



FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS

**DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS Y COMPUTACIÓN**

**“IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE
INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO.
APLICACIÓN AL BARCO TREASURE OF GALAPAGOS”.**

AUTOR:

ALEX NIETO

Directora:

Ing. Suyana Arcos

Agradecimientos

A mis familiares.

A mi hermano Juan Nieto el cual supo darme la idea principal y sobre todo ayudarme en los requerimientos y lo principal a idea de tesis. A mis padres Ing. Juan Nieto e Ing. María Terán siendo el sustento económico en mis estudios, para culminar mi carrera universitaria.

A mi pareja

A Cristina Rivera que fue gran apoyo incondicional durante la elaboración de la tesis y gran fuente de ayuda para redacción en la misma.

A mis profesores

Los principales en el desarrollo Ing. Suyana Arcos, Ing. Alfredo Calderón e Ing. Oswaldo Espinoza los cuales fueron los principales asesores para el desarrollo de esta tesis.

A mis compañeros de trabajo

Que fueron un apoyo más en la colaboración de ideas e innovaciones que sirvieron para optimizar mi tesis y no perder tiempo.

A mis amigos.

Los cuales entre varias vivencias experimentadas entre todos se logró mantener los niveles de estrés bajos en los momentos críticos del desarrollo.

Tabla de contenidos

AGRADECIMIENTOS	I
TABLA DE CONTENIDOS	1
ÍNDICE DE ILUSTRACIONES.....	7
ÍNDICE DE TABLAS.....	11
RESUMEN	1
INTRODUCCIÓN	3
1. CAPÍTULO: ANTECEDENTES.....	4
1.1. <i>Objetivos</i>	4
1.2. <i>Perfil de Usuarios</i>	4
1.3. <i>Funciones Desarrolladas en el Área</i>	5
1.4. <i>Procesos que se Llevan a Cabo</i>	5
1.5. <i>Problemas del Proceso</i>	5
2. CAPÍTULO: MARCO TEORICO.	6
2.1. <i>Técnicas de Desarrollo de Aplicaciones</i>	6
2.1.1. <i>Diseño Ágil de Aplicaciones (Extreme Programming)</i>	6
2.1.2. <i>Rad (Diseño Rápido de Aplicaciones)</i>	7
2.1.2.1. <i>Planificación de los Requisitos</i>	8
2.1.2.2. <i>Diseño</i>	8
2.1.2.3. <i>Construcción</i> :.....	8
2.1.2.4. <i>Implementación</i> :	8
2.1.2.5. <i>Modelado de Gestión</i> :	8
2.1.2.6. <i>Modelado de Datos</i> :	8
2.1.2.7. <i>Modelado del Proceso</i> :.....	9
2.1.2.8. <i>Generación de Aplicaciones</i>	9
2.1.2.9. <i>Entorno RAD Sobre Navegadores Web</i>	9
2.1.3. <i>JAD (Diseño de Aplicaciones por Juntas)</i>	10
2.1.3.1. <i>¿Qué es JAD?</i>	11
2.1.3.2. <i>Instalaciones Para el Desarrollo de JAD</i>	11
2.1.3.3. <i>Realizar la Sesión JAD</i>	11
2.1.3.4. <i>Preparación</i>	11
2.1.3.5. <i>El Período de Sesiones</i>	12
2.1.3.6. <i>Beneficios Principales</i>	12
2.2. <i>Análisis de Herramientas de Desarrollo</i>	13
2.2.1. <i>Sistema de Gestión de Bases de Datos</i>	13
2.2.1.1. <i>Objetivos</i>	13
2.2.1.2. <i>Ventajas</i>	14
2.2.1.3. <i>MySQL</i>	14
2.2.2. <i>Lenguajes De Programación</i>	15
2.2.2.1. <i>.NET</i>	15
2.2.2.2. <i>PHP</i>	16
2.2.2.3. <i>JSP</i>	16
2.2.3. <i>Herramientas Case</i>	17
2.2.4. <i>Servidores de Aplicaciones (Servidor Web)</i>	19
2.3. <i>Arquitectura De Aplicaciones</i>	21
2.3.1. <i>Cliente/Servidor</i>	21
2.3.1.1. <i>Cliente</i>	22

2.3.1.2.	Servidor	22
2.3.1.3.	Características	23
2.3.1.4.	Ventajas	24
2.3.1.5.	Desventajas	25
2.3.2.	Por Capas	25
2.3.2.1.	Capa de presentación	27
2.3.2.2.	Capa de negocio	28
2.3.2.3.	Capa de datos	28
3.	CAPÍTULO: CASO DE ESTUDIO	29
3.1.	<i>Proceso Pedido y Entrega de Insumos y Víveres en el Barco Treasure Of Galápagos.....</i>	<i>29</i>
3.1.1.	Objetivos	29
3.1.1.1.	General	29
3.1.1.2.	Específicos	29
3.1.2.	Misión	29
3.1.3.	Visión.....	29
3.1.4.	Cuadro Jerárquico	30
3.1.5.	Mapa de Procesos del Barco	31
3.1.6.	Principales Actividades del Proceso de Pedido y Entrega de Insumos y Víveres.....	31
3.1.7.	Selección de las Herramientas Para la Construcción De la Aplicación.	32
3.1.7.1.	Técnica De Desarrollo De Aplicaciones.....	32
3.1.7.2.	Arquitectura De Aplicación.....	32
3.1.7.3.	Lenguaje De Programación.....	33
3.1.7.4.	Sistema Gestor de Base de Datos	36
3.1.7.5.	Servidor De Aplicación.....	38
3.1.7.6.	IDE Para Desarrollo.....	39
3.2.	<i>Cuadro Resumen de Selección de Tecnología.....</i>	<i>39</i>
4.	CAPÍTULO: LEVANTAMIENTO DEL PROCESO	40
4.1.	<i>Mapa de Procesos.....</i>	<i>40</i>
4.1.1.	Descripción.....	40
4.1.2.	Actores	40
4.1.3.	Procesos Involucrados.....	40
4.1.4.	Cadenas de Valor	41
4.1.4.1.	Procesos Centrales	41
4.1.4.2.	Procesos de Soporte.....	41
4.1.5.	Mapa de procesos.....	42
4.1.6.	Gráficos de Soporte.....	43
4.1.6.1.	Gráfico Proceso Administrar Recursos	43
4.1.6.2.	Gráfico Proceso Administrar Pedido.....	44
4.1.6.1.	Gráfico Proceso Administrar Entrega	45
4.2.	<i>SIPOC</i>	<i>46</i>
5.	CAPÍTULO: CONSTRUCCIÓN DEL PRODUCTO	48
5.1.	<i>Inicio</i>	<i>48</i>
5.1.1.	Análisis De Requerimientos	48
5.1.2.	Identificación De Actores	48
5.1.3.	Especificaciones De Requerimientos.....	49
5.1.3.1.	Funcionales.....	49
5.1.3.2.	No Funcionales	50
5.1.4.	Arquitectura	51
5.1.5.	Diseño	52
5.1.5.1.	Diseño Entidad/Relación (E/R)	52
5.1.5.2.	Diagramas UML	53
5.2.	<i>Elaboración.....</i>	<i>71</i>
5.2.1.	Prototipo Ejecutable	71

5.2.1.1.	Ventana Inicial (Página de Inicio).....	71
5.2.1.2.	Ventana Principal de Administrador (Página Principal Administrador).....	72
5.2.1.3.	Ventana Principal de Solicitante (Página Principal Solicitante).	73
5.2.1.4.	Ventana Principal de Aprobador (Página Principal Aprobador).	74
5.2.1.5.	Ventana Principal de Controlador (Página Principal Controlador).	75
5.2.1.6.	Ventana Principal de Proveedor (Página Principal Proveedor).....	76
5.2.1.7.	Ventana Administración Usuarios (Pagina Administración Usuarios).	77
5.2.1.8.	Ventana Administración de Departamentos (Pagina de Administración de Departamentos). ...	78
5.2.1.9.	Ventana Administración de Productos (Pagina de Administración de Productos).....	79
5.2.1.10.	Ventana Administración de Pedidos (Pagina de Administración de Pedidos).	80
5.2.1.11.	Ventana Administración de Estado Pedidos (Pagina de Administración Estado de Pedidos).....	81
5.2.1.12.	Ventana Administración de Despachos (Pagina de Administración de Despachos).....	82
5.2.1.13.	Ventana Administración de Estado de Despachos (Pagina de Administración de Estado de Despachos).	83
5.2.1.14.	Ventana Administración de Bodega (Pagina de Administración de Bodega).	84
5.2.2.	Lista De Riesgos.....	85
5.2.2.1.	Descripción.	85
5.2.2.2.	Riesgos.....	86
5.3.	Construcción.	91
5.3.1.	Implementación y Pruebas.....	91
5.3.1.1.	Ventana Inicial Login:	91
5.3.1.2.	Administrador:.....	93
5.3.1.3.	Solicitante:.....	135
5.3.1.4.	Aprobador:	145
5.3.1.5.	Proveedor:.....	150
5.3.1.6.	Controlador:	162
5.4.	Transición.	167
5.4.1.	Seguridades.....	167
5.4.2.	Costos.....	167
5.4.3.	Mantenimiento.	169
6.	CAPÍTULO: CONCLUSIONES Y RECOMENDACIONES.	170
6.1.	<i>Conclusiones.</i>	170
6.2.	<i>Recomendaciones.</i>	172
	BIBLIOGRAFÍA.	173
	TRABAJOS CITADOS.	175
	ANEXO A	176
1.	MANUAL DE USUARIO	176
1.1.	<i>Módulo Usuarios</i>	176
1.1.1.	Ingreso Nuevo Usuario.....	176
1.1.2.	Reporte de Usuarios en el Sistema.	178
1.1.3.	Edición de Usuarios.	179
1.2.	<i>Módulo Departamentos.</i>	180
1.2.1.	Ingreso Nuevo Departamento.....	180
1.2.1.	Reporte de Departamentos en el Sistema.	181
1.2.2.	Edición de Departamentos.....	182
1.3.	<i>Módulo Productos</i>	183
1.3.1.	Ingreso Nuevo Producto	183
1.3.2.	Reporte de Productos en el Sistema.	184
1.3.3.	Edición de Productos.....	185
1.4.	<i>Módulo Pedidos.</i>	186
1.4.1.	Nuevo Pedido	186

1.4.2.	Reporte de Pedidos no Aprobados.	187
1.4.3.	Edición de Pedido.....	188
1.4.4.	Selección de Pedido Para Aprobación.....	189
1.5.	<i>Módulo Despachos.</i>	191
1.5.1.	Nuevo Despacho.	191
1.5.2.	Reporte de Despachos no Comprados.....	192
1.5.3.	Edición de Despacho.	193
1.5.4.	Selección de Despacho Para Procesar.....	194
1.6.	<i>Módulo de Manejo de Bodega.</i>	196
1.6.1.	Ventana de Administrador.....	196
1.6.2.	Ventana de Solicitante	197
ANEXO B		198
1. MANUAL TÉCNICO		198
1.1.	<i>Login</i>	198
1.1.1.	Contenedor	198
1.1.2.	Controlador	200
1.1.3.	Contenedor de Vista	201
1.1.4.	Controlador de Vista	201
1.1.5.	Archivo JS	202
1.2.	<i>Perfil Administrador</i>	203
1.2.1.	Contenedor.	203
1.2.2.	Contenedor de Vista.	205
1.2.3.	Controlador Bodega.	206
1.2.4.	Controlador Departamento.	207
1.2.5.	Controlador Despacho.	209
1.2.6.	Controlador Main.....	214
1.2.7.	Controlador Pedido.....	215
1.2.8.	Controlador Producto.	219
1.2.9.	Controlador Usuario.....	221
1.2.10.	Helper.....	223
1.2.10.1.	Menú.	223
1.2.11.	Vistas de Bodega.	224
1.2.11.1.	Select.	224
1.2.12.	Vistas de Departamento.....	225
1.2.12.1.	Create.	225
1.2.12.2.	Select.	226
1.2.13.	Vistas de Despacho.	227
1.2.13.1.	Create.	227
1.2.13.2.	Procesar.....	228
1.2.13.3.	Revisar.	229
1.2.13.4.	Select.	230
1.2.14.	Vistas de Main.....	231
1.2.14.1.	Main.	231
1.2.15.	Vistas de Pedido.	232
1.2.15.1.	Aprobar.	232
1.2.15.2.	Create.	233
1.2.15.3.	Revisar.....	234
1.2.15.4.	Select.	235
1.2.16.	Vistas de Producto.	236
1.2.16.1.	Create.	236
1.2.16.2.	Select.	237
1.2.17.	Vistas Usuario.....	238

1.2.17.1.	Create	238
1.2.17.2.	Select	240
1.2.18.	Archivos JS de Bodega	241
1.2.18.1.	Select	241
1.2.1.	Archivos JS de Departamento.	243
1.2.1.1.	Create	243
1.2.1.2.	Select	244
1.2.1.	Archivos JS de Despacho.	245
1.2.1.1.	Create	245
1.2.1.2.	Procesar.....	247
1.2.1.3.	Revisar.....	248
1.2.1.4.	Select	249
1.2.2.	Archivos JS de Pedido.....	250
1.2.2.1.	Aprobar.	250
1.2.2.2.	Create	251
1.2.2.3.	Revisar.....	252
1.2.2.4.	Select	252
1.2.3.	Archivos JS de Producto	253
1.2.3.1.	Create	253
1.2.3.2.	Select	254
1.2.4.	Archivos JS Usuario.	255
1.2.4.1.	Create	255
1.2.4.2.	Select	258
1.3.	<i>Perfil Control.</i>	259
1.3.1.	Contenedor.	259
1.3.2.	Contenedor de Vista	261
1.3.3.	Controlador de Despacho.	262
1.3.4.	Controlador Main.....	265
1.3.5.	Helper.....	266
1.3.5.1.	Menú.	266
1.3.6.	Vistas de Despacho.	267
1.3.6.1.	Procesar.....	267
1.3.6.2.	Revisar.....	268
1.3.7.	Vistas de Main.....	269
1.3.8.	Archivos JS de Despacho.....	270
1.3.8.1.	Procesar.....	270
1.3.8.2.	Revisar.....	271
1.4.	<i>Perfil Aprobador.</i>	272
1.4.1.	Contenedor.	272
1.4.2.	Contenedor de Vista	274
1.4.3.	Controlador de Pedido.....	275
1.4.4.	Controlador de Main.....	279
1.4.5.	Helper.....	280
1.4.5.1.	Menú.	280
1.4.6.	Vistas de Pedido.....	281
1.4.6.1.	Aprobar.	281
1.4.6.2.	Revisar.....	282
1.4.7.	Vistas de Main.....	283
1.4.8.	Archivos JS de Pedido.....	284
1.4.8.1.	Aprobar.	284
1.4.8.2.	Revisar.....	285
1.5.	<i>Perfil Proveedor.</i>	286
1.5.1.	Contenedor.	286
1.5.2.	Contenedor Vista.....	288

1.5.3.	Controlador Despacho.....	289
1.5.4.	Controlador Main.....	294
1.5.5.	Helper.....	295
1.5.5.1.	Menú.....	295
1.5.6.	Vistas de Despacho.....	296
1.5.6.1.	Create.....	296
1.5.6.2.	Procesar.....	297
1.5.6.3.	Revisar.....	298
1.5.6.4.	Select.....	299
1.5.7.	Vistas de Main.....	300
1.5.8.	Archivos JS de Despacho.....	301
1.5.8.1.	Create.....	301
1.5.8.2.	Procesar.....	303
1.5.8.3.	Revisar.....	304
1.5.8.4.	Select.....	305
1.6.	Perfil Solicitante.....	306
1.6.1.	Contenedor.....	306
1.6.2.	Contenedor de Vista.....	308
1.6.3.	Controlador de Bodega.....	309
1.6.4.	Controlador de Pedido.....	311
1.6.5.	Controlador de Main.....	314
1.6.6.	Helper.....	315
1.6.6.1.	Menú.....	315
1.6.7.	Vistas de Bodega.....	316
1.6.7.1.	Select.....	316
1.6.8.	Vistas de Pedido.....	317
1.6.8.1.	Create.....	317
1.6.8.2.	Select.....	319
1.6.9.	Vistas de Main.....	320
1.6.10.	Archivos JS de Bodega.....	321
1.6.10.1.	Select.....	321
1.6.11.	Archivos JS de Pedido.....	322
1.6.11.1.	Create.....	322
1.6.11.2.	Select.....	324
1.7.	Modelos.....	325
1.7.1.	Departamento.....	325
1.7.2.	Despacho.....	325
1.7.3.	Pedido.....	325
1.7.4.	Pedido Despacho por Producto.....	325
1.7.5.	Perfil.....	326
1.7.6.	Producto.....	326
1.7.7.	Producto por Bodega.....	326
1.7.8.	Solicitante.....	327
1.8.	Mensajes Flash.....	328
1.9.	Archivo SQL de Instalación.....	329

Índice de ilustraciones

ILUSTRACIÓN 1: CAPAS HERRAMIENTAS CASE. REFERENCIA: HTTP://OCW.UA.ES/INGENIERIA-ARQUITECTURA/PROGRAMACION-EN-INTERNET/03C-APLICACIONESWEB.PDF	18
ILUSTRACIÓN 2: OPCIONES DE INTEGRACION. REFERENCIA: HTTP://OCW.UA.ES/INGENIERIA-ARQUITECTURA/PROGRAMACION-EN-INTERNET/03C-APLICACIONESWEB.PDF	19
ILUSTRACIÓN 3: MODELO CLIENTE/SERVIDOR. REFERENCIA: HTTP://WWW.KALIPEDIA.COM/TECNOLOGIA/TEMA/COMUNICACIONES/GRAFICOS-MODELO-CLIENTE-SERVIDOR.HTML?X1=20070821KLPINGINF_30.EES&X=20070821KLPINGINF_71.KES	21
ILUSTRACIÓN 4: MODELO 3 CAPAS REFERENCIA: HTTP://WWW.DESARROLLOWEB.COM/ARTICULOS/ARQUITECTURA-CLIENTE-SERVIDOR.HTML	26
ILUSTRACIÓN 5: MODELO 4 CAPAS WEB. REFERENCIA: HTTP://WWW.DESARROLLOWEB.COM/ARTICULOS/ARQUITECTURA-CLIENTE-SERVIDOR.HTML	27
ILUSTRACIÓN 6: MODELO 3 CAPAS WEB. REFERENCIA: HTTP://WWW.DESARROLLOWEB.COM/ARTICULOS/ARQUITECTURA-CLIENTE-SERVIDOR.HTML	27
ILUSTRACIÓN 7: CUADRO JERÁRQUICO. AUTOR: JUAN FERNANDO NIETO.....	30
ILUSTRACIÓN 8: MAPA DE PROCESOS. AUTOR: JUAN FERNANDO NIETO Y ALEX NIETO.	31
ILUSTRACIÓN 9: COMPARACIÓN GESTORES BASES DE DATOS. REFERENCIA: HTTP://ALBERTO121888.FILES.WORDPRESS.COM/2010/08/ENSAYO-POSTGRESQL-VS-MYSQL-VS-ORACLE.PD	38
ILUSTRACIÓN 11: CADENA DE VALOR. AUTOR: ALEX NIETO.	41
ILUSTRACIÓN 12: MAPA DE PROCESOS. AUTOR: ALEX NIETO.	43
ILUSTRACIÓN 13: PROCESO ADMINISTRAR RECURSOS. AUTOR: ALEX NIETO.	43
ILUSTRACIÓN 14: PROCESO ADMINISTRAR PEDIDO. AUTOR: ALEX NIETO.	44
ILUSTRACIÓN 15: PROCESO ADMINISTRAR ENTREGA. AUTOR: ALEX NIETO.....	45
ILUSTRACIÓN 16: SIPOC. AUTOR: ALEX NIETO.....	46
ILUSTRACIÓN 17: MODELO SIPOC PROCESOS. AUTOR: ALEX NIETO.....	47
ILUSTRACIÓN 18: ACTORES. AUTOR: ALEX NIETO.	49
ILUSTRACIÓN 19: ARQUITECTURA DEL PRODUCTO. AUTOR: ALEX NIETO	51
ILUSTRACIÓN 20: DISEÑO E/R. AUTOR: ALEX NIETO.....	52
ILUSTRACIÓN 21: DIAGRAMA DP. AUTOR: ALEX NIETO.....	53
ILUSTRACIÓN 22: DIAGRAMA MD. AUTOR: ALEX NIETO.....	54
ILUSTRACIÓN 23: DIAGRAMA GUI. AUTOR: ALEX NIETO.	55
ILUSTRACIÓN 24: DIAGRAMA DE DESPLIEGUE. AUTOR: ALEX NIETO.....	56
ILUSTRACIÓN 25: INGRESO DE PRODUCTOS. AUTOR: ALEX NIETO.	57
ILUSTRACIÓN 26: MODIFICAR PRODUCTO. AUTOR: ALEX NIETO.....	58
ILUSTRACIÓN 27: ELIMINAR PRODUCTO. AUTOR: ALEX NIETO.	59
ILUSTRACIÓN 28: CONSULTA GENERAL PRODUCTOS. AUTOR: ALEX NIETO.	60
ILUSTRACIÓN 29: CONSULTA PARÁMETROS PRODUCTOS. AUTOR: ALEX NIETO.	60
ILUSTRACIÓN 30: INGRESAR PEDIDO. AUTOR: ALEX NIETO.	61
ILUSTRACIÓN 31: MODIFICAR PEDIDO. AUTOR: ALEX NIETO.	62
ILUSTRACIÓN 32: ELIMINAR PEDIDO. AUTOR: ALEX NIETO.	63
ILUSTRACIÓN 33: CONSULTA GENERAL PEDIDOS. AUTOR: ALEX NIETO.	63
ILUSTRACIÓN 34: CONSULTA POR PARÁMETRO PEDIDOS. AUTOR: ALEX NIETO.	64
ILUSTRACIÓN 35: MODIFICAR ESTADO PEDIDOS. AUTOR: ALEX NIETO.	65
ILUSTRACIÓN 36: INGRESAR DESPACHO. AUTOR: ALEX NIETO.....	66
ILUSTRACIÓN 37: MODIFICAR DESPACHO. AUTOR: ALEX NIETO.	67
ILUSTRACIÓN 38: ELIMINAR DESPACHO. AUTOR: ALEX NIETO.....	68
ILUSTRACIÓN 39: CONSULTA GENERAL DESPACHOS. AUTOR: ALEX NIETO.....	68
ILUSTRACIÓN 40: CONSULTA POR PARÁMETRO DESPACHOS. AUTOR: ALEX NIETO.....	69

ILUSTRACIÓN 41: MODIFICAR ESTADO DESPACHO. AUTOR: ALEX NIETO.....	70
ILUSTRACIÓN 42: VENTANA INICIAL (PROTOTIPO). AUTOR: ALEX NIETO	71
ILUSTRACIÓN 43: VENTANA PRINCIPAL DE ADMINISTRADOR (PROTOTIPO). AUTOR: ALEX NIETO.....	72
ILUSTRACIÓN 44: VENTANA PRINCIPAL SOLICITANTE (PROTOTIPO). AUTOR: ALEX NIETO	73
ILUSTRACIÓN 45: VENTANA PRINCIPAL DE APROBADOR (PROTOTIPO). AUTOR: ALEX NIETO.....	74
ILUSTRACIÓN 46: VENTANA PRINCIPAL DE CONTROLADOR (PROTOTIPO). AUTOR: ALEX NIETO	75
ILUSTRACIÓN 47: VENTANA PRINCIPAL DE PROVEEDOR (PROTOTIPO). AUTOR: ALEX NIETO	76
ILUSTRACIÓN 48: VENTANA ADMINISTRACIÓN USUARIOS (PROTOTIPO). AUTOR: ALEX NIETO	77
ILUSTRACIÓN 49: VENTANA ADMINISTRACIÓN DE DEPARTAMENTOS (PROTOTIPO). AUTOR: ALEX NIETO	78
ILUSTRACIÓN 50: VENTANA ADMINISTRACIÓN DE PRODUCTOS (PROTOTIPO). AUTOR: ALEX NIETO	79
ILUSTRACIÓN 51: VENTANA ADMINISTRACIÓN DE PEDIDOS (PROTOTIPO). AUTOR: ALEX NIETO	80
ILUSTRACIÓN 52: VENTANA ADMINISTRACIÓN DE ESTADO PEDIDOS (PROTOTIPO). AUTOR: ALEX NIETO.....	81
ILUSTRACIÓN 53: VENTANA ADMINISTRACIÓN DE DESPACHOS (PROTOTIPO). AUTOR: ALEX NIETO.....	82
ILUSTRACIÓN 54: VENTANA ADMINISTRACIÓN DE ESTADO DE DESPACHOS (PROTOTIPO). AUTOR: ALEX NIETO	83
ILUSTRACIÓN 55: VENTANA ADMINISTRACIÓN DE BODEGA (PROTOTIPO). AUTOR: ALEX NIETO	84
ILUSTRACIÓN 56: VENTANA INICIAL LOGIN. AUTOR: ALEX NIETO.....	91
ILUSTRACIÓN 57: VENTANA INICIAL LOGIN (VALIDACIONES USUARIO). AUTOR: ALEX NIETO.....	92
ILUSTRACIÓN 58: VENTANA PRINCIPAL (ADMINISTRADOR). AUTOR: ALEX NIETO.....	93
ILUSTRACIÓN 59: VENTANA NUEVO USUARIO (ADMINISTRADOR). AUTOR: ALEX NIETO.	94
ILUSTRACIÓN 60: INGRESO NUEVO USUARIO (ADMINISTRADOR). AUTOR: ALEX NIETO.	94
ILUSTRACIÓN 61: VENTANA NUEVO USUARIO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	95
ILUSTRACIÓN 62: VENTANA SELECCIÓN USUARIO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	97
ILUSTRACIÓN 63: VENTANA EDICIÓN USUARIO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	98
ILUSTRACIÓN 64: INGRESO EDICIÓN USUARIO (ADMINISTRADOR). AUTOR: ALEX NIETO.	98
ILUSTRACIÓN 65: VENTANA EDICIÓN USUARIO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	99
ILUSTRACIÓN 66: VENTANA NUEVO DEPARTAMENTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	101
ILUSTRACIÓN 67: INGRESO NUEVO DEPARTAMENTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	101
ILUSTRACIÓN 68: VENTANA NUEVO DEPARTAMENTO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	102
ILUSTRACIÓN 69: SELECCIÓN DEPARTAMENTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	103
ILUSTRACIÓN 70: EDICIÓN DEPARTAMENTO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	104
ILUSTRACIÓN 71: INGRESO EDICIÓN DEPARTAMENTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	104
ILUSTRACIÓN 72: EDICIÓN DEPARTAMENTO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	105
ILUSTRACIÓN 73: NUEVO PRODUCTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	106
ILUSTRACIÓN 74: INGRESO NUEVO PRODUCTO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	106
ILUSTRACIÓN 75: NUEVO PRODUCTO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	107
ILUSTRACIÓN 76: SELECCIÓN PRODUCTO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	108
ILUSTRACIÓN 77: EDICIÓN PRODUCTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	109
ILUSTRACIÓN 78: INGRESO EDICIÓN PRODUCTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	109
ILUSTRACIÓN 79: EDICIÓN PRODUCTO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	110
ILUSTRACIÓN 80: NUEVO PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	111
ILUSTRACIÓN 81: INGRESO 1 NUEVO PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	111
ILUSTRACIÓN 82: INGRESO 2 NUEVO PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	112
ILUSTRACIÓN 83: NUEVO PEDIDO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	113
ILUSTRACIÓN 84: SELECCIÓN PEDIDO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	114
ILUSTRACIÓN 85: EDICIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	115
ILUSTRACIÓN 86: INGRESO 1 EDICIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	115
ILUSTRACIÓN 87: INGRESO 2 EDICIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	116
ILUSTRACIÓN 88: EDICIÓN PEDIDO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	117

ILUSTRACIÓN 89: APROBACIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	118
ILUSTRACIÓN 90: REVISIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	119
ILUSTRACIÓN 91: INGRESO 1 REVISIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	119
ILUSTRACIÓN 92: INGRESO 2 REVISIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	120
ILUSTRACIÓN 93: REVISIÓN PEDIDO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	121
ILUSTRACIÓN 94: NUEVO DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	122
ILUSTRACIÓN 95: INGRESO 1 NUEVO DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	122
ILUSTRACIÓN 96: INGRESO 2 NUEVO DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	123
ILUSTRACIÓN 97: NUEVO DESPACHO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	124
ILUSTRACIÓN 98: SELECCIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	125
ILUSTRACIÓN 99: EDICIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	126
ILUSTRACIÓN 100: INGRESO 1 EDICIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	126
ILUSTRACIÓN 101: INGRESO 2 EDICIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	127
ILUSTRACIÓN 102: EDICIÓN DESPACHO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	128
ILUSTRACIÓN 103: ESTADO DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	129
ILUSTRACIÓN 104: REVISIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	130
ILUSTRACIÓN 105: INGRESO 1 REVISIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	130
ILUSTRACIÓN 106: INGRESO 2 REVISIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	131
ILUSTRACIÓN 107: REVISIÓN DESPACHO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	132
ILUSTRACIÓN 108: BODEGA (ADMINISTRADOR). AUTOR: ALEX NIETO.	133
ILUSTRACIÓN 109: INGRESO BODEGA (ADMINISTRADOR). AUTOR: ALEX NIETO.	133
ILUSTRACIÓN 110: BODEGA VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	134
ILUSTRACIÓN 111: VENTANA PRINCIPAL (SOLICITANTE). AUTOR: ALEX NIETO.	135
ILUSTRACIÓN 112: NUEVO PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	136
ILUSTRACIÓN 113: INGRESO 1 NUEVO PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	136
ILUSTRACIÓN 114: INGRESO 2 NUEVO PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	137
ILUSTRACIÓN 115: NUEVO PEDIDO VALIDACIONES (SOLICITANTE). AUTOR: ALEX NIETO.	138
ILUSTRACIÓN 116: SELECCIÓN PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	139
ILUSTRACIÓN 117: EDICIÓN PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	140
ILUSTRACIÓN 118: INGRESO 1 EDICIÓN PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	140
ILUSTRACIÓN 119: INGRESO 2 EDICIÓN PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	141
ILUSTRACIÓN 120: EDICIÓN PEDIDO VALIDACIONES (SOLICITANTE). AUTOR: ALEX NIETO.	142
ILUSTRACIÓN 121: BODEGA (SOLICITANTE). AUTOR: ALEX NIETO.	143
ILUSTRACIÓN 122: INGRESO BODEGA (SOLICITANTE). AUTOR: ALEX NIETO.	143
ILUSTRACIÓN 123: BODEGA VALIDACIONES (SOLICITANTE). AUTOR: ALEX NIETO.	144
ILUSTRACIÓN 124: VENTANA PRINCIPAL (APROBADOR). AUTOR: ALEX NIETO.	145
ILUSTRACIÓN 125: APROBACIÓN PEDIDO (APROBADOR). AUTOR: ALEX NIETO.	146
ILUSTRACIÓN 126: REVISIÓN PEDIDO (APROBADOR). AUTOR: ALEX NIETO.	147
ILUSTRACIÓN 127: INGRESO 1 REVISIÓN PEDIDO (APROBADOR). AUTOR: ALEX NIETO.	147
ILUSTRACIÓN 128: INGRESO 2 REVISIÓN PEDIDO (APROBADOR). AUTOR: ALEX NIETO.	148
ILUSTRACIÓN 129: REVISIÓN PEDIDO VALIDACIONES (APROBADOR). AUTOR: ALEX NIETO.	149
ILUSTRACIÓN 130: VENTANA PRINCIPAL (PROVEEDOR). AUTOR: ALEX NIETO.	150
ILUSTRACIÓN 131: NUEVO DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	151
ILUSTRACIÓN 132: INGRESO 1 NUEVO DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	151
ILUSTRACIÓN 133: INGRESO 2 NUEVO DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	152
ILUSTRACIÓN 134: NUEVO DESPACHO VALIDACIONES (PROVEEDOR). AUTOR: ALEX NIETO.	153
ILUSTRACIÓN 135: SELECCIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	154
ILUSTRACIÓN 136: EDICIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	155

ILUSTRACIÓN 137: INGRESO 1 EDICIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.....	155
ILUSTRACIÓN 138: INGRESO 2 EDICIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.....	156
ILUSTRACIÓN 139: EDICIÓN DESPACHO VALIDACIONES (PROVEEDOR). AUTOR: ALEX NIETO.	157
ILUSTRACIÓN 140: ESTADO DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	158
ILUSTRACIÓN 141: REVISIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.....	159
ILUSTRACIÓN 142: INGRESO 1 REVISIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	159
ILUSTRACIÓN 143: INGRESO 2 REVISIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	160
ILUSTRACIÓN 144: REVISIÓN DESPACHO VALIDACIONES (PROVEEDOR). AUTOR: ALEX NIETO.	161
ILUSTRACIÓN 145: VENTANA PRINCIPAL (CONTROLADOR). AUTOR: ALEX NIETO.....	162
ILUSTRACIÓN 146: ESTADO DESPACHO (CONTROLADOR). AUTOR: ALEX NIETO.	163
ILUSTRACIÓN 147: REVISIÓN DESPACHO (CONTROLADOR). AUTOR: ALEX NIETO.....	164
ILUSTRACIÓN 148: INGRESO 1 DESPACHO (CONTROLADOR). AUTOR: ALEX NIETO.....	164
ILUSTRACIÓN 149: INGRESO 2 REVISIÓN DESPACHO (CONTROLADOR). AUTOR: ALEX NIETO.	165
ILUSTRACIÓN 150: REVISIÓN DESPACHO VALIDACIONES (CONTROLADOR). AUTOR: ALEX NIETO.	166

Índice de tablas

TABLA 1: CUADRO COMPARATIVO KAHANA VS SMARTY AUTOR: ALEX NIETO	33
TABLA 2: CUADRO COMPARATIVO PHP Y JSP. AUTOR: ALEX NIETO	34
TABLA 3: CUADRO COMPARATIVO PHP Y .NET. AUTOR: ALEX NIETO	35
TABLA 4: CUADRO COMPARATIVO MYSQL VS POSTGRESQL. AUTOR: ALEX NIETO	37
TABLA 5: CUADRO RESUMEN SELECCIÓN TECNOLOGÍA. AUTOR: ALEX NIETO.....	39
TABLA 6: TABLA REQUERIMIENTOS CLIENTE. AUTOR: ALEX NIETO.	50
TABLA 7: TABLA REQUERIMIENTOS SERVIDOR. AUTOR: ALEX NIETO.	50
TABLA 8: INGRESO LOGIN. AUTOR: ALEX NIETO.	91
TABLA 9: INGRESOS INICIAL LOGIN (VALIDACIONES USUARIO). AUTOR: ALEX NIETO.	92
TABLA 10: INGRESO NUEVO USUARIO (ADMINISTRADOR). AUTOR: ALEX NIETO.	95
TABLA 11: INGRESO NUEVO USUARIO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	96
TABLA 12: INGRESO EDICIÓN USUARIO (ADMINISTRADOR). AUTOR: ALEX NIETO.	99
TABLA 13: INGRESO EDICIÓN USUARIO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	100
TABLA 14: INGRESO NUEVO DEPARTAMENTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	101
TABLA 15: INGRESO NUEVO DEPARTAMENTO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	102
TABLA 16: INGRESO EDICIÓN DEPARTAMENTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	104
TABLA 17: EDICIÓN DEPARTAMENTO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	105
TABLA 18: INGRESO NUEVO PRODUCTO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	106
TABLA 19: NUEVO PRODUCTO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	107
TABLA 20: INGRESO EDICIÓN PRODUCTO (ADMINISTRADOR). AUTOR: ALEX NIETO.	109
TABLA 21: EDICIÓN PRODUCTO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	110
TABLA 22: INGRESO NUEVO PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	112
TABLA 23: NUEVO PEDIDO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	113
TABLA 24: INGRESO EDICIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.	116
TABLA 25: EDICIÓN PEDIDO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	117
TABLA 26: INGRESO REVISIÓN PEDIDO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	120
TABLA 27: REVISIÓN PEDIDO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	121
TABLA 28: INGRESO NUEVO DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	123
TABLA 29: NUEVO DESPACHO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	124
TABLA 30: INGRESO EDICIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.....	127
TABLA 31: EDICIÓN DESPACHO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	128
TABLA 32: INGRESO REVISIÓN DESPACHO (ADMINISTRADOR). AUTOR: ALEX NIETO.	131
TABLA 33: REVISIÓN DESPACHO VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.....	132
TABLA 34: INGRESO BODEGA (ADMINISTRADOR). AUTOR: ALEX NIETO.	133
TABLA 35: BODEGA VALIDACIONES (ADMINISTRADOR). AUTOR: ALEX NIETO.	134
TABLA 36: INGRESO NUEVO PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	137
TABLA 37: NUEVO PEDIDO VALIDACIONES (SOLICITANTE). AUTOR: ALEX NIETO.....	138
TABLA 38: INGRESO EDICIÓN PEDIDO (SOLICITANTE). AUTOR: ALEX NIETO.	141
TABLA 39: EDICIÓN PEDIDO VALIDACIONES (SOLICITANTE). AUTOR: ALEX NIETO.	142
TABLA 40: INGRESO BODEGA (SOLICITANTE). AUTOR: ALEX NIETO.	143
TABLA 41: BODEGA VALIDACIONES (SOLICITANTE). AUTOR: ALEX NIETO.	144
TABLA 42: INGRESO REVISIÓN PEDIDO (APROBADOR). AUTOR: ALEX NIETO.....	148
TABLA 43: REVISIÓN PEDIDO VALIDACIONES (APROBADOR). AUTOR: ALEX NIETO.....	149
TABLA 44: INGRESO NUEVO DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.....	152
TABLA 45: NUEVO DESPACHO VALIDACIONES (PROVEEDOR). AUTOR: ALEX NIETO.....	153
TABLA 46: INGRESO EDICIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.	156
TABLA 47: EDICIÓN DESPACHO VALIDACIONES (PROVEEDOR). AUTOR: ALEX NIETO.	157

TABLA 48: INGRESO REVISIÓN DESPACHO (PROVEEDOR). AUTOR: ALEX NIETO.....	160
TABLA 49: REVISIÓN DESPACHO VALIDACIONES (PROVEEDOR). AUTOR: ALEX NIETO.	161
TABLA 50: INGRESO REVISIÓN DESPACHO (CONTROLADOR). AUTOR: ALEX NIETO.....	165
TABLA 51: REVISIÓN DESPACHO VALIDACIONES (CONTROLADOR). AUTOR: ALEX NIETO.	166
TABLA 52: COSTOS INICIALES. AUTOR: ALEX NIETO.	167
TABLA 53: COSTOS ADMINISTRATIVOS. AUTOR: ALEX NIETO.....	167
TABLA 54: COSTOS OPERACIÓN. AUTOR: ALEX NIETO.....	168
TABLA 55: COSTOS SOPORTE TÉCNICO. AUTOR: ALEX NIETO.....	168
TABLA 56: COSTOS TOTALES. AUTOR: ALEX NIETO.	168

Resumen

En esta disertación de grado se presenta principalmente el desarrollo del sistema para control de insumos y víveres de un barco turístico aplicado al Treasure Of Galápagos; en los primeros capítulos se explica la problemática del tema en general y a su vez el estudio de las técnicas para la construcción y desarrollo de sistemas web.

Dentro del primer capítulo se explica, de manera general, cómo se lleva a cabo en la actualidad el manejo del proceso; con sus respectivos perfiles de usuario y funciones desarrolladas, junto con un pequeño resumen de los problemas que se presentan y los procesos que se llevan a cabo.

En el segundo capítulo se puede encontrar las diferentes técnicas de desarrollo, junto a los principales lenguajes y gestores de bases de datos que se encuentran en el mundo de la informática siendo los más usados actualmente en diferentes sistemas y páginas web.

Luego se desarrolla el tercer capítulo donde se ven los objetivos, misión, visión y principales actividades del proceso de requisiciones y entrega de productos para el barco turístico; en el mismo capítulo se hace un análisis de las herramientas estudiadas anteriormente. Para elegir aquellas que se usaran en el proyecto, se hace uso de cuadros comparativos para llegar a las decisiones y así elegir el uso de las herramientas, lenguajes de programación y gestores de bases de datos.

A través de levantamiento de procesos se puede hacer un análisis del proceso principal y sus subprocesos los cuales deben ser estudiados para generar un producto de calidad, con la ayuda de algunos diagramas entre los cuales se encuentran:

- Cadenas de Valor.
- Mapa de Procesos.
- Sipoc.

Luego de este análisis se procedió a hacer la construcción del sistema en sí, previamente realizados los procesos de desarrollo de software los cuales son:

- Análisis de Requerimientos.
- Identificación de Actores.
- Especificación de Requerimientos.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

- Arquitectura.
- Diseño.
- Diagramas UML (Incluido diagramas de despliegue y de secuencia).

Una vez realizado estos pasos se procedió a elaborar un prototipo ejecutable demostrado mediante capturas de pantalla que muestran la funcionalidad del sistema y posteriormente proseguir con la construcción del sistema completo; no antes de proceder a realizar una lista detallada de riesgos que representa el proyecto.

En la construcción final del sistema en el cual se muestra la implementación y pruebas del mismo para saber cómo funciona en general, esto se realizara con capturas de pantalla para tener conocimiento de cómo interactúa el usuario con el sistema y el mismo con el manejo de datos. Dentro del mismo capítulo se encuentra la información a detalle de que errores que se pueden cometer el usuario y a su vez cómo el sistema reaccionará a tales errores, junto a la interacción del sistema con el gestor de base de datos que permite hacer análisis a detalle de qué ingresa o qué no ingresa al mismo.

Finalmente se hace como se conoce en proyectos informáticos, “la transición”, la cual consiste en hacer un análisis de seguridades, costos del proyecto, despliegue y mantenimiento del sistema terminado.

Para la terminación de esta disertación de grado se podrá encontrar las conclusiones y las recomendaciones necesarias que implicaron el desarrollo de esta disertación de grado.

Introducción

Como se conoce actualmente el desarrollo de sistemas informáticos representa una necesidad para la automatización de procesos que ayudan a las empresas ya sean grandes, medianas o pequeñas.

A su vez, esta disertación de grado toca un tema nuevo o innovador, debido a que se hace el desarrollo del sistema informático para un barco turístico el cual se encuentra funcionando actualmente; con la finalidad de que se llegue a implementar en totalidad.

El sistema desarrollado, aparte de generar un ahorro de tiempo en la mayoría de los procesos automatizados también, genera una metodología de trabajo más formal o disciplinario para el desarrollo del proceso en el cual se trabaja día a día.

Dentro de esta disertación de grado se encuentran los pasos a detalle para poder generar un sistema informático de calidad, mediante un previo estudio de las técnicas utilizadas actualmente para el desarrollo de proyectos informáticos los cuales utilizan varias compañías de prestigio; a su vez, también se puede entender la manera cómo el sistema interactúa con el usuario final.

En los capítulos intermedios y finales se puede entender cómo el proceso toma forma para la elaboración del sistema informático y para el caso particular junto con las técnicas actuales de desarrollo haciéndolo de manera efectiva y rápida.

1. CAPÍTULO: ANTECEDENTES.

1.1. Objetivos

- Objetivo General: Desarrollar e implementar un sistema de información para llevar el control de información de víveres e insumos para la administración de un barco dedicado al turismo.
- Objetivos Específicos:
 - Analizar el proceso de control de información de víveres e insumos en un barco turístico.
 - Diseñar y desarrollar un sistema de información para el proceso.
 - Implementar el sistema de información.
 - Capacitar al personal para el uso del sistema.
 - Generar reportes finales de lo implementado.

1.2. Perfil de Usuarios

Se han determinado los siguientes tipos de usuarios del sistema que se implementará, tanto para la generación de pedidos como la aprobación de los mismos, considerando su jerarquía:

1. Rol de Administrador:
 - Administrador del Sitio.
2. Rol de Aprobador de Pedidos:
 - Capitán de Barco y
3. Rol de Controlador:
 - Capitán de Tierra.
4. Rol de Proveedor:
 - Proveedor.
5. Rol de Solicitante:
 - Cocinero.
 - Ayudante de Cocina.
 - Barman.
 - Ama de llaves.

1.3. Funciones Desarrolladas en el Área

Las funciones que se desarrollan en el área de compras y adquisiciones es un poco simple al momento de generar órdenes, se desarrollan las siguientes funciones:

- Hacer pedidos.
- Generar pedidos.
- Aprobar pedidos.
- Despachar pedidos.

1.4. Procesos que se Llevan a Cabo

Procesos principales:

- Generar pedido.
- Aprobar pedido.
- Obtener producto.
- Entregar producto.

1.5. Problemas del Proceso

El principal problema en el proceso es la desorganización al momento de generar los pedidos ya que a veces existe duplicación de pedidos o a su vez mala generación de los mismos.

La obtención de la información es deficiente, ya sea por cantidad de productos entregados o también el total de gasto generado por los pedidos entregados en la semana o al mes o al trimestre de entrega de pedidos.

Todo movimiento realizado se lo documenta en papel, al ser solo de esta manera se mantiene con un riesgo de pérdida, manipulación y error de información acerca de los movimientos de pedidos y entregas.

2. CAPÍTULO: MARCO TEORICO.

2.1. Técnicas de Desarrollo de Aplicaciones

A continuación se presentarán los diferentes tipos de técnicas para el desarrollo de aplicaciones, con la finalidad de escoger una de ellas y aplicarla en el barco.

2.1.1. *Diseño Ágil de Aplicaciones (Extreme Programming)*

La Programación Extrema o Extreme Programming es una metodología de desarrollo de la Ingeniería de Software formulada por Kent Beck.

Es el más destacado de los procesos ágiles de desarrollo de software. La programación extrema se diferencia de las metodologías tradicionales principalmente en que brinda más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de Extreme Programming consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. La capacidad de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

La simplicidad es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hace que la complejidad aumente exponencialmente.

Para mantener la simplicidad es necesaria la reorganización del código, ésta es la manera de mantener el código simple a medida que crece.

También se aplica la simplicidad en la documentación. De esta manera, el código debe comentarse en su justa medida, intentando que el código esté autocomentado. Para ello se deben elegir adecuadamente los nombres de las variables, métodos y clases. Los nombres extensos no disminuyen la eficiencia del código ni el tiempo de desarrollo, gracias a las herramientas de autocompletado y reorganización que existen actualmente.

Aplicando la simplicidad junto con la autoría colectiva del código y la programación por parejas se asegura que cuanto más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real.

Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante.

Considérense los problemas que derivan de tener ciclos muy largos. Meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. Por ejemplo, las pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes en el código.

Ventajas:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.

Desventajas:

- Es recomendable emplearlo solo en proyectos a corto plazo.
- Altas comisiones en caso de fallar.

2.1.2. Rad (Diseño Rápido de Aplicaciones)

El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

Hoy en día se suele utilizar para referirnos al desarrollo rápido de interfaces gráficas de usuario tales como Glade, o entornos de desarrollo integrado completos. Algunas de las plataformas más conocidas son Visual Studio, Lazarus, Gambas, Delphi, FoxPro, Anjuta, Game Maker, Velneo o Clarion. En el área de la autoría multimedia, software como Neosoft Neoboo y MediaChance Multimedia Builder proveen plataformas de desarrollo rápido de aplicaciones, dentro de ciertos límites.

2.1.2.1. *Planificación de los Requisitos.*

Requiere que usuarios con un vasto conocimiento de los procesos de la compañía determinen cuáles serán las funciones del sistema. Debe darse una discusión estructurada sobre los problemas de la compañía que necesitan solución.

2.1.2.2. *Diseño.*

Se hace un análisis de las actividades de la compañía en relación al sistema propuesto. Los usuarios participan en talleres que descomponen funciones y definen entidades asociadas con el sistema.

2.1.2.3. *Construcción:*

Se afirman los requisitos, repasar los resultados y se hacen pruebas al sistema. También se crean las instrucciones, rutinas y procedimientos para operar la nueva aplicación.

2.1.2.4. *Implementación:*

La implementación y el manejo del cambio del viejo al nuevo sistema. Se hacen pruebas y se adiestran los usuarios. Los cambios organizacionales y la operación del nuevo sistema se hacen en paralelo con el viejo sistema hasta que el nuevo se establezca completamente.

2.1.2.5. *Modelado de Gestión:*

El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera?, ¿Quién la genera?, ¿A dónde va la información?, ¿Quién la procesa?.

2.1.2.6. *Modelado de Datos:*

El flujo de información definido como parte de la fase de modelado de gestión se redefine como un conjunto de objetos de datos necesarios para apoyar la empresa.

2.1.2.7. *Modelado del Proceso:*

Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión.

2.1.2.8. *Generación de Aplicaciones.*

El proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas para facilitar la construcción del software.

2.1.2.9. *Entorno RAD Sobre Navegadores Web*

Muchos entornos RAD permiten desarrollar interfaces de bases de datos basadas en navegadores web; que permiten acceder a la base de datos desde cualquier lugar a través de Internet, esto reduce los costos y permite a los usuarios externos tener acceso a una base de datos; sin embargo, las limitaciones propias de los navegadores web y los protocolos de Internet hacen a este enfoque no adecuado para sistemas donde se requieran respuestas interactivas muy rápidas.

Ventajas:

- Comprar puede ahorrar dinero en comparación con construir.
- Los entregables pueden ser fácilmente trasladados a otra plataforma.
- El desarrollo se realiza a un nivel de abstracción mayor.
- Visibilidad temprana (Ingeniería de Software).
- Mayor flexibilidad.
- Menor codificación manual.
- Mayor involucramiento de los usuarios.
- Posiblemente menos fallas.
- Posiblemente menor costo.
- Ciclos de desarrollo más pequeños.
- Interfaz gráfica estándar.

Desventajas:

- Para proyectos en gran escala se requiere recursos humanos suficientes como para crear el número suficiente de equipos.
- Debe haber un compromiso fuerte entre todas las partes para completar el sistema en el tiempo necesario.
- No es adecuado cuando los riesgos técnicos son muy alto.

2.1.3. JAD (Diseño de Aplicaciones por Juntas)

Al elaborar un sistema de información, se sabe que se deben de seguir pasos y etapas para realizar software de calidad; dentro de la etapa de Análisis se ve la determinación de requerimientos, que se pueden obtener mediante diversas técnicas, entre ellas la entrevista, que es un método en el que se interactúa personalmente con los usuarios del sistema.

A través de la entrevista se puede estar consciente de las necesidades de los usuarios; proporciona además detalles que llegan a ser muy importantes para la creación del sistema, y hace que el entrevistado se dé cuenta que la empresa está interesada en su opinión y lo hace sentir parte de ella.

La entrevista no puede ser realizada por cualquier persona y de cualquier manera, hay pasos o procedimientos a seguir para lograr el éxito en la obtención de información clave para la realización del sistema. Así se observa que la persona encargada de realizar las entrevistas dentro de la empresa debe ser alguien afable, alguien con quien sea grato hablar pero que también sea experto en su campo, un analista que sepa obtener la información necesaria.

Ésta es una técnica muy usada debido a sus grandes resultados porque se logra tener conexión con las partes, que son las que en realidad interactúan el sistema de la organización y que no se deben pasar por alto al analizar las opciones de obtención de datos. A pesar de los grandes resultados y el beneficio que trae la entrevista dentro de las empresas, también conlleva una gran desventaja, tiende a ser tardía, por el gran detalle que se le aplica a esta técnica; lleva mucho tiempo entrevistar a las personas que se cree nos darán la información necesaria, pues las entrevistas son de persona a persona.

2.1.3.1. ¿Qué es JAD?

Diseño Conjunto de Aplicaciones (JAD), también conocido como el Desarrollo de Aplicaciones Comunes, es una técnica para determinar rápidamente los requerimientos del sistema mediante la obtención de aportes de una muestra representativa de las partes interesadas. Un equipo adecuado compuesto de los principales usuarios, administradores y analistas de sistemas se ensambla. Posteriormente, el equipo se reúne en una sesión intensiva para recopilar datos, una lluvia de ideas, debatir ideas, conciliar las diferencias, identificar y priorizar las necesidades y generar soluciones alternativas. JAD también puede ser utilizado para realizar análisis de viabilidad, análisis de costo / beneficio y análisis de riesgo. A menudo, el diseño de especificaciones tales como diagramas de flujo de datos, diagramas entidad relación, y diagramas de flujo del sistema se generan durante el período de sesiones JAD.

2.1.3.2. Instalaciones Para el Desarrollo de JAD

Como mínimo, una sala de conferencias lo suficientemente grande como para acomodar a todos los miembros del equipo y están equipadas con pizarras o tableros, un proyector debe estar disponible. Con la aparición de los sistemas de reunión electrónica, el grupo de sistemas de soporte de decisiones, y la ingeniería de software asistida por ordenador, los requisitos adicionales pueden incluir equipos para llevar a cabo reuniones por medios electrónicos, servicios de tele-conferencia y una estación de maestros equipados con software.

2.1.3.3. Realizar la Sesión JAD

Una sesión JAD es un curso intensivo (por lo general) de dos o tres días de reunión de todo el equipo de JAD. Se espera que los miembros del equipo que están en período de sesiones JAD, ejerzan una atención completa para la programación de actividades, las actividades conflictivas no se encuentran dentro de la programación.

2.1.3.4. Preparación

Antes del inicio de la sesión JAD, los analistas responsables de los sistemas de información dicen que los consultores deben:

- Definir el alcance del sistema.
- Identificar los problemas, limitaciones y restricciones.

- Estimar las necesidades de recursos (tiempo, presupuesto, personal) para desarrollar el sistema.
- Identificar los gastos preliminares, beneficios, riesgos e impactos del proyecto.
- Identificar la naturaleza y los atributos principales del proyecto, las dependencias del proyecto y la interrelación del proyecto.
- Identificar los sub-proyectos apropiados. (El proyecto es a veces, descompuesto en varias sub-proyectos, debido a la oportunidad y / o las limitaciones presupuestarias.)
- Realizar el análisis de los antecedentes necesarios para definir parámetros fundamentales como el número de usuarios, el tamaño de la base de datos, el rendimiento requerido y la respuesta mínimo aceptable veces.
- Planificar la sesión JAD.

2.1.3.5. *El Período de Sesiones*

Una sesión JAD comienza con una descripción del material recogido durante la etapa de preparación. Una vez que los participantes entienden el problema, comienza el proceso de identificación de las dimensiones del problema, las posibles causas, necesidades y soluciones alternativas.

Durante una sesión JAD, es responsabilidad del moderador gestionar con eficacia el período de sesiones, para asegurar que el equipo se mantenga enfocado en los temas del programa, para alentar a todos los miembros del equipo a participar, y para resolver los conflictos generados durante el período de sesiones. Debido a que el equipo está compuesto en gran parte del personal no técnico, es importante que los analistas de los sistemas de información traten de minimizar el uso de términos técnicos.

2.1.3.6. *Beneficios Principales*

- Ahorro del tiempo en comparación con las entrevistas tradicionales persona a persona.
- Desarrollo rápido por medio del JAD (debido a que las entrevistas de usuarios no son realizadas seriamente a lo largo de semanas o meses, el desarrollo puede ser más rápido).
- Permite mejoras en el sistema de información, ya que su naturaleza interactiva y la alta visibilidad permiten que el JAD ayude a los usuarios a involucrarse desde el principio en el proyecto del sistema y traten su realimentación seriamente.
- Beneficia el desarrollo creativo de los diseños.

2.2. Análisis de Herramientas de Desarrollo

A continuación, se presenta el análisis y descripción de las herramientas que serán utilizadas en el proyecto:

2.2.1. Sistema de Gestión de Bases de Datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico) y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

Existen programas denominados Sistemas Gestores de Bases de Datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos SGBD, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

2.2.1.1. Objetivos

- *Abstracción de la información:* Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- *Independencia:* La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- *Consistencia:* En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad

determinada que tiene determinadas condiciones, por ejemplo que los menores de edad no pueden tener licencia de conducir. El sistema no debería aceptar datos de un conductor menor de edad. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.

- *Seguridad:* La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentre segura.
- *Manejo de transacciones:* Una transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que luego de una ejecución en la que se produce una falla se obtiene lo mismo que si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.

2.2.1.2. Ventajas

- Simplifican la programación de equipos de consistencia.
- Manejando las políticas de respaldo adecuadas, garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores.
- Organizan los datos con un impacto mínimo en el código de los programas.
- La correcta dedicación por parte de los desarrolladores disminuyen drásticamente los tiempos de proceso y aumentan la calidad del sistema desarrollado.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

2.2.1.3. MySQL

La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a base de datos en Internet. Sistema de Gestión de Base de Datos. Una implementación Cliente Servidor, basado en el álgebra relacional, se caracteriza por disponer toda la información contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos.

Es un software de código abierto escrito en C y C++, accesible para cualquiera para usarlo y modificarlo. MySQL usa el GPL (GNU Licencia Publica General) no cuesta dinero a menos que lo incluyamos en un software comercial.

2.2.2. Lenguajes De Programación

2.2.2.1. .NET

.Net es la nueva plataforma de desarrollo que ha lanzado al mercado Microsoft, y en la que ha estado trabajado durante los últimos años.

Microsoft .NET es un conjunto de tecnologías de software, compuesto de varios lenguajes de programación que se ejecutan bajo el .NET Framework. Es además un entorno completamente orientado a objetos y que es capaz de ejecutarse bajo cualquier plataforma.

Podríamos dividir el entorno .NET en las siguientes partes:

- .NET Framework, que es el entorno de trabajo de la plataforma .NET y que la engloba completamente. Toda la plataforma .NET forma parte de .NET framework.
- Lenguajes .NET. Destacan C# y VB.NET, y recientemente J#, aunque existen más lenguajes y con toda seguridad surgirán nuevos.
- El Common Runtime Language CRL, que es el motor de ejecución común a todos los lenguajes .NET.
- MSIL, Microsoft Intermedial language, es el lenguaje intermedio al que compilan las aplicaciones (Asemmblies) .NET. Este lenguaje intermedio es interpretado por el CRL en tiempo de ejecución.
- CLS, Common Language Specification, que engloban las pautas que deben cumplir los lenguajes .NET. Es está característica la que va a permitir a otras compañías producir lenguajes compatibles con .NET.
- ADO.NET, es la nueva interfaz de bases de datos. No se trata de una evolución de ADO, sino que se trata de una interfaz completamente nueva.
- ASP.NET, es la nueva tecnología para páginas web dinámicas completamente integrada dentro del entorno .NET. Representa una auténtica revolución en el desarrollo Web (Internet e Intranet).
- Biblioteca de clases .NET, es el conjunto de clases que componen el .NET framework y que permiten realizar casi cualquier tarea de una manera fácil y rápida.

2.2.2.2. *PHP*

PHP usa una mezcla entre interpretación y compilación para intentar ofrecer a los programadores la mejor mezcla entre rendimiento y flexibilidad.

PHP compila para el código en una serie de instrucciones (llamadas opcodes) siempre que estas son accedidas. Estas instrucciones son ejecutadas una por una hasta que el script termina. Esto es diferente a la manera convencional de compilación de lenguajes como C++ donde el código es compilado a código ejecutable que es ejecutado después. Php es recompilado cada vez que se solicita un script.

Esta constante recompilación puede parecer una pérdida de tiempo, pero no lo es porque no hay que preocuparse de la recompilación cada vez que realizas cambios.

Además, proporciona una regeneración muy rápida durante el desarrollo. Si se obtienen errores en el código, PHP rechazará compilar la página hasta que se arregle el problema, y se tendrá que ejecutar el código línea a línea hasta encontrar el problema.

Una ventaja importante de interpretar el código es que toda la memoria usada por el código es manejada por PHP y el lenguaje automáticamente vacía esta memoria cuando el script finaliza. Esto significa que no tiene que preocuparse de las conexiones a la base de datos, porque PHP lo hará de por sí. Tampoco quiere decir que se deba dejar hacer todo a PHP, los buenos programadores hacen esas funciones programándolas en código teniendo un mayor control del script.ssd.

2.2.2.3. *JSP*

La implementación actual de JSP está limitada como un lenguaje script. El diseño de JSP permitirá a otros lenguajes scripts ser soportados en un futuro. JSP ayuda en la creación de nuevos tags que son interpretados por el procesador JSP, permitiendo la extensión del lenguaje. Esta nueva propuesta tiene que ver con la codificación dentro de la propia página. Una vez que se accede a la misma, se ejecuta el código contenido, y la información desplegada en el cliente pasa a ser HTML puro generado en el momento de la consulta.

Los Java Server Pages nos permitirán tener código java para la manipulación de los datos y poder construir páginas de manera dinámica, que se actualicen directamente con la información de la base de datos. Finalmente, el usuario vera el sitio como un documento HTML.

2.2.3. Herramientas Case

En este punto se hablará de las herramientas case debido a que en el proyecto se utilizará un framework que aplica las mismas. Se indican los detalles a continuación:

Son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

El empleo de herramientas Case permiten integrar el proceso de ciclo de vida:

- Análisis de datos y procesos integrados mediante un repositorio.
- Generación de interfaces entre el análisis y el diseño.
- Generación del código a partir del diseño.
- Control de mantenimiento.

Actualmente, la tendencia en el desarrollo de software está enfocada hacia las microcomputadoras como plataformas de Ingeniería de Software, que se interconectan mediante redes para que puedan comunicarse de forma efectiva. La base de datos del proyecto, está disponible a través de un servidor de archivos en red que es accesible desde todas las estaciones de trabajo. Un sistema operativo que gestiona el hardware, la red y las herramientas mantienen todo el entorno unido.

La arquitectura de entorno, compuesta por la plataforma hardware y el soporte del sistema operativo, constituye la base del CASE. Pero el entorno CASE, en sí mismo, necesita otros componentes. Un conjunto de servicios de portabilidad constituyen un puente entre las herramientas CASE y su marco de integración y la arquitectura de entorno. El marco de integración es un conjunto de programas especializados que permite a cada herramienta CASE comunicarse con las demás, para crear una base de datos de proyectos y mostrar una apariencia homogénea al usuario final. Los servicios de portabilidad permiten que las herramientas CASE y su marco de integración puedan migrar a través de diferentes plataformas hardware y sistemas operativos sin grandes esfuerzos de adaptación. Ver Ilustración 1.



Ilustración 1: Capas Herramientas Case. Referencia: <http://ocw.ua.es/ingenieria-arquitectura/programacion-en-internet/03c-AplicacionesWeb.pdf>

La mayoría de las herramientas Case no han sido construidas utilizando todos los bloques componentes. Muchas de éstas son soluciones puntuales, esto es, una herramienta se utiliza como ayuda en una actividad concreta de Ingeniería de Software, pero no se comunica directamente con otras herramientas, porque no está unida a una base de datos de proyectos. Aunque esta situación no es la ideal, una herramienta Case puede ser utilizada eficientemente, aun siendo una solución puntual.

En el nivel más bajo del espectro de integración está la herramienta individual. Cuando las herramientas proporcionan facilidades para el intercambio de datos, el nivel de integración aumenta ligeramente. Estas herramientas generan una salida en un formato estándar compatible con otras herramientas que puedan leer ese formato. En algunos casos, los que construyen herramientas CASE complementarias trabajan juntos para establecer un puente entre ellas. Utilizando este enfoque, la compatibilidad entre herramientas puede generar productos finales que serían difíciles de desarrollar utilizando cada herramienta por separado. La integración por fuente única se da cuando un constructor de herramientas CASE integra diferentes herramientas y las vende como un único paquete. Aunque este enfoque es bastante efectivo, la mayoría de los entornos provenientes de una misma fuente tienen una arquitectura cerrada que hace difícil añadir nuevas herramientas de otros vendedores.

Al final del espectro de integración está el entorno de soporte de proyectos integrado. Se crean estándares para cada uno de los bloques componentes. Los vendedores de herramientas CASE utilizan estos estándares IPSE para construir herramientas entre sí. Ver Ilustración 2.

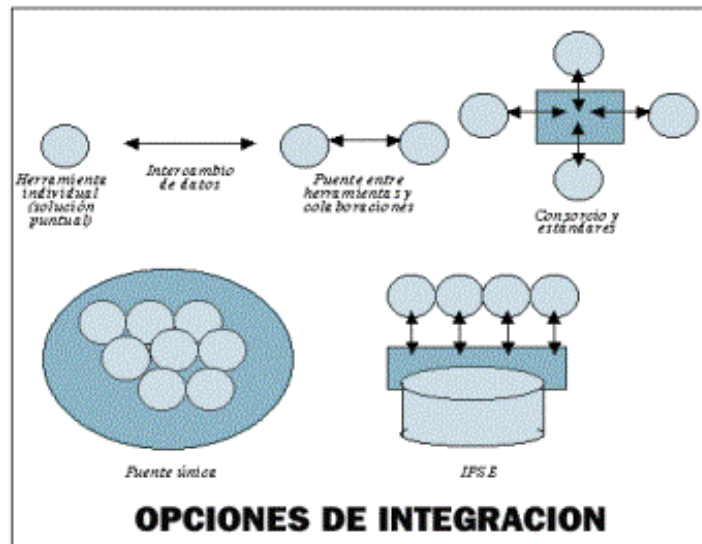


Ilustración 2: Opciones de Integración. Referencia: <http://ocw.ua.es/ingenieria-arquitectura/programacion-en-internet/03c-AplicacionesWeb.pdf>

La principal ventaja de la utilización de una herramienta CASE, es la mejora de la calidad de los desarrollos realizados y en segundo término, el aumento de la productividad. Para conseguir estos dos objetivos es conveniente contar con una organización y una metodología de trabajo además de la propia herramienta.

La mejora de calidad se consigue reduciendo sustancialmente muchos de los problemas de análisis y diseño inherentes a los proyectos de mediano y gran tamaño.

La mejora de productividad se consigue a través de la automatización de determinadas tareas como la generación de código y la reutilización de objetos o módulos.

2.2.4. Servidores de Aplicaciones (Servidor Web)

Se denomina servidor de aplicaciones a un servidor en una red de computadores que ejecuta ciertas aplicaciones, un servidor de aplicaciones generalmente gestiona la mayor parte o la totalidad de las funciones de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones.

Un servidor web es un programa que implementa el protocolo HTTP; es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones y que responde a estas peticiones adecuadamente mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Un servidor web se mantiene a la espera de peticiones HTTP por parte de un cliente HTTP que solemos conocer como navegador de internet, el cliente realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web.

Servidores web importantes:

- Apache.
- Cherokee.

Servidor HTTP Apache

Es un servidor web HTTP de código abierto; presenta entre otras características configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. Es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP, Perl o Python.

2.3. Arquitectura De Aplicaciones

2.3.1. Cliente/Servidor

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor y este envía uno o varios mensajes con la respuesta. En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. Ej. Ilustración 3.

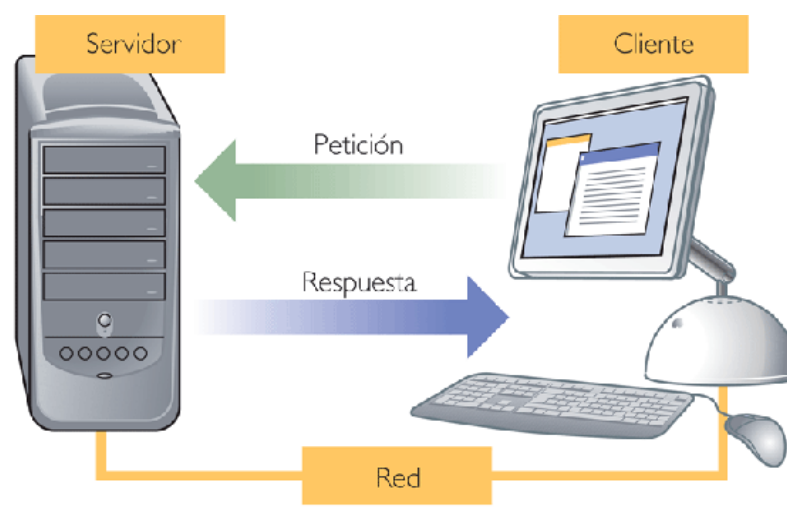


Ilustración 3: Modelo Cliente/Servidor. Referencia:

http://www.kalipedia.com/tecnologia/tema/comunicaciones/graficos-modelo-cliente-servidor.html?x1=20070821klpinginf_30.Ees&x=20070821klpinginf_71.Kes

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas Cliente/Servidor es mediante la explotación de las Computadoras Personales (PC) a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo mainframe. Usualmente, la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario, aunque esto puede variar. En otras palabras, la arquitectura Cliente/Servidor es una extensión de programación modular en la que la

base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento.

Esta arquitectura permite distribuir físicamente los procesos y los datos en forma más eficiente lo que en computación distribuida afecta directamente el tráfico de la red, reduciéndolo grandemente.

2.3.1.1. *Cliente*

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor.

El Cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario, además de acceder a los servicios distribuidos en cualquier parte de una red o Internet, si el Servidor que provee la aplicación está en una parte muy remota.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

2.3.1.2. *Servidor*

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él.

El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.

- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

2.3.1.3. Características

Las características básicas de una arquitectura Cliente/Servidor son:

- Combinación de un cliente que interactúa con el usuario y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente/Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

2.3.1.4. *Ventajas*

Entre las principales ventajas del esquema Cliente/Servidor están:

- Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas. Esto constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- El esquema Cliente/Servidor facilita la integración entre sistemas diferentes y comparte información permitiendo por ejemplo, que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. De esta manera, podemos integrar Computadoras Personales con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
- Al favorecer el uso de interfaces gráficas interactivas, los sistemas Construidos bajo este esquema tienen mayor interacción y más intuitiva con el usuario. En la utilización de interfaces gráficas para el usuario, el esquema Cliente/Servidor presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- Una ventaja adicional del uso del esquema Cliente/Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes.
- La estructura inherentemente modular facilita además, la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales pero permitiendo la integración de la información relevante a nivel global.

2.3.1.5. *Desventajas*

Entre las principales desventajas del esquema Cliente/Servidor están:

- El mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- Se cuenta con muy escasas herramientas para la administración y ajuste del desempeño de los sistemas.
- Es importante que los clientes y los servidores utilicen el mismo mecanismo, lo cual implica que se deben tener mecanismos generales que existan en diferentes plataformas.
- Además, hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos.
- La seguridad de un esquema Cliente/Servidor es otra preocupación importante. Por ejemplo, se deben hacer verificaciones en el cliente y en el servidor.
- El desempeño es otro de los aspectos que se deben tener en cuenta en el esquema Cliente Servidor. Problemas de este estilo pueden presentarse por congestión en la red, dificultad de tráfico de datos, etc.

2.3.2. *Por Capas*

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad e integración.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario la utilización del sistema (GUI). Ver Ilustración 4

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

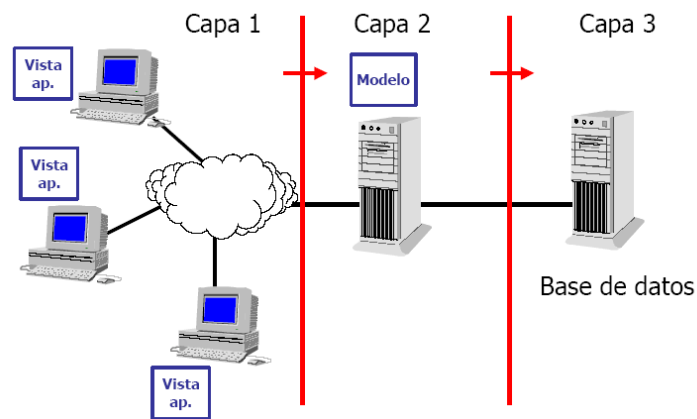


Ilustración 4: Modelo 3 Capas Referencia: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>

Si establecemos una separación entre la capa de interfaz gráfica (cliente), replicada en cada uno de los entornos de usuario y la capa modelo, que quedaría centralizada en un servidor de aplicaciones, obtenemos una potente arquitectura que nos otorga algunas ventajas:

- Centralización de los aspectos de seguridad y transaccionalidad que serían responsabilidad del modelo.
- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.
- Mayor sencillez de los clientes.

Si intentamos aplicar esto a las aplicaciones web, debido a la obligatoria sencillez del software cliente que será un navegador web nos encontramos con una doble posibilidad:

- Crear un modelo de 4 capas, separando cliente, servidor web, modelo y almacén de datos. Esto nos permite una mayor extensibilidad en caso de que existan también clientes no web en el sistema que trabajarían directamente contra el servidor del modelo.
- Mantener el número de capas en 3, integrando interfaz web y modelo en un mismo servidor aunque conservando su independencia funcional. Ésta es la distribución en capas más común en las aplicaciones web. Ver Ilustración 5 e Ilustración 6.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

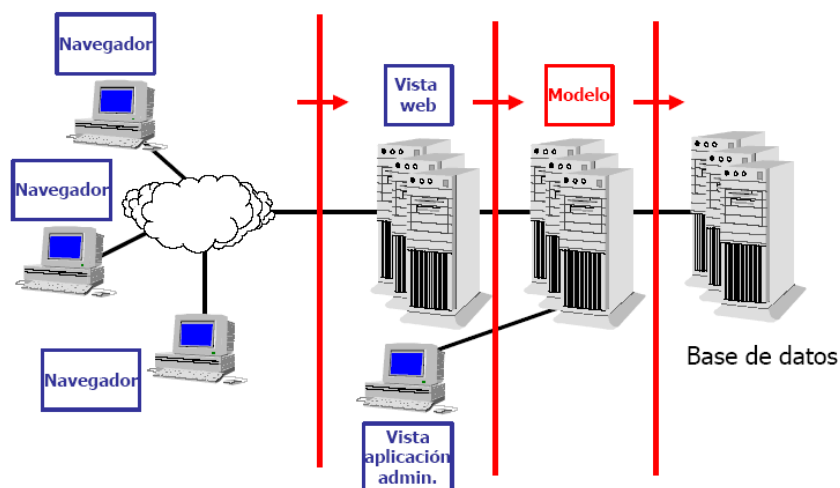


Ilustración 5: modelo 4 capas web. Referencia: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>

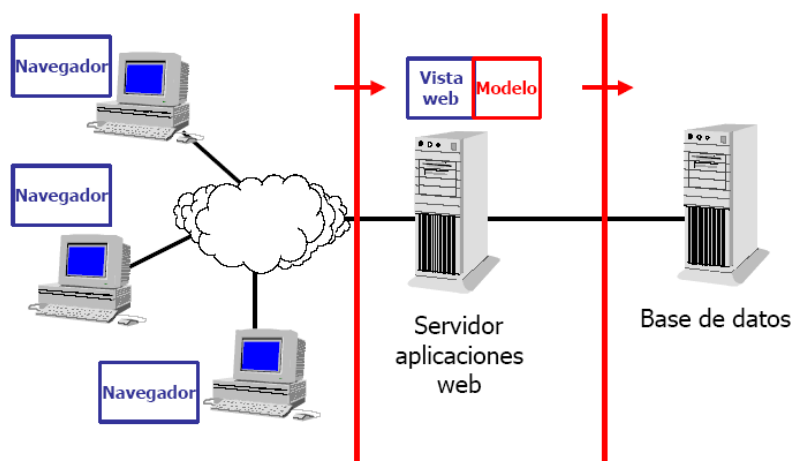


Ilustración 6: Modelo 3 Capas Web. Referencia: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>

2.3.2.1. Capa de presentación

Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. También es conocida como interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa de negocio.

2.3.2.2. Capa de negocio

Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio e incluso de lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También, se consideran aquí los programas de aplicación.

2.3.2.3. Capa de datos

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

3. CAPÍTULO: CASO DE ESTUDIO.

3.1. Proceso Pedido y Entrega de Insumos y Víveres en el Barco Treasure Of Galápagos

En este capítulo se detallara algunos puntos de la problemática del proyecto y sus posibles herramientas que se elegirán para el mismo.

3.1.1. Objetivos

3.1.1.1. General

Satisfacer el stock en materia prima del barco Treasure of Galápagos.

3.1.1.2. Específicos

- Generar solicitud de materia prima.
- Entrega de materia prima.
- Control de stock en materia prima.

3.1.2. Misión

La empresa está dedicada al servicio personalizado en cruceros de galápagos, pionera en el mercado, comprometidos con la satisfacción de todos sus clientes, promoviendo siempre el liderazgo y el trabajo en equipo, avanzando con la mejor calidad y eficiencia.

3.1.3. Visión

Liderar Galápagos en cadenas de cruceros catamarán de primera categoría, dentro de los próximos 5 años de servicio.

3.1.4. Cuadro Jerárquico.

El cuadro indica perfiles y tareas dentro del Barco Treasure of Galápagos. Ver Ilustración 7.

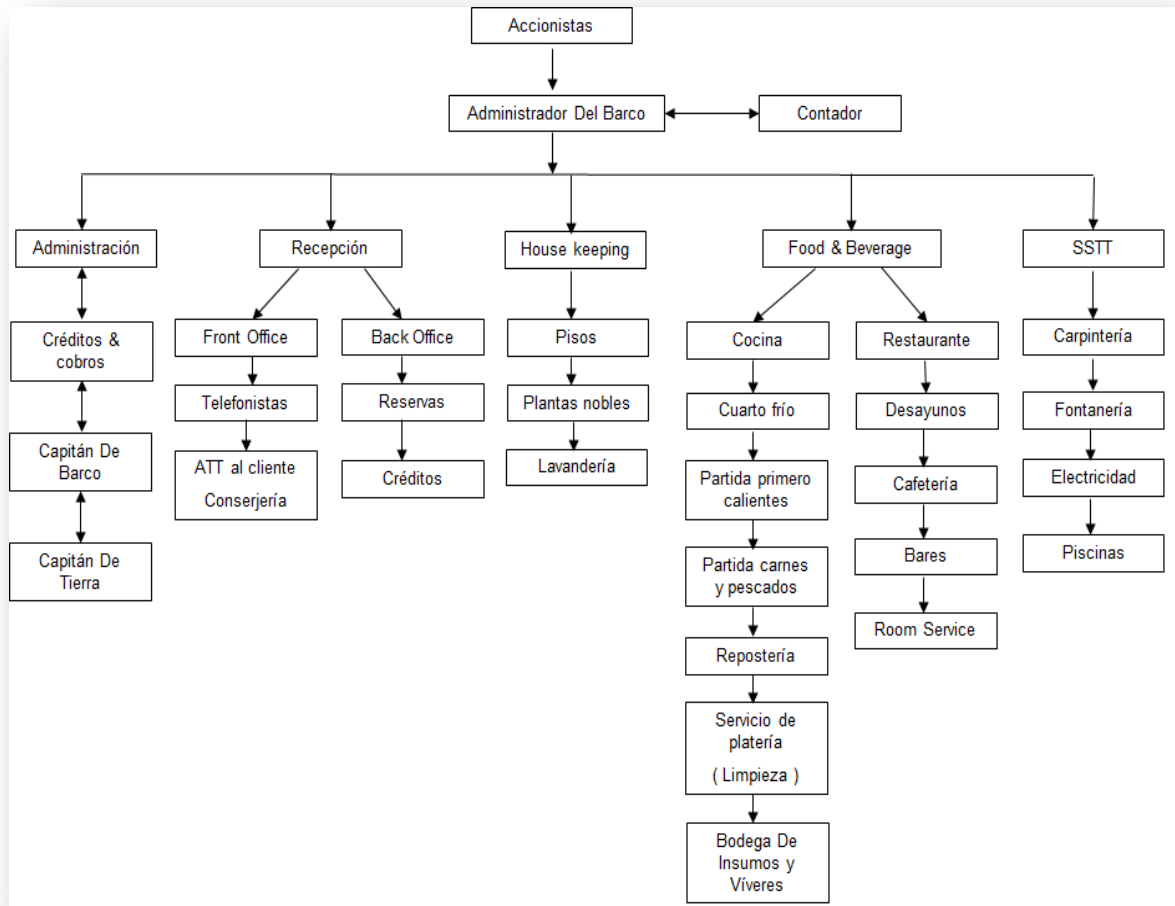


Ilustración 7: Cuadro Jerárquico. Autor: Juan Fernando Nieto.

3.1.5. Mapa de Procesos del Barco.

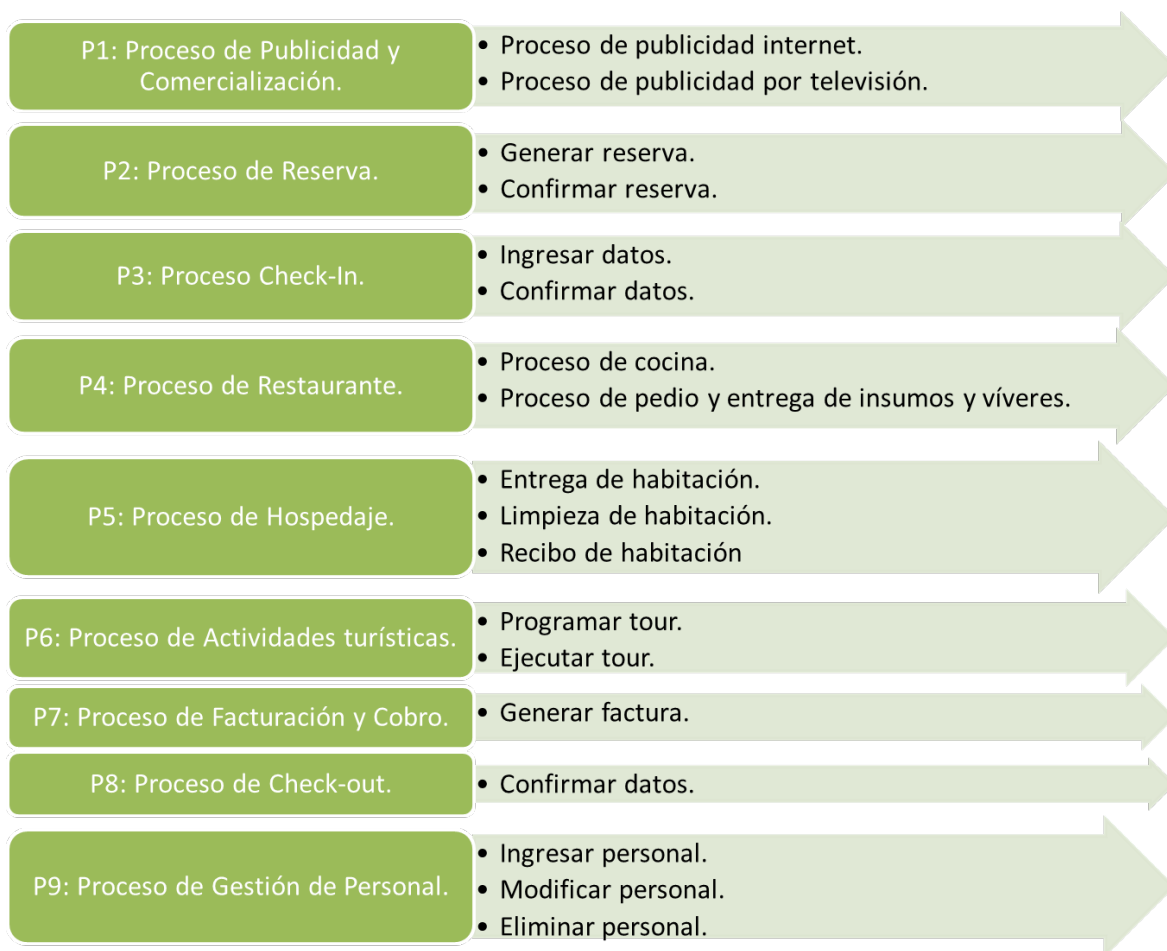


Ilustración 8: Mapa de Procesos. Autor: Juan Fernando Nieto y Alex Nieto.

3.1.6. Principales Actividades del Proceso de Pedido y Entrega de Insumos y Víveres.

A continuación se detallan las actividades que se realizan en el proceso de administración de insumos y víveres del barco:

1. Realizar un inventario de materia prima existente.
2. Verificar de stock en bodega de materia prima.
3. Realizar pedido de materia prima faltante a logística.
4. Entregar requisición a bodeguero.
5. Enviar a proveedores el pedido de materia prima.
6. Recibir materia prima para realizar nuevo stock de materia prima.

3.1.7. Selección de las Herramientas Para la Construcción De la Aplicación.

Dentro de la selección de herramientas existen varios puntos a tratarse, los cuales se los ha medido y se los ha probado uno por uno para poder sacar a relucir la mejor de opciones al momento de elegir las herramientas que formarán parte del proyecto y aplicación del mismo.

3.1.7.1. Técnica De Desarrollo De Aplicaciones.

La técnica para desarrollar el sistema planteado será por medio de RAD (Diseño Rápido de Aplicaciones), el cual ya fue descrito en el capítulo anterior, demuestra su eficiencia frente a las otras metodologías de desarrollo debido a que tiene mayor contacto con el cliente.

El cliente dentro del desarrollo RAD tiene mayor interacción con el cliente que la técnica de Diseño Ágil (Extreme Programming), pero a su vez mantiene menos contacto con el cliente que JAD (Diseño de Aplicaciones por Juntas) el cual se lo desarrolla para grandes proyectos y con mayor número de procesos dentro del proyecto.

En este caso, el cliente como tal tendrá la ventaja de mantenerse al tanto del desarrollo del proyecto y a su vez poder aportar con nuevos requerimientos en la parte de interacción con el usuario; al momento que se guarda esta interacción el proyecto se vuelve más estable y es de mayor confianza ya que el cliente obtiene un producto que se va adaptando con sus necesidades.

Para el desarrollo del sistema se procederá a una planificación a corto tiempo manteniendo el esquema de trabajo que plantea RAD con el cual se logra la finalización del sistema en corto tiempo.

3.1.7.2. Arquitectura De Aplicación.

Debido a que es un aplicativo web se mantendrá la política de la arquitectura por capas, con lo cual tenemos un esquema de tres capas el cual consistirá en la capa de presentación, la capa del negocio (la interacción entre la capa de presentación y capa de datos) y la capa de datos (sistema gestor de base de datos); esto facilita la modalidad de trabajo en el momento de desarrollar el sistema.

Para facilitar esta arquitectura de aplicación se usara el framework Kohana para el desarrollo de tres capas, el cual facilita el trabajo al momento de usar el lenguaje de

programación PHP; en lugar de utilizar el template que es también usado para PHP Smarty el cual no puede ser usado en este caso porque no respeta la arquitectura de 3 capas para el desarrollo del sistema a implementar.

En el siguiente cuadro comparativo veremos las diferencias entre ambos para poder elegir la correcta herramienta para el desarrollo del aplicativo. Ver Tabla 1.

KOHANA	SMARTY	SUPERIOR
Framework	Template	KOHANA
Extremadamente ligero	Ligero	KOHANA
PHP 5 Estricto	un script que solo incluye programación.	KOHANA
Curva de aprendizaje fácil, pero extensa	Script Complejo de entender	KOHANA
Se puede usar con Doctrine ORM, Twig y hasta con tus propias clases sin problemas.	X	KOHANA
IDE – Trabaja muy bien con Netbeans.	IDE	Empate

Tabla 1: Cuadro Comparativo KAHANA VS SMARTY Autor: Alex Nieto

3.1.7.3. Lenguaje De Programación.

Después de un análisis de los mismos lenguajes presentados anteriormente y dado su interactividad entre los mismos se pudieron elegir 3 de ellos en especial el más importante que es del lado del servidor es PHP; el cual por ser flexible para la aplicación mediana en la que se trabajara es compatible con los lenguajes de programación de lado del cliente los cuales son JavaScript y Ajax para que el cliente tenga una armonía de comunicación con el sistema desarrollado.

PHP al ser un lenguaje de libre desarrollo no tiene un costo de licenciamiento por lo cual no se debe pagar el uso del mismo.

A continuación, se presentará la comparación de los lenguajes antes vistos para ver el motivo por lo cual se eligió PHP en lugar de los otros.

PHP vs JSP (JAVA)

En el mundo del desarrollo web libre siempre están los dos lenguajes de programación más potentes los cuales son PHP y JSP, con lo cual después de hacer una descripción en el capítulo anterior, vamos ahora a comparar ambas herramientas en el uso de este sistema, por lo cual se elegiría a PHP por su versatilidad y también porque existe ya un framework libre llamado Kohana mantiene un¹Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

En el siguiente cuadro mostramos las ventajas más básicas de cada lenguaje de programación. Ver Tabla 2.

VENTAJAS		
	Lenguaje PHP	Lenguaje JSP
	Muy fácil de aprender.	Usa Visual Basic Script, siendo fácil para los usuarios.
	Lenguaje muy rápido.	
	Clases y herencia.	
	Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otra	Comunicación óptima con SQL Server.
	Capacidad de expandir su potencial utilizando módulos.	
	Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.	
	Es libre.	
	Incluye gran cantidad de funciones.	Soporta el lenguaje JScript (Javascript de Microsoft).
	No requiere definición de tipos de variables ni manejo detallado del bajo nivel.	
Total a favor	9	3

Tabla 2: Cuadro Comparativo PHP y JSP. Autor: Alex Nieto

¹ (Wikipedia, 2013)

En conclusión, debido a que PHP es un lenguaje rápido, por su facilidad al manejar variables y a su vez mantener información y documentación en su página oficial; es muy conveniente al momento del desarrollo y entregar soporte, debido a que se mantiene al día en lo último del desarrollo de este lenguaje interpretado.

PHP vs .NET

Dentro del desarrollo web también hay que hacer la comparativa dentro del campo de código libre y código propietario el cual la tecnología .NET (propiedad de Microsoft); mantiene un costo bastante elevado el cual para proyectos pequeños no es conveniente pagar un porcentaje circunstancial. A su vez PHP es bastante robusto para el desarrollo hasta incluso para proyectos de media escala con lo cual podemos generar un crecimiento alto.

En el siguiente cuadro mostramos las ventajas más básicas de cada lenguaje de programación. Ver Tabla 3.

VENTAJAS		
	Lenguaje PHP	Lenguaje .NET
	Muy fácil de aprender.	
	Lenguaje muy rápido.	Incremento de velocidad de respuesta del servidor.
	Clases y herencia.	Completamente orientado a objetos.
	Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otra	Mayor velocidad y Mayor seguridad.
	Capacidad de expandir su potencial utilizando módulos.	Controles de usuario y personalizados.
	Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.	
	Es libre.	
	Incluye gran cantidad de funciones.	División entre la capa de aplicación o diseño y el código.
	No requiere definición de tipos de variables ni manejo detallado del bajo nivel.	Facilita el mantenimiento de grandes aplicaciones.
Total a favor	9	6

Tabla 3: Cuadro comparativo PHP y .NET. Autor: Alex Nieto

Para concluir entre estos dos podemos ver claramente que PHP es más sencillo de aprender y a su vez nuevamente el tener documentación al día favorece el desarrollo de las aplicaciones; a su vez .NET le gana en su incremento lamentablemente es solo con un tipo de gestor de bases de datos que es también de propiedad de Microsoft, PHP es más dinámico al momento de interactuar con un servidor ya que funciona sobre varias plataformas de gestión en bases de datos.

Conclusión

La conclusión que podemos llegar a dar por final es que PHP cumple con todos los requerimientos para el desarrollo de este sistema que se lo implementara con el lenguaje de programación PHP. Tanto por demostrar que es la mejor opción después de analizar las ventajas de cada lenguaje.

3.1.7.4. Sistema Gestor de Base de Datos.

Para el desarrollo de una base de datos que interactúe de manera adecuada con el lenguaje de programación elegido, nos disponemos a usar la que viene incluido en la herramienta de desarrollo XAMPP el cual es la plataforma MySQL, el cual es el más estable para las aplicaciones desarrolladas con PHP.

A continuación vamos a hacer una comparativa entre los 2 gestores de bases de datos libres mayormente usados en el mundo del desarrollo web.

MySQL vs PostgreSQL

Ambos por ser caracterizados los mejores gestores de bases de datos libres es importante denotar la comparativa significativa entre los dos haciendo una comparativa simple; a su vez que MySQL trabaja de manera más estable con el lenguaje de programación PHP de forma nativa.

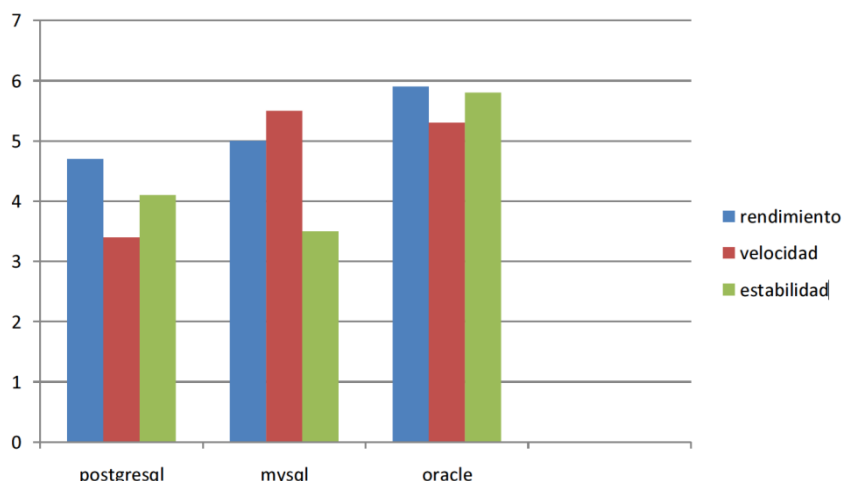
En el siguiente mostramos las ventajas de ambos gestores libres para dar una conclusión de cual gestor es el más aceptable para el desarrollo del sistema. Ver Tabla 4.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

PostgreSQL	MySQL	SUPERIOR
Posee una gran escalabilidad.		PostgreSQL
Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.	Carece de soporte para transacciones, rollback's y subconsultas.	PostgreSQL
Tiene la capacidad de comprobar la integridad referencial.	No maneja la integridad referencial.	PostgreSQL
Consume gran cantidad de recursos.	Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.	MySQL
Tiene un límite de 8K por fila.	Configurable	MySQL
	Gran facilidad de configuración e instalación	MySQL
	Tiene una probabilidad muy reducida de corromper los datos.	MySQL
	El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet	MySQL

Tabla 4: Cuadro Comparativo MySQL vs PostgreSQL. Autor: Alex Nieto

En el siguiente grafica se puede observar la comparación de los tres gestores de bases de datos más utilizados. Ver Ilustración 7.



²Ilustración 9: Comparación Gestores Bases de Datos. Referencia:

<http://alberto121888.files.wordpress.com/2010/08/ensayo-postgresql-vs-mysql-vs-oracle.pdf>

En conclusión, debido a que en la implementación web requiere velocidad de respuesta al momento de generar consultas se elige el gestor de bases de datos MySQL debido a que se encuentra también en el aplicativo de desarrollo XAMPP y a su vez también funciona con mayor estabilidad con el lenguaje de programación PHP.

3.1.7.5. Servidor De Aplicación.

Dentro del servidor web, como tal, se debe optar por elegir a la herramienta Apache el cual es con el que el lenguaje de programación PHP mantiene una armonía excepcional y el cual puede ser utilizado libremente ya que su licenciamiento es ³La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o simplemente sus siglas del inglés GNU GPL) es la licencia más ampliamente usada y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software.

Al ser una licencia GPL esto abarata el costo de producción debido a que no se tiene que pagar la licencia, uno de los aplicativos que contiene servidor apache para el desarrollo del proyecto es la herramienta XAMPP la cual fue escogida anteriormente para el

² (BAUTISTA, 2010)

³ (Wikipedia, Wikipedia, 2013)

desarrollo y por lo tanto, también se observa y se puede concluir que es una herramienta bastante completa para el desarrollo de aplicaciones web.

3.1.7.6. IDE Para Desarrollo.

Dentro de las herramientas IDE ⁴Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios. Tenemos una gran variedad pero por uso general y también por orden de popularidad se eligió NetBeans debido a que es el IDE que mejor se acopla al trabajo con el framework KOHANA.

NetBeans como tal es un IDE que se creó colaborativamente y el cual tiene soporte a diferencia de otros IDEs que no poseen soporte con lo cual lo hace más estable para trabajar y también porque tiene varios complementos tanto para el desarrollo web y a su vez para el desarrollo, no solo en PHP, sino para varios lenguajes de programación a fin de desarrollar aplicaciones de escritorio, móviles y de web.

3.2. Cuadro Resumen de Selección de Tecnología.

Tecnología	Elección	Razón
Arquitectura	En capas	El Proyecto se ejecutara vía Web
Lenguaje de Programación	PHP	Es un lenguaje fácil de aprender y libre
Gestor De Base de Datos	MySql	Es el mejor para este proyecto de mediano tamaño
Framework	Kohana	Es el que mejor se adapta a la arquitectura en capas y trabaja con PHP
IDE	NetBeans	Es un IDE de desarrollo libre que se adapta a las herramientas elegidas.

Tabla 5: Cuadro Resumen Selección Tecnología. Autor: Alex Nieto.

⁴ (Wikipedia, Wikipedia, 2013)

4. CAPÍTULO: LEVANTAMIENTO DEL PROCESO

4.1. Mapa de Procesos

4.1.1. Descripción

Debido a que en el proceso intervienen distintos personajes (actores), los cuales tienen diferentes permisos para el desarrollo del macro proceso, diferentes escenarios y actividades se describirán en este capítulo.

4.1.2. Actores

- Administrador.
- Capitán y Proveedor.
- Proveedor.
- Cocinero.
- Ayudante de Cocina.
- Barman.
- Ama de llaves.

4.1.3. Procesos Involucrados

A continuación se mostraran los macro procesos, con sus subprocesos del proceso de administración los insumos y víveres.

- Administrar Recursos.
 - Revisar productos.
 - Egresar productos.
 - Descartar productos.
- Administrar Pedido.
 - Generar pedido.
 - Revisar pedido.
 - Aprobar de pedido.
 - Enviar de pedido.

- Administrar Entrega.
 - Recibir pedido.
 - Obtener productos.
 - Generar despacho.
 - Entregar productos.

4.1.4. Cadenas de Valor

Para un mejor entendimiento de los procesos se muestra en el gráfico de cadena de procesos en función de orden específico de cada ciclo durante el proceso general. Ver Ilustración 8.

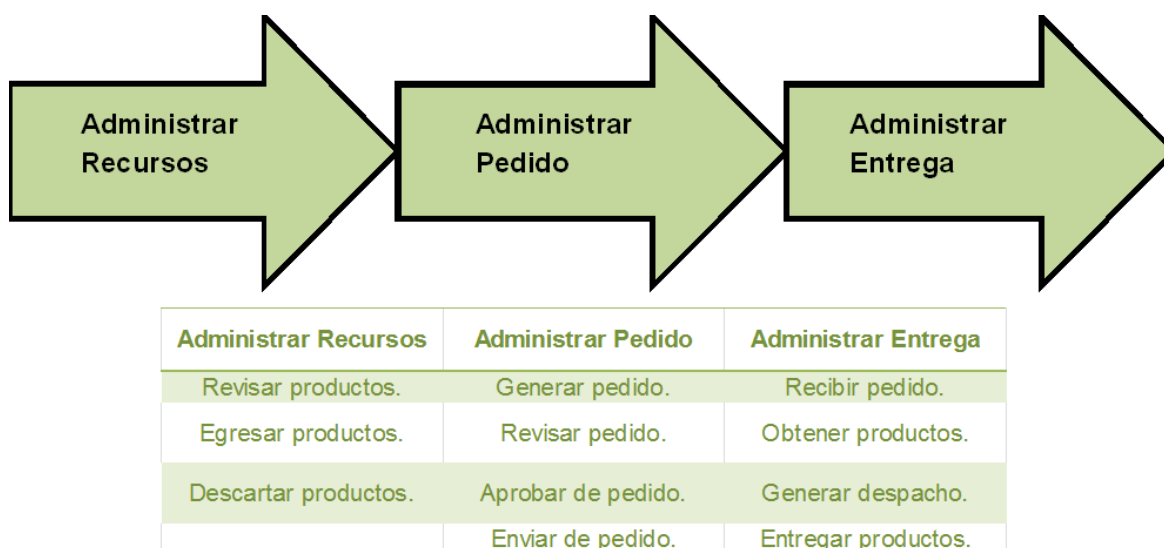


Ilustración 10: Cadena De Valor. Autor: Alex Nieto.

4.1.4.1. Procesos Centrales

- Administrar Recursos.
- Administrar Pedido.
- Administrar Entrega.

4.1.4.2. Procesos de Soporte

- Revisar Productos.
- Egresar Productos.
- Descartar Productos.
- Generar Pedido.

- Revisar Pedido.
- Aprobar Pedido.
- Enviar Pedido.
- Recibir Pedido.
- Obtener Productos.
- Generar Despacho.
- Entregar Productos.

4.1.5. Mapa de procesos.

A continuación se presenta el mapa de procesos:



Ilustración 11: Mapa De Procesos. Autor: Alex Nieto.

4.1.6. Gráficos de Soporte.

4.1.6.1. Gráfico Proceso Administrar Recursos.



Ilustración 12: Proceso Administrar Recursos. Autor: Alex Nieto.

Nombre Proceso: Administrar Pedido.

Propósito: Automatizar el proceso revisión de inventario.

Actividades: Generar Informe.

Unidad Organizacional: Bodega.

E1: Código Producto.

E2: Datos Usuario.

E3: Datos Producto.

C1: Formato de presentación.

C2: Condiciones del producto.

M1: Bodega.

M2: Tecnología.

M3: Recursos Humanos.

S1: Lista de productos.

4.1.6.2. Gráfico Proceso Administrar Pedido.

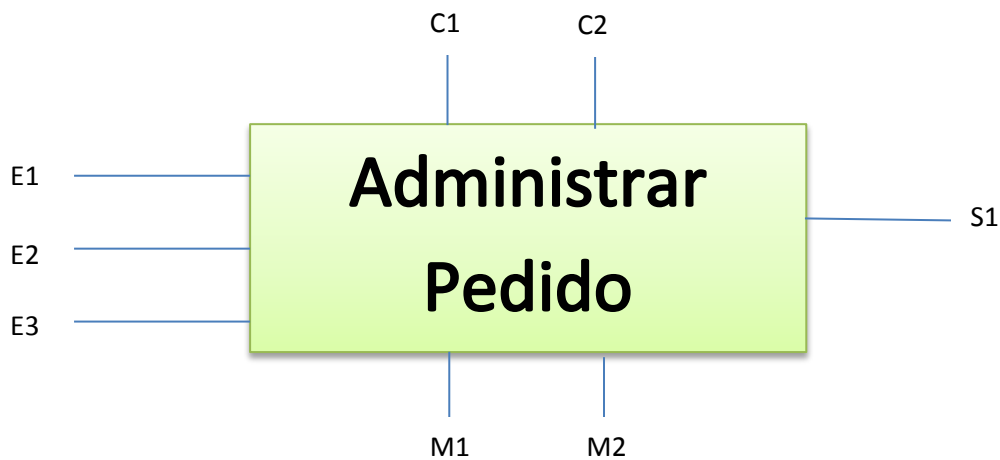


Ilustración 13: Proceso Administrar Pedido. Autor: Alex Nieto.

Nombre Proceso: Administrar Pedido.

Propósito: Automatizar el proceso de generación de pedidos.

Actividades: Generar Pedido.

Unidad Organizacional: Administración local.

E1: Código Pedido.

E2: Datos Usuario (realiza petición).

E3: Datos Usuario (capitán).

C1: Formato de presentación.

C2: Condiciones del producto.

M1: Administración.

M2: Tecnología.

M3: Recursos Humanos.

S1: Documento de pedido.

4.1.6.1. Gráfico Proceso Administrar Entrega.

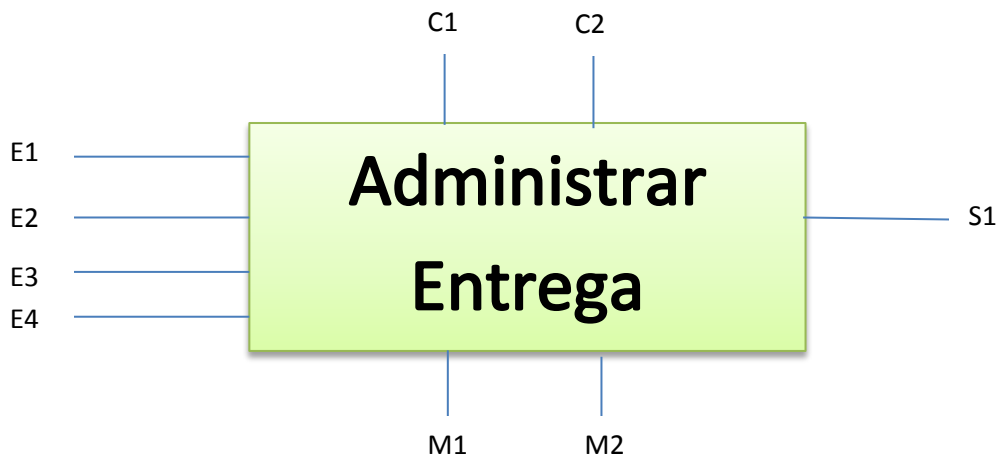


Ilustración 14: Proceso Administrar Entrega. Autor: Alex Nieto.

Nombre Proceso: Administrar Entrega.

Propósito: Automatizar el proceso de entrega de productos.

Actividades: Generar Despacho.

Unidad Organizacional: Proveedor.

E1: Código Despacho.

E2: Datos Usuario.

E3: Código Producto.

E4: Datos Producto.

C1: Formato de presentación.

C2: Condiciones del producto.

M1: Proveedor.

M2: Tecnología.

M3: Recursos Humanos.

S1: Documento de despacho.

4.2. SIPOC

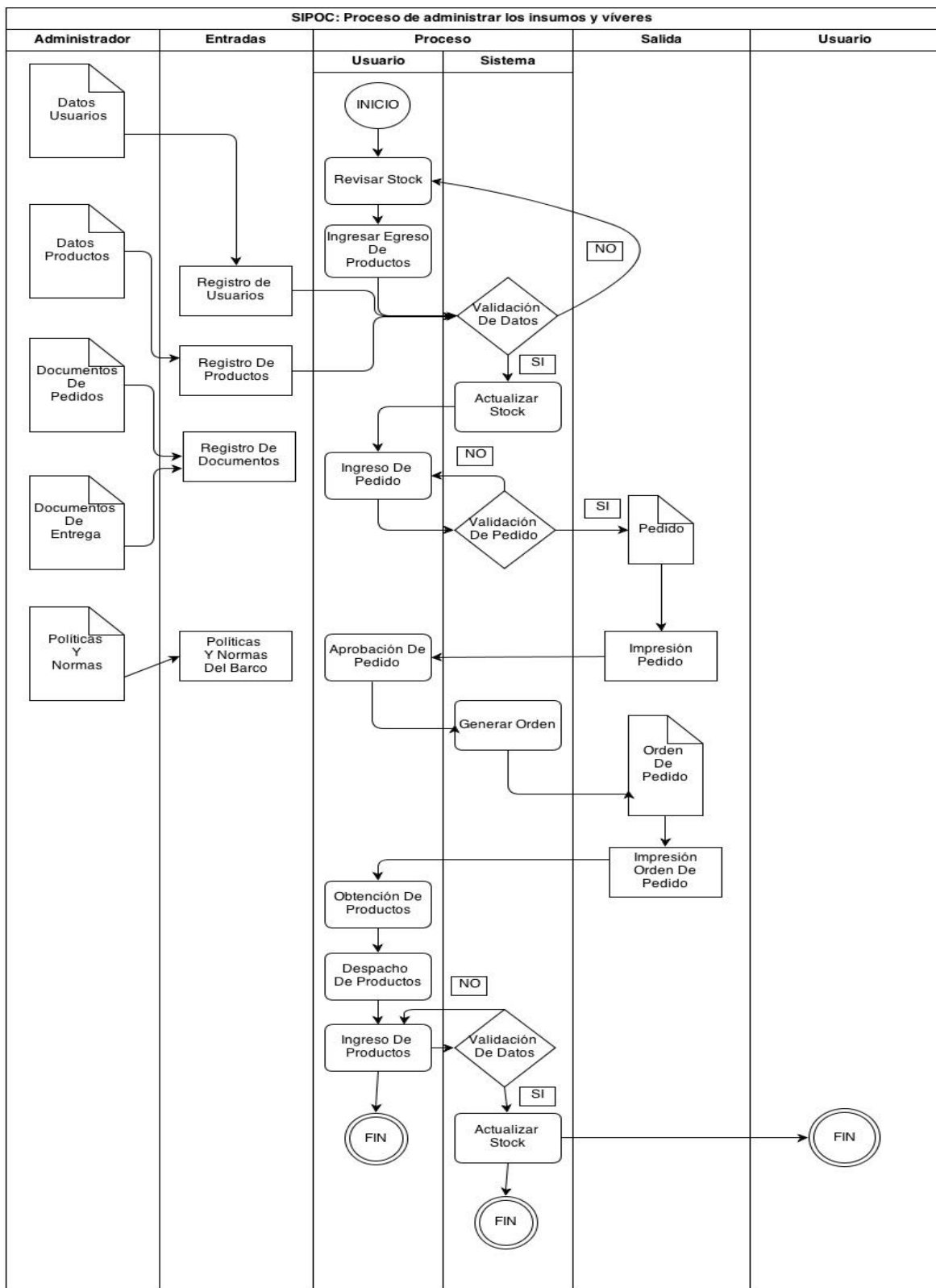


Ilustración 15: SIPOC. Autor: Alex Nieto

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO.
 APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

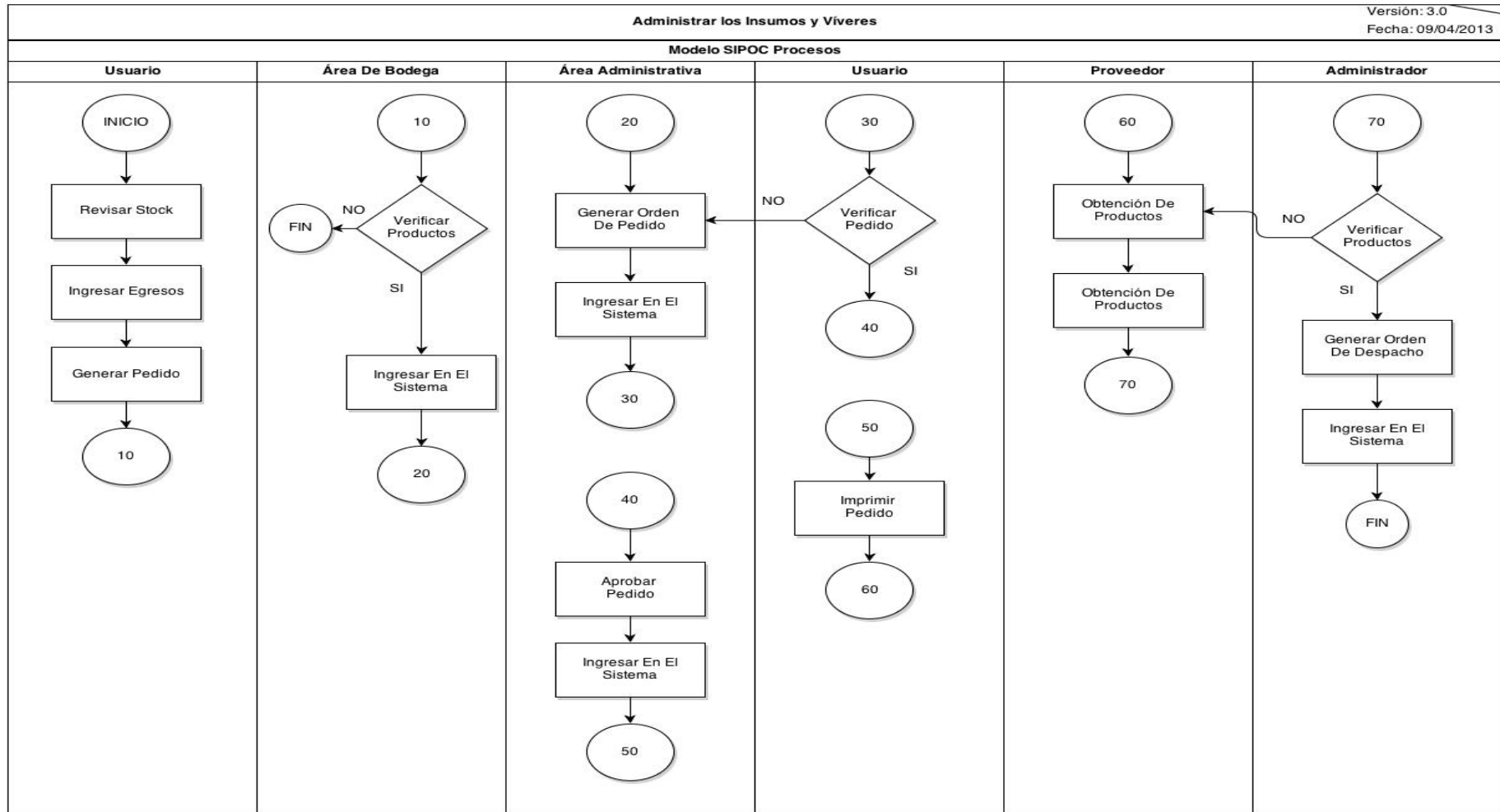


Ilustración 16: Modelo SIPOC Procesos. Autor: Alex Nieto.

5. CAPÍTULO: CONSTRUCCIÓN DEL PRODUCTO

5.1. Inicio

A continuación se presentarán los diferentes diagramas descriptivos para el desarrollo de la aplicación final con la finalidad de hacer entender a detalle las funciones de la misma.

5.1.1. *Análisis De Requerimientos*

Dentro del levantamiento podremos conocer en detalle los aspectos más importantes para el desarrollo del mismo los cuales son:

- Identificación de actores.
- Especificación de funcionalidades.
- Arquitectura.
- Diseño.

5.1.2. *Identificación De Actores.*

Los actores principales que se pudieron identificar en el sistema son puntualmente cinco tipos de actores que intervienen en el sistema completo.

Administrador: La persona encargada de administrar el sistema; siendo el responsable de la administración de productos y usuarios en el sistema

Solicitante: Las personas que generan una solicitud con productos que se requieren en inventario o stock.

Administrador de pedidos: Las personas que son responsables de revisar el pedido y de ponerlo en estado de aprobado con lo cual pasa a ser solicitado al proveedor.

Controlador: Las personas que son los encargados de ver el estado de los pedidos generados después de que el estado pasa han aprobado.

Proveedor: Las personas encargadas de la adquisición de productos los cuales serán posteriormente enviados a su destino de bodega para ingreso en stock; son los encargados de que el pedido pase por los estados: comprado, enviado y entregado una vez que llegue a su destino.

Los detalles se pueden observar en la Ilustración 15.

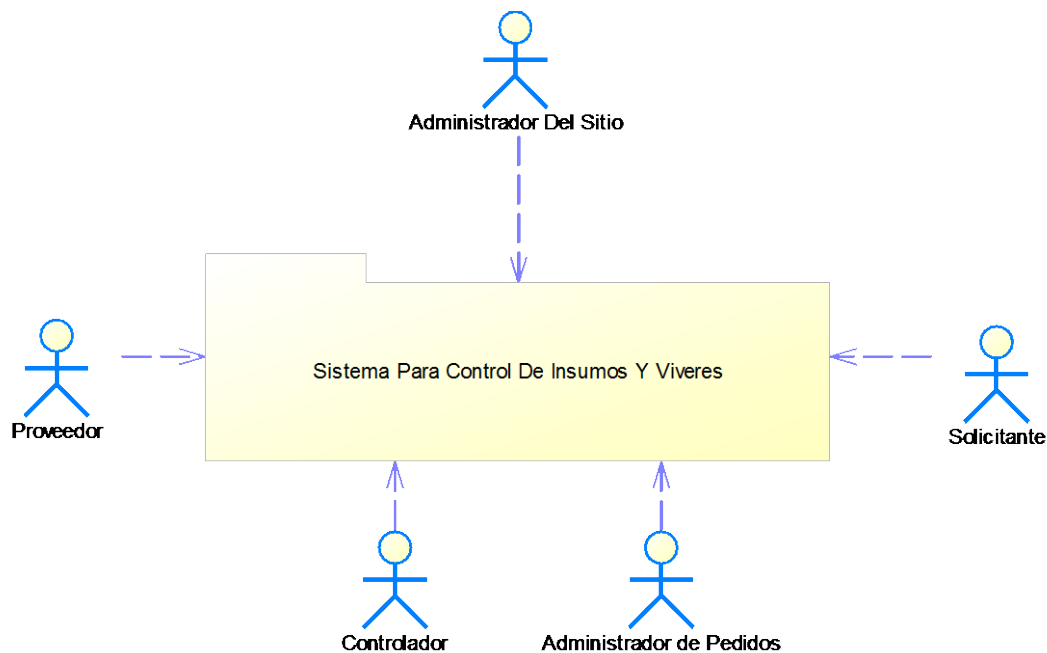


Ilustración 17: Actores. Autor: Alex Nieto.

5.1.3. Especificaciones De Requerimientos

5.1.3.1. Funcionales

- El sistema administrará el control de insumos y víveres dentro de un barco turístico.
- El sistema generará órdenes de pedidos para la requisición de productos próximos a agotarse o agotados en bodega (inventario).
- El sistema generará estados para los pedidos.
- El sistema generará órdenes de despacho para el cumplimiento de los pedidos generados y aprobados.
- El sistema registrará a los actores involucrados durante todo el proceso de adquisiciones.
- El sistema registrará todos los productos que consten en inventario con su respectiva cantidad disponible.
- El sistema descontará productos del stock dependiendo de los gastos por parte del personal.
- El sistema generará un informe en tiempo real sobre los estados de los pedidos generados.
- El sistema generará reportes sobre los estados de los pedidos a fin de mes.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

- El sistema permitirá ahorrar el tiempo de respuesta entre pedidos y adquisiciones.
- El sistema generará un informe con la cantidad de productos en tiempo real de los productos en stock.

5.1.3.2. *No Funcionales*

Tecnología

- Computador personal.
- Tablet.
- Celular.

Licenciamiento

El sistema contiene los siguientes componentes:

Lenguaje: Php

BDD: MySql

Lo que implica un licenciamiento open source para para el lenguaje de programación; y un licenciamiento GNU para la base de datos; lo cual permite libre uso sin paga por su uso.

Requerimientos

Cliente:

Hardware	Navegador Web
Computador <ul style="list-style-type: none">• Memoria RAM: 256 MB• Procesador: Núcleo Simple	Internet Explorer. Google Chrome Mozilla Firefox

Tabla 6: Tabla Requerimientos Cliente. Autor: Alex Nieto.

Servidor:

Hardware	Sistemas Operativos	Software
Memoria RAM: 256 MB	Windows Server 2008	MySql5.0 o superior
Espacio en Disco: 1 GB	Windows Server 2012	Php 5.3 o superior
Procesador: Núcleo Simple	Linux (Recomendado)	

Tabla 7: Tabla Requerimientos Servidor. Autor: Alex Nieto.

5.1.4. Arquitectura

El sistema será construido bajo una arquitectura en tres capas en un ambiente o trabajo de cliente servidor. (Ver Ilustración 16).



Ilustración 18: Arquitectura del Producto. Autor: Alex Nieto

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO.
 APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

5.1.5. Diseño.

5.1.5.1. Diseño Entidad/Relación (E/R).

En el siguiente cuadro se explica la relación entre las tablas para que se impactaran en la base de datos para el uso del mismo, nótese que todas las tablas guardan relación entre sí, esto quiere decir, que no puede crearse un elemento sin antes tener otro antes creado. Ver Ilustración 20

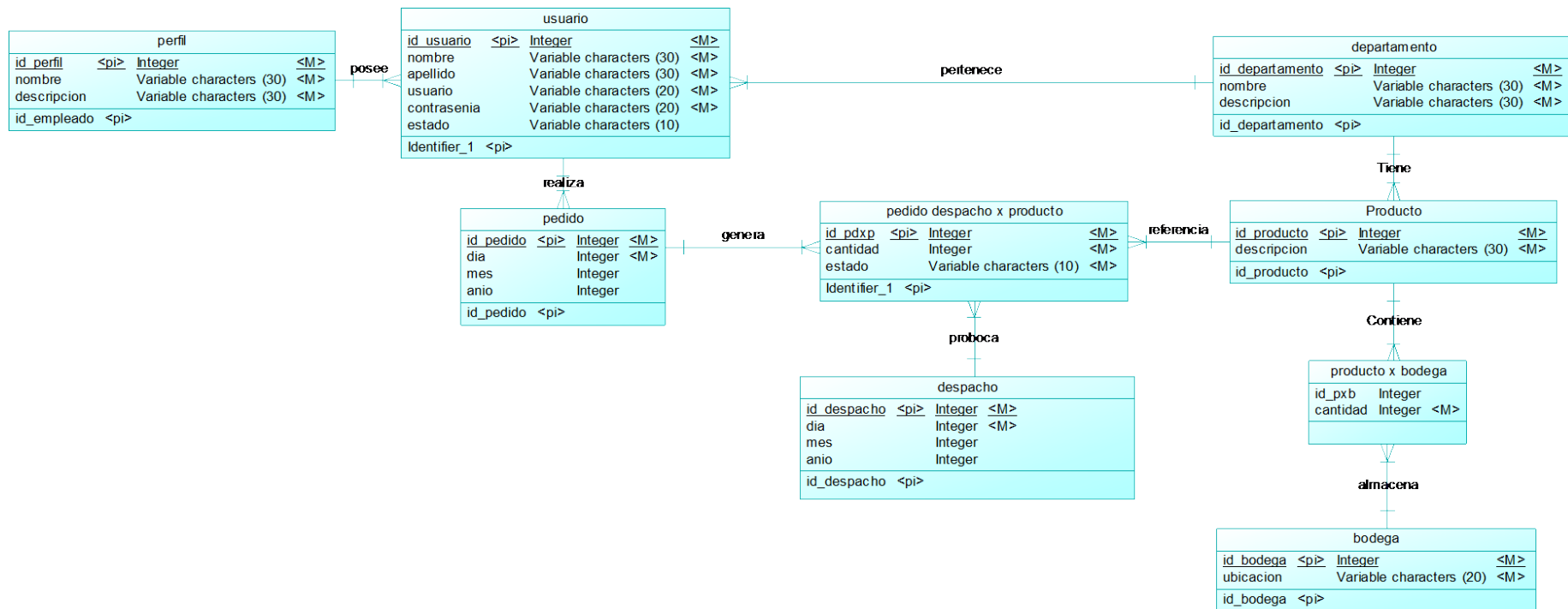


Ilustración 19: Diseño E/R. Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO.
 APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

5.1.5.2. Diagramas UML

Clase DP.

En el diagrama se puede comprender la manera en la que el DP (Data Process) interactúa en el sistema a través de métodos ya prediseñados para sus funciones en el sistema completo. Ver Ilustración 21.

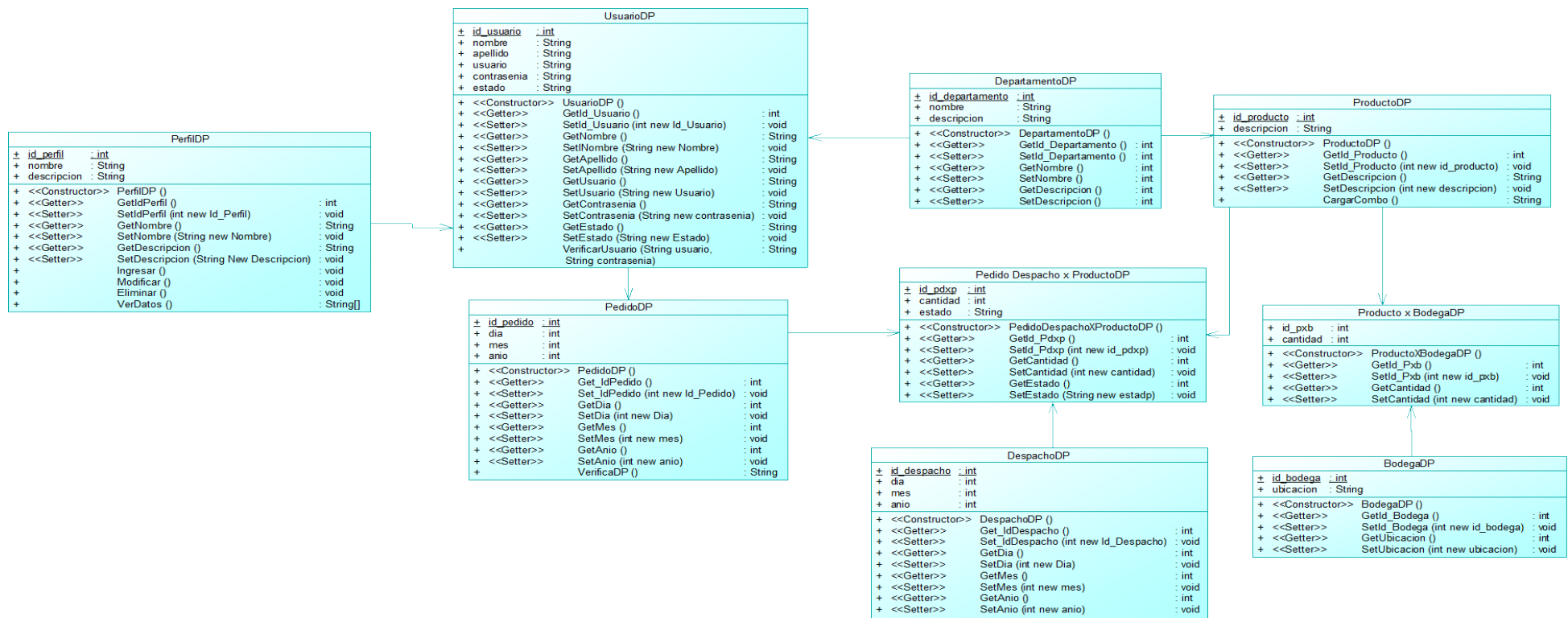


Ilustración 20: Diagrama DP. Autor: Alex Nieto

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO.
 APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Clase MD.

En el diagrama se puede comprender la manera en la que el MD (Management Data) interactúa con la parte de manejo de datos los cuales van relacionados directamente con la base de datos. Ver Ilustración 20.

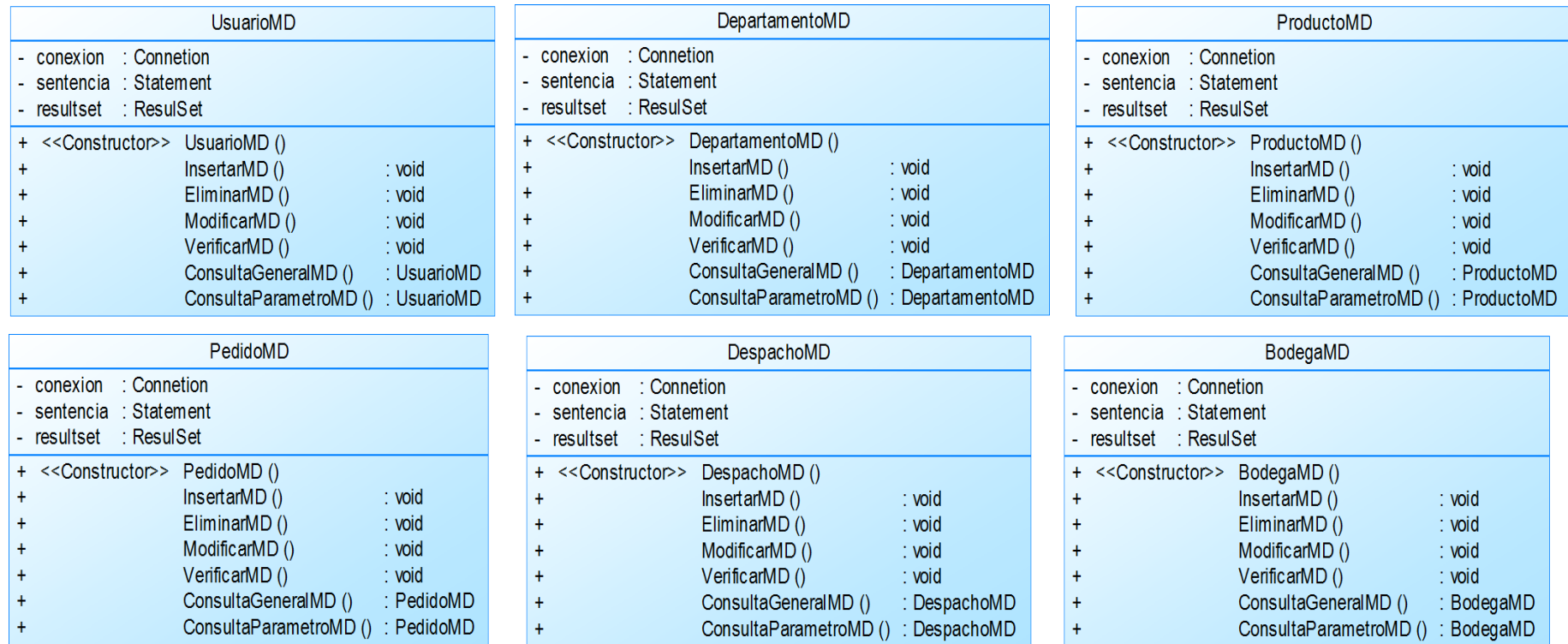


Ilustración 21: Diagrama MD. Autor: Alex Nieto

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO.
 APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Clase GUI.

En el diagrama se puede comprender la manera en que esta empaquetado el GUI (Graphical User Interface) así como el despliegue de esta interfaz dependiente de una interfaz superior. Ver Ilustración 23.

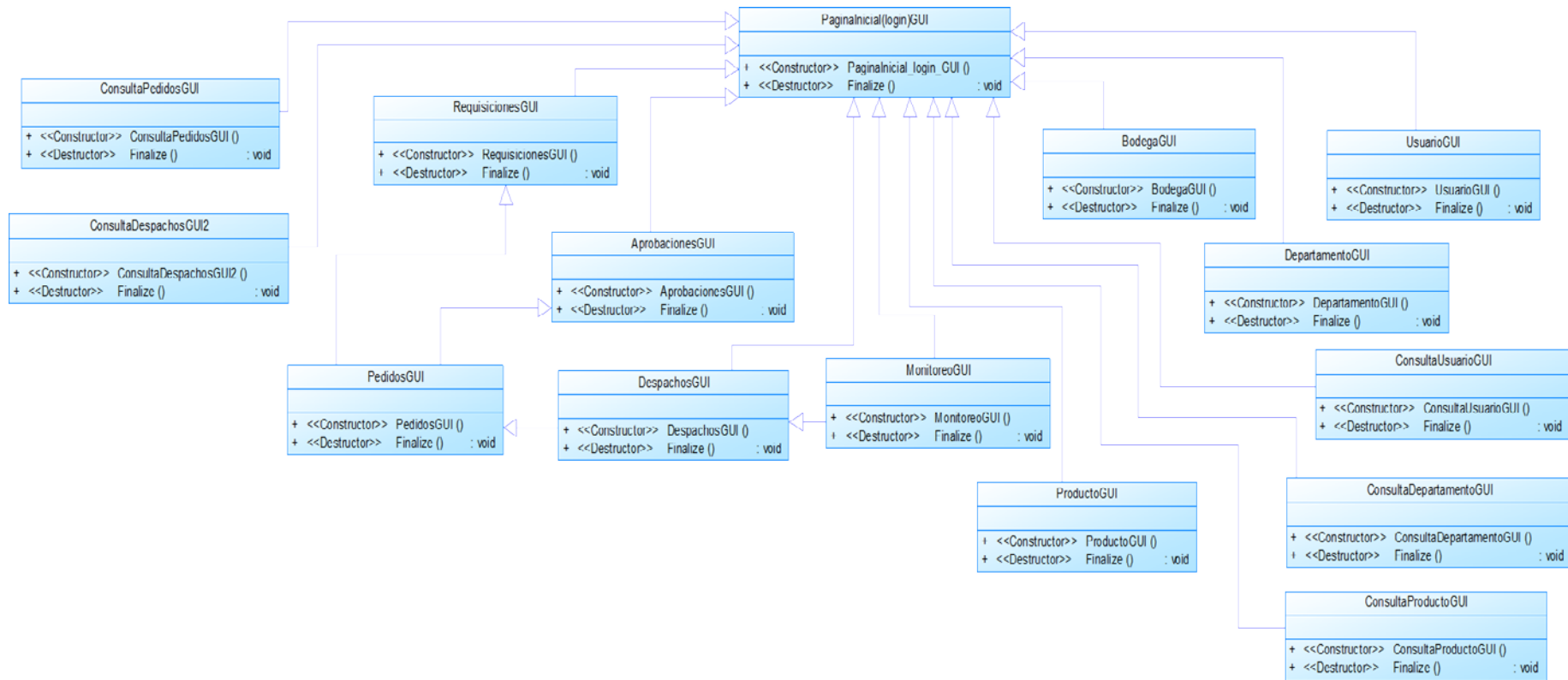


Ilustración 22: Diagrama GUI. Autor: Alex Nieto.

Diagramas de Despliegue:

En el diagrama se puede comprender la cantidad de servidores que se presentan en el sistema y su interacción entre sí para el manejo de los procesos. Ver Ilustración 24.

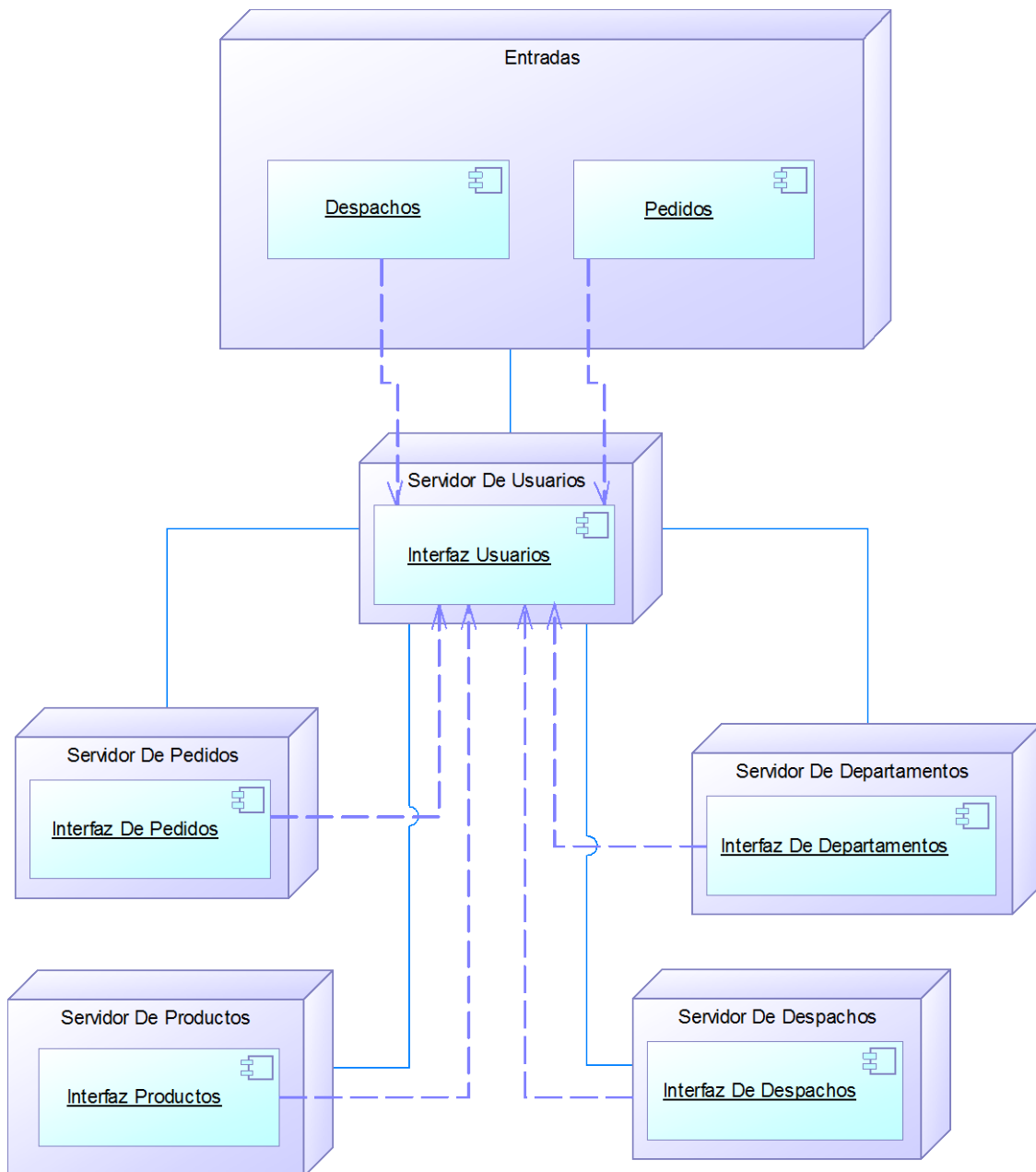


Ilustración 23: Diagrama de Despliegue. Autor: Alex Nieto.

Diagramas de Secuencia:

Proceso: Administración de Productos (Ingreso de Productos).

En el diagrama se presenta al administrador del sitio cómo interactúa con el programa para el ingreso de productos para generar pedidos. Ver Ilustración 25.

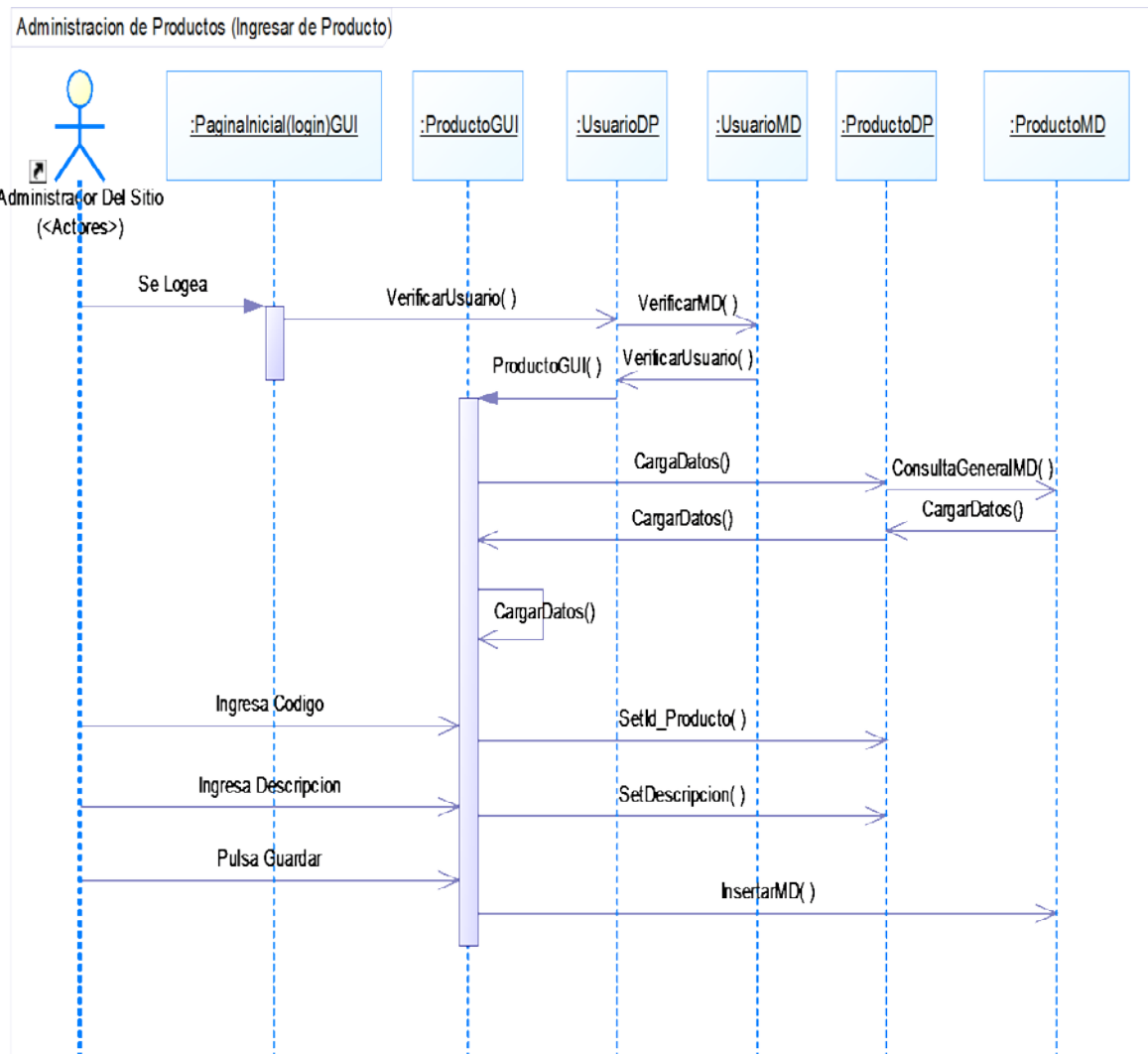


Ilustración 24: Ingreso de Productos. Autor: Alex Nieto.

Proceso: Administración de Productos (Modificar de Productos).

En el diagrama se presenta al administrador del sitio cómo interactúa con el programa para la modificación de productos. Ver Ilustración 26.

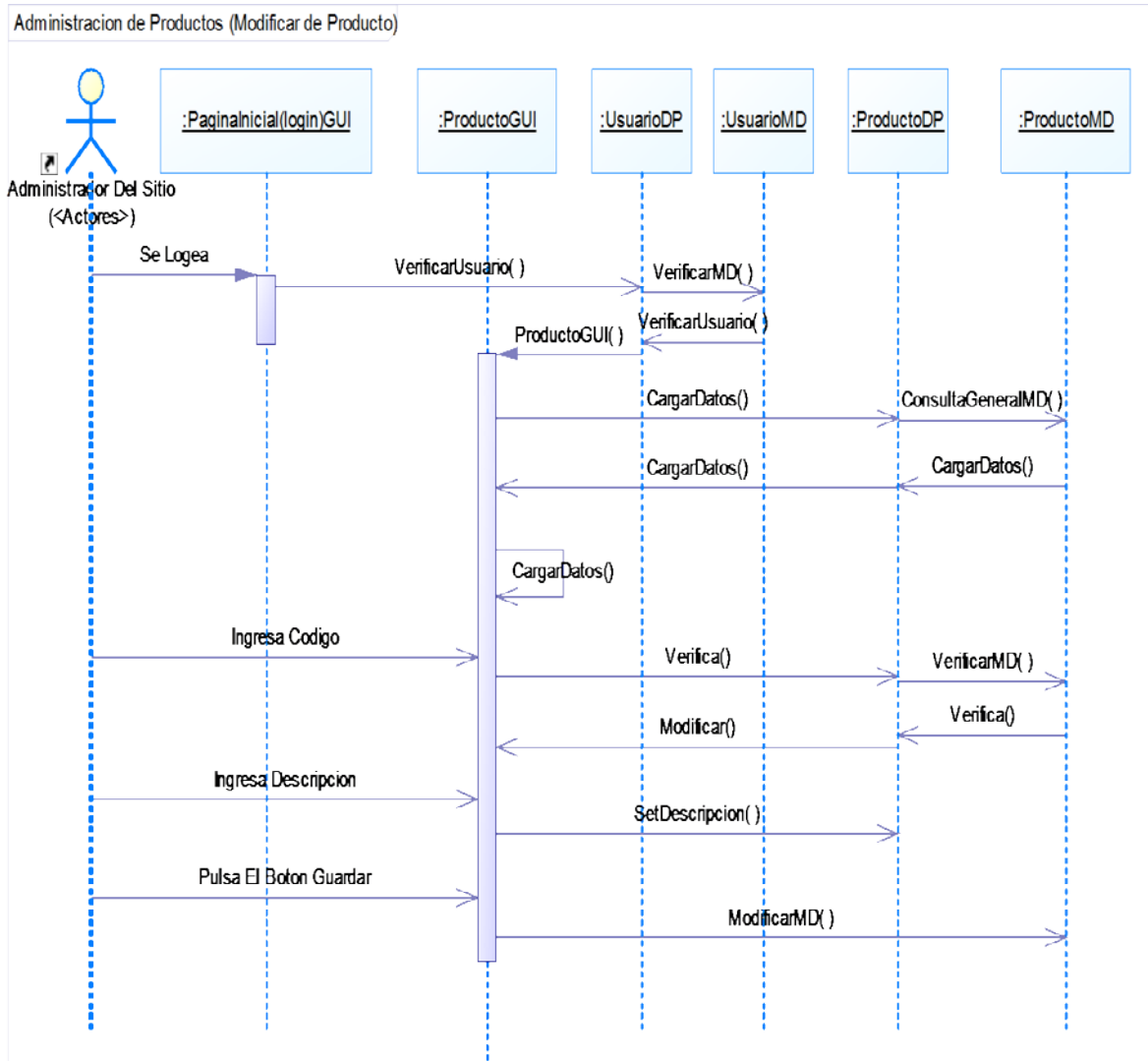


Ilustración 25: Modificar Producto. Autor: Alex Nieto.

Proceso: Administración de Productos (Eliminar Productos).

En el diagrama se presenta al administrador del sitio cómo interactúa con el programa para la eliminación de productos. Ver Ilustración 27.

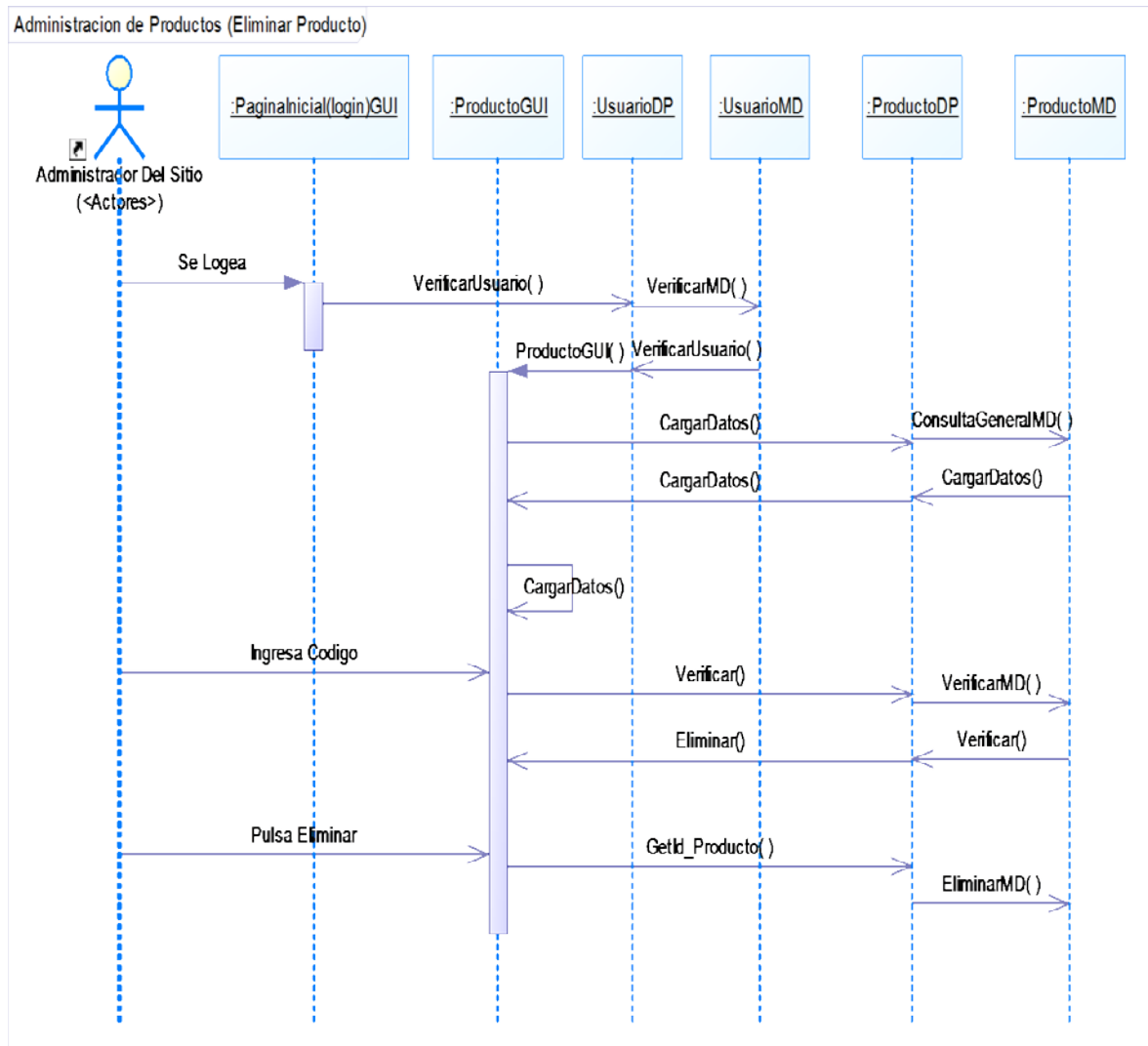


Ilustración 26: Eliminar Producto. Autor: Alex Nieto.

Proceso: Administración de Productos (Consulta General).

En el diagrama se presenta al administrador del sitio cómo interactúa con el programa para la consulta de productos. Ver Ilustración 28.

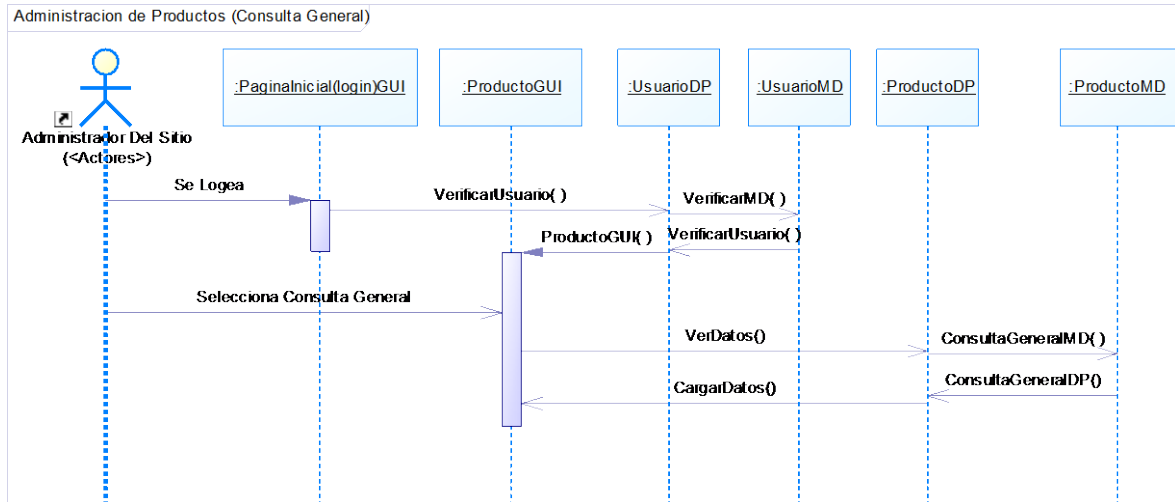


Ilustración 27: Consulta General Productos. Autor: Alex Nieto.

Proceso: Administración de Productos (Consulta por Parámetros).

En el diagrama se presenta al solicitante cómo interactúa con el programa para la consulta de productos para generar pedidos. Ver Ilustración 29.

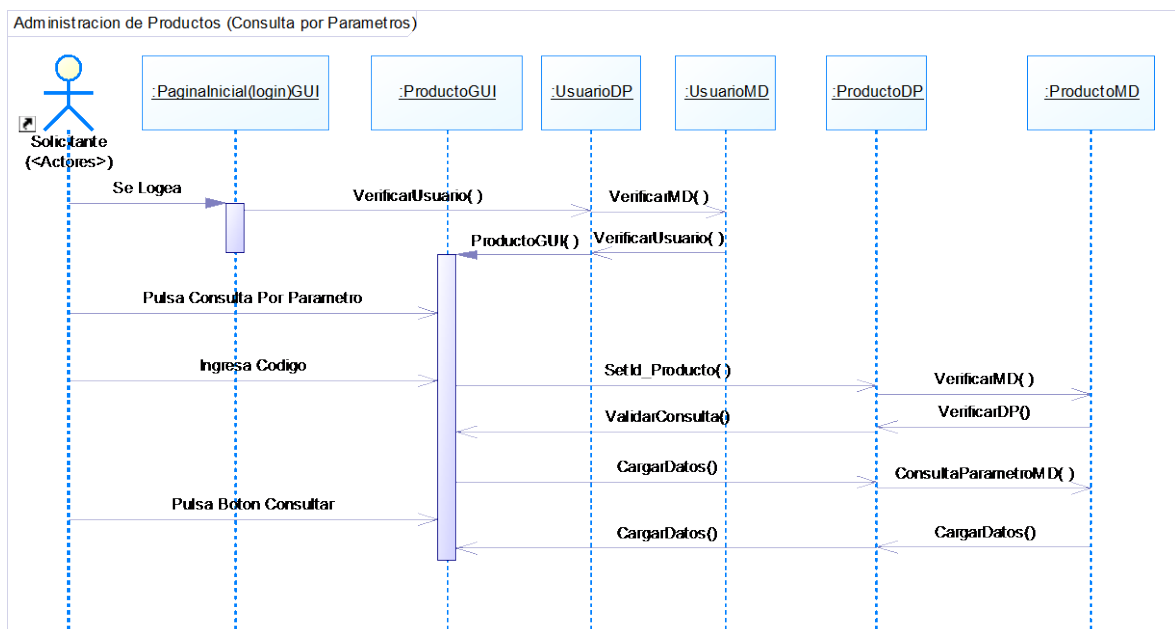


Ilustración 28: Consulta Parámetros Productos. Autor: Alex Nieto.

Proceso: Administración de Pedidos (Ingresar Pedido).

En el diagrama se presenta al solicitante como interactúa con el programa para generar pedidos. Ver Ilustración 30.

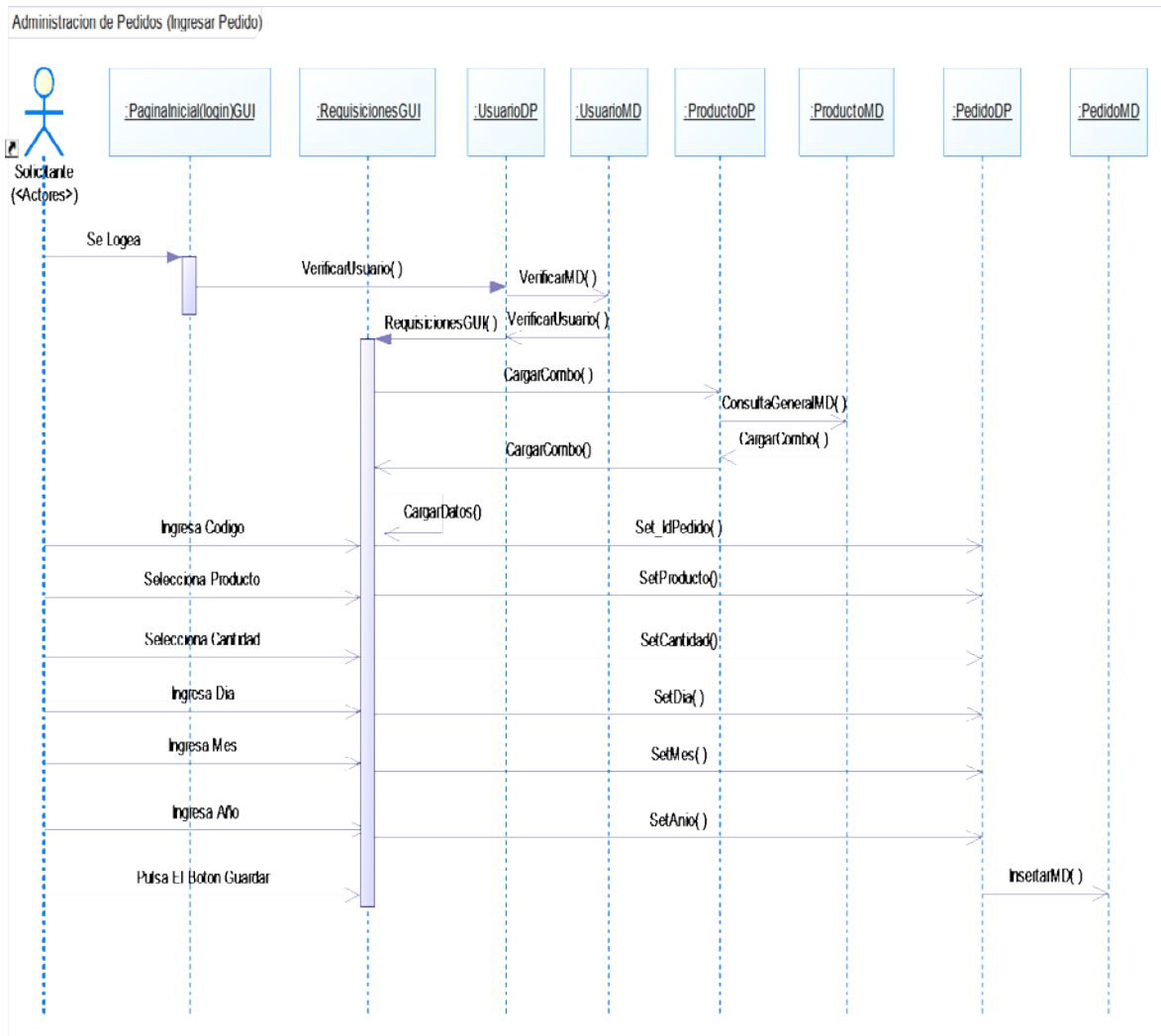


Ilustración 29: Ingresar Pedido. Autor: Alex Nieto.

Proceso: Administración de Pedidos (Modificar Pedido).

En el diagrama se presenta al solicitante cómo interactúa con el programa para modificar un pedido. Ver Ilustración 31.

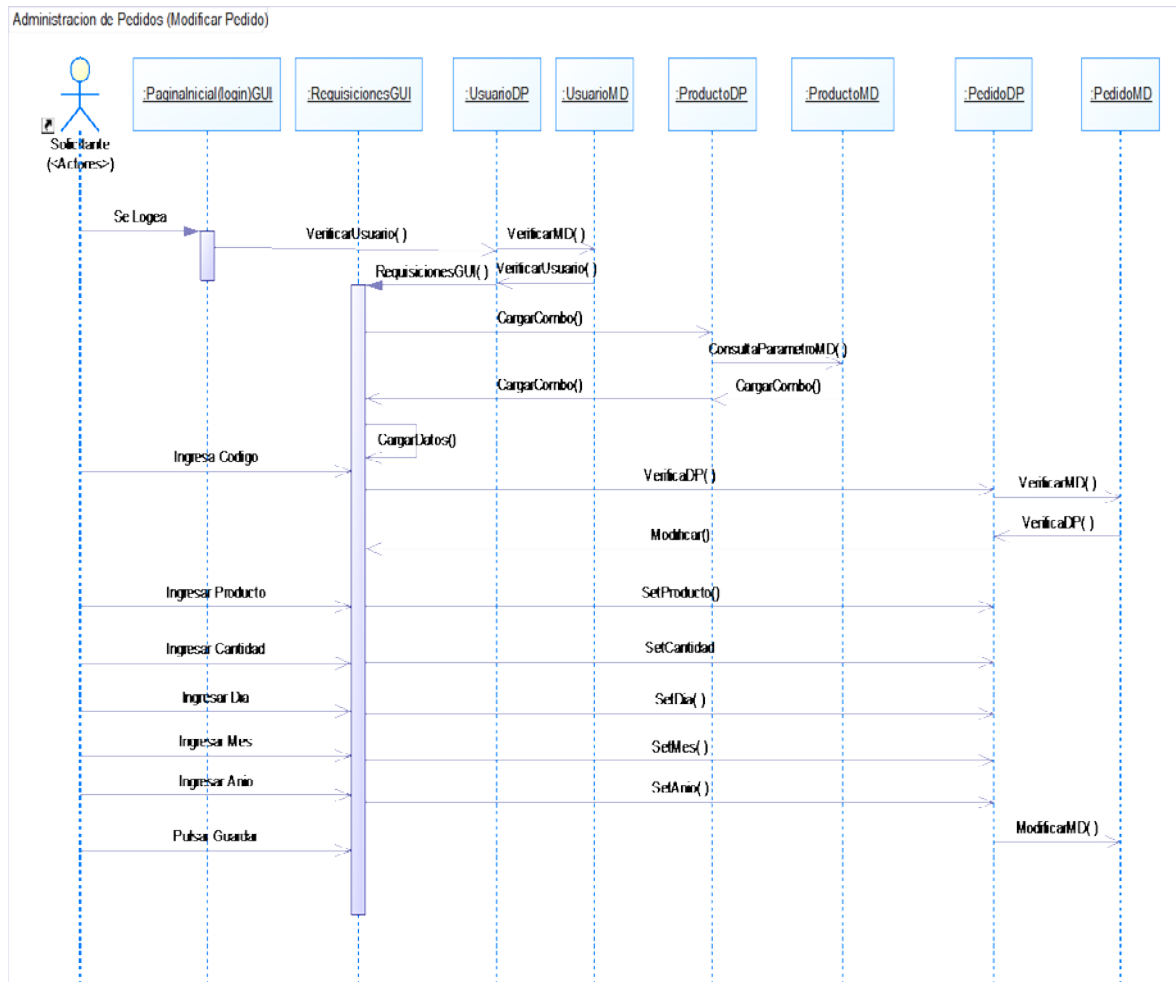


Ilustración 30: Modificar Pedido. Autor: Alex Nieto.

Proceso: Administración de Pedidos (Eliminar Pedido).

En el diagrama se presenta al administrador de pedidos cómo interactúa con el programa para la eliminación de pedidos. Ver Ilustración 32.

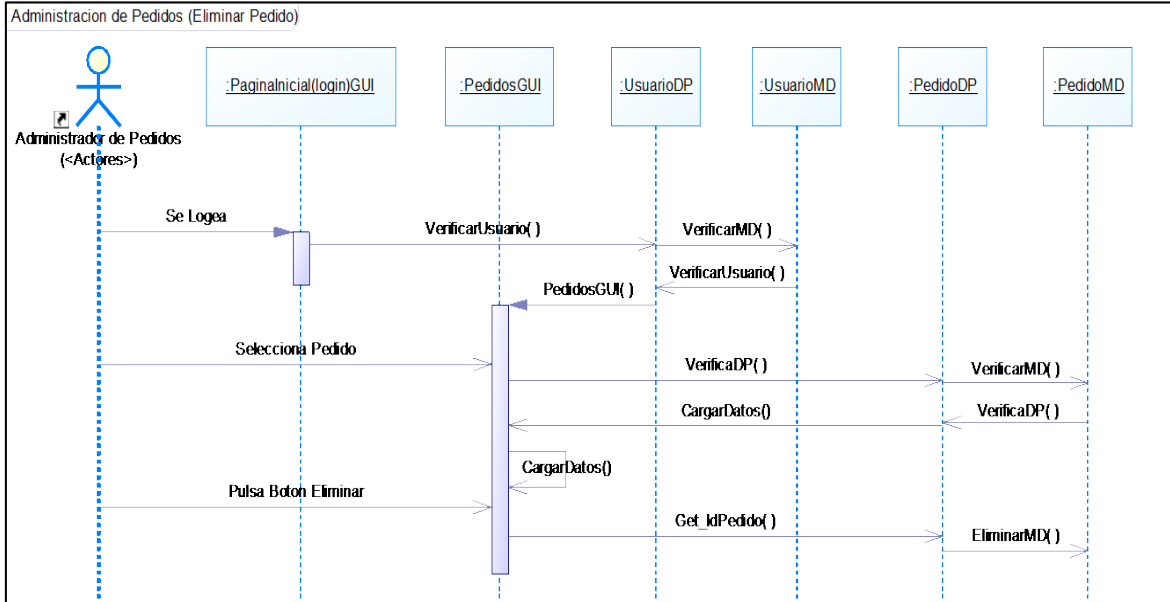


Ilustración 31: Eliminar Pedido. Autor: Alex Nieto.

Proceso: Administración de Pedidos (Consulta General).

En el diagrama se presenta al administrador de pedidos cómo interactúa con el programa para la consulta general de pedidos. Ver Ilustración 33.

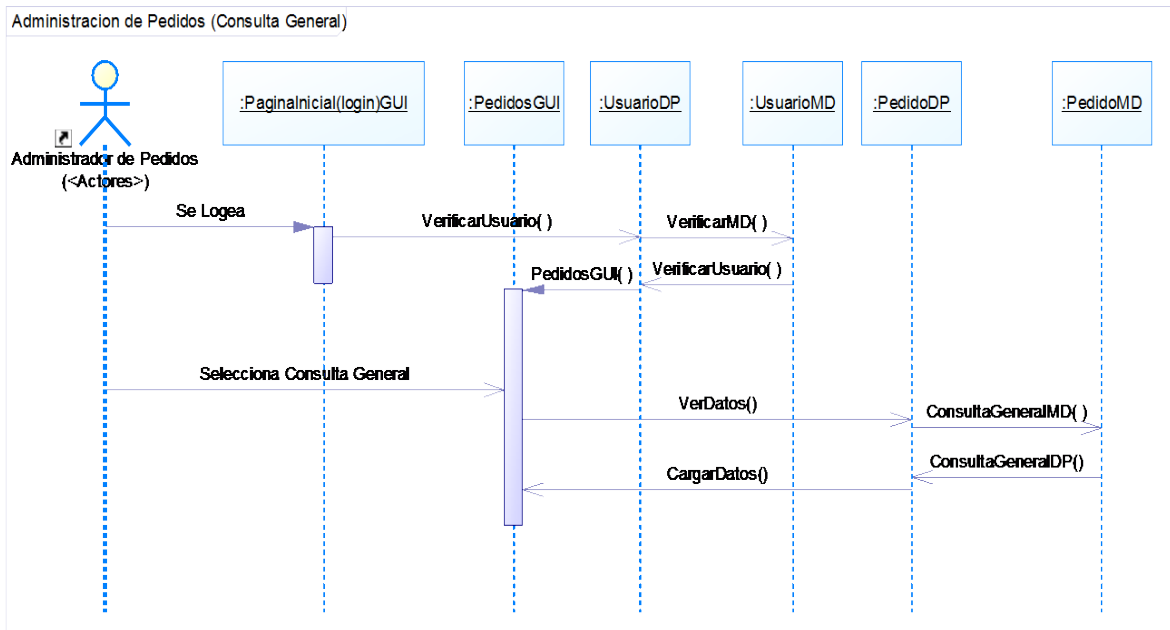


Ilustración 32: Consulta General Pedidos. Autor: Alex Nieto.

Proceso: Administración de Pedidos (Consulta por Parámetro).

En el diagrama se presenta al administrador de pedidos cómo interactúa con el programa para la consulta por parámetros pedidos. Ver Ilustración 34.

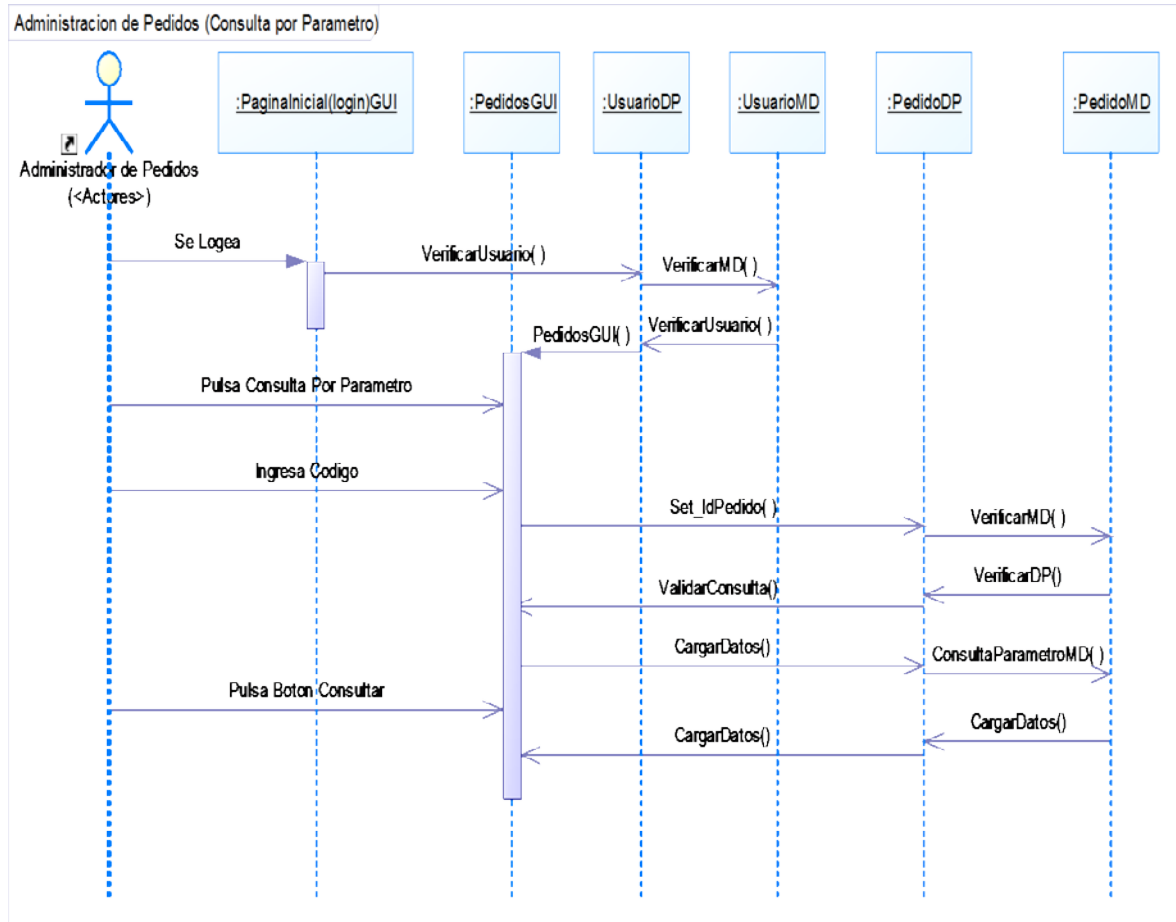


Ilustración 33: Consulta Por Parámetro Pedidos. Autor: Alex Nieto.

Proceso: Administración Pedidos (Modificar Estado).

En el diagrama se presenta al administrador de pedidos cómo interactúa con el programa para cambiar el estado de los pedidos (aprobar pedidos o rechazar pedidos). Ver Ilustración 35.

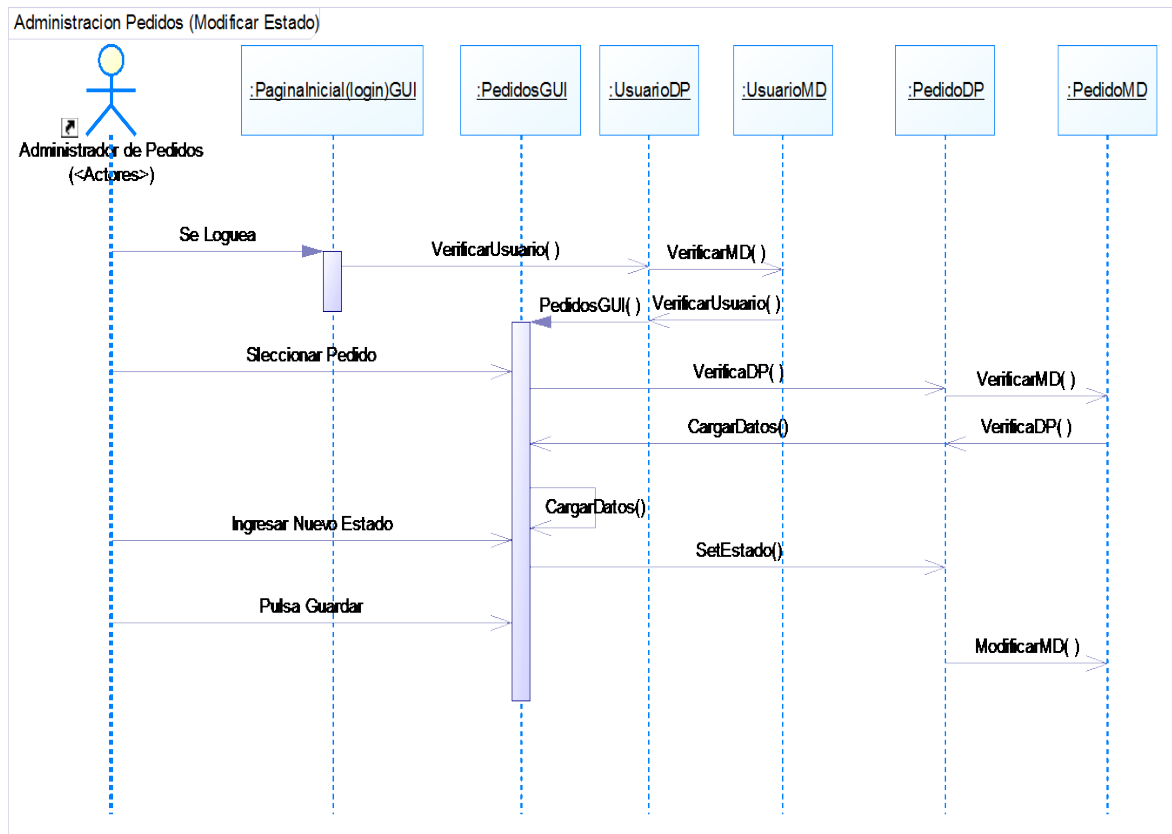


Ilustración 34: Modificar Estado Pedidos. Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Proceso: Administración de Despachos (Ingresar Despacho).

En el diagrama se presenta al proveedor cómo interactúa con el programa para el ingreso de un nuevo despacho. Ver Ilustración 36.

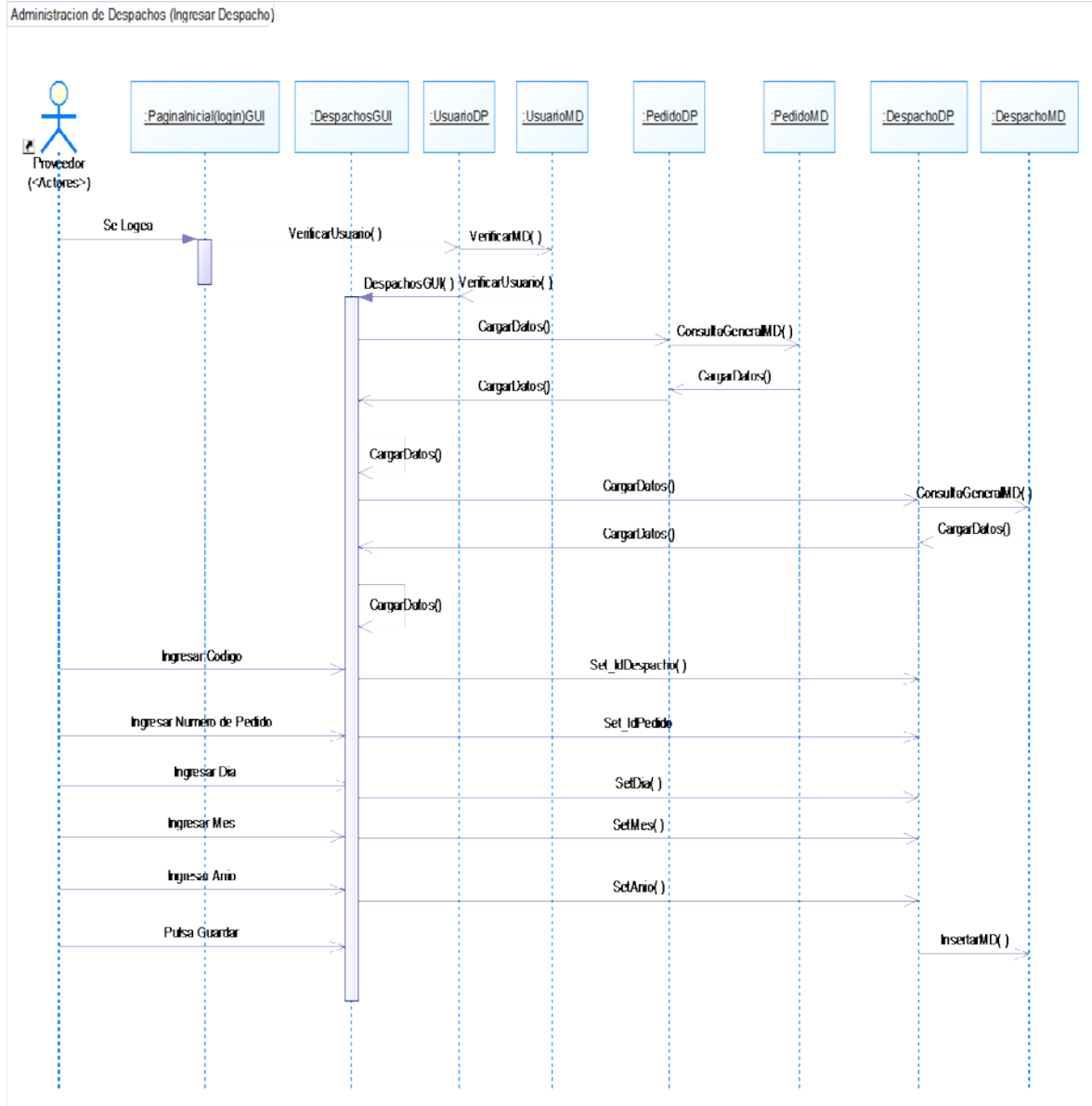


Ilustración 35: Ingresar Despacho. Autor: Alex Nieto

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Proceso: Administración de Despachos (Modificar Despacho).

En el diagrama se presenta al proveedor cómo interactúa con el programa para la modificación de despachos. Ver Ilustración 37.

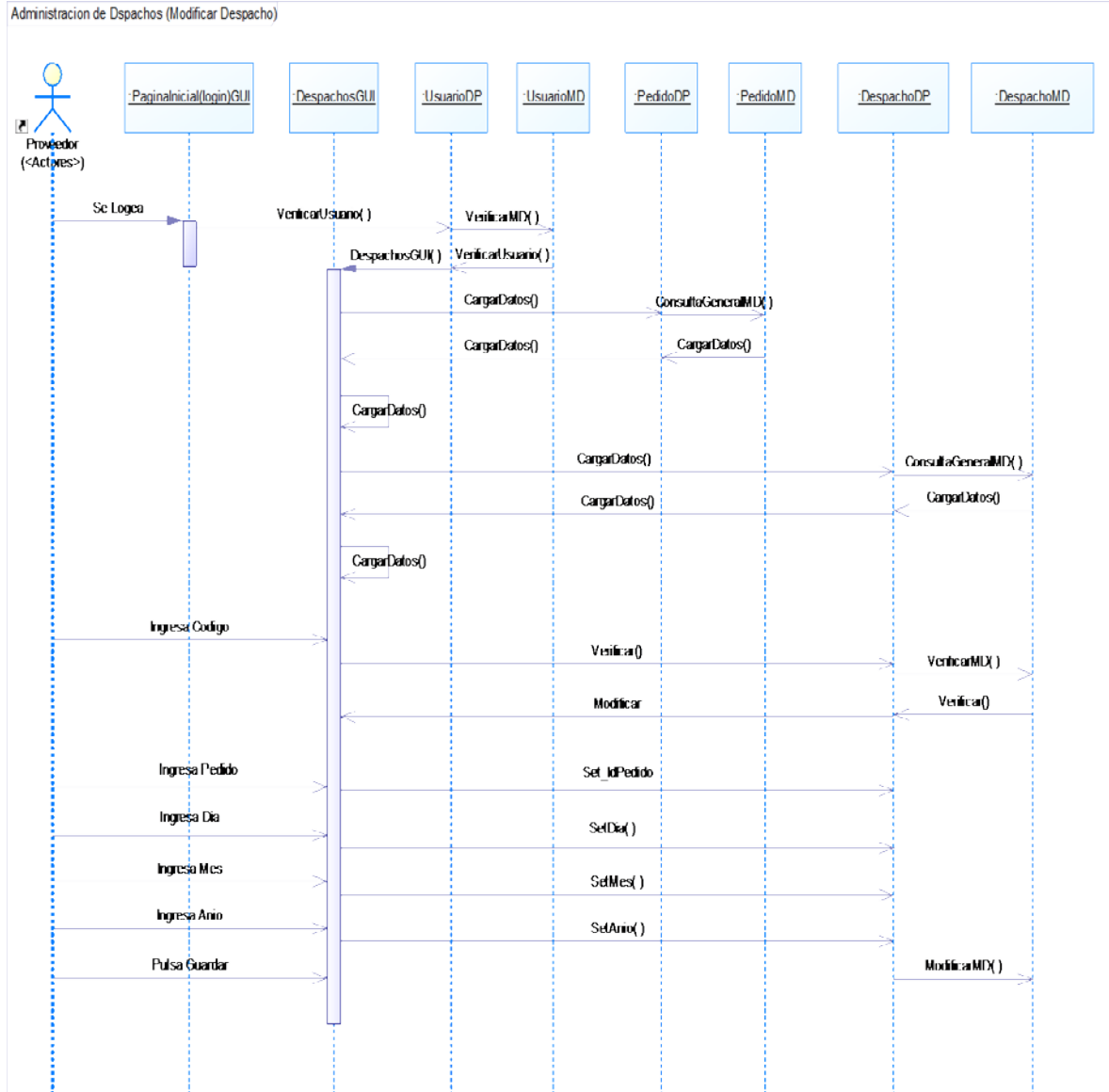


Ilustración 36: Modificar Despacho. Autor: Alex Nieto.

Proceso: Administración de Despachos (Eliminar Despacho).

En el diagrama se presenta al proveedor cómo interactúa con el programa para la eliminación de despachos. Ver Ilustración 38.

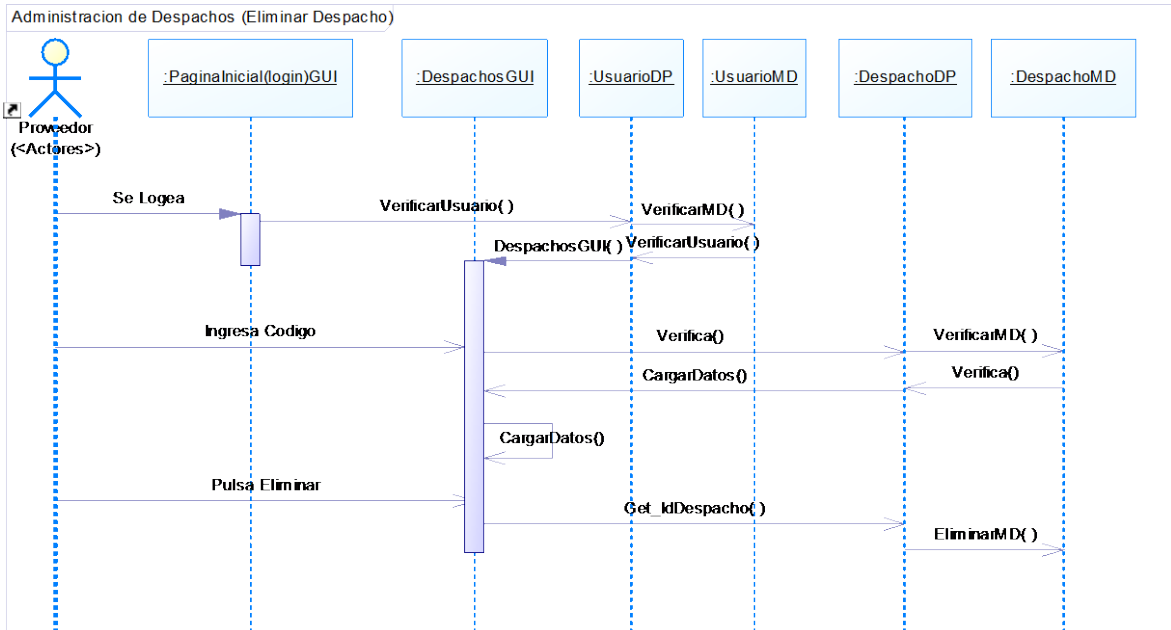


Ilustración 37: Eliminar Despacho. Autor: Alex Nieto.

Proceso: Administración de Despachos (Consulta General).

En el diagrama se presenta al proveedor cómo interactúa con el programa para la consulta general de despachos. Ver Ilustración 39.

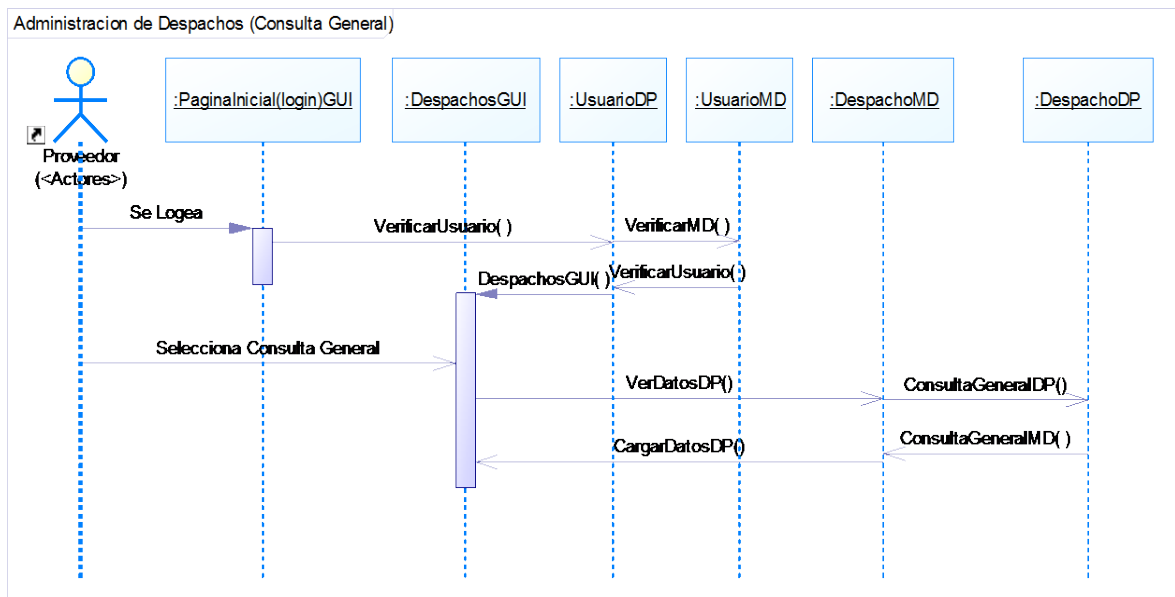


Ilustración 38: Consulta General Despachos. Autor: Alex Nieto.

Proceso: Administración de Despachos (Consulta por Parámetro).

En el diagrama se presenta al proveedor cómo interactúa con el programa para la consulta de despachos por parámetros. Ver Ilustración 40.

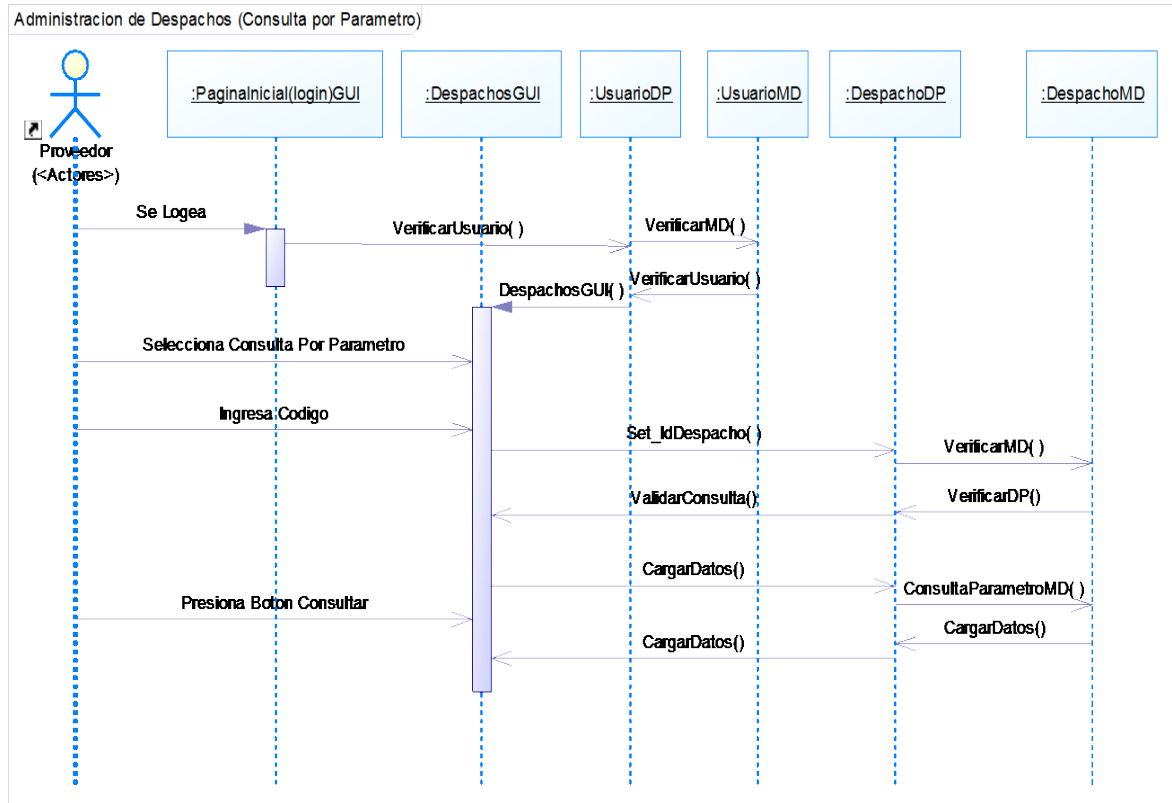


Ilustración 39: Consulta Por Parámetro Despachos. Autor: Alex Nieto.

Proceso: Administración Despachos (Modificar Estado).

En el diagrama se presenta al proveedor cómo interactúa con el programa para el cambio del estado de los despachos (comprado, enviado, entregado). Ver Ilustración 41.

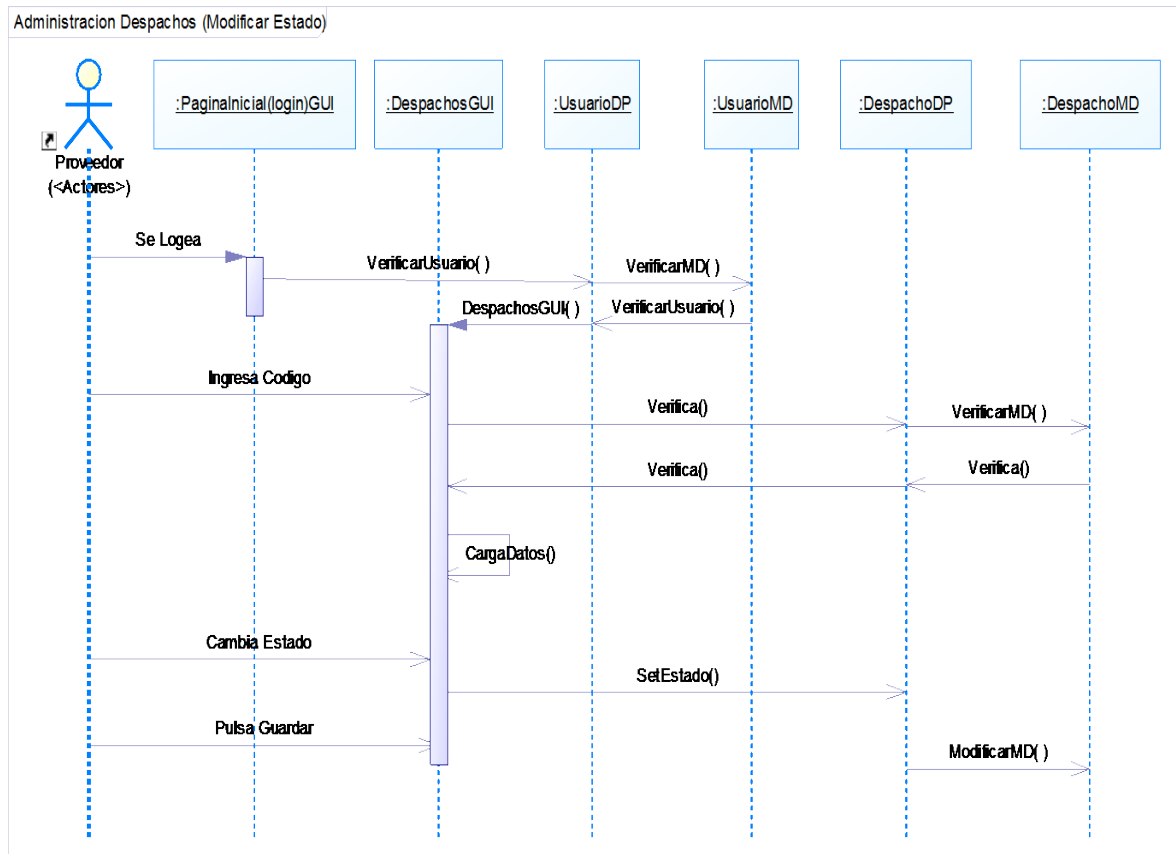


Ilustración 40: Modificar Estado Despacho. Autor: Alex Nieto.

5.2. Elaboración.

En este capítulo veremos la parte de elaboración del sistema con sus respectivos análisis y detalle.

5.2.1. Prototipo Ejecutable.

En esta sección se puede visualizar las diferentes interfaces para las tareas puntuales de cada trabajador que tiene un papel importante en todo el proceso de requisiciones y entrega.

5.2.1.1. Ventana Inicial (Página de Inicio).

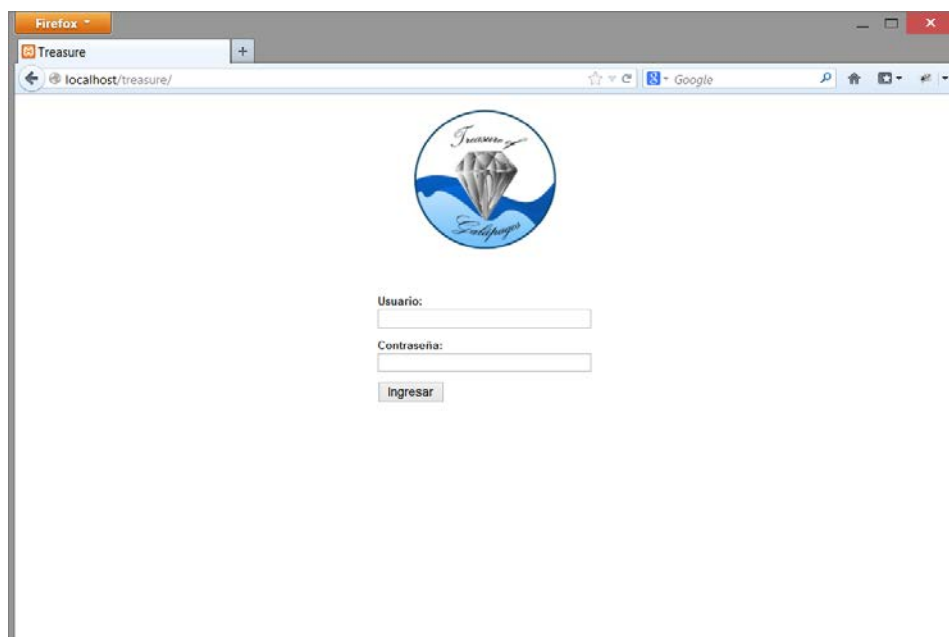


Ilustración 41: Ventana Inicial (Prototipo). Autor: Alex Nieto

En esta ventana se hace el ingreso al sistema (login) de cada usuario con su perfil y así direccionar a la página principal de cada uno con sus distintos privilegios de usuario e interfaces.

5.2.1.2. *Ventana Principal de Administrador (Página Principal Administrador).*



Ilustración 42: Ventana Principal de Administrador (Prototipo). Autor: Alex Nieto

Esta es la ventana principal (Página Principal Administrador) en la cual tenemos entre las siguientes opciones principales:

- Usuario:
 - Nuevo, Modificar.
- Departamentos:
 - Nuevo, Modificar.
- Productos:
 - Nuevo, Modificar.
- Pedidos:
 - Nuevo, Modificar, Aprobar.
- Despachos:
 - Nuevo, Modificar, Procesar.
- Bodega:
 - Manejo.

5.2.1.3. *Ventana Principal de Solicitante (Página Principal Solicitante).*

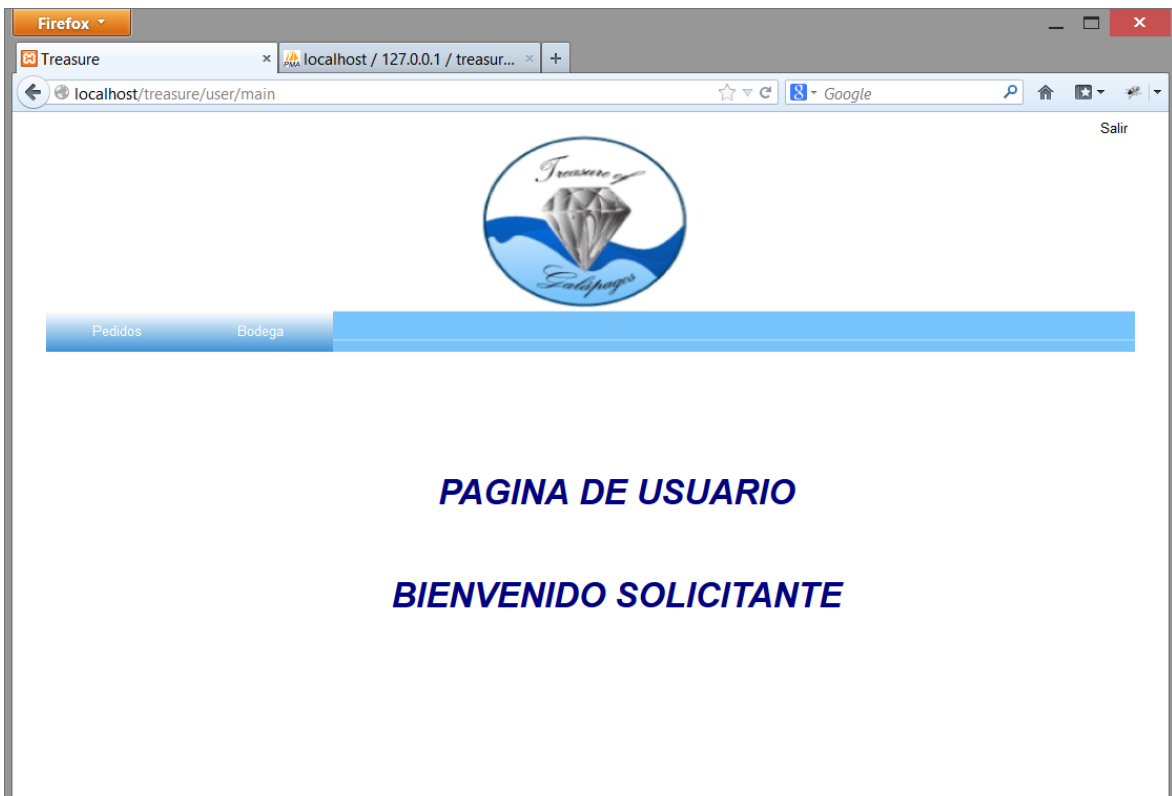


Ilustración 43: Ventana Principal Solicitante (Prototipo). Autor: Alex Nieto

Esta es la Ventana Principal de Solicitante (Página Principal Solicitante) en la cual se tienen las siguientes opciones principales:

- Pedidos:
 - Nuevo, Modificar.
- Bodega:
 - Manejo.

5.2.1.4. *Ventana Principal de Aprobador (Página Principal Aprobador).*



Ilustración 44: Ventana Principal de Aprobador (Prototipo). Autor: Alex Nieto

Esta es la Ventana Principal de Aprobador (Página Principal Aprobador) en la cual tenemos las siguientes opciones principales:

- Pedidos:
 - Aprobar.

5.2.1.5. *Ventana Principal de Controlador (Página Principal Controlador).*



Ilustración 45: Ventana Principal de Controlador (Prototipo). Autor: Alex Nieto

Esta es la Ventana Principal de Controlador (Página Principal Controlador) en la cual tenemos las siguientes opciones principales:

- Despachos:
 - Procesar.

5.2.1.6. *Ventana Principal de Proveedor (Página Principal Proveedor).*



Ilustración 46: Ventana Principal de Proveedor (Prototipo). Autor: Alex Nieto

Esta es la Ventana Principal de Proveedor (Página Principal Proveedor) en la cual tenemos las siguientes opciones principales:

- Despachos:
 - Nuevo, Modificar, Procesar.

5.2.1.7. *Ventana Administración Usuarios (Pagina Administración Usuarios).*

The screenshot shows a web browser window with the title 'Treasure Administrador' and the URL 'localhost/treasure/admin/user/create'. The page features a logo for 'Treasure of Galapagos' at the top center, which includes a diamond and the text 'Treasure of Galapagos'. Below the logo is a horizontal navigation menu with the following items: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', 'Bodega', and a highlighted 'Nuevo Usuario' button. The main content area is titled 'NUEVO USUARIO' and includes a note '(*) Campos Obligatorios'. The form contains the following fields and controls:

* Nombre:	<input type="text"/>	* Cargo:	<input type="text" value="-Seleccione-"/>
* Apellido:	<input type="text"/>		
* Usuario:	<input type="text"/>	* Estado:	<input type="text" value="-Seleccione-"/>
* Contraseña:	<input type="text"/>	* Confirmar Contraseña:	<input type="text"/>

At the bottom center of the form is a 'Guardar' button.

Ilustración 47: Ventana Administración Usuarios (Prototipo). Autor: Alex Nieto

Esta ventana permite el ingreso, modificación y cambio de estado (solo los usuarios que estén activos podrán ingresar al sistema) de usuarios que harán uso del sistema, este es el único método de agregación de usuarios para el uso del sistema por lo tanto el único en privilegios de creación de usuarios es el Administrador del Sistema.

5.2.1.8. *Ventana Administración de Departamentos (Pagina de Administración de Departamentos).*

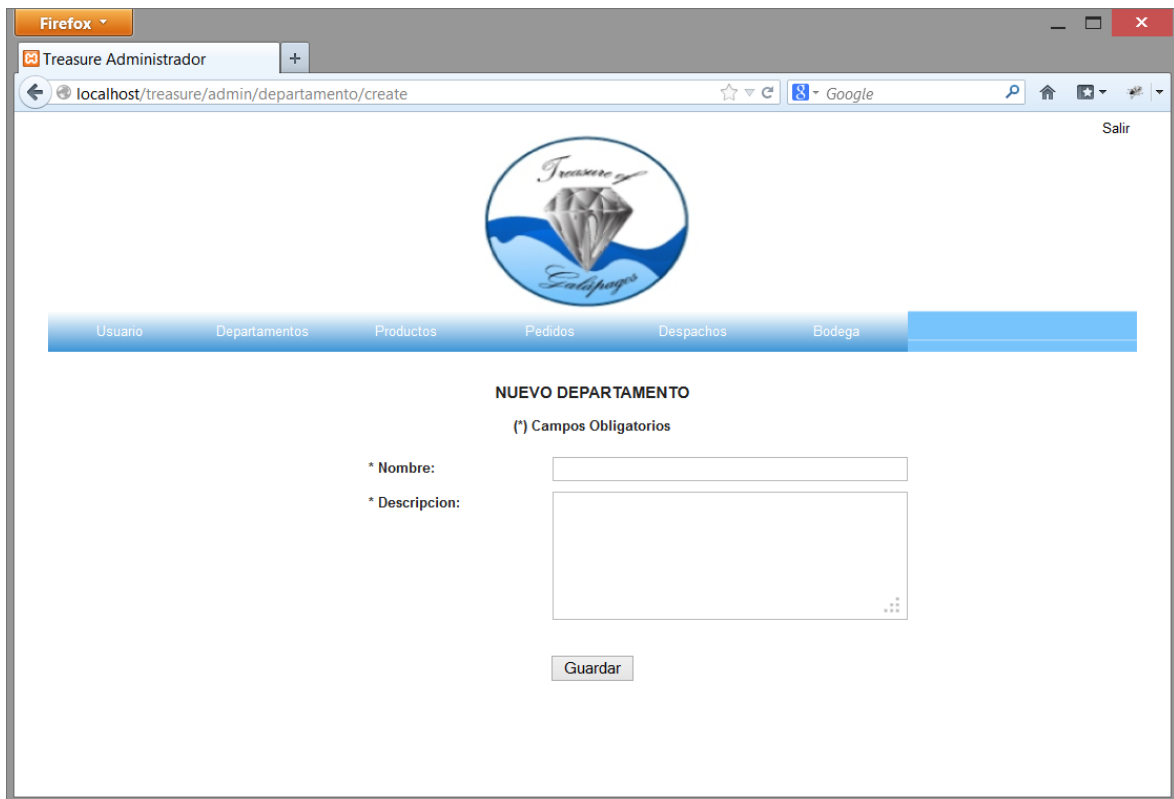


Ilustración 48: Ventana Administración de Departamentos (Prototipo). Autor: Alex Nieto

Esta ventana permite el ingreso y modificación de departamentos que pertenecen los solicitantes, este es el único método de agregación de departamentos para el uso del sistema por lo tanto el único en privilegios de creación de usuarios es el Administrador del sistema.

5.2.1.9. *Ventana Administración de Productos (Pagina de Administración de Productos).*

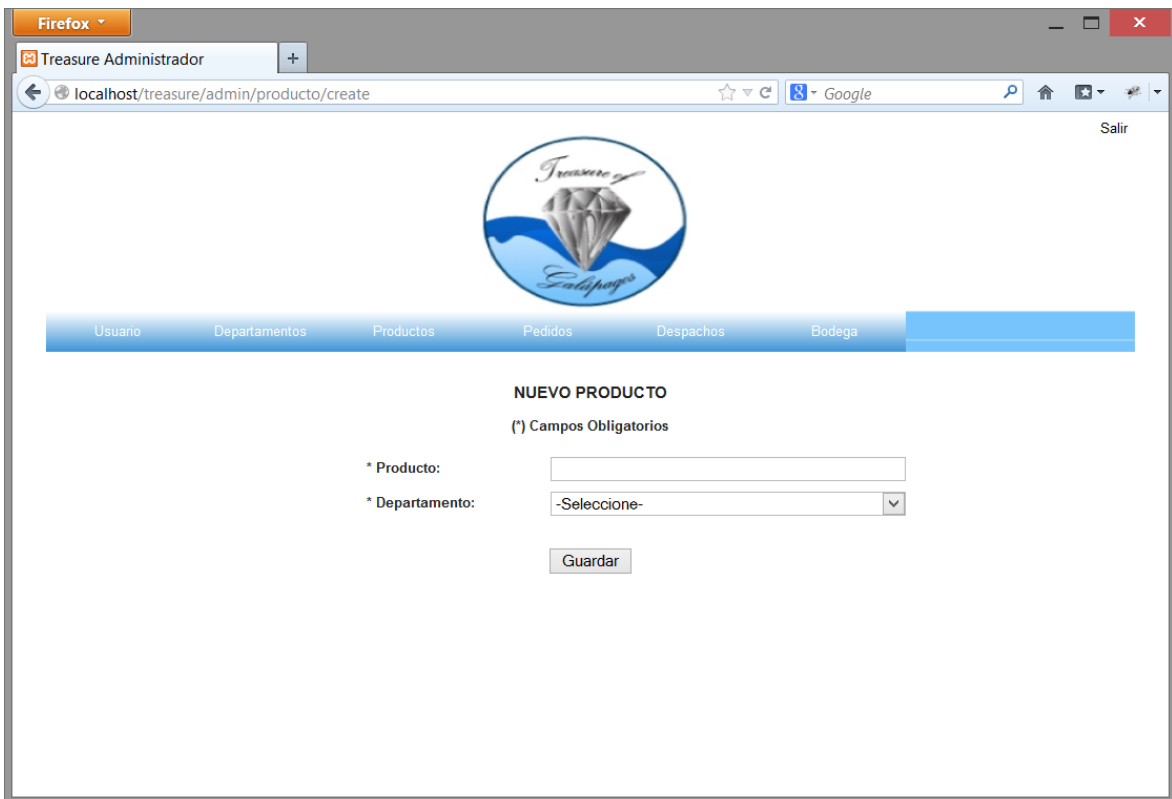


Ilustración 49: Ventana Administración de Productos (Prototipo). Autor: Alex Nieto

Esta ventana permite el ingreso y modificación de productos que pertenecen al proceso de pedidos por departamento, este es el único método de agregación de productos para el uso del sistema por lo tanto el único en privilegios de creación de usuarios es el Administrador del Sistema.

5.2.1.10. *Ventana Administración de Pedidos (Pagina de Administración de Pedidos).*

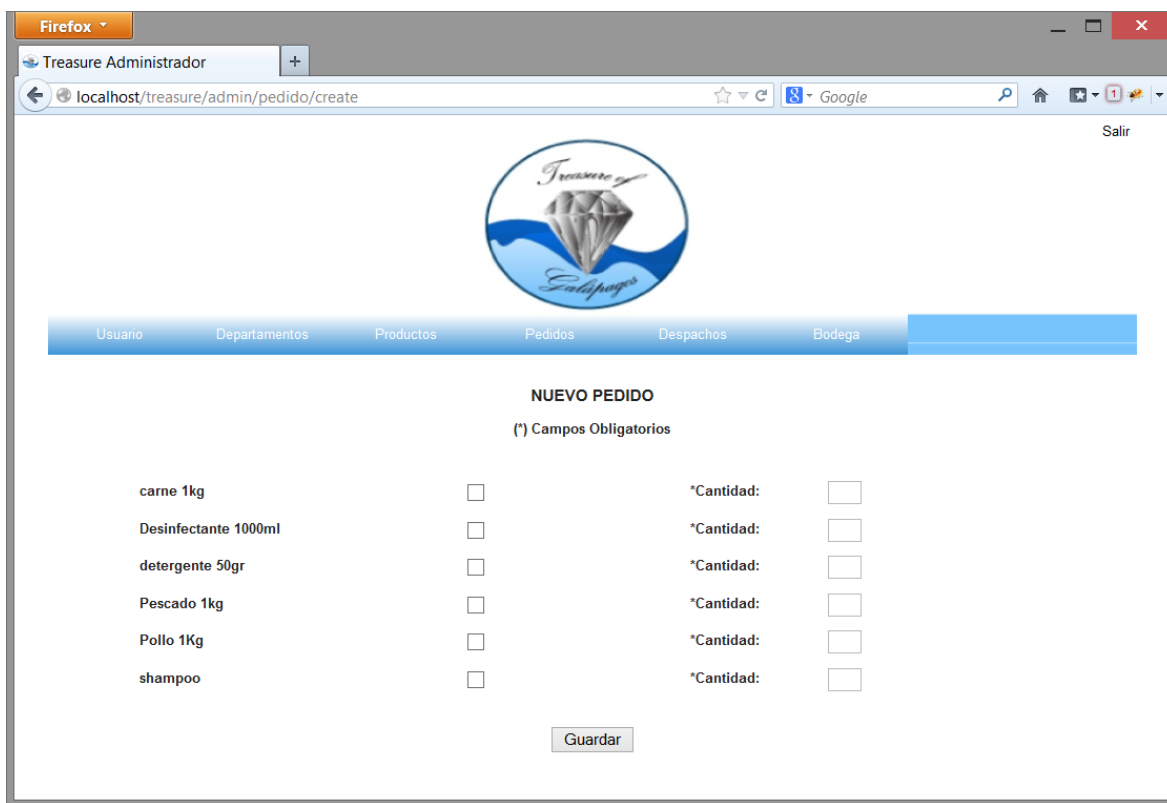


Ilustración 50: Ventana Administración de Pedidos (Prototipo). Autor: Alex Nieto

Esta ventana permite el ingreso y modificación de productos que se van a incluir en un nuevo pedido, los productos que se muestran depende del usuario que ingrese y que solo podrá observar los productos del departamento al que pertenece; los usuarios que puedes ingresar a esta ventana son:

- Administrador. (ve todos los productos sin importar departamento).
- Solicitante.
- Aprobador. (solo puede revisar no modificar los productos del pedido).

5.2.1.11. *Ventana Administración de Estado Pedidos (Pagina de Administración Estado de Pedidos).*

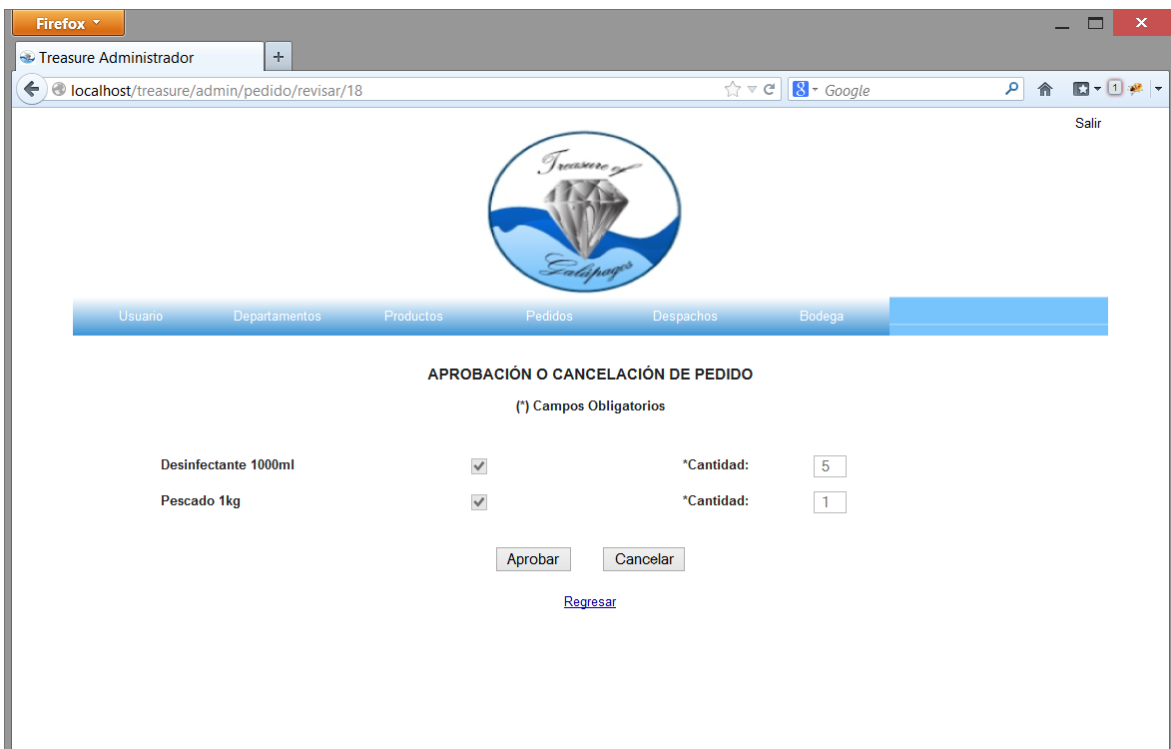


Ilustración 51: Ventana Administración de Estado Pedidos (Prototipo). Autor: Alex Nieto

Esta ventana permite la modificación de estado de pedidos que se incluirán en la lista para despachos, los productos que se muestran depende de los productos que el solicitante que creó el pedido; los usuarios que pueden ingresar a esta ventana son:

- Administrador.
- Aprobador.

5.2.1.12. *Ventana Administración de Despachos (Pagina de Administración de Despachos).*

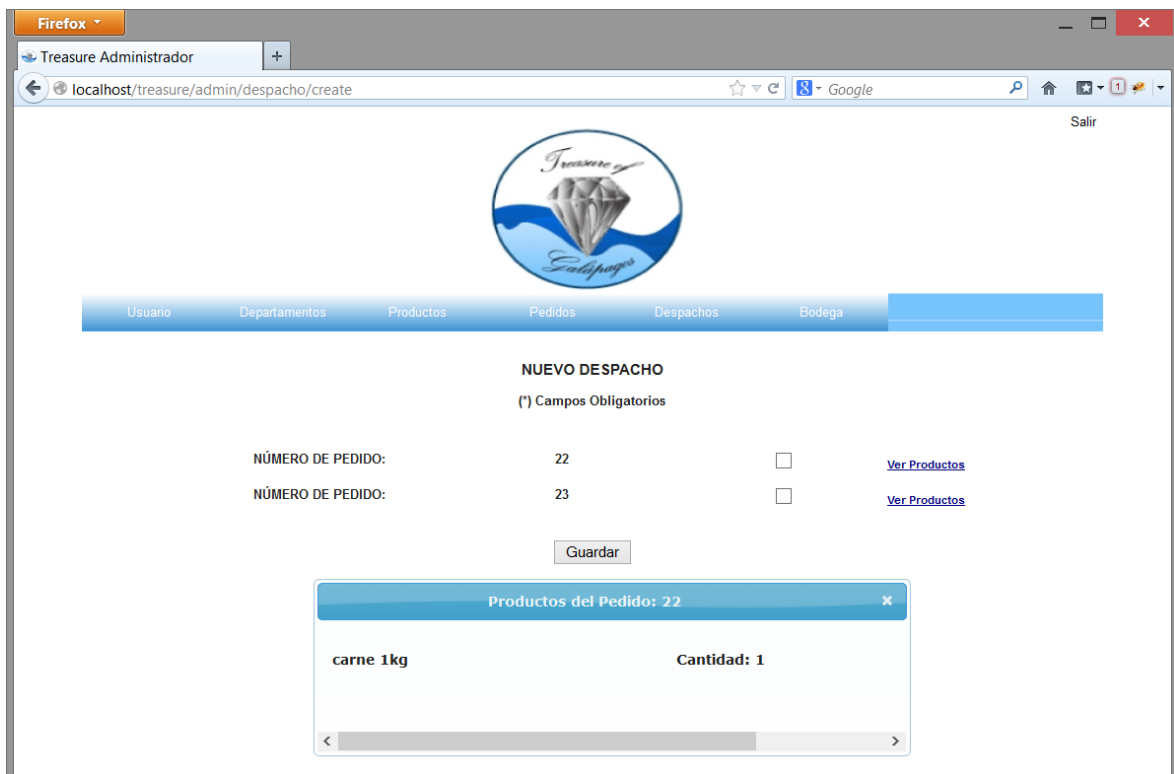


Ilustración 52: Ventana Administración de Despachos (Prototipo). Autor: Alex Nieto

Esta ventana permite el ingreso y modificación de despachos, los pedidos que se muestran depende de los pedidos que estén en aprobados; los usuarios que puedes ingresar a esta ventana son:

- Administrador.
- Proveedor.

5.2.1.13. *Ventana Administración de Estado de Despachos (Pagina de Administración de Estado de Despachos).*

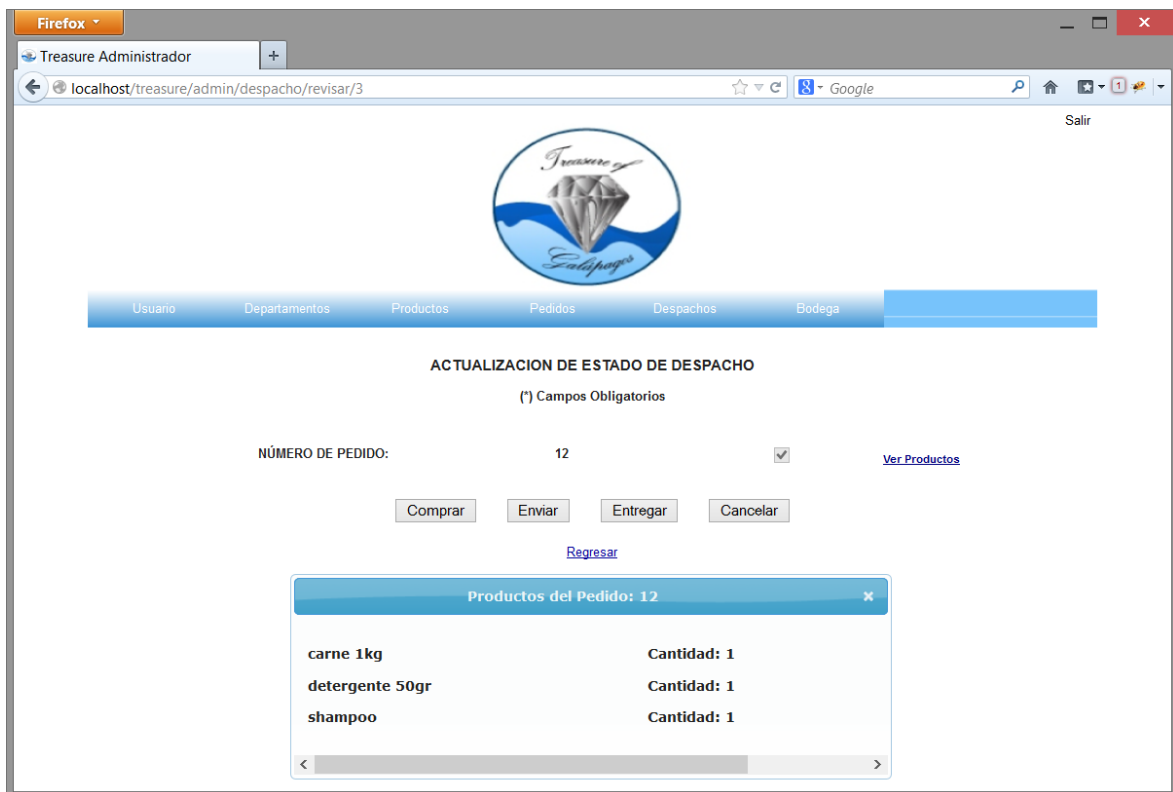


Ilustración 53: Ventana Administración de Estado de Despachos (Prototipo). Autor: Alex Nieto

Esta ventana permite la modificación de estado de despachos para saber en qué etapa se encuentra el pedido, los productos que se muestran depende de los productos que el solicitante que creo el pedido; los usuarios que pueden ingresar a esta ventana son:

- Administrador. (Maneja todos los estados)
- Proveedor. (Maneja todos los estados a excepción del entregado)
- Controlador. (Maneja únicamente el estado de entregado)

5.2.1.14. *Ventana Administración de Bodega (Pagina de Administración de Bodega).*

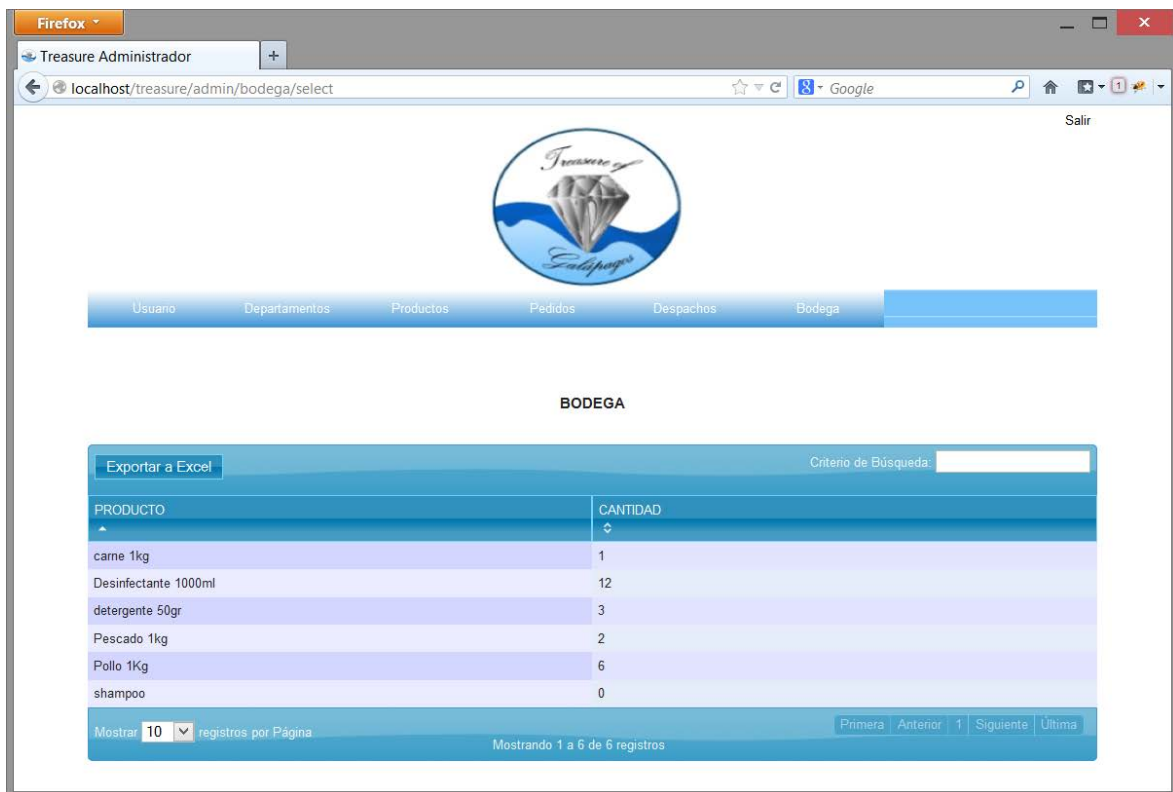


Ilustración 54: Ventana Administración de Bodega (Prototipo). Autor: Alex Nieto

Esta ventana permite la modificación de la cantidad de productos que se encuentran en stock (bodega); los usuarios que pueden ingresar a esta ventana son:

- Administrador. (Puede aumentar o disminuir productos)
- Solicitante. (Solo puede generar egresos de los productos según el departamento al que este asignado).

5.2.2. Lista De Riesgos.

5.2.2.1. Descripción.

Propósito:

El propósito de generar una lista de riesgos es el de realizar un análisis durante la fase media del proyecto. Para cada riesgo observado se detallarán sus efectos y las acciones tomadas para reducir la probabilidad de que reaparezcan.

Perspectiva general:

Se listarán todos los riesgos identificados y analizados cada uno con la siguiente estructura descriptiva:

- Magnitud: La escala utilizada para estimar el grado de daño que genera el riesgo al momento de convertirse en realidad será:
 - Despreciable. (Grado 1).
 - Marginal. (Grado 2).
 - Crítico. (Grado 3).
 - Catastrófico. (Grado 4).
- Descripción: Breve explicación del riesgo.
- Impacto: Descripción de cómo afecta al proyecto directamente.
- Indicadores: Manera de medir el riesgo generado.
- Plan de acción: Pasos a realizarse en el caso de que el riesgo se convierta en una realidad y como poder hacer que su impacto mantenga niveles no críticos o nulos; esto se realiza antes de que el riesgo se vuelva una realidad
- Plan de contingencia: Medidas a tomar en el proyecto una vez que el riesgo se haya transformado en una realidad.

5.2.2.2. Riesgos.

Cambio de Requisitos.

- Magnitud:
 - Etapa inicial: Despreciable.
 - Etapa de Elaboración: Marginal.
 - Etapa de Construcción: Crítico
 - Etapa de Implementación: Critico.
- Descripción:

El cliente puede generar nuevos requisitos o modificar requisitos que ya fueron dados a conocer en cualquier etapa del desarrollo del proyecto. Generalmente este riesgo aparece en las primeras fases; debido a que durante el proceso el cliente aún está descubriendo más necesidades respecto al producto deseado.
- Impacto:

La implementación sobre el desarrollo del proyecto o modificación de partes ya generadas requerirá cambios en la parte de documentación y producto final. Modificaciones o implementaciones resultan más factibles realizarlas durante las primeras fases del proyecto, pero pueden suponer perturbaciones en las etapas finales.
- Indicadores:

Cantidad de cambios en los requisitos.
- Plan de acción:

Realizar un buen análisis de requerimientos en las primeras etapas de desarrollo para poder generar un producto de calidad.
- Plan de Contingencia:

Durante las primeras etapas de desarrollo se incorporaran o se realizaran los cambios necesarios a requerimientos antes establecidos.

Durante las etapas finales se valorará la importancia de los cambios o nuevas requisiciones frente al tiempo que requiere su desarrollo, a la vez analizar el impacto en otros requisitos y la documentación que se realice.

Falta de experiencia en tareas de planificación.

- Magnitud:
 - Marginal.
- Descripción:

El desarrollador se enfrenta por primera vez a una planificación de un proyecto de alto grado.
- Impacto:

La planificación guía el proyecto. Un error puede influir de manera directa en los resultados esperados. Sin embargo, la planeación modular o por etapas reduce el impacto de los errores, permitiendo que puedan ser corregidos o absorbidos en próximas etapas de desarrollo
- Indicadores:

Las diferencias directas entre el documento de planeación frente al desarrollo real.
- Plan de acción:

Realizar autoevaluaciones de continuidad por parte del desarrollador.
- Plan de Contingencia:

Realizar una retroalimentación a comparar el tiempo planificado con el real y generar una replanificación de las fases siguientes.

Falta de experiencia con las herramientas utilizadas.

- Magnitud:
 - Etapa inicial: Despreciable.
 - Etapa de Elaboración: Marginal.
 - Etapa de Construcción: Crítico
 - Etapa de Implementación: Critico.
- Descripción:

El desarrollador presenta inconvenientes al momento de elaborar el proyecto debido a la inexperiencia de las herramientas en general (recursos software).
- Impacto:

Generación de retrasos frente a la planificación original.
- Indicadores:

Cantidad de avances lentos o nulos.
- Plan de acción:

Una parte del tiempo de desarrollo del proyecto se destinará al aprendizaje de las herramientas.
- Plan de Contingencia:

De generarse un retraso de alto impacto se deberá acudir a asistencia externa y/o consultoría por medio de Internet mediante foros, manuales en línea o libros. En último recurso, el menos probable, se planteará un cambio de herramienta.

Diseño erróneo.

- Magnitud:
 - Etapa de Elaboración: Despreciable.
 - Etapa de Construcción: Crítico.
- Descripción:

La elaboración del sistema no es el adecuado. Durante la etapa de construcción puede darse el caso que el diseño no presenta el nivel de detalle o este mal enfocado, bien por naturaleza del problema o por la limitante en las herramientas de desarrollo escogidas.
- Impacto:

Generación de retrasos frente a la planificación original.
Requiere la actualización o modificación de los artefactos de diseño.
- Indicadores:

La arquitectura no cumple las expectativas de diseño.
Se complica la implementación del proyecto.
- Plan de acción:

Generar un prototipo ejecutable para analizar el diseño elegido y poder comprobar la arquitectura de la misma. En caso de encontrarse errores puede reestructurarse la parte de la arquitectura.
- Plan de Contingencia:

Si se convierte en una realidad en la fase de elaboración, se revisara y modificara la documentación de diseño afectada.
Si se presenta durante la fase de construcción, se hará un estudio para la solución dentro del tiempo plazo que se dispone.
Se reajustara la planificación si es necesario.

Falta de un experto.

- Magnitud:
 - Marginal.
- Descripción:

No existe un experto del dominio en el equipo de desarrollo a quien consultar.
- Impacto:

Generación de retrasos frente a la planificación original.
- Indicadores:

Cantidad de avances lentos o nulos.
- Plan de acción:

Autoaprendizaje durante la ejecución del proyecto.
- Plan de Contingencia:

De generarse un retraso de alto impacto se deberá acudir a asistencia externa y/o consultoría por medio de internet mediante foros, manuales en línea o libros.

Pérdida de datos.

- Magnitud:
 - Crítico.
- Descripción:

Los datos requeridos no aparecen en los lugares que deberían aparecer.
- Impacto:

Es variable puede generar desde un retraso hasta una catástrofe.
- Indicadores:

Porcentaje de la información total perdida.
- Plan de acción:

Se utilizarán varios repositorios en nubes de información.
Se realizarán copias de seguridad de los equipos en los que se trabajen.
- Plan de Contingencia:

Recuperar versiones anteriores que permitan el seguimiento o reconstrucción de la información perdida.

5.3. Construcción.

En este capítulo se describirá la implementación para cada perfil de usuario existente y pruebas del sistema en detalle.

5.3.1. Implementación y Pruebas.

A continuación, se detallara por cada perfil sus funciones y manejo del proceso central con las respectivas pruebas de ingreso, validaciones de usuario y de funcionamiento.

La primera ventana Login es general para todos los usuarios.

5.3.1.1. Ventana Inicial Login:

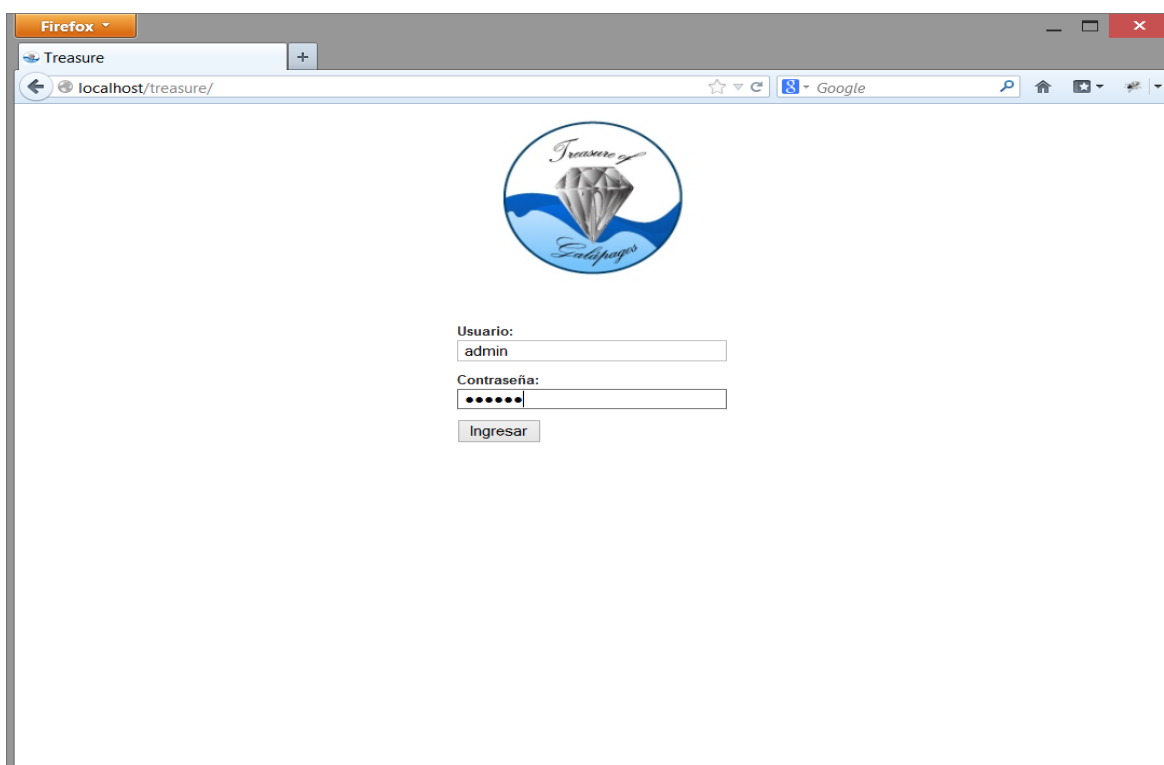


Ilustración 55: Ventana Inicial Login. Autor: Alex Nieto.

En esta ventana principal el usuario escribe sus credenciales, el sistema automáticamente le direccionara a la página principal dependiendo del perfil que tenga.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Usuario.	Usuario Valido.	Usuario no valido.	Ventana Principal (según el usuario).
Contraseña.	Contraseña correspondiente al usuario.	Contraseña no correspondiente al usuario.	Ventana Principal (según el usuario).

Tabla 8: Ingreso Login. Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

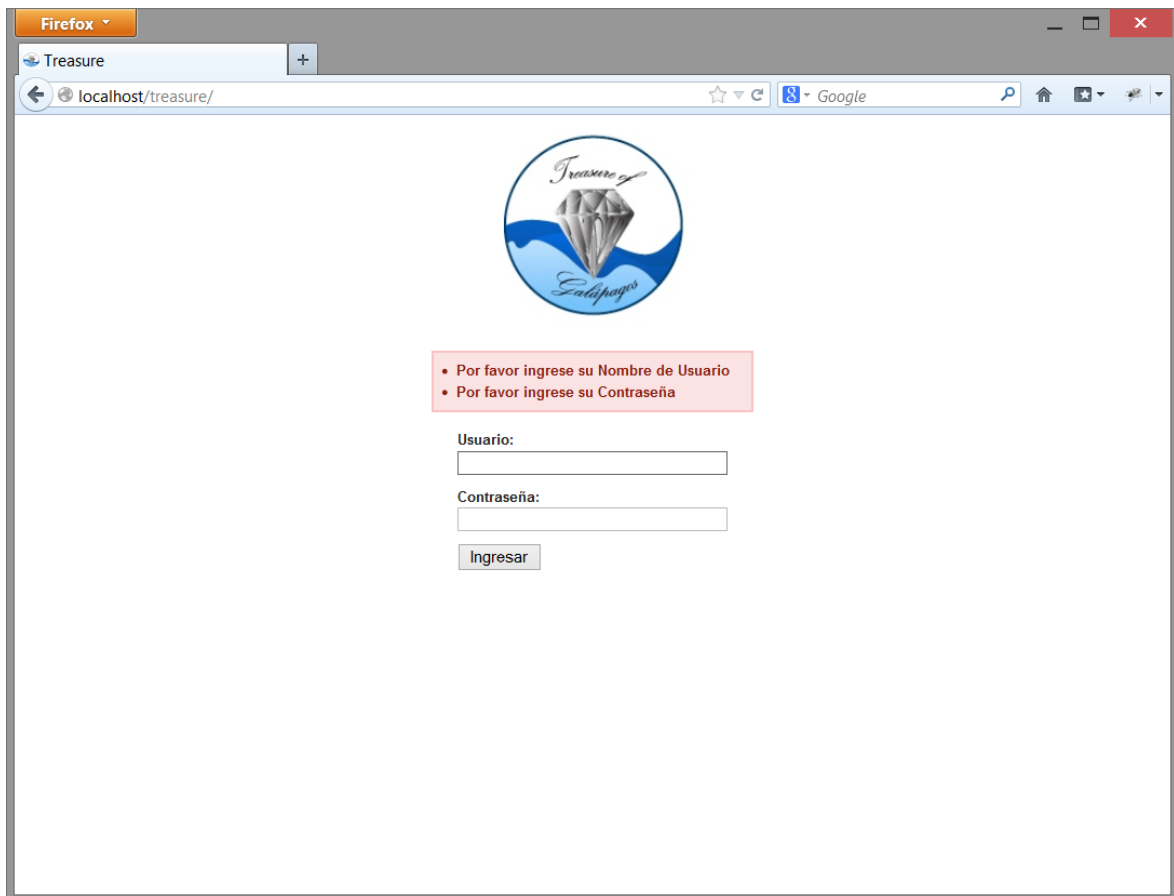


Ilustración 56: Ventana Inicial Login (Validaciones Usuario). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Usuario	Identificador del usuario: 20 Caracteres	Identificador del usuario: + 1 Caracteres	Mensaje con lo que falta de llenar
Contraseña	Clave de Ingreso: 32 Caracteres	Clave Ingreso: + 1 Caracteres	Mensaje con lo que falta de llenar

Tabla 9: Ingresos Inicial Login (Validaciones Usuario). Autor: Alex Nieto.

5.3.1.2. *Administrador:*

Ventana Principal:

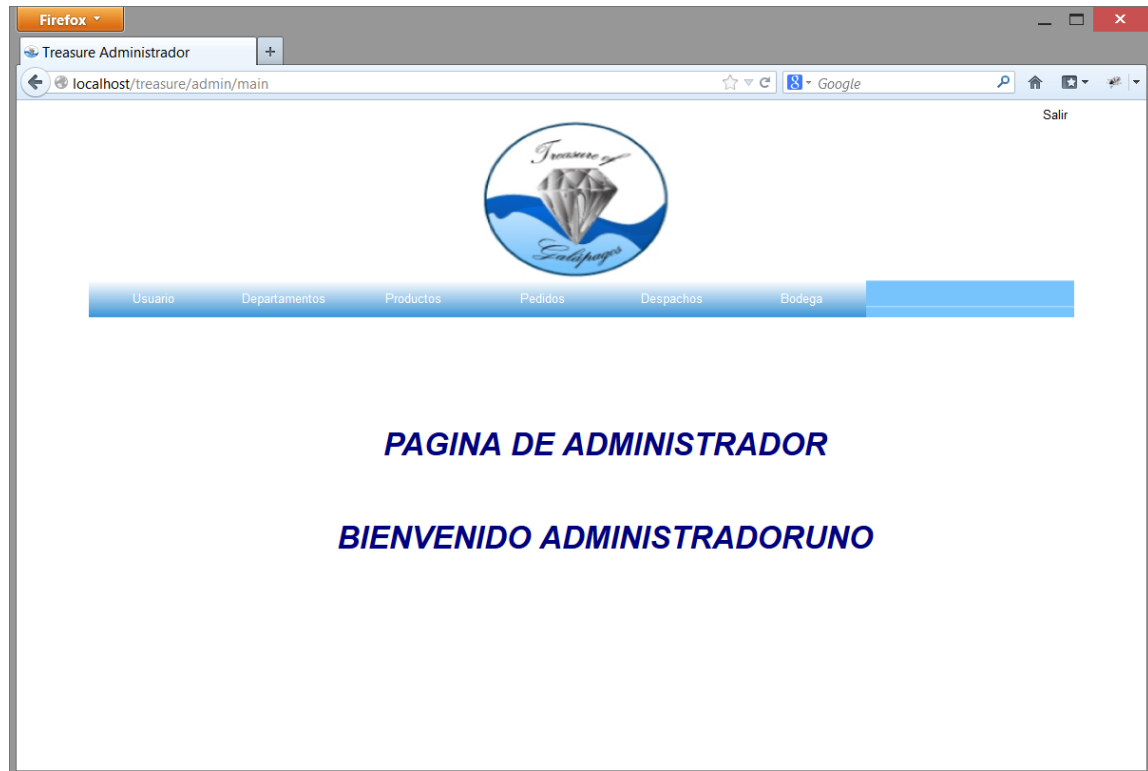


Ilustración 57: Ventana Principal (Administrador). Autor: Alex Nieto.

EL usuario que posee un perfil de administrador se dirige a esta página en donde presenta un menú con todos los privilegios para control del sistema.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de Nuevo Usuario:

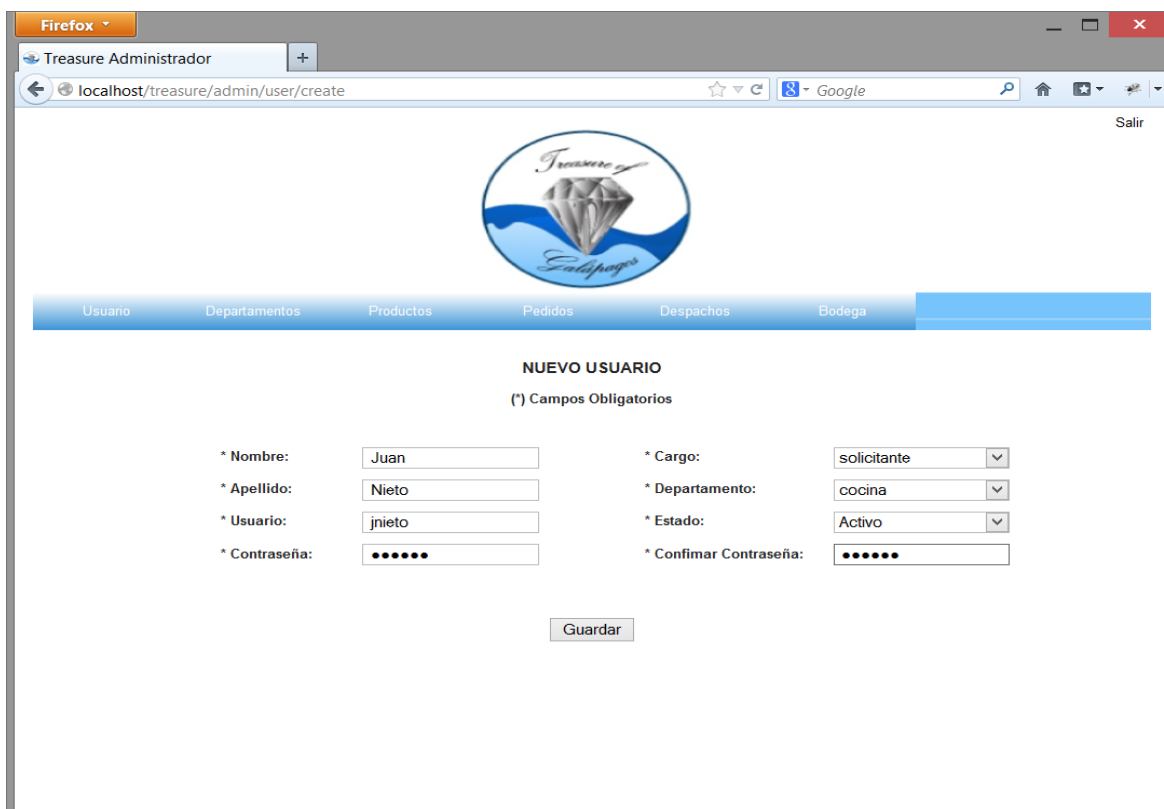


Ilustración 58: Ventana Nuevo Usuario (Administrador). Autor: Alex Nieto.

En esta ventana el administrador puede ingresar un nuevo usuario para que haga uso del sistema una vez lleno los campos requeridos se procede a guardar.

Ingreso:

	ID_USUARIO	ID_PERFIL	ID_DEPARTAMENTO	NOMBRE	APELLIDO	USUARIO	CONTRASENIA	ESTADO
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	1	NULL	administradoruno	administradoruno	admin	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	11	3	NULL	aprobador	aprobador	aprobador	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	12	4	NULL	controlador	controlador	controlador	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	14	5	NULL	proveedor	proveedor	proveedor	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	26	2	1	solicitante	solicitante	solicitante	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	31	2	3	solicitantedos	solicitantedos	solicitantedos	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	32	2	2	solicitantetres	solicitantetres	solicitantetres	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	33	2	2	solicitante	cuatro	solicitantecuatro	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	34	2	1	Juan	Nieto	jnieto	e10adc3949ba59abbe56e057f20f883e	Activo

Ilustración 59: Ingreso Nuevo Usuario (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de usuario: máximo de 30 caracteres.	Nombre de usuario: + 30 caracteres.	Registro almacenado en base de datos Campo VARCHAR(30)

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Apellido	Apellido de usuario: máximo de 30 caracteres.	Apellido de usuario: + 30 caracteres.	Registro almacenado en base de datos Campo VARCHAR(30)
Usuario	Usuario de Ingreso: 20 Letras	Usuario de Ingreso: + 20 caracteres.	Registro almacenado en base de datos Campo VARCHAR (11).
Contraseña	Contraseña de Ingreso: 11 Letras	Usuario de Ingreso: + 20 caracteres. MD5 encriptado.	Registro almacenado en base de datos Campo VARCHAR (32).
Cargo	Cargo: 1 id	Cargo: Sin errores truncado con Combo Box	Registro almacenado en base de datos Campo INT(11)
Departamento	Departamento: 1 id	Departamento: Sin errores truncado con Combo Box	Registro almacenado en base de datos Campo INT(11)
Estado	Estado: ACTIVO, INACTIVO	Estado: Sin errores truncado con Combo Box	Registro almacenado en base de datos enum('Activo', 'Inactivo')

Tabla 10: Ingreso Nuevo Usuario (Administrador). Autor: Alex Nieto.

Pruebas validaciones de usuario.

Firefox

Treasure Administrador

localhost/treasure/admin/user/create

Salir

Usuario Departamentos Productos Pedidos Despachos Bodega

NUEVO USUARIO
(*) Campos Obligatorios

- El campo "Nombre" es obligatorio.
- El campo "Cargo" es obligatorio.
- El campo "Apellido" es obligatorio.
- El campo "Usuario" es obligatorio.
- El campo "Estado" es obligatorio.
- El campo "Contraseña" es obligatorio.
- El campo "Confirmar Contraseña" es obligatorio.

* Nombre:

* Apellido:

* Usuario:

* Contraseña:

* Cargo:

* Estado:

* Confirmar Contraseña:

Guardar

Ilustración 60: Ventana Nuevo Usuario Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de usuario: máximo de 30 caracteres.	Nombre de usuario: + 30 caracteres.	Mensaje con lo que falta de llenar
Apellido	Apellido de usuario: máximo de 30 caracteres.	Apellido de usuario: + 30 caracteres.	Mensaje con lo que falta de llenar
Usuario	Usuario de Ingreso: 20 Letras	Usuario de Ingreso: + 20 caracteres.	Mensaje con lo que falta de llenar
Contraseña	Contraseña de Ingreso: 11 Letras	Usuario de Ingreso: + 20 caracteres. MD5 encriptado.	Mensaje con lo que falta de llenar
Cargo	Cargo: 1 id	Cargo: Sin errores truncado con Combo Box	Mensaje con lo que falta de llenar
Departamento	Departamento: 1 id	Departamento: Sin errores truncado con Combo Box	Mensaje con lo que falta de llenar
Estado	Estado: ACTIVO, INACTIVO	Estado: Sin errores truncado con Combo Box	Mensaje con lo que falta de llenar

Tabla 11: Ingreso Nuevo Usuario Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Usuario:

Exportar a Excel

Criterio de Búsqueda

NOMBRE	APELLIDO	CARGO	DEPARTAMENTO	USUARIO	ESTADO	EDITAR
administradoruno	administradoruno	administrador	NINGUNO	admin	Activo	
aprobador	aprobador	aprobador	NINGUNO	aprobador	Activo	
controlador	controlador	controlador	NINGUNO	controlador	Activo	
proveedor	proveedor	proveedor	NINGUNO	proveedor	Activo	
solicitante	solicitante	solicitante	cocina	solicitante	Activo	
solicitante	cuatro	solicitante	limpieza	solicitantecuatro	Activo	
solicitantedos	solicitantedos	solicitante	habitaciones	solicitantedos	Activo	
solicitantetres	solicitantetres	solicitante	limpieza	solicitantetres	Activo	

Mostrar 10 registros por Página

Mostrando 1 a 8 de 8 registros

Primera Anterior 1 Siguiente Ultima

Ilustración 61: Ventana Selección Usuario (Administrador). Autor: Alex Nieto.

Esta ventana cumple dos funciones mantener un reporte de los usuarios registrados y a su vez funciona para poder hacer selección de un usuario para modificar los atributos en el caso de que estén incorrectos.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de edición de Usuario:

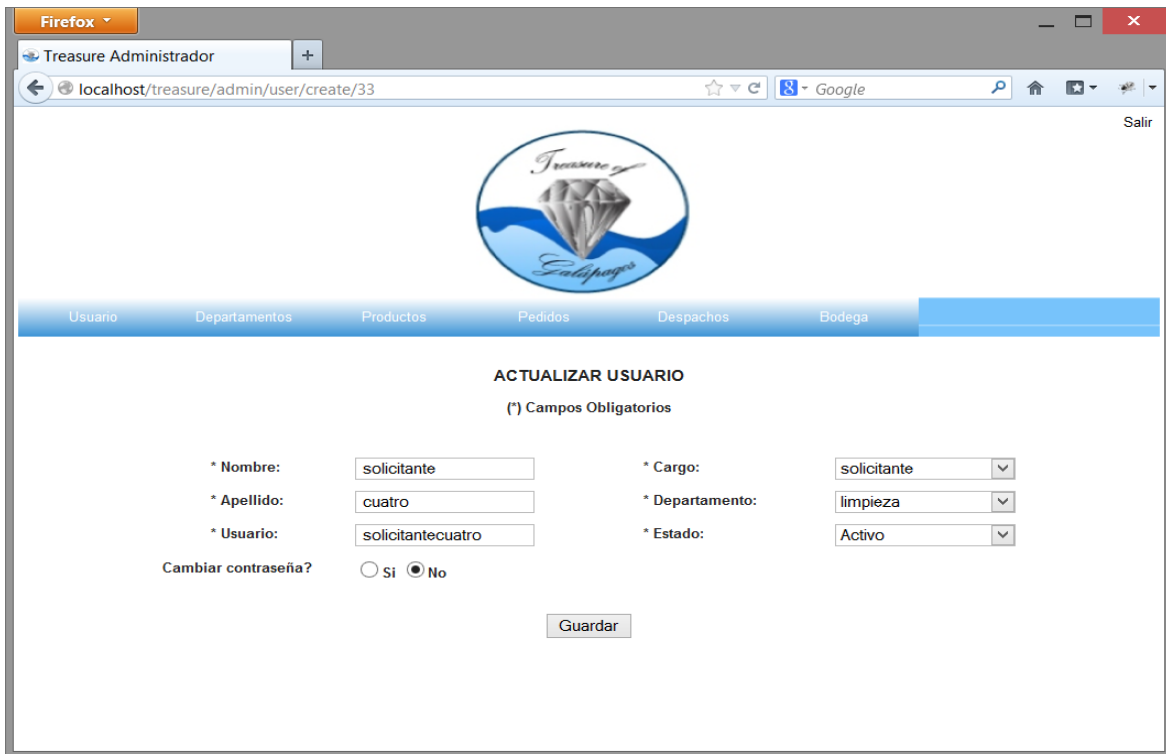


Ilustración 62: Ventana Edición Usuario (Administrador). Autor: Alex Nieto.

Esta ventana se puede editar la información de cada usuario dependiendo de la selección realizada en la anterior ventana, siempre y cuando los datos estén completos se puede guardar.

Ingreso:

	ID_USUARIO	ID_PERFIL	ID_DEPARTAMENTO	NOMBRE	APELLIDO	USUARIO	CONTRASENIA	ESTADO
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	1		administradoruno	administradoruno	admin	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	11	3		aprobador	aprobador	aprobador	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	12	4		controlador	controlador	controlador	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	14	5		proveedor	proveedor	proveedor	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	26	2	1	solicitante	solicitante	solicitante	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	31	2	3	solicitantedos	solicitantedos	solicitantedos	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	32	2	2	solicitantetres	solicitantetres	solicitantetres	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	33	2	2	solicitante	cuatro	solicitantecuatro	e10adc3949ba59abbe56e057f20f883e	Activo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	34	2	1	Juan	Nieto	jnieto	e10adc3949ba59abbe56e057f20f883e	Activo

Ilustración 63: Ingreso Edición Usuario (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de usuario: máximo de 30 caracteres.	Nombre de usuario: + 30 caracteres.	Registro almacenado en base de datos Campo VARCHAR(30)

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Apellido	Apellido de usuario: máximo de 30 caracteres.	Apellido de usuario: + 30 caracteres.	Registro almacenado en base de datos Campo VARCHAR(30)
Usuario	Usuario de Ingreso: 20 Letras	Usuario de Ingreso: + 20 caracteres.	Registro almacenado en base de datos Campo VARCHAR (11).
Contraseña	Contraseña de Ingreso: 11 Letras	Usuario de Ingreso: + 20 caracteres. MD5 encriptado.	Registro almacenado en base de datos Campo VARCHAR (32).
Cargo	Cargo: 1 id	Cargo: Sin errores truncado con Combo Box	Registro almacenado en base de datos Campo INT(11)
Departamento	Departamento: 1 id	Departamento: Sin errores truncado con Combo Box	Registro almacenado en base de datos Campo INT(11)
Estado	Estado: ACTIVO, INACTIVO	Estado: Sin errores truncado con Combo Box	Registro almacenado en base de datos enum('Activo', 'Inactivo')

Tabla 12: Ingreso Edición Usuario (Administrador). Autor: Alex Nieto.

Pruebas validaciones de usuario.

Firefox

Treasure Administrador

localhost/treasure/admin/user/create/34

Salir

Usuario Departamentos Productos Pedidos Despachos Bodega

ACTUALIZAR USUARIO

(*) Campos Obligatorios

- El campo "Nombre" es obligatorio.
- El campo "Apellido" es obligatorio.
- El campo "Usuario" es obligatorio.
- El campo "Estado" es obligatorio.
- El campo "Contraseña" es obligatorio.
- El campo "Confirmar Contraseña" es obligatorio.

* Nombre:

* Apellido:

* Usuario:

Cambiar contraseña? Si No

* Contraseña:

* Cargo: solicitante

* Departamento: -Seleccione-

* Estado: -Seleccione-

* Confirmar Contraseña:

Guardar

Ilustración 64: Ventana Edición Usuario Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de usuario: máximo de 30 caracteres.	Nombre de usuario: + 30 caracteres.	Mensaje con lo que falta de llenar
Apellido	Apellido de usuario: máximo de 30 caracteres.	Apellido de usuario: + 30 caracteres.	Mensaje con lo que falta de llenar
Usuario	Usuario de Ingreso: 20 Letras	Usuario de Ingreso: + 20 caracteres.	Mensaje con lo que falta de llenar
Contraseña	Contraseña de Ingreso: 11 Letras	Usuario de Ingreso: + 20 caracteres. MD5 encriptado.	Mensaje con lo que falta de llenar
Cargo	Cargo: 1 id	Cargo: Sin errores truncado con Combo Box	Mensaje con lo que falta de llenar
Departamento	Departamento: 1 id	Departamento: Sin errores truncado con Combo Box	Mensaje con lo que falta de llenar
Estado	Estado: ACTIVO, INACTIVO	Estado: Sin errores truncado con Combo Box	Mensaje con lo que falta de llenar

Tabla 13: Ingreso Edición Usuario Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de nuevo Departamento:

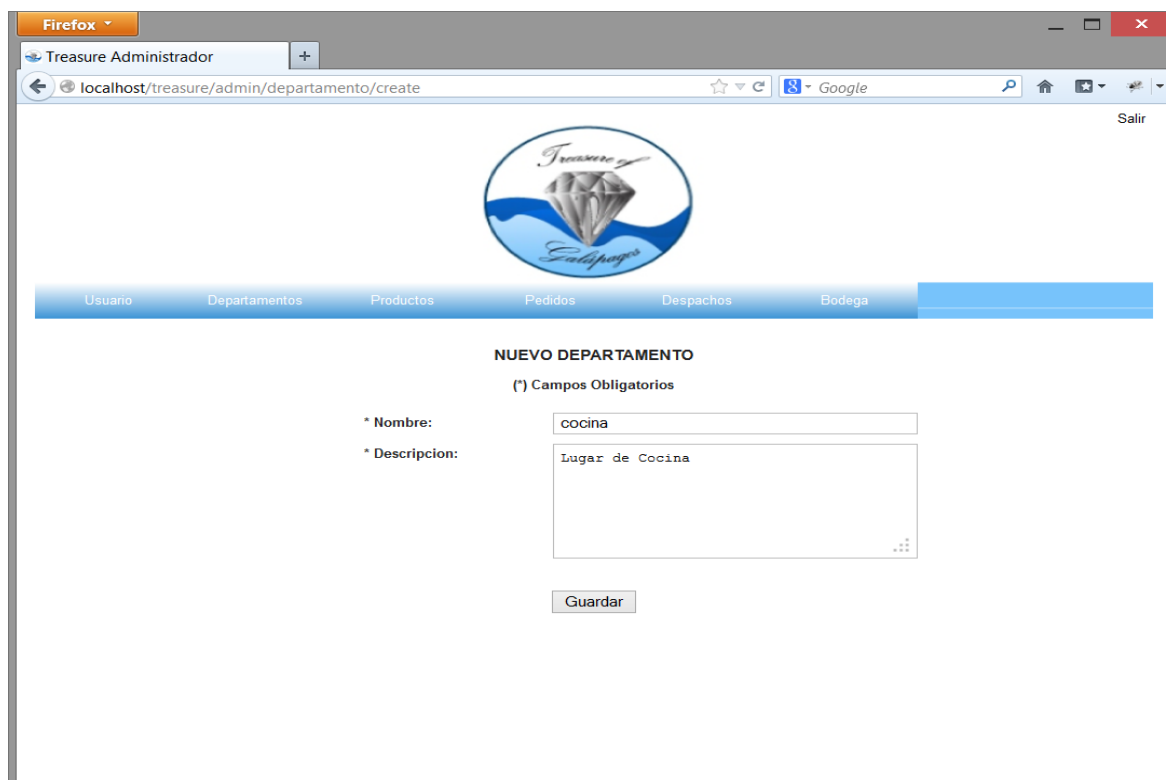


Ilustración 65: Ventana Nuevo Departamento (Administrador). Autor: Alex Nieto.

En esta ventana el administrador puede ingresar un nuevo departamento una vez lleno los campos requeridos se procede a guardar.

Ingreso:

	ID_DEPARTAMENTO	NOMBRE	DESCRIPCION
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	cocina	Lugar de Cocina
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	limpieza	Lugar de limpieza
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	habitaciones	Lugar de Limpieza
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	7	atencion al cliente	atencion al cliente

Ilustración 66: Ingreso Nuevo Departamento (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de departamento: Máximo de 50 caracteres.	Nombre de departamento: + 50 caracteres.	Registro almacenado en base de datos Campo VARCHAR(50)
Descripción	Descripción de departamento: Máximo de n caracteres.	Descripción de departamento: + n caracteres.	Registro almacenado en base de datos Campo TEXT

Tabla 14: Ingreso Nuevo Departamento (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

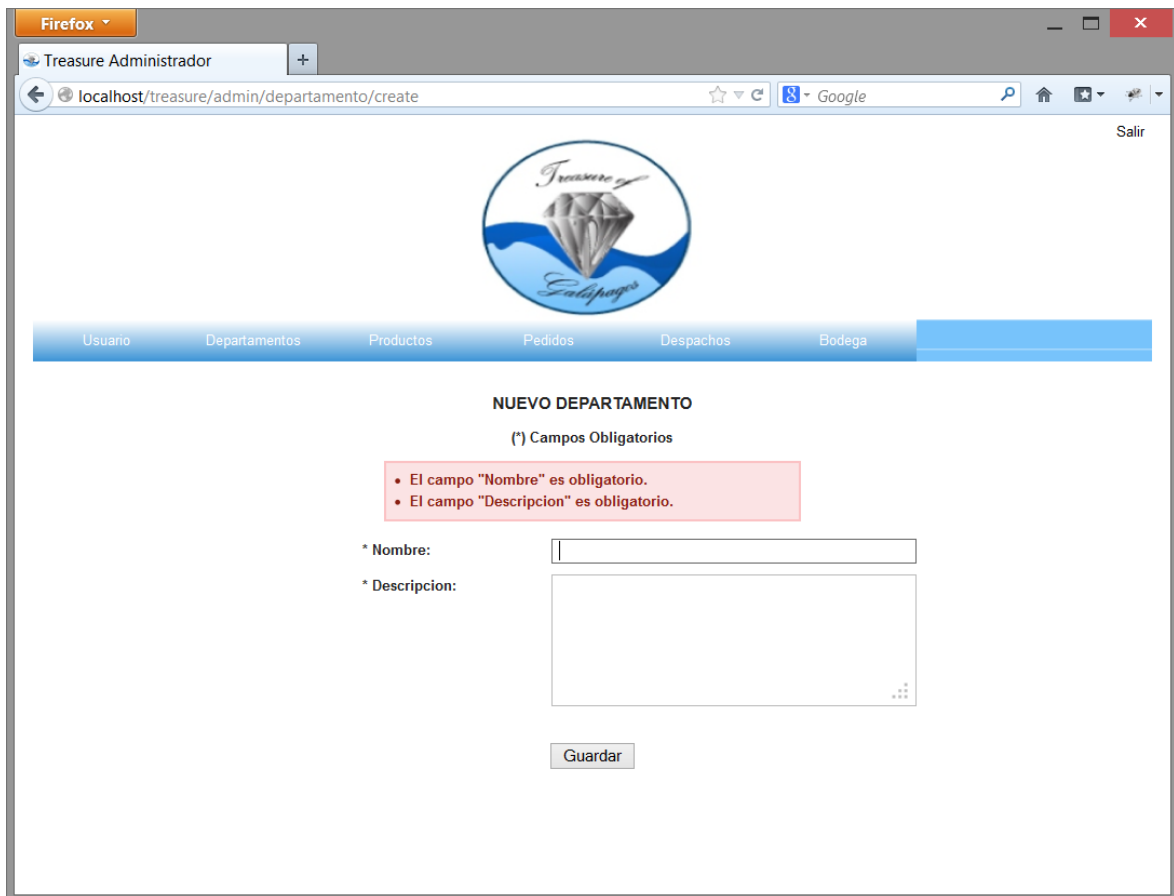


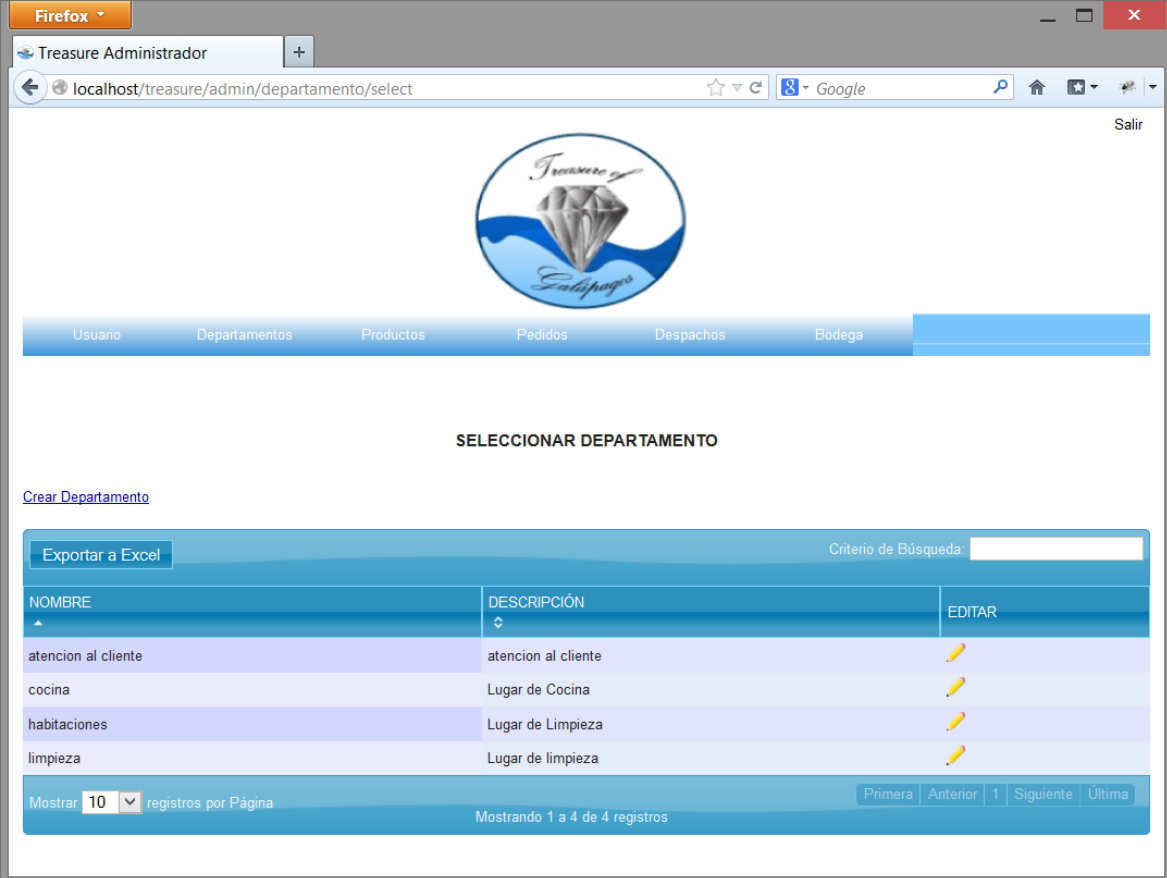
Ilustración 67: Ventana Nuevo Departamento Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de departamento: Máximo de 50 caracteres.	Nombre de departamento: + 50 caracteres.	Mensaje con lo que falta de llenar
Descripción	Descripción de departamento: Máximo de n caracteres.	Descripción de departamento: + n caracteres.	Mensaje con lo que falta de llenar

Tabla 15: Ingreso Nuevo Departamento Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Departamento:



The screenshot shows a web browser window titled 'Treasure Administrador' with the URL 'localhost/treasure/admin/departamento/select'. The page features a logo for 'Treasure of Galapagos' and a navigation menu with options: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', 'Bodega', and 'Salir'. The main content area is titled 'SELECCIONAR DEPARTAMENTO' and includes a link 'Crear Departamento'. Below this is a table with columns 'NOMBRE', 'DESCRIPCIÓN', and 'EDITAR'. The table contains four rows of department data. At the bottom, there is a pagination control showing 'Mostrar 10 registros por Página' and 'Mostrando 1 a 4 de 4 registros'.

NOMBRE	DESCRIPCIÓN	EDITAR
atencion al cliente	atencion al cliente	
cocina	Lugar de Cocina	
habitaciones	Lugar de Limpieza	
limpieza	Lugar de limpieza	

Ilustración 68: Selección Departamento (Administrador). Autor: Alex Nieto.

En esta ventana al igual que en la de usuario, tiene 2 funciones generar un reporte de los departamentos que están ingresados en el sistema y para elección de edición de cualquier departamento.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de edición de Departamento:

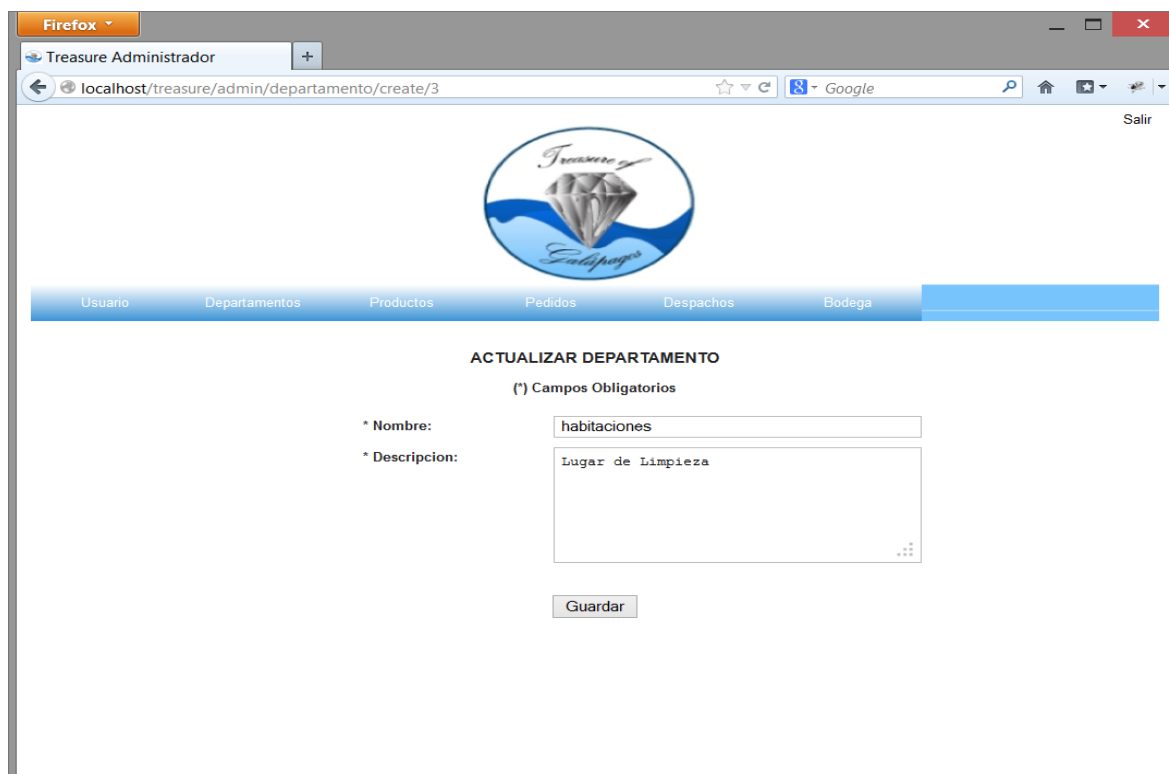


Ilustración 69: Edición Departamento (Administrador). Autor: Alex Nieto.

En esta ventana se edita el departamento seleccionado en la ventana anterior y al guardar genera los cambios al departamento seleccionado.

Ingreso:

ID_DEPARTAMENTO	NOMBRE	DESCRIPCION
1	cocina	Lugar de Cocina
2	limpieza	Lugar de limpieza
3	habitaciones	Lugar de Limpieza
7	atencion al cliente	atencion al cliente

Ilustración 70: Ingreso Edición Departamento (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de departamento: Máximo de 50 caracteres.	Nombre de departamento: + 50 caracteres.	Registro almacenado en base de datos Campo VARCHAR(50)
Descripción	Descripción de departamento: Máximo de n caracteres.	Descripción de departamento: + n caracteres.	Registro almacenado en base de datos Campo TEXT

Tabla 16: Ingreso Edición Departamento (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario:

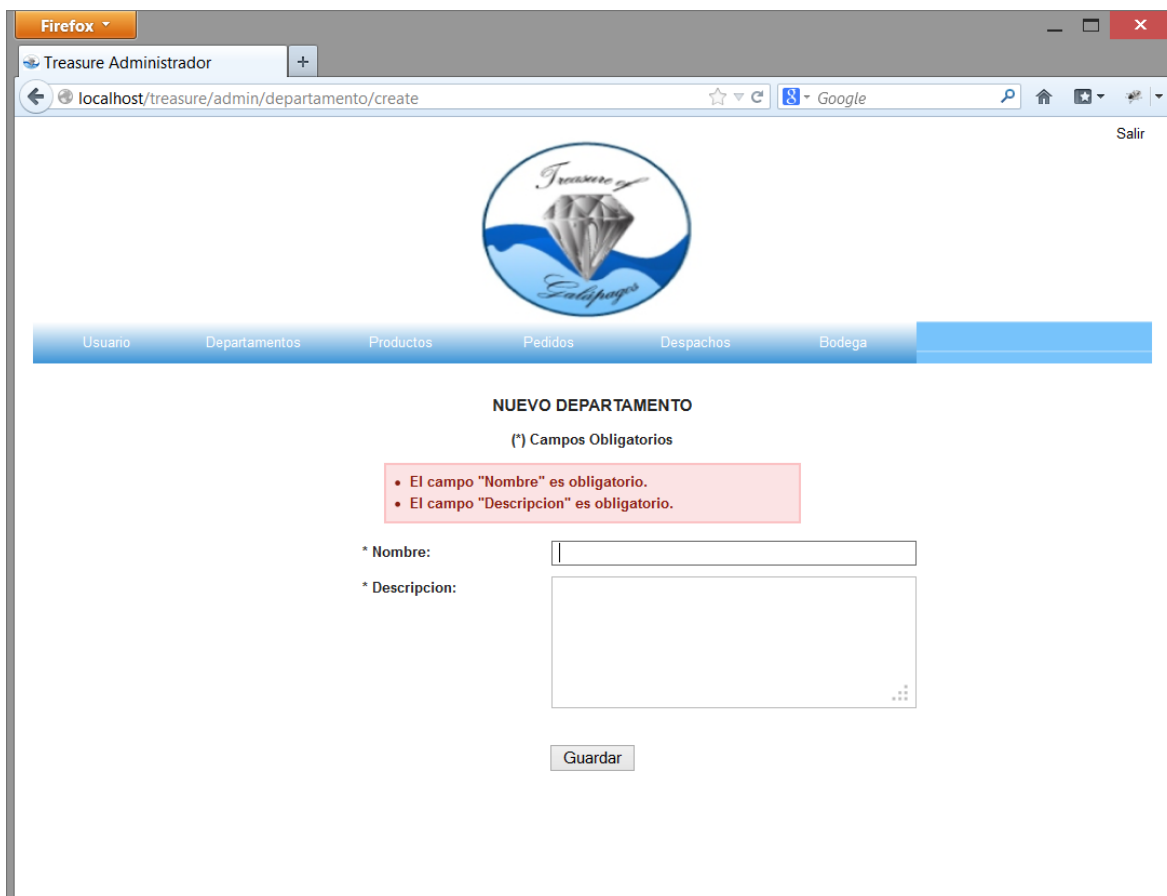


Ilustración 71: Edición Departamento Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de departamento: Máximo de 50 caracteres.	Nombre de departamento: + 50 caracteres.	Mensaje con lo que falta de llenar
Descripción	Descripción de departamento: Máximo de n caracteres.	Descripción de departamento: + n caracteres.	Mensaje con lo que falta de llenar

Tabla 17: Edición Departamento Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de nuevo Producto:

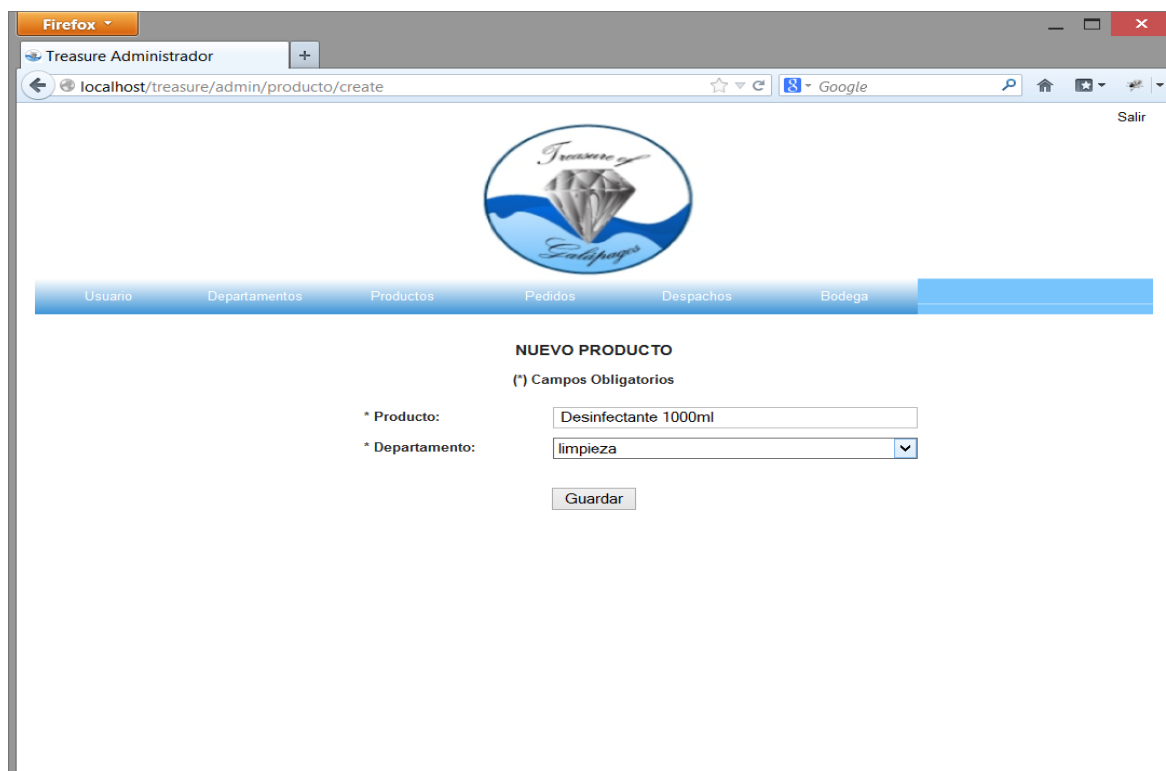


Ilustración 72: Nuevo Producto (Administrador). Autor: Alex Nieto.

En esta ventana se ingresa el producto y se direcciona a qué departamento pertenece una vez lleno se guarda en el sistema.

Ingreso:

	ID_PRODUCTO	ID_DEPARTAMENTO	DESCRIPCION
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	1	carne 1kg
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	3	Desinfectante 1000ml
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	2	detergente 50gr

Ilustración 73: Ingreso Nuevo Producto (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Nombre de producto: Máximo de 50 caracteres.	Nombre de producto: + 50 caracteres.	Registro almacenado en base de datos Campo TEXT
Departamento	Departamento de producto: Máximo de n caracteres.	Departamento de producto: Sin errores truncado con Combo Box	Registro almacenado en base de datos Campo Int(11)

Tabla 18: Ingreso Nuevo Producto (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

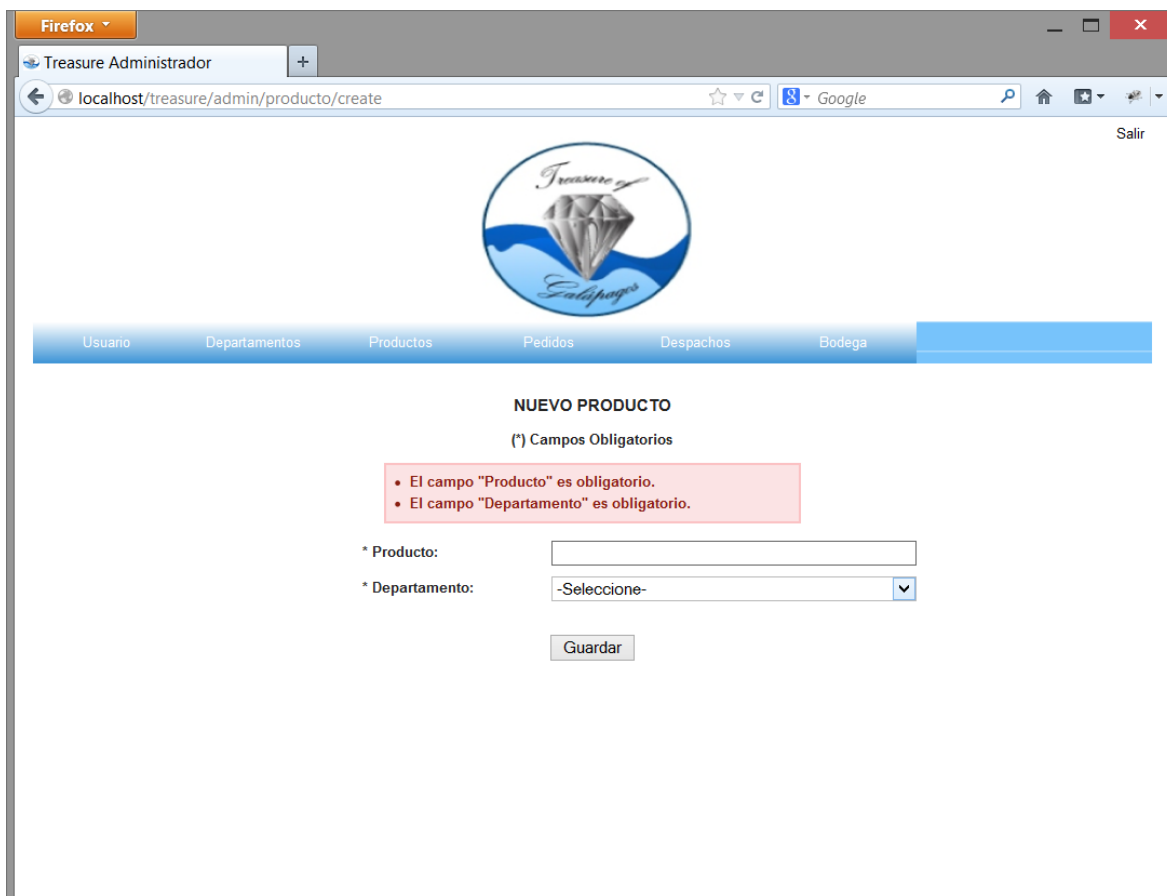


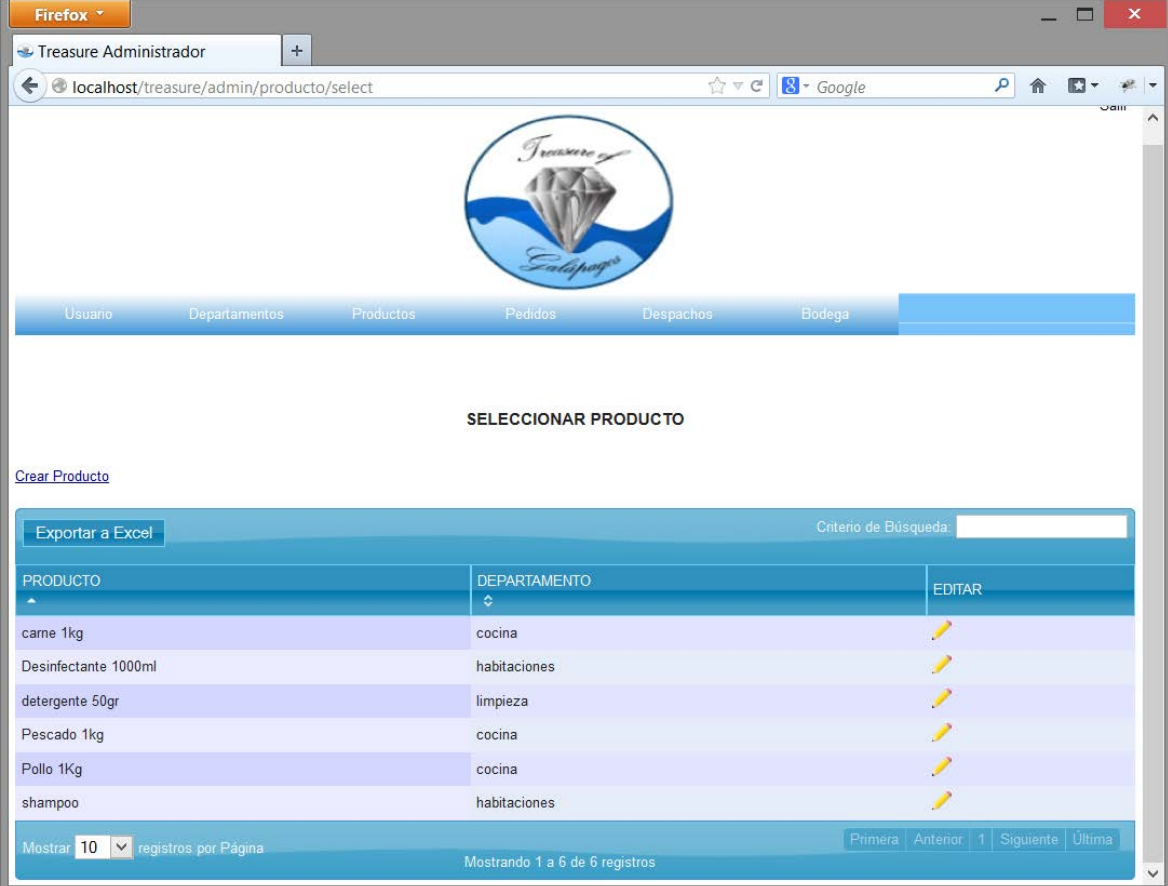
Ilustración 74: Nuevo Producto Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de producto: Máximo de 50 caracteres.	Nombre de producto: + 50 caracteres.	Mensaje con lo que falta de llenar
Descripción	Departamento de producto: Máximo de n caracteres.	Departamento de producto: Sin errores truncado con Combo Box	Mensaje con lo que falta de llenar







Tabla 19: Nuevo Producto Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de producto:



The screenshot displays the 'Selección de Producto' interface within the 'Treasure Administrador' application. The browser address bar shows 'localhost/treasure/admin/producto/select'. The application header includes a logo and a navigation menu with options: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', and 'Bodega'. The main content area is titled 'SELECCIONAR PRODUCTO' and features a 'Crear Producto' link. Below this is a search bar labeled 'Criterio de Búsqueda:' and an 'Exportar a Excel' button. A table lists the following products and departments:

PRODUCTO	DEPARTAMENTO	EDITAR
carne 1kg	cocina	
Desinfectante 1000ml	habitaciones	
detergente 50gr	limpieza	
Pescado 1kg	cocina	
Pollo 1Kg	cocina	
shampoo	habitaciones	

At the bottom of the table, there is a pagination control showing 'Mostrar 10 registros por Página' and 'Mostrando 1 a 6 de 6 registros', along with navigation buttons for 'Primera', 'Anterior', '1', 'Siguiente', and 'Última'.

Ilustración 75: Selección Producto (Administrador). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de qué productos existen en el sistema y a su vez la posibilidad de elegir alguno para su edición.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de edición de productos:

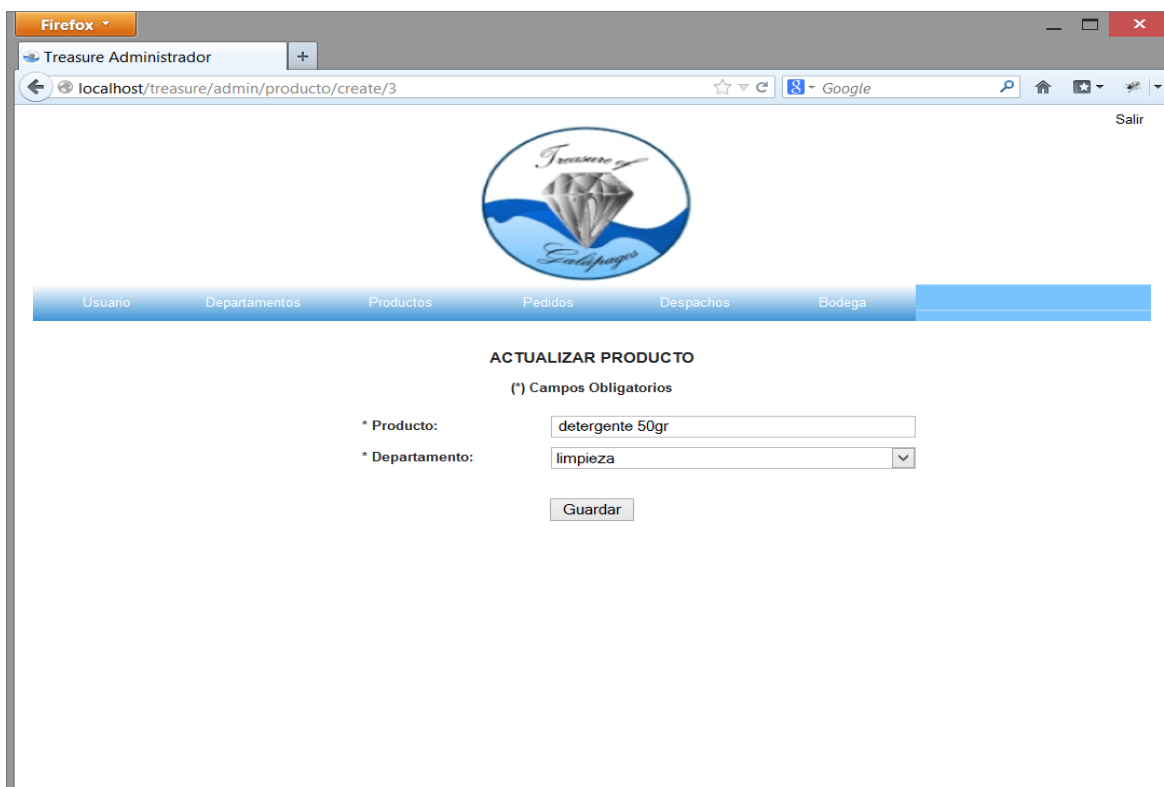


Ilustración 76: Edición Producto (Administrador). Autor: Alex Nieto.

Esta ventana muestra los datos del producto seleccionado en la ventana anterior la cual se edita o se cambia de departamento para uso en el sistema.

Ingreso:

	ID_PRODUCTO	ID_DEPARTAMENTO	DESCRIPCION
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	1	carne 1kg
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	3	Desinfectante 1000ml
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	2	detergente 50gr

Ilustración 77: Ingreso Edición Producto (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Nombre de producto: Máximo de 50 caracteres.	Nombre de producto: + 50 caracteres.	Registro almacenado en base de datos Campo TEXT
Departamento	Departamento de producto: Máximo de n caracteres.	Departamento de producto: Sin errores truncado con Combo Box	Registro almacenado en base de datos Campo Int(11)

Tabla 20: Ingreso Edición Producto (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

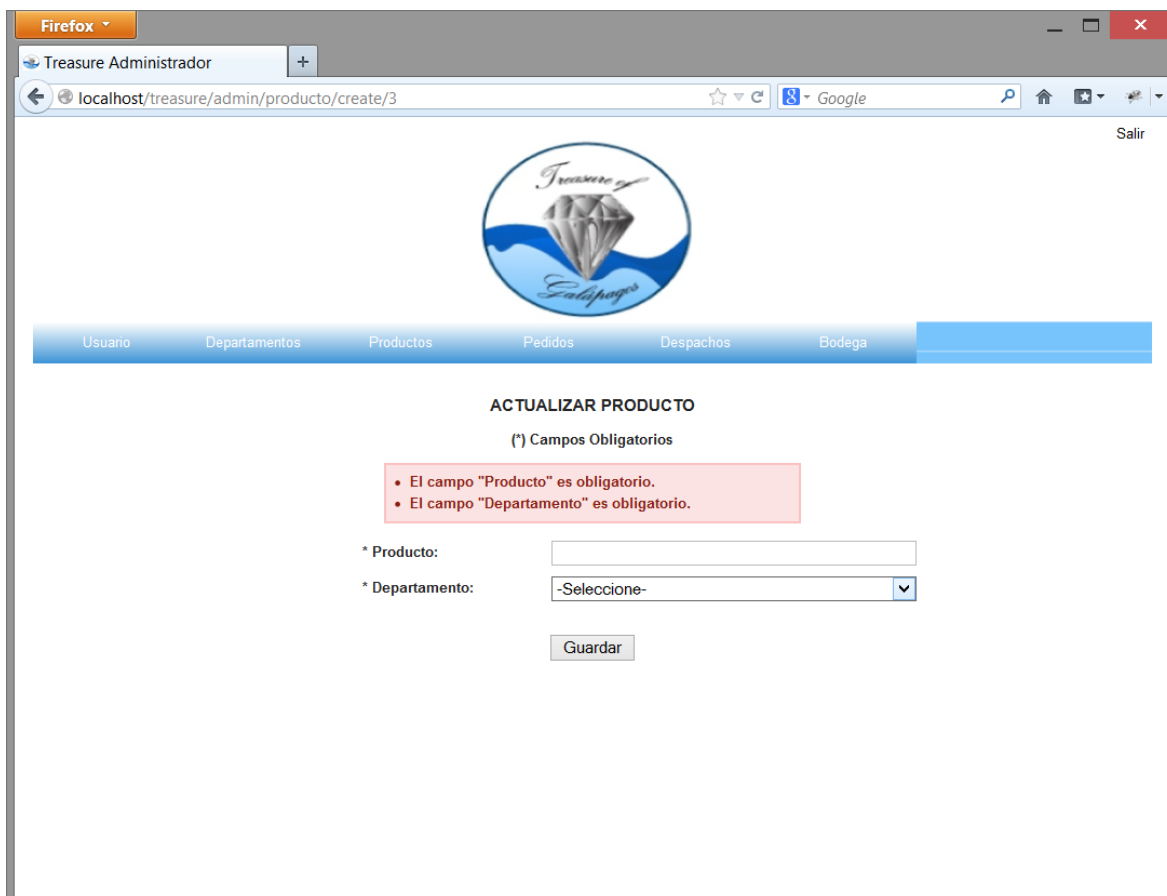


Ilustración 78: Edición Producto Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Nombre	Nombre de producto: Máximo de 50 caracteres.	Nombre de producto: + 50 caracteres.	Mensaje con lo que falta de llenar
Descripción	Departamento de producto: Máximo de n caracteres.	Departamento de producto: Sin errores truncado con Combo Box	Mensaje con lo que falta de llenar

Tabla 21: Edición Producto Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de nuevo Pedido:

Ilustración 79: Nuevo Pedido (Administrador). Autor: Alex Nieto.

En esta ventana se ingresa los productos para generar un nuevo pedido al cual después puede ser cambiado de “pedido” a “aprobado” o “cancelado”.

Ingreso:

Tabla pedidos

← T →				ID_PEDIDO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO
<input type="checkbox"/>	Editar	Copiar	Borrar	10	1	5	3	2013	20	10
<input type="checkbox"/>	Editar	Copiar	Borrar	11	1	6	3	2013	14	20
<input type="checkbox"/>	Editar	Copiar	Borrar	12	1	8	3	2013	0	15
<input type="checkbox"/>	Editar	Copiar	Borrar	14	1	8	3	2013	0	17
<input type="checkbox"/>	Editar	Copiar	Borrar	15	1	8	3	2013	0	18
<input type="checkbox"/>	Editar	Copiar	Borrar	16	1	8	3	2013	0	43
<input type="checkbox"/>	Editar	Copiar	Borrar	17	1	8	3	2013	19	30
<input type="checkbox"/>	Editar	Copiar	Borrar	18	1	11	3	2013	21	30

Ilustración 80: Ingreso 1 Nuevo Pedido (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Tabla Pedido Despacho X Producto

←T→		ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	 Editar  Copiar  Borrar	35	18	2	NULL	5	PEDIDO
<input type="checkbox"/>	 Editar  Copiar  Borrar	36	18	6	NULL	1	PEDIDO

Ilustración 81: Ingreso 2 Nuevo Pedido (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Selección de productos: Los productos en lista.	Selección de productos: Sin elección de productos en lista.	Registro almacenado en base de datos Campo Int(11)
Cantidad	Cantidad de producto: Máximo 2 dígitos.	Cantidad de producto: + de 2 dígitos. - de 1 dígito.	Registro almacenado en base de datos Campo Int(11)

Tabla 22: Ingreso Nuevo Pedido (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

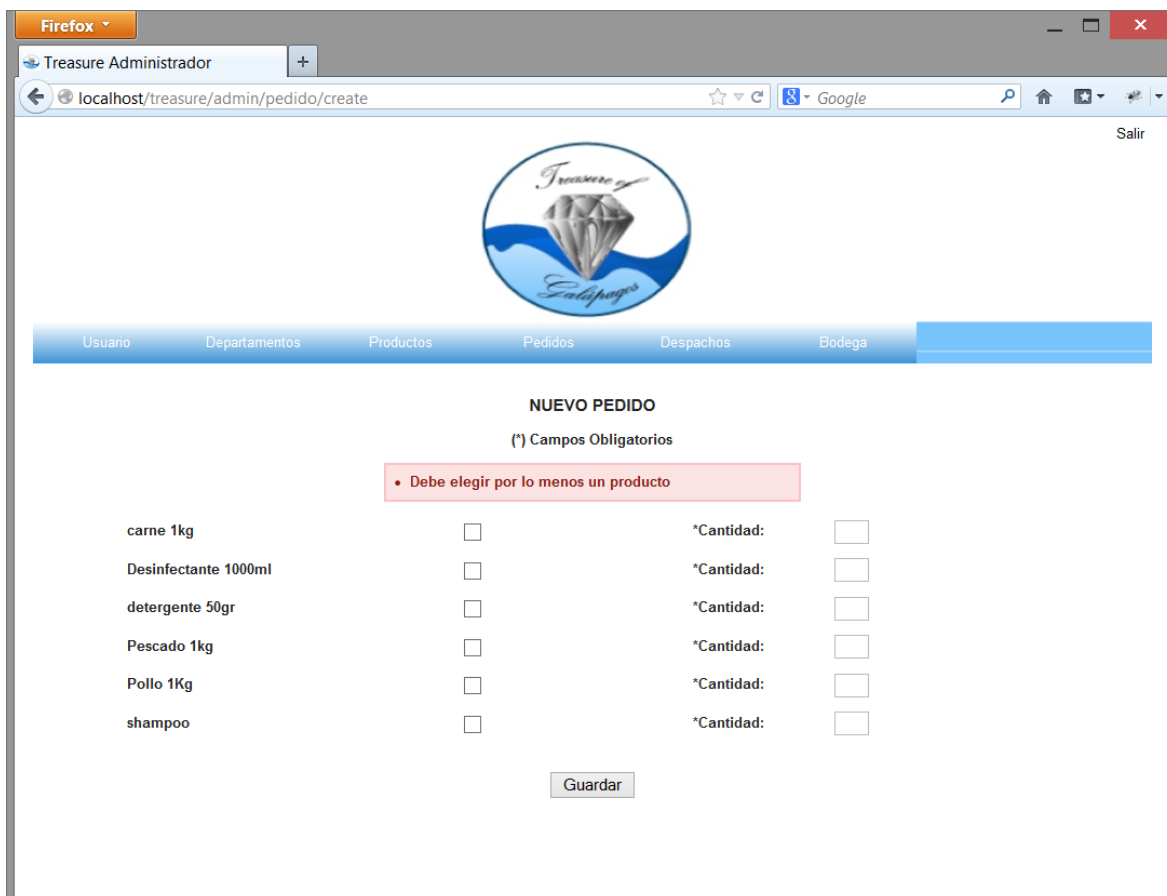


Ilustración 82: Nuevo Pedido Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Selección de productos: Los productos en lista.	Selección de productos: Sin elección de productos en lista.	Mensaje con lo que falta de llenar
Cantidad	Cantidad de producto: Máximo 2 dígitos.	Cantidad de producto: + de 2 dígitos. - de 1 dígito.	Mensaje con lo que falta de llenar

Tabla 23: Nuevo Pedido Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Pedidos:

The screenshot shows a web application interface for 'Treasure of Galapagos'. The browser address bar indicates the URL is localhost/treasure/admin/pedido/select. The page features a navigation menu with options: Usuario, Departamentos, Productos, Pedidos (highlighted), Despachos, and Bodega. Below the menu, the title 'SELECCIONAR PEDIDO PARA MODIFICAR' is centered. A link 'Crear Pedido' is visible on the left. The main content area contains a table with columns: NÚMERO DE PEDIDO, USUARIO, ANIO, MES, DIA, and EDITAR. The table lists three records. Below the table, there is a pagination control showing 'Mostrar 10 registros por Página' and 'Mostrando 1 a 3 de 3 registros'. A search bar labeled 'Criterio de Búsqueda:' is located at the top right of the table area.

NÚMERO DE PEDIDO	USUARIO	ANIO	MES	DIA	EDITAR
18	administradoruno administradoruno	2013	3	11	
25	solicitante solicitante	2013	3	14	
26	solicitantedos solicitantedos	2013	3	15	

Ilustración 83: Selección Pedido Validaciones (Administrador). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los pedidos existentes en el sistema y a su vez la posibilidad de elegir alguno para su edición.

Ventana de edición de Pedidos:

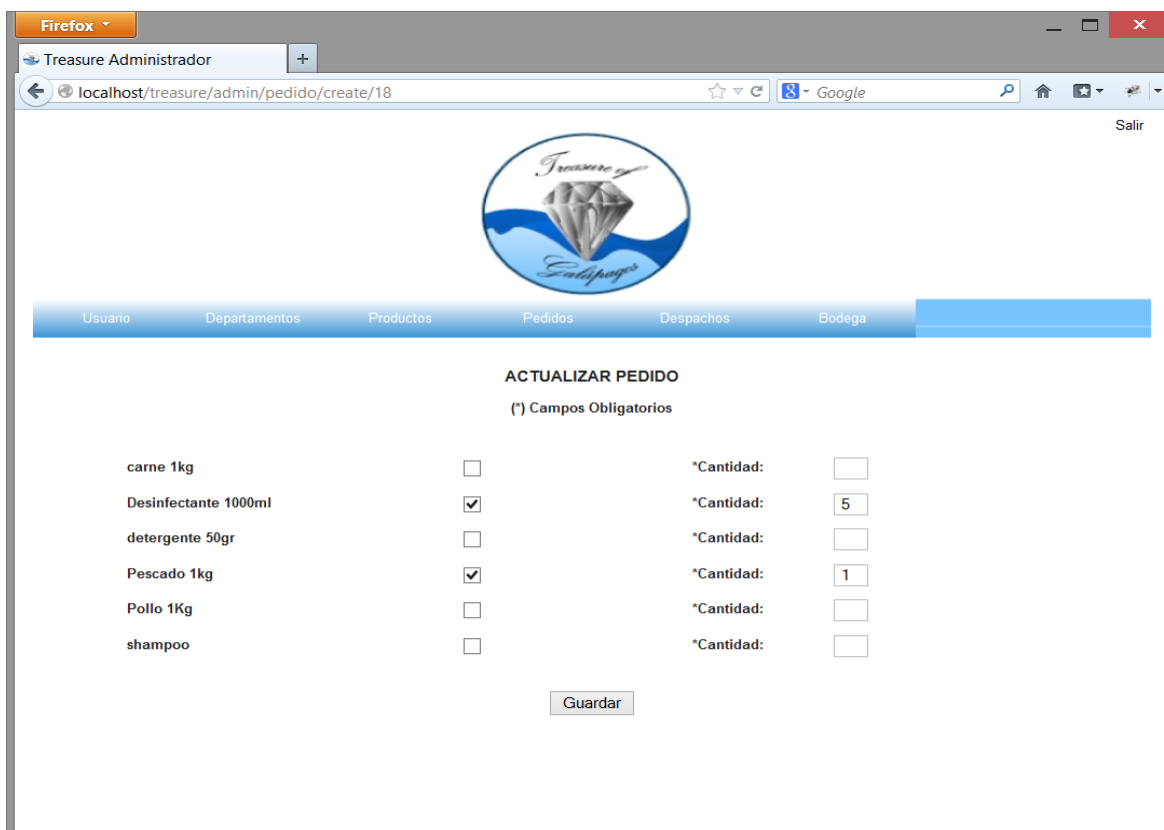


Ilustración 84: Edición Pedido (Administrador). Autor: Alex Nieto.

Esta ventana muestra los datos del Pedido seleccionado en la ventana anterior la cual se edita o se cambia de productos y cantidad de los mismos para ingresarlos nuevamente en el sistema.

Ingreso:

Tabla pedidos

				ID_PEDIDO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO			
<input type="checkbox"/>		Editar		Copiar		Borrar	10	1	5	3	2013	20	10
<input type="checkbox"/>		Editar		Copiar		Borrar	11	1	6	3	2013	14	20
<input type="checkbox"/>		Editar		Copiar		Borrar	12	1	8	3	2013	0	15
<input type="checkbox"/>		Editar		Copiar		Borrar	14	1	8	3	2013	0	17
<input type="checkbox"/>		Editar		Copiar		Borrar	15	1	8	3	2013	0	18
<input type="checkbox"/>		Editar		Copiar		Borrar	16	1	8	3	2013	0	43
<input type="checkbox"/>		Editar		Copiar		Borrar	17	1	8	3	2013	19	30
<input type="checkbox"/>		Editar		Copiar		Borrar	18	1	11	3	2013	21	30

Ilustración 85: Ingreso 1 Edición Pedido (Administrador). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

		ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	Editar Copiar Borrar	35	18	2	NULL	5	PEDIDO
<input type="checkbox"/>	Editar Copiar Borrar	36	18	6	NULL	1	PEDIDO

Ilustración 86: Ingreso 2 Edición Pedido (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Selección de productos: Los productos en lista.	Selección de productos: Sin elección de productos en lista.	Registro almacenado en base de datos Campo Int(11)
Cantidad	Cantidad de producto: Máximo 2 dígitos.	Cantidad de producto: + de 2 dígitos. - de 1 dígito.	Registro almacenado en base de datos Campo Int(11)

Tabla 24: Ingreso Edición Pedido (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

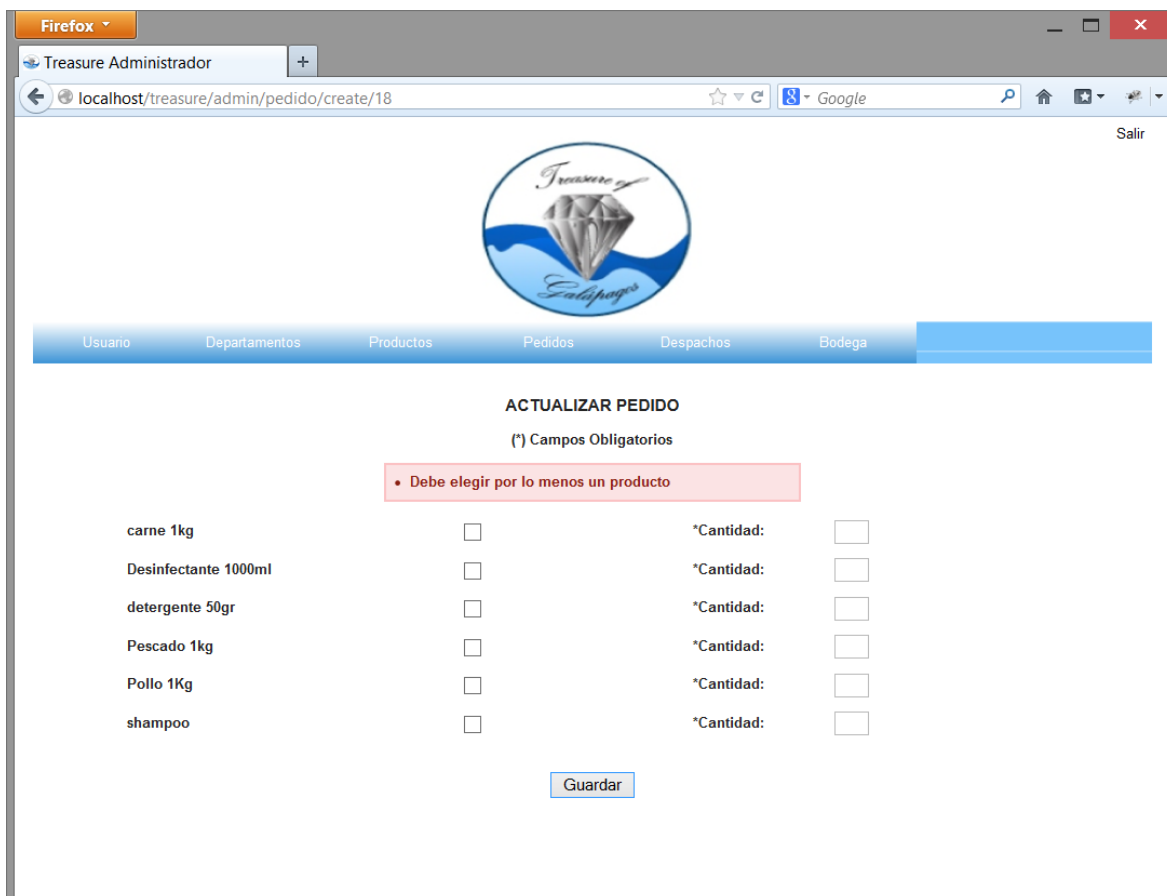


Ilustración 87: Edición Pedido Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Selección de productos: Los productos en lista.	Selección de productos: Sin elección de productos en lista.	Mensaje con lo que falta de llenar
Cantidad	Cantidad de producto: Máximo 2 dígitos.	Cantidad de producto: + de 2 dígitos. - de 1 dígito.	Mensaje con lo que falta de llenar

Tabla 25: Edición Pedido Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Pedidos para Aprobación:

The screenshot shows a web application interface for 'Treasure Administrador'. The browser address bar shows 'localhost/treasure/admin/pedido/aprobar'. The page features a navigation menu with options: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', and 'Bodega'. Below the menu is a search bar labeled 'Criterio de Búsqueda:'. The main content area is titled 'SELECCIONAR PEDIDO PARA APROBACIÓN' and includes a 'Crear Pedido' link and an 'Exportar a Excel' button. A table displays three pending orders:

NÚMERO DE PEDIDO	USUARIO	ANIO	MES	DIA	REVISAR
18	administradoruno administradoruno	2013	3	11	
25	solicitante solicitante	2013	3	14	
26	solicitantados solicitantados	2013	3	15	

At the bottom, there is a pagination control showing 'Mostrar 10 registros por Página' and 'Mostrando 1 a 3 de 3 registros', with navigation buttons for 'Primera', 'Anterior', '1', 'Siguiente', and 'Última'.

Ilustración 88: Aprobación Pedido (Administrador). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los pedidos que se encuentran en estado de "Pedido" pero aún no están aprobados en el sistema y a su vez la posibilidad de elegir alguno para su aprobación o cancelación.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de revisión de Pedidos:

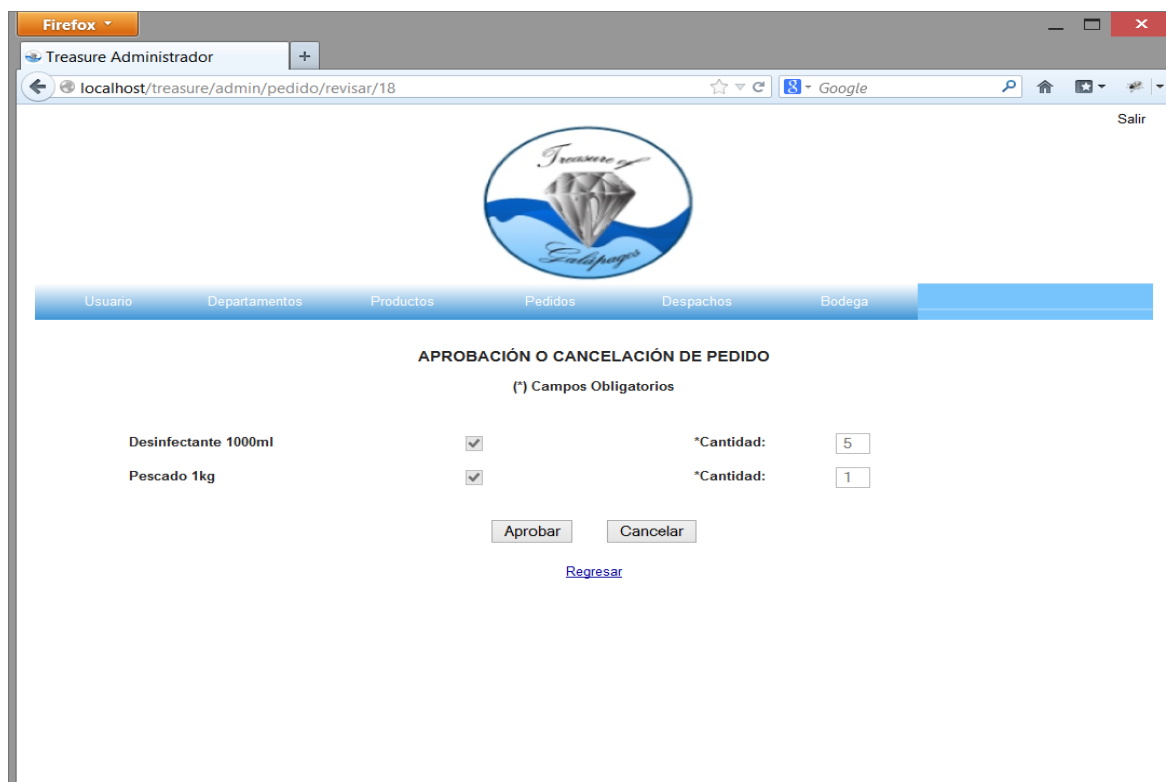


Ilustración 89: Revisión Pedido (Administrador). Autor: Alex Nieto.

Esta ventana muestra los datos del pedido seleccionado en la ventana anterior la cual puede "Aprobar" o "Cancelar" el pedido seleccionado.

Ingreso:

Tabla pedidos

← T →				ID_PEDIDO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO
<input type="checkbox"/>				10	1	5	3	2013	20	10
<input type="checkbox"/>				11	1	6	3	2013	14	20
<input type="checkbox"/>				12	1	8	3	2013	0	15
<input type="checkbox"/>				14	1	8	3	2013	0	17
<input type="checkbox"/>				15	1	8	3	2013	0	18
<input type="checkbox"/>				16	1	8	3	2013	0	43
<input type="checkbox"/>				17	1	8	3	2013	19	30
<input type="checkbox"/>				18	1	11	3	2013	21	30

Ilustración 90: Ingreso 1 Revisión Pedido (Administrador). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

		ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	  	35	18	2	NULL	5	APROBADO
<input type="checkbox"/>	  	36	18	6	NULL	1	APROBADO

Ilustración 91: Ingreso 2 Revisión Pedido (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Aprobar	Clic en aceptar confirmación.	Aprobar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('APROBADO')
Botón Cancelar	Clic en aceptar confirmación.	Cancelar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('CANCELADO')

Tabla 26: Ingreso Revisión Pedido (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

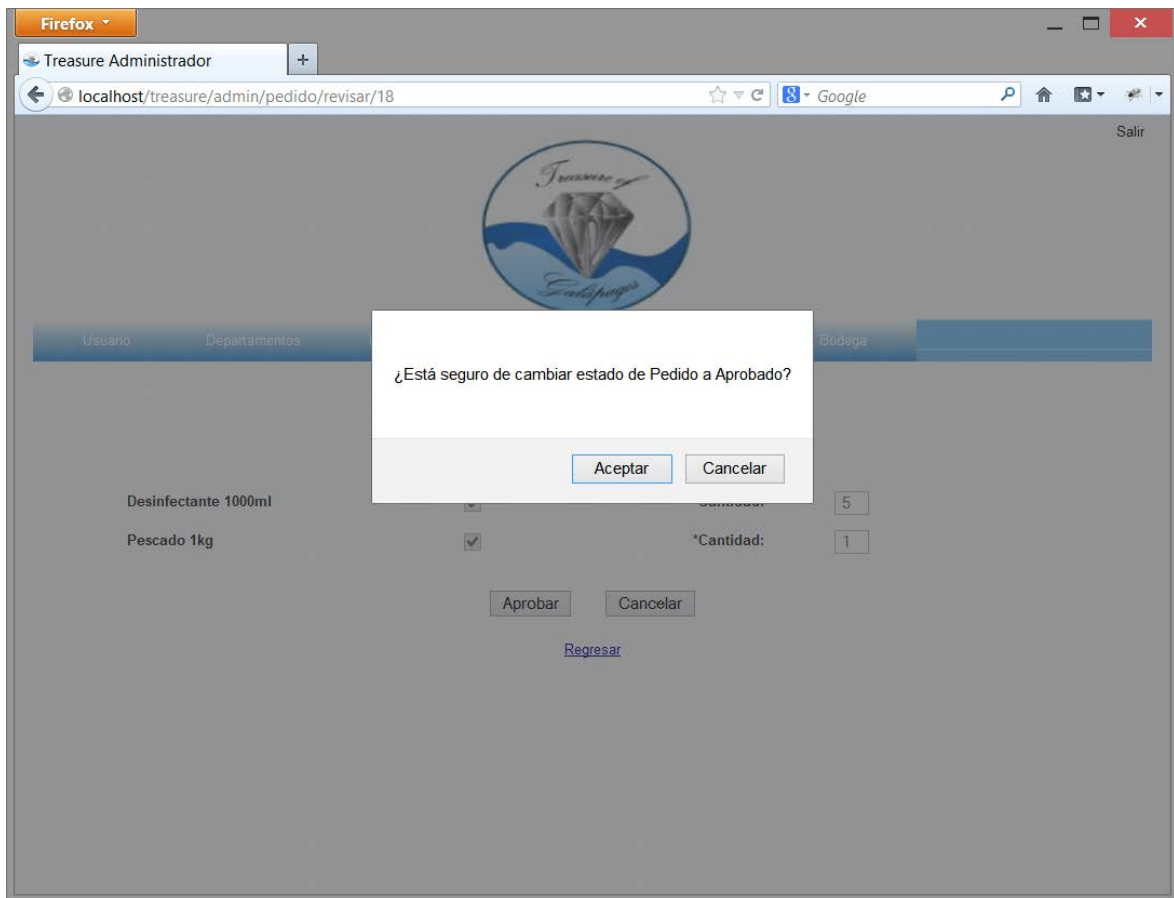


Ilustración 92: Revisión Pedido Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Aprobar	Clic en aceptar confirmación.	Aprobar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar
Botón Cancelar	Clic en aceptar confirmación.	Cancelar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar

Tabla 27: Revisión Pedido Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de nuevo Despacho:

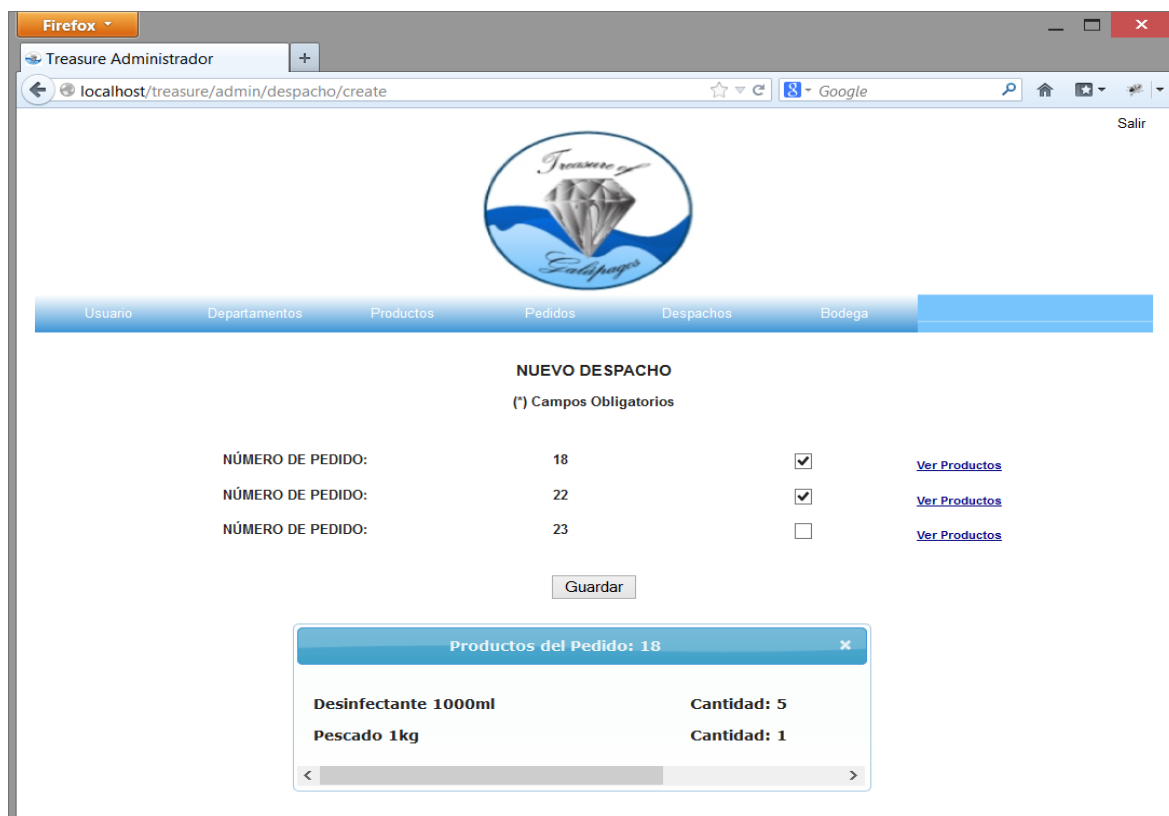


Ilustración 93: Nuevo Despacho (Administrador). Autor: Alex Nieto.

En esta ventana se selecciona los pedidos “Aprobados” con la opción de revisar los productos que contiene cada pedido.

Ingreso:

Tabla despachos

← T →				ID_DESPACHO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO
<input type="checkbox"/>	Editar	Copiar	Borrar	1	1	7	3	2013	18	22
<input type="checkbox"/>	Editar	Copiar	Borrar	3	1	8	3	2013	0	37
<input type="checkbox"/>	Editar	Copiar	Borrar	4	1	8	3	2013	19	33
<input type="checkbox"/>	Editar	Copiar	Borrar	6	1	9	3	2013	1	28
<input type="checkbox"/>	Editar	Copiar	Borrar	7	1	9	3	2013	1	31
<input type="checkbox"/>	Editar	Copiar	Borrar	8	1	9	3	2013	1	35
<input type="checkbox"/>	Editar	Copiar	Borrar	9	1	9	3	2013	1	42
<input type="checkbox"/>	Editar	Copiar	Borrar	10	1	11	3	2013	21	34
<input type="checkbox"/>	Editar	Copiar	Borrar	11	1	22	3	2013	0	17

Ilustración 94: Ingreso 1 Nuevo Despacho (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Tabla Pedido Despacho X Producto










← T →				ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	35	18	2	11	5	EN PROCESO
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	36	18	6	11	1	EN PROCESO
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	38	22	1	11	1	EN PROCESO

Ilustración 95: Ingreso 2 Nuevo Despacho (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Pedido	Selección de pedidos: Los pedidos en lista.	Selección de pedidos: Sin elección de productos en lista.	Registro almacenado en base de datos Campo Int(11)

Tabla 28: Ingreso Nuevo Despacho (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

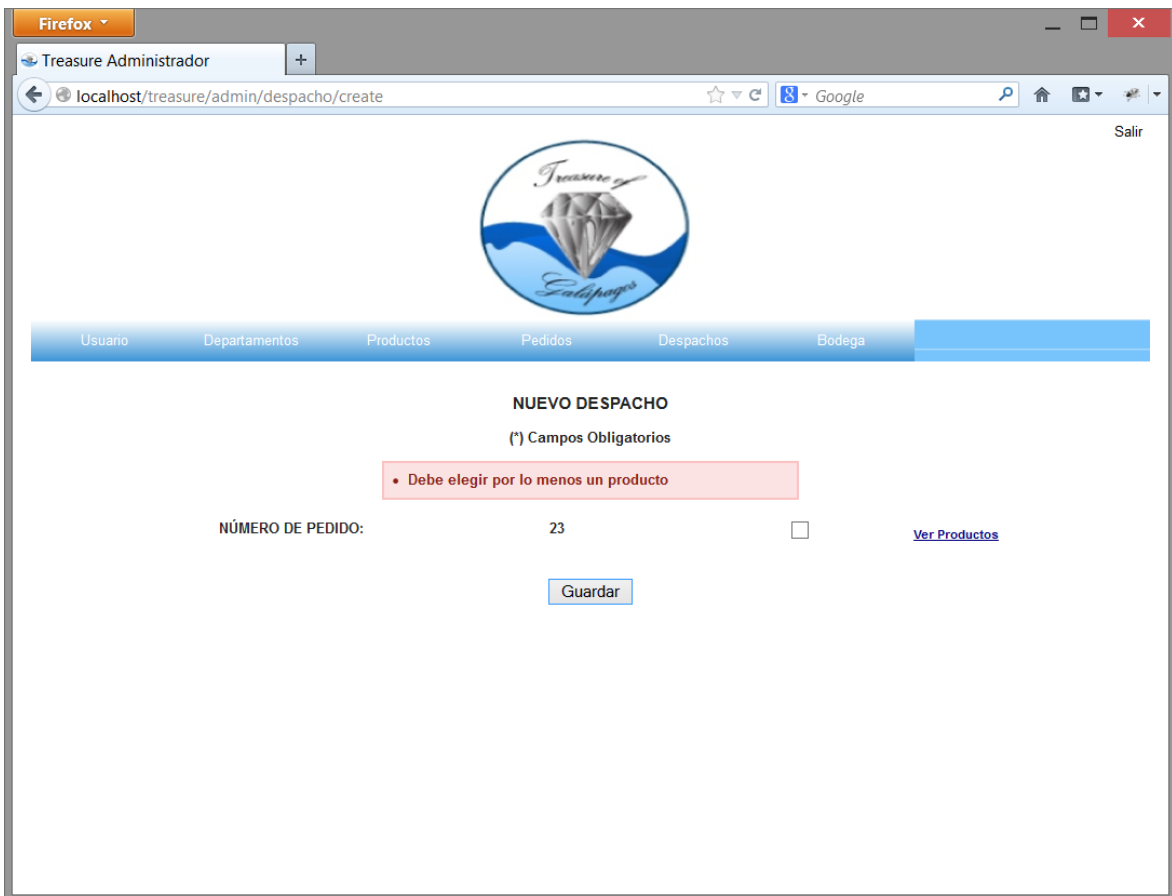


Ilustración 96: Nuevo Despacho Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Pedido	Selección de pedidos: Los pedidos en lista.	Selección de pedidos: Sin elección de productos en lista.	Mensaje con lo que falta de llenar

Tabla 29: Nuevo Despacho Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Despachos:

The screenshot shows a web application interface for 'Treasure of Galapagos'. The browser address bar shows 'localhost/treasure/admin/despacho/select'. The application has a navigation menu with options: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', and 'Bodega'. The 'Despachos' option is highlighted. Below the menu, the title 'SELECCIONAR DESPACHO PARA MODIFICAR' is displayed. There is a link 'Crear Despacho' and a button 'Exportar a Excel'. A search bar labeled 'Criterio de Búsqueda:' is present. The main content is a table with the following data:

NÚMERO DE DESPACHO	USUARIO	ANIO	MES	DIA	EDITAR
4	administradoruno administradoruno	2013	3	8	
6	administradoruno administradoruno	2013	3	9	
11	administradoruno administradoruno	2013	3	22	

At the bottom of the table, there is a pagination control showing 'Mostrar 10 registros por Página' and 'Mostrando 1 a 3 de 3 registros'. Navigation buttons for 'Primera', 'Anterior', '1', 'Siguiente', and 'Ultima' are also visible.

Ilustración 97: Selección Despacho (Administrador). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los despachos que existen en estado "EN PROCESO" en el sistema y a su vez la posibilidad de elegir alguno para su edición.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de edición de Despachos:

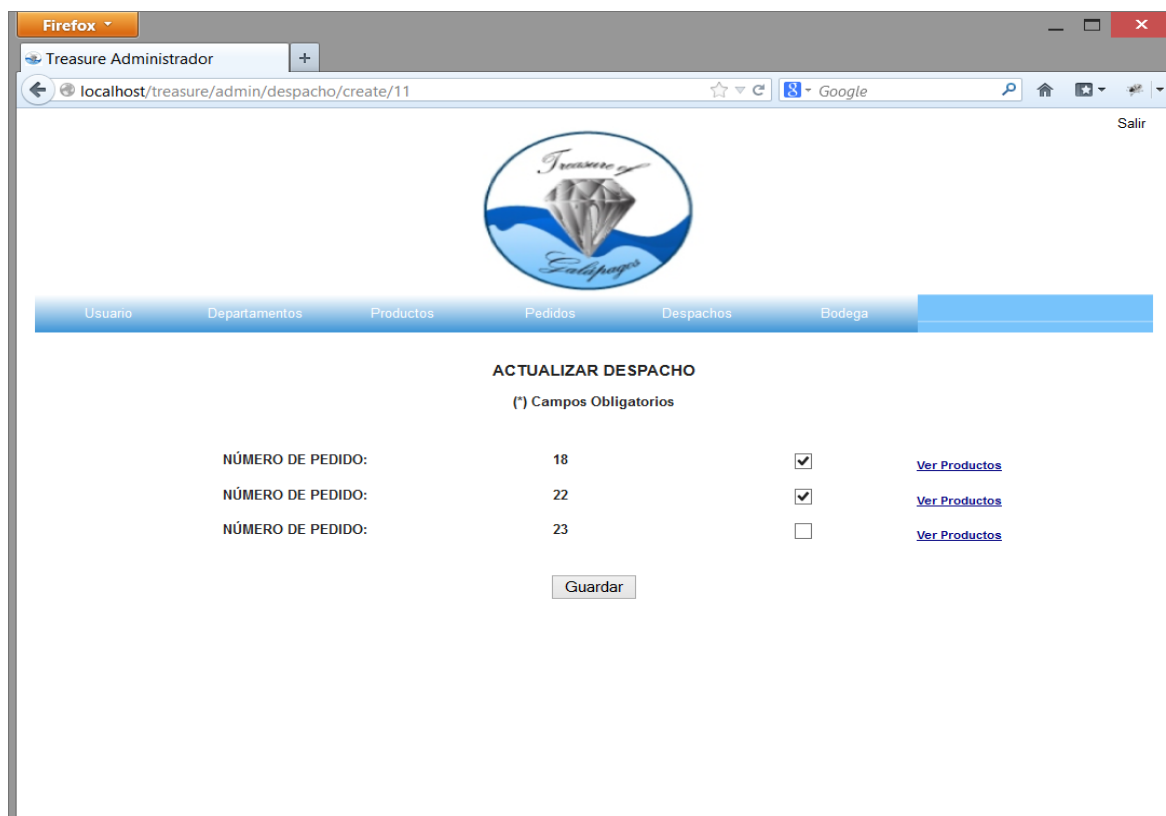


Ilustración 98: Edición Despacho (Administrador). Autor: Alex Nieto.

Esta ventana muestra los datos del despacho seleccionado en la ventana anterior, la cual edita o cambia los pedidos seleccionados para ingreso en el sistema.

Ingreso:

Tabla despachos

← T →				ID_DESPACHO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO			
<input type="checkbox"/>		Editar		Copiar		Borrar	1	1	7	3	2013	18	22
<input type="checkbox"/>		Editar		Copiar		Borrar	3	1	8	3	2013	0	37
<input type="checkbox"/>		Editar		Copiar		Borrar	4	1	8	3	2013	19	33
<input type="checkbox"/>		Editar		Copiar		Borrar	6	1	9	3	2013	1	28
<input type="checkbox"/>		Editar		Copiar		Borrar	7	1	9	3	2013	1	31
<input type="checkbox"/>		Editar		Copiar		Borrar	8	1	9	3	2013	1	35
<input type="checkbox"/>		Editar		Copiar		Borrar	9	1	9	3	2013	1	42
<input type="checkbox"/>		Editar		Copiar		Borrar	10	1	11	3	2013	21	34
<input type="checkbox"/>		Editar		Copiar		Borrar	11	1	22	3	2013	0	17

Ilustración 99: Ingreso 1 Edición Despacho (Administrador). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

← T →				ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	Editar	Copiar	Borrar	35	18	2	11	5	EN PROCESO
<input type="checkbox"/>	Editar	Copiar	Borrar	36	18	6	11	1	EN PROCESO
<input type="checkbox"/>	Editar	Copiar	Borrar	38	22	1	11	1	EN PROCESO

Ilustración 100: Ingreso 2 Edición Despacho (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Pedido	Selección de pedidos: Los pedidos en lista.	Selección de pedidos: Sin elección de productos en lista.	Registro almacenado en base de datos Campo Int(11)

Tabla 30: Ingreso Edición Despacho (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

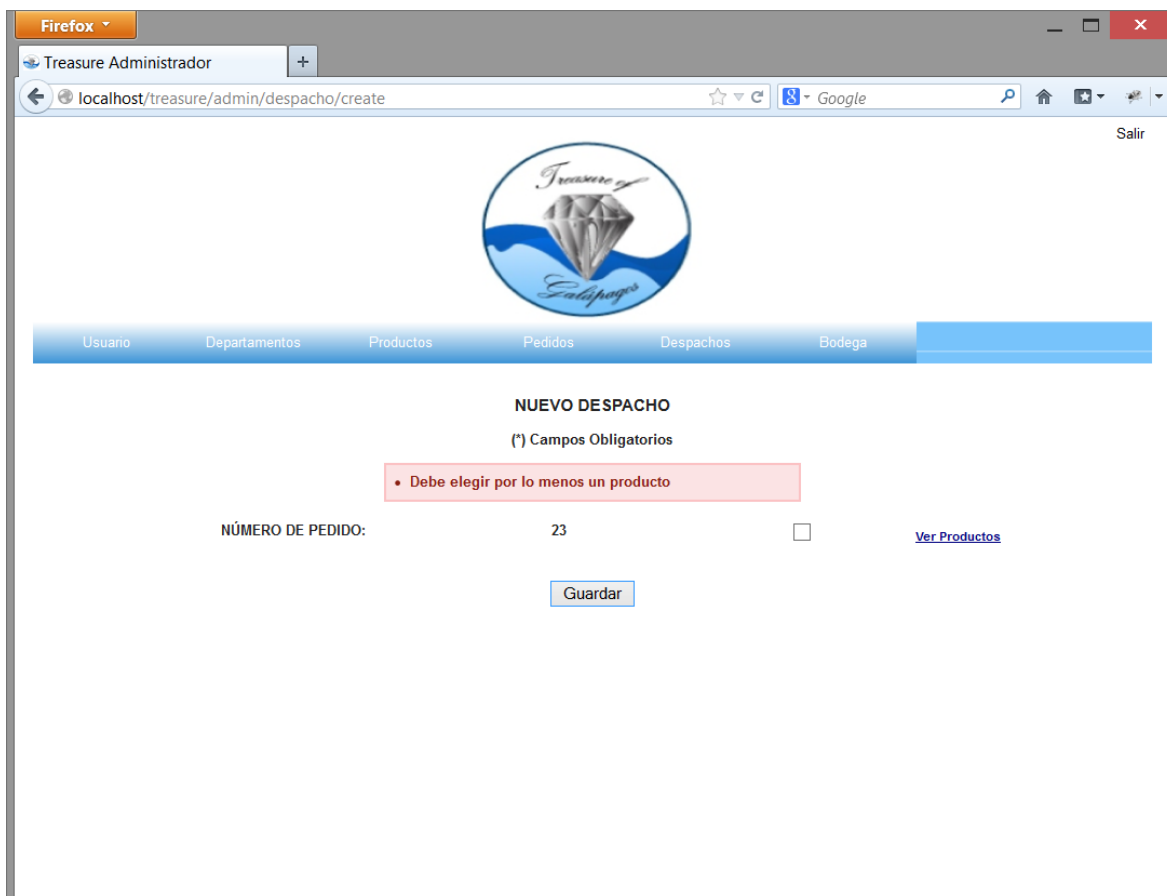


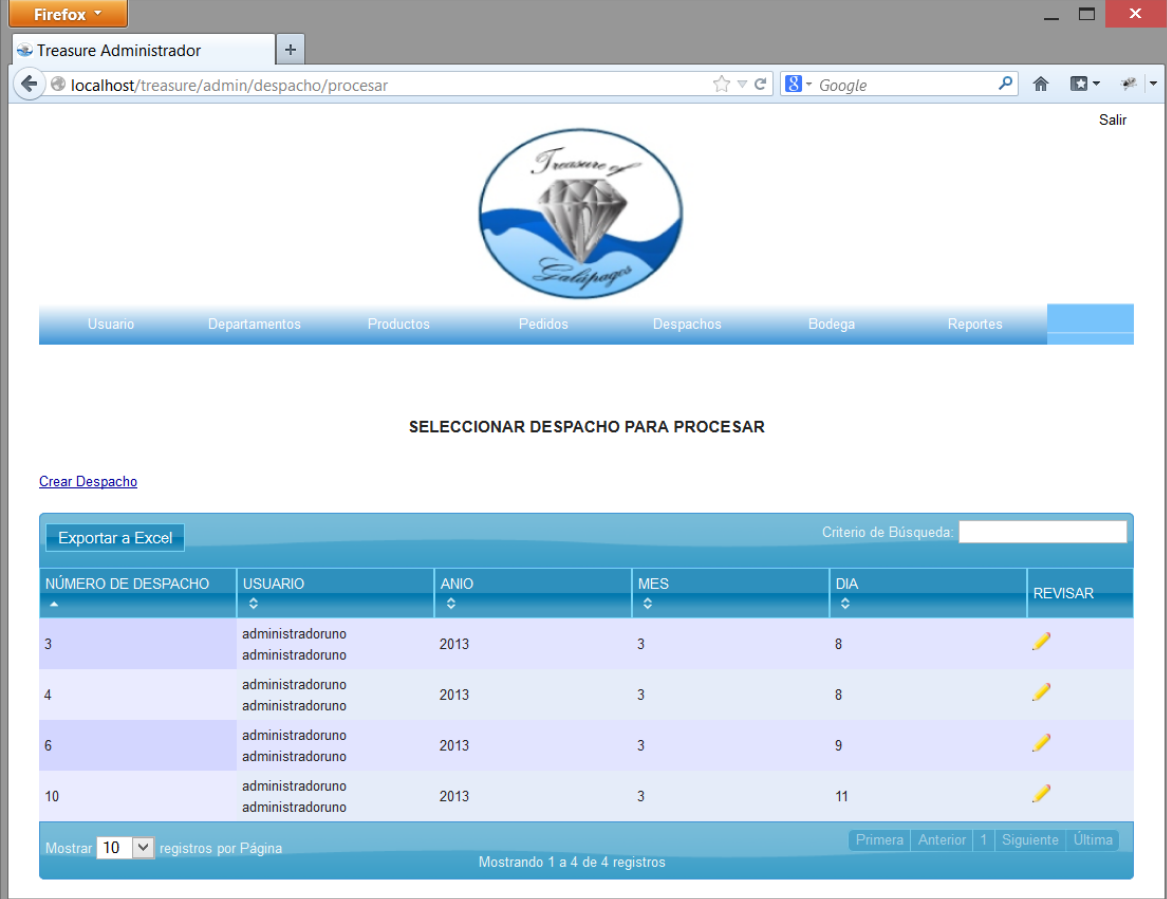
Ilustración 101: Edición Despacho Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Pedido	Selección de pedidos: Los pedidos en lista.	Selección de pedidos: Sin elección de productos en lista.	Mensaje con lo que falta de llenar

Tabla 31: Edición Despacho Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Despachos para Cambio de Estado:



The screenshot shows a web browser window with the URL `localhost/treasure/admin/despacho/procesar`. The page features a navigation menu with options: Usuario, Departamentos, Productos, Pedidos, Despachos, Bodega, Reportes, and a highlighted 'Salir' button. The main content area is titled 'SELECCIONAR DESPACHO PARA PROCESAR' and includes a 'Crear Despacho' link. Below this is a table with columns: 'NÚMERO DE DESPACHO', 'USUARIO', 'ANIO', 'MES', 'DIA', and 'REVISAR'. The table contains four rows of data, each with a yellow pencil icon in the 'REVISAR' column. At the bottom of the table, there is a pagination control showing 'Mostrar 10 registros por Página' and 'Mostrando 1 a 4 de 4 registros'.

NÚMERO DE DESPACHO	USUARIO	ANIO	MES	DIA	REVISAR
3	administradoruno administradoruno	2013	3	8	
4	administradoruno administradoruno	2013	3	8	
6	administradoruno administradoruno	2013	3	9	
10	administradoruno administradoruno	2013	3	11	

Ilustración 102: Estado Despacho (Administrador). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los despachos que se encuentran en estado de “En Proceso”, con la posibilidad de cambiar su estado.

Ventana de revisión de Despacho:

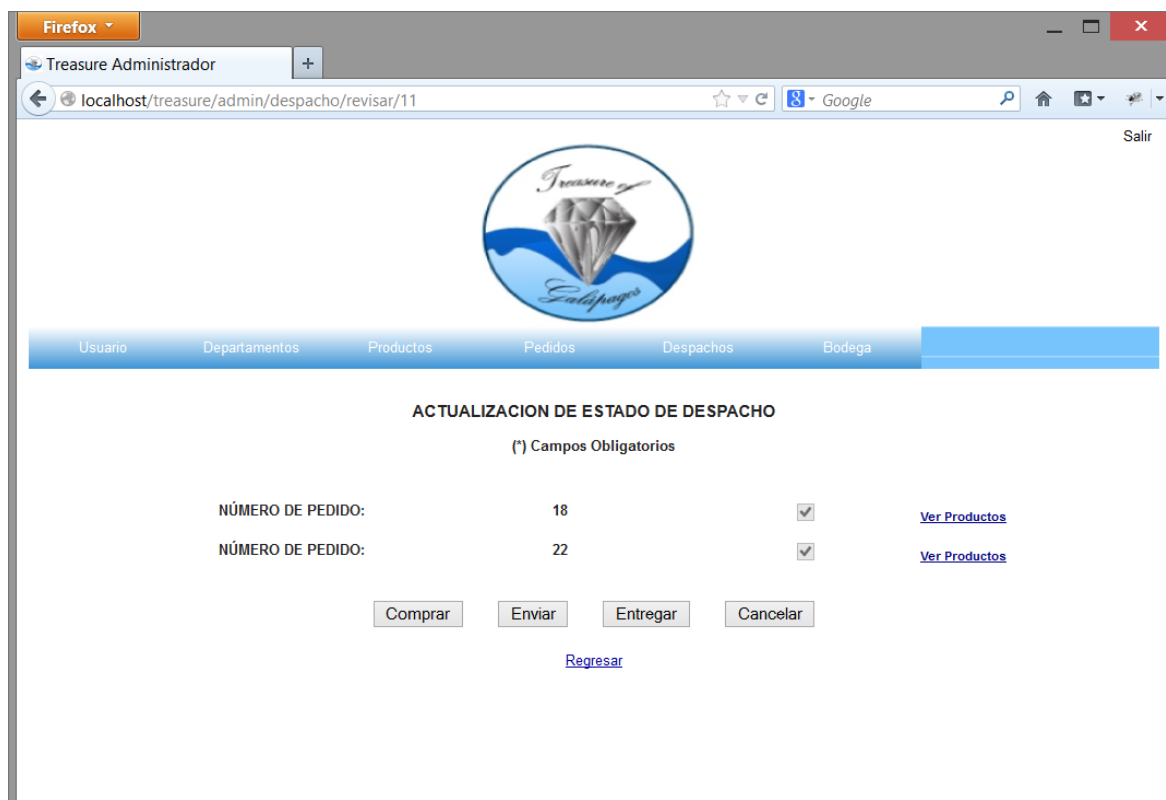


Ilustración 103: Revisión Despacho (Administrador). Autor: Alex Nieto.

Esta ventana muestra los datos del Despacho seleccionado en la ventana anterior, la cual puede cambiar su estado a: “Comprar”, “Enviar”, “Entregar” o “Cancelar”.

Ingreso:

Tabla Despachos

				ID_DESPACHO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO
<input type="checkbox"/>	Editar	Copiar	Borrar	1	1	7	3	2013	18	22
<input type="checkbox"/>	Editar	Copiar	Borrar	3	1	8	3	2013	0	37
<input type="checkbox"/>	Editar	Copiar	Borrar	4	1	8	3	2013	19	33
<input type="checkbox"/>	Editar	Copiar	Borrar	6	1	9	3	2013	1	28
<input type="checkbox"/>	Editar	Copiar	Borrar	7	1	9	3	2013	1	31
<input type="checkbox"/>	Editar	Copiar	Borrar	8	1	9	3	2013	1	35
<input type="checkbox"/>	Editar	Copiar	Borrar	9	1	9	3	2013	1	42
<input type="checkbox"/>	Editar	Copiar	Borrar	10	1	11	3	2013	21	34
<input type="checkbox"/>	Editar	Copiar	Borrar	11	1	22	3	2013	0	17

Ilustración 104: Ingreso 1 Revisión Despacho (Administrador). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

← T →				ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>				35	18	2	11	5	COMPRADO
<input type="checkbox"/>				36	18	6	11	1	COMPRADO
<input type="checkbox"/>				38	22	1	11	1	COMPRADO

Ilustración 105: Ingreso 2 Revisión Despacho (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Comprar	Clic en Comprar confirmación.	Comprar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('COMPRADO')
Botón Enviar	Clic en Enviar confirmación.	Enviar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('ENVIADO')
Botón Entregar	Clic en Entregar confirmación.	Entregar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('ENTREGADO')
Botón Cancelar	Clic en Cancelar confirmación.	Cancelar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('CANCELADO')

Tabla 32: Ingreso Revisión Despacho (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

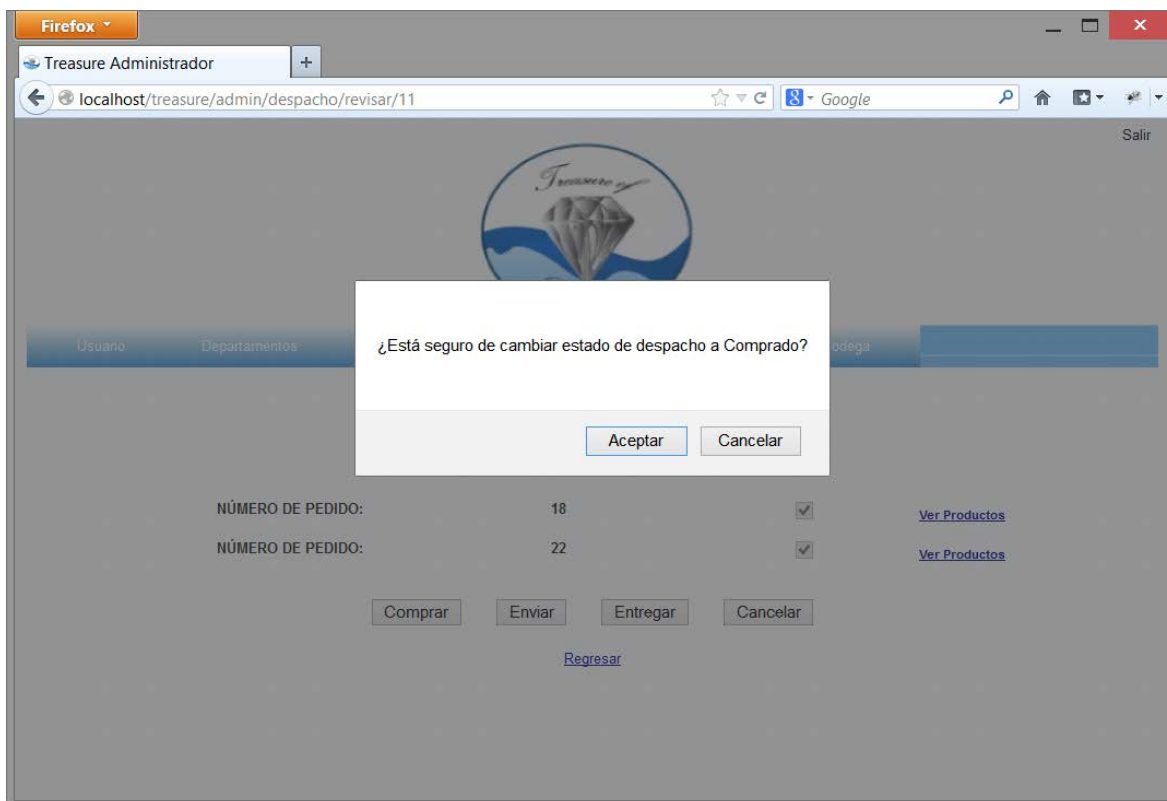


Ilustración 106: Revisión Despacho Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Comprar	Clic en Comprar confirmación.	Comprar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar
Botón Enviar	Clic en Enviar confirmación.	Enviar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar
Botón Entregar	Clic en Entregar confirmación.	Entregar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar
Botón Cancelar	Clic en Cancelar confirmación.	Cancelar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar

Tabla 33: Revisión Despacho Validaciones (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de manejo de bodega:

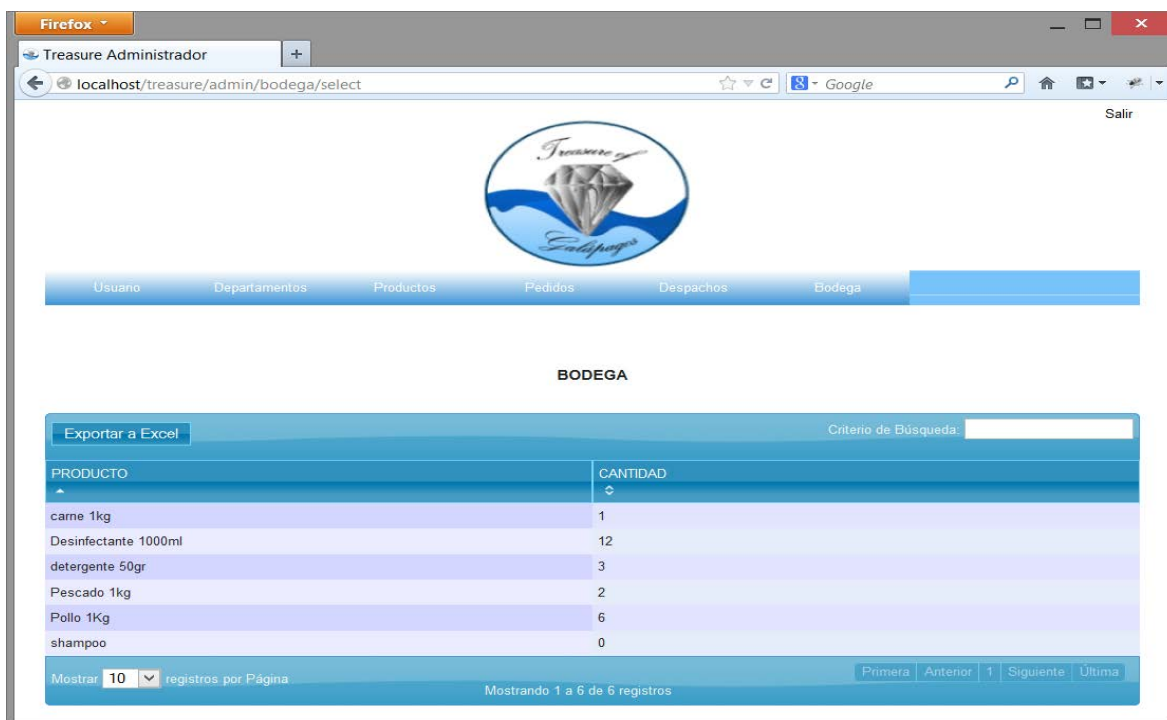


Ilustración 107: Bodega (Administrador). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los Productos que se encuentran en stock o en la bodega, con la posibilidad de cambiar su cantidad directamente.

Ingreso:

Tabla Producto X Bodega

	ID_PXB	ID_PRODUCTO	ID_BODEGA	CANTIDAD
<input type="checkbox"/> Editar Copiar Borrar	3	1	1	1
<input type="checkbox"/> Editar Copiar Borrar	4	2	1	12
<input type="checkbox"/> Editar Copiar Borrar	5	3	1	3
<input type="checkbox"/> Editar Copiar Borrar	6	4	1	0

Ilustración 108: Ingreso Bodega (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Campo Cantidad	Cantidad de producto: Máximo de 2 dígitos.	Cantidad de producto: Máximo de 2 dígitos. Campo vacío.	Registro almacenado en base de datos Campo Int(11)

Tabla 34: Ingreso Bodega (Administrador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

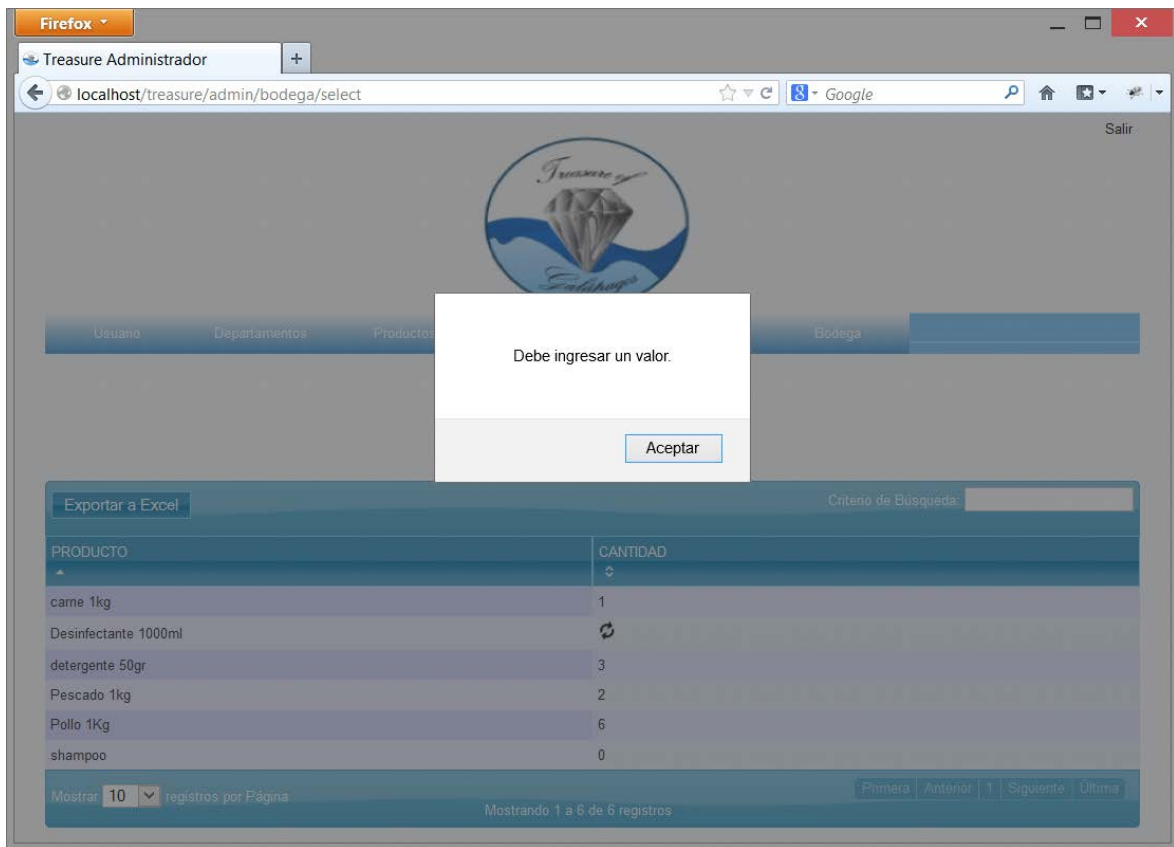


Ilustración 109: Bodega Validaciones (Administrador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Campo Cantidad	Cantidad de producto: Máximo de 2 dígitos.	Cantidad de producto: Máximo de 2 dígitos. Campo vacío.	Alerta de seguridad:

Tabla 35: Bodega Validaciones (Administrador). Autor: Alex Nieto.

5.3.1.3. *Solicitante:*

Ventana Principal:

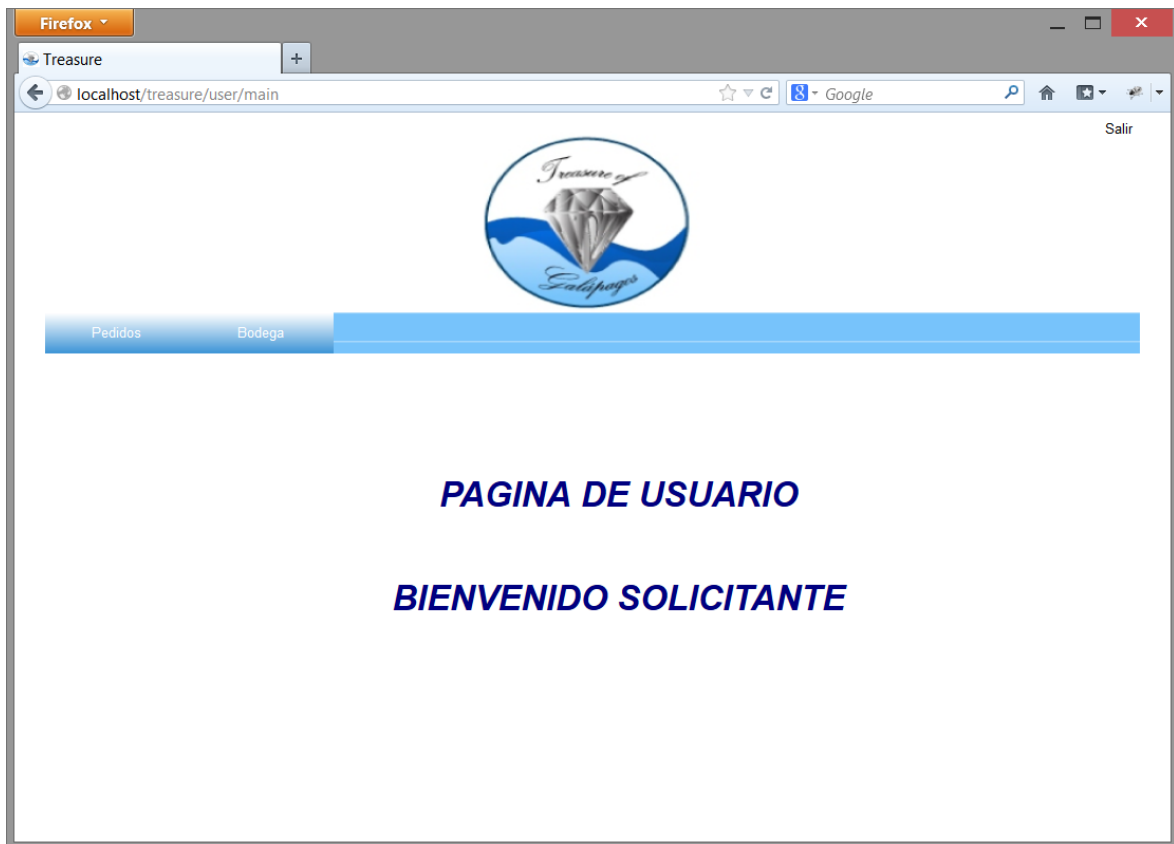


Ilustración 110: Ventana Principal (Solicitante). Autor: Alex Nieto.

El usuario que posee un perfil de solicitante se dirige a esta página en donde presenta un menú con todas las opciones disponibles para un perfil de solicitante de pedidos.

Ventana de nuevo Pedido:

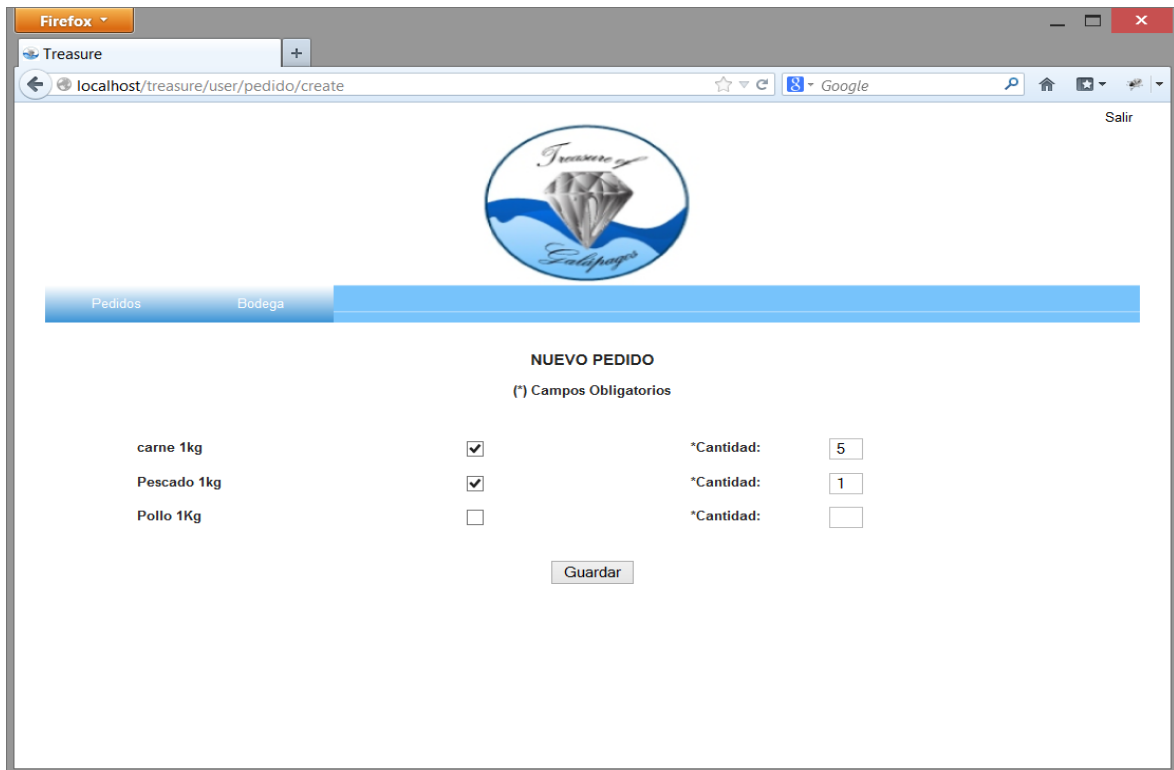


Ilustración 111: Nuevo Pedido (Solicitante). Autor: Alex Nieto.

En esta ventana se ingresa los productos para generar un nuevo pedido el mismo puede ser cambiado de “pedido” a “aprobado” o “cancelado”. (Los productos enlistados dependen de la persona que inició la sesión).

Ingreso:

Tabla pedidos

← T →				ID_PEDIDO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	10	1	5	3	2013	20	10
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	11	1	6	3	2013	14	20
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	12	1	8	3	2013	0	15
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	14	1	8	3	2013	0	17
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	15	1	8	3	2013	0	18
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	16	1	8	3	2013	0	43
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	17	1	8	3	2013	19	30
<input type="checkbox"/>	✎ Editar	📄 Copiar	🗑️ Borrar	18	1	11	3	2013	21	30

Ilustración 112: Ingreso 1 Nuevo Pedido (Solicitante). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

← T →				ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>				35	18	2	NULL	5	PEDIDO
<input type="checkbox"/>				36	18	6	NULL	1	PEDIDO

Ilustración 113: Ingreso 2 Nuevo Pedido (Solicitante). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Selección de productos: Los productos en lista.	Selección de productos: Sin elección de productos en lista.	Registro almacenado en base de datos Campo Int(11)
Cantidad	Cantidad de producto: Máximo 2 dígitos.	Cantidad de producto: + de 2 dígitos. - de 1 dígito.	Registro almacenado en base de datos Campo Int(11)

Tabla 36: Ingreso Nuevo Pedido (Solicitante). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

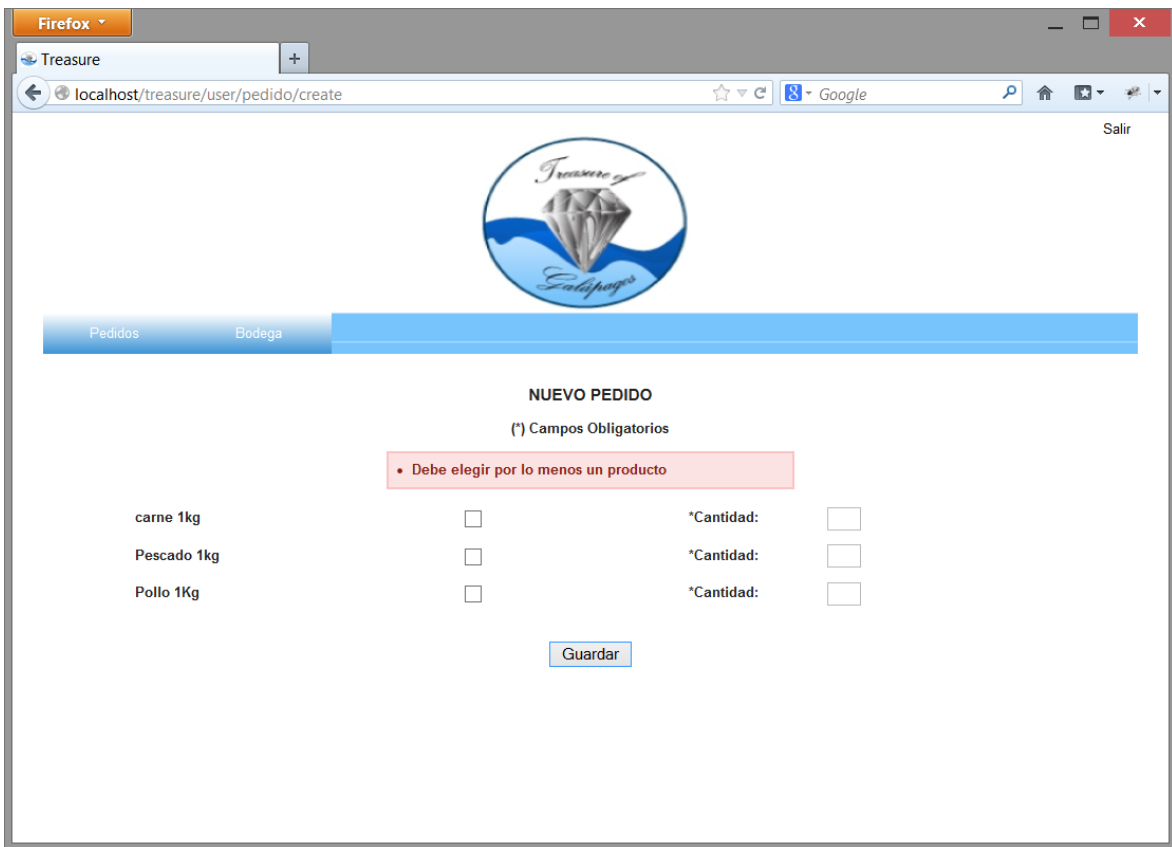


Ilustración 114: Nuevo Pedido Validaciones (Solicitante). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Selección de productos: Los productos en lista.	Selección de productos: Sin elección de productos en lista.	Mensaje con lo que falta de llenar
Cantidad	Cantidad de producto: Máximo 2 dígitos.	Cantidad de producto: + de 2 dígitos. - de 1 dígito.	Mensaje con lo que falta de llenar

Tabla 37: Nuevo Pedido Validaciones (Solicitante). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Pedidos:

Firefox

Treasure

localhost/treasure/user/pedido/select

Google

Salir

Treasure of Galapagos

Pedidos Bodega

SELECCIONAR PEDIDO PARA MODIFICAR

Crear Pedido

Exportar a Excel

Criterio de Búsqueda:

NÚMERO DE PEDIDO	USUARIO	ANIO	MES	DIA	EDITAR
25	solicitante solicitante	2013	3	14	

Mostrar 10 registros por Página

Mostrando 1 a 1 de 1 registros

Primera Anterior 1 Siguiente Última

Ilustración 115: Selección Pedido (Solicitante). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los pedidos existen en el sistema y a su vez la posibilidad de elegir alguno para su edición.

Ventana de edición de Pedidos:

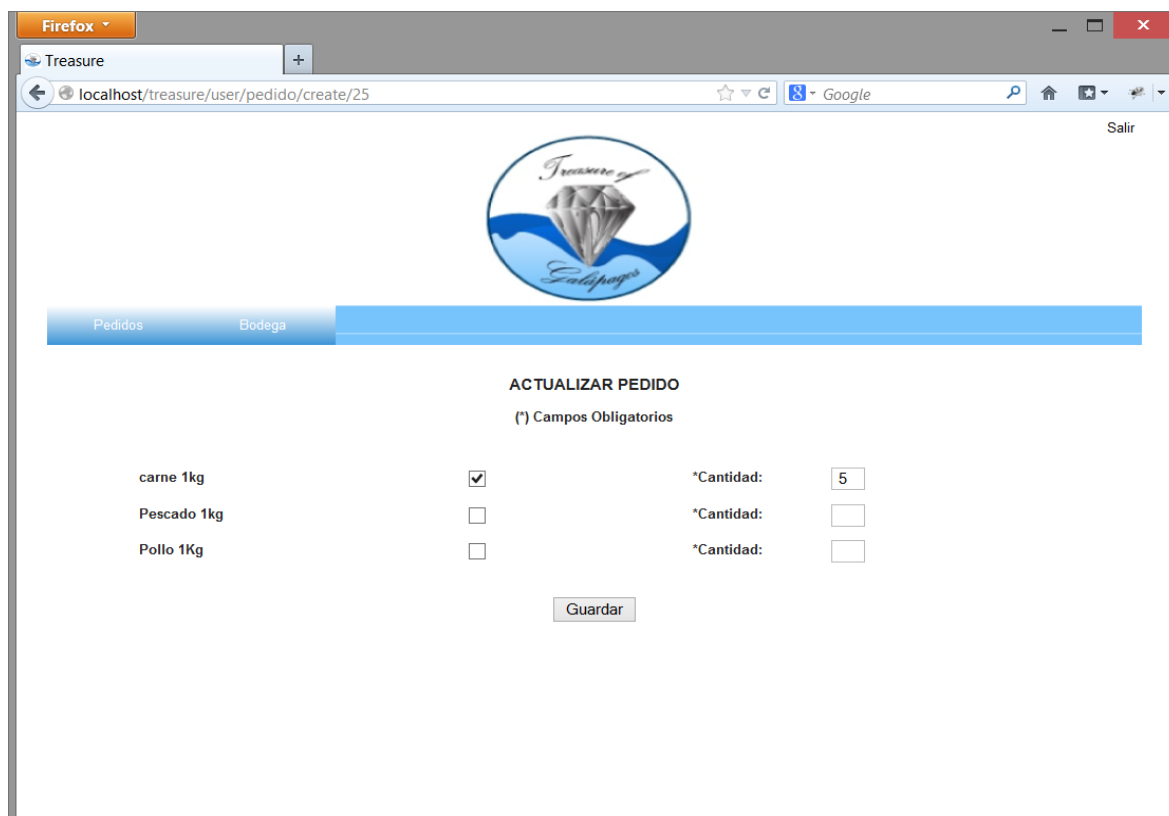


Ilustración 116: Edición Pedido (Solicitante). Autor: Alex Nieto.

Esta ventana muestra los datos del Pedido seleccionado en la ventana anterior la cual se edita o se cambia de productos y cantidad de los mismos para ingresarlos nuevamente en el sistema. (Los productos enlistados dependen se la persona que inició la sesión).

Ingreso:

Tabla pedidos

				ID_PEDIDO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO			
<input type="checkbox"/>		Editar		Copiar		Borrar	10	1	5	3	2013	20	10
<input type="checkbox"/>		Editar		Copiar		Borrar	11	1	6	3	2013	14	20
<input type="checkbox"/>		Editar		Copiar		Borrar	12	1	8	3	2013	0	15
<input type="checkbox"/>		Editar		Copiar		Borrar	14	1	8	3	2013	0	17
<input type="checkbox"/>		Editar		Copiar		Borrar	15	1	8	3	2013	0	18
<input type="checkbox"/>		Editar		Copiar		Borrar	16	1	8	3	2013	0	43
<input type="checkbox"/>		Editar		Copiar		Borrar	17	1	8	3	2013	19	30
<input type="checkbox"/>		Editar		Copiar		Borrar	18	1	11	3	2013	21	30

Ilustración 117: Ingreso 1 Edición Pedido (Solicitante). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

← T →		ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	Editar	35	18	2	NULL	5	PEDIDO

Ilustración 118: Ingreso 2 Edición Pedido (Solicitante). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Selección de productos: Los productos en lista.	Selección de productos: Sin elección de productos en lista.	Registro almacenado en base de datos Campo Int(11)
Cantidad	Cantidad de producto: Máximo 2 dígitos.	Cantidad de producto: + de 2 dígitos. - de 1 dígito.	Registro almacenado en base de datos Campo Int(11)

Tabla 38: Ingreso Edición Pedido (Solicitante). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

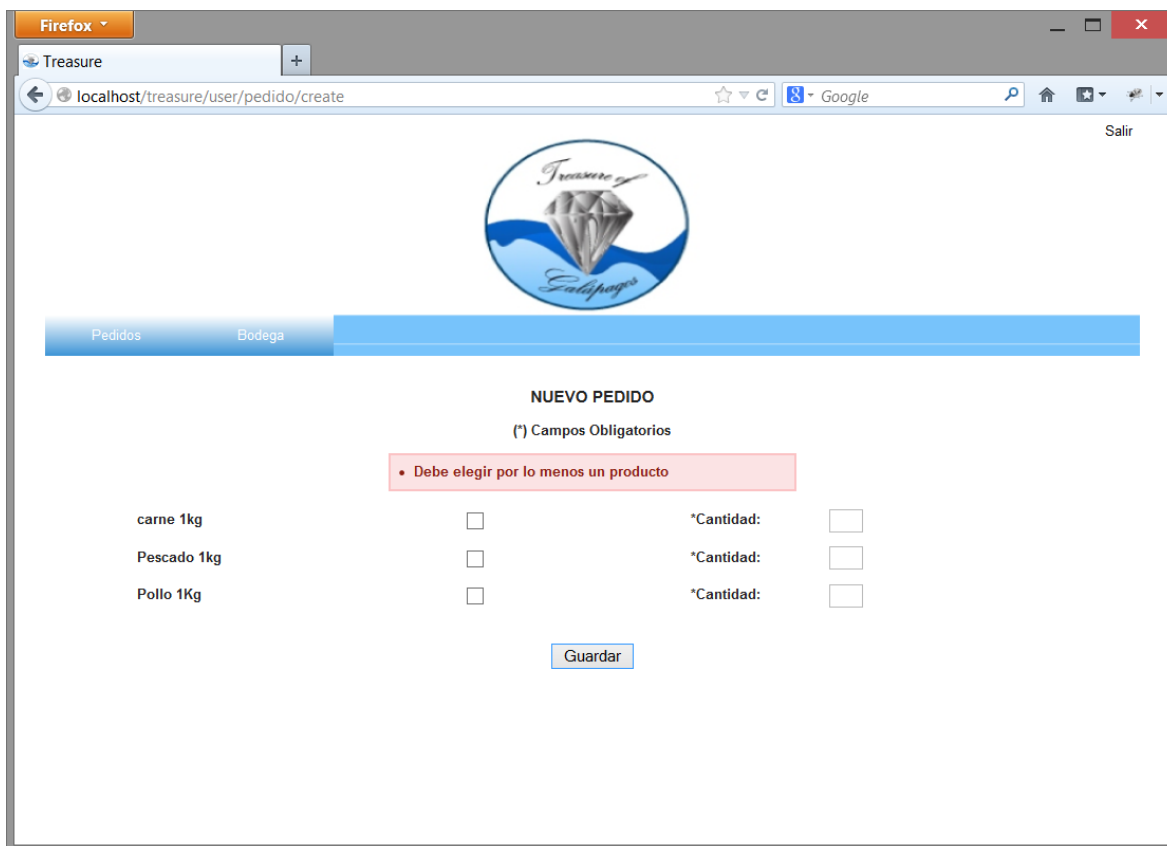


Ilustración 119: Edición Pedido Validaciones (Solicitante). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Producto	Selección de productos: Los productos en lista.	Selección de productos: Sin elección de productos en lista.	Mensaje con lo que falta de llenar
Cantidad	Cantidad de producto: Máximo 2 dígitos.	Cantidad de producto: + de 2 dígitos. - de 1 dígito.	Mensaje con lo que falta de llenar

Tabla 39: Edición Pedido Validaciones (Solicitante). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de Manejo de Bodega:

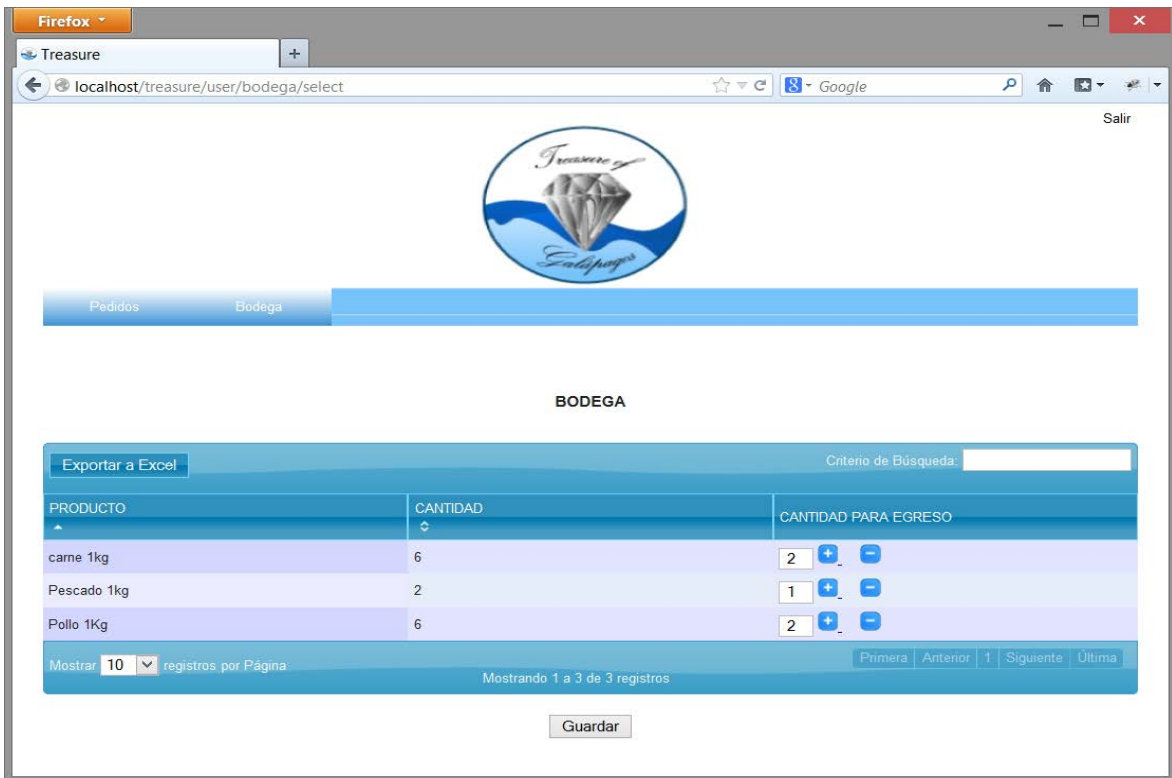


Ilustración 120: Bodega (Solicitante). Autor: Alex Nieto.

En esta ventana se ingresa a cantidad de productos que van a salir de bodega; pueden generarse solo egresos. (Los productos enlistados dependen se la persona que inició la sesión).

Ingreso:

Tabla producto x bodega

		ID_PXB	ID_PRODUCTO	ID_BODEGA	CANTIDAD
<input type="checkbox"/>	Editar Copiar Borrar	3	1	1	4
<input type="checkbox"/>	Editar Copiar Borrar	7	5	1	4
<input type="checkbox"/>	Editar Copiar Borrar	8	6	1	1

Ilustración 121: Ingreso Bodega (Solicitante). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Cantidad para Egreso	Cantidad para Egreso: Botón de más y de menos.	Cantidad para Egreso: Puede ingresar más cantidad de la disponible o cantidades negativas.	Registro almacenado en base de datos Campo Int(11)

Tabla 40: Ingreso Bodega (Solicitante). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

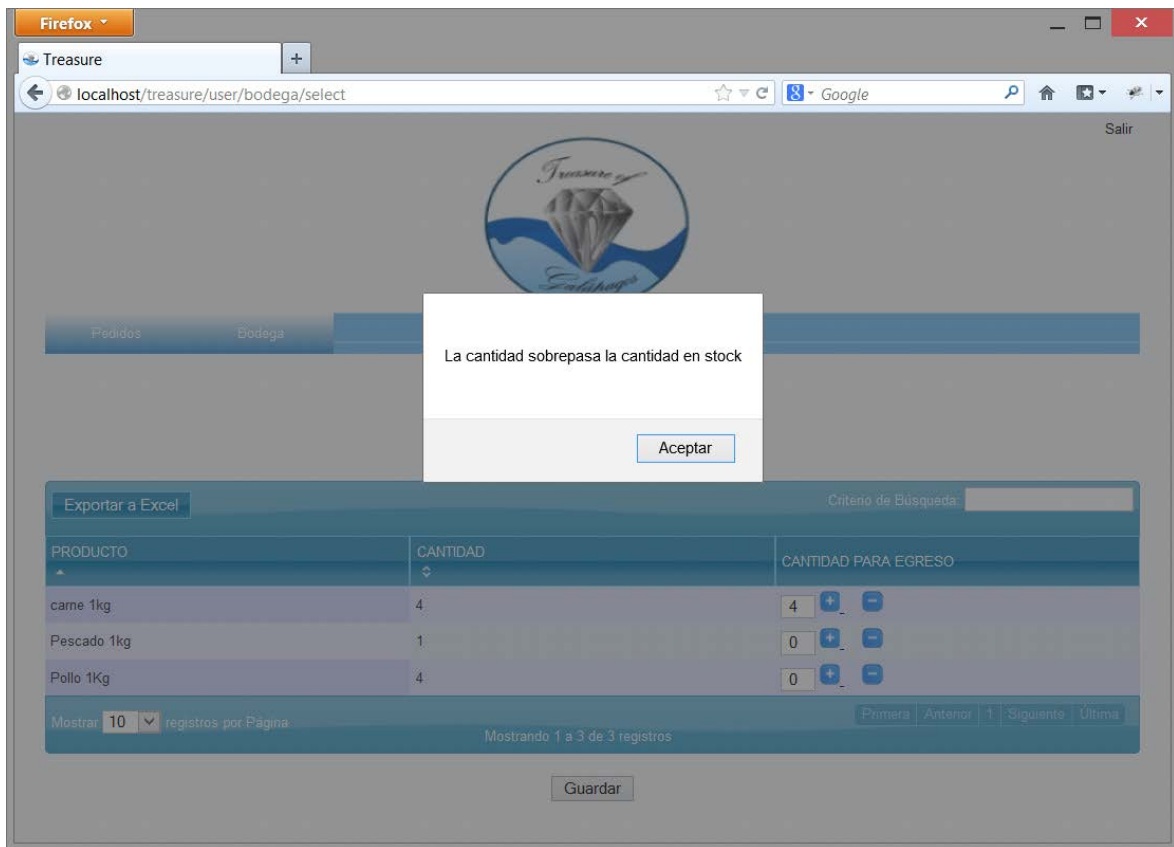


Ilustración 122: Bodega Validaciones (Solicitante). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Cantidad para Egreso	Cantidad para Egreso: Botón de más y de menos.	Cantidad para Egreso: Puede ingresar más cantidad de la disponible o cantidades negativas.	Alerta de seguridad.

Tabla 41: Bodega Validaciones (Solicitante). Autor: Alex Nieto.

5.3.1.4. *Aprobador:*

Ventana Principal:



Ilustración 123: Ventana Principal (Aprobador). Autor: Alex Nieto.

El usuario que posee un perfil de aprobador se dirige a esta página en donde se presenta un menú con todas las opciones disponibles para un perfil de aprobador de pedidos.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Pedidos para Aprobación:

Firefox

Treasure Administrador

localhost/treasure/manager/pedido/aprobar

Salir

Pedidos

SELECCIONAR PEDIDO PARA APROBACIÓN

Exportar a Excel

Criterio de Búsqueda:

NÚMERO DE PEDIDO	USUARIO	AÑO	MES	DÍA	REVISAR
18	administradoruno administradoruno	2013	3	11	
25	solicitante solicitante	2013	3	14	
26	solicitantedos solicitantedos	2013	3	15	

Mostrar 10 registros por Página

Mostrando 1 a 3 de 3 registros

Primera Anterior 1 Siguiente Última

Ilustración 124: Aprobación Pedido (Aprobador). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los pedidos que se encuentran en estado de "Pedido" pero aún no están aprobados en el sistema y a su vez la posibilidad de elegir alguno para su aprobación o cancelación.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de revisión de Pedidos:

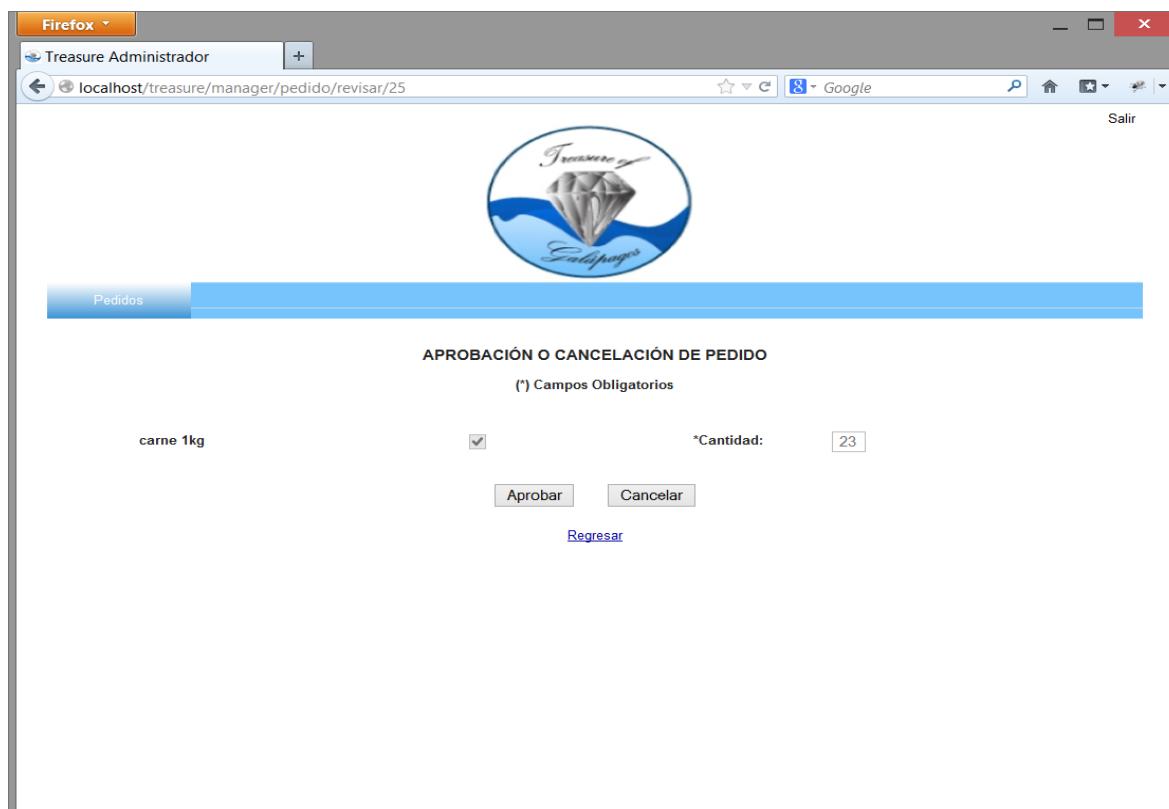


Ilustración 125: Revisión Pedido (Aprobador). Autor: Alex Nieto.

Esta ventana muestra los datos del pedido seleccionado en la ventana anterior la cual puede “Aprobar” o “Cancelar” el pedido seleccionado.

Ingreso:

Tabla pedidos

				ID_PEDIDO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO			
<input type="checkbox"/>		Editar		Copiar		Borrar	10	1	5	3	2013	20	10
<input type="checkbox"/>		Editar		Copiar		Borrar	11	1	6	3	2013	14	20
<input type="checkbox"/>		Editar		Copiar		Borrar	12	1	8	3	2013	0	15
<input type="checkbox"/>		Editar		Copiar		Borrar	14	1	8	3	2013	0	17
<input type="checkbox"/>		Editar		Copiar		Borrar	15	1	8	3	2013	0	18
<input type="checkbox"/>		Editar		Copiar		Borrar	16	1	8	3	2013	0	43
<input type="checkbox"/>		Editar		Copiar		Borrar	17	1	8	3	2013	19	30
<input type="checkbox"/>		Editar		Copiar		Borrar	18	1	11	3	2013	21	30

Ilustración 126: Ingreso 1 Revisión Pedido (Aprobador). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

		ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	Editar Copiar Borrar	35	18	2	NULL	5	APROBADO
<input type="checkbox"/>	Editar Copiar Borrar	36	18	6	NULL	1	APROBADO

Ilustración 127: Ingreso 2 Revisión Pedido (Aprobador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Aprobar	Clic en aceptar confirmación.	Aprobar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('APROBADO')
Botón Cancelar	Clic en aceptar confirmación.	Cancelar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('CANCELADO')

Tabla 42: Ingreso Revisión Pedido (Aprobador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

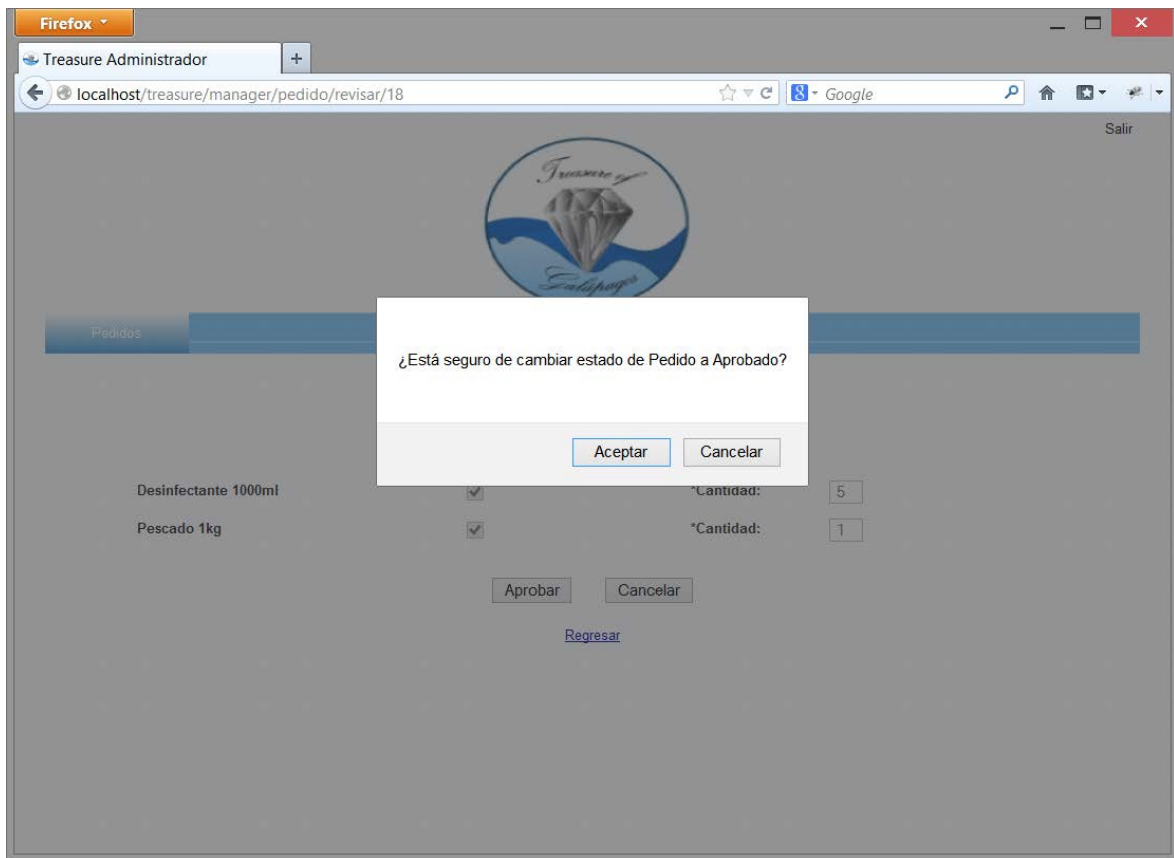


Ilustración 128: Revisión Pedido Validaciones (Aprobador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Aprobar	Clic en aceptar confirmación.	Aprobar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar
Botón Cancelar	Clic en aceptar confirmación.	Cancelar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar

Tabla 43: Revisión Pedido Validaciones (Aprobador). Autor: Alex Nieto.

5.3.1.5. *Proveedor:*

Ventana Principal:

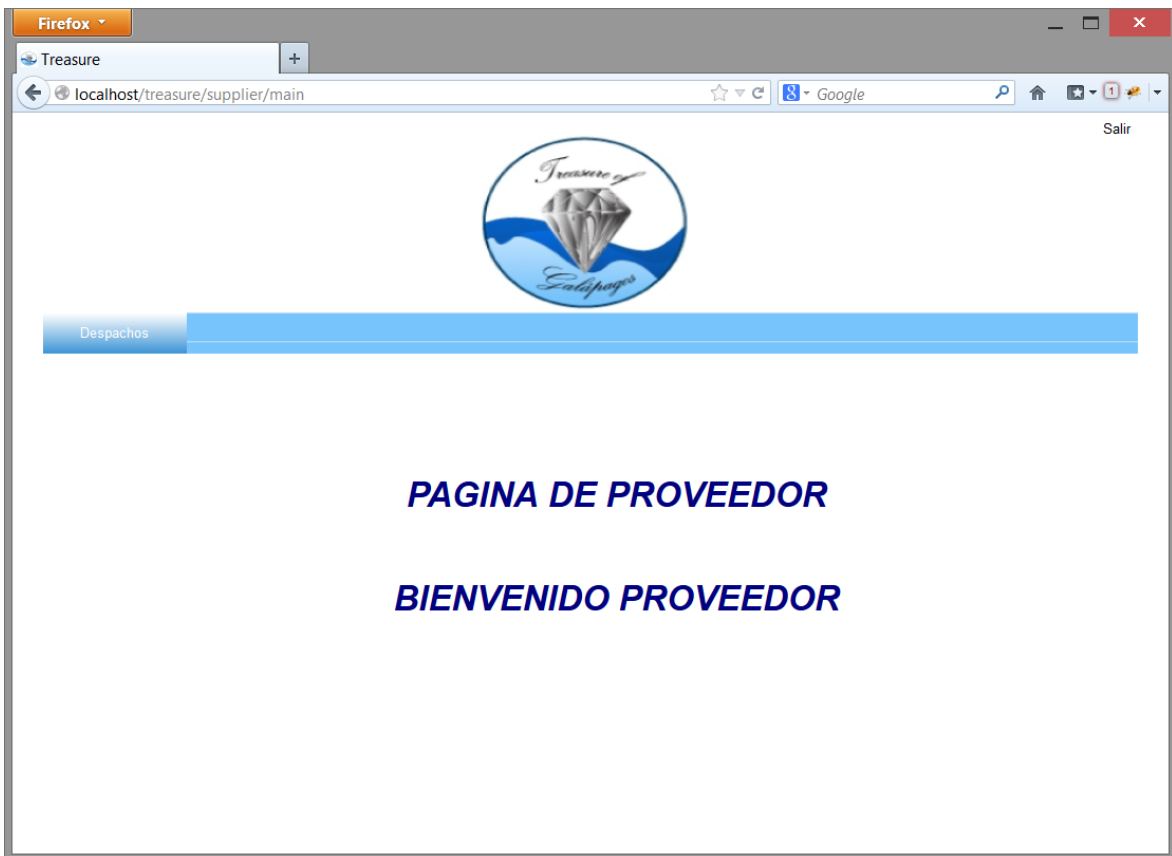


Ilustración 129: Ventana Principal (Proveedor). Autor: Alex Nieto.

El usuario que posee un perfil de proveedor se dirige a esta página en donde presenta un menú con todas las opciones disponibles para un perfil de proveedor.

Ventana de nuevo Despacho:

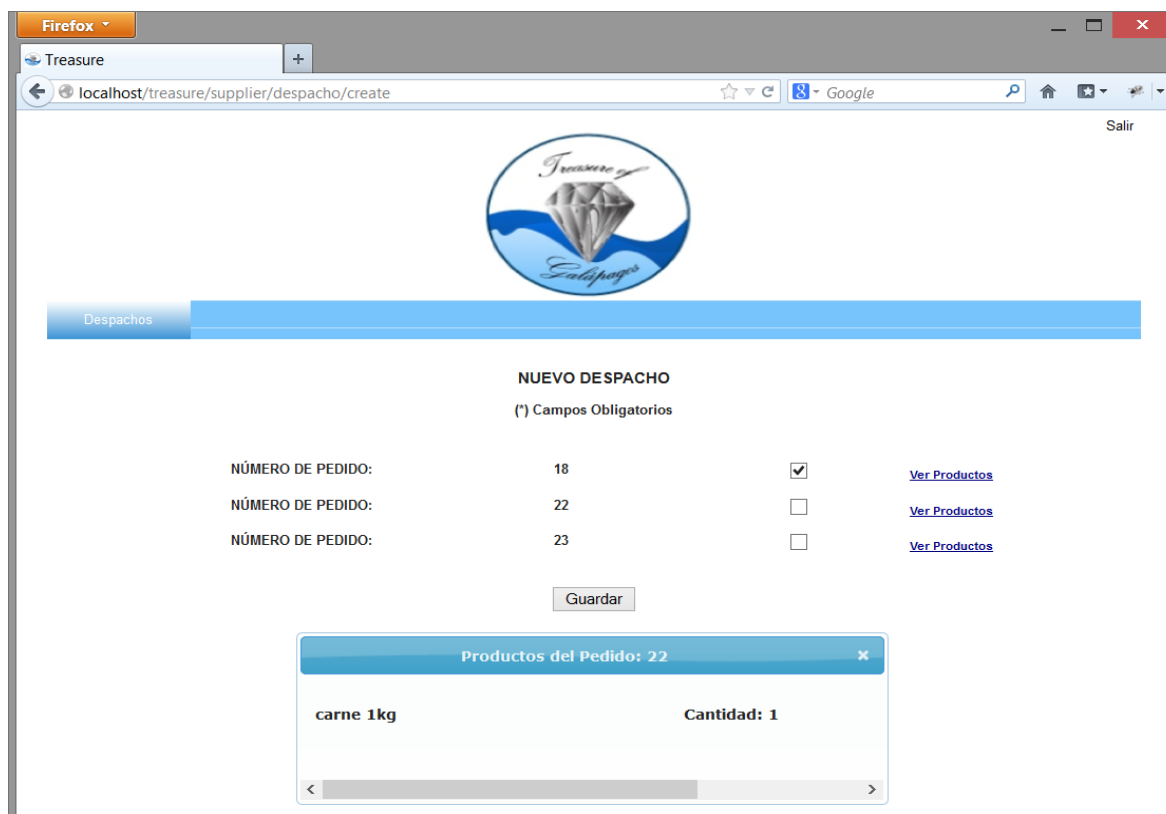


Ilustración 130: Nuevo Despacho (Proveedor). Autor: Alex Nieto.

En esta ventana se selecciona los pedidos “Aprobados” con la opción de revisar los productos que contiene cada pedido.

Ingreso:

Tabla despachos

				ID_DESPACHO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO			
<input type="checkbox"/>		Editar		Copiar		Borrar	1	1	7	3	2013	18	22
<input type="checkbox"/>		Editar		Copiar		Borrar	3	1	8	3	2013	0	37
<input type="checkbox"/>		Editar		Copiar		Borrar	4	1	8	3	2013	19	33
<input type="checkbox"/>		Editar		Copiar		Borrar	6	1	9	3	2013	1	28
<input type="checkbox"/>		Editar		Copiar		Borrar	7	1	9	3	2013	1	31
<input type="checkbox"/>		Editar		Copiar		Borrar	8	1	9	3	2013	1	35
<input type="checkbox"/>		Editar		Copiar		Borrar	9	1	9	3	2013	1	42
<input type="checkbox"/>		Editar		Copiar		Borrar	10	1	11	3	2013	21	34
<input type="checkbox"/>		Editar		Copiar		Borrar	11	1	22	3	2013	0	17

Ilustración 131: Ingreso 1 Nuevo Despacho (Proveedor). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto










← T →				ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	35	18	2	11	5	EN PROCESO
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	36	18	6	11	1	EN PROCESO
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	38	22	1	11	1	EN PROCESO

Ilustración 132: Ingreso 2 Nuevo Despacho (Proveedor). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Pedido	Selección de pedidos: Los pedidos en lista.	Selección de pedidos: Sin elección de productos en lista.	Registro almacenado en base de datos Campo Int(11)

Tabla 44: Ingreso Nuevo Despacho (Proveedor). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

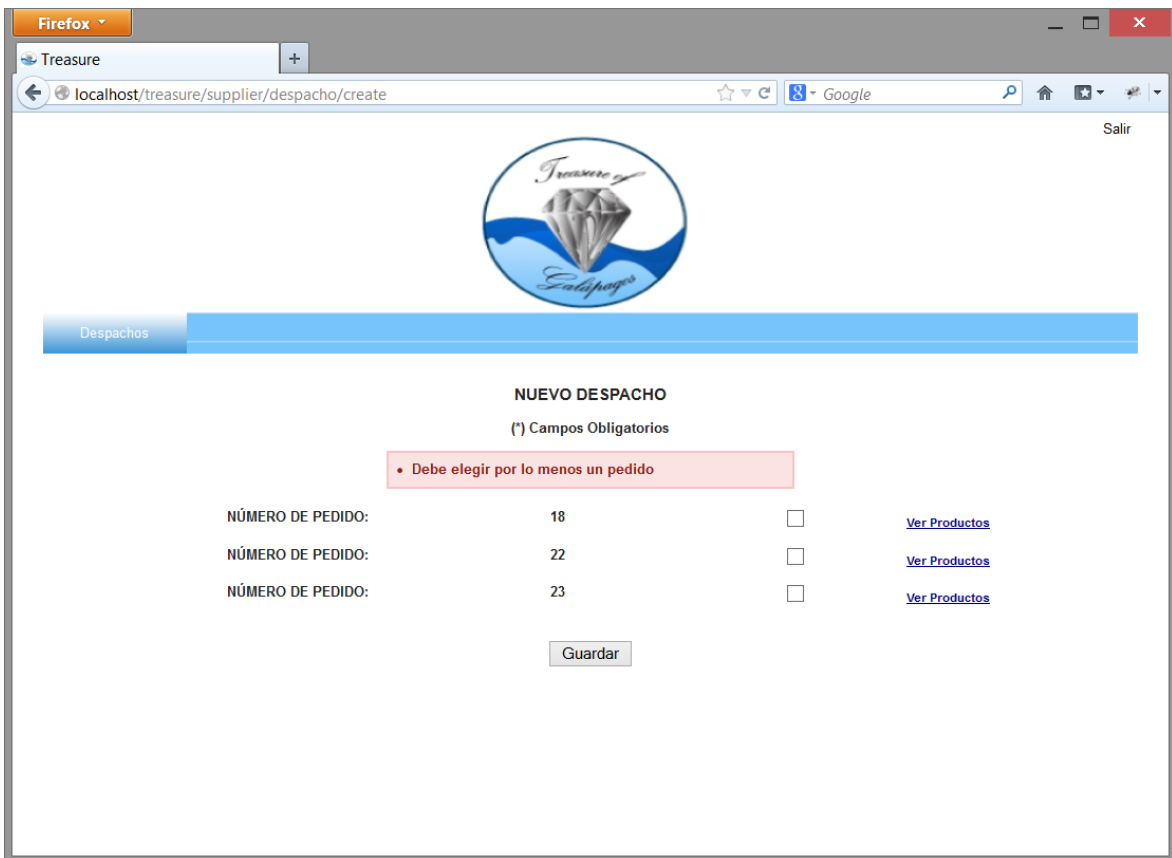


Ilustración 133: Nuevo Despacho Validaciones (Proveedor). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Pedido	Selección de pedidos: Los pedidos en lista.	Selección de pedidos: Sin elección de productos en lista.	Mensaje con lo que falta de llenar

Tabla 45: Nuevo Despacho Validaciones (Proveedor). Autor: Alex Nieto.

Ventana de selección de Despachos:

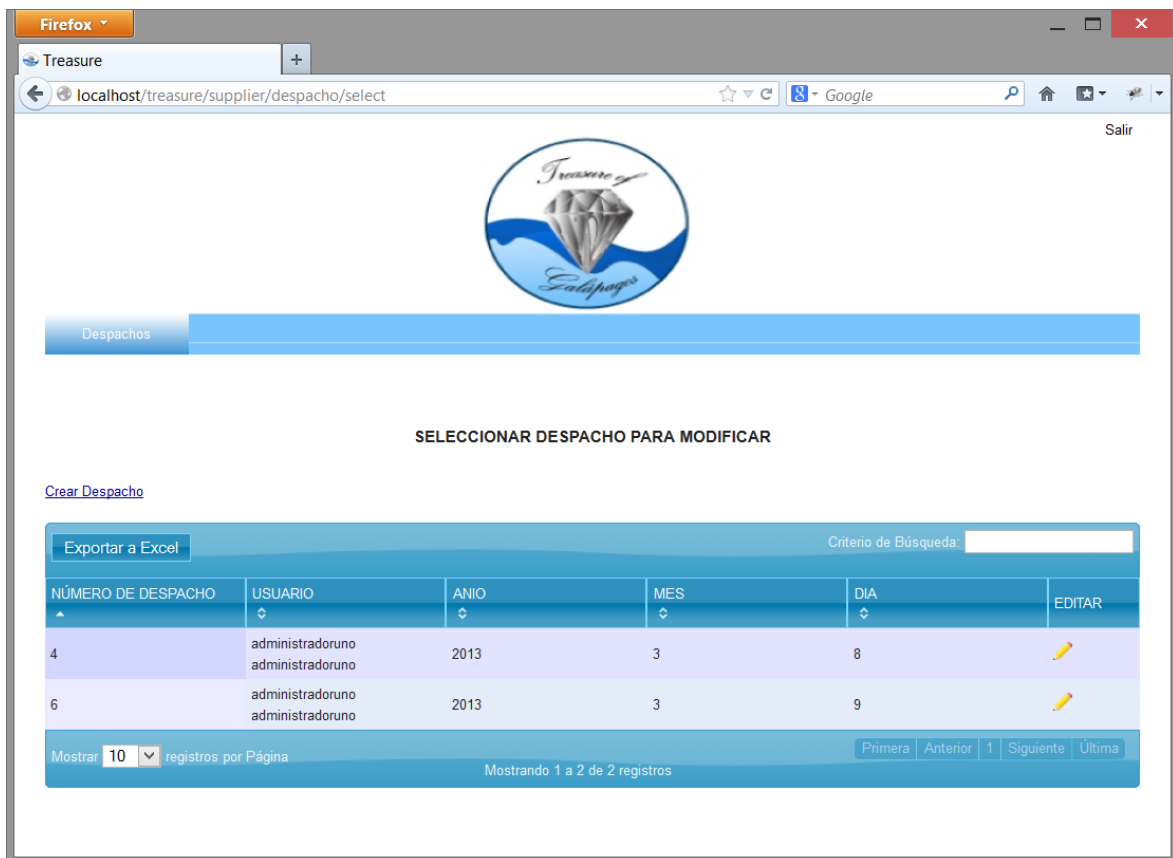


Ilustración 134: Selección Despacho (Proveedor). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los despachos que existen en estado “EN PROCESO” en el sistema y a su vez la posibilidad de elegir alguno para su edición. (Se visualizan solo los despachos creados por el usuario que los generó).

Ventana de edición de Despachos:

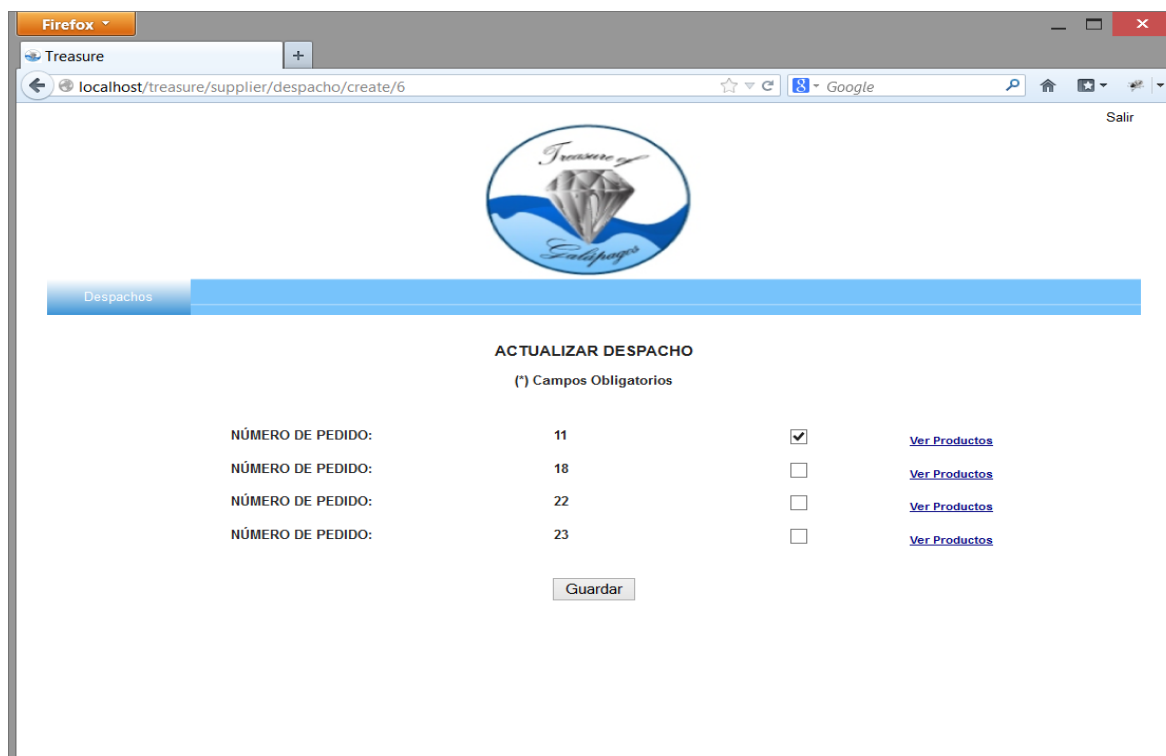


Ilustración 135: Edición Despacho (Proveedor). Autor: Alex Nieto.

Esta ventana muestra los datos del despacho seleccionado en la ventana anterior, la cual edita o cambia los pedidos seleccionados para ingreso en el sistema.

Ingreso:

Tabla despachos

		ID_DESPACHO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO
<input type="checkbox"/>	Editar Copiar Borrar	1	1	7	3	2013	18	22
<input type="checkbox"/>	Editar Copiar Borrar	3	1	8	3	2013	0	37
<input type="checkbox"/>	Editar Copiar Borrar	4	1	8	3	2013	19	33
<input type="checkbox"/>	Editar Copiar Borrar	6	1	9	3	2013	1	28
<input type="checkbox"/>	Editar Copiar Borrar	7	1	9	3	2013	1	31
<input type="checkbox"/>	Editar Copiar Borrar	8	1	9	3	2013	1	35
<input type="checkbox"/>	Editar Copiar Borrar	9	1	9	3	2013	1	42
<input type="checkbox"/>	Editar Copiar Borrar	10	1	11	3	2013	21	34
<input type="checkbox"/>	Editar Copiar Borrar	11	1	22	3	2013	0	17

Ilustración 136: Ingreso 1 Edición Despacho (Proveedor). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

← T →				ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	Editar	Copiar	Borrar	35	18	2	11	5	EN PROCESO
<input type="checkbox"/>	Editar	Copiar	Borrar	36	18	6	11	1	EN PROCESO
<input type="checkbox"/>	Editar	Copiar	Borrar	38	22	1	11	1	EN PROCESO

Ilustración 137: Ingreso 2 Edición Despacho (Proveedor). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Pedido	Selección de pedidos: Los pedidos en lista.	Selección de pedidos: Sin elección de productos en lista.	Registro almacenado en base de datos Campo Int(11)

Tabla 46: Ingreso Edición Despacho (Proveedor). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

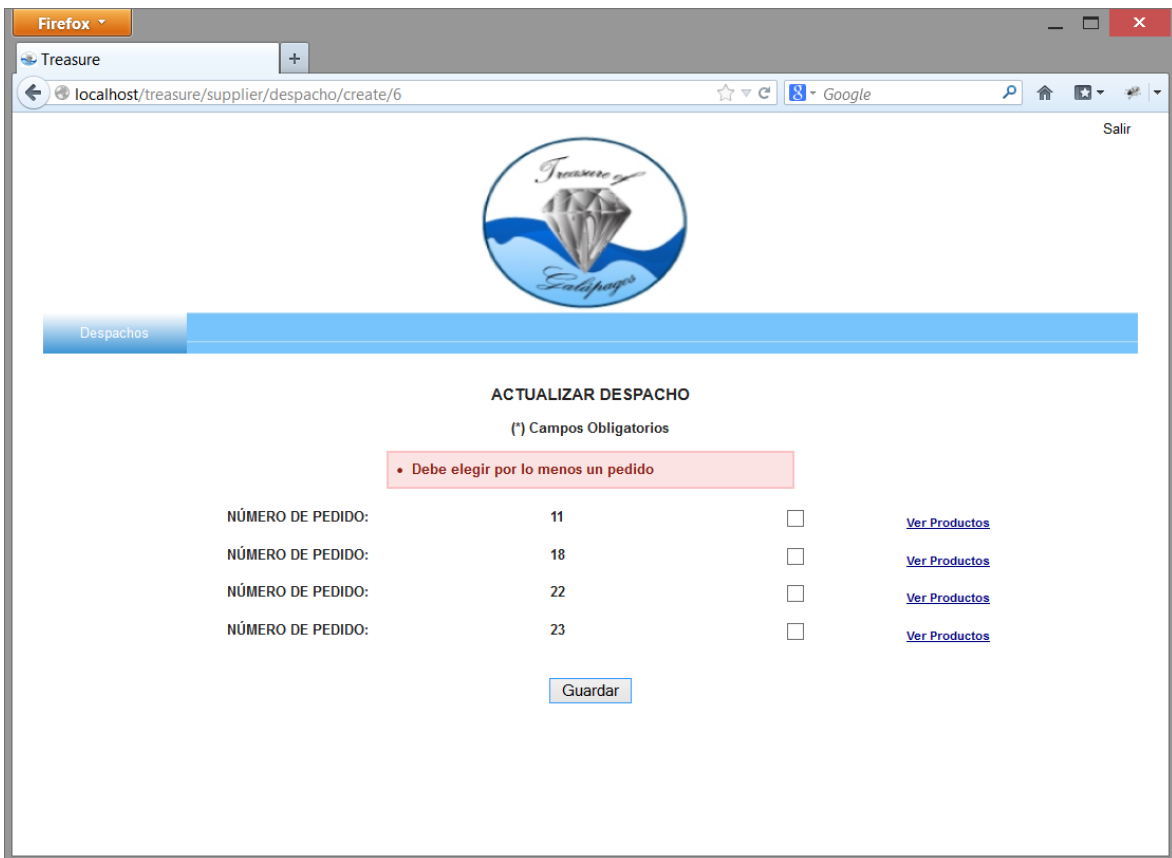


Ilustración 138: Edición Despacho Validaciones (Proveedor). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Pedido	Selección de pedidos: Los pedidos en lista.	Selección de pedidos: Sin elección de productos en lista.	Mensaje con lo que falta de llenar

Tabla 47: Edición Despacho Validaciones (Proveedor). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Despachos para Cambio de Estado:

Firefox

Treasure

localhost/treasure/supplier/despacho/procesar

Salir

Despachos

SELECCIONAR DESPACHO PARA PROCESAR

[Crear Despacho](#)

Exportar a Excel

Criterio de Búsqueda:

NÚMERO DE DESPACHO	USUARIO	ANIO	MES	DIA	REVISAR
3	administradoruno administradoruno	2013	3	8	
4	administradoruno administradoruno	2013	3	8	
6	administradoruno administradoruno	2013	3	9	
10	administradoruno administradoruno	2013	3	11	

Mostrar 10 registros por Página

Mostrando 1 a 4 de 4 registros

Primera Anterior 1 Siguiente Última

Ilustración 139: Estado Despacho (Proveedor). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los despachos que se encuentran en estado de “En Proceso”, con la posibilidad de cambiar su estado.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de revisión de Despacho:

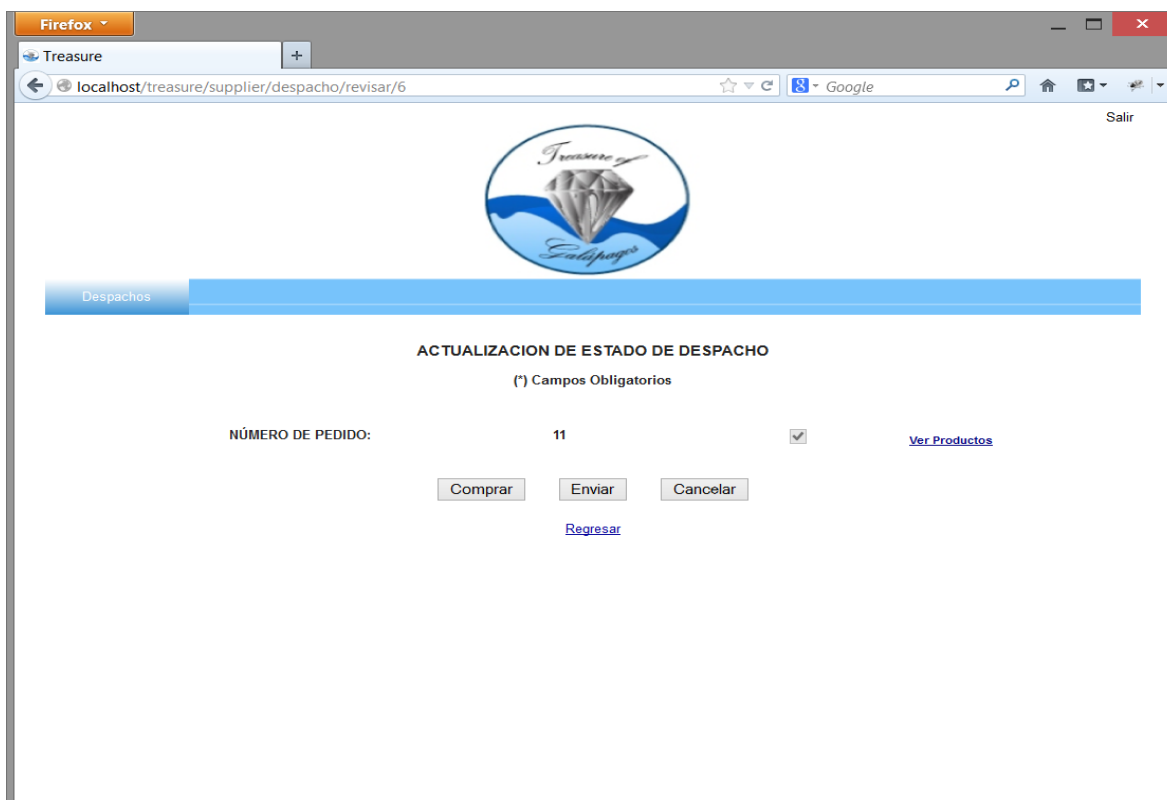


Ilustración 140: Revisión Despacho (Proveedor). Autor: Alex Nieto.

Esta ventana muestra los datos del Despacho seleccionado en la ventana anterior, la cual puede cambiar su estado a: “Comprar”, “Enviar” o “Cancelar”.

Ingreso:

Tabla Despachos

				ID_DESPACHO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO
<input type="checkbox"/>	Editar	Copiar	Borrar	1	1	7	3	2013	18	22
<input type="checkbox"/>	Editar	Copiar	Borrar	3	1	8	3	2013	0	37
<input type="checkbox"/>	Editar	Copiar	Borrar	4	1	8	3	2013	19	33
<input type="checkbox"/>	Editar	Copiar	Borrar	6	1	9	3	2013	1	28
<input type="checkbox"/>	Editar	Copiar	Borrar	7	1	9	3	2013	1	31
<input type="checkbox"/>	Editar	Copiar	Borrar	8	1	9	3	2013	1	35
<input type="checkbox"/>	Editar	Copiar	Borrar	9	1	9	3	2013	1	42
<input type="checkbox"/>	Editar	Copiar	Borrar	10	1	11	3	2013	21	34
<input type="checkbox"/>	Editar	Copiar	Borrar	11	1	22	3	2013	0	17

Ilustración 141: Ingreso 1 Revisión Despacho (Proveedor). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

		ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	Editar Copiar Borrar	35	18	2	11	5	COMPRADO
<input type="checkbox"/>	Editar Copiar Borrar	36	18	6	11	1	COMPRADO
<input type="checkbox"/>	Editar Copiar Borrar	38	22	1	11	1	COMPRADO

Ilustración 142: Ingreso 2 Revisión Despacho (Proveedor). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Comprar	Clic en Comprar confirmación.	Comprar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('COMPRADO')
Botón Enviar	Clic en Enviar confirmación.	Enviar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('ENVIADO')
Botón Entregar	Clic en Entregar confirmación.	Entregar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('ENTREGADO')
Botón Cancelar	Clic en Cancelar confirmación.	Cancelar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('CANCELADO')

Tabla 48: Ingreso Revisión Despacho (Proveedor). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

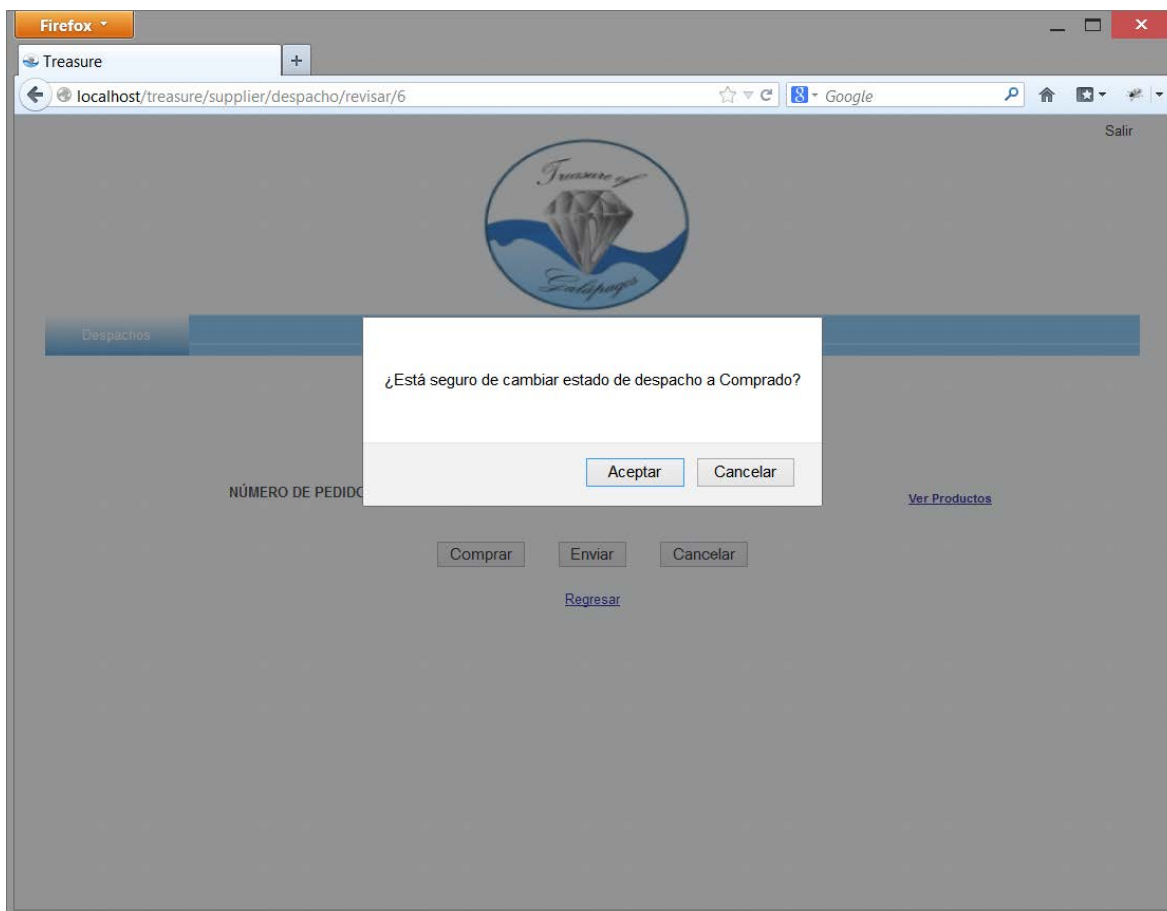


Ilustración 143: Revisión Despacho Validaciones (Proveedor). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Comprar	Clic en Comprar confirmación.	Comprar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar
Botón Enviar	Clic en Enviar confirmación.	Enviar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar
Botón Cancelar	Clic en Cancelar confirmación.	Cancelar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar

Tabla 49: Revisión Despacho Validaciones (Proveedor). Autor: Alex Nieto.

5.3.1.6. *Controlador:*

Ventana Principal:



Ilustración 144: Ventana Principal (Controlador). Autor: Alex Nieto.

El usuario que posee un perfil de controlador se dirige a esta página en donde presenta un menú con todas las opciones disponibles para un perfil de controlador.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de selección de Despachos para Cambio de Estado:

Firefox

Treasure

localhost/treasure/control/despacho/procesar

Salir

Despachos

SELECCIONAR DESPACHO PARA PROCESAR

[Crear Despacho](#)

Exportar a Excel

Criterio de Búsqueda:

NÚMERO DE DESPACHO	USUARIO	ANIO	MES	DIA	REVISAR
3	administradoruno administradoruno	2013	3	8	
4	administradoruno administradoruno	2013	3	8	
6	administradoruno administradoruno	2013	3	9	
10	administradoruno administradoruno	2013	3	11	

Mostrar 10 registros por Página

Mostrando 1 a 4 de 4 registros

Primera Anterior 1 Siguiente Última

Ilustración 145: Estado Despacho (Controlador). Autor: Alex Nieto.

Esta ventana cumple dos funciones: la de generar un reporte de los despachos que se encuentran en estado de “Enviado”, con la posibilidad de cambiar su estado.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Ventana de revisión de Despacho:

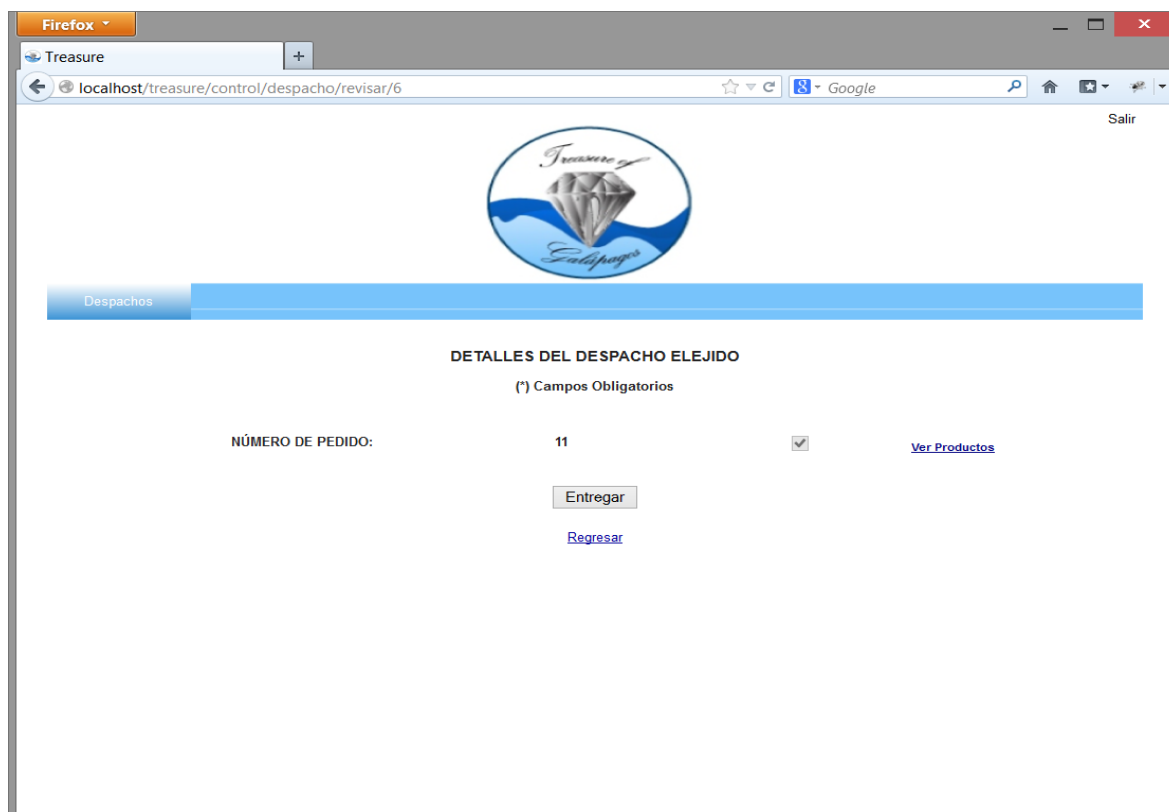


Ilustración 146: Revisión Despacho (Controlador). Autor: Alex Nieto.

Esta ventana muestra los datos del Despacho seleccionado en la ventana anterior la cual puede cambiar su estado a: “Comprar”, “Enviar” o “Cancelar”.

Ingreso:

Tabla Despachos

		ID_DESPACHO	ID_USUARIO	DIA	MES	ANIO	HORA	MINUTO
<input type="checkbox"/>	Editar Copiar Borrar	1	1	7	3	2013	18	22
<input type="checkbox"/>	Editar Copiar Borrar	3	1	8	3	2013	0	37
<input type="checkbox"/>	Editar Copiar Borrar	4	1	8	3	2013	19	33
<input type="checkbox"/>	Editar Copiar Borrar	6	1	9	3	2013	1	28
<input type="checkbox"/>	Editar Copiar Borrar	7	1	9	3	2013	1	31
<input type="checkbox"/>	Editar Copiar Borrar	8	1	9	3	2013	1	35
<input type="checkbox"/>	Editar Copiar Borrar	9	1	9	3	2013	1	42
<input type="checkbox"/>	Editar Copiar Borrar	10	1	11	3	2013	21	34
<input type="checkbox"/>	Editar Copiar Borrar	11	1	22	3	2013	0	17

Ilustración 147: Ingreso 1 Despacho (Controlador). Autor: Alex Nieto.

Tabla Pedido Despacho X Producto

←T→		ID_PDXP	ID_PEDIDO	ID_PRODUCTO	ID_DESPACHO	CANTIDAD	ESTADO
<input type="checkbox"/>	Editar Copiar Borrar	35	18	2	11	5	COMPRADO
<input type="checkbox"/>	Editar Copiar Borrar	36	18	6	11	1	COMPRADO
<input type="checkbox"/>	Editar Copiar Borrar	38	22	1	11	1	COMPRADO

Ilustración 148: Ingreso 2 Revisión Despacho (Controlador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Entregar	Clic en Entregar confirmación.	Entregar pedido: Sin errores truncado con Alert	Registro almacenado en base de datos Campo enum('ENTREGADO')

Tabla 50: Ingreso Revisión Despacho (Controlador). Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Pruebas validaciones de usuario.

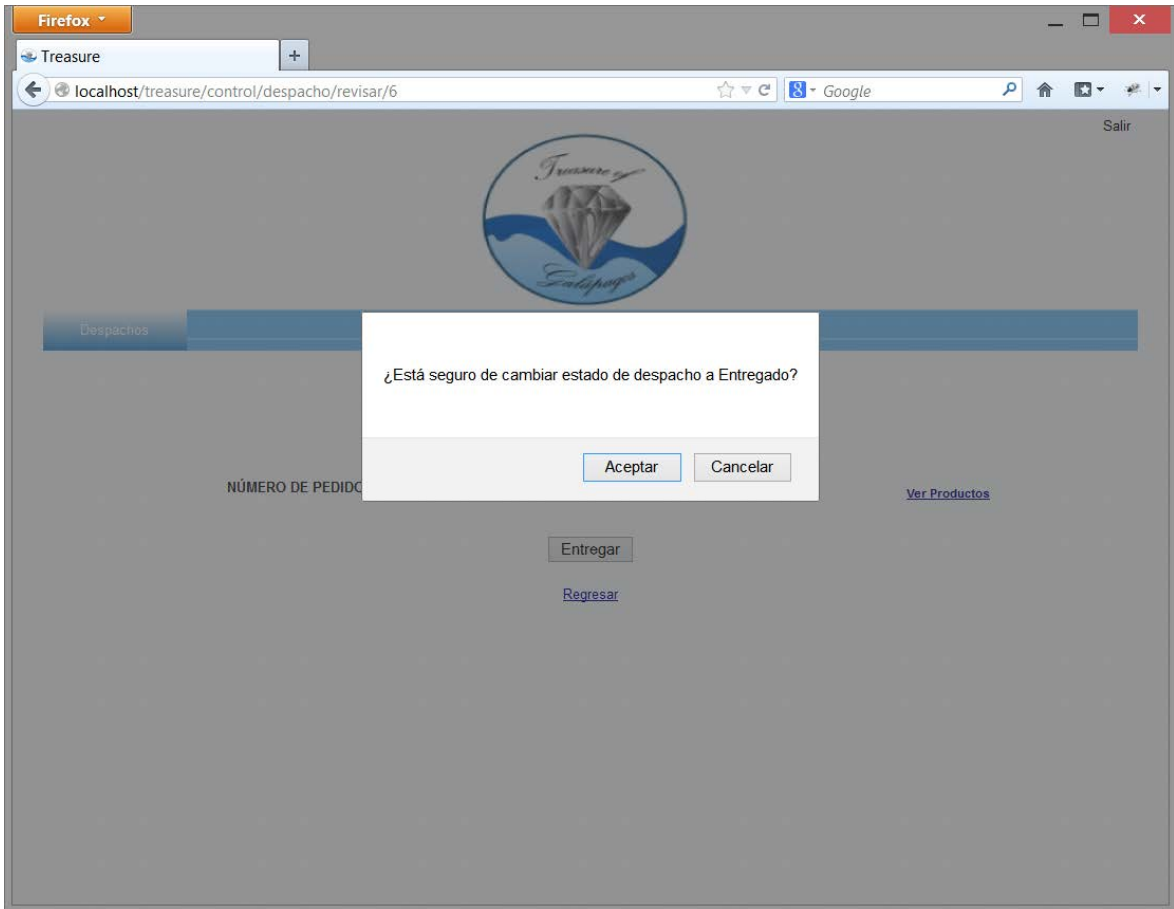


Ilustración 149: Revisión Despacho Validaciones (Controlador). Autor: Alex Nieto.

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Botón Entregar	Clic en Entregar confirmación.	Entregar pedido: Sin errores truncado con Alert	Alerta de seguridad: Confirmar o cancelar

Tabla 51: Revisión Despacho Validaciones (Controlador). Autor: Alex Nieto.

5.4. Transición.

5.4.1. Seguridad.

El sistema aparte de poseer seguridades en las ventanas para que únicamente usuarios autorizados ingresen, también cuenta con la seguridad de datos por medio de la configuración de seguridad que brinda MySQL, el cual, aparte de tener los datos protegidos con contraseña también puede ser escalable al punto de generar log de las interacciones que produce cada usuario.

5.4.2. Costos.

Los costos de acuerdo al proyecto desarrollado, teniendo en cuenta que son costos actualizados al mes de marzo de 2013, son los siguientes:

Tabla de costos iniciales del proyecto	
Costos del Hardware	\$ -
Costos del Software	\$ 1.500,00
Costos de Instalación	\$ 350,00
Costos de Configuración	\$ 350,00
Total costos iniciales	\$ 2.200,00

Tabla 52: Costos Iniciales. Autor: Alex Nieto.

Costos de Administración	
Costo promedio anual de un Ingeniero de Sistemas	\$ 14.400,00
Número de Ingenieros	1
Porcentaje del tiempo dedicado a la administración de la solución	30%
Costo Promedio Por Año	\$ 4.320,00
Número de años de la solución	1
Costo estimado en el tiempo de vida de la solución	\$ 1.080,00

Tabla 53: Costos Administrativos. Autor: Alex Nieto.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Costos de Operación	
Costo promedio por hora de un Ingeniero de Sistemas de planta	\$ 6,82
Número de incidentes promedio al año	10
Tiempo aproximado para resolver la falla (en horas)	3,00
Costo Promedio Por Año	\$ 204,55
Número de años de la solución	1
Costo estimado en el tiempo de vida de la solución	\$ 204,55

Tabla 54: Costos Operación. Autor: Alex Nieto.

Costos de Soporte Técnico	
Costo promedio por hora de un ingeniero de soporte	\$ 50,00
Número de incidentes promedio al año	25
Tiempo aproximado para dar soporte técnico por incidente (en horas)	1
Costo Promedio Por Año	\$ 1.250,00
Número de años de la solución	1
Costo estimado en el tiempo de vida de la solución	\$ 1.250,00

Tabla 55: Costos Soporte Técnico. Autor: Alex Nieto

Resumen de Costos de la Solución	
Costo Inicial de la Solución	\$ 2.200,00
Costo de Administración	\$ 1.080,00
Costo de Operación	\$ 204,55
Costo de Soporte	\$ 1.250,00
Costo de Hosting Anual	\$ 50,00
Total de la Solución	\$ 4.734,55

Tabla 56: Costos Totales. Autor: Alex Nieto.

5.4.3. *Mantenimiento.*

El mantenimiento se hace de manera versátil ya que al tratarse de un sitio web situado en un dominio propietario, su accesibilidad es prácticamente a nivel global lo cual facilita dar soporte o asistencia.

6. CAPÍTULO: CONCLUSIONES Y RECOMENDACIONES.

6.1. Conclusiones.

Producto del desarrollo de la presente disertación de grado concluyo que:

- ◆ Debido a que el proyecto se orientó a un proceso en especial se pudo manejar de manera más efectiva por medio de la herramienta SIPOC, la cual fue muy útil para conocer el proceso a fondo.
- ◆ Las constantes reuniones con el cliente fueron satisfactorias para el análisis de requerimientos para puesta a punto del sistema que se implantó en el barco turístico. El desarrollo de sistemas resulta mucho más sencillo cuando se genera un buen análisis de requerimientos previamente.
- ◆ La identificación de actores que utilizarán el sistema final resultó de gran ayuda para la creación de perfiles en el sistema.
- ◆ El uso de diagramas UML para el diseño del sistema fue de gran ayuda al momento del desarrollo del proyecto final por la facilidad de entendimiento gráfico que tiene. El levantamiento del proceso también sirvió para la generación de diagramas UML y las especificaciones principales del sistema.
- ◆ El Framework Kohana más las metodologías de Desarrollo Rápido de Aplicaciones (RAD) utilizados para este proyecto fueron de ayuda para un desarrollo rápido del sistema. La facilidad de aprendizaje dada por el lenguaje elegido facilita la solución de problemas en la etapa de desarrollo del sistema.
- ◆ El análisis de costos nos resulta útil para tener idea de qué tan viable es o no el desarrollar el sistema.
- ◆ El desarrollo de un prototipo ejecutable ayudó a tener una mejor perspectiva de los requerimientos del cliente.
- ◆ La elección de herramientas libres facilitó la búsqueda de información técnica de las mismas.
- ◆ El uso de tablas para la prueba de validaciones de usuario resultaron muy prácticas al momento de finalizar la construcción del sistema.
- ◆ La arquitectura Cliente/Servidor fue de ayuda al momento de implantar el sistema en la empresa.

- ◆ El análisis de seguridad funcionó para identificar las vulnerabilidades del sistema.

6.2. Recomendaciones.

Producto del desarrollo de la presente disertación de grado se presentan las siguientes recomendaciones:

- ◆ Elegir el método de desarrollo que se acople de mejor manera al proyecto dependiendo de sus requerimientos.
- ◆ Hacer una lista de riesgos antes de comenzar el proyecto.
- ◆ Hacer un análisis de costos a profundidad para conocer la viabilidad del proyecto.
- ◆ Generar un buen diagrama de casos de uso para entender a fondo el proceso en general.
- ◆ Tomarse el tiempo necesario para el diseño de la base de datos.
- ◆ Hacer los debidos comentarios en las líneas de código para poder entender cada módulo desarrollado.
- ◆ Se debería implementar un log en la base de datos para conocer el movimiento por parte de todos los usuarios que ingresan al sistema y generar un mayor control dentro del sistema.
- ◆ Realizar periódicamente copias de seguridad de la base de datos para mantener respaldos de la información.
- ◆ Generar una base de datos estable y escalable para que facilite el crecimiento en el futuro.
- ◆ Generar un producto final con la posibilidad de realizar cambios con facilidad y escalable.
- ◆ Para la segunda versión se debería implementar el manejo de contenedores de los productos, stock mínimo de productos y la integración a un sistema más grande.
- ◆ Una vez entregado el proyecto al usuario final dar capacitación sobre el sistema.

Bibliografía.

DE LIBROS:

- Lic. Karla Armijos, (2011). Geografía Turística Ecuatoriana: Guía Didáctica (1ª ed.). Ecuador: Editorial de la Universidad Técnica Particular de Loja.
- Gurría Di-Bella, Manuel, (1991). Introducción al Turismo: Concepto de turismo, Oferta y demanda turística, Marketing turístico (1ª ed.). México: Trillas.
- Ward Cunningham, (2003). Extreme Programming Pocket Guide (1ª ed.). Estados Unidos: O'Reilly.
- Mario A. Fernández Fernández, (2003). El Control: fundamento de la gestión por procesos y la calidad total (2ª ed.). España: Esic.
- Boehm, B.W., (1991). Software Risk Management: Principles and practices, IEEE Software, Estados Unidos.
- Charette, R., (1989). Software Engineering Risk analysis and management, McGraw-Hill.
- Karolak, (1996). D.W. Software Engineering Risk Management, IEEE Computer Society Press, USA.
- Pressman, R.S., (2001). Ingeniería del Software, un enfoque práctico, Mc Graw Hill.

DE INTERNET:

- http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software, Enciclopedia Libre "Ingeniería de Software", WIKIPEDIA.ORG, 30/09/2012.
- <http://es.wikipedia.org/wiki/Diagramas>, Enciclopedia Libre "Diagramas", WIKIPEDIA.ORG, 30/09/2012.
- <http://es.scribd.com/doc/5366642/02-Analisis-de-problemas-de-estacionamiento>, Documento "ANÁLISIS DE PROBLEMAS DE ESTACIONAMIENTO EN GRANDES INSTITUCIONES: APLICACIÓN AL CASO DE UN CAMPOS UNIVERSITARIO", Patricio Donoso Ibáñez, 03/09/2012.
- <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>, Diapositivas "Metodologías de Desarrollo de Software", Grupo Alarcos, 03/09/2012.
- <http://www.buenastareas.com/ensayos/Sistema-De-Monitoreo-Para-Ingreso-De/582443.html>, Ensayo "Sistema De Monitoreo Para Ingreso De Automoviles A Parqueaderos", "Sitio Web de Apoyo Educativo", ComScore, 16/10/2012.
- <http://www.sznet.com.ar/sistemas/Informacion.php>, Documento "Información sobre Código de Barras", sz.net Empresa ayuda Áreas Informáticas, 16/10/2012
- <http://es.wikipedia.org/wiki/Fluctuaci%C3%B3n>, Enciclopedia Libre "Fluctuación", WIKIPEDIA.ORG, 16/10/2012.
- <http://es.wikipedia.org/wiki/Decodificador>, Enciclopedia Libre "Decodificador", WIKIPEDIA.ORG, 16/10/2012.
- <http://es.wikipedia.org/wiki/Diodo>, Enciclopedia Libre "Diodo", WIKIPEDIA.ORG, 16/10/2012.
- <http://www.docolutions.com.mx/es/soluciones/servicios/digitalizacion.html>, Web DocSolutions Documento "Digitalización", DocSolutions, 16/10/2012.

- <http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node9.html> , Web de Información “Necesidad de una metodología”, José R. Álvarez y Manuel Arias – UNED, 16/10/2012
- <http://es.scribd.com/doc/51066630/64/Metodologias-orientadas-a-objetos>, Biblioteca en Línea, Documento “SISTEMAS DE INFORMACIÓNEN LA ERA DIGITAL”, Luis Alberto Mamani Pinto, 16/10/2012.
- <http://glbrtlmb.blogdiario.com/i2006-09/>, Blog de Opinión Libre Documento “ANALISIS DISEÑO ORIENTADO A OBJETOS”, BlogDiario.com, 16/10/2012.
- <http://ccia.ei.uvigo.es/docencia/SCS>, PDF Online “Modelo Cliente/Servidor”, FJRP, FMBR, 16/10/2012.

INVESTIGADOR

- Alex Nieto, Estudiante de Ingeniería en Sistemas, PUCE.

Trabajos Citados.

Wikipedia. (24 de 01 de 2013). Recuperado el 31 de 01 de 2013, de http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

BAUTISTA, L. A. (24 de 08 de 2010). *Wordpress*. Recuperado el 31 de 01 de 2013, de <http://alberto121888.files.wordpress.com/2010/08/ensayo-postgresql-vs-mysql-vs-oracle.pdf>

Wikipedia. (29 de 01 de 2013). *Wikipedia*. Recuperado el 31 de 01 de 2013, de http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado

Wikipedia. (23 de 01 de 2013). *Wikipedia*. (Wikipedia) Recuperado el 31 de 01 de 2013, de http://es.wikipedia.org/wiki/GNU_General_Public_License

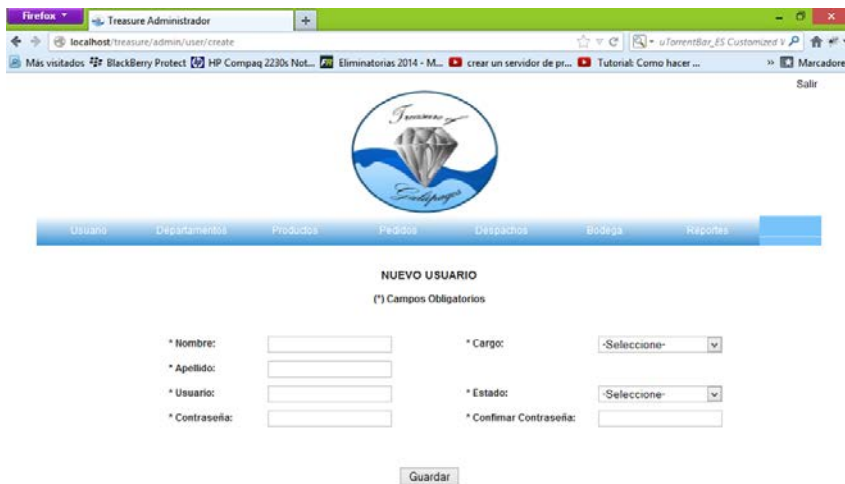
Anexo A

1. Manual de Usuario

1.1. Módulo Usuarios.

En este módulo se hacen los ingreso, modificación y cambio de estado de usuarios que hacen uso del sistema, manejando una lista de perfiles previamente instalado, con lo cual restringe a cada usuario cumplir con su rol. (Los nombres de usuario no se pueden repetir ni contener números).

1.1.1. Ingreso Nuevo Usuario.



The screenshot shows a web browser window titled 'Treasure Administrador' with the URL 'localhost:treasure/admin/user/create'. The page features a logo for 'Treasure of Galapagos' and a navigation menu with items: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', 'Bodega', and 'Reportes'. The main content area is titled 'NUEVO USUARIO' and includes a note '(*) Campos Obligatorios'. The form contains the following fields:

* Nombre:	<input type="text"/>	* Cargo:	<input type="text" value="-Seleccione-"/>
* Apellido:	<input type="text"/>	* Estado:	<input type="text" value="-Seleccione-"/>
* Usuario:	<input type="text"/>	* Confirmar Contraseña:	<input type="text"/>
* Contraseña:	<input type="text"/>		

Below the fields is a 'Guardar' button.

Se llenan los campos correspondientes; todos los campos en este formulario son obligatorios; en el caso de que algún campo obligatorio falte se desplegara una alerta con un mensaje correspondiente al error.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

Firefox - Treasure Administrador

localhost:treasure/admin/users/create

Más visitados BlackBerry Protect HP Compaq 2230c Not... Eliminatorias 2014 - M... crear un servidor de pr... Tutorial: Como hacer ... Marcadores

Salir

Usuario Departamentos Productos Pedidos Despachos Boques Reportes

NUEVO USUARIO
(*) Campos Obligatorios

- El campo "Nombre" es obligatorio.
- El campo "Cargo" es obligatorio.
- El campo "Apellido" es obligatorio.
- El campo "Usuario" es obligatorio.
- El campo "Estado" es obligatorio.
- El campo "Contraseña" es obligatorio.
- El campo "Confirmar Contraseña" es obligatorio.

* Nombre: * Cargo:

* Apellido: * Estado:

* Usuario: * Contraseña:

* Confirmar Contraseña:

Guardar

Nota: el caso de que se quiera ingresar un solicitante se especificara el departamento que aparecerá si se elige en la opción cargo.

1.1.2. Reporte de Usuarios en el Sistema.

En esta página se pueden observar todos los usuarios que se encuentran ingresados en el sistema ya estén activos o inactivos. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.

SELECCIONAR USUARIO

[Clear Usuario](#)

Exportar a Excel

NOMBRE	APELLIDO	CARGO	DEPARTAMENTO	USUARIO	ESTADO	EDITAR
administradoruno	administradoruno	administrador	NINGUNO	admin	Activo	
solicitante	solicitante	solicitante	cocina	cocina	Activo	
solicitante	solicitante	solicitante	habitaciones	habitaciones	Activo	
solicitante	solicitante	solicitante	limpieza	limpieza	Activo	

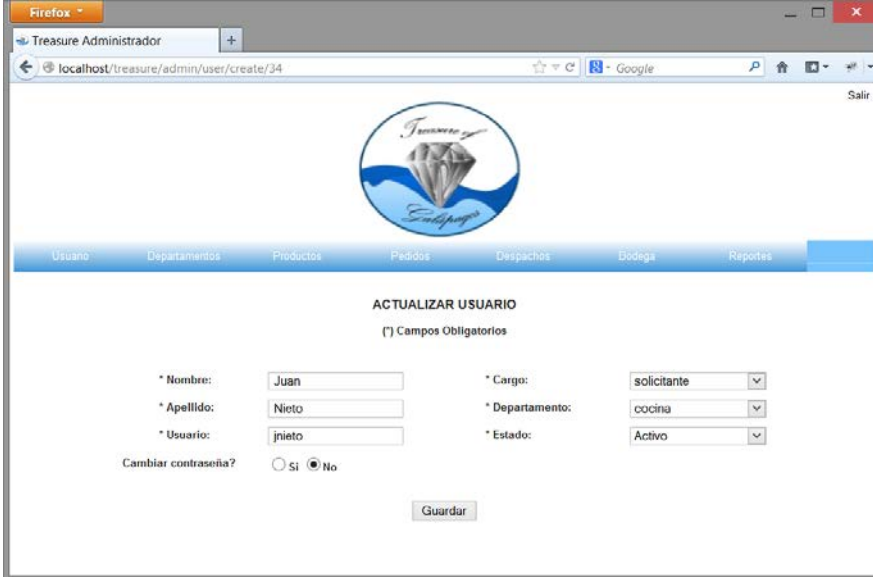
Mostrar 10 registros por Página

Mostrando 1 a 4 de 4 registros

Primera Anterior 1 Siguiente Última

Nota: Si se da click en la parte del lápiz el sistema redirige a la pantalla de edición.

1.1.3. Edición de Usuarios.



The screenshot shows a web browser window titled 'Treasure Administrador' with the URL 'localhost/treasure/admin/user/create/34'. The page features a logo for 'Treasure of Galapagos' and a navigation menu with items: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', 'Bodega', 'Reportes', and 'Salir'. The main content area is titled 'ACTUALIZAR USUARIO' and includes a note '(*) Campos Obligatorios'. The form contains the following fields:

* Nombre:	<input type="text" value="Juan"/>	* Cargo:	<input type="text" value="solicitante"/>
* Apellido:	<input type="text" value="Niето"/>	* Departamento:	<input type="text" value="cocina"/>
* Usuario:	<input type="text" value="jniето"/>	* Estado:	<input type="text" value="Activo"/>

Below the fields, there is a section for 'Cambiar contraseña?' with radio buttons for 'Si' and 'No' (selected). A 'Guardar' button is located at the bottom of the form.

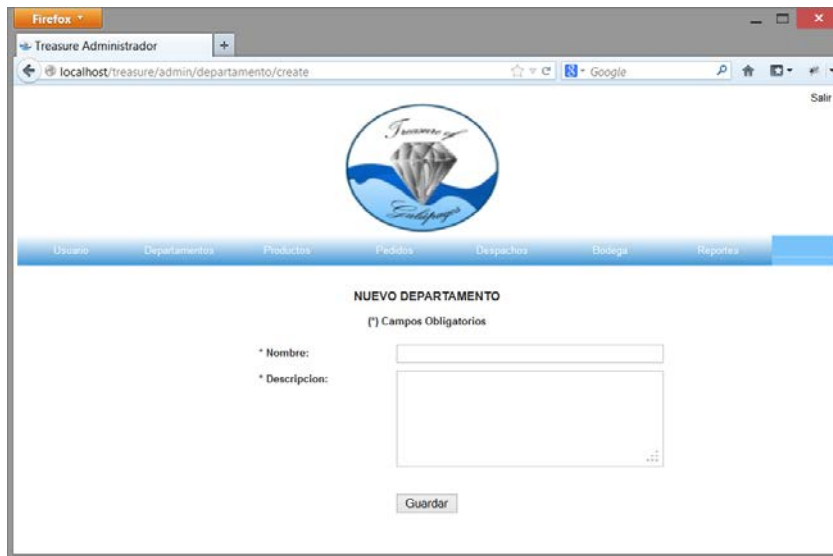
Esta pantalla que es parecida a la de ingreso de usuario, en la cual se pueden editar los campos que se deseen; también sirve para cambiar o no la contraseña del usuario.

Nota: Esta ventana tiene los mismos controles que la página de ingreso de usuario.

1.2. Módulo Departamentos.

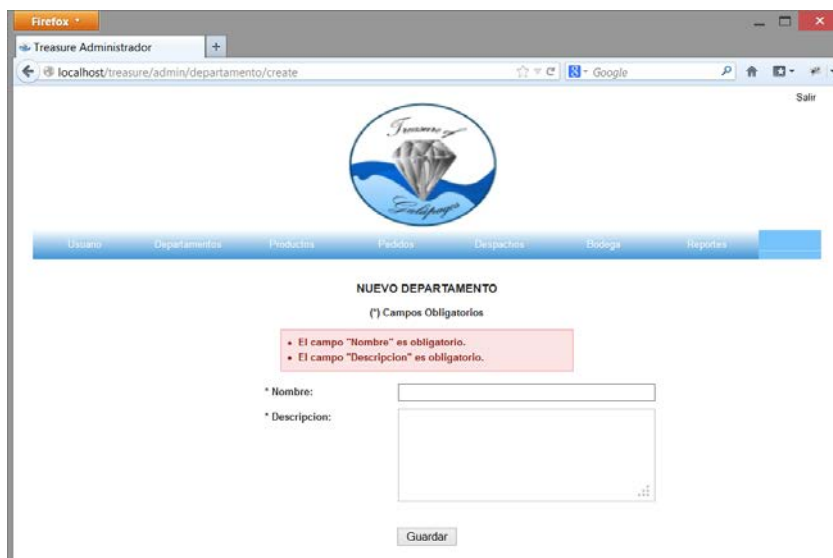
En este módulo se hacen los ingresos y modificaciones de los departamentos que existen en el sistema, el departamento una vez creado no se lo puede eliminar. (Los nombres de los departamentos no se pueden repetir).

1.2.1. Ingreso Nuevo Departamento.



The screenshot shows a web browser window titled 'Treasure Administrador' with the URL 'localhost/treasure/admin/departamento/create'. The page features a logo for 'Treasure of Galapagos' and a navigation menu with items: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', 'Bodega', 'Reportes', and 'Salir'. The main content area is titled 'NUEVO DEPARTAMENTO' and includes the text '(*) Campos Obligatorios'. There are two input fields: '* Nombre:' and '* Descripción:'. A 'Guardar' button is located at the bottom of the form.

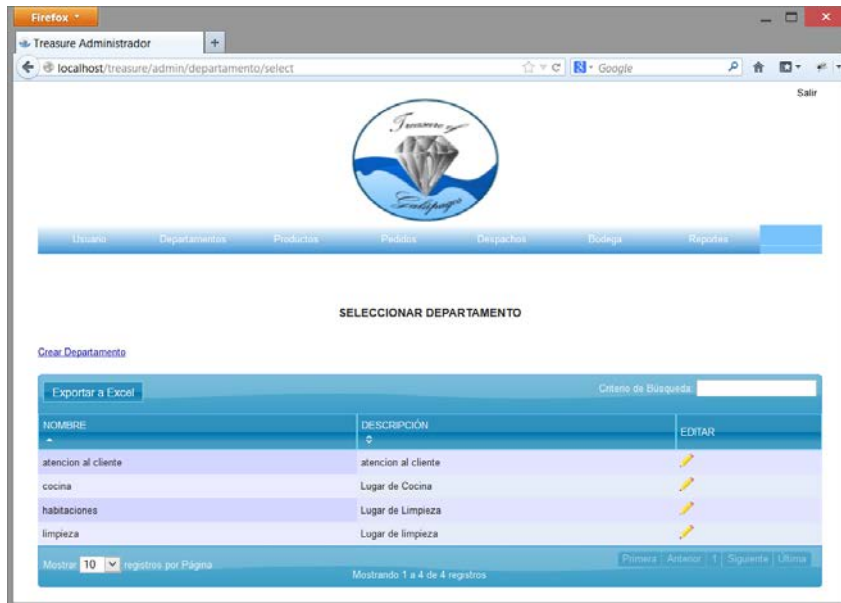
Se llenan los campos correspondientes; todos los campos en este formulario son obligatorios; en el caso de que algún campo obligatorio falte se desplegara una alerta con un mensaje correspondiente al error.



This screenshot shows the same 'Nuevo Departamento' form as above, but with an error message displayed in a red box. The error message reads: '• El campo "Nombre" es obligatorio. • El campo "Descripción" es obligatorio.' The 'Nombre' and 'Descripción' fields are empty, and the 'Guardar' button is still visible at the bottom.

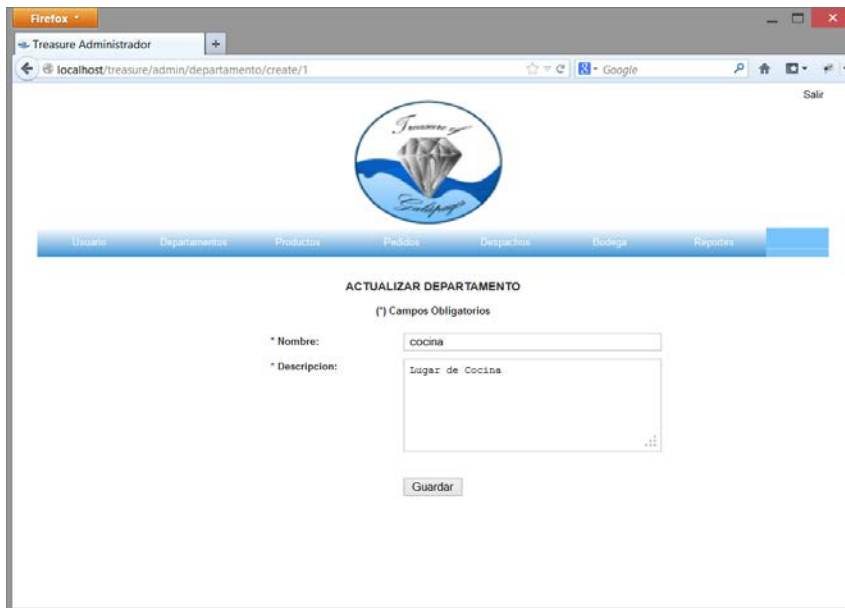
1.2.1. Reporte de Departamentos en el Sistema.

En esta página se pueden observar todos los departamentos que se encuentran ingresados en el sistema. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.



Nota: Si se da click en la parte del lápiz el sistema redirige a la pantalla de edición.

1.2.2. Edición de Departamentos.



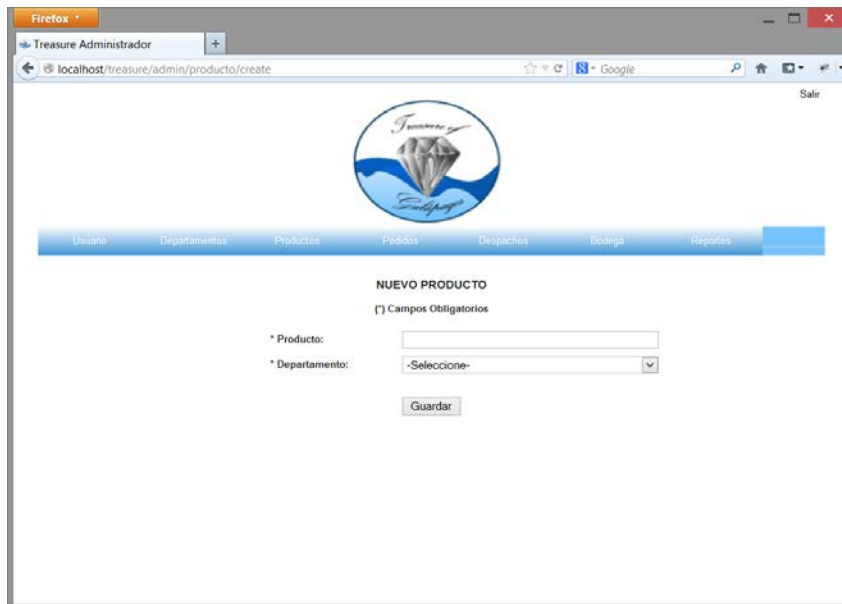
Esta pantalla que es parecida a la de ingreso de departamento, en la cual se pueden editar los campos que se deseen.

Nota: Esta ventana tiene los mismos controles que la página de ingreso de departamentos.

1.3. Módulo Productos.

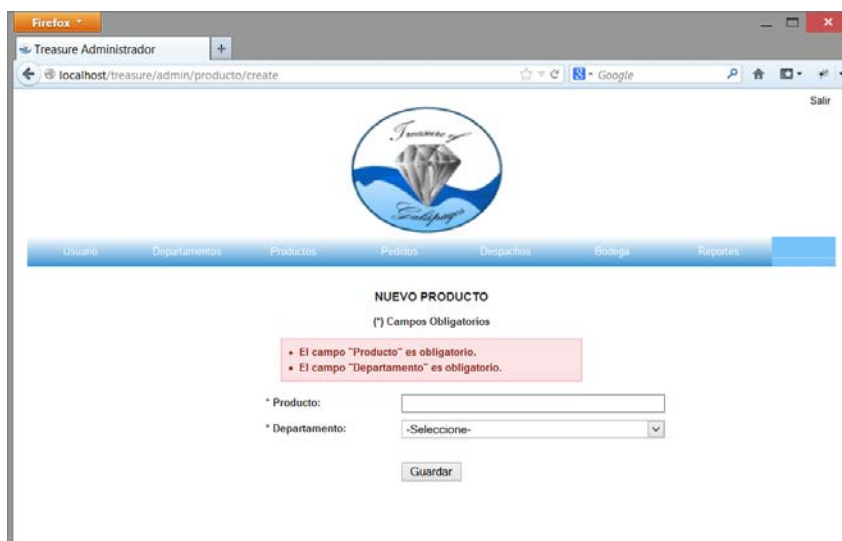
En este módulo se hacen los ingresos y modificaciones de los productos que existen en el sistema, el producto una vez creado no se lo puede eliminar. (Los nombres de los productos no se pueden repetir).

1.3.1. Ingreso Nuevo Producto



The screenshot shows a web browser window titled 'Treasure Administrador' with the URL 'localhost/treasure/admin/producto/create'. The page features a logo for 'Treasure of Galapagos' and a navigation menu with items: 'Usuario', 'Departamentos', 'Productos', 'Pedidos', 'Despachos', 'Bodega', and 'Reportes'. The main content area is titled 'NUEVO PRODUCTO' and includes a note '(*) Campos Obligatorios'. There are two input fields: '* Producto:' (a text box) and '* Departamento:' (a dropdown menu with '-Seleccione-' selected). A 'Guardar' button is located below the fields.

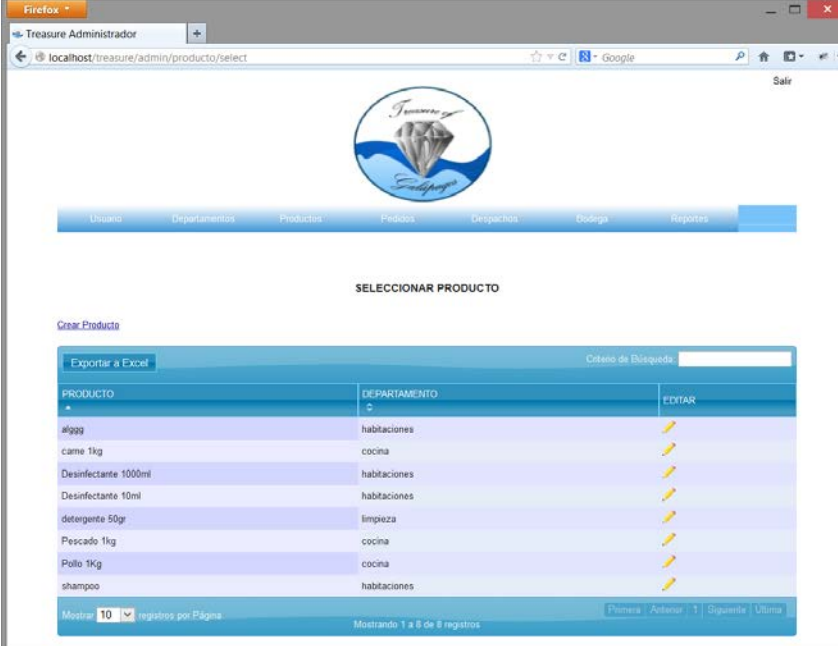
Se llenan los campos correspondientes; todos los campos en este formulario son obligatorios; en el caso de que algún campo obligatorio falte se desplegara una alerta con un mensaje correspondiente al error.



This screenshot shows the same 'Nuevo Producto' form as above, but with an error message displayed in a red box. The message reads: '• El campo "Producto" es obligatorio. • El campo "Departamento" es obligatorio.' The input fields and the 'Guardar' button are still visible below the error message.

1.3.2. Reporte de Productos en el Sistema.

En esta página se pueden observar todos los productos que se encuentran ingresados en el sistema. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.



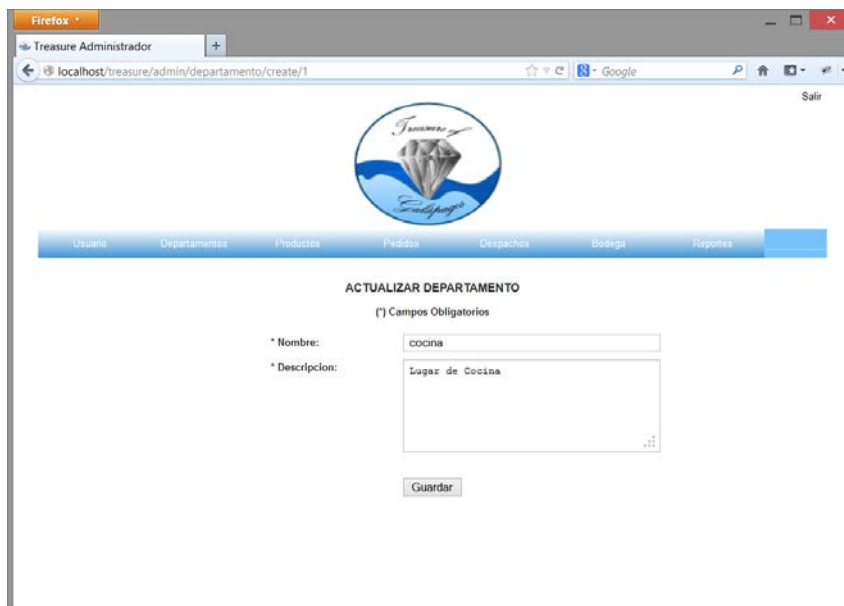
The screenshot displays the 'Selección de Producto' interface. At the top, there is a navigation menu with options: Inicio, Departamentos, Productos, Pedidos, Despedidos, Bodega, Reportes, and Salir. Below the menu is a search bar labeled 'Criterio de Búsqueda'. The main content area features a table with the following data:

PRODUCTO	DEPARTAMENTO	EDITAR
aligg	habitaciones	[pencil icon]
carne 1kg	cocina	[pencil icon]
Desinfectante 1000ml	habitaciones	[pencil icon]
Desinfectante 10ml	habitaciones	[pencil icon]
detergente 50gr	limpieza	[pencil icon]
Pescado 1kg	cocina	[pencil icon]
Pollo 1Kg	cocina	[pencil icon]
shampoo	habitaciones	[pencil icon]

At the bottom of the table, there is a pagination control showing 'Mostrar 10 registros por Página' and 'Mostrando 1 a 8 de 8 registros'.

Nota: Si se da click en la parte del lápiz el sistema redirige a la pantalla de edición.

1.3.3. Edición de Productos.



Esta pantalla que es parecida a la de ingreso de productos, en la cual se pueden editar los campos que se deseen.

Nota: Esta ventana tiene los mismos controles que la página de ingreso de productos.

1.4. Módulo Pedidos.

En este módulo podremos entender cómo generar y procesar pedidos en el sistema.

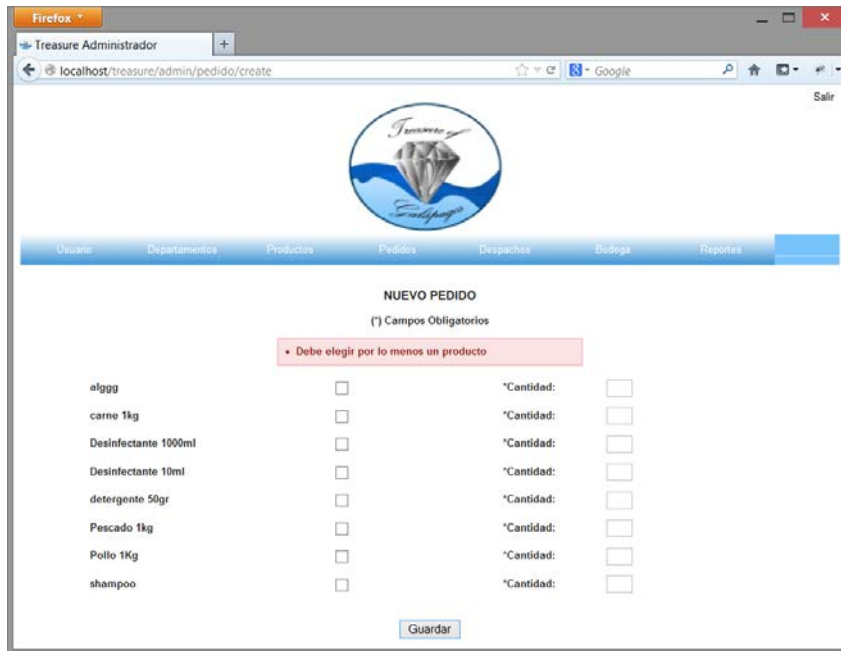
1.4.1. Nuevo Pedido.

Producto	Seleccionar	Cantidad
alggg	<input type="checkbox"/>	<input type="text"/>
carne 1kg	<input type="checkbox"/>	<input type="text"/>
Desinfectante 1000ml	<input type="checkbox"/>	<input type="text"/>
Desinfectante 10ml	<input type="checkbox"/>	<input type="text"/>
detergente 50gr	<input type="checkbox"/>	<input type="text"/>
Pescado 1kg	<input type="checkbox"/>	<input type="text"/>
Pollo 1Kg	<input type="checkbox"/>	<input type="text"/>
shampoo	<input type="checkbox"/>	<input type="text"/>

Guardar

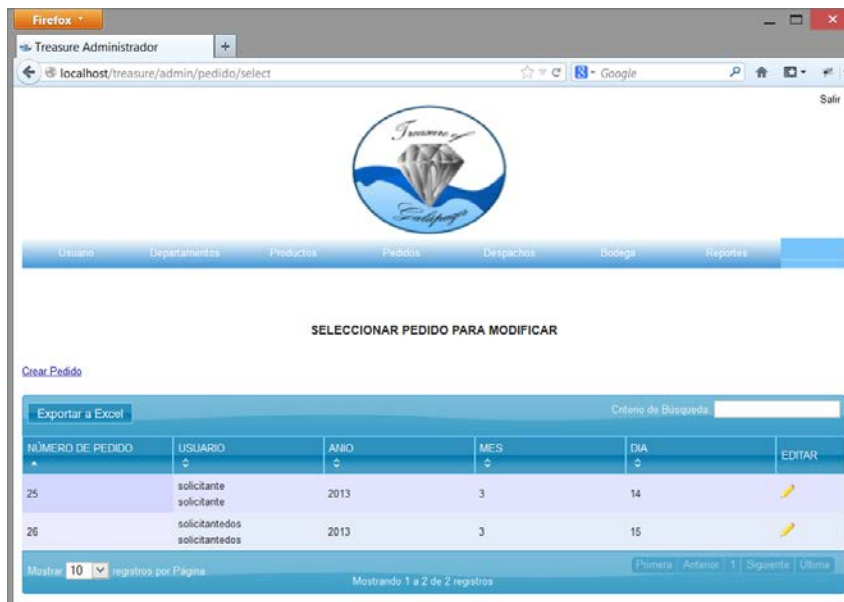
En esta página se observan todos los productos en forma de una lista para a facilidad de elección de los mismos; los productos desplegados varían dependiendo del usuario que ingreso al sistema, los solicitantes solo pueden ver los productos del departamento al cual pertenecen. El administrador puede ver todos los productos registrados. Es necesario elegir por lo menos un producto, si se elige uno se debe especificar una cantidad positiva. Si se comete un error se desplegara el respectivo mensaje de error.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS



1.4.2. Reporte de Pedidos no Aprobados.

En esta página se pueden observar todos los pedidos que se encuentran ingresados en el sistema en estado de 'Pedido'. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.



Nota: Si se da click en la parte del lápiz el sistema redirige a la pantalla de edición.

1.4.3. Edición de Pedido.

Item	Check	*Cantidad:
olggg	<input type="checkbox"/>	<input type="text"/>
carne 1kg	<input checked="" type="checkbox"/>	23
Desinfectante 1000ml	<input type="checkbox"/>	<input type="text"/>
Desinfectante 10ml	<input type="checkbox"/>	<input type="text"/>
detergente 50gr	<input type="checkbox"/>	<input type="text"/>
Pescado 1kg	<input type="checkbox"/>	<input type="text"/>
Pollo 1Kg	<input type="checkbox"/>	<input type="text"/>
shampoo	<input type="checkbox"/>	<input type="text"/>

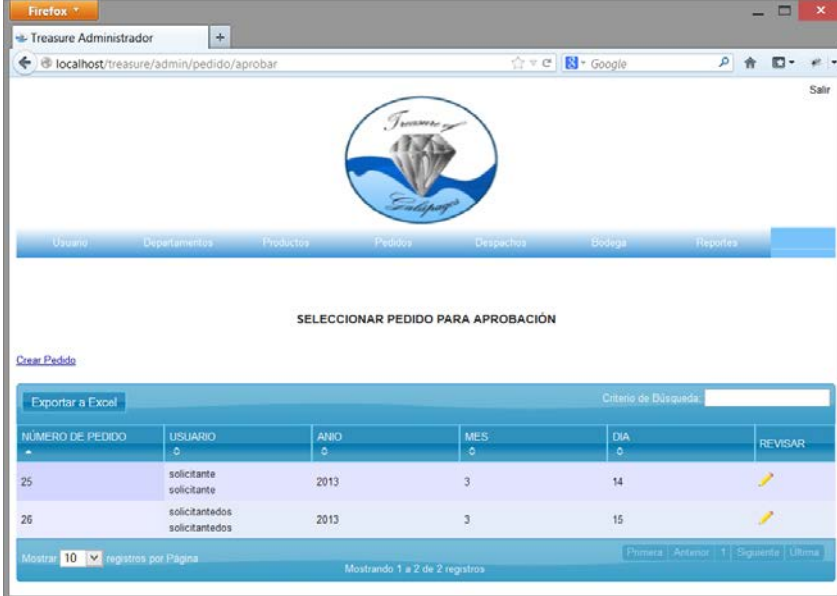
Guardar

Esta pantalla que es parecida a la de ingreso de pedidos, en la cual se pueden editar los campos que se deseen.



Nota: Esta ventana tiene los mismos controles que la página de ingreso de pedidos.

1.4.4. Selección de Pedido Para Aprobación.

En esta página se pueden observar todos los pedidos que se encuentran ingresados en el sistema en estado de 'Pedido'. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.



The screenshot shows a web browser window with the URL `localhost:treasure/admin/pedido/aprobar`. The page title is 'Treasure Administrador'. The main content area is titled 'SELECCIONAR PEDIDO PARA APROBACIÓN'. Below the title, there is a 'Crear Pedido' link and a search bar labeled 'Criterio de Búsqueda:'. A table displays the following data:

NÚMERO DE PEDIDO	USUARIO	AÑO	MES	DÍA	REVISAR
25	solicitante	2013	3	14	
26	solicitantados	2013	3	15	

At the bottom of the table, there is a pagination control showing 'Mostrar 10 registros por Página' and 'Mostrando 1 a 2 de 2 registros'. Navigation buttons for 'Primera', 'Anterior', '1', 'Siguiente', and 'Última' are also present.

Nota: Si se da click en la parte del lápiz el sistema redirige a la pantalla de revisión.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

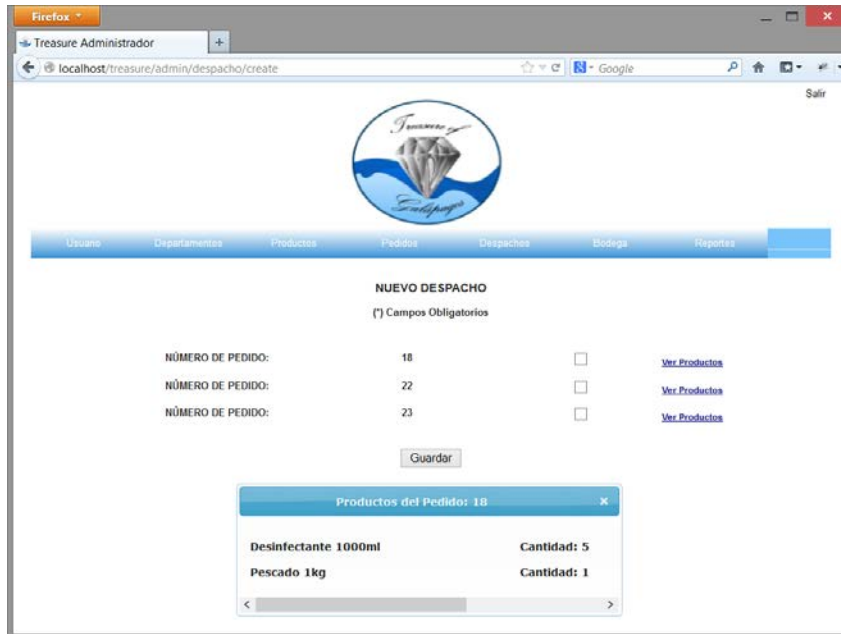


En esta página se puede observar los productos que pertenecen al pedido y se le da las dos posibilidades de cambiar el estado del pedido. Puede 'Aprobar' o 'Cancelar' los pedidos.

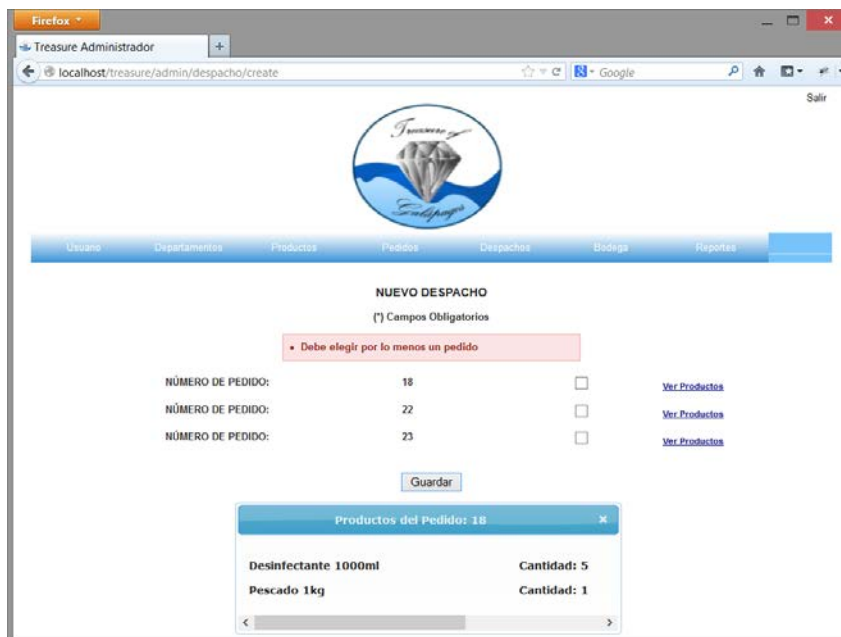
Nota: Una vez aprobados o cancelados no se podrán visualizar los pedidos en la página anterior.

1.5. Módulo Despachos.

1.5.1. Nuevo Despacho.

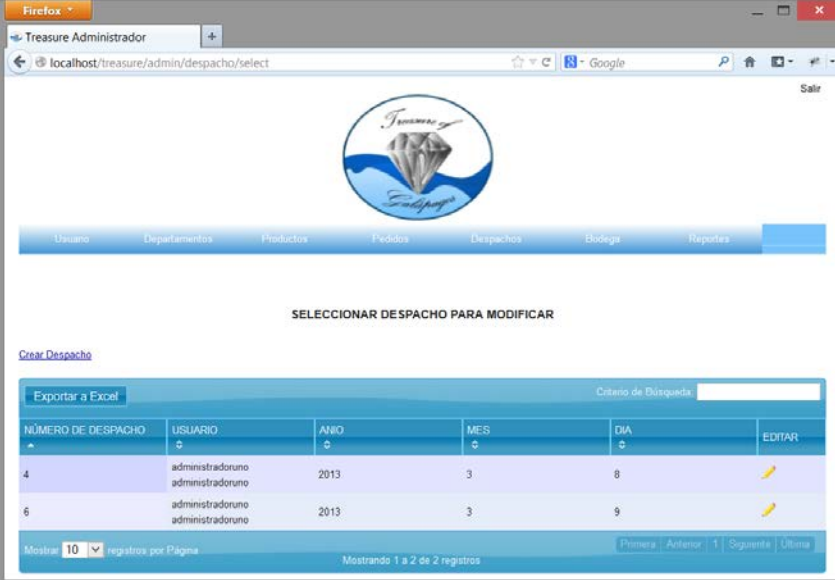


En esta página se observan todos los pedidos en estado de 'Aprobados' en forma de una lista para a facilidad de elección de los mismos; se puede hacer click en 'ver productos' para visualizar los productos que están incluidos en ese pedido. Es necesario elegir por lo menos un pedido. Si se comete un error se desplegara el respectivo mensaje de error.



1.5.2. Reporte de Despachos no Comprados.



En esta página se pueden observar todos los despachos que se encuentran ingresados en el sistema en estado de 'En Proceso'. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.



SELECCIONAR DESPACHO PARA MODIFICAR

[Crear Despacho](#)

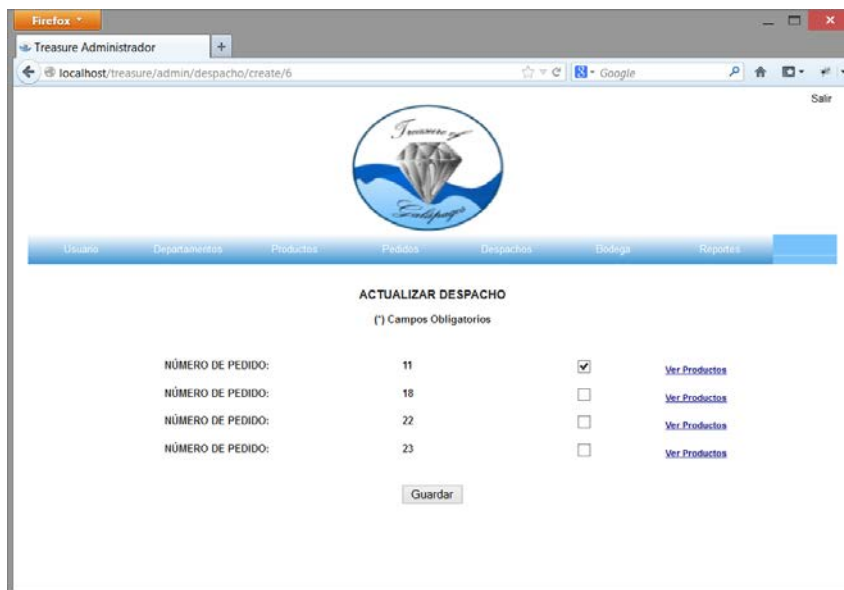
Exportar a Excel

NÚMERO DE DESPACHO	USUARIO	AÑO	MES	DÍA	EDITAR
4	administradoruno	2013	3	8	
6	administradoruno	2013	3	9	

Mostrar registros por Página Mostrando 1 a 2 de 2 registros [Primera](#) [Anterior](#) | 1 | [Siguiente](#) [Última](#)

Nota: Si se da click en la parte del lápiz el sistema redirige a la pantalla de edición.

1.5.3. Edición de Despacho.



Esta pantalla que es parecida a la de ingreso de despachos, en la cual se pueden editar los campos que se deseen.

Nota: Esta ventana tiene los mismos controles que la página de ingreso de despachos.

1.5.4. Selección de Despacho Para Procesar.

En esta página se pueden observar todos los despachos que se encuentran ingresados en el sistema. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.

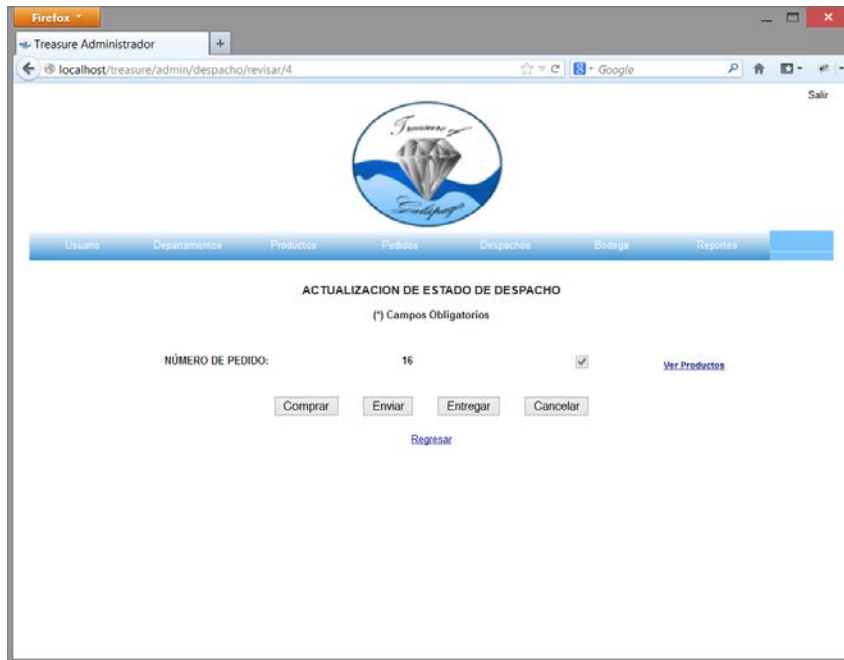
Exportar a Excel

NUMERO DE DESPACHO	USUARIO	AÑO	MES	DIA	REVISAR
3	administradoruno	2013	3	8	
4	administradoruno	2013	3	8	
6	administradoruno	2013	3	9	
10	administradoruno	2013	3	11	

Mostrar 10 registros por Página Primera | Anterior | Siguiente | Última

Nota: Si se da click en la parte del lápiz el sistema redirige a la pantalla de revisión. El Administrador puede ver todos los despachos a excepción de los que se encuentren en 'Entregado', El Proveedor solo puede ver los que no estén en 'Entregado'. Y el controlador solo puede ver los que estén en 'Enviado'.

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS



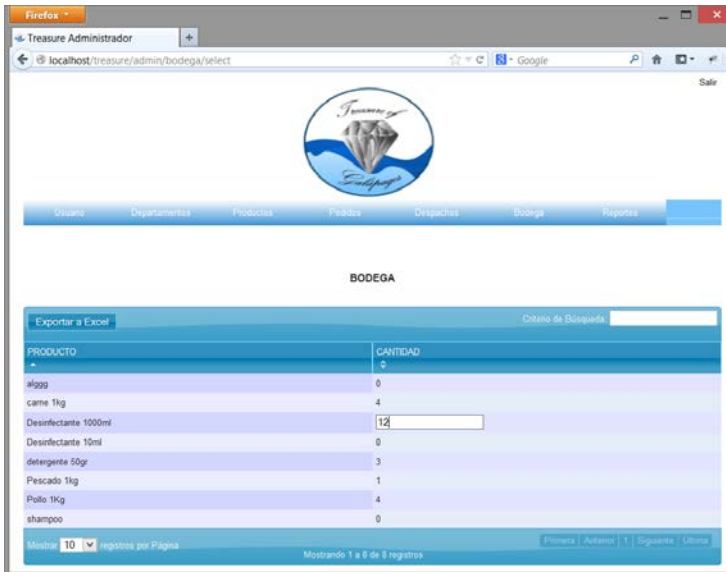
En esta página se puede observar los despachos que están en estado 'En Proceso'. Se puede cambiar su estado: 'Comprar' o 'Enviar' o 'Entregar' o 'Cancelar'.

Nota: Los botones varían según el usuario. El administrador tiene todos los privilegios, el proveedor solo puede 'Comprar', 'Enviar' o 'Cancelar' mientras que el controlador solo puede confirmar el estado 'Entregar'.

1.6. Módulo de Manejo de Bodega.

En este módulo se verá el manejo de la bodega o stock que se puede hacer con dos perfiles diferentes tanto para administrador y para un solicitante que solo puede hacer egresos en la bodega.

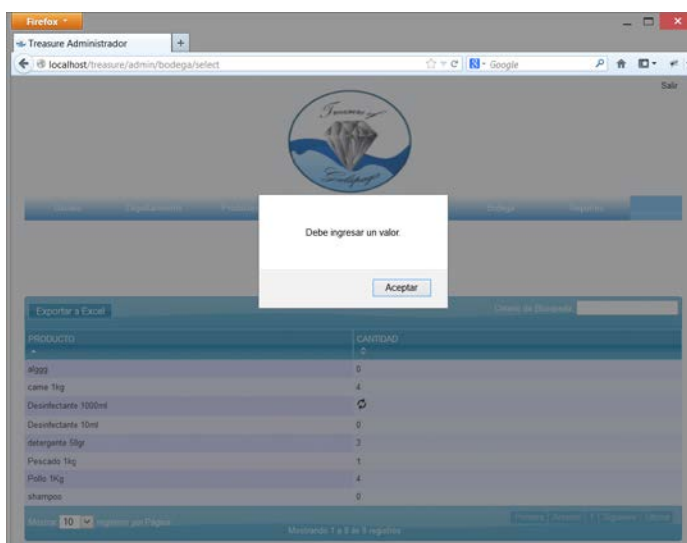
1.6.1. Ventana de Administrador.



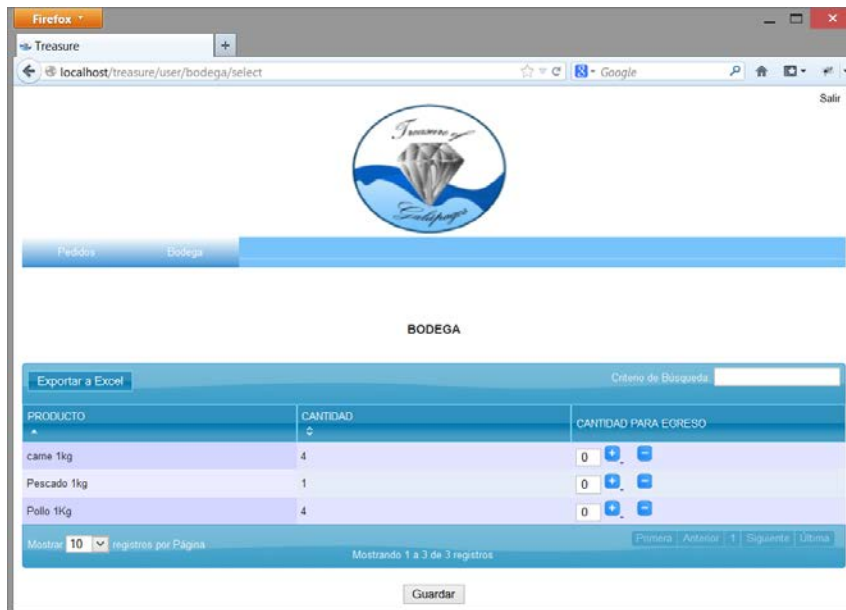
PRODUCTO	CANTIDAD
Huevo	0
carne 1kg	4
Desinfectante 1000ml	12
Desinfectante 10ml	0
detergente 50gr	3
Pescado 1kg	1
Pollo 1kg	4
shampoo	0

En esta ventana el administrador basta con que de doble click para que genere una edición directa en la bodega de la cantidad de productos que constan en la bodega.

Nota: Esta página también cuenta con controles de errores no se admiten letras o que se deje en blanco.

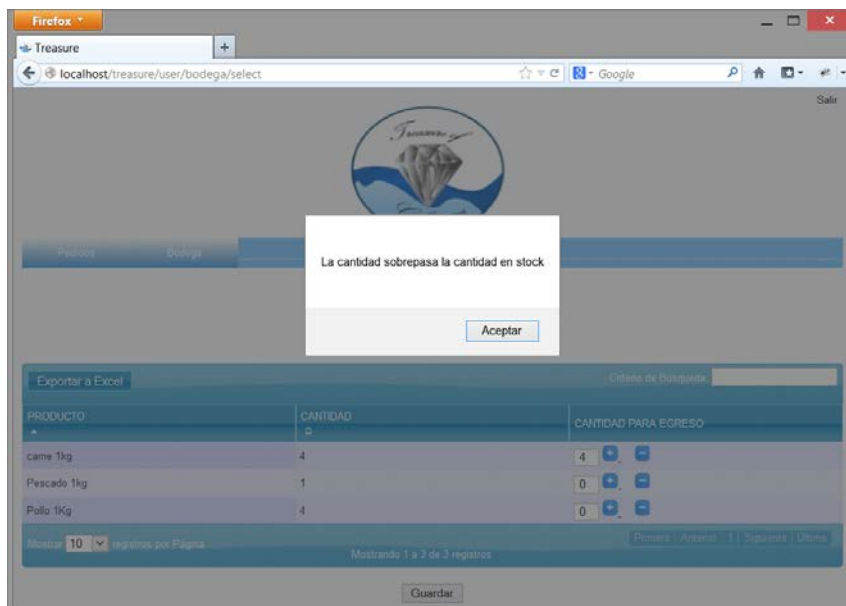


1.6.2. Ventana de Solicitante



En esta ventana se puede dar click en el signo de '+' o de '-' para hacer el egreso correspondiente en la bodega.

Nota: Esta página presenta control de usuarios, no se pueden generar egresos negativos ni tampoco mayores al stock que se tiene en bodega.



Anexo B

1. Manual Técnico

Debido a que se utilizó un framework que tiene código propio y estandarizado se incluirá el código generado para el sistema de manera esquemática para su entendimiento.

1.1. Login

1.1.1. Contenedor

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Containers_Login extends Controller_Template {  
  
    /**  
     * El contenedor que va utilizar el controlador.  
     * Debe referirse a un archivo .php que se encuentra en  
     * la carpeta views.  
     * @var String  
     */  
    public $template = 'containers/login';  
  
    /**  
     * El atributo que va a ser utilizado en cada acción  
     * NULL por defecto  
     * @var View  
     */  
    public $view;  
  
    /**  
     * @see system/classes/kohana/controller/Kohana_Controller_Template::before()  
     */  
    public function before()  
    {  
        //Required because we are extending Controller_Template  
        parent::before();  
  
        if ( $this->auto_render )  
        {  
            //Sets the "Default" title set on the container  
            $this->template->title = 'Treasure';  
            $this->template->content = '';  
  
            $this->template->styles = array();  
            $this->template->scripts = array();  
        }  
  
        //Cleans the view so that every action can set it  
        //by itself without problems  
        $this->view = NULL;  
    }  
  
    /**  
     * @see system/classes/kohana/controller/Kohana_Controller_Template::after()  
     */  
    public function after()  
    {  
        if ( $this->auto_render )  
        {  
            $action = $this->request->action();  
            $controller = $this->request->controller();  
  
            //Build the styles array  
            $css_file = 'media/css/controllers/'.$controller.'/'.$action.'.css';  
            $styles = array(  
                'media/css/screen.css' => 'screen, projection',
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
'media/css/src/print.css' => 'print',
'media/css/jquery-ui/smoothness/jquery-ui-1.8.16.custom.css' => 'all',
'media/css/dataTables/TableTools.css' => 'screen',
'media/css/dataTables/TableTools_JUI.css' => 'screen',
'media/css/dataTables/demo_table_jui.css' => 'screen',
'media/css/src/ie.css' => 'screen, projection',
'media/css/styles.css' => 'screen',
);

//VALIDATING FILE
if(file_exists($scss_file) )
{
    $styles = array_merge( $styles, array($scss_file=>'screen') );
}

//Build the scripts array
$js_file = 'media/js/controller/'.$_controller.'/'.$_action.'.js';
$scripts = array(
    'media/js/jqueryLibrary/jquery-1.6.2.min.js',
    'media/js/jqueryLibrary/jquery-ui-1.8.16.custom.min.js',
    'media/js/jqueryLibrary/jquery-validation.js',
    'media/js/dataTables/jquery.dataTables.js',
    'media/js/dataTables/TableTools.js',
    'media/js/dataTables/ZeroClipboard.js',
    'media/js/globalfunctions.js',
);

//VALIDATING FILE
if(file_exists($js_file) )
{
    $scripts = array_merge( $scripts, array($js_file) );
}

//Set styles and scripts on the template (container). This can also be
//done from any action method
$this->template->styles = array_merge( $styles, $this->template->styles );
$this->template->scripts = array_merge( $scripts, $this->template->scripts );

//If $this->view is set display it on the
//containers $content
$this->template->content = $this->view;
}

//Required because we are extending Controller_Template
parent::after();
}
}
```

1.1.2. Controlador

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Login extends Controller_Containers_Login {
```

```
    /**
```

```
     * DEFAULT ACTION
```

```
     */
```

```
    public function action_index()
```

```
    {
```

```
        Session::instance()->set( 'treasure_usuario', FALSE );
```

```
        //IF USER SUBMITTED FORM
```

```
        if( $this->request->post() )
```

```
        {
```

```
            //VERIFYING USER
```

```
            $user = Model_User::authenticate( arr::get( $this->request->post(), 'txtUsername' ), md5( arr::get( $this->request->post(), 'txtPassword' ) ) );
```

```
            if( $user )
```

```
            {
```

```
                $username_usr = $user->USUARIO;
```

```
                $type = $user->ID_PERFIL;
```

```
                $id = $user->ID_USUARIO;
```

```
                $usertype_usr = Model_Perfil::verifica_perfil( $type );
```

```
            }
```

```
            if (isset($username_usr))
```

```
            {
```

```
                Session::instance()->set('treasure_usuario', $id);
```

```
                Session::instance()->set('exceptions', array());
```

```
                if($usertype_usr==1)
```

```
                {
```

```
                    HTTP::redirect('admin/main');
```

```
                }
```

```
                if($usertype_usr==2)
```

```
                {
```

```
                    HTTP::redirect('user/main');
```

```
                }
```

```
                if($usertype_usr==3)
```

```
                {
```

```
                    HTTP::redirect('manager/main');
```

```
                }
```

```
                if($usertype_usr==4)
```

```
                {
```

```
                    HTTP::redirect('control/main');
```

```
                }
```

```
                if($usertype_usr==5)
```

```
                {
```

```
                    HTTP::redirect('supplier/main');
```

```
                }
```

```
            }
```

```
            else
```

```
            {
```

```
                $this->view = View::factory( 'controller/login' );
```

```
                $this->view->set( 'error_validation', 1 );
```

```
            }
```

```
            else
```

```
            {
```

```
                $this->view = View::factory( 'controller/login' );
```

```
                $this->view->set( 'error_validation', 0 );
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

1.1.3. Contenedor de Vista

```
<!doctype html>
<html lang="<?php echo l18n::$lang ?>">
<head>
  <link rel="shortcut icon" type="image/x-icon" href="/treasure/media/img/icon.ico">
  <meta charset="utf-8">
  <title><?php echo $title ?></title>
  <?php foreach ($styles as $file => $type) echo HTML::style($file, array('media' => $type)), PHP_EOL ?>
  <?php foreach ($scripts as $file) echo HTML::script($file), PHP_EOL ?>
</head>
<body>
  <div class="container">
    <div class="span-24 content_container">
      <div class="span-24 content">
        <div class="prepend-10 span-12">
          <div class="span-12 login">&nbsp;</div>
        </div>
        <div class="span-24 ">
          <?php echo $content ?>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

1.1.4. Controlador de Vista

```
<div class="span-24">
  <?php if( $error_validation == 1 ){?>
    <div class="prepend-8 span-8 append-8" style=" padding-top: 20px;">
      <div class="span-8 error" align="center">
        <ul>
          <li>El usuario o la contraseña está incorrectos.</li>
        </ul>
      </div>
    </div>
    <?php }?>
    <div class="prepend-8 span-8 append-8">
      <ul id="alerts" class="alerts">
      </ul>
    </div>

    <form method="post" id="frmLogin" name="frmLogin">
      <br />
      <div class="prepend-9 span-6">
        <div class="span-6 line">
          <label class="span-6" for="txtUsername">Usuario:</label>
          <input type="text" class="span-6" name="txtUsername" id="txtUsername" title="Por favor ingrese su Nombre de Usuario" />
        </div>
        <div class="span-6 line">
          <label class="span-6" for="txtPassword">Contraseña:</label>
          <input type="password" class="span-6" name="txtPassword" id="txtPassword" title="Por favor ingrese su Contraseña" />
        </div>
        <div class="span-6 line">
          <input type="submit" id="btnSave" name="btnSave" value="Ingresar" />
        </div>
      </div>
    </form>
  </div>
```

1.1.5. Archivo JS

```
$(document).ready(function(){
  /*FORM VALIDATIONS*/
  $("#frmLogin").validate({
    errorLabelContainer: "#alerts",
    wrapper: "li",
    onfocusout: false,
    onkeyup: false,
    rules: {
      txtUsername: {
        required: true
      },
      txtPassword: {
        required: true
      }
    },
    submitHandler: function(form) {
      $("#btnSave").attr('disabled', 'true');
      $("#btnSave").val('Espere por favor...');
      form.submit();
    }
  });
});

//DIALOG FOR VALIDATION ALERTS
$("#dialog_validation").dialog({
  autoOpen: false,
  //height: 200,
  width: 380,
  maxHeight: 200,
  maxWidth: 380,
  modal: true,
  hide: 'slide',
  show: 'slide',
  title: 'Alertas'
});
//END DIALOG FOR VALIDATION

//DIALOG FOR ERROR ALERTS
if($("#has_error").val()){
  $("#dialog_errors").dialog({
    autoOpen: false,
    //height: 200,
    width: 380,
    maxHeight: 200,
    maxWidth: 380,
    modal: true,
    hide: 'slide',
    show: 'slide',
    title: 'Error'
  });
}

$("#dialog_errors").dialog('open');
}
//END DIALOG FOR ERRORS

$("#btnSave").bind('click',function(){
  if($("#frmLogin").valid()){
    $("#btnSave").val('Espere por favor...');
    $("#frmLogin").submit();
  }
  }else{
    $("#dialog_validation").dialog('open');
  }
});
$("#body").bind('keypress',function(e){
  if(e.which == 13){
    $("#btnSave").focus();
    $("#btnSave").click();
  }
});
});
```

1.2. Perfil Administrador

1.2.1. Contenedor.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Admin_Containers_Admin extends Controller_Template {
```

```
    public $template = 'controller/admin/containers/admin';
```

```
    /**
     * El atributo que va a ser utilizado en cada acción
     * NULL por defecto
     * @var View
     */
    public $view;
```

```
    /**
     * @see system/classes/kohana/controller/Kohana_Controller_Template::before()
     */
    public function before()
```

```
    {
        Session::instance();
        //Required because we are extending Controller_Template
        parent::before();
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario', FALSE))->find();
```

```
        if(!Session::instance()->get('treasure_usuario', FALSE) OR Session::instance()->get('treasure_usuario', FALSE)=="")
        OR $user->ID_PERFIL!=1)
```

```
        {
            HTTP::redirect('login');
        }
        //$this->_check_permissions();
```

```
        if ( $this->auto_render )
        {
            //Sets the "Default" title set on the container
            $this->template->title = 'Treasure Administrador';
            $this->template->content = '';
```

```
            $this->template->styles = array();
            $this->template->scripts = array();
        }
```

```
        //Cleans the view so that every action can set it
        //by itself without problems
        $this->view = NULL;
    }
```

```
    /**
     * @see system/classes/kohana/controller/Kohana_Controller_Template::after()
     */
    public function after()
```

```
    {
        if ( $this->auto_render )
        {
            $action = $this->request->action();
            $controller = $this->request->controller();
            $directory = $this->request->directory();
```

```
            //Build the styles array
            $css_file = 'media/css/controller/'.$directory.'/'.$controller.'/'.$action.'.css';
            $styles = array(
                'media/css/screen.css' => 'screen, projection',
                'media/css/src/print.css' => 'print',
                'media/css/jquery-ui/smoothness/jquery-ui-1.8.16.custom.css' => 'all',
                'media/css/dataTables/TableTools.css' => 'screen',
                'media/css/dataTables/TableTools_JUI.css' => 'screen',
                'media/css/dataTables/demo_table_jui.css' => 'screen',
                'media/css/src/ie.css' => 'screen, projection',
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
'media/css/styles.css' => 'screen',
'media/css/menu.css' => 'screen',
    'media/css/menu-style.css' => 'screen',
'media/css/jquery-ui/jquery.autocomplete.css' => 'all',
);

//VALIDATING FILE
if(file_exists($css_file) )
{
    $styles = array_merge( $styles, array($css_file=>'screen') );
}

//Build the scripts array
$js_file = 'media/js/controller/'.$_directory.'/'.$controller.'/'.$action.'.js';
$scripts = array(
    'media/js/jqueryLibrary/jquery-1.6.2.min.js',
    'media/js/jqueryLibrary/jquery-ui-1.8.16.custom.min.js',
    'media/js/jqueryLibrary/jquery-validation.js',
        'media/js/jqueryLibrary/jquery.autocomplete.js',
        'media/js/dataTables/jquery.dataTables.js',
    'media/js/dataTables/TableTools.js',
    'media/js/dataTables/ZeroClipboard.js',
        'media/js/dataTables/dataTables.editor.js',
    'media/js/globalfunctions.js',
    'media/js/ajaxupload.js',
    'media/js/system.js',
    'media/js/mbMenu.js',
);

//VALIDATING FILE
if(file_exists($js_file) )
{
    $scripts = array_merge( $scripts, array($js_file) );
}

//Set styles and scripts on the template (container). This can also be
//done from any action method
$this->template->styles = array_merge( $styles, $this->template->styles );
$this->template->scripts = array_merge( $scripts, $this->template->scripts );

//Get Menu
//Uncomment once we have all the menu logic created
$menu=FALSE;
if($menu){
    $this->template->menu = $menu;
}else{
    $this->template->menu = false;
}

//If $this->view is set display it on the
//containers $content
$this->template->content = $this->view;
}

//Required because we are extending Controller_Template
parent::after();
}

}
```

1.2.2. Contenedor de Vista.

```
<!doctype html>
<html lang="<?php echo l18n::$lang ?>">
  <head>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/x-icon" href="/treasure/media/img/icon.ico">
    <title><?php echo $title ?></title>
    <?php foreach ($styles as $file => $type) echo HTML::style($file, array('media' => $type)), PHP_EOL ?>
    <?php foreach ($scripts as $file) echo HTML::script($file), PHP_EOL ?>
    <script type="text/javascript">
      var document_root = '<?php echo Url::site();?>';
    </script>
  </head>
  <body>
    <div class="container" id="admin_container">
      <div class="span-24 general_bg">
        <div class="span-24 last " style=" padding-top: 5px;">
          <div class="prepend-23 span-1 last admin_banner">
            <strong><a href="<?php echo Url::site();?>" style="text-decoration:none; color: black; font-weight:
normal;">Salir</a></strong>
          </div>
        </div>

        <div class="span-24 ">
          <?php echo Helper_Admin_Menu::menu(); ?>
        </div>
      </div>
      <div class="span-24 content">
        <div class="prepend-2 span-20 append-2 line last" id="container-flash-messages" style=" padding-top:
15px;">
          <?php FlashMessenger::factory()->get_messages(); ?>
        </div>
        <?php echo $content ?>
      </div>
      <div id="spinner" style="display: none;">
        <div id="ajax_loader"></div>
        <div class="ajax_loader_text">Cargando...</div>
      </div>
    </div>
  </body>
</html>
```

1.2.3. Controlador Bodega.

```
<?php
```

```
class Controller_Admin_Bodega extends Controller_Admin_Containers_Admin {  
  
    public function action_index()  
    {  
        HTTP::redirect('admin/bodega/select');  
    }  
  
    public function action_editcontent()  
    {  
        if($this->request->is_ajax())  
        {  
            $pxb_id = arr::get($this->request->post(), 'pxb_id');  
            $cantidad = arr::get($this->request->post(), 'cantidad');  
            try  
            {  
                DB::update('producto_x_bodega')->set(array('CANTIDAD'=>$cantidad))->where('ID_PXB', '=', $pxb_id)  
                    ->execute();  
                $content = (int)$cantidad;  
                echo json_encode($content);  
            }  
            catch(Exception $e)  
            {  
                echo json_encode($e->getMessage());  
            }  
            $this->auto_render = FALSE;  
        }  
        else  
        {  
            HTTP::redirect('admin/main');  
        }  
    }  
  
    public function action_select()  
    {  
        $this->view = View::factory('controller/admin/bodega/select');  
        $productos = ORM::factory('producto')->find_all();  
        $bodega = ORM::factory('productoxbodega')->find_all();  
        $producto = array();  
  
        foreach ($productos as $value) {  
            $producto = $producto + array( $value->ID_PRODUCTO=>$value->DESCRIPCION);  
        }  
  
        if(! count($bodega))  
            $bodega=NULL;  
  
        $this->view->set('bodega', $bodega);  
        $this->view->set('producto', $producto);  
    }  
}  
?>
```

1.2.4. Controlador Departamento.

```
<?php
class Controller_Admin_Departamento extends Controller_Admin_Containers_Admin {

    public function action_index()
    {
        HTTP::redirect('admin/departamento/create/');
    }

    public function action_create()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/admin/departamento/create' );

        //IF USER SUBMITTED FORM
        $ID_DEPARTAMNTO = $this->request->param('params');
        $departamento = ORM::factory('departamento')->where('ID_DEPARTAMENTO', '=', $ID_DEPARTAMNTO)->find();

        //VALIDATE POST
        if( $this->request->post() )
        {

            try
            {
                $values= $this->request->post();

                $departamento->values($values);

                if(!$ID_DEPARTAMNTO)
                    $departamento_nuevo = DB::insert('departamento', array('ID_DEPARTAMENTO', 'NOMBRE',
'DESCRIPCION')->values(array(NULL, $values['NOMBRE'], $values['DESCRIPCION']))->execute();
                else
                    $departamento_nuevo = DB::update('departamento')->set (array('NOMBRE'=>$values['NOMBRE'],
'DESCRIPCION'=>$values['DESCRIPCION']))->where ('ID_DEPARTAMENTO', '=', $departamento-
>ID_DEPARTAMENTO)->execute();

                if($departamento->ID_DEPARTAMENTO)
                    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:departamento:create:update'));
                else
                    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:departamento:create:success'));
            }
            catch(Exception $e)
            {
                echo $e->getMessage();
                FlashMessenger::factory()->set_message('error', $e->getMessage());
            }
            HTTP::redirect('admin/departamento/select/');
        }
        $this->view->set( 'departamento', $departamento );
    }

    public function action_checknombre()
    {
        $this->auto_render=false;
        if ($this->request->is_ajax())
        {
            $ID = arr::get($this->request->post(), 'id');

            $NOMBRE = strtolower(arr::get($this->request->post(), 'NOMBRE'));

            $valid = ORM::factory('departamento')->where(DB::expr("LOWER(NOMBRE)", 'like', $NOMBRE)->find());

            $user_id = $valid->ID_DEPARTAMENTO;

            if($ID!=")
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
{
    if($ID == $user_id)
    {
        echo json_encode(true);
    }
    else
    {
        echo json_encode(false);
    }
}
else
{
    if($user_id==null)
    {
        echo json_encode(true);
    }
    else
    {
        echo json_encode(false);
    }
}
}
}

public function action_select()
{
    $this->view = View::factory('controller/admin/departamento/select');
    $departamentos = ORM::factory('departamento')->find_all();

    if(! $departamentos->count())
        $departamentos=NULL;

    $this->view->set('departamentos', $departamentos);
}
?>
```

1.2.5. Controlador Despacho.

```

<?php
class Controller_Admin_Despacho extends Controller_Admin_Containers_Admin {

    public function action_index()
    {
        HTTP::redirect('admin/despacho/create/');
    }

    public function action_create()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/admin/despacho/create' );

        //IF USER SUBMITTED FORM
        $ID_DESPACHO = $this->request->param('params');
        if($ID_DESPACHO)
            $pedidos = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM `pedido_despacho_x_producto`
            WHERE `ESTADO`='EN PROCESO' AND `ID_DESPACHO` = ".$ID_DESPACHO.">execute()->as_array();
        else
            $pedidos = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM `pedido_despacho_x_producto`
            WHERE `ESTADO`='EN PROCESO' AND `ID_DESPACHO` is NULL")->execute()->as_array();
        $this->view->set( 'pedidos', $pedidos);
        $pedidos_sin_despacho = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM
        `pedido_despacho_x_producto` WHERE `ESTADO`='APROBADO'">execute()->as_array();
        $this->view->set( 'pedidos_sin_despacho', $pedidos_sin_despacho);
        $despacho = ORM::factory('despacho')->where('ID_DESPACHO', '=', $ID_DESPACHO)->find();
        $this->view->set( 'despacho', $despacho );
        $user = ORM::factory('user')->where('ID_USUARIO', '=', $despacho->ID_USUARIO)->find();
        $this->view->set( 'user', $user->ID_USUARIO );
        $validar = count($pedidos);
        $this->view->set( 'validar', $validar );

        //VALIDATE POST
        if( $this->request->post() )
        {

            try
            {

                $values= $this->request->post();
                if($ID_DESPACHO)
                {
                    $Liberapxps = ORM::factory('pedidodespachoxproducto')->where('ID_DESPACHO', '=', $ID_DESPACHO)
                    ->find_all();
                    foreach ($Liberapxps as $Liberapxp) {
                        DB::update('pedido_despacho_x_producto')-
                        >set(array('ID_DESPACHO'=>NULL,'ESTADO'=>'APROBADO'))->where('ID_PDXP', '=', $Liberapxp->ID_PDXP)
                        ->execute();
                    }
                }

                $pedidoProceso = array();
                foreach ($values as $value) {
                    if(strpos($value, 'ID_PEDIDO_')!== false)
                    {
                        $aux = explode('_', $value);
                        $pedidoProceso = $pedidoProceso + array($aux[2]=>$aux[2]);
                    }
                }
                if(!$ID_DESPACHO)
                {
                    $tiempo = array('d'=>(int)date('d'),'m'=>(int)date('m'),'Y'=>(int)date('Y'),'H'=>(int)date('H'),'i'=>(int)date('i'));
                    $despacho->ID_USUARIO = Session::instance()->get('treasure_usuario');
                    $despacho->DIA = $tiempo['d'];
                    $despacho->MES = $tiempo['m'];
                    $despacho->ANIO = $tiempo['Y'];
                    $despacho->HORA = $tiempo['H'];
                    $despacho->MINUTO = $tiempo['i'];
                }
            }
        }
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

$despacho->save();

$pedido2 = ORM::factory('despacho')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))
->and_where('DIA', '=', $tiempo['d'])
->and_where('MES', '=', $tiempo['m'])
->and_where('ANIO', '=', $tiempo['Y'])
->and_where('HORA', '=', $tiempo['H'])
->and_where('MINUTO', '=', $tiempo['i'])
->find();
foreach ($pedidoProceso as $value) {
    DB::update('pedido_despacho_x_producto')->set(array('ID_DESPACHO'=>$pedido2-
>ID_DESPACHO,'ESTADO'=>'EN PROCESO'))->where('ID_PEDIDO', '=', $value)
->execute();
}
}
else
{
    $pedido2 = ORM::factory('despacho')->where('ID_USUARIO', '=', $despacho->ID_USUARIO)
->and_where('DIA', '=', $despacho->DIA)
->and_where('MES', '=', $despacho->MES)
->and_where('ANIO', '=', $despacho->ANIO)
->and_where('HORA', '=', $despacho->HORA)
->and_where('MINUTO', '=', $despacho->MINUTO)
->find();
foreach ($pedidoProceso as $value) {
    DB::update('pedido_despacho_x_producto')->set(array('ID_DESPACHO'=>$pedido2-
>ID_DESPACHO,'ESTADO'=>'EN PROCESO'))->where('ID_PEDIDO', '=', $value)
->execute();
}
}

if($ID_DESPACHO)
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:create:update'));
else
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:create:success'));
}
catch(Exception $e)
{
    echo $e->getMessage();
    FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('admin/despacho/select/');
}
$this->view->set( 'despacho', $despacho );
}

public function action_get_content()
{
    $content="";
    if($this->request->is_ajax())
    {
        $pedido_id = arr::get($this->request->post(), 'pedido_id');
        try
        {
            $closed = ORM::factory('pedidodespachoxproducto')
->where('ID_PEDIDO', '=', $pedido_id)
->find_all()->as_array();
            foreach ($closed as $value) {
                $nombre_producto = ORM::factory('producto')->where('ID_PRODUCTO', '=', $value->ID_PRODUCTO)-
>find();
                $content = $content.'<div class="span-17 line"><div class="span-8 label"> '.$nombre_producto-
>DESCRIPCION.'</div><div class="span-8 label"> Cantidad: '.$value->CANTIDAD.'</div> </div> ';
            }
            echo json_encode($content);
        }
        catch(Exception $e)
        {
            echo json_encode($e->getMessage());
        }
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

        $this->auto_render = FALSE;
    }
    else
    {
        HTTP::redirect('admin/main');
    }
}

public function action_select()
{
    $this->view = View::factory('controller/admin/despacho/select');
    $despachosp = DB::query(Database::SELECT, "SELECT Distinct('ID_DESPACHO') FROM `pedido_despacho_x_producto` WHERE `ESTADO`='EN PROCESO' AND `ID_DESPACHO` is not NULL")->execute()->as_array();
    $despacho = array();
    foreach ($despachosp as $value) {
        $despachos = ORM::factory('despacho')->where('ID_DESPACHO', '=', $value)->find();
        $despacho = $despacho + array($despachos->ID_DESPACHO=>$despachos->as_array());
    }
    $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
    $user = array();
    foreach ($users as $value) {
        $nombres = $value->NOMBRE.'<br>'. $value->APELLIDO;
        $user = $user + array($value->ID_USUARIO=>$nombres);
    }

    if(! count($despacho))
        $despacho=NULL;
    $this->view->set('user', $user);
    $this->view->set('despacho', $despacho);
}

public function action_procesar()
{
    $this->view = View::factory('controller/admin/despacho/procesar');
    $despachosp = DB::query(Database::SELECT, "SELECT Distinct('ID_DESPACHO') FROM `pedido_despacho_x_producto` WHERE `ESTADO` in ('EN PROCESO','COMPRADO','ENVIADO') AND `ID_DESPACHO` is not NULL")->execute()->as_array();
    $despacho = array();
    foreach ($despachosp as $value) {
        $despachos = ORM::factory('despacho')->where('ID_DESPACHO', '=', $value)->find();
        $despacho = $despacho + array($despachos->ID_DESPACHO=>$despachos->as_array());
    }
    $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
    $user = array();
    foreach ($users as $value) {
        $nombres = $value->NOMBRE.'<br>'. $value->APELLIDO;
        $user = $user + array($value->ID_USUARIO=>$nombres);
    }
    if(! count($despacho))
        $despacho=NULL;
    $this->view->set('user', $user);
    $this->view->set('despacho', $despacho);
}

public function action_revisar()
{
    //IF USER SUBMITTED FORM
    $this->view = View::factory('controller/admin/despacho/revisar' );

    //IF USER SUBMITTED FORM
    $ID_DESPACHO = $this->request->param('params');
    $pedidos = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM `pedido_despacho_x_producto` WHERE `ESTADO` in ('EN PROCESO','COMPRADO','ENVIADO') AND `ID_DESPACHO` = ".$ID_DESPACHO."")->execute()->as_array();
    $this->view->set('pedidos', $pedidos);
    $despacho = ORM::factory('despacho')->where('ID_DESPACHO', '=', $ID_DESPACHO)->find();
    $this->view->set('despacho', $despacho );
    $user = ORM::factory('user')->where('ID_USUARIO', '=', $despacho->ID_USUARIO)->find();
    $this->view->set('user', $user->ID_USUARIO );
    $validar = count($pedidos);
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

$this->view->set( 'validar', $validar );

//VALIDATE POST
if( $this->request->post() )
{
    try
    {
        $values= $this->request->post();
        $proceso = 0;
        $pedidoProceso = array();
        echo Debug::vars($values);
        foreach ( $values as $key=>$value ) {
            if(strpos($value, 'Comprar')!= false)
            {
                $proceso = 1;
            }
            if(strpos($value, 'Enviar')!= false)
            {
                $proceso = 2;
            }
            if(strpos($value, 'Entregar')!= false)
            {
                $proceso = 3;
            }
            if(strpos($value, 'Cancelar')!= false)
            {
                $proceso = 4;
            }
            if(strpos($key, 'ID_PEDIDO_')!= false)
            {
                $aux = explode('_', $key);
                $pedidoProceso = $pedidoProceso + array($aux[2]=>$aux[2]);
            }
        }
        switch ( $proceso )
        {
            case 1:foreach ( $pedidoProceso as $value ) {
                DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'COMPRADO'))->where('ID_PEDIDO', '=', $value)
                ->execute();
            }
            FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:comprado'));
            break;
            case 2:foreach ( $pedidoProceso as $value ) {
                DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'ENVIADO'))->where('ID_PEDIDO', '=', $value)
                ->execute();
            }
            FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:enviado'));
            break;
            case 3:foreach ( $pedidoProceso as $value ) {
                $producto = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $value)->find_all();
                foreach ( $producto as $value2 ) {
                    $cantidad = ORM::factory('productoxbodega')->where('ID_PRODUCTO', '=', $value2-
>ID_PRODUCTO)->find();
                    $valor = $cantidad->CANTIDAD + $value2->CANTIDAD;
                    DB::update('producto_x_bodega')->set(array('CANTIDAD'=>$valor,))->where('ID_PRODUCTO',
'=', $value2->ID_PRODUCTO)->execute();
                }
                DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'ENTREGADO'))->where('ID_PEDIDO', '=', $value)
                ->execute();
            }
            FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:entregado'));
            break;
            case 4:foreach ( $pedidoProceso as $value ) {

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'CANCELADO'))->where('ID_PEDIDO', '=', $value)
->execute();
}
FlashMessenger::factory()->set_message('error', Kohana::message('admin',
'admin:despacho:estado:cancelado'));
break;
default : HTTP::redirect('admin/main');
break;
}
}
catch(Exception $e)
{
echo $e->getMessage();
FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('admin/despacho/procesar/');
}
$this->view->set( 'despacho', $despacho );
}
?>
```

1.2.6. Controlador Main.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Admin_Main extends Controller_Admin_Containers_Admin {
```

```
    public function action_index()
```

```
    {
```

```
        $option = $this->request->param('params');
```

```
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))->find();
```

```
        $name = ".strtoupper ($user->NOMBRE)";
```

```
        $this->view = View::factory( 'controller/admin/main' );
```

```
        $this->view->set('name',$name);
```

```
    }
```

```
}
```

1.2.7. Controlador Pedido.

```

<?php
class Controller_Admin_Pedido extends Controller_Admin_Containers_Admin {

    public function action_index()
    {
        HTTP::redirect('admin/pedido/create/');
    }

    public function action_create()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/admin/pedido/create' );

        //IF USER SUBMITTED FORM
        $ID_PEDIDO = $this->request->param('params');
        $pedido = ORM::factory('pedido')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find();
        $producto = ORM::factory('producto')->order_by('DESCRIPCION')->find_all()->as_array('ID_PRODUCTO',
'DESCRIPCION');
        $this->view->set( 'producto', $producto );
        $user = ORM::factory('user')->where('ID_USUARIO', '=', $pedido->ID_USUARIO)->find();
        $this->view->set( 'user', $user->ID_USUARIO );
        if($ID_PEDIDO)
        {
            $PXP = ORM::factory('pedidodespachoproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find_all()-
>as_array('ID_PRODUCTO', 'CANTIDAD' );
            $productos = count($PXP);
        }
        else
        {
            $PXP = "";
            $productos = 0;
        }
        $this->view->set( 'PXP', $PXP );
        $this->view->set( 'productos', $productos );

        //VALIDATE POST
        if( $this->request->post() )
        {
            try
            {
                $values= $this->request->post();
                if($ID_PEDIDO)
                {
                    $Eliminapxs = ORM::factory('pedidodespachoproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)
->find_all();
                    foreach ($Eliminapxs as $Eliminapxp) {
                        DB::delete('pedido_despacho_x_producto')->where('ID_PDXP', '=', $Eliminapxp->ID_PDXP)
->execute(Database::instance());
                    }
                }

                $productosIngreso = array();
                foreach ($values as $key => $value) {
                    if(strpos($key, 'CANTIDAD_')== false)
                    {
                        $aux = explode('_', $key);
                        $productosIngreso = $productosIngreso + array($aux[1]==>$value);
                    }
                }
                if(!$ID_PEDIDO)
                {
                    $tiempo = array('d'=>(int)date('d'),'m'=>(int)date('m'),'Y'=>(int)date('Y'),'H'=>(int)date('H'),'i'=>(int)date('i'));
                    $pedido->ID_USUARIO = Session::instance()->get('treasure_usuario');
                    $pedido->DIA = $tiempo['d'];
                }
            }
        }
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
$pedido->MES = $tiempo['m'];
$pedido->ANIO = $tiempo['Y'];
$pedido->HORA = $tiempo['H'];
$pedido->MINUTO = $tiempo['i'];
$pedido->save();
```

```
$pedido2 = ORM::factory('pedido')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))
->and_where('DIA', '=', $tiempo['d'])
->and_where('MES', '=', $tiempo['m'])
->and_where('ANIO', '=', $tiempo['Y'])
->and_where('HORA', '=', $tiempo['H'])
->and_where('MINUTO', '=', $tiempo['i'])
->find();
foreach ($productosIngreso as $key=>$value) {
    $PXP2 = ORM::factory('pedidodespachoxproducto');
    $PXP2->ID_PEDIDO = $pedido2->ID_PEDIDO;
    $PXP2->ID_PRODUCTO = $key;
    $PXP2->CANTIDAD = $value;
    $PXP2->ESTADO = 'PEDIDO';
    $PXP2->save();
}
}
else
{
    $pedido2 = ORM::factory('pedido')->where('ID_USUARIO', '=', $pedido->ID_USUARIO)
->and_where('DIA', '=', $pedido->DIA)
->and_where('MES', '=', $pedido->MES)
->and_where('ANIO', '=', $pedido->ANIO)
->and_where('HORA', '=', $pedido->HORA)
->and_where('MINUTO', '=', $pedido->MINUTO)
->find();
    foreach ($productosIngreso as $key=>$value) {
        $PXP2 = ORM::factory('pedidodespachoxproducto');
        $PXP2->ID_PEDIDO = $pedido2->ID_PEDIDO;
        $PXP2->ID_PRODUCTO = $key;
        $PXP2->CANTIDAD = $value;
        $PXP2->ESTADO = 'PEDIDO';
        $PXP2->save();
    }
}
}
```

```
if($ID_PEDIDO)
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:pedido:create:update'));
else
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:pedido:create:success'));
}
catch(Exception $e)
{
    echo $e->getMessage();
    FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('admin/pedido/select');
}
$this->view->set('pedido', $pedido);
}
```

```
public function action_select()
{
    $this->view = View::factory('controller/admin/pedido/select');
    $pedidosp = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM
'pedido_despacho_x_producto' WHERE 'ESTADO'='PEDIDO' AND 'ID_DESPACHO' is NULL")->execute()->as_array();
    $pedido = array();
    foreach ($pedidosp as $value) {
        $pedidos = ORM::factory('pedido')->where('ID_PEDIDO', '=', $value)->find();
        $pedido = $pedido + array($pedidos->ID_PEDIDO=>$pedidos->as_array());
    }
    $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
    $user = array();
    foreach ($users as $value) {
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

        $nombres = $value->NOMBRE.'  
'; $value->APELLIDO;
        $user = $user + array($value->ID_USUARIO=>$nombres);
    }

    if(! count($pedido))
        $pedido=NULL;
    $this->view->set('user', $user);
    $this->view->set('pedido', $pedido);
}

public function action_aprobar()
{
    $this->view = View::factory('controller/admin/pedido/aprobar');
    $pedidosp = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM
    'pedido_despacho_x_producto' WHERE 'ESTADO'='PEDIDO' AND 'ID_DESPACHO' is NULL")->execute()->as_array();
    $pedido = array();
    foreach ($pedidosp as $value) {
        $pedidos = ORM::factory('pedido')->where('ID_PEDIDO', '=', $value)->find();
        $pedido = $pedido + array($pedidos->ID_PEDIDO=>$pedidos->as_array());
    }
    $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
    $user = array();
    foreach ($users as $value) {
        $nombres = $value->NOMBRE.'  
'; $value->APELLIDO;
        $user = $user + array($value->ID_USUARIO=>$nombres);
    }

    if(! count($pedido))
        $pedido=NULL;
    $this->view->set('user', $user);
    $this->view->set('pedido', $pedido);
}

public function action_revisar()
{
    //IF USER SUBMITTED FORM
    $this->view = View::factory( 'controller/admin/pedido/revisar' );

    //IF USER SUBMITTED FORM
    $ID_PEDIDO = $this->request->param('params');
    $pedido = ORM::factory('pedido')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find();
    $producto = ORM::factory('producto')->order_by('DESCRIPCION')->find_all()->as_array('ID_PRODUCTO',
'DESCRIPCION');
    $this->view->set( 'producto', $producto );
    if($ID_PEDIDO)
    {
        $PXP = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find_all()-
>as_array('ID_PRODUCTO', 'CANTIDAD' );
    }
    else
    {
        $PXP = "";
    }
    $this->view->set( 'PXP', $PXP );

    //VALIDATE POST
    if( $this->request->post() )
    {
        try
        {
            $values= $this->request->post();
            if(isset($values['btnCancel']))
            {
                $pedidos = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)-
>and_where('ESTADO', '=', 'PEDIDO')->find_all();
                foreach ($pedidos as $value) {

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
        DB::update('pedido_despacho_x_producto')->set(array('ESTADO' => 'CANCELADO'))-
>where('ID_PDXP', '=', $value->ID_PDXP)->execute();
    }
    FlashMessenger::factory()->set_message('error', 'Se ha Cancelado el Pedido exitosamente.');
```

```
    }
    else
    {
        $pedidos = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)-
>and_where('ESTADO', '=', 'PEDIDO')->find_all();
        foreach ($pedidos as $value) {
            DB::update('pedido_despacho_x_producto')->set(array('ESTADO' => 'APROBADO'))->where('ID_PDXP',
'=', $value->ID_PDXP)->execute();
        }
        FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:pedido:revisar:aprobado')));
    }
}
}
catch(Exception $e)
{
    echo $e->getMessage();
    FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('admin/pedido/aprobar/');
}
$this->view->set('pedido', $pedido );
}
}
?>
```

1.2.8. Controlador Producto.

```
<?php
class Controller_Admin_Producto extends Controller_Admin_Containers_Admin {

    public function action_index()
    {
        HTTP::redirect('admin/producto/create/');
    }

    public function action_create()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/admin/producto/create' );

        //IF USER SUBMITTED FORM
        $ID_PRODUCTO = $this->request->param('params');
        $producto = ORM::factory('producto')->where('ID_PRODUCTO', '=', $ID_PRODUCTO)->find();
        $this->view->set('producto', $producto);
        $departamento = ORM::factory('departamento')->order_by('NOMBRE')->find_all()->as_array('ID_DEPARTAMENTO',
'NOMBRE');
        $this->view->set('departamento', $departamento);

        //VALIDATE POST
        if( $this->request->post() )
        {
            try
            {
                $values= $this->request->post();

                $producto->values($values);

                if(!$ID_PRODUCTO)
                {
                    $producto_nuevo = DB::insert('producto', array('ID_PRODUCTO', 'ID_DEPARTAMENTO',
'DESCRIPCION')->values(array(NULL, $values['ID_DEPARTAMENTO'], $values['DESCRIPCION'])->execute());
                    $aux = mysql_insert_id();
                    $producto_nuevo = DB::insert('producto_x_bodega', array('ID_PXB', 'ID_PRODUCTO', 'ID_BODEGA',
'CANTIDAD')->values(array(NULL, $aux, 1, 0))->execute());
                }
                else
                {
                    $producto_nuevo = DB::update('producto')->set
(array('ID_DEPARTAMENTO'=>$values['ID_DEPARTAMENTO'], 'DESCRIPCION'=>$values['DESCRIPCION'])->where
('ID_PRODUCTO', '=', $producto->ID_PRODUCTO)->execute());
                }

                if($ID_PRODUCTO)
                    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:producto:create:update'));
                else
                    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:producto:create:success'));
            }
            catch(Exception $e)
            {
                echo $e->getMessage();
                FlashMessenger::factory()->set_message('error', $e->getMessage());
            }
            HTTP::redirect('admin/producto/select/');
        }
        $this->view->set('producto', $producto );
    }

    public function action_checkdescripcion()
    {
        $this->auto_render=false;
        if ($this->request->is_ajax())
        {
            $ID = arr::get($this->request->post(), 'id');
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
$DESCRIPCION = arr::get($this->request->post(), 'DESCRIPCION');  
$DESCRIPCION = strtolower($DESCRIPCION);
```

```
$valid = ORM::factory('producto')->where(DB::expr("LOWER(DESCRIPCION)", '=', $DESCRIPCION)->find();
```

```
$producto_id = $valid->ID_PRODUCTO;
```

```
if($ID!="")  
{  
    if($ID == $producto_id)  
    {  
        echo json_encode(true);  
    }  
    else  
    {  
        if($producto_id==null)  
        {  
            echo json_encode(true);  
        }  
        else  
        {  
            echo json_encode(false);  
        }  
    }  
}  
else  
{  
    if($producto_id==null)  
    {  
        echo json_encode(true);  
    }  
    else  
    {  
        echo json_encode(false);  
    }  
}  
}  
}  
  
public function action_select()  
{  
    $this->view = View::factory('controller/admin/producto/select');  
    $productos = ORM::factory('producto')->find_all();  
    $departamento = ORM::factory('departamento')->order_by('NOMBRE')->find_all()-  
>as_array('ID_DEPARTAMENTO', 'NOMBRE');  
    $this->view->set('departamento', $departamento);  
  
    if(! $productos->count())  
        $productos=NULL;  
  
    $this->view->set('productos', $productos);  
}  
}  
?>
```

1.2.9. Controlador Usuario.

```

<?php
class Controller_Admin_User extends Controller_Admin_Containers_Admin {

    public function action_index()
    {
        HTTP::redirect('admin/user/create/');
    }

    public function action_create()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/admin/user/create ');

        //IF USER SUBMITTED FORM
        $user_id = $this->request->param('params');
        $user = ORM::factory('user')->where('ID_USUARIO', '=', $user_id)->find();

        //VARIABLES FOR GET THE PROFILE OF USER FROM TABLE PERFILES
        $perfil = ORM::factory('perfil')->order_by('NOMBRE')->find_all()->as_array('ID_PERFIL', 'NOMBRE');
        $this->view->set('perfil', $perfil);

        //VARIABLES PARA OBTENER LOS DEPARTAMENTOS DEL USUARIO
        $departamento = ORM::factory('departamento')->order_by('NOMBRE')->find_all()->as_array('ID_DEPARTAMENTO', 'NOMBRE');
        $this->view->set('departamento', $departamento);

        //VALIDATE POST
        if( $this->request->post() )
        {

            try
            {

                $values= $this->request->post();

                if($values['CONTRASENIA']==='')
                    $values[ 'CONTRASENIA' ] = $user->CONTRASENIA;
                else
                    $values[ 'CONTRASENIA' ] = md5($values[ 'CONTRASENIA' ]);
                $user->values($values);
                if(strlen($values['ID_DEPARTAMENTO'])==0)
                    $values['ID_DEPARTAMENTO']=NULL;

                if(!$user_id)
                    $usuario_nuevo = DB::insert('usuario', array('ID_USUARIO', 'ID_PERFIL', 'ID_DEPARTAMENTO', 'NOMBRE', 'APELLIDO', 'USUARIO', 'CONTRASENIA', 'ESTADO'))->values(array(NULL, $values['ID_PERFIL'], $values['ID_DEPARTAMENTO'], $values['NOMBRE'], $values['APELLIDO'], $values['USUARIO'], $values['CONTRASENIA'], $values['ESTADO']))->execute();
                else
                    $usuario_nuevo = DB::update('usuario')->set( array('ID_PERFIL'=>$values['ID_PERFIL'], 'ID_DEPARTAMENTO'=>$values['ID_DEPARTAMENTO'], 'NOMBRE'=>$values['NOMBRE'], 'APELLIDO'=>$values['APELLIDO'], 'USUARIO'=>$values['USUARIO'], 'CONTRASENIA'=>$values['CONTRASENIA'], 'ESTADO'=>$values['ESTADO']))->where ('ID_USUARIO', '=', $user->ID_USUARIO)->execute ();

                if($user_id)
                    FlashMessenger::factory()->set_message('success', Kohana::message('admin', 'admin:user:create:update'));
                else
                    FlashMessenger::factory()->set_message('success', Kohana::message('admin', 'admin:user:create:success'));

            }
            catch(Exception $e)
            {
                echo $e->getMessage();
                FlashMessenger::factory()->set_message('error', $e->getMessage());
            }
        }
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
        HTTP::redirect('admin/user/select/');
    }
    $this->view->set('user', $user);
}

public function action_checkuser()
{
    $this->auto_render=false;
    if ($this->request->is_ajax())
    {
        $id = arr::get($this->request->post(), 'id');
        $usuario = arr::get($this->request->post(), 'USUARIO');

        $usuario = strtolower($usuario);

        $valid = ORM::factory('user')->where(DB::expr("LOWER(USUARIO)", 'LIKE', $usuario)->find();

        $user_id = $valid->ID_USUARIO;

        if($id!="")
        {
            if($id == $user_id)
            {
                echo json_encode(true);
            }
            else
            {
                echo json_encode(false);
            }
        }
        else
        {
            if($user_id==null)
            {
                echo json_encode(true);
            }
            else
            {
                echo json_encode(false);
            }
        }
    }
}

public function action_select()
{
    $this->view = View::factory('controller/admin/user/select');
    $users = ORM::factory('user')->find_all();
    $perfiles = ORM::factory('perfil')->find_all();
    $departamentos = ORM::factory('departamento')->find_all()->as_array();
    $perfiles_array = array();
    $departamentos_array = array(NULL=>'NINGUNO');
    foreach ($perfiles as $perfil)
        $perfiles_array[$perfil->ID_PERFIL] = $perfil->NOMBRE;
    foreach ($departamentos as $departamento)
        $departamentos_array[$departamento->ID_DEPARTAMENTO] = $departamento->NOMBRE;

    if(!$users->count())
        $users=NULL;

    $this->view->set('users', $users);
    $this->view->set('perfiles_array', $perfiles_array);
    $this->view->set('departamentos_array', $departamentos_array);
}
?>
```

1.2.10. Helper

1.2.10.1. Menú.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Helper_Admin_Menu {  
    public static function menu($seg1 = null, $seg2 = null, $seg3 = null)  
    {  
        $items = array();  
  
        /*USUARIO*/  
        $items[] = array('id' => 1, 'parent_id' => 0, 'title' => 'Usuario', 'link' => '#');  
        $items[] = array('id' => 2, 'parent_id' => 1, 'title' => 'Nuevo', 'link' => 'admin/user/create');  
        $items[] = array('id' => 3, 'parent_id' => 1, 'title' => 'Modificar', 'link' => 'admin/user/select');  
  
        /*DEPARTAMENTOS*/  
        $items[] = array('id' => 4, 'parent_id' => 0, 'title' => 'Departamentos', 'link' => '#');  
        $items[] = array('id' => 5, 'parent_id' => 4, 'title' => 'Nuevo', 'link' => 'admin/departamento/create');  
        $items[] = array('id' => 6, 'parent_id' => 4, 'title' => 'Modificar', 'link' => 'admin/departamento/select');  
  
        /*PRODUCTOS*/  
        $items[] = array('id' => 7, 'parent_id' => 0, 'title' => 'Productos', 'link' => '#');  
        $items[] = array('id' => 8, 'parent_id' => 7, 'title' => 'Nuevo', 'link' => 'admin/producto/create');  
        $items[] = array('id' => 9, 'parent_id' => 7, 'title' => 'Modificar', 'link' => 'admin/producto/select');  
  
        /*PEDIDOS*/  
        $items[] = array('id' => 10, 'parent_id' => 0, 'title' => 'Pedidos', 'link' => '#');  
        $items[] = array('id' => 11, 'parent_id' => 10, 'title' => 'Nuevo', 'link' => 'admin/pedido/create');  
        $items[] = array('id' => 12, 'parent_id' => 10, 'title' => 'Modificar', 'link' => 'admin/pedido/select');  
        $items[] = array('id' => 13, 'parent_id' => 10, 'title' => 'Aprobar', 'link' => 'admin/pedido/aprobar');  
  
        /*DESPACHOS*/  
        $items[] = array('id' => 14, 'parent_id' => 0, 'title' => 'Despachos', 'link' => '#');  
        $items[] = array('id' => 15, 'parent_id' => 14, 'title' => 'Nuevo', 'link' => 'admin/despacho/create');  
        $items[] = array('id' => 16, 'parent_id' => 14, 'title' => 'Modificar', 'link' => 'admin/despacho/select');  
        $items[] = array('id' => 17, 'parent_id' => 14, 'title' => 'Procesar', 'link' => 'admin/despacho/procesar');  
  
        /*BODEGA*/  
        $items[] = array('id' => 20, 'parent_id' => 0, 'title' => 'Bodega', 'link' => '#');  
        $items[] = array('id' => 21, 'parent_id' => 20, 'title' => 'Manejo', 'link' => 'admin/bodega/select');  
  
        /*REPORTES*/  
        $items[] = array('id' => 22, 'parent_id' => 0, 'title' => 'Reportes', 'link' => '#');  
        $items[] = array('id' => 23, 'parent_id' => 22, 'title' => 'Pedidos', 'link' => 'admin/reporte/pedido');  
        $items[] = array('id' => 24, 'parent_id' => 22, 'title' => 'Despachos', 'link' => 'admin/reporte/despacho');  
        $items[] = array('id' => 25, 'parent_id' => 22, 'title' => 'Productos', 'link' => 'admin/reporte/producto');  
        $items[] = array('id' => 26, 'parent_id' => 22, 'title' => 'Departamentos', 'link' => 'admin/reporte/departamento');  
  
        // Build the current path  
        $current = "";  
        $current .= $seg1 ? $seg1 : "";  
        $current .= $seg2 ? ('/' . $seg2): "";  
        $current .= $seg3 ? ('/' . $seg3): "";  
  
        // Let's get the body of the current page  
        //$body = ORM::factory('entry')->where('link','=', $current)->find()->body;  
        $options = array('link_prepend' => "", 'current_path' => $current);  
  
        return MenuStructure::factory($items, $options)->get_menu();  
    }  
}
```

??

1.2.11. Vistas de Bodega.

1.2.11.1. Select.

```
<div class="span-24">
  <br><br>
  <div class="title">Bodega</div>
  <?php if($bodega): ?>
  <div class="span-24"><br>
    <table id="tblBodega" >
      <thead>
        <tr>
          <th class="span-3">PRODUCTO</th>
          <th class="span-3">CANTIDAD</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($bodega as $bod): ?>
          <tr>
            <td><?php echo $producto[$bod->ID_PRODUCTO] ?></td>
            <td id="<?php echo $bod->ID_PXB ?>">
              <span id="cantidad_<?php echo $bod->ID_PXB; ?>" class="text" style="display: inline;"><?php
echo $bod->CANTIDAD; ?></span>
              <input type="text" value="<?php echo $bod->CANTIDAD; ?>" class="editbox" style="display:
none;" id="cantidad_input_<?php echo $bod->ID_PXB; ?>"
            </td>
          </tr>
        </tr>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br/>
  <div class="alerts" style="text-align: center">
    No existen Productos registrados en bodega.
  </div>
  </div>
  <?php endif: ?>
</div>
```

1.2.12. Vistas de Departamento.

1.2.12.1. Create.

```
<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title"><?php echo ($departamento->ID_DEPARTAMENTO)?'Actualizar:'Nuevo'; ?> Departamento</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>

    <div class="span-10 prepend-5 last">
      <ul id="alerts" class="alerts"></ul>
    </div>

    <form action="" method="post" name="frmDepartamento" id="frmDepartamento"><br/><br/>
    <div class="span-20 line">
      <div class="prepend-5 span-4 label">
        * Nombre:
      </div>
      <div class="span-8 label">
        <input class="span-8" type="text" name="NOMBRE" id="NOMBRE" value="<?php echo $departamento-
>NOMBRE; ?>"/>
      </div>
    </div>

    <div class="span-20 line" id="passContainer">
      <div class="prepend-5 span-4 label">
        * Descripcion:
      </div>
      <div class="span-8 label">
        <textarea class="span-8" name="DESCRIPCION" id="DESCRIPCION"><?php echo str_replace("\n", "\n",
$departamento->DESCRIPCION); ?></textarea>
      </div>
    </div>
    <input type="hidden" name="ID_DEPARTAMENTO" id="ID_DEPARTAMENTO" value="<?php echo
$departamento->ID_DEPARTAMENTO; ?>"/>
    <input type="hidden" name="nickname" id="nickname" value=""/>
    <div class="span-20 line" style="text-align: center;"><br/>
    <input id="btnSave" type="submit" value="Guardar"/>
  </div>
</form>
</div>
</div>
```

1.2.12.2. *Select.*

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Departamento</div>
  <?php if($departamentos): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/departamento/create');?>">Crear Departamento</a><br><br>
    <table id="tblDepartamentos">
      <thead>
        <tr>
          <th class="span-3">NOMBRE</th>
          <th class="span-3">DESCRIPCION</th>
          <th class="span-1">EDITAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($departamentos as $departamento): ?>
          <tr>
            <td><?php echo $departamento->NOMBRE ?></td>
            <td><?php echo $departamento->DESCRIPCION ?></td>
            <td>
              <a href="<?php echo URL::site('admin/departamento/create/'.$departamento->ID_DEPARTAMENTO);?>">
                </img>
              </a>
            </td>
          </tr>
        <?php endforeach; ?>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Departamentos registrados.
    </div>
  </div>
  <?php endif; ?>
</div>
```

1.2.13. Vistas de Despacho.

1.2.13.1. Create.

```
<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title"><?php echo ($despacho->ID_DESPACHO)?'Actualizar':'Nuevo'; ?> Despacho</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>
    <div class="prepend-5 span-10" align="center">
      <div id="dialog_errores">
        <ul id="alerts" class="alerts"></ul>
      </div>
    </div>
    <div id="dialog" title="">
      <p id="pedido_productos">
        <div id="contenido"></div>
      </p>
    </div><br/><br/>
    <form action="" method="post" name="frmDespacho" id="frmDespacho">
      <?php if($despacho->ID_DESPACHO): ?>
      <?php foreach ($pedidos as $value): ?>
        <div class="span-20 line ">
          <div class="prepend-2 span-5 label"> N&Uacute;mero de Pedido: </div>
          <div class="prepend-2 span-10">
            <?php echo Form::checkbox("$value['ID_PEDIDO'].", 'ID_PEDIDO_'.$value['ID_PEDIDO'].",
FALSE, array('id'=>'ID_PEDIDO_'.$value['ID_PEDIDO'].", 'class' => 'span-5', 'checked' => 'checked'));?>
            <div class="span-3 label"> <?php echo $value['ID_PEDIDO']?> </div>
            <a class='des-pedido-link' pedido_id='<?php echo $value['ID_PEDIDO']?>' href="#">Ver
Productos</a>
          </div>
        </div>
      <?php endforeach; ?>
      <?php foreach ($pedidos_sin_despacho as $value): ?>
        <div class="span-20 line ">
          <div class="prepend-2 span-5 label"> N&Uacute;mero de Pedido: </div>
          <div class="prepend-2 span-10">
            <?php echo Form::checkbox("$value['ID_PEDIDO'].", 'ID_PEDIDO_'.$value['ID_PEDIDO'].",
FALSE, array('id'=>'ID_PEDIDO_'.$value['ID_PEDIDO'].", 'class' => 'span-5'));?>
            <div class="span-3 label"> <?php echo $value['ID_PEDIDO']?> </div>
            <a class='des-pedido-link' pedido_id='<?php echo $value['ID_PEDIDO']?>' href="#">Ver
Productos</a>
          </div>
        </div>
      <?php endforeach; ?>
      <?php else: ?>
      <?php foreach ($pedidos_sin_despacho as $value): ?>
        <div class="span-20 line ">
          <div class="prepend-2 span-5 label"> N&Uacute;mero de Pedido: </div>
          <div class="prepend-2 span-10">
            <?php echo Form::checkbox("$value['ID_PEDIDO'].", 'ID_PEDIDO_'.$value['ID_PEDIDO'].",
FALSE, array('id'=>'ID_PEDIDO_'.$value['ID_PEDIDO'].", 'class' => 'span-5'));?>
            <div class="span-3 label"> <?php echo $value['ID_PEDIDO']?> </div>
            <a class='des-pedido-link' pedido_id='<?php echo $value['ID_PEDIDO']?>' href="#">Ver
Productos</a>
          </div>
        </div>
      <?php endforeach; ?>
      <?php endif; ?>
      <input type="hidden" name="validar" id="validar" value="<?php echo ($validar)?$validar:0;?>" />
      <input type="hidden" name="user" id="user" value="<?php echo ($user)?$user:";?>" />
      <div class="span-20 " align="center"><br>
      <input type="submit" id="btnSave" name="btnSave" value="Guardar" />
    </div>
  </form>
</div>
</div>
```

1.2.13.2. Procesar.

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Despacho Para Procesar</div>
  <?php if($despacho): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/despacho/create');?>">Crear Despacho</a><br><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">N&Uacute;MERO DE DESPACHO</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">REVISAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($despacho as $desp): ?>
          <tr>
            <td><?php echo $desp["ID_DESPACHO"] ?></td>
            <td><?php echo $user[$desp["ID_USUARIO"]]?></td>
            <td><?php echo $desp["ANIO"]?></td>
            <td><?php echo $desp["MES"]?></td>
            <td><?php echo $desp["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('admin/despacho/revisar/'.$desp["ID_DESPACHO"]);?>">
                </img>
              </a>
            </td>
          </tr>
        </tr>
      </tbody>
    </table>
  </div>
  <?php elseif: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Despachos registrados.
    </div>
  </div>
  <?php endif; ?>
</div>
```

1.2.13.3. Revisar.

```

<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title">Actualizacion de Estado de Despacho</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>

    <div class="prepend-5 span-10" align="center">
      <div id="dialog_errores">
        <ul id="alerts" class="alerts"></ul>
      </div>
    </div>

    <div id="dialog" title="">
      <p id="pedido_productos">
        <div id="contenido"></div>
      </p>
    </div>

  <br/><br/><br/>

  <form action="" method="post" name="frmDespacho" id="frmDespacho">
    <?php if($despacho->ID_DESPACHO): ?>
      <?php foreach ($pedidos as $value): ?>
        <div class="span-20 line ">
          <div class="prepend-2 span-5 label"> N&Uacute;mero DE PEDIDO: </div>
          <div class="prepend-2 span-10">
            <?php echo Form::checkbox("$value['ID_PEDIDO'].", 'ID_PEDIDO_'.$value['ID_PEDIDO'].",
FALSE, array('id'=>'ID_PEDIDO_'.$value['ID_PEDIDO'].', 'class' => 'span-5', 'checked' =>
'checked', 'disabled'=>'disable')));?>
            <div class="span-3 label"> <?php echo $value['ID_PEDIDO']?> </div>
            <a class='des-pedido-link' pedido_id='<?php echo $value['ID_PEDIDO']?>' href='>Ver
Productos</a>
            <input type="hidden" name="<?php echo 'ID_PEDIDO_'.$value['ID_PEDIDO'].'?>" id="<?php
echo 'ID_PEDIDO_'.$value['ID_PEDIDO'].'?>" value="<?php echo $value['ID_PEDIDO'].'?>" />
          </div>
        </div>
      <?php endforeach; ?>
    <?php endif; ?>
    <div class="span-20 " align="center"><br>
      <input type="submit" id="btn_Comprar" name="btn_Comprar" value="Comprar" />&emsp;&emsp;
      <input type="submit" id="btn_Enviar" name="btn_Enviar" value="Enviar" />&emsp;&emsp;
      <input type="submit" id="btn_Entregar" name="btn_Entregar" value="Entregar" />&emsp;&emsp;
      <input type="submit" id="btn_Cancelar" name="btn_Cancelar" value="Cancelar" />
    </div>
  </form>
  <div class="span-20 " align="center"><br>
    <a href="javascript:window.history.go(-1);">Regresar</a>
  </div>
</div>
</div>

```

1.2.13.4. *Select.*

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Despacho Para Modificar</div>
  <?php if($despacho): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/despacho/create');?>">Crear Despacho</a><br><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">N&Uacute;MERO DE DESPACHO</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">EDITAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($despacho as $desp): ?>
          <tr>
            <td><?php echo $desp["ID_DESPACHO"] ?></td>
            <td><?php echo $user[$desp["ID_USUARIO"]]?></td>
            <td><?php echo $desp["ANIO"]?></td>
            <td><?php echo $desp["MES"]?></td>
            <td><?php echo $desp["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('admin/despacho/create/'.$desp["ID_DESPACHO"]);?>">
                </img>
              </a>
            </td>
          </tr>
        </tr>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Despachos registrados.
    </div>
  </div>
  <?php endif; ?>
</div>
```

1.2.14. Vistas de Main.

1.2.14.1. Main.

```
<div class="span-24">
  <div class="prepend-2 span-20">

    <div class="prepend-5 spen-20 title_main_page center">
      PAGINA DE ADMINISTRADOR<br><br>
      BIENVENIDO <?php echo $name?>
    </div>

  </div>
</div>
```

1.2.15. Vistas de Pedido.

1.2.15.1. Aprobar.

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Pedido Para Aprobaci&oaacute;n</div>
  <?php if($pedido): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/pedido/create');?>">Crear Pedido</a><br><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">N&Uacute;MERO DE PEDIDO</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">REVISAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($pedido as $ped): ?>
          <tr>
            <td><?php echo $ped["ID_PEDIDO"] ?></td>
            <td><?php echo $user[$ped["ID_USUARIO"]]?></td>
            <td><?php echo $ped["ANIO"]?></td>
            <td><?php echo $ped["MES"]?></td>
            <td><?php echo $ped["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('admin/pedido/revisar/'.$ped["ID_PEDIDO"]); ?>">
                </img>
              </a>
            </td>
          </tr>
        </tr>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Pedidos para procesar.
    </div>
  </div>
  <?php endif: ?>
</div>
```

1.2.15.2. Create.

```

<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title"><?php echo ($pedido->ID_PEDIDO)?'Actualizar':'Nuevo'; ?> Pedido</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>
    <div class="prepend-5 span-10" align="center">
      <div id="dialog_errores">
        <ul id="alerts" class="alerts"></ul>
      </div>
    </div><br/><br/>
    <form action="" method="post" name="frmPedido" id="frmPedido">
      <?php if($pedido->ID_PEDIDO): ?>
        <?php foreach ($producto as $key => $value): ?>
          <div class="span-20 line ">
            <div class="span-10">
              <?php if(array_key_exists($key,$PXP)): ?>
                <?php echo Form::checkbox(".$key.", $key, FALSE,
array('id'=>'ID_PRODUCTO_'. $key.', 'class' => 'span-5', 'checked' => 'checked' ));?>
                <?php else: ?>
                <?php echo Form::checkbox(".$key.", $key, FALSE,
array('id'=>'ID_PRODUCTO_'. $key.', 'class' => 'span-5'));?>
                <?php endif; ?>
                <div class="span-5 label"> <?php echo $value?> </div>
            </div>
            <div class="span-10 last">
              <div class="prepend-2 span-3 label"> *Cantidad: </div>
              <?php if(array_key_exists($key,$PXP)): ?>
                <input type="text" class="span-1 last" name="CANTIDAD_<?php
echo $key?>" id='CANTIDAD_<?php echo $key?>' value="<?php echo $PXP[$key]?>" title='El campo
"Cantidad" es obligatorio.' maxlength="2"/>
                <?php else: ?>
                <input type="text" class="span-1 last" name="CANTIDAD_<?php
echo $key?>" id='CANTIDAD_<?php echo $key?>' value="" title='El campo "Cantidad" es obligatorio.'
disabled="disable" maxlength="2"/>
                <?php endif; ?>
                <input type="hidden" name="<?php echo $key?>" id="<?php echo $key?>"
value="<?php echo $key?>"/>
            </div>
          </div>
        <?php endforeach; ?>
        <?php else: ?>
        <?php foreach ($producto as $key => $value): ?>
          <div class="span-20 line ">
            <div class="span-10">
              <?php echo Form::checkbox(".$key.", $key, FALSE,
array('id'=>'ID_PRODUCTO_'. $key.', 'class' => 'span-5'));?>
            </div>
            <div class="span-5 label"> <?php echo $value?> </div>
          </div>
          <div class="span-10 last">
            <div class="prepend-2 span-3 label"> *Cantidad: </div>
            <input type="text" class="span-1 last" name="CANTIDAD_<?php echo $key?>"
id='CANTIDAD_<?php echo $key?>' value="" title='El campo "Cantidad" es obligatorio.'
disabled="disable" maxlength="2"/>
            <input type="hidden" name="<?php echo $key?>" id="<?php echo $key?>"
value="<?php echo $key?>"/>
          </div>
        <?php endforeach; ?>
      </div>
      <div id="lineas" name="lineas" clas="span-20 line"><br/><br/>

```

```

        </div>
        <?php endif; ?>
        <input type="hidden" name="id" id="id" value="<?php echo ($pedido->ID_PEDIDO)?$pedido-
>ID_PEDIDO:":"?>"/>
        <input type="hidden" name="validar" id="validar" value="<?php echo
($productos)?$productos:0;?>" />
        <input type="hidden" name="user" id="user" value="<?php echo ($user)?$user:":"?>" />
        <input type="hidden" name="productos" id="productos" value="" />
        <div class="span-20" align="center"><br>
            <input type="submit" id="btnSave" name="btnSave" value="Guardar" />
        </div>
    </form>
</div>
</div>

```

1.2.15.3. Revisar.

```

<div class="span-24">
    <div class="prepend-2 span-20">
        <div class="title">Aprobaci&oacute;n o Cancelaci&oacute;n de Pedido</div>
        <div class="span-20 required_fields line" align="center">
            <label>(*) Campos Obligatorios</label>
        </div>
        <div class="prepend-5 span-10" align="center">
            <div id="dialog_errores">
                <ul id="alerts" class="alerts"></ul>
            </div>
        </div><br><br><br>
        <form action="" method="post" name="frmPedido" id="frmPedido">
            <?php if($pedido->ID_PEDIDO): ?>
                <?php foreach ($producto as $key => $value): ?>
                    <?php if(array_key_exists($key,$PXP)): ?>
                        <div class="span-20 line ">
                            <div class="span-10">
                                <?php echo Form::checkbox(".$key.", $key, FALSE, array('id'=>'ID_PRODUCTO_' . $key.', 'class'
=> 'span-5','checked' => 'checked','disabled' => 'disable' ));?>
                                <div class="span-5 label"> <?php echo $value?> </div>
                            </div>
                            <div class="span-10 last">
                                <div class="prepend-2 span-3 label"> *Cantidad: </div>
                                <input type="text" class="span-1 last" name="CANTIDAD_<?php echo $key?>"
id='CANTIDAD_<?php echo $key?>' value="<?php echo $PXP[$key]?>" title="El campo "Cantidad" es obligatorio."
disabled="disable" maxlength="2"/>
                            </div>
                        </div>
                    <?php endif; ?>
                <?php endforeach; ?>
            <?php endif; ?>
            <input type="hidden" name="id" id="id" value="<?php echo ($pedido->ID_PEDIDO)?$pedido-
>ID_PEDIDO:":"?>"/>
            <div class="span-20" align="center"><br>
                <input type="submit" id="btnSave" name="btnSave" value="Aprobar" />&emsp;&emsp;
                <input type="submit" id="btnCancel" name="btnCancel" value="Cancelar" />
            </div>
        </form>
        <div class="span-20" align="center"><br>
            <a href="javascript:window.history.go(-1);">Regresar</a>
        </div>
    </div>
</div>

```

1.2.15.4. *Select.*

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Pedido Para Modificar</div>
  <?php if($pedido): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/pedido/create');?>">Crear Pedido</a><br><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">Nº de Pedido</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">EDITAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($pedido as $ped): ?>
          <tr>
            <td><?php echo $ped["ID_PEDIDO"] ?></td>
            <td><?php echo $user[$ped["ID_USUARIO"]] ?></td>
            <td><?php echo $ped["ANIO"] ?></td>
            <td><?php echo $ped["MES"] ?></td>
            <td><?php echo $ped["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('admin/pedido/create/'.$ped["ID_PEDIDO"]); ?>">
                </img>
              </a>
            </td>
          </tr>
        </tr>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Pedidos registrados.
    </div>
  </div>
  <?php endif; ?>
</div>
```

1.2.16. Vistas de Producto.

1.2.16.1. Create.

```
<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title"><?php echo ($producto->ID_PRODUCTO)?'Actualizar':'Nuevo'; ?> Producto</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>

    <div class="span-10 prepend-5 last">
      <ul id="alerts" class="alerts"></ul>
    </div>

    <form action="" method="post" name="frmProducto" id="frmProducto"><br/><br/>
      <div class="span-20 line">
        <div class="prepend-5 span-4 label">
          * Producto:
        </div>
        <div class="span-8 label">
          <input class="span-8" type="text" name="DESCRIPCION" id="DESCRIPCION" value="<?php echo
$producto->DESCRIPCION; ?>"/>
        </div>
      </div>

      <div class="span-20 line ">
        <div class="prepend-5 span-4 label">* Departamento:</div>
        <div class="span-8 last" id="cargardepartamento">
          <?php if($producto->ID_PRODUCTO): ?>
            <?php echo Form::select("ID_DEPARTAMENTO", array("=>'-Seleccione-')+$departamento,
$producto->ID_DEPARTAMENTO, array('id'=>'ID_DEPARTAMENTO','class' => 'span-8 select last')); ?>
          <?php else: ?>
            <?php echo Form::select("ID_DEPARTAMENTO", array("=>'-Seleccione-')+$departamento,
$producto->ID_DEPARTAMENTO, array('id'=>'ID_DEPARTAMENTO','class' => 'span-8 select last')); ?>
          <?php endif; ?>
        </div>
      </div>
      <input type="hidden" name="ID_PRODUCTO" id="ID_PRODUCTO" value="<?php echo $producto-
>ID_PRODUCTO; ?>"/>
      <input type="hidden" name="nickname" id="nickname" value=""/>
      <div class="span-20 line" style="text-align: center;"><br/>
        <input id="btnSave" type="submit" value="Guardar"/>
      </div>
    </form>
  </div>
</div>
```

1.2.16.2. *Select.*

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Producto</div>
  <?php if($productos): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/producto/create');?>">Crear Producto</a><br><br>
    <table id="tblProductos">
      <thead>
        <tr>
          <th class="span-3">PRODUCTO</th>
          <th class="span-3">DEPARTAMENTO</th>
          <th class="span-1">EDITAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($productos as $producto): ?>
          <tr>
            <td><?php echo $producto->DESCRIPCION ?></td>
            <td><?php echo $departamento[$producto->ID_DEPARTAMENTO]?></td>
            <td>
              <a href="<?php echo URL::site('admin/producto/create/'. $producto->ID_PRODUCTO); ?>">
                </img>
              </a>
            </td>
          </tr>
        </tbody>
        <?php endforeach; ?>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Productos registrados.
    </div>
  </div>
  <?php endif; ?>
</div>
```

1.2.17. Vistas Usuario.

1.2.17.1. Create.

```

<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title"><?php echo ($user->ID_USUARIO)?'Actualizar':'Nuevo'; ?> Usuario</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>

    <div class="prepend-5 span-10" align="center">
      <div id="dialog_errores">
        <ul id="alerts" class="alerts"></ul>
      </div>
    </div>

    <br/><br/><br/>

    <form action="" method="post" name="frmUser" id="frmUser">

      <div class="span-20 line ">
        <div id="managerContainer">
          <div class="prepend-2 span-3 label">* Nombre:</div>
          <div class="span-5">
            <input type="text" class="span-4" name="NOMBRE" id="NOMBRE" value="<?php echo $user->NOMBRE;
?>" title="El campo "Nombre" es obligatorio." />
          </div>
          <div class="prepend-1 span-4 label">* Cargo:</div>
          <div class="span-2 last" id="cargarperfil">
            <?php if($user->ID_USUARIO): ?>
              <?php echo Form::select("ID_PERFIL", array("=>"-Seleccione-")+ $perfil, $user->ID_PERFIL,
array('id'=>'ID_PERFIL','class' => 'span-4 select last')); ?>
            <?php else: ?>
              <?php echo Form::select("ID_PERFIL", array("=>"-Seleccione-")+ $perfil, $user->ID_PERFIL,
array('id'=>'ID_PERFIL','class' => 'span-4 select last')); ?>
            <?php endif; ?>
          </div>
        </div>

        <div class="span-20 line ">
          <div class="prepend-2 span-3 label">* Apellido:</div>
          <div class="span-5">
            <input type="text" class="span-4" name="APELLIDO" id="APELLIDO" value="<?php echo $user-
>APELLIDO; ?>" title="El campo "Nombre" es obligatorio." />
          </div>
          <div id="cargardepartamento" style="display:<?php echo($user->ID_PERFIL!=1)?'block':'none';
?>">
            <div class="prepend-1 span-4 label">* Departamento:</div>
            <div class="span-2 last">
              <?php if($user->ID_USUARIO): ?>
                <?php echo Form::select("ID_DEPARTAMENTO", array("=>"-Seleccione-")+ $departamento,
$user->ID_DEPARTAMENTO, array('id'=>'ID_DEPARTAMENTO','class' => 'span-4 select last')); ?>
              <?php else: ?>
                <?php echo Form::select("ID_DEPARTAMENTO", array("=>"-Seleccione-")+ $departamento,
$user->ID_DEPARTAMENTO, array('id'=>'ID_DEPARTAMENTO','class' => 'span-4 select last')); ?>
              <?php endif; ?>
            </div>
          </div>

          <div class="span-20 line ">
            <div class="span-10" id="containerusuario">
              <div class="prepend-2 span-3 label">* Usuario:</div>
              <div class="span-5 last">
                <input type="text" class="span-4" name="USUARIO" id="USUARIO" value="<?php echo $user-
>USUARIO; ?>" maxlength="20" />
              </div>
            </div>

            <div class="prepend-1 span-4 label" id="containerstatus">* Estado:</div>

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

        <div class="span-2 last" id="statuscontainer">
            <?php if(!$user->ID_USUARIO): ?>
                <?php echo Form::select('ESTADO', array("=>'-Seleccione-')+Model_User::$status_values,
$user->ESTADO, array('id'=>'ESTADO', 'class'=>'span-4 select')) ;?>
            <?php else: ?>
                <?php echo Form::select('ESTADO', array("=>'-Seleccione-')+Model_User::$status_values,
$user->ESTADO, array('id'=>'ESTADO', 'class'=>'span-4 select')) ;?>
            <?php endif; ?>
        </div>

    </div>

    <?php if($user->ID_USUARIO): ?>
        <div class="span-20 line">
            <div class="prepend-1 span-4 label">Cambiar contrase&ntilde;a?</div>
            <div class="span-4 label">
                <input type="radio" id="change_pass_yes" name="change_pass" value="yes"/>Si
                <input type="radio" id="change_pass_no" name="change_pass" value="no"
checked="checked"/>No
            </div>
        </div>
    <?php endif; ?>

    <div class="span-20 line" id="passContainer" style="display:<?php echo($user->ID_USUARIO)?'none':'block';
?>">
        <div class="span-20 line ">
            <div class="prepend-2 span-3 label">* Contrase&ntilde;a:</div>
            <div class="span-5 line">
                <input type="password" class="span-4" name="CONTRASENIA" id="CONTRASENIA"
maxlength="15"/>
            </div>

            <div class="prepend-1 span-4 label" id="containerstatus">* Confimar Contrase&ntilde;a:</div>
            <div class="span-2 line last" id="statuscontainer">
                <input type="password" class="span-4" name="confPassword" id="confPassword" maxlength="15"/>
            </div>
        </div>

        <input type="hidden" name="id" id="id" value="<?php echo ($user->ID_USUARIO)?$user->ID_USUARIO:":"?>"/>
        <input type="hidden" name="nickname" id="nickname" value="<?php echo ($user->USUARIO)?$user-
>USUARIO:":"?>"/>
        <div class="span-20 " align="center"><br>
            <input type="submit" id="btnSave" name="btnSave" value="Guardar" />
        </div>
    </form>
</div>
</div>

```

1.2.17.2. *Select.*

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Usuario</div>
  <?php if($users): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/user/create');?>">Crear Usuario</a><br><br>
    <table id="tblUsuarios">
      <thead>
        <tr>
          <th class="span-3">NOMBRE</th>
          <th class="span-3">APELLIDO</th>
          <th class="span-3">CARGO</th>
          <th class="span-3">DEPARTAMENTO</th>
          <th class="span-3">USUARIO</th>
          <th class="span-2">ESTADO</th>
          <th class="span-1">EDITAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($users as $user): ?>
        <tr>
          <td><?php echo $user->NOMBRE ?></td>
          <td><?php echo $user->APELLIDO ?></td>
          <td><?php echo $perfiles_array[$user->ID_PERFIL]?></td>
          <td><?php echo $departamentos_array[$user->ID_DEPARTAMENTO]?></td>
          <td><?php echo $user->USUARIO ?></td>
          <td><?php echo $user->ESTADO ?></td>
          <td>
            <a href="<?php echo URL::site('admin/user/create/',$user->ID_USUARIO);?>">
              </img>
            </a>
          </td>
        </tr>
        <?php endforeach; ?>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Usuarios registrados.
    </div>
  </div>
  <?php endif; ?>
</div>
```

1.2.18. Archivos JS de Bodega

1.2.18.1. Select

```
$(document).ready(function(){
  $('#tblBodega').dataTable({
    "bJQueryUI": true,
    "sPaginationType": "full_numbers",
    "oLanguage": {
      "oPaginate": {
        "sFirst": "Primera",
        "sLast": "&Uacute;ltimea",
        "sNext": "Siguiente",
        "sPrevious": "Anterior"
      },
      "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
      "sZeroRecords": "No se encontraron resultados",
      "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
      "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
      "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
      "sSearch": "Criterio de B&uacute;squeda:",
      "sProcessing": "Filtrando.."
    },
    "aoColumns": [
      { "bSortable": true },
      { "bSortable": true }
    ],
    "sDom": '<H"Tr><F"pli>',
    "oTableTools": {
      "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
      "aButtons": [
        {
          "sExtends": "xls",
          "sButtonText": "Exportar a Excel"
        }
      ]
    }
  });

  $('[id^="cantidad_"]').bind('click',function(){
    var ID=$(this).attr('id');
    ID = ID.split('_');
    ID = ID[1];

    $("#cantidad_"+ID).hide();
    $("#cantidad_input_"+ID).show();

  }).change(function()
  {
    var ID=$(this).attr('id');
    ID = ID.split('_');
    ID = ID[2];
    var cantidad = $("#cantidad_input_"+ID).val();
    var original = $("#cantidad_"+ID).html();
    var validar = cantidad.length;
    $("#cantidad_"+ID).html(''); // Loading image

    if(validar>0)
    {
      if(!isNaN(cantidad))
      {
        $.ajax({
          type: "POST",
          url:document_root+'admin/bodega/editcontent',
          data: {
            pxb_id: ID,
            cantidad: cantidad
          },
          cache: false,
          success: function(data)

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
    {
        $("#cantidad_"+ID).html(data);
        $("#cantidad_input_"+ID).val(data);
    }
    });
}
else
{
    alert('Escriba solo n\xfameros.');
```

```
    $("#cantidad_"+ID).html(original);
    $("#cantidad_input_"+ID).val(original);
}
}
else
{
    alert('Debe ingresar un valor.');
```

```
    $("#cantidad_"+ID).html(original);
    $("#cantidad_input_"+ID).val(original);
}
}
});

$(".editbox").mouseup(function()
{
    return false
});

$(document).mouseup(function()
{
    $(".editbox").hide();
    $(".text").show();
});
});
```

1.2.1. Archivos JS de Departamento.

1.2.1.1. Create

```
$(document).ready(function(){
  /*FORM VALIDATIONS*/
  $("#frmDepartamento").validate({
    errorLabelContainer: "#alerts",
    wrapper: "i",
    onfocusout: false,
    onkeyup: false,
    async: false,
    rules: {
      NOMBRE: {
        required: true,
        maxlength: 50,
        textOnly: true,
        remote: {
          url: document_root+"admin/departamento/checknombre",
          type: 'post',
          data: {
            id: $("#ID_DEPARTAMENTO").val()
          }
        }
      },
      DESCRIPCION: {
        required: true,
        textOnly: true
      }
    },
    messages: {
      NOMBRE: {
        required: 'El campo "Nombre" es obligatorio.',
        maxlength: 'El campo "Nombre" debe contener máximo 50 caracteres.',
        textOnly: 'El campo "Nombre" solo puede contener letras.',
        remote: 'El Nombre del departamento ya existe.'
      },
      DESCRIPCION: {
        required: 'El campo "Descripcion" es obligatorio.',
        textOnly: 'El campo "Descripcion" solo puede contener letras.'
      }
    },
    submitHandler: function(form) {
      $("#type_id").attr('disabled', false);
      $("#btnSave").attr('disabled', 'true');
      $("#btnSave").val('Espere por favor...');
      form.submit();
    }
  });
  /*END FORM VALIDATIONS*/

  /* LISTENERS */
  $("#NOMBRE").bind('change', function(){
    var nick = $("#NOMBRE").val()
    $("#nickname").val(nick);
  });

  /* END LISTENERS */
});
```

1.2.1.2. *Select.*

```
$(document).ready(function(){
    $('#tblDepartamentos').dataTable({
        "bJQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPaginate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltime",
                "sNext": "Siguiete",
                "sPrevious": "Anterior"
            },
            "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
            "sZeroRecords": "No se encontraron resultados",
            "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
            "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
            "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
            "sSearch": "Criterio de B&uacute;squeda:",
            "sProcessing": "Filtrando.."
        },
        "aoColumns": [
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": false}
        ],
        "sDom": '<"H"Tfr>t<"F"pli>',
        "oTableTools": {
            "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
            "aButtons": [
                {"sExtends": "xls",
                "sButtonText": "Exportar a Excel"}
            ]
        }
    });
});
```

1.2.1. Archivos JS de Despacho.

1.2.1.1. Create.

```
$(document).ready(function(){
    var validacheck=$("#validar").val();

    /*FORM VALIDATIONS*/
    $("#frmDespacho").validate({
        errorLabelContainer: "#alerts",
        wrapper: "i",
        onfocusout: false,
        onkeyup: false,
        async: false,
        rules: {
            validar: {
                required: true,
                min: true
            }
        },
        messages: {
            validar: {
                required: 'Debe elegir por lo menos un pedido',
                min: 'Debe elegir por lo menos un pedido'
            }
        },
        submitHandler: function(form) {
            $("#btnSave").attr('disabled', 'true');
            $("#btnSave").val('Espere por favor...');
            form.submit();
        }
    });
    /*END FORM VALIDATIONS*/

    /* LISTENERS */
    $('[id^="ID_PEDIDO_"]').bind('click',function(){
        if($(this).is(':checked'))
        {
            validacheck++;
            $("#validar").val(validacheck);
        }
        else
        {
            validacheck--;
            $("#validar").val(validacheck);
        }
    });

    /* END LISTENERS */
    $(function() {
        $("#dialog").dialog({
            autoOpen: false,
            height: 350,
            width: 550,
            show: "blind",
            hide: "blind"
        });
    });
    $("#des-pedido-link").click(function(e){
        e.preventDefault();
        var titulo_pedido = $(this).attr("pedido_id");
        var contenido_pedido = "";
        $("#contenido").empty();
        $.ajax({
            url: document_root+'admin/despacho/get_content',
            async: false,
            type: "post",
            data: {
                pedido_id: $(this).attr("pedido_id")
            }
        });
    });
});
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
    }},
    dataType: 'json',
    success: function(data){
        contenido_pedido = data
    }
});

$( "#dialog" ).dialog( "open" );
$( "#ui-dialog-title-dialog" ).html('Productos del Pedido: '+titulo_pedido);
$( "#contenido" ).append(contenido_pedido);

});

});
```

1.2.1.2. *Procesar.*

```
$(document).ready(function(){
    $('#tblPedidos').dataTable({
        "bjQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPagninate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltime",
                "sNext": "Siguiete",
                "sPrevious": "Anterior"
            },
        },
        "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
        "sZeroRecords": "No se encontraron resultados",
        "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
        "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
        "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
        "sSearch": "Criterio de B&uacute;squeda:",
        "sProcessing": "Filtrando.."
    },
    "aoColumns": [
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": false}
    ],
    "sDom" : '<"H"Tfr>t<"F"pli>',
    "oTableTools": {
        "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
        "aButtons": [{
            "sExtends": "xls",
            "sButtonText": "Exportar a Excel"
        }]
    }
});
});
```

1.2.1.3. Revisar.

```

$(document).ready(function(){
    var validacheck=$("#validar").val();
    $('[id^="btn_"]').bind('click',function(){
        var ID=$(this).attr('id');
        ID = ID.split('_');
        ID = ID[1];
        if(ID=='Comprar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Comprado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Enviar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Enviado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Entregar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Entregado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Cancelar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Cancelado?\n\nNota: esto tambien cancela los pedidos aprobados anteriormente.','Confirmar Accion'))
            {
                return false;
            }
        }
    });
    /* END LISTENERS */
    $(function() {
        $("#dialog").dialog({
            autoOpen: false,
            height: 350,
            width: 550,
            show: "blind",
            hide: "blind"
        });
    });
    $("#des-pedido-link").click(function(e){
        e.preventDefault();
        var titulo_pedido = $(this).attr("pedido_id");
        var contenido_pedido = "";
        $("#contenido").empty();
        $.ajax({
            url:document_root+'admin/despacho/get_content',
            async: false,
            type: "post",
            data: {
                pedido_id: $(this).attr("pedido_id")
            },
            dataType: 'json',
            success: function(data){
                contenido_pedido = data
            }
        });
        $("#dialog").dialog("open");
        $("#ui-dialog-title-dialog").html('Productos del Pedido: '+titulo_pedido);
        $("#contenido").append(contenido_pedido);
    });
});

```

1.2.1.4. *Select.*

```
$(document).ready(function(){
    $('#tblPedidos').dataTable({
        "bjQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPagninate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltime",
                "sNext": "Siguiete",
                "sPrevious": "Anterior"
            },
        },
        "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
        "sZeroRecords": "No se encontraron resultados",
        "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
        "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
        "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
        "sSearch": "Criterio de B&uacute;squeda:",
        "sProcessing": "Filtrando.."
    },
    "aoColumns": [
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": false}
    ],
    "sDom" : '<"H"Tfr>t<"F"pli>',
    "oTableTools": {
        "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
        "aButtons": [{
            "sExtends": "xls",
            "sButtonText": "Exportar a Excel"
        }]
    }
});
});
```

1.2.2. Archivos JS de Pedido.

1.2.2.1. Aprobar.

```
$(document).ready(function(){
  $('#tblPedidos').dataTable({
    "bJQueryUI": true,
    "sPaginationType": "full_numbers",
    "oLanguage": {
      "oPaginate": {
        "sFirst": "Primera",
        "sLast": "&Uacute;ltimea",
        "sNext": "Siguiente",
        "sPrevious": "Anterior"
      },
      "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
      "sZeroRecords": "No se encontraron resultados",
      "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
      "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
      "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
      "sSearch": "Criterio de B&uacute;squeda:",
      "sProcessing": "Filtrando.."
    },
    "aoColumns": [
      {"bSortable": true},
      {"bSortable": true},
      {"bSortable": true},
      {"bSortable": true},
      {"bSortable": true},
      {"bSortable": false}
    ],
    "sDom": '<H"Tr>t<"F"pli>',
    "oTableTools": {
      "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
      "aButtons": [
        {"sExtends": "xls",
        "sButtonText": "Exportar a Excel"}
      ]
    }
  });
});
```

1.2.2.2. Create.

```
$(document).ready(function(){
  var validacheck=$("#validar").val();
  /*FORM VALIDATIONS*/
  $("#frmPedido").validate({
    errorLabelContainer: "#alerts",
    wrapper: "i",
    onfocusout: false,
    onkeyup: false,
    async:false,
    rules: {
      validar: {
        required: true,
        min: true
      }
    },
    messages:{
      validar: {
        required: 'Debe elegir por lo menos un producto',
        min: 'Debe elegir por lo menos un producto'
      }
    },
    submitHandler: function(form) {
      $("#btnSave").attr('disabled', 'true');
      $("#btnSave").val('Espere por favor...');
      form.submit();
    }
  });
  /*END FORM VALIDATIONS*/
  /* LISTENERS */
  $('[id^="ID_PRODUCTO_"]').bind('click',function(){
    if($(this).is(':checked'))
    {
      $("#CANTIDAD_"+$(this).val()+").attr('disabled',false);
      validacheck++;
      $("#validar").val(validacheck);
      $("#CANTIDAD_"+$(this).val()+").rules('add',{
        required:true,
        maxlength: 2,
        digits: true,
        messages: {
          required:'El campo "Cantidad" es obligatorio.',
          maxlength: 'El campo "Cantidad" debe contener m&aacute;ximo 2 digitos.',
          digits: 'El campo "Cantidad" debe contener solo digitos.'
        }
      });
    }
    else
    {
      $("#CANTIDAD_"+$(this).val()+").attr('disabled',true);
      $("#CANTIDAD_"+$(this).val()+").val('');
      validacheck--;
      $("#validar").val(validacheck);
    }
  });
  /* END LISTENERS */
});
//ADD PASSWORD RULES
function addAniadirRules(aumenta)
{
  $("#CANTIDAD_"+aumenta).rules('add',{
    required:true,
    maxlength: 2,
    digits: true,
    messages: {
      required:'El campo "Cantidad" es obligatorio.',
      maxlength: 'El campo "Cantidad" debe contener m&aacute;ximo 2 digitos.',
      digits: 'El campo "Contrase&ntilde;a" debe contener solo digitos.'
    }
  });
});
```

1.2.2.3. Revisar.

```
$(document).ready(function(){
    $('#btnSave').bind('click',function(){
        if(!confirm('¿Está seguro de cambiar estado de Pedido a Aprobado?','Confirmar Accion'))
        {
            return false;
        }
    });
    $('#btnCancel').bind('click',function(){
        if(!confirm('¿Está seguro de cambiar estado de Pedido a Cancelado?','Confirmar Accion'))
        {
            return false;
        }
    });
});
```

1.2.2.4. Select.

```
$(document).ready(function(){
    $('#tblPedidos').dataTable({
        "bJQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPaginate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltima",
                "sNext": "Siguiente",
                "sPrevious": "Anterior"
            },
            "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
            "sZeroRecords": "No se encontraron resultados",
            "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
            "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
            "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
            "sSearch": "Criterio de B&uacute;squeda:",
            "sProcessing": "Filtrando.."
        },
        "aoColumns": [
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": false}
        ],
        "sDom": '<H"Tfr>t<"F"pli>',
        "oTableTools": {
            "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
            "aButtons": [
                "sExtends": "xls",
                "sButtonText": "Exportar a Excel"
            ]
        }
    });
});
```

1.2.3. Archivos JS de Producto.

1.2.3.1. Create.

```
$(document).ready(function(){
  /*FORM VALIDATIONS*/
  $("#frmProducto").validate({
    errorLabelContainer: "#alerts",
    wrapper: "i",
    onfocusout: false,
    onkeyup: false,
    async: false,
    rules: {
      DESCRIPCION: {
        required: true,
        remote: {
          url: document_root+"admin/producto/checkdescripcion",
          type: 'post',
          data: {
            id: $("#ID_PRODUCTO").val()
          }
        },
        maxlength: 50
      },
      ID_DEPARTAMENTO: {
        required: true
      }
    },
    messages: {
      DESCRIPCION: {
        required: 'El campo "Producto" es obligatorio.',
        maxlength: 'El campo "Producto" debe contener máximo 50 caracteres.',
        remote: 'El Producto ingresado ya existe.'
      },
      ID_DEPARTAMENTO: {
        required: 'El campo "Departamento" es obligatorio.'
      }
    },
    submitHandler: function(form) {
      $("#type_id").attr('disabled', false);
      $("#btnSave").attr('disabled', 'true');
      $("#btnSave").val('Espere por favor...');
      form.submit();
    }
  });
  /*END FORM VALIDATIONS*/
});
```

1.2.3.2. *Select.*

```
$(document).ready(function(){
    $('#tblProductos').dataTable({
        "bJQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPaginate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltime",
                "sNext": "Siguiete",
                "sPrevious": "Anterior"
            },
            "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
            "sZeroRecords": "No se encontraron resultados",
            "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
            "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
            "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
            "sSearch": "Criterio de B&uacute;squeda:",
            "sProcessing": "Filtrando.."
        },
        "aoColumns": [
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": false}
        ],
        "sDom": '<H"Tfr>t<"F"pli>',
        "oTableTools": {
            "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
            "aButtons": [
                {"sExtends": "xls",
                "sButtonText": "Exportar a Excel"}
            ]
        }
    });
});
```

1.2.4. Archivos JS Usuario.

1.2.4.1. Create.

```
$(document).ready(function(){
  /*FORM VALIDATIONS*/
  $("#frmUser").validate({
    errorLabelContainer: "#alerts",
    wrapper: "i",
    onfocusout: false,
    onkeyup: false,
    async: false,
    rules: {
      NOMBRE: {
        required: true,
        maxlength: 30,
        textOnly: true
      },
      APELLIDO: {
        required: true,
        maxlength: 30,
        textOnly: true
      },
      ID_PERFIL:{
        required: true
      },
      USUARIO:{
        required: true,
        remote:{
          url: document_root+"admin/user/checkuser",
          type: 'post',
          data:{
            id: $('#id').val()
          }
        },
        maxlength: 20,
        minlength: 5,
        textOnly: true
      },
      ESTADO: {
        required: true
      }
    },
    messages:{
      NOMBRE: {
        required: 'El campo "Nombre" es obligatorio.',
        maxlength: 'El campo "Nombre" debe contener máximo 50 caracteres.',
        textOnly: 'El campo "Nombre" solo puede contener letras.'
      },
      APELLIDO: {
        required: 'El campo "Apellido" es obligatorio.',
        maxlength: 'El campo "Apellido" debe contener máximo 50 caracteres.',
        textOnly: 'El campo "Apellido" solo puede contener letras.'
      },
      ID_PERFIL:{
        required: 'El campo "Cargo" es obligatorio.'
      },
      USUARIO:{
        required: 'El campo "Usuario" es obligatorio.',
        remote: 'El Usuario ingresado ya existe.',
        maxlength: 'El campo "Usuario" debe contener máximo 20 caracteres.',
        minlength: 'El campo "Usuario" debe contener mínimo 5 caracteres.',
        textOnly: 'El campo "Usuario" debe contener solo caracteres.'
      },
      ESTADO: {
        required: 'El campo "Estado" es obligatorio.'
      }
    },
    submitHandler: function(form) {
      $('#type_id').attr('disabled', false);
      $('#btnSave').attr('disabled', true);
      $('#btnSave').val('Espere por favor...');
    }
  });
});
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
        form.submit();
    }
});
/*END FORM VALIDATIONS*/

//VALIDATES FIELDS WHEN IT'S AN UPDATE
if(!$("#id").val())
addPasswordRules();

if($("#cargarperfil option:selected").val()!==2)
{
    $("#cargardepartamento").css('display','none');
    $("#ID_DEPARTAMENTO").rules('remove');
    $("#ID_DEPARTAMENTO").val("");
}

/* LISTENERS */
$([id^="change_pass_"]).bind('click',function(){
    if($(this).val()=='yes'){
        $("#passContainer").css('display','block');
        addPasswordRules();
    }else{
        $("#passContainer").css('display','none');
        $("#CONTRASENIA").rules('remove');
        $("#confPassword").rules('remove');
        $("#password").val("");
        $("#confPassword").val("");
    }
});

$("#cargarperfil").bind('change',function(){
    if($("#cargarperfil option:selected").val()===2){
        $("#cargardepartamento").css('display','block');
        addDepartamentoRules();
    }else{
        $("#cargardepartamento").css('display','none');
        $("#ID_DEPARTAMENTO").rules('remove');
        $("#ID_DEPARTAMENTO").val("");
    }
});

$("#USUARIO").bind('change',function(){
    var nick = $("#USUARIO").val()
    $("#nickname").val(nick);
});

/* END LISTENERS */

});

//ADD PASSWORD RULES
function addPasswordRules()
{
    $("#CONTRASENIA").rules('add',{
        required:true,
        minlength: 6,
        maxlength: 15,
        messages: {
            required:'El campo "Contrase&ntilde;a" es obligatorio.',
            minlength: 'El campo "Contrase&ntilde;a" debe contener m&iacute;nimo 6 caracteres.',
            maxlength: 'El campo "Contrase&ntilde;a" debe contener m&aacute;ximo 15 caracteres.'
        }
    });
    $("#confPassword").rules('add',{
        required:true,
        minlength: 6,
        maxlength: 15,
        equalTo: "#CONTRASENIA",
    });
}
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
messages: {
  required: 'El campo "Confirmar Contrase&ntilde;a" es obligatorio.',
  minlength: 'El campo "Confirmar Contrase&ntilde;a" debe contener m&iacute;nimo 6 caracteres.',
  maxlength: 'El campo "Confirmar Contrase&ntilde;a" debe contener m&aacute;ximo 15 caracteres.',
  equalTo: 'El campo "Contrase&ntilde;a" y "Confirmar Contrase&ntilde;a" deben coincidir.'
}
});
}
```

```
function addDepartaentoRules()
{
  $('#ID_DEPARTAMENTO').rules('add',{
    required:true,
    messages: {
      required: 'El campo "Departamento" es obligatorio.'
    }
  });
}
```

1.2.4.2. Select.

```
$(document).ready(function(){
    $('#tblUsuarios').dataTable({
        "bJQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPaginate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltime",
                "sNext": "Siguiente",
                "sPrevious": "Anterior"
            },
            "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
            "sZeroRecords": "No se encontraron resultados",
            "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
            "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
            "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
            "sSearch": "Criterio de B&uacute;squeda:",
            "sProcessing": "Filtrando.."
        },
        "aoColumns": [
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": false}
        ],
        "sDom": '<H"Tfr>t<"F"pli>',
        "oTableTools": {
            "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
            "aButtons": [
                "sExtends": "xls",
                "sButtonText": "Exportar a Excel"
            ]
        }
    });
});
```

1.3. Perfil Control.

1.3.1. Contenedor.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Control_Containers_Control extends Controller_Template {
```

```
    public $template = 'controller/control/containers/control';
```

```
    /**
```

```
     * El atributo que va a ser utilizado en cada acción
```

```
     * NULL por defecto
```

```
     * @var View
```

```
     */
```

```
    public $view;
```

```
    /**
```

```
     * @see system/classes/kohana/controller/Kohana_Controller_Template::before()
```

```
     */
```

```
    public function before()
```

```
    {
```

```
        Session::instance();
```

```
        //Required because we are extending Controller_Template
```

```
        parent::before();
```

```
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario', FALSE))-
```

```
>find();
```

```
        if(!Session::instance()->get('treasure_usuario', FALSE) OR Session::instance()->get('treasure_usuario', FALSE)=="
```

```
OR $user->ID_PERFIL=4)
```

```
        {
```

```
            HTTP::redirect('login');
```

```
        }
```

```
        //$this->_check_permissions();
```

```
        if ( $this->auto_render )
```

```
        {
```

```
            //Sets the "Default" title set on the container
```

```
            $this->template->title = 'Treasure';
```

```
            $this->template->content = '';
```

```
            $this->template->styles = array();
```

```
            $this->template->scripts = array();
```

```
        }
```

```
        //Cleans the view so that every action can set it
```

```
        //by itself without problems
```

```
        $this->view = NULL;
```

```
    }
```

```
    /**
```

```
     * @see system/classes/kohana/controller/Kohana_Controller_Template::after()
```

```
     */
```

```
    public function after()
```

```
    {
```

```
        if ( $this->auto_render )
```

```
        {
```

```
            $action = $this->request->action();
```

```
            $controller = $this->request->controller();
```

```
            $directory = $this->request->directory();
```

```
            //Build the styles array
```

```
            $css_file = 'media/css/controller/'.$directory.'/'.$controller.'/'.$action.'.css';
```

```
            $styles = array(
```

```
                'media/css/screen.css' => 'screen, projection',
```

```
                'media/css/src/print.css' => 'print',
```

```
                'media/css/jquery-ui/smoothness/jquery-ui-1.8.16.custom.css' => 'all',
```

```
                'media/css/dataTables/TableTools.css' => 'screen',
```

```
                'media/css/dataTables/TableTools_JUI.css' => 'screen',
```

```
                'media/css/dataTables/demo_table_jui.css' => 'screen',
```

```
                'media/css/src/ie.css' => 'screen, projection',
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
'media/css/styles.css' => 'screen',
'media/css/menu.css' => 'screen',
    'media/css/menu-style.css' => 'screen',
'media/css/jquery-ui/jquery.autocomplete.css' => 'all',
);

//VALIDATING FILE
if(file_exists($css_file) )
{
    $styles = array_merge( $styles, array($css_file=>'screen') );
}

//Build the scripts array
$js_file = 'media/js/controller/'.$_DIRECTORY__.$controller.'/'.$_ACTION__.'.js';
$scripts = array(
    'media/js/jqueryLibrary/jquery-1.6.2.min.js',
    'media/js/jqueryLibrary/jquery-ui-1.8.16.custom.min.js',
    'media/js/jqueryLibrary/jquery-validation.js',
        'media/js/jqueryLibrary/jquery.autocomplete.js',
        'media/js/dataTables/jquery.dataTables.js',
    'media/js/dataTables/TableTools.js',
    'media/js/dataTables/ZeroClipboard.js',
        'media/js/dataTables/dataTables.editor.js',
    'media/js/globalfunctions.js',
    'media/js/ajaxupload.js',
    'media/js/system.js',
    'media/js/mbMenu.js',
);

//VALIDATING FILE
if(file_exists($js_file) )
{
    $scripts = array_merge( $scripts, array($js_file) );
}

//Set styles and scripts on the template (container). This can also be
//done from any action method
$this->template->styles = array_merge( $styles, $this->template->styles );
$this->template->scripts = array_merge( $scripts, $this->template->scripts );

//Get Menu
//Uncomment once we have all the menu logic created
$menu=FALSE;//$this->_get_menu( Model_Globalfunctions::get_menu( Session::instance()-
>get('discotecas_serial_prf', FALSE) ) );
if($menu){
    $this->template->menu = $menu;
}else{
    $this->template->menu = false;
}

//If $this->view is set display it on the
//containers $content
$this->template->content = $this->view;
}

//Required because we are extending Controller_Template
parent::after();
}
}
```

1.3.2. Contenedor de Vista.

```
<!doctype html>
<html lang="<?php echo l18n::$lang ?>">
  <head>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/x-icon" href="/treasure/media/img/icon.ico">
    <title><?php echo $title ?></title>
    <?php foreach ($styles as $file => $type) echo HTML::style($file, array('media' => $type)), PHP_EOL ?>
    <?php foreach ($scripts as $file) echo HTML::script($file), PHP_EOL ?>
    <script type="text/javascript">
      var document_root = '<?php echo Url::site();?>';
    </script>
  </head>
  <body>
    <div class="container" id="admin_container">
      <div class="span-24 general_bg">
        <div class="span-24 last " style=" padding-top: 5px;">
          <div class="prepend-23 span-1 last admin_banner">
            <strong><a href="<?php echo Url::site();?>" style="text-decoration:none; color: black; font-weight:
normal;">Salir</a></strong>
          </div>
        </div>

        <div class="span-24 ">
          <?php echo Helper_Control_Menu::menu(); ?>
        </div>
      </div>
      <div class="span-24 content">
        <div class="prepend-2 span-20 append-2 line last" id="container-flash-messages" style=" padding-top:
15px;">
          <?php FlashMessenger::factory()->get_messages(); ?>
        </div>
        <?php echo $content ?>
      </div>
      <div id="spinner" style="display: none;">
        <div id="ajax_loader"></div>
        <div class="ajax_loader_text">Cargando...</div>
      </div>
    </div>
  </body>
</html>
```

1.3.3. Controlador de Despacho.

```

<?php
class Controller_Control_Despacho extends Controller_Control_Containers_Control {

    public function action_index()
    {
        HTTP::redirect('control/despacho/create/');
    }

    public function action_get_content()
    {
        $content="";
        if($this->request->is_ajax())
        {
            $pedido_id = arr::get($this->request->post(), 'pedido_id');
            try
            {
                $closed = ORM::factory('pedidodespachoxproducto')
                ->where('ID_PEDIDO','=',$pedido_id)
                ->find_all()->as_array();
                foreach ($closed as $value) {
                    $nombre_producto = ORM::factory('producto')->where('ID_PRODUCTO', '=', $value->ID_PRODUCTO)-
                >find();
                    $content = $content.'<div class="span-17 line"><div class="span-8 label"> '.$nombre_producto-
                >DESCRIPCION.'</div><div class="span-8 label"> Cantidad: '.$value->CANTIDAD.'</div> </div> ';
                }
                echo json_encode($content);
            }
            catch(Exception $e)
            {
                echo json_encode($e->getMessage());
            }
        }

        $this->auto_render = FALSE;
    }
    else
    {
        HTTP::redirect('control/main');
    }
}

    public function action_procesar()
    {
        $this->view = View::factory('controller/control/despacho/procesar');
        $despachosp = DB::query(Database::SELECT, "SELECT Distinct('ID_DESPACHO') FROM
        `pedido_despacho_x_producto` WHERE `ESTADO` in ('EN PROCESO','COMPRADO','ENVIADO') AND `ID_DESPACHO`
        is not NULL")->execute()->as_array();
        $despacho = array();
        foreach ($despachosp as $value) {
            $despachos = ORM::factory('despacho')->where('ID_DESPACHO', '=', $value)->find();
            $despacho = $despacho + array($despachos->ID_DESPACHO=>$despachos->as_array());
        }
        $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
        $user = array();
        foreach ($users as $value) {
            $nombres = $value->NOMBRE.'<br>'.$value->APELLIDO;
            $user = $user + array($value->ID_USUARIO=>$nombres);
        }
        if(! count($despacho))
            $despacho=NULL;
        $this->view->set('user', $user);
        $this->view->set('despacho', $despacho);
    }

    public function action_revisar()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/control/despacho/revisar' );
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

//IF USER SUBMITTED FORM
$ID_DESPACHO = $this->request->param("params");
$pedidos = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM `pedido_despacho_x_producto`
WHERE `ESTADO` in ('EN PROCESO','COMPRADO','ENVIADO') AND `ID_DESPACHO` = ".$ID_DESPACHO."")-
>execute()->as_array();
$this->view->set('pedidos', $pedidos);
$despacho = ORM::factory('despacho')->where('ID_DESPACHO', '=', $ID_DESPACHO)->find();
$this->view->set('despacho', $despacho);
$user = ORM::factory('user')->where('ID_USUARIO', '=', $despacho->ID_USUARIO)->find();
$this->view->set('user', $user->ID_USUARIO);
$validar = count($pedidos);
$this->view->set('validar', $validar);

//VALIDATE POST
if( $this->request->post() )
{
    try
    {
        $values= $this->request->post();
        $proceso = 0;
        $pedidoProceso = array();
        echo Debug::vars($values);
        foreach ($values as $key=>$value) {
            if(strpos($value, 'Comprar')!= false)
            {
                $proceso = 1;
            }
            if(strpos($value, 'Enviar')!= false)
            {
                $proceso = 2;
            }
            if(strpos($value, 'Entregar')!= false)
            {
                $proceso = 3;
            }
            if(strpos($value, 'Cancelar')!= false)
            {
                $proceso = 4;
            }
            if(strpos($key, 'ID_PEDIDO_')!= false)
            {
                $aux = explode('_', $key);
                $pedidoProceso = $pedidoProceso + array($aux[2]==>$aux[2]);
            }
        }
        switch ($proceso)
        {
            case 1:foreach ($pedidoProceso as $value) {
                DB::update('pedido_despacho_x_producto'-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'COMPRADO'))->where('ID_PEDIDO', '=', $value)
->execute();
            }
            FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:comprado'));
            break;
            case 2:foreach ($pedidoProceso as $value) {
                DB::update('pedido_despacho_x_producto'-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'ENVIADO'))->where('ID_PEDIDO', '=', $value)
->execute();
            }
            FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:enviado'));
            break;
            case 3:foreach ($pedidoProceso as $value) {
                $producto = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $value)->find_all();
                foreach ($producto as $value2) {
                    $cantidad = ORM::factory('productobodega')->where('ID_PRODUCTO', '=', $value2-
>ID_PRODUCTO)->find();
                    $valor = $cantidad->CANTIDAD + $value2->CANTIDAD;
                    DB::update('producto_x_bodega')->set(array('CANTIDAD'=>$valor,))->where('ID_PRODUCTO',
'=', $value2->ID_PRODUCTO)->execute();
                }
            }
        }
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
        }
        DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'ENTREGADO'))->where('ID_PEDIDO', '=', $value)
        ->execute();
    }
}

FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:entregado'));
break;
case 4:foreach ($pedidoProceso as $value) {
    DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'CANCELADO'))->where('ID_PEDIDO', '=', $value)
    ->execute();
}
FlashMessenger::factory()->set_message('error', Kohana::message('admin',
'admin:despacho:estado:cancelado'));
break;
default : HTTP::redirect('control/main');
break;
}
}
catch(Exception $e)
{
    echo $e->getMessage();
    FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('control/despacho/procesar/');
}
$this->view->set( '$despacho', $despacho );
}
?>
```

1.3.4. Controlador Main.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Control_Main extends Controller_Control_Containers_Control {
```

```
    public function action_index()
    {
        $option = $this->request->param('params');
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))->find();
        $name = ".strtoupper ($user->NOMBRE)";
        $this->view = View::factory( 'controller/control/main' );
        $this->view->set('name',$name);
    }
}
```

1.3.5. Helper

1.3.5.1. Menú.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Helper_Control_Menu {  
    public static function menu($seg1 = null, $seg2 = null, $seg3 = null)  
    {  
        $items = array();  
  
        /*DESPACHOS*/  
        $items[] = array('id' => 14, 'parent_id' => 0, 'title' => 'Despachos', 'link' => '#');  
        $items[] = array('id' => 17, 'parent_id' => 14, 'title' => 'Procesar', 'link' => 'control/despacho/procesar');
```

```
        // Build the current path  
        $current = "";  
        $current .= $seg1 ? $seg1 : "";  
        $current .= $seg2 ? ("/" . $seg2): "";  
        $current .= $seg3 ? ("/" . $seg3): "";  
  
        // Let's get the body of the current page  
        //$body = ORM::factory('entry')->where('link','=', $current)->find()->body;  
        $options = array('link_prepend' => "", 'current_path' => $current);  
  
        return MenuStructure::factory($items, $options)->get_menu();  
    }  
}
```

```
?>
```

1.3.6. Vistas de Despacho.

1.3.6.1. Procesar.

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Despacho Para Procesar</div>
  <?php if($despacho): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/despacho/create');?>">Crear Despacho</a><br><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">Nº</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">REVISAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($despacho as $desp): ?>
          <tr>
            <td><?php echo $desp["ID_DESPACHO"] ?></td>
            <td><?php echo $user[$desp["ID_USUARIO"]]?></td>
            <td><?php echo $desp["ANIO"]?></td>
            <td><?php echo $desp["MES"]?></td>
            <td><?php echo $desp["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('control/despacho/revisar/'.$desp["ID_DESPACHO"]); ?>">
                </img>
              </a>
            </td>
          </tr>
        <?php endforeach; ?>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Despachos registrados.
    </div>
  </div>
  <?php endif; ?>
</div>
```

1.3.6.2. *Revisar.*

```

<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title">Detalles del despacho elegido</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>

    <div class="prepend-5 span-10" align="center">
      <div id="dialog_errores">
        <ul id="alerts" class="alerts"></ul>
      </div>
      <div id="dialog" title="">
        <p id="pedido_productos">
          <div id="contenido"></div>
        </p>
      </div>
    </div>

    <br/><br/><br/>

    <form action="" method="post" name="frmDespacho" id="frmDespacho">
      <?php if($despacho->ID_DESPACHO): ?>
        <?php foreach ($pedidos as $value): ?>
          <div class="span-20 line ">
            <div class="prepend-2 span-5 label"> N&Uacute;mero DE PEDIDO: </div>
            <div class="prepend-2 span-10">
              <?php echo Form::checkbox("$value[ID_PEDIDO].", 'ID_PEDIDO_'.$value[ID_PEDIDO].",
FALSE, array('id'=>'ID_PEDIDO_'.$value[ID_PEDIDO].', 'class' => 'span-5', 'checked' =>
'checked', 'disabled'=>'disable'));?>
              <div class="span-3 label"> <?php echo $value[ID_PEDIDO]?> </div>
              <a class='des-pedido-link' pedido_id='<?php echo $value[ID_PEDIDO]?>' href="">Ver
Productos</a>
              <input type="hidden" name="<?php echo 'ID_PEDIDO_'.$value[ID_PEDIDO]."?>" id="<?php
echo 'ID_PEDIDO_'.$value[ID_PEDIDO]."?>" value="<?php echo $value[ID_PEDIDO]?>" />
            </div>
          </div>
        <?php endforeach; ?>
        <?php endif; ?>
        <div class="span-20 " align="center"><br>
          <input type="submit" id="btn_Entregar" name="btn_Entregar" value="Entregar" />
        </div>
      </form>
      <div class="span-20 " align="center"><br>
        <a href="javascript:window.history.go(-1);">Regresar</a>
      </div>
    </div>
  </div>

```

1.3.7. *Vistas de Main.*

```
<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="prepend-5 span-20 title_main_page center">
      PAGINA DE CONTROLADOR<br><br>
      BIENVENIDO <?php echo $name?>
    </div>
  </div>
</div>
```

1.3.8. Archivos JS de Despacho.

1.3.8.1. Procesar.

```
$(document).ready(function(){
  $('#tblPedidos').dataTable({
    "bJQueryUI": true,
    "sPaginationType": "full_numbers",
    "oLanguage": {
      "oPaginate": {
        "sFirst": "Primera",
        "sLast": "&Uacute;ltimea",
        "sNext": "Siguiente",
        "sPrevious": "Anterior"
      },
      "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
      "sZeroRecords": "No se encontraron resultados",
      "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
      "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
      "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
      "sSearch": "Criterio de B&uacute;squeda:",
      "sProcessing": "Filtrando.."
    },
    "aoColumns": [
      {"bSortable": true},
      {"bSortable": true},
      {"bSortable": true},
      {"bSortable": true},
      {"bSortable": true},
      {"bSortable": false}
    ],
    "sDom": '<H"Tr>t<"F"pli>',
    "oTableTools": {
      "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
      "aButtons": [
        {"sExtends": "xls",
         "sButtonText": "Exportar a Excel"}
      ]
    }
  });
});
```

1.3.8.2. Revisar.

```
$(document).ready(function(){
    var validacheck=$("#validar").val();
    $('[id^="btn_"]').bind('click',function(){
        var ID=$(this).attr('id');
        ID = ID.split('_');
        ID = ID[1];
        if(ID=='Comprar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Comprado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Enviar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Enviado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Entregar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Entregado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Cancelar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Cancelado?\n\nNota: esto tambien cancela los
pedidos aprobados anteriormente.','Confirmar Accion'))
            {
                return false;
            }
        }
    });
}
/* END LISTENERS */
$(function() {
    $("#dialog").dialog({
        autoOpen: false,
        height: 350,
        width: 550,
        show: "blind",
        hide: "blind"
    });
});
$("#des-pedido-link").click(function(e){
    e.preventDefault();
    var titulo_pedido = $(this).attr("pedido_id");
    var contenido_pedido = "";
    $("#contenido").empty();
    $.ajax({
        url:document_root+control/despacho/get_content',
        async: false,
        type: "post",
        data: {
            pedido_id: $(this).attr("pedido_id")
        },
        dataType: 'json',
        success: function(data){
            contenido_pedido = data
        }
    });
    $("#dialog").dialog("open");
    $("#ui-dialog-title-dialog").html('Productos del Pedido: '+titulo_pedido);
    $("#contenido").append(contenido_pedido);
});
});
```

1.4. Perfil Aprobador.

1.4.1. Contenedor.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Manager_Containers_Manager extends Controller_Template {
```

```
    public $template = 'controller/manager/containers/manager';
```

```
    /**
```

```
     * El atributo que va a ser utilizado en cada acción
```

```
     * NULL por defecto
```

```
     * @var View
```

```
     */
```

```
    public $view;
```

```
    /**
```

```
     * @see system/classes/kohana/controller/Kohana_Controller_Template::before()
```

```
     */
```

```
    public function before()
```

```
    {
```

```
        Session::instance();
```

```
        //Required because we are extending Controller_Template
```

```
        parent::before();
```

```
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario', FALSE))-
```

```
>find();
```

```
        if(!Session::instance()->get('treasure_usuario', FALSE) OR Session::instance()->get('treasure_usuario', FALSE)=="
```

```
OR $user->ID_PERFIL=3)
```

```
        {
```

```
            HTTP::redirect('login');
```

```
        }
```

```
        //$this->_check_permissions();
```

```
    if ( $this->auto_render )
```

```
    {
```

```
        //Sets the "Default" title set on the container
```

```
        $this->template->title = 'Treasure Administrador';
```

```
        $this->template->content = '';
```

```
        $this->template->styles = array();
```

```
        $this->template->scripts = array();
```

```
    }
```

```
    //Cleans the view so that every action can set it
```

```
    //by itself without problems
```

```
    $this->view = NULL;
```

```
    }
```

```
    /**
```

```
     * @see system/classes/kohana/controller/Kohana_Controller_Template::after()
```

```
     */
```

```
    public function after()
```

```
    {
```

```
        if ( $this->auto_render )
```

```
        {
```

```
            $action = $this->request->action();
```

```
            $controller = $this->request->controller();
```

```
            $directory = $this->request->directory();
```

```
            //Build the styles array
```

```
            $css_file = 'media/css/controller/'.$directory.'/'.$controller.'/'.$action.'.css';
```

```
            $styles = array(
```

```
                'media/css/screen.css' => 'screen, projection',
```

```
                'media/css/src/print.css' => 'print',
```

```
                'media/css/jquery-ui/smoothness/jquery-ui-1.8.16.custom.css' => 'all',
```

```
                'media/css/dataTables/TableTools.css' => 'screen',
```

```
                'media/css/dataTables/TableTools_JUI.css' => 'screen',
```

```
                'media/css/dataTables/demo_table_jui.css' => 'screen',
```

```
                'media/css/src/ie.css' => 'screen, projection',
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
'media/css/styles.css' => 'screen',
'media/css/menu.css' => 'screen',
    'media/css/menu-style.css' => 'screen',
'media/css/jquery-ui/jquery.autocomplete.css' => 'all',
);

//VALIDATING FILE
if(file_exists($css_file) )
{
    $styles = array_merge( $styles, array($css_file=>'screen') );
}

//Build the scripts array
$js_file = 'media/js/controller/'.$_DIRECTORY__.$controller.'/'.$_ACTION__.'.js';
$scripts = array(
    'media/js/jqueryLibrary/jquery-1.6.2.min.js',
    'media/js/jqueryLibrary/jquery-ui-1.8.16.custom.min.js',
    'media/js/jqueryLibrary/jquery-validation.js',
        'media/js/jqueryLibrary/jquery.autocomplete.js',
        'media/js/dataTables/jquery.dataTables.js',
    'media/js/dataTables/TableTools.js',
    'media/js/dataTables/ZeroClipboard.js',
        'media/js/dataTables/dataTables.editor.js',
    'media/js/globalfunctions.js',
    'media/js/ajaxupload.js',
    'media/js/system.js',
    'media/js/mbMenu.js',
);

//VALIDATING FILE
if(file_exists($js_file) )
{
    $scripts = array_merge( $scripts, array($js_file) );
}

//Set styles and scripts on the template (container). This can also be
//done from any action method
$this->template->styles = array_merge( $styles, $this->template->styles );
$this->template->scripts = array_merge( $scripts, $this->template->scripts );

//Get Menu
//Uncomment once we have all the menu logic created
$menu=FALSE;//$this->_get_menu( Model_Globalfunctions::get_menu( Session::instance()-
>get('discotecas_serial_prf', FALSE) ) );
if($menu){
    $this->template->menu = $menu;
}else{
    $this->template->menu = false;
}

//If $this->view is set display it on the
//containers $content
$this->template->content = $this->view;
}

//Required because we are extending Controller_Template
parent::after();
}
}
```

1.4.2. Contenedor de Vista.

```
<!doctype html>
<html lang="<?php echo l18n::$lang ?>">
  <head>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/x-icon" href="/treasure/media/img/icon.ico">
    <title><?php echo $title ?></title>
    <?php foreach ($styles as $file => $type) echo HTML::style($file, array('media' => $type)), PHP_EOL ?>
    <?php foreach ($scripts as $file) echo HTML::script($file), PHP_EOL ?>
    <script type="text/javascript">
      var document_root = '<?php echo Url::site();?>';
    </script>
  </head>
  <body>
    <div class="container" id="admin_container">
      <div class="span-24 general_bg">
        <div class="span-24 last " style=" padding-top: 5px;">
          <div class="prepend-23 span-1 last admin_banner">
            <strong><a href="<?php echo Url::site();?>" style="text-decoration:none; color: black; font-weight:
normal;">Salir</a></strong>
          </div>
        </div>

        <div class="span-24 ">
          <?php echo Helper_Manager_Menu::menu(); ?>
        </div>
      </div>
      <div class="span-24 content">
        <div class="prepend-2 span-20 append-2 line last" id="container-flash-messages" style=" padding-top:
15px;">
          <?php FlashMessenger::factory()->get_messages(); ?>
        </div>
        <?php echo $content ?>
      </div>
      <div id="spinner" style="display: none;">
        <div id="ajax_loader"></div>
        <div class="ajax_loader_text">Cargando...</div>
      </div>
    </div>
  </body>
</html>
```

1.4.3. Controlador de Pedido.

```
<?php
class Controller_Manager_Pedido extends Controller_Manager_Containers_Manager {

    public function action_index()
    {
        HTTP::redirect('manager/pedido/create/');
    }

    public function action_create()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/manager/pedido/create');

        //IF USER SUBMITTED FORM
        $ID_PEDIDO = $this->request->param('params');
        $pedido = ORM::factory('pedido')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find();
        $producto = ORM::factory('producto')->order_by('DESCRIPCION')->find_all()->as_array('ID_PRODUCTO',
'DESCRIPCION');
        $this->view->set('producto', $producto);
        $user = ORM::factory('user')->where('ID_USUARIO', '=', $pedido->ID_USUARIO)->find();
        $this->view->set('user', $user->ID_USUARIO);
        if($ID_PEDIDO)
        {
            $PXP = ORM::factory('pedidodespachoproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find_all()-
>as_array('ID_PRODUCTO', 'CANTIDAD');
            $productos = count($PXP);
        }
        else
        {
            $PXP = "";
            $productos = 0;
        }
        $this->view->set('PXP', $PXP);
        $this->view->set('productos', $productos);

        //VALIDATE POST
        if( $this->request->post() )
        {
            try
            {
                $values= $this->request->post();
                if($ID_PEDIDO)
                {
                    $Eliminapxs = ORM::factory('pedidodespachoproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)
->find_all();
                    foreach ($Eliminapxs as $Eliminapxp) {
                        DB::delete('pedido_despacho_x_producto')->where('ID_PDXP', '=', $Eliminapxp->ID_PDXP)
->execute(Database::instance());
                    }
                }

                $productosIngreso = array();
                foreach ($values as $key => $value) {
                    if(strpos($key, 'CANTIDAD_')!== false)
                    {
                        $aux = explode('_', $key);
                        $productosIngreso = $productosIngreso + array($aux[1]==>$value);
                    }
                }
                if(!$ID_PEDIDO)
                {
                    $tiempo = array('d'=>(int)date('d'),'m'=>(int)date('m'),'Y'=>(int)date('Y'),'H'=>(int)date('H'),'i'=>(int)date('i'));
                    $pedido->ID_USUARIO = Session::instance()->get('treasure_usuario');
                    $pedido->DIA = $tiempo['d'];
                }
            }
        }
    }
}
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
$pedido->MES = $tiempo['m'];
$pedido->ANIO = $tiempo['Y'];
$pedido->HORA = $tiempo['H'];
$pedido->MINUTO = $tiempo['i'];
$pedido->save();
```

```
$pedido2 = ORM::factory('pedido')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))
->and_where('DIA', '=', $tiempo['d'])
->and_where('MES', '=', $tiempo['m'])
->and_where('ANIO', '=', $tiempo['Y'])
->and_where('HORA', '=', $tiempo['H'])
->and_where('MINUTO', '=', $tiempo['i'])
->find();
foreach ($productosIngreso as $key=>$value) {
    $PXP2 = ORM::factory('pedidodespachoxproducto');
    $PXP2->ID_PEDIDO = $pedido2->ID_PEDIDO;
    $PXP2->ID_PRODUCTO = $key;
    $PXP2->CANTIDAD = $value;
    $PXP2->ESTADO = 'PEDIDO';
    $PXP2->save();
}
}
else
{
    $pedido2 = ORM::factory('pedido')->where('ID_USUARIO', '=', $pedido->ID_USUARIO)
->and_where('DIA', '=', $pedido->DIA)
->and_where('MES', '=', $pedido->MES)
->and_where('ANIO', '=', $pedido->ANIO)
->and_where('HORA', '=', $pedido->HORA)
->and_where('MINUTO', '=', $pedido->MINUTO)
->find();
    foreach ($productosIngreso as $key=>$value) {
        $PXP2 = ORM::factory('pedidodespachoxproducto');
        $PXP2->ID_PEDIDO = $pedido2->ID_PEDIDO;
        $PXP2->ID_PRODUCTO = $key;
        $PXP2->CANTIDAD = $value;
        $PXP2->ESTADO = 'PEDIDO';
        $PXP2->save();
    }
}
}
```

```
if($ID_PEDIDO)
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:pedido:create:update'));
else
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:pedido:create:success'));
}
catch(Exception $e)
{
    echo $e->getMessage();
    FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('manager/pedido/select/');
}
$this->view->set('pedido', $pedido );
}
```

```
public function action_select()
{
    $this->view = View::factory('controller/manager/pedido/select');
    $pedidosp = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM
'pedido_despacho_x_producto' WHERE 'ESTADO'='PEDIDO' AND 'ID_DESPACHO' is NULL")->execute()->as_array();
    $pedido = array();
    foreach ($pedidosp as $value) {
        $pedidos = ORM::factory('pedido')->where('ID_PEDIDO', '=', $value)->find();
        $pedido = $pedido + array($pedidos->ID_PEDIDO=>$pedidos->as_array());
    }
    $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
    $user = array();
    foreach ($users as $value) {
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

        $nombres = $value->NOMBRE.'  
'; $value->APELLIDO;
        $user = $user + array($value->ID_USUARIO=>$nombres);
    }

    if(! count($pedido))
        $pedido=NULL;
    $this->view->set('user', $user);
    $this->view->set('pedido', $pedido);
}

public function action_aprobar()
{
    $this->view = View::factory('controller/manager/pedido/aprobar');
    $pedidosp = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM
    'pedido_despacho_x_producto' WHERE 'ESTADO'='PEDIDO' AND 'ID_DESPACHO' is NULL")->execute()->as_array();
    $pedido = array();
    foreach ($pedidosp as $value) {
        $pedidos = ORM::factory('pedido')->where('ID_PEDIDO', '=', $value->find());
        $pedido = $pedido + array($pedidos->ID_PEDIDO=>$pedidos->as_array());
    }
    $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
    $user = array();
    foreach ($users as $value) {
        $nombres = $value->NOMBRE.'  
'; $value->APELLIDO;
        $user = $user + array($value->ID_USUARIO=>$nombres);
    }

    if(! count($pedido))
        $pedido=NULL;
    $this->view->set('user', $user);
    $this->view->set('pedido', $pedido);
}

public function action_revisar()
{
    //IF USER SUBMITTED FORM
    $this->view = View::factory( 'controller/manager/pedido/revisar' );

    //IF USER SUBMITTED FORM
    $ID_PEDIDO = $this->request->param('params');
    $pedido = ORM::factory('pedido')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find();
    $producto = ORM::factory('producto')->order_by('DESCRIPCION')->find_all()->as_array('ID_PRODUCTO',
'DESCRIPCION');
    $this->view->set( 'producto', $producto );
    if($ID_PEDIDO)
    {
        $PXP = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find_all()-
>as_array('ID_PRODUCTO', 'CANTIDAD' );
    }
    else
    {
        $PXP = "";
    }
    $this->view->set( 'PXP', $PXP );

    //VALIDATE POST
    if( $this->request->post() )
    {
        try
        {
            $values= $this->request->post();
            if(isset($values['btnCancel']))
            {
                $pedidos = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)-
>and_where('ESTADO', '=', 'PEDIDO')->find_all();
                foreach ($pedidos as $value) {

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
        DB::update('pedido_despacho_x_producto')->set(array('ESTADO' => 'CANCELADO'))->where('ID_PDXP', '=', $value->ID_PDXP)->execute();
    }
    FlashMessenger::factory()->set_message('error', 'Se ha Cancelado el Pedido exitosamente.');
```

```
    }
    else
    {
        $pedidos = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)->and_where('ESTADO', '=', 'PEDIDO')->find_all();
        foreach ($pedidos as $value) {
            DB::update('pedido_despacho_x_producto')->set(array('ESTADO' => 'APROBADO'))->where('ID_PDXP', '=', $value->ID_PDXP)->execute();
        }
        FlashMessenger::factory()->set_message('success', Kohana::message('admin', 'admin:pedido:revisar:aprobado'));
```

```
    }
    }
    catch(Exception $e)
    {
        echo $e->getMessage();
        FlashMessenger::factory()->set_message('error', $e->getMessage());
    }
    HTTP::redirect('manager/pedido/aprobar/');
```

```
    }
    $this->view->set('pedido', $pedido );
}
?>
```

1.4.4. Controlador de Main.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Manager_Main extends Controller_Manager_Containers_Manager {
```

```
    public function action_index()
    {
        $option = $this->request->param('params');
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))->find();
        $name = ".strtoupper ($user->NOMBRE)";
        $this->view = View::factory( 'controller/manager/main' );
        $this->view->set('name',$name);
    }
}
```

1.4.5. Helper

1.4.5.1. Menú.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Helper_Manager_Menu {  
    public static function menu($seg1 = null, $seg2 = null, $seg3 = null)  
    {  
        $items = array();  
  
        /*PEDIDOS*/  
        $items[] = array('id' => 1, 'parent_id' => 0, 'title' => 'Pedidos', 'link' => '#');  
        $items[] = array('id' => 2, 'parent_id' => 1, 'title' => 'Aprobar', 'link' => 'manager/pedido/aprobar');
```

```
        // Build the current path  
        $current = "";  
        $current .= $seg1 ? $seg1 : "";  
        $current .= $seg2 ? ('/' . $seg2): "";  
        $current .= $seg3 ? ('/' . $seg3): "";  
  
        // Let's get the body of the current page  
        //$body = ORM::factory('entry')->where('link','=', $current)->find()->body;  
        $options = array('link_prepend' => "", 'current_path' => $current);  
  
        return MenuStructure::factory($items, $options)->get_menu();  
    }  
}
```

```
?>
```

1.4.6. Vistas de Pedido.

1.4.6.1. Aprobar.

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Pedido Para Aprobaci&oacute;n</div>
  <?php if($pedido): ?>
  <div class="span-24"><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">N&Uacute;MERO DE PEDIDO</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">REVISAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($pedido as $ped): ?>
          <tr>
            <td><?php echo $ped["ID_PEDIDO"] ?></td>
            <td><?php echo $user[$ped["ID_USUARIO"]]?></td>
            <td><?php echo $ped["ANIO"]?></td>
            <td><?php echo $ped["MES"]?></td>
            <td><?php echo $ped["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('manager/pedido/revisar/'. $ped["ID_PEDIDO"]); ?>">
                </img>
              </a>
            </td>
          </tr>
        <?php endforeach; ?>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br/>
  <div class="alerts" style="text-align: center">
    No existen Pedidos para procesar.
  </div>
  </div>
  <?php endif; ?>
</div>
```

1.4.6.2. *Revisar.*

```

<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title">Aprobaci&oacute;n o Cancelaci&oacute;n de Pedido</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>

    <div class="prepend-5 span-10" align="center">
      <div id="dialog_errores">
        <ul id="alerts" class="alerts"></ul>
      </div>
    </div>

    <br/><br/><br/>

    <form action="" method="post" name="frmPedido" id="frmPedido">
      <?php if($pedido->ID_PEDIDO): ?>
        <?php foreach ($producto as $key => $value): ?>
          <?php if(array_key_exists($key,$PXP)): ?>
            <div class="span-20 line ">
              <div class="span-10">
                <?php echo Form::checkbox(".$key.", $key, FALSE, array('id'=>'ID_PRODUCTO_'. $key.', 'class'
=> 'span-5','checked' => 'checked','disabled' => 'disable' ));?>
                <div class="span-5 label"> <?php echo $value?> </div>
              </div>
              <div class="span-10 last">
                <div class="prepend-2 span-3 label"> *Cantidad: </div>
                <input type="text" class="span-1 last" name="CANTIDAD_<?php echo $key?>"
id='CANTIDAD_<?php echo $key?>' value="<?php echo $PXP[$key]?>" title="El campo "Cantidad" es obligatorio."
disabled="disable" maxlength="2"/>
              </div>
            </div>
          <?php endforeach; ?>
        <?php endif; ?>
        <input type="hidden" name="id" id="id" value="<?php echo ($pedido->ID_PEDIDO)?$pedido-
>ID_PEDIDO:;?>"/>
        <div class="span-20 " align="center"><br>
          <input type="submit" id="btnSave" name="btnSave" value="Aprobar" />&emsp;&emsp;
          <input type="submit" id="btnCancel" name="btnCancel" value="Cancelar" />
        </div>
      </form>
      <div class="span-20 " align="center"><br>
        <a href="javascript:window.history.go(-1);">Regresar</a>
      </div>
    </div>
  </div>

```

1.4.7. *Vistas de Main.*

```
<div class="span-24">
  <div class="prepend-2 span-20">

    <div class="prepend-5 spen-20 title_main_page center">
      PAGINA DE APROBADOR<br><br>
      BIENVENIDO <?php echo $name?>
    </div>
  </div>
</div>
```

1.4.8. Archivos JS de Pedido.

1.4.8.1. Aprobar.

```
$(document).ready(function(){
    $('#tblPedidos').dataTable({
        "bJQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPaginate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltimea",
                "sNext": "Siguiente",
                "sPrevious": "Anterior"
            },
            "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
            "sZeroRecords": "No se encontraron resultados",
            "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
            "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
            "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
            "sSearch": "Criterio de B&uacute;squeda:",
            "sProcessing": "Filtrando.."
        },
        "aoColumns": [
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": true},
            {"bSortable": false}
        ],
        "sDom": '<H"Tr>t<"F"pli>',
        "oTableTools": {
            "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
            "aButtons": [
                {"sExtends": "xls",
                "sButtonText": "Exportar a Excel"}
            ]
        }
    });
});
```

1.4.8.2. Revisar.

```
$(document).ready(function(){
```

```
    $('#btnSave').bind('click',function(){
        if(!confirm('¿Está seguro de cambiar estado de Pedido a Aprobado?','Confirmar Accion'))
        {
            return false;
        }
    });

    $('#btnCancel').bind('click',function(){
        if(!confirm('¿Está seguro de cambiar estado de Pedido a Cancelado?','Confirmar Accion'))
        {
            return false;
        }
    });
});
```

1.5. Perfil Proveedor.

1.5.1. Contenedor.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Supplier_Containers_Supplier extends Controller_Template {
```

```
    public $template = 'controller/supplier/containers/supplier';
```

```
    /**
```

```
     * El atributo que va a ser utilizado en cada acción
```

```
     * NULL por defecto
```

```
     * @var View
```

```
     */
```

```
    public $view;
```

```
    /**
```

```
     * @see system/classes/kohana/controller/Kohana_Controller_Template::before()
```

```
     */
```

```
    public function before()
```

```
    {
```

```
        Session::instance();
```

```
        //Required because we are extending Controller_Template
```

```
        parent::before();
```

```
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario', FALSE))-
```

```
>find();
```

```
        if(!Session::instance()->get('treasure_usuario', FALSE) OR Session::instance()->get('treasure_usuario', FALSE)=="
```

```
OR $user->ID_PERFIL=5)
```

```
        {
```

```
            HTTP::redirect('login');
```

```
        }
```

```
        //$this->_check_permissions();
```

```
        if ( $this->auto_render )
```

```
        {
```

```
            //Sets the "Default" title set on the container
```

```
            $this->template->title = 'Treasure';
```

```
            $this->template->content = '';
```

```
            $this->template->styles = array();
```

```
            $this->template->scripts = array();
```

```
        }
```

```
        //Cleans the view so that every action can set it
```

```
        //by itself without problems
```

```
        $this->view = NULL;
```

```
    }
```

```
    /**
```

```
     * @see system/classes/kohana/controller/Kohana_Controller_Template::after()
```

```
     */
```

```
    public function after()
```

```
    {
```

```
        if ( $this->auto_render )
```

```
        {
```

```
            $action = $this->request->action();
```

```
            $controller = $this->request->controller();
```

```
            $directory = $this->request->directory();
```

```
            //Build the styles array
```

```
            $css_file = 'media/css/controller/'.$directory.'/'.$controller.'/'.$action.'.css';
```

```
            $styles = array(
```

```
                'media/css/screen.css' => 'screen, projection',
```

```
                'media/css/src/print.css' => 'print',
```

```
                'media/css/jquery-ui/smoothness/jquery-ui-1.8.16.custom.css' => 'all',
```

```
                'media/css/dataTables/TableTools.css' => 'screen',
```

```
                'media/css/dataTables/TableTools_JUI.css' => 'screen',
```

```
                'media/css/dataTables/demo_table_jui.css' => 'screen',
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
'media/css/src/ie.css' => 'screen, projection',
'media/css/styles.css' => 'screen',
'media/css/menu.css' => 'screen',
    'media/css/menu-style.css' => 'screen',
'media/css/jquery-ui/jquery.autocomplete.css' => 'all',
);

//VALIDATING FILE
if(file_exists($css_file) )
{
    $styles = array_merge( $styles, array($css_file=>'screen') );
}

//Build the scripts array
$js_file = 'media/js/controller/'.$_.$directory.'/'.$_.$controller.'/'.$_.$action.'.js';
$scripts = array(
    'media/js/jqueryLibrary/jquery-1.6.2.min.js',
    'media/js/jqueryLibrary/jquery-ui-1.8.16.custom.min.js',
    'media/js/jqueryLibrary/jquery-validation.js',
        'media/js/jqueryLibrary/jquery.autocomplete.js',
    'media/js/dataTables/jquery.dataTables.js',
    'media/js/dataTables/TableTools.js',
    'media/js/dataTables/ZeroClipboard.js',
        'media/js/dataTables/dataTables.editor.js',
    'media/js/globalfunctions.js',
    'media/js/ajaxupload.js',
    'media/js/system.js',
    'media/js/mbMenu.js',
);

//VALIDATING FILE
if(file_exists($js_file) )
{
    $scripts = array_merge( $scripts, array($js_file) );
}

//Set styles and scripts on the template (container). This can also be
//done from any action method
$this->template->styles = array_merge( $styles, $this->template->styles );
$this->template->scripts = array_merge( $scripts, $this->template->scripts );

//Get Menu
//Uncomment once we have all the menu logic created
$menu=FALSE;//$this->_get_menu( Model_Globalfunctions::get_menu( Session::instance()-
>get('discotecas_serial_prf', FALSE) ) );
if($menu){
    $this->template->menu = $menu;
}else{
    $this->template->menu = false;
}

//If $this->view is set display it on the
//containers $content
$this->template->content = $this->view;
}

//Required because we are extending Controller_Template
parent::after();
}
}
```

1.5.2. Contenedor Vista.

```
<!doctype html>
<html lang="<?php echo l18n::$lang ?>">
  <head>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/x-icon" href="/treasure/media/img/icon.ico">
    <title><?php echo $title ?></title>
    <?php foreach ($styles as $file => $type) echo HTML::style($file, array('media' => $type)), PHP_EOL ?>
    <?php foreach ($scripts as $file) echo HTML::script($file), PHP_EOL ?>
    <script type="text/javascript">
      var document_root = '<?php echo Url::site();?>';
    </script>
  </head>
  <body>
    <div class="container" id="admin_container">
      <div class="span-24 general_bg">
        <div class="span-24 last " style=" padding-top: 5px;">
          <div class="prepend-23 span-1 last admin_banner">
            <strong><a href="<?php echo Url::site();?>" style="text-decoration:none; color: black; font-weight:
normal;">Salir</a></strong>
          </div>
        </div>

        <div class="span-24 ">
          <?php echo Helper_Supplier_Menu::menu(); ?>
        </div>
      </div>
      <div class="span-24 content">
        <div class="prepend-2 span-20 append-2 line last" id="container-flash-messages" style=" padding-top:
15px;">
          <?php FlashMessenger::factory()->get_messages(); ?>
        </div>
        <?php echo $content ?>
      </div>
      <div id="spinner" style="display: none;">
        <div id="ajax_loader"></div>
        <div class="ajax_loader_text">Cargando...</div>
      </div>
    </div>
  </body>
</html>
```

1.5.3. Controlador Despacho.

```

<?php
class Controller_Supplier_Despacho extends Controller_Supplier_Containers_Supplier {

    public function action_index()
    {
        HTTP::redirect('supplier/despacho/create/');
    }

    public function action_create()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/supplier/despacho/create' );

        //IF USER SUBMITTED FORM
        $ID_DESPACHO = $this->request->param('params');
        if($ID_DESPACHO)
            $pedidos = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM `pedido_despacho_x_producto`
            WHERE `ESTADO`='EN PROCESO' AND `ID_DESPACHO` = ".$ID_DESPACHO.">execute()->as_array();
        else
            $pedidos = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM `pedido_despacho_x_producto`
            WHERE `ESTADO`='EN PROCESO' AND `ID_DESPACHO` is NULL")>execute()->as_array();
        $this->view->set( 'pedidos', $pedidos);
        $pedidos_sin_despacho = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM
        `pedido_despacho_x_producto` WHERE `ESTADO`='APROBADO'")>execute()->as_array();
        $this->view->set( 'pedidos_sin_despacho', $pedidos_sin_despacho);
        $despacho = ORM::factory('despacho')->where('ID_DESPACHO', '=', $ID_DESPACHO)->find();
        $this->view->set( 'despacho', $despacho );
        $user = ORM::factory('user')->where('ID_USUARIO', '=', $despacho->ID_USUARIO)->find();
        $this->view->set( 'user', $user->ID_USUARIO );
        $validar = count($pedidos);
        $this->view->set( 'validar', $validar );

        //VALIDATE POST
        if( $this->request->post() )
        {

            try
            {

                $values= $this->request->post();
                if($ID_DESPACHO)
                {
                    $Liberapxps = ORM::factory('pedidodespachoxproducto')->where('ID_DESPACHO', '=', $ID_DESPACHO)
                    ->find_all();
                    foreach ($Liberapxps as $Liberapxp) {
                        DB::update('pedido_despacho_x_producto')-
                        >set(array('ID_DESPACHO'=>NULL,'ESTADO'=>'APROBADO'))->where('ID_PDXP', '=', $Liberapxp->ID_PDXP)
                        ->execute();
                    }
                }

                $pedidoProceso = array();
                foreach ($values as $value) {
                    if(strpos($value, 'ID_PEDIDO_')!== false)
                    {
                        $aux = explode('_', $value);
                        $pedidoProceso = $pedidoProceso + array($aux[2]=>$aux[2]);
                    }
                }
                if(!$ID_DESPACHO)
                {
                    $tiempo = array('d'=>(int)date('d'),'m'=>(int)date('m'),'Y'=>(int)date('Y'),'H'=>(int)date('H'),'i'=>(int)date('i'));
                    $despacho->ID_USUARIO = Session::instance()->get('treasure_usuario');
                    $despacho->DIA = $tiempo['d'];
                    $despacho->MES = $tiempo['m'];
                    $despacho->ANIO = $tiempo['Y'];
                    $despacho->HORA = $tiempo['H'];
                    $despacho->MINUTO = $tiempo['i'];
                }
            }
            catch (Exception $e)
            {
                $this->view->set('error', $e->getMessage());
            }
        }
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

$despacho->save();

$pedido2 = ORM::factory('despacho')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))
->and_where('DIA', '=', $tiempo['d'])
->and_where('MES', '=', $tiempo['m'])
->and_where('ANIO', '=', $tiempo['Y'])
->and_where('HORA', '=', $tiempo['H'])
->and_where('MINUTO', '=', $tiempo['i'])
->find();
foreach ($pedidoProceso as $value) {
    DB::update('pedido_despacho_x_producto')->set(array('ID_DESPACHO'=>$pedido2-
>ID_DESPACHO,'ESTADO'=>'EN PROCESO'))->where('ID_PEDIDO', '=', $value)
->execute();
}
}
else
{
    $pedido2 = ORM::factory('despacho')->where('ID_USUARIO', '=', $despacho->ID_USUARIO)
->and_where('DIA', '=', $despacho->DIA)
->and_where('MES', '=', $despacho->MES)
->and_where('ANIO', '=', $despacho->ANIO)
->and_where('HORA', '=', $despacho->HORA)
->and_where('MINUTO', '=', $despacho->MINUTO)
->find();
foreach ($pedidoProceso as $value) {
    DB::update('pedido_despacho_x_producto')->set(array('ID_DESPACHO'=>$pedido2-
>ID_DESPACHO,'ESTADO'=>'EN PROCESO'))->where('ID_PEDIDO', '=', $value)
->execute();
}
}

if($ID_DESPACHO)
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:create:update'));
else
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:create:success'));
}
catch(Exception $e)
{
    echo $e->getMessage();
    FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('supplier/despacho/select');
}
$this->view->set( 'despacho', $despacho );
}

public function action_get_content()
{
    $content="";
    if($this->request->is_ajax())
    {
        $pedido_id = arr::get($this->request->post(), 'pedido_id');
        try
        {
            $closed = ORM::factory('pedidodespachoxproducto')
->where('ID_PEDIDO', '=', $pedido_id)
->find_all()->as_array();
            foreach ($closed as $value) {
                $nombre_producto = ORM::factory('producto')->where('ID_PRODUCTO', '=', $value->ID_PRODUCTO)-
>find();
                $content = $content.'<div class="span-17 line"><div class="span-8 label"> '.$nombre_producto-
>DESCRIPCION.'</div><div class="span-8 label"> Cantidad: '.$value->CANTIDAD.'</div> </div> ';
            }
            echo json_encode($content);
        }
        catch(Exception $e)
        {
            echo json_encode($e->getMessage());
        }
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

        $this->auto_render = FALSE;
    }
    else
    {
        HTTP::redirect('supplier/main');
    }
}

public function action_select()
{
    $this->view = View::factory('controller/supplier/despacho/select');
    $despachosp = DB::query(Database::SELECT, "SELECT Distinct('ID_DESPACHO') FROM `pedido_despacho_x_producto` WHERE `ESTADO`='EN PROCESO' AND `ID_DESPACHO` is not NULL")->execute()->as_array();
    $despacho = array();
    foreach ($despachosp as $value) {
        $despachos = ORM::factory('despacho')->where('ID_DESPACHO', '=', $value)->find();
        $despacho = $despacho + array($despachos->ID_DESPACHO=>$despachos->as_array());
    }
    $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
    $user = array();
    foreach ($users as $value) {
        $nombres = $value->NOMBRE.'<br>'. $value->APELLIDO;
        $user = $user + array($value->ID_USUARIO=>$nombres);
    }

    if(! count($despacho))
        $despacho=NULL;
    $this->view->set('user', $user);
    $this->view->set('despacho', $despacho);
}

public function action_procesar()
{
    $this->view = View::factory('controller/supplier/despacho/procesar');
    $despachosp = DB::query(Database::SELECT, "SELECT Distinct('ID_DESPACHO') FROM `pedido_despacho_x_producto` WHERE `ESTADO` in ('EN PROCESO','COMPRADO','ENVIADO') AND `ID_DESPACHO` is not NULL")->execute()->as_array();
    $despacho = array();
    foreach ($despachosp as $value) {
        $despachos = ORM::factory('despacho')->where('ID_DESPACHO', '=', $value)->find();
        $despacho = $despacho + array($despachos->ID_DESPACHO=>$despachos->as_array());
    }
    $users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
    $user = array();
    foreach ($users as $value) {
        $nombres = $value->NOMBRE.'<br>'. $value->APELLIDO;
        $user = $user + array($value->ID_USUARIO=>$nombres);
    }
    if(! count($despacho))
        $despacho=NULL;
    $this->view->set('user', $user);
    $this->view->set('despacho', $despacho);
}

public function action_revisar()
{
    //IF USER SUBMITTED FORM
    $this->view = View::factory('controller/supplier/despacho/revisar' );

    //IF USER SUBMITTED FORM
    $ID_DESPACHO = $this->request->param('params');
    $pedidos = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM `pedido_despacho_x_producto` WHERE `ESTADO` in ('EN PROCESO','COMPRADO','ENVIADO') AND `ID_DESPACHO` = ".$ID_DESPACHO."")->execute()->as_array();
    $this->view->set('pedidos', $pedidos);
    $despacho = ORM::factory('despacho')->where('ID_DESPACHO', '=', $ID_DESPACHO)->find();
    $this->view->set('despacho', $despacho );
    $user = ORM::factory('user')->where('ID_USUARIO', '=', $despacho->ID_USUARIO)->find();
    $this->view->set('user', $user->ID_USUARIO );
    $validar = count($pedidos);
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```

$this->view->set( 'validar', $validar );

//VALIDATE POST
if( $this->request->post() )
{
    try
    {
        $values= $this->request->post();
        $proceso = 0;
        $pedidoProceso = array();
        echo Debug::vars($values);
        foreach ( $values as $key=>$value ) {
            if(strpos($value, 'Comprar')!= false)
            {
                $proceso = 1;
            }
            if(strpos($value, 'Enviar')!= false)
            {
                $proceso = 2;
            }
            if(strpos($value, 'Entregar')!= false)
            {
                $proceso = 3;
            }
            if(strpos($value, 'Cancelar')!= false)
            {
                $proceso = 4;
            }
            if(strpos($key, 'ID_PEDIDO_')!= false)
            {
                $aux = explode('_', $key);
                $pedidoProceso = $pedidoProceso + array($aux[2]=>$aux[2]);
            }
        }
        switch ( $proceso )
        {
            case 1:foreach ( $pedidoProceso as $value ) {
                DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'COMPRADO'))->where('ID_PEDIDO', '=', $value)
->execute();
            }
            FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:comprado'));
            break;
            case 2:foreach ( $pedidoProceso as $value ) {
                DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'ENVIADO'))->where('ID_PEDIDO', '=', $value)
->execute();
            }
            FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:enviado'));
            break;
            case 3:foreach ( $pedidoProceso as $value ) {
                $producto = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $value)->find_all();
                foreach ( $producto as $value2 ) {
                    $cantidad = ORM::factory('productoxbodega')->where('ID_PRODUCTO', '=', $value2-
>ID_PRODUCTO)->find();
                    $valor = $cantidad->CANTIDAD + $value2->CANTIDAD;
                    DB::update('producto_x_bodega')->set(array('CANTIDAD'=>$valor,))->where('ID_PRODUCTO',
'=', $value2->ID_PRODUCTO)->execute();
                }
                DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'ENTREGADO'))->where('ID_PEDIDO', '=', $value)
->execute();
            }
            FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:despacho:estado:entregado'));
            break;
            case 4:foreach ( $pedidoProceso as $value ) {

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
DB::update('pedido_despacho_x_producto')-
>set(array('ID_DESPACHO'=>$ID_DESPACHO,'ESTADO'=>'CANCELADO'))->where('ID_PEDIDO', '=', $value)
->execute();
}
FlashMessenger::factory()->set_message('error', Kohana::message('admin',
'admin:despacho:estado:cancelado'));
break;
default : HTTP::redirect('supplier/main');
break;
}
}
catch(Exception $e)
{
echo $e->getMessage();
FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('supplier/despacho/procesar/');
}
$this->view->set( '$despacho', $despacho );
}
?>
```

1.5.4. Controlador Main.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_Supplier_Main extends Controller_Supplier_Containers_Supplier {
```

```
    public function action_index()
    {
        $option = $this->request->param('params');
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))->find();
        $name = ".strtoupper ($user->NOMBRE)";
        $this->view = View::factory( 'controller/supplier/main' );
        $this->view->set('name',$name);
    }
}
```

1.5.5. Helper

1.5.5.1. Menú.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Helper_Supplier_Menu {
    public static function menu($seg1 = null, $seg2 = null, $seg3 = null)
    {
        $items = array();

        /*DESPACHOS*/
        $items[] = array('id' => 1, 'parent_id' => 0, 'title' => 'Despachos', 'link' => '#');
        $items[] = array('id' => 2, 'parent_id' => 1, 'title' => 'Nuevo', 'link' => 'supplier/despacho/create');
        $items[] = array('id' => 3, 'parent_id' => 1, 'title' => 'Modificar', 'link' => 'supplier/despacho/select');
        $items[] = array('id' => 4, 'parent_id' => 1, 'title' => 'Procesar', 'link' => 'supplier/despacho/procesar');

        // Build the current path
        $current = "";
        $current .= $seg1 ? $seg1 : "";
        $current .= $seg2 ? ('/' . $seg2) : "";
        $current .= $seg3 ? ('/' . $seg3) : "";

        // Let's get the body of the current page
        //$body = ORM::factory('entry')->where('link','=', $current)->find()->body;
        $options = array('link_prepend' => '', 'current_path' => $current);

        return MenuStructure::factory($items, $options)->get_menu();
    }
}
?>
```

1.5.6. Vistas de Despacho.

1.5.6.1. Create.

```

<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title"><?php echo ($despacho->ID_DESPACHO)?'Actualizar':'Nuevo';?> Despacho</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>
    <div class="prepend-5 span-10" align="center">
      <div id="dialog_errores">
        <ul id="alerts" class="alerts"></ul>
      </div>
    </div>
    <div id="dialog" title="">
      <p id="pedido_productos">
        <div id="contenido"></div>
      </p>
    </div>
    <form action="" method="post" name="frmDespacho" id="frmDespacho">
      <?php if($despacho->ID_DESPACHO):?>
      <?php foreach ($pedidos as $value):?>
        <div class="span-20 line">
          <div class="prepend-2 span-5 label"> N&Uacute;mero DE PEDIDO: </div>
          <div class="prepend-2 span-10">
            <?php echo Form::checkbox("$value['ID_PEDIDO'].", 'ID_PEDIDO_'.$value['ID_PEDIDO'].",
FALSE, array('id'=>'ID_PEDIDO_'.$value['ID_PEDIDO'].", 'class' => 'span-5', 'checked' => 'checked')));?>
            <div class="span-3 label"> <?php echo $value['ID_PEDIDO'];?> </div>
            <a class='des-pedido-link' pedido_id='<?php echo $value['ID_PEDIDO'];?>' href="#">Ver
Productos</a>
          </div>
        </div>
      <?php endforeach;?>
      <?php foreach ($pedidos_sin_despacho as $value):?>
        <div class="span-20 line">
          <div class="prepend-2 span-5 label"> N&Uacute;mero DE PEDIDO: </div>
          <div class="prepend-2 span-10">
            <?php echo Form::checkbox("$value['ID_PEDIDO'].", 'ID_PEDIDO_'.$value['ID_PEDIDO'].",
FALSE, array('id'=>'ID_PEDIDO_'.$value['ID_PEDIDO'].", 'class' => 'span-5')));?>
            <div class="span-3 label"> <?php echo $value['ID_PEDIDO'];?> </div>
            <a class='des-pedido-link' pedido_id='<?php echo $value['ID_PEDIDO'];?>' href="#">Ver
Productos</a>
          </div>
        </div>
      <?php endforeach;?>
      <?php else: ?>
      <?php foreach ($pedidos_sin_despacho as $value):?>
        <div class="span-20 line">
          <div class="prepend-2 span-5 label"> N&Uacute;mero DE PEDIDO: </div>
          <div class="prepend-2 span-10">
            <?php echo Form::checkbox("$value['ID_PEDIDO'].", 'ID_PEDIDO_'.$value['ID_PEDIDO'].",
FALSE, array('id'=>'ID_PEDIDO_'.$value['ID_PEDIDO'].", 'class' => 'span-5')));?>
            <div class="span-3 label"> <?php echo $value['ID_PEDIDO'];?> </div>
            <a class='des-pedido-link' pedido_id='<?php echo $value['ID_PEDIDO'];?>' href="#">Ver
Productos</a>
          </div>
        </div>
      <?php endforeach;?>
      <?php endif;?>
      <input type="hidden" name="validar" id="validar" value="<?php echo ($validar)?$validar:0;?>" />
      <input type="hidden" name="user" id="user" value="<?php echo ($user)?$user:"?>" />
      <div class="span-20" align="center"><br>
        <input type="submit" id="btnSave" name="btnSave" value="Guardar" />
      </div>
    </form>
  </div>
</div>

```

1.5.6.2. *Procesar.*

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Despacho Para Procesar</div>
  <?php if($despacho): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('admin/despacho/create');?>">Crear Despacho</a><br><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">N&Uacute;MERO DE DESPACHO</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">REVISAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($despacho as $desp): ?>
          <tr>
            <td><?php echo $desp["ID_DESPACHO"] ?></td>
            <td><?php echo $user[$desp["ID_USUARIO"]]?></td>
            <td><?php echo $desp["ANIO"]?></td>
            <td><?php echo $desp["MES"]?></td>
            <td><?php echo $desp["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('supplier/despacho/revisar/'. $desp["ID_DESPACHO"]); ?>">
                </img>
              </a>
            </td>
          </tr>
        </tr>
      </tbody>
    </table>
  </div>
  <?php elseif: ?>
  <div class="prepend-7 span-10 append-7 line last"><br/>
  <div class="alerts" style="text-align: center">
    No existen Despachos registrados.
  </div>
  </div>
  <?php endif; ?>
</div>
```


1.5.6.4. Select.

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Despacho Para Modificar</div>
  <?php if($despacho): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('supplier/despacho/create');?>">Crear Despacho</a><br><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">N&Uacute;MERO DE DESPACHO</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">EDITAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($despacho as $desp): ?>
          <tr>
            <td><?php echo $desp["ID_DESPACHO"] ?></td>
            <td><?php echo $user[$desp["ID_USUARIO"]]?></td>
            <td><?php echo $desp["ANIO"]?></td>
            <td><?php echo $desp["MES"]?></td>
            <td><?php echo $desp["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('supplier/despacho/create/'.$desp["ID_DESPACHO"]);?>">
                </img>
              </a>
            </td>
          </tr>
        </tr>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Despachos registrados.
    </div>
  </div>
  <?php endif; ?>
</div>
```

1.5.7. *Vistas de Main.*

```
<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="prepend-5 spen-20 title_main_page center">
      PAGINA DE PROVEEDOR<br><br>
      BIENVENIDO <?php echo $name?>
    </div>
  </div>
</div>
```

1.5.8. Archivos JS de Despacho.

1.5.8.1. Create.

```
$(document).ready(function(){
    var validacheck=$( "#validar" ).val();

    /*FORM VALIDATIONS*/
    $("#frmDespacho").validate({
        errorLabelContainer: "#alerts",
        wrapper: "i",
        onfocusout: false,
        onkeyup: false,
        async: false,
        rules: {
            validar: {
                required: true,
                min: true
            }
        },
        messages: {
            validar: {
                required: 'Debe elegir por lo menos un pedido',
                min: 'Debe elegir por lo menos un pedido'
            }
        },
        submitHandler: function(form) {
            $("#btnSave").attr('disabled', 'true');
            $("#btnSave").val('Espere por favor...');
            form.submit();
        }
    });
    /*END FORM VALIDATIONS*/

    /* LISTENERS */
    $('[id^="ID_PEDIDO_"]').bind('click', function(){
        if($(this).is(':checked'))
        {
            validacheck++;
            $("#validar").val(validacheck);
        }
        else
        {
            validacheck--;
            $("#validar").val(validacheck);
        }
    });

    /* END LISTENERS */
    $(function() {
        $("#dialog").dialog({
            autoOpen: false,
            height: 350,
            width: 550,
            show: "blind",
            hide: "blind"
        });
    });
    $("#des-pedido-link").click(function(e){
        e.preventDefault();
        var titulo_pedido = $(this).attr("pedido_id");
        var contenido_pedido = "";
        $("#contenido").empty();
        $.ajax({
            url: document_root+'supplier/despacho/get_content',
            async: false,
            type: "post",
            data: {
                pedido_id: $(this).attr("pedido_id")
            }
        });
    });
});
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
    }},  
    dataType: 'json',  
    success: function(data){  
        contenido_pedido = data  
    }  
    });  
  
    $("#dialog").dialog("open");  
    $("#ui-dialog-title-dialog").html('Productos del Pedido: '+titulo_pedido);  
    $("#contenido").append(contenido_pedido);  
  
    });  
  
});
```

1.5.8.2. Procesar.

```
$(document).ready(function(){
  $("#tblPedidos").dataTable({
    "bjQueryUI": true,
    "sPaginationType": "full_numbers",
    "oLanguage": {
      "oPagniate": {
        "sFirst": "Primera",
        "sLast": "&Uacute;ltime",
        "sNext": "Siguiete",
        "sPrevious": "Anterior"
      },
    },
    "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
    "sZeroRecords": "No se encontraron resultados",
    "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
    "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
    "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
    "sSearch": "Criterio de B&uacute;squeda:",
    "sProcessing": "Filtrando.."
  },
  "aoColumns": [
    {"bSortable": true},
    {"bSortable": true},
    {"bSortable": true},
    {"bSortable": true},
    {"bSortable": true},
    {"bSortable": false}
  ],
  "sDom" : '<"H"Tfr>t<"F"pli>',
  "oTableTools": {
    "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
    "aButtons": [{
      "sExtends": "xls",
      "sButtonText": "Exportar a Excel"
    }]
  }
});
});
```

1.5.8.3. Revisar.

```
$(document).ready(function(){
    var validacheck=$("#validar").val();
    $('[id^="btn_"]').bind('click',function(){
        var ID=$(this).attr('id');
        ID = ID.split('_');
        ID = ID[1];
        if(ID=='Comprar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Comprado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Enviar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Enviado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Entregar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Entregado?','Confirmar Accion'))
            {
                return false;
            }
        }
        if(ID=='Cancelar')
        {
            if(!confirm('¿Está seguro de cambiar estado de despacho a Cancelado?\n\nNota: esto tambien cancela los pedidos aprobados anteriormente.','Confirmar Accion'))
            {
                return false;
            }
        }
    });
    /* END LISTENERS */
    $(function() {
        $("#dialog").dialog({
            autoOpen: false,
            height: 350,
            width: 550,
            show: "blind",
            hide: "blind"
        });
    });
    $("#des-pedido-link").click(function(e){
        e.preventDefault();
        var titulo_pedido = $(this).attr("pedido_id");
        var contenido_pedido = "";
        $("#contenido").empty();
        $.ajax({
            url:document_root+supplier/despacho/get_content',
            async: false,
            type: "post",
            data: ({
                pedido_id: $(this).attr("pedido_id")
            }),
            dataType: 'json',
            success: function(data){
                contenido_pedido = data
            }
        });
        $("#dialog").dialog("open");
        $("#ui-dialog-title-dialog").html('Productos del Pedido: '+titulo_pedido);
        $("#contenido").append(contenido_pedido);
    });
});
```

1.5.8.4. *Select.*

```
$(document).ready(function(){
    $('#tblPedidos').dataTable({
        "bjQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPagniate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltime",
                "sNext": "Siguiete",
                "sPrevious": "Anterior"
            },
        },
        "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
        "sZeroRecords": "No se encontraron resultados",
        "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
        "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
        "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
        "sSearch": "Criterio de B&uacute;squeda:",
        "sProcessing": "Filtrando.."
    },
    "aoColumns": [
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": false}
    ],
    "sDom" : '<"H"Tfr>t<"F"pli>',
    "oTableTools": {
        "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
        "aButtons": [{
            "sExtends": "xls",
            "sButtonText": "Exportar a Excel"
        }]
    }
});
});
```

1.6. Perfil Solicitante.

1.6.1. Contenedor.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_User_Containers_User extends Controller_Template {
```

```
    public $template = 'controller/user/containers/user';
```

```
    /**
     * El atributo que va a ser utilizado en cada acción
     * NULL por defecto
     * @var View
     */
    public $view;
```

```
    /**
     * @see system/classes/kohana/controller/Kohana_Controller_Template::before()
     */
    public function before()
```

```
    {
        Session::instance();
        //Required because we are extending Controller_Template
        parent::before();
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario', FALSE))->find();
```

```
        if(!Session::instance()->get('treasure_usuario', FALSE) OR Session::instance()->get('treasure_usuario', FALSE)=="")
        OR $user->ID_PERFIL!=2)
```

```
        {
            HTTP::redirect('login');
        }
```

```
        //$this->_check_permissions();
```

```
        if ( $this->auto_render )
        {
            //Sets the "Default" title set on the container
            $this->template->title = 'Treasure';
            $this->template->content = '';
```

```
            $this->template->styles = array();
            $this->template->scripts = array();
        }
```

```
        //Cleans the view so that every action can set it
        //by itself without problems
        $this->view = NULL;
    }
```

```
    /**
     * @see system/classes/kohana/controller/Kohana_Controller_Template::after()
     */
    public function after()
```

```
    {
        if ( $this->auto_render )
        {
            $action = $this->request->action();
            $controller = $this->request->controller();
            $directory = $this->request->directory();
```

```
            //Build the styles array
            $css_file = 'media/css/controller/'.$directory.'/'.$controller.'/'.$action.'.css';
            $styles = array(
                'media/css/screen.css' => 'screen, projection',
                'media/css/src/print.css' => 'print',
                'media/css/jquery-ui/smoothness/jquery-ui-1.8.16.custom.css' => 'all',
                'media/css/dataTables/TableTools.css' => 'screen',
                'media/css/dataTables/TableTools_JUI.css' => 'screen',
                'media/css/dataTables/demo_table_jui.css' => 'screen',
                'media/css/src/ie.css' => 'screen, projection',
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
'media/css/styles.css' => 'screen',
'media/css/menu.css' => 'screen',
    'media/css/menu-style.css' => 'screen',
'media/css/jquery-ui/jquery.autocomplete.css' => 'all',
);

//VALIDATING FILE
if(file_exists($css_file) )
{
    $styles = array_merge( $styles, array($css_file=>'screen') );
}

//Build the scripts array
$js_file = 'media/js/controller/'.$_directory.'/'.$_controller.'/'.$_action.'.js';
$scripts = array(
    'media/js/jqueryLibrary/jquery-1.6.2.min.js',
    'media/js/jqueryLibrary/jquery-ui-1.8.16.custom.min.js',
    'media/js/jqueryLibrary/jquery-validation.js',
        'media/js/jqueryLibrary/jquery.autocomplete.js',
        'media/js/dataTables/jquery.dataTables.js',
    'media/js/dataTables/TableTools.js',
    'media/js/dataTables/ZeroClipboard.js',
        'media/js/dataTables/dataTables.editor.js',
    'media/js/globalfunctions.js',
    'media/js/ajaxupload.js',
    'media/js/system.js',
    'media/js/mbMenu.js',
);

//VALIDATING FILE
if(file_exists($js_file) )
{
    $scripts = array_merge( $scripts, array($js_file) );
}

//Set styles and scripts on the template (container). This can also be
//done from any action method
$this->template->styles = array_merge( $styles, $this->template->styles );
$this->template->scripts = array_merge( $scripts, $this->template->scripts );

//Get Menu
//Uncomment once we have all the menu logic created
$menu=FALSE;//$this->_get_menu( Model_Globalfunctions::get_menu( Session::instance()-
>get('discotecas_serial_prf', FALSE) ) );
if($menu){
    $this->template->menu = $menu;
}else{
    $this->template->menu = false;
}

//If $this->view is set display it on the
//containers $content
$this->template->content = $this->view;
}

//Required because we are extending Controller_Template
parent::after();
}
}
```

1.6.2. Contenedor de Vista.

```
<!doctype html>
<html lang="<?php echo l18n::$lang ?>">
  <head>
    <meta charset="utf-8">
    <link rel="shortcut icon" type="image/x-icon" href="/treasure/media/img/icon.ico">
    <title><?php echo $title ?></title>
    <?php foreach ($styles as $file => $type) echo HTML::style($file, array('media' => $type)), PHP_EOL ?>
    <?php foreach ($scripts as $file) echo HTML::script($file), PHP_EOL ?>
    <script type="text/javascript">
      var document_root = '<?php echo Url::site();?>';
    </script>
  </head>
  <body>
    <div class="container" id="admin_container">
      <div class="span-24 general_bg">
        <div class="span-24 last " style=" padding-top: 5px;">
          <div class="prepend-23 span-1 last admin_banner">
            <strong><a href="<?php echo Url::site();?>" style="text-decoration:none; color: black; font-weight:
normal;">Salir</a></strong>
          </div>
        </div>

        <div class="span-24 ">
          <?php echo Helper_User_Menu::menu(); ?>
        </div>
      </div>
      <div class="span-24 content">
        <div class="prepend-2 span-20 append-2 line last" id="container-flash-messages" style=" padding-top:
15px;">
          <?php FlashMessenger::factory()->get_messages(); ?>
        </div>
        <?php echo $content ?>
      </div>
      <div id="spinner" style="display: none;">
        <div id="ajax_loader"></div>
        <div class="ajax_loader_text">Cargando...</div>
      </div>
    </div>
  </body>
</html>
```

1.6.3. Controlador de Bodega.

```
<?php
class Controller_User_Bodega extends Controller_User_Containers_User {

    public function action_index()
    {
        HTTP::redirect('user/bodega/select/');
    }

    public function action_editcontent()
    {
        if($this->request->is_ajax())
        {
            $pxb_id = arr::get($this->request->post(), 'pxb_id');
            $cantidad = arr::get($this->request->post(), 'cantidad');
            try
            {
                DB::update('producto_x_bodega')->set(array('CANTIDAD'=>$cantidad))->where('ID_PXB', '=', $pxb_id)
                    ->execute();
                $content = (int)$cantidad;
                echo json_encode($content);
            }
            catch(Exception $e)
            {
                echo json_encode($e->getMessage());
            }
            $this->auto_render = FALSE;
        }
        else
        {
            HTTP::redirect('user/main');
        }
    }

    public function action_select()
    {
        $this->view = View::factory('controller/user/bodega/select');
        $departamento = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))-
>find();
        $productos = ORM::factory('producto')->where('ID_DEPARTAMENTO', '=', $departamento->ID_DEPARTAMENTO)-
>find_all();
        $producto = array();
        foreach ($productos as $value) {
            $producto = $producto + array($value->ID_PRODUCTO => $value->ID_PRODUCTO);
        }
        $bodega = ORM::factory('productoxbodega')->where('ID_PRODUCTO', 'IN', $producto)->find_all();
        $producto = array();
        foreach ($productos as $value) {
            $producto = $producto + array($value->ID_PRODUCTO=>$value->DESCRIPCION);
        }

        if(! count($bodega))
            $bodega=NULL;

        if( $this->request->post() )
        {
            try
            {
                $values= $this->request->post();

                $producto_resta = array();

                foreach ($values as $key => $value) {
                    if(strpos($key, 'cantidad_rest_')!= false)
                    {
                        $aux = explode('_', $key);
                        if($value!=0)
                            $producto_resta = $producto_resta + array($aux[2] => $value);
                    }
                }
            }
        }
    }
}
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
    }  
  }  
  
  if(count($producto_resta))  
  {  
    foreach ($producto_resta as $key => $value) {  
      $aux = ORM::factory('productoxbodega')->where('ID_PXB', '=', $key)->find();  
      $aux = $aux->CANTIDAD - $value;  
      $actualizar_cantidad = DB::update('producto_x_bodega')->set (array('CANTIDAD'=>$aux))->where  
( 'ID_PXB', '=', $key)->execute ();  
    }  
  }  
  if(count($producto_resta))  
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',  
'admin:producto:create:cantidad'));  
  else  
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',  
'admin:producto:create:nada'));  
  
  }  
  catch(Exception $e)  
  {  
    echo $e->getMessage();  
    FlashMessenger::factory()->set_message('error', $e->getMessage());  
  }  
  HTTP::redirect('user/main/');  
}  
  
$this->view->set('bodega', $bodega);  
$this->view->set('producto', $producto);  
}  
?>
```

1.6.4. Controlador de Pedido.

```

<?php
class Controller_User_Pedido extends Controller_User_Containers_User {

    public function action_index()
    {
        HTTP::redirect('user/pedido/create/');
    }

    public function action_create()
    {
        //IF USER SUBMITTED FORM
        $this->view = View::factory('controller/user/pedido/create' );

        //IF USER SUBMITTED FORM
        $ID_PEDIDO = $this->request->param('params');
        $pedido = ORM::factory('pedido')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find();
        $departamento = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))->find();
        $producto = ORM::factory('producto')->where('ID_DEPARTAMENTO', '=', $departamento->ID_DEPARTAMENTO)-
        >order_by('DESCRIPCION')->find_all()->as_array('ID_PRODUCTO', 'DESCRIPCION');
        $this->view->set( 'producto', $producto );
        $user = ORM::factory('user')->where('ID_USUARIO', '=', $pedido->ID_USUARIO)->find();
        $this->view->set( 'user', $user->ID_USUARIO );
        if($ID_PEDIDO)
        {
            $PXP = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)->find_all()-
            >as_array('ID_PRODUCTO', 'CANTIDAD' );
            $productos = count($PXP);
        }
        else
        {
            $PXP = "";
            $productos = 0;
        }
        $this->view->set( 'PXP', $PXP );
        $this->view->set( 'productos', $productos );

        //VALIDATE POST
        if( $this->request->post() )
        {
            try
            {
                $values= $this->request->post();
                if($ID_PEDIDO)
                {
                    $Eliminapxps = ORM::factory('pedidodespachoxproducto')->where('ID_PEDIDO', '=', $ID_PEDIDO)
                    ->find_all();
                    foreach ($Eliminapxps as $Eliminapxp) {
                        DB::delete('pedido_despacho_x_producto')->where('ID_PDXP', '=', $Eliminapxp->ID_PDXP)
                        ->execute(Database::instance());
                    }
                }

                $productosIngreso = array();
                foreach ($values as $key => $value) {
                    if(strpos($key, 'CANTIDAD_')!= false)
                    {
                        $aux = explode('_', $key);
                        $productosIngreso = $productosIngreso + array($aux[1]==>$value);
                    }
                }
                if(!$ID_PEDIDO)
                {
                    $tiempo = array('d'=>(int)date('d'),'m'=>(int)date('m'),'Y'=>(int)date('Y'),'H'=>(int)date('H'),'i'=>(int)date('i'));
                    $pedido->ID_USUARIO = Session::instance()->get('treasure_usuario');
                }
            }
        }
    }
}

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
$pedido->DIA = $tiempo['d'];
$pedido->MES = $tiempo['m'];
$pedido->ANIO = $tiempo['Y'];
$pedido->HORA = $tiempo['H'];
$pedido->MINUTO = $tiempo['i'];
$pedido->save();
```

```
$pedido2 = ORM::factory('pedido')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))
->and_where('DIA', '=', $tiempo['d'])
->and_where('MES', '=', $tiempo['m'])
->and_where('ANIO', '=', $tiempo['Y'])
->and_where('HORA', '=', $tiempo['H'])
->and_where('MINUTO', '=', $tiempo['i'])
->find();

foreach ($productosIngreso as $key=>$value) {
    $PXP2 = ORM::factory('pedidospachoproducto');
    $PXP2->ID_PEDIDO = $pedido2->ID_PEDIDO;
    $PXP2->ID_PRODUCTO = $key;
    $PXP2->CANTIDAD = $value;
    $PXP2->ESTADO = 'PEDIDO';
    $PXP2->save();
}
}
else
{
    $pedido2 = ORM::factory('pedido')->where('ID_USUARIO', '=', $pedido->ID_USUARIO)
->and_where('DIA', '=', $pedido->DIA)
->and_where('MES', '=', $pedido->MES)
->and_where('ANIO', '=', $pedido->ANIO)
->and_where('HORA', '=', $pedido->HORA)
->and_where('MINUTO', '=', $pedido->MINUTO)
->find();

    foreach ($productosIngreso as $key=>$value) {
        $PXP2 = ORM::factory('pedidospachoproducto');
        $PXP2->ID_PEDIDO = $pedido2->ID_PEDIDO;
        $PXP2->ID_PRODUCTO = $key;
        $PXP2->CANTIDAD = $value;
        $PXP2->ESTADO = 'PEDIDO';
        $PXP2->save();
    }
}
}
```

```
if($ID_PEDIDO)
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:pedido:create:update'));
else
    FlashMessenger::factory()->set_message('success', Kohana::message('admin',
'admin:pedido:create:success'));
}
catch(Exception $e)
{
    echo $e->getMessage();
    FlashMessenger::factory()->set_message('error', $e->getMessage());
}
HTTP::redirect('user/pedido/select');
}
$this->view->set('pedido', $pedido);
}
```

```
public function action_select()
{
    $this->view = View::factory('controller/user/pedido/select');
    $pedidos_usuario = ORM::factory('pedido')->where('ID_USUARIO', '=', Session::instance()-
>get('treasure_usuario'))->find_all();
    if($pedidos_usuario->count())
    {
        $pedido_usuario = array();
        foreach ($pedidos_usuario as $value) {
            $pedido_usuario = $pedido_usuario + array($value->ID_PEDIDO => $value->ID_PEDIDO);
        }
        $pedido_usuario = implode(',', $pedido_usuario);
    }
}
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
$pedidosp = DB::query(Database::SELECT, "SELECT Distinct('ID_PEDIDO') FROM
`pedido_despacho_x_producto` WHERE `ID_PEDIDO` IN (". $pedido_usuario. ") AND `ESTADO`='PEDIDO' AND
`ID_DESPACHO` is NULL")->execute()->as_array();
$pedido = array();
foreach ($pedidosp as $value) {
    $pedidos = ORM::factory('pedido')->where('ID_PEDIDO', '=', $value)->find();
    $pedido = $pedido + array($pedidos->ID_PEDIDO=>$pedidos->as_array());
}
}
else
{
    $pedido = NULL;
}

$users = ORM::factory('user')->distinct('ID_USUARIO')->find_all()->as_array();
$user = array();
foreach ($users as $value) {
    $nombres = $value->NOMBRE.'<br>'.$value->APELLIDO;
    $user = $user + array($value->ID_USUARIO=>$nombres);
}
if(! count($pedido))
    $pedido=NULL;
$this->view->set('user', $user);
$this->view->set('pedido', $pedido);
}
?>
```

1.6.5. Controlador de Main.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Controller_User_Main extends Controller_User_Containers_User {
```

```
    public function action_index()
    {
        $option = $this->request->param('params');
        $user = ORM::factory('user')->where('ID_USUARIO', '=', Session::instance()->get('treasure_usuario'))->find();
        $name = ".strtoupper ($user->NOMBRE)";
        $this->view = View::factory( 'controller/user/main' );
        $this->view->set('name',$name);
    }
}
```

1.6.6. Helper

1.6.6.1. Menú.

```
<?php defined('SYSPATH') or die('No direct script access.');
```

```
class Helper_User_Menu {  
    public static function menu($seg1 = null, $seg2 = null, $seg3 = null)  
    {  
        $items = array();  
  
        /*PEDIDOS*/  
        $items[] = array('id' => 1, 'parent_id' => 0, 'title' => 'Pedidos', 'link' => '#');  
        $items[] = array('id' => 2, 'parent_id' => 1, 'title' => 'Nuevo', 'link' => 'user/pedido/create');  
        $items[] = array('id' => 3, 'parent_id' => 1, 'title' => 'Modificar', 'link' => 'user/pedido/select');  
  
        /*BODEGA*/  
        $items[] = array('id' => 4, 'parent_id' => 0, 'title' => 'Bodega', 'link' => '#');  
        $items[] = array('id' => 5, 'parent_id' => 4, 'title' => 'Manejo', 'link' => 'user/bodega/select');  
  
        // Build the current path  
        $current = "";  
        $current .= $seg1 ? $seg1 : "";  
        $current .= $seg2 ? ("/" . $seg2): "";  
        $current .= $seg3 ? ("/" . $seg3): "";  
  
        // Let's get the body of the current page  
        //$body = ORM::factory('entry')->where('link','=', $current)->find()->body;  
        $options = array('link_prepend' => "", 'current_path' => $current);  
  
        return MenuStructure::factory($items, $options)->get_menu();  
    }  
}
```

```
?>
```

1.6.7. Vistas de Bodega.

1.6.7.1. Select.

```
<div class="span-24">
  <br><br>
  <div class="title">Bodega</div>
  <?php if($bodega): ?>
  <div class="span-24"><br>
    <form action="" method="post" name="frmBodega" id="frmBodega">
      <table id="tblBodega" >
        <thead>
          <tr>
            <th class="span-3">PRODUCTO</th>
            <th class="span-3">CANTIDAD</th>
            <th class="span-3">CANTIDAD PARA EGRESO</th>
          </tr>
        </thead>
        <tbody>
          <?php foreach($bodega as $bod): ?>
          <tr>
            <td><?php echo $producto[$bod->ID_PRODUCTO] ?></td>
            <td>
              <?php echo $bod->CANTIDAD; ?>
              <input type="hidden" name="cantidad_resta_<?php echo $bod->ID_PXB; ?>"
id="cantidad_resta_<?php echo $bod->ID_PXB; ?>" value="0"/>
              <input type="hidden" name="cantidad_original_<?php echo $bod->ID_PXB; ?>"
id="cantidad_original_<?php echo $bod->ID_PXB; ?>" value="<?php echo $bod->CANTIDAD; ?>"/>
            </td>
            <td id="<?php echo $bod->ID_PXB ?>">
              <input type="text" value="0" class="span-1 last" id="cantidad_input_<?php echo $bod->ID_PXB;
?>" readonly="readonly"/>
              <a class="span-1 last" id="boton_mas_<?php echo $bod->ID_PXB; ?>">
                </img>
              </a>&nbsp;
              <a class="span-1 last" id="boton_menos_<?php echo $bod->ID_PXB; ?>">
                </img>
              </a>
            </td>
          </tr>
          <?php endforeach; ?>
        </tbody>
      </table>
      <div class="span-24 " align="center"><br>
      <input type="submit" id="btnSave" name="btnSave" value="Guardar" />
      </div>
    </form>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
  <div class="alerts" style="text-align: center">
    No existen Productos registrados en bodega.
  </div>
  </div>
  <?php endif; ?>
</div>
```

1.6.8. Vistas de Pedido.

1.6.8.1. Create.

```

<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="title"><?php echo ($pedido->ID_PEDIDO)?'Actualizar':'Nuevo'; ?> Pedido</div>
    <div class="span-20 required_fields line" align="center">
      <label>(*) Campos Obligatorios</label>
    </div>

    <div class="prepend-5 span-10" align="center">
      <div id="dialog_errores">
        <ul id="alerts" class="alerts"></ul>
      </div>
    </div>

    <br/><br/><br/>

    <form action="" method="post" name="frmPedido" id="frmPedido">
      <?php if($pedido->ID_PEDIDO): ?>
        <?php foreach ($producto as $key => $value): ?>

          <div class="span-20 line ">
            <div class="span-10">
              <?php if(array_key_exists($key,$PXP)): ?>
                <?php echo Form::checkbox(".$key.", $key, FALSE,
array('id'=>'ID_PRODUCTO_'. $key.', 'class' => 'span-5', 'checked' => 'checked' )); ?>

                <?php else: ?>
                <?php echo Form::checkbox(".$key.", $key, FALSE,
array('id'=>'ID_PRODUCTO_'. $key.', 'class' => 'span-5')); ?>

                <?php endif; ?>
              <div class="span-5 label"> <?php echo $value?> </div>
            </div>
            <div class="span-10 last">
              <div class="prepend-2 span-3 label"> *Cantidad: </div>
              <?php if(array_key_exists($key,$PXP)): ?>
                <input type="text" class="span-1 last" name="CANTIDAD_<?php echo $key?>"
id='CANTIDAD_<?php echo $key?>' value="<?php echo $PXP[$key]?>" title="El campo "Cantidad" es obligatorio.'
maxlength="2"/>

                <?php else: ?>
                <input type="text" class="span-1 last" name="CANTIDAD_<?php echo $key?>"
id='CANTIDAD_<?php echo $key?>' value="" title="El campo "Cantidad" es obligatorio.' disabled="disable"
maxlength="2"/>

                <?php endif; ?>
              <input type="hidden" name="<?php echo $key?>" id="<?php echo $key?>"
value="<?php echo $key?>"/>
            </div>
          </div>

          <?php endforeach; ?>

          <?php else: ?>
            <?php foreach ($producto as $key => $value): ?>
              <div class="span-20 line ">
                <div class="span-10">
                  <?php echo Form::checkbox(".$key.", $key, FALSE, array('id'=>'ID_PRODUCTO_'. $key.', 'class'
=> 'span-5')); ?>

                  <div class="span-5 label"> <?php echo $value?> </div>
                </div>
                <div class="span-10 last">
                  <div class="prepend-2 span-3 label"> *Cantidad: </div>
                  <input type="text" class="span-1 last" name="CANTIDAD_<?php echo $key?>"
id='CANTIDAD_<?php echo $key?>' value="" title="El campo "Cantidad" es obligatorio.' disabled="disable"
maxlength="2"/>

                  <input type="hidden" name="<?php echo $key?>" id="<?php echo $key?>" value="<?php echo
$key?>"/>
                </div>
              </div>
            </div>
          </div>

```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
</div>

<?php endforeach; ?>

<div id="lineas" name="lineas" clas="span-20 line"><br><br>

</div>

<?php endif; ?>

<input type="hidden" name="id" id="id" value="<?php echo ($pedido->ID_PEDIDO)?$pedido-
>ID_PEDIDO:":"?>"/>
<input type="hidden" name="validar" id="validar" value="<?php echo ($productos)?$productos:0;?>" />
<input type="hidden" name="user" id="user" value="<?php echo ($user)?$user:":"?>" />
<input type="hidden" name="productos" id="productos" value="" />
<div class="span-20 " align="center"><br>
<input type="submit" id="btnSave" name="btnSave" value="Guardar" />
</div>
</form>
</div>
</div>
```

1.6.8.2. *Select.*

```
<div class="span-24">
  <br><br>
  <div class="title">Seleccionar Pedido Para Modificar</div>
  <?php if($pedido): ?>
  <div class="span-24"><br>
    <a href="<?php echo URL::site('user/pedido/create');?>">Crear Pedido</a><br><br>
    <table id="tblPedidos">
      <thead>
        <tr>
          <th class="span-3">Nº de Pedido</th>
          <th class="span-3">USUARIO</th>
          <th class="span-3">ANIO</th>
          <th class="span-3">MES</th>
          <th class="span-3">DIA</th>
          <th class="span-1">EDITAR</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach($pedido as $ped): ?>
          <tr>
            <td><?php echo $ped["ID_PEDIDO"] ?></td>
            <td><?php echo $user[$ped["ID_USUARIO"]] ?></td>
            <td><?php echo $ped["ANIO"] ?></td>
            <td><?php echo $ped["MES"] ?></td>
            <td><?php echo $ped["DIA"] ?></td>
            <td>
              <a href="<?php echo URL::site('user/pedido/create/'.$ped["ID_PEDIDO"]); ?>">
                </img>
              </a>
            </td>
          </tr>
        </tr>
      </tbody>
    </table>
  </div>
  <?php else: ?>
  <div class="prepend-7 span-10 append-7 line last"><br>
    <div class="alerts" style="text-align: center">
      No existen Pedidos registrados.
    </div>
  </div>
  <?php endif: ?>
</div>
```

1.6.9. *Vistas de Main.*

```
<div class="span-24">
  <div class="prepend-2 span-20">
    <div class="prepend-5 span-20 title_main_page center">
      PAGINA DE USUARIO<br><br>
      BIENVENIDO <?php echo $name?>
    </div>
  </div>
</div>
```

1.6.10. Archivos JS de Bodega.

1.6.10.1. Select.

```
$(document).ready(function(){
  $('#tblBodega').dataTable({
    "bJQueryUI": true,
    "sPaginationType": "full_numbers",
    "oLanguage": {
      "oPaginate": {
        "sFirst": "Primera",
        "sLast": "&Uacute;ltimea",
        "sNext": "Siguiente",
        "sPrevious": "Anterior"
      },
      "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
      "sZeroRecords": "No se encontraron resultados",
      "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
      "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
      "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
      "sSearch": "Criterio de B&uacute;squeda:",
      "sProcessing": "Filtrando.."
    },
    "aoColumns": [
      { "bSortable": true },
      { "bSortable": true },
      { "bSortable": false }
    ],
    "sDom": '<"H"Tfr<t<"F"pli>',
    "oTableTools": {
      "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
      "aButtons": [
        {
          "sExtends": "xls",
          "sButtonText": "Exportar a Excel"
        }
      ]
    }
  });
  $('#[id^="boton_mas_"]').bind('click',function(){
    var ID = $(this).attr('id');
    ID = ID.split('_');
    ID = ID[2];
    var aux = parseInt($("#cantidad_resta_"+ID).val(), 10);
    aux = aux + 1;
    if(aux>$("#cantidad_original_"+ID).val())
      alert('La cantidad sobrepasa la cantidad en stock');
    else
    {
      $("#cantidad_resta_"+ID).val(aux);
      $("#cantidad_input_"+ID).val(aux);
    }
  });
  $('#[id^="boton_menos_"]').bind('click',function(){
    var ID = $(this).attr('id');
    ID = ID.split('_');
    ID = ID[2];
    var aux = parseInt($("#cantidad_resta_"+ID).val(), 10);
    aux = aux - 1;
    if(aux<0)
      alert('La cantidad no puede ser menor a 0');
    else
    {
      $("#cantidad_resta_"+ID).val(aux);
      $("#cantidad_input_"+ID).val(aux);
    }
  });
  $('#btnSave').bind('click',function(){
    if(!confirm('¿Está seguro de hacer la actualización de bodega?','Confirmar Accion'))
    {
      return false;
    }
  });
});
```

1.6.11. Archivos JS de Pedido.

1.6.11.1. Create.

```
$(document).ready(function(){
  var validacheck=$("#validar").val();

  /*FORM VALIDATIONS*/
  $("#frmPedido").validate({
    errorLabelContainer: "#alerts",
    wrapper: "i",
    onfocusout: false,
    onkeyup: false,
    async: false,
    rules: {
      validar: {
        required: true,
        min: true
      }
    },
    messages: {
      validar: {
        required: 'Debe elegir por lo menos un producto',
        min: 'Debe elegir por lo menos un producto'
      }
    },
    submitHandler: function(form) {
      $("#btnSave).attr('disabled', 'true');
      $("#btnSave).val('Espere por favor...');
      form.submit();
    }
  });
  /*END FORM VALIDATIONS*/

  /* LISTENERS */
  $('[id^="ID_PRODUCTO_"]').bind('click',function(){
    if($(this).is(':checked'))
    {
      $("#CANTIDAD_"+$(this).val()+").attr('disabled',false);
      validacheck++;
      $("#validar").val(validacheck);
      $("#CANTIDAD_"+$(this).val()+").rules('add',{
        required:true,
        maxlength: 2,
        digits: true,
        messages: {
          required:'El campo "Cantidad" es obligatorio.',
          maxlength:'El campo "Cantidad" debe contener m&acute;ximo 2 digitos.',
          digits:'El campo "Cantidad" debe contener solo digitos.'
        }
      });
    }
    else
    {
      $("#CANTIDAD_"+$(this).val()+").attr('disabled',true);
      $("#CANTIDAD_"+$(this).val()+").val('');
      validacheck--;
      $("#validar").val(validacheck);
    }
  });

  /* END LISTENERS */

});

//ADD PASSWORD RULES
function addAniadirRules(aumenta)
{
  $("#CANTIDAD_"+aumenta).rules('add',{
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
required: true,  
maxlength: 2,  
digits: true,  
messages: {  
  required: 'El campo "Cantidad" es obligatorio.',  
  maxlength: 'El campo "Cantidad" debe contener máximo 2 dígitos.',  
  digits: 'El campo "Contraseña" debe contener solo dígitos.'  
}  
});  
}
```

1.6.11.2. Select.

```
$(document).ready(function(){
    $('#tblPedidos').dataTable({
        "bjQueryUI": true,
        "sPaginationType": "full_numbers",
        "oLanguage": {
            "oPagninate": {
                "sFirst": "Primera",
                "sLast": "&Uacute;ltime",
                "sNext": "Siguiete",
                "sPrevious": "Anterior"
            },
        },
        "sLengthMenu": "Mostrar _MENU_ registros por P&aacute;gina",
        "sZeroRecords": "No se encontraron resultados",
        "sInfo": "Mostrando _START_ a _END_ de _TOTAL_ registros",
        "sInfoEmpty": "Mostrando 0 a 0 de 0 registros",
        "sInfoFiltered": "(filtrado de _MAX_ registros en total)",
        "sSearch": "Criterio de B&uacute;squeda:",
        "sProcessing": "Filtrando.."
    },
    "aoColumns": [
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": true},
        {"bSortable": false}
    ],
    "sDom" : '<"H"Tfr>t<"F"pli>',
    "oTableTools": {
        "sSwfPath": document_root+"media/swf/copy_cvs_xls_pdf.swf",
        "aButtons": [{
            "sExtends": "xls",
            "sButtonText": "Exportar a Excel"
        }]
    }
});
});
```

1.7. Modelos.

1.7.1. Departamento.

```
<?php defined('SYSPATH') or die('No direct access allowed.');
```

```
class Model_Departamento extends ORM {
```

```
    protected $_table_name = 'departamento';
```

```
}
```

```
?>
```

1.7.2. Despacho.

```
<?php defined('SYSPATH') or die('No direct access allowed.');
```

```
class Model_Despacho extends ORM {
```

```
    protected $_table_name = 'despacho';
```

```
}
```

```
?>
```

1.7.3. Pedido.

```
<?php defined('SYSPATH') or die('No direct access allowed.');
```

```
class Model_Pedido extends ORM {
```

```
    protected $_table_name = 'pedido';
```

```
}
```

```
?>
```

1.7.4. Pedido Despacho por Producto.

```
<?php defined('SYSPATH') or die('No direct access allowed.');
```

```
class Model_Pedidodespachoxproducto extends ORM {
```

```
    public $status_values_pedido = array('PEDIDO' => 'Pedido', 'APROBADO' => 'Aprobado', 'CANCELADO' => 'Cancelado');
```

```
    public $status_values_despacho = array('CANCELADO' => 'Cancelado', 'EN PROCESO' => 'En Proceso', 'COMPRADO' => 'Comprado', 'ENVIADO' => 'Enviado', 'ENTREGADO' => 'Entregado');
```

```
    public $status_values_completo = array('PEDIDO' => 'Pedido', 'APROBADO' => 'Aprobado', 'CANCELADO' => 'Cancelado', 'CANCELADO' => 'Cancelado', 'EN PROCESO' => 'En Proceso', 'COMPRADO' => 'Comprado', 'ENVIADO' => 'Enviado', 'ENTREGADO' => 'Entregado');
```

```
    protected $_table_name = 'pedido_despacho_x_producto';
```

```
}
```

```
?>
```

1.7.5. Perfil.

```
<?php defined('SYSPATH') or die('No direct access allowed.');
```

```
class Model_Perfil extends ORM {
```

```
    public static $status_values = array('Activo' => 'Activo', 'Inactivo' => 'Inactivo');
```

```
    protected $_table_name = 'perfil';
```

```
    public static function verifica_perfil($id_perfil)
```

```
    {
```

```
        $perfil = ORM::factory('perfil')->where('!ID_PERFIL', '=', $id_perfil)->find();
```

```
        if (strtolower(trim($perfil->NOMBRE))=='administrador')
```

```
        {
```

```
            $ret = 1;
```

```
            return $ret;
```

```
        }
```

```
        if (strtolower(trim($perfil->NOMBRE))=='solicitante')
```

```
        {
```

```
            $ret = 2;
```

```
            return $ret;
```

```
        }
```

```
        if (strtolower(trim($perfil->NOMBRE))=='aprovador')
```

```
        {
```

```
            $ret = 3;
```

```
            return $ret;
```

```
        }
```

```
        if (strtolower(trim($perfil->NOMBRE))=='controlador')
```

```
        {
```

```
            $ret = 4;
```

```
            return $ret;
```

```
        }
```

```
        if (strtolower(trim($perfil->NOMBRE))=='proveedor')
```

```
        {
```

```
            $ret = 5;
```

```
            return $ret;
```

```
        }
```

```
        else
```

```
            return 0;
```

```
    }
```

```
}
```

```
?>
```

1.7.6. Producto.

```
<?php defined('SYSPATH') or die('No direct access allowed.');
```

```
class Model_Producto extends ORM {
```

```
    protected $_table_name = 'producto';
```

```
}
```

```
?>
```

1.7.7. Producto por Bodega.

```
<?php defined('SYSPATH') or die('No direct access allowed.');
```

```
class Model_productoxbodega extends ORM {
```

```
    protected $_table_name = 'producto_x_bodega';
```

```
}
```

```
?>
```

1.7.8. Solicitante.

```
<?php defined('SYSPATH') or die('No direct access allowed.');
```

```
class Model_User extends ORM {
```

```
    public static $status_values = array('Activo' => 'Activo', 'Inactivo' => 'Inactivo');
```

```
    protected $_table_name = 'usuario';
```

```
    public static function authenticate($username, $password)
```

```
    {
```

```
        $user = ORM::factory('user')->where('USUARIO', '=', $username)->find();
```

```
        if (($user->loaded())!=false)
```

```
        {
```

```
            return false;
```

```
        }
```

```
        if ($user->ESTADO != 'Activo')
```

```
        {
```

```
            return false;
```

```
        }
```

```
        if ($password != $user->CONTRASENIA)
```

```
        {
```

```
            return false;
```

```
        }
```

```
        return $user;
```

```
    }
```

```
}
```

```
?>
```

1.8. Mensajes Flash.

```
<?php defined('SYSPATH') OR die('No direct access allowed.');
```

```
class FlashMessenger {  
    protected static $_instance;  
    // Tipos of message  
    private $_statuses = array(  
        'success' => 'success',  
        'error' => 'error',  
        'notice' => 'notice'  
    );  
    public static function factory()  
    {  
        if (!isset(FlashMessenger::$_instance))  
        {  
            // Create a new session instance  
            FlashMessenger::$_instance = new FlashMessenger;  
        }  
        return FlashMessenger::$_instance;  
    }  
  
    public function set_message($status, $message)  
    {  
        if(in_array($status, $this->_statuses))  
        {  
            $_SESSION['flash'][$status][] = $message;  
        }  
    }  
    private function _render($class)  
    {  
        $html = '';  
        if (isset($_SESSION['flash']) && is_array($_SESSION['flash']))  
        {  
            $html .= '<div id="flash-messages" class="'. $class .' last">';  
            foreach ($_SESSION['flash'] as $k => $v)  
            {  
                $html .= "<div class='{ $k }'>";  
                $html .= "<span class='close'>x</span>";  
                $html .= '<ul>';  
                foreach($v as $message)  
                {  
                    $html .= '<li>'. $message . '</li>';  
                }  
                $html .= '<ul>';  
                $html .= '</div>';  
            }  
            $html .= '</div>';  
        }  
        echo $html;  
        return $this;  
    }  
    private function _add_css()  
    {  
        echo HTML::style('media/css/flash-messenger.css');  
        return $this;  
    }  
  
    private function _add_js()  
    {  
        echo HTML::script('media/js/flash-messenger.js');  
        return $this;  
    }  
  
    private function _clear()  
    {  
        $_SESSION['flash'] = array();  
    }  
  
    public function get_messages($class = 'prepend-5 span-10 append-5')  
    {  
        $this->_render($class)->_add_css()->_add_js()->_clear();  
    }  
}
```

1.9. Archivo SQL de Instalación.

```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 25-03-2013 a las 05:05:26
-- Versión del servidor: 5.5.27
-- Versión de PHP: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de datos: `treasure`
--

-----

--
-- Estructura de tabla para la tabla `bodega`
--

DROP TABLE IF EXISTS `bodega`;
CREATE TABLE IF NOT EXISTS `bodega` (
  `ID_BODEGA` int(11) NOT NULL AUTO_INCREMENT,
  `UBICACION` varchar(20) COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`ID_BODEGA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=2;

-----

--
-- Estructura de tabla para la tabla `departamento`
--

DROP TABLE IF EXISTS `departamento`;
CREATE TABLE IF NOT EXISTS `departamento` (
  `ID_DEPARTAMENTO` int(11) NOT NULL AUTO_INCREMENT,
  `NOMBRE` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
  `DESCRIPCION` text COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`ID_DEPARTAMENTO`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=8;

-----

--
-- Estructura de tabla para la tabla `despacho`
--

DROP TABLE IF EXISTS `despacho`;
CREATE TABLE IF NOT EXISTS `despacho` (
  `ID_DESPACHO` int(11) NOT NULL AUTO_INCREMENT,
  `ID_USUARIO` int(11) NOT NULL,
  `DIA` int(11) NOT NULL,
  `MES` int(11) NOT NULL,
  `ANIO` int(11) NOT NULL,
  `HORA` int(11) NOT NULL,
  `MINUTO` int(11) NOT NULL,
  PRIMARY KEY (`ID_DESPACHO`),
  KEY `ID_USUARIO` (`ID_USUARIO`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=12;

-----
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
--
-- Estructura de tabla para la tabla `pedido`
--
DROP TABLE IF EXISTS `pedido`;
CREATE TABLE IF NOT EXISTS `pedido` (
  `ID_PEDIDO` int(11) NOT NULL AUTO_INCREMENT,
  `ID_USUARIO` int(11) NOT NULL,
  `DIA` int(11) NOT NULL,
  `MES` int(11) NOT NULL,
  `ANIO` int(11) NOT NULL,
  `HORA` int(11) NOT NULL,
  `MINUTO` int(11) NOT NULL,
  PRIMARY KEY (`ID_PEDIDO`),
  KEY `FK_REALIZA` (`ID_USUARIO`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=27 ;

-----

--
-- Estructura de tabla para la tabla `pedido_despacho_x_producto`
--
DROP TABLE IF EXISTS `pedido_despacho_x_producto`;
CREATE TABLE IF NOT EXISTS `pedido_despacho_x_producto` (
  `ID_PDXP` int(11) NOT NULL AUTO_INCREMENT,
  `ID_PEDIDO` int(11) NOT NULL,
  `ID_PRODUCTO` int(11) NOT NULL,
  `ID_DESPACHO` int(11) DEFAULT NULL,
  `CANTIDAD` int(11) NOT NULL,
  `ESTADO` enum('PEDIDO','CANCELADO','APROBADO','EN PROCESO','COMPRADO','ENVIADO','ENTREGADO')
  COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`ID_PDXP`),
  KEY `FK_GENERA` (`ID_PEDIDO`),
  KEY `FK_PROBOCA` (`ID_DESPACHO`),
  KEY `FK_REFERENCIA` (`ID_PRODUCTO`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=41 ;

-----

--
-- Estructura de tabla para la tabla `perfil`
--
DROP TABLE IF EXISTS `perfil`;
CREATE TABLE IF NOT EXISTS `perfil` (
  `ID_PERFIL` int(11) NOT NULL AUTO_INCREMENT,
  `NOMBRE` varchar(20) COLLATE latin1_spanish_ci NOT NULL,
  `DESCRIPCION` text COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`ID_PERFIL`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=6 ;

-----

--
-- Estructura de tabla para la tabla `producto`
--
DROP TABLE IF EXISTS `producto`;
CREATE TABLE IF NOT EXISTS `producto` (
  `ID_PRODUCTO` int(11) NOT NULL AUTO_INCREMENT,
  `ID_DEPARTAMENTO` int(11) NOT NULL,
  `DESCRIPCION` text COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`ID_PRODUCTO`),
  KEY `FK_TIENE` (`ID_DEPARTAMENTO`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=7 ;

-----

--
-- Estructura de tabla para la tabla `producto_x_bodega`
--
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
DROP TABLE IF EXISTS `producto_x_bodega`;
CREATE TABLE IF NOT EXISTS `producto_x_bodega` (
  `ID_PXB` int(11) NOT NULL AUTO_INCREMENT,
  `ID_PRODUCTO` int(11) NOT NULL,
  `ID_BODEGA` int(11) NOT NULL,
  `CANTIDAD` int(11) NOT NULL,
  PRIMARY KEY (`ID_PXB`),
  UNIQUE KEY `ID_PRODUCTO` (`ID_PRODUCTO`),
  KEY `FK_ALMACENA` (`ID_BODEGA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=9;

-----

--
-- Estructura de tabla para la tabla `usuario`
--

DROP TABLE IF EXISTS `usuario`;
CREATE TABLE IF NOT EXISTS `usuario` (
  `ID_USUARIO` int(11) NOT NULL AUTO_INCREMENT,
  `ID_PERFIL` int(11) NOT NULL,
  `ID_DEPARTAMENTO` int(11) DEFAULT NULL,
  `NOMBRE` varchar(30) COLLATE latin1_spanish_ci NOT NULL,
  `APELLIDO` varchar(30) COLLATE latin1_spanish_ci NOT NULL,
  `USUARIO` varchar(20) COLLATE latin1_spanish_ci NOT NULL,
  `CONTRASENIA` varchar(32) COLLATE latin1_spanish_ci NOT NULL,
  `ESTADO` enum('Activo','Inactivo') COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`ID_USUARIO`),
  KEY `FK_POSEE` (`ID_PERFIL`),
  KEY `ID_DEPARTAMENTO` (`ID_DEPARTAMENTO`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci AUTO_INCREMENT=35;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `despacho`
--
ALTER TABLE `despacho`
  ADD CONSTRAINT `despacho_ibfk_1` FOREIGN KEY (`ID_USUARIO`) REFERENCES `usuario` (`ID_USUARIO`);

--
-- Filtros para la tabla `pedido`
--
ALTER TABLE `pedido`
  ADD CONSTRAINT `FK_REALIZA` FOREIGN KEY (`ID_USUARIO`) REFERENCES `usuario` (`ID_USUARIO`);

--
-- Filtros para la tabla `pedido_despacho_x_producto`
--
ALTER TABLE `pedido_despacho_x_producto`
  ADD CONSTRAINT `FK_GENERA` FOREIGN KEY (`ID_PEDIDO`) REFERENCES `pedido` (`ID_PEDIDO`),
  ADD CONSTRAINT `FK_PROBOCA` FOREIGN KEY (`ID_DESPACHO`) REFERENCES `despacho` (`ID_DESPACHO`),
  ADD CONSTRAINT `FK_REFERENCIA` FOREIGN KEY (`ID_PRODUCTO`) REFERENCES `producto`
  (`ID_PRODUCTO`);

--
-- Filtros para la tabla `producto`
--
ALTER TABLE `producto`
  ADD CONSTRAINT `FK_TIENE` FOREIGN KEY (`ID_DEPARTAMENTO`) REFERENCES `departamento`
  (`ID_DEPARTAMENTO`);

--
-- Filtros para la tabla `producto_x_bodega`
--
ALTER TABLE `producto_x_bodega`
  ADD CONSTRAINT `FK_ALMACENA` FOREIGN KEY (`ID_BODEGA`) REFERENCES `bodega` (`ID_BODEGA`),
  ADD CONSTRAINT `FK_CONTIENE` FOREIGN KEY (`ID_PRODUCTO`) REFERENCES `producto` (`ID_PRODUCTO`);
```

IMPLEMENTACION DE UN SISTEMA INFORMATICO PARA CONTROL DE INSUMOS Y VIVERES PARA LA ADMINISTRACION DE UN BARCO TURISTICO. APLICACIÓN AL BARCO TREASURE OF GALAPAGOS

```
--
-- Filtros para la tabla `usuario`
--
ALTER TABLE `usuario`
  ADD CONSTRAINT `FK_PERTENECE` FOREIGN KEY (`ID_DEPARTAMENTO`) REFERENCES `departamento`
  (`ID_DEPARTAMENTO`),
  ADD CONSTRAINT `FK_POSEE` FOREIGN KEY (`ID_PERFIL`) REFERENCES `perfil` (`ID_PERFIL`);

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

--
-- Base de datos: `treasure`
--

--
-- Volcado de datos para la tabla `perfil`
--

INSERT INTO `perfil` (`ID_PERFIL`, `NOMBRE`, `DESCRIPCION`) VALUES
(1, 'administrador', 'El encargado de administración del sitio.),
(2, 'solicitante', 'La persona que genera los pedidos para ser procesados.),
(3, 'aprovador', 'La persona que administra los pedidos y puede cambiarlos en 3 estados: solicitado, aprobado, cancelado.),
(4, 'controlador', 'La persona que revisa los estados de los despachos.),
(5, 'proveedor', 'La persona que genera y cambia el estado a los despachos.);

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

--
-- Base de datos: `treasure`
--

--
-- Volcado de datos para la tabla `usuario`
--

INSERT INTO `usuario` (`ID_USUARIO`, `ID_PERFIL`, `ID_DEPARTAMENTO`, `NOMBRE`, `APELLIDO`, `USUARIO`,
`CONTRASENIA`, `ESTADO`) VALUES
(1, 1, NULL, 'administradoruno', 'administradoruno', 'admin', 'e10adc3949ba59abbe56e057f20f883e', 'Activo');

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```