

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

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

“DESARROLLO E IMPLANTACIÓN DEL PROYECTO CÓDIGO DE
BARRAS PARA TAME CARGO”

NOMBRES

HÉCTOR EDUARDO MALDONADO ESPINOSA

PAÚL SEBASTIÁN LÓPEZ CRUZ

DIRECTOR: MTR. JAVIER CÓNDOR

QUITO, 2014

AGRADECIMIENTO

Agradezco a Dios por ser el programador de mi vida,
a mis padres, por todo su esfuerzo, amor y guía,
a mi hermana por todo su apoyo y cariño,
a mi tía Elba por su bondad y por ser como una madre,
a mis amigos quienes alegran más aún mi vida,
a mi cuñado por su apoyo, generosidad y sencillez,
a mis profesores por compartirnos su conocimiento,
a mis compañeros que fueron parte de mi segundo hogar.

Eduardo Maldonado

A los que siempre estuvieron,
a los que tuvieron buenos y malos propósitos
por cuanto todos ellos me han conducido a este momento.

A todos todo.

Paúl López C.

DEDICATORIA

Dedico este trabajo a Dios, Marcia, Elba, Isela, Eduardo, pilares fundamentales de mi vida,
a Camila y Rafaela, las princesas de la familia,
a mis abuelos, trabajadores ejemplares,
a mis tíos y primos, con quienes he pasado muchos momentos agradables,
a mis amigos, quienes siempre están en las buenas y en las malas,
a la memoria de los seres que ya no están.

Eduardo Maldonado

A nuestro Señor,
a mis padres
y a mi futura esposa.

Paúl López. C.

Índice

1	INTRODUCCIÓN.....	1
1.1	Tema.....	1
1.2	Datos de la Institución.....	1
1.2.1	Nombre.....	1
1.2.2	Actividad.....	1
1.2.3	Ubicación.....	1
1.2.4	Características.....	1
1.2.5	Contexto.....	1
1.3	Justificación.....	2
1.4	Diagnóstico.....	2
1.5	Objetivos.....	3
1.5.1	Objetivo General.....	3
1.5.2	Objetivos Específicos.....	3
2	FUNDAMENTOS TEORICOS.....	4
2.1	Marco Teórico.....	4
2.1.1	El transporte aéreo de carga.....	4
2.1.2	El código de barras.....	7
2.1.2.1	Elementos.....	8
2.1.3	El lector óptico.....	8
2.1.3.1	Principales razones para utilizar un lector de códigos de barras en un negocio	9
2.1.3.2	Cómo se leen los códigos de barras.....	10
2.1.4	El Asistente Personal Digital (PDA).....	11
2.1.5	La impresora térmica de etiquetas.....	11
2.1.5.1	Características.....	12

2.1.5.2	Desventajas.....	13
2.1.5.3	Diseño.....	13
2.1.5.4	Aplicaciones	13
2.2	Metodología y Técnicas.....	13
2.2.1	Metodología de desarrollo (Scrum y XP).....	13
2.2.1.1	Scrum	13
2.2.1.2	Extreme Programming (XP).....	16
2.2.2	Arquitectura de la aplicación.....	20
2.2.2.1	¿Qué es la arquitectura de Software?	20
2.2.2.2	¿Para qué sirve?.....	21
2.2.2.3	¿Qué es un arquitecto de Software?	21
2.2.2.4	Arquitecturas más comunes	21
2.2.2.5	Modelos o Vistas	25
2.2.3	Lenguaje de Programación	25
2.2.3.1	Componentes	26
2.2.4	Entorno de desarrollo integrado	27
2.2.4.1	Componentes.....	27
2.2.4.2	Características	27
2.2.4.3	Ventajas.....	27
2.2.5	Versionador de Código	28
2.2.5.1	Conceptos en el Versionamiento.....	28
2.3	Herramientas.....	29
2.3.1	Visual Studio 2010	29
2.3.1.1	Licenciamiento	30
2.3.2	Internet Information Server (IIS).....	31
2.3.2.1	Características	31

2.3.2.2	Arquitectura.....	31
2.3.3	Gestor de base de datos DB2.....	32
2.3.3.1	Características	32
2.3.3.2	Ventajas.....	33
2.3.3.3	Compatibilidad.....	33
2.3.4	Aqua Data Studio	33
2.3.5	Herramienta de Modelamiento PowerDesigner	34
2.3.5.1	Características generales	35
2.3.5.2	Objetivos de la herramienta CASE	35
2.3.6	Team Foundation Server	36
2.3.6.1	Capas de Team Foundation Server.....	36
2.3.6.2	Componentes de Team Foundation Server	36
3	SELECCIÓN DE ELEMENTOS PARA EL PROYECTO	39
3.1	Scrum y eXtreme Programming.....	39
3.1.1	Procesos de Scrum.....	39
3.1.2	Prácticas de eXtreme Programming	40
3.2	Web Forms	40
3.2.1	El rol de HTTP	41
3.2.2	El rol de HTML	41
3.2.3	El rol del Script en el lado del cliente.....	42
3.2.4	Rol de los Postback al servidor.....	43
3.2.4.1	Postbacks de acuerdo a ASP.NET.....	45
3.3	Web Services	45
3.3.1	Estándares de los Servicios Web.....	46
3.4	Visual C# 2010	47
3.4.1	Versiones	47

3.4.2	Características.....	47
3.4.3	Tipos de proyecto	48
4	APLICACIÓN DE UN SISTEMA DE CÓDIGO DE BARRAS AL CASO DE ESTUDIO “TAME CARGO”	50
4.1	Antecedentes.....	50
4.2	Necesidades	50
4.3	Requerimientos Funcionales – Casos de Uso.....	55
4.3.1	Recepción	55
4.3.2	Despacho	58
4.3.3	Chequeo	59
4.3.4	Entrega.....	61
4.4	Requerimientos no funcionales	63
5	DESARROLLO DEL SISTEMA.....	64
5.1	Definición del Product Backlog	64
5.2	Sprints.....	66
5.2.1	Sprint 1	66
5.2.1.1	Fase de Planificación.....	66
5.2.1.2	Fase de Seguimiento.....	71
5.2.1.3	Fase de Revisión.....	86
5.2.2	Sprint 2	88
5.2.2.1	Fase de Planificación.....	88
5.2.2.2	Fase de Seguimiento.....	93
5.2.2.3	Fase de Revisión.....	107
5.2.3	Sprint 3	109
5.2.3.1	Fase de Planificación.....	109
5.2.3.2	Fase de Seguimiento.....	113

5.2.3.3	Fase de Revisión.....	127
5.2.4	Sprint 4	130
5.2.4.1	Fase de Planificación.....	130
5.2.4.2	Fase de Seguimiento.....	135
5.2.4.3	Fase de Revisión.....	149
6	COMENTARIOS, CONCLUSIONES Y RECOMENDACIONES	153
6.1	Comentarios.....	153
6.2	Conclusiones.....	153
6.3	Recomendaciones	154
7	BIBLIOGRAFÍA	156

Índice de Figuras

Figura 2-1 Ejemplo de código de barras.....	8
Figura 2-2: Forma correcta de lectura	10
Figura 2-3: Forma incorrecta de lectura	10
Figura 2-4: Proceso Scrum	16
Figura 2-5: Arquitectura monolítica.....	22
Figura 2-6: Arquitectura cliente servidor	22
Figura 2-7: Arquitectura modelo vista controlador	23
Figura 2-8: Arquitectura de 3 niveles	24
Figura 2-9: Arquitectura de n niveles	25
Figura 2-10: Orden de ejecución de un lenguaje de programación.....	26
Figura 2-11: Arquitectura IIS	32
Figura 2-12: Estructura de Team Foundation Server	38
Figura 3-1: Ejecución de páginas ASP.NET usando el Internet Information Server como servidor HTTP, la librería ISAPI permite aprovechar los recursos de la plataforma .NET en el desarrollo de aplicaciones Web	41
Figura 3-2: Uso del atributo "action" en la etiqueta <form>.....	44
Figura 3-3: Datos transmitidos mediante el método "GET"	44
Figura 3-4: Uso del método POST	44
Figura 3-5: Los servicios Web XML permiten un alto grado de interoperabilidad	46
Figura 5-1: Product Backlog	65
Figura 5-2: Historias de Usuario incluidas en el Sprint 1	66
Figura 5-3: Sprint 1 Backlog	68
Figura 5-4: Esfuerzo de cada tarea para la primera historia de usuario	69
Figura 5-5: Esfuerzo de cada tarea para la segunda historia de usuario.....	70
Figura 5-6: Esfuerzo de cada tarea para la tercera historia de usuario	70

Figura 5-7: Esfuerzo de cada tarea para la cuarta historia de usuario	71
Figura 5-8: Esfuerzo de cada tarea para la quinta historia de usuario	71
Figura 5-9: Panel de Tareas al final del día 1	73
Figura 5-10: Panel de Tareas al final del día 2	74
Figura 5-11: Panel de Tareas al final del día 3	75
Figura 5-12: Panel de Tareas al final del día 4	76
Figura 5-13: Panel de Tareas al final del día 5	77
Figura 5-14: Panel de Tareas al final del día 6	78
Figura 5-15: Panel de Tareas al final del día 7	79
Figura 5-16: Panel de Tareas al final del día 8	80
Figura 5-17: Panel de Tareas al final del día 9	81
Figura 5-18: Panel de Tareas al final del día 10	82
Figura 5-19: Burn down chart	83
Figura 5-20: Reporte de defectos inicial.....	83
Figura 5-21: Reporte de defectos final	84
Figura 5-22: Extracto de una de las nuevas tablas integradas "Taquilla" y posibilidad de realizar el CRUD	86
Figura 5-23: Selección y despliegue de información de Manifiestos.....	86
Figura 5-24: Captura automática de peso marcado por la balanza.....	87
Figura 5-25: Despliegue de ubicaciones de entrega de paquetes en ciudad de destino.....	87
Figura 5-26: Botón que permite regresar al menú inmediato superior	87
Figura 5-27: Historias de Usuario incluidas en el Sprint 2	89
Figura 5-28: Sprint 2 Backlog	90
Figura 5-29: Esfuerzo de cada tarea para la primera historia de usuario	91
Figura 5-30: Esfuerzo de cada tarea para la segunda historia de usuario.....	91
Figura 5-31: Esfuerzo de cada tarea para la tercera historia de usuario	92

Figura 5-32: Esfuerzo de cada tarea para la cuarta historia de usuario	92
Figura 5-33: Esfuerzo de cada tarea para la quinta historia de usuario	93
Figura 5-34: Panel de Tareas al final del día 11	94
Figura 5-35: Panel de Tareas al final del día 12	95
Figura 5-36: Panel de Tareas al final del día 13	96
Figura 5-37: Panel de Tareas al final del día 14	97
Figura 5-38: Panel de Tareas al final del día 15	98
Figura 5-39: Panel de Tareas al final del día 16	99
Figura 5-40: Panel de Tareas al final del día 17	100
Figura 5-41: Panel de Tareas al final del día 18	101
Figura 5-42: Panel de Tareas al final del día 19	102
Figura 5-43: Panel de Tareas al final del día 20	103
Figura 5-44: Burn down chart	104
Figura 5-45: Reporte de defectos inicial.....	104
Figura 5-46: Reporte de defectos final	105
Figura 5-47: Ingreso de información del paquete según categoría.....	107
Figura 5-48: Categorización de paquetes en módulo de Chequeo	107
Figura 5-49: Control de usuarios con acceso al sistema.....	107
Figura 5-50: Despliegue de todos los destinos nacionales para envío de paquetes.....	108
Figura 5-51: Despliegue de información relacionada al usuario en ventana despacho.....	108
Figura 5-52: Historias de Usuario incluidas en el Sprint 3	110
Figura 5-53: Sprint 3 Backlog	111
Figura 5-54: Esfuerzo de cada tarea para la primera historia de usuario	112
Figura 5-55: Esfuerzo de cada tarea para la segunda historia de usuario	112
Figura 5-56: Esfuerzo de cada tarea para la tercera historia de usuario	113
Figura 5-57: Esfuerzo de cada tarea para la cuarta historia de usuario	113

Figura 5-58: Panel de Tareas al final del día 21	114
Figura 5-59: Panel de Tareas al final del día 22	115
Figura 5-60: Panel de Tareas al final del día 23	116
Figura 5-61: Panel de Tareas al final del día 24	117
Figura 5-62: Panel de Tareas al final del día 25	118
Figura 5-63: Panel de Tareas al final del día 26	119
Figura 5-64: Panel de Tareas al final del día 27	120
Figura 5-65: Panel de Tareas al final del día 28	121
Figura 5-66: Panel de Tareas al final del día 29	122
Figura 5-67: Panel de Tareas al final del día 30	123
Figura 5-68: Burn down chart	124
Figura 5-69: Reporte de defectos inicial.....	124
Figura 5-70: Reporte de defectos final	125
Figura 5-71: Botón que permite la impresión de la Taquilla.....	127
Figura 5-72: Modelo de taquilla impresa o reimpresa.....	127
Figura 5-73: Ventana que permite la reimpresión de taquillas.....	128
Figura 5-74: Despliegue de menú de opciones para módulo de recepción	128
Figura 5-75: Despliegue de menú principal de opciones	129
Figura 5-76: Historias de Usuario incluidas en el Sprint 4	131
Figura 5-77: Sprint 4 Backlog	132
Figura 5-78: Esfuerzo de cada tarea para la primera historia de usuario	133
Figura 5-79: Esfuerzo de cada tarea para la segunda historia de usuario.....	134
Figura 5-80: Esfuerzo de cada tarea para la tercera historia de usuario	134
Figura 5-81: Esfuerzo de cada tarea para la cuarta historia de usuario	135
Figura 5-82: Esfuerzo de cada tarea para la quinta historia de usuario.....	135
Figura 5-83: Panel de Tareas al final del día 31	136

Figura 5-84: Panel de Tareas al final del día 32	137
Figura 5-85: Panel de Tareas al final del día 33	138
Figura 5-86: Panel de Tareas al final del día 34	139
Figura 5-87: Panel de Tareas al final del día 35	140
Figura 5-88: Panel de Tareas al final del día 36	141
Figura 5-89: Panel de Tareas al final del día 37	142
Figura 5-90: Panel de Tareas al final del día 38	143
Figura 5-91: Panel de Tareas al final del día 39	144
Figura 5-92: Panel de Tareas al final del día 40	145
Figura 5-93: Burn down chart	146
Figura 5-94: Reporte de defectos inicial.....	146
Figura 5-95: Reporte de defectos final	147
Figura 5-96: Lectura de código de barras de la taquilla	149
Figura 5-97: Visualización de encargos pendientes de procesar.....	149
Figura 5-98: Pantalla que permite la modificación del estado de una taquilla.....	150
Figura 5-99: Ingreso de ubicación para paquetes chequeados	150
Figura 5-100: Opción para salir del Sistema desde el menú principal	151

Índice de Tablas

Tabla 3-1: Versiones de C#	47
Tabla 4-1: Reconocimiento de necesidades.....	55
Tabla 5-1: Encuesta de la reunión de retrospectiva del equipo del sprint 1	85
Tabla 5-2: Programa de mejora del sprint 1	88
Tabla 5-3: Encuesta de la reunión de retrospectiva del equipo del sprint 2	106
Tabla 5-4: Programa de mejora del sprint 2	109
Tabla 5-5: Encuesta de la reunión de retrospectiva del equipo del sprint 3	126
Tabla 5-6: Programa de mejora del sprint 3	130
Tabla 5-7: Encuesta de la reunión de retrospectiva del equipo del sprint 4	148
Tabla 5-8: Programa de mejora del sprint 4	152

1 INTRODUCCIÓN

En este capítulo, se expondrá de manera general el proyecto “Código de Barras para TAME Cargo”, brindando un enfoque preliminar sobre el tema a tratar en este documento. Se definirá el problema que se desea solventar, y los resultados esperados.

1.1 Tema

Desarrollo e implantación del Proyecto Código de Barras para TAME Cargo.

1.2 Datos de la Institución

1.2.1 Nombre

TAME EP.

1.2.2 Actividad

Aerocomercial.

1.2.3 Ubicación

Avenida Amazonas N24-260 y Colón. Edificio TAME.

1.2.4 Características

Empresa Pública TAME Línea Aérea del Ecuador, es la aerolínea pionera del Ecuador, su principal actividad es el transporte aéreo de personas y carga en rutas nacionales e internacionales.

Con respecto al transporte aéreo de carga, su compromiso es: “satisfacer las necesidades de transporte de carga, a través de una amplia oferta de productos, desde Carga General hasta aquellos productos que por su ciclo de vida, naturaleza, manejo o prioridad necesitan un especial cuidado, como: Animales Vivos, Mercancías Peligrosas, Restos Humanos, Carga Perecedera y Otros”.

1.2.5 Contexto

TAME EP inicialmente estaba ligada directamente a la Fuerza Aérea Ecuatoriana, sin embargo, en la actualidad forma parte de las empresas del Estado; cabe mencionar que su funcionamiento es totalmente autónomo en los campos administrativo y financiero, no recibe

aportaciones gubernamentales y crece sustentada en los ingresos que genera su propia actividad.

1.3 Justificación

Sin duda alguna, la vertiginosidad con que la tecnología ha pasado de ser un lujo a una necesidad, ha llevado a las empresas a emprender una evaluación y reestructuración de sus procesos, desde los más básicos hasta los más complejos, a fin de optimizar tiempo, dinero y recurso humano.

Precisamente este es el caso que afronta TAME EP. La propuesta para nuestro proyecto, es el mejoramiento de los procesos que intervienen en el servicio de transporte de paquetes TAME CARGO.

El principal motivo para desarrollar este tema, es el de aumentar la eficiencia de los procesos que intervienen en TAME CARGO, mediante el uso de dispositivos tecnológicos como Asistentes Personales Digitales (PDA) y lectores ópticos, que no representan mayor inversión y traerán réditos importantes a la empresa; estos dispositivos se emplearán para automatizar procesos como la lectura de la información de un paquete a partir de un código de barras, evitando así el desperdicio de recursos que se ven reflejados en perjuicio para la empresa e insatisfacción de los clientes.

1.4 Diagnóstico

Antes del inicio del presente proyecto, podíamos encontrarnos con un proyecto dependiente de recurso humano cuyo horario laboral debía ajustarse a la demanda de los clientes para garantizarles el servicio, esta condición, derivaba en un servicio ineficiente por una sobrecarga de trabajo.

El único sistema que gobernaba al servicio TAME CARGO, era SIGETAME, creado en SNAP, un ambiente de trabajo integrado para el desarrollo y mantenimiento de sistemas sobre el servidor AS/400 de IBM.

Debido a la creciente demanda del servicio TAME CARGO, se puso en evidencia la necesidad de una aplicación que permita agilizar los procesos involucrados desde que el cliente entrega un paquete, hasta que éste es entregado en su destino.

Se pretende con este proyecto aumentar el nivel de satisfacción del cliente para contribuir con los valores institucionales de la empresa.

1.5 Objetivos

1.5.1 Objetivo General

- Desarrollar e Implementar el Proyecto “Código de Barras para TAME Cargo”.

1.5.2 Objetivos Específicos

- Desarrollar servicios web.
- Desarrollar aplicación web optimizada para computadores de escritorio
- Desarrollar aplicación web optimizada para Asistentes Personales Digitales.
- Reducir el uso de papel al minimizar impresiones.
- Llevar de una manera más adecuada el registro de los paquetes transportados.
- Familiarizar a los involucrados con el uso de nuevas tecnologías.

2 FUNDAMENTOS TEORICOS

En este capítulo, se analizarán a profundidad los agentes que tienen injerencia directa sobre el proyecto:

- Aspectos teóricos técnicos y de negocio.
- Metodología de desarrollo que se utilizará para gestionar el proyecto.
- Herramientas que se emplearán durante el desarrollo del proyecto.

2.1 Marco Teórico

2.1.1 El transporte aéreo de carga

Día tras día cientos de aviones surcan el cielo, el transporte aéreo es el más seguro y rápido de todos los medios de transporte. No solamente se transportan pasajeros, sino que llevan sin descanso toda clase de mercancías de un extremo del planeta a otro. Las distancias y el tiempo no son un obstáculo, todo viaja por los aires y generalmente llega puntualmente a su destino.

Cada día varios medios de transporte, entre ellos, los aviones, transportan mercancías mediante una intrincada red de transporte en todo el planeta. A lo largo de esta red, un ciclo continuo de carga y descarga mantiene repletos los almacenes y las fábricas en actividad. En cada escala hay una zona de carga. Quizás la mayoría no se haya detenido a pensar en las zonas de carga, pero quienes forman parte de la industria del transporte saben que es el punto fundamental de sus negocios.

(Uriza, 2011) nos da a conocer que el punto de conexión que logra unir el transporte y el almacenaje es la zona de carga. Hoy en día los trabajadores van desde los camiones hacia los almacenes y viceversa en elevadores de paleta a motor. La actividad es frenética y debe ser así. Los empleados son los encargados de hacer llegar los encargos a cientos o miles de kilómetros. La actividad se mantiene los 7 días a la semana.

Todo comienza cuando los camiones que recogen la carga proveniente de los aviones se sitúan en la puerta de la zona de carga. Debido a la gran cantidad de productos, cada uno debe presentar una etiqueta con un código de barras. Los agentes utilizan escáneres de mano especiales para verificar cada paquete. Luego los operadores conducen cada mercancía hasta el almacén y estos son colocados en estanterías que a veces superan los 3 metros de altura. Pero eso solo es la mitad del trabajo ya que los miles de productos recibidos deben ser

entregados y con rapidez. Los artículos perecederos como los lácteos o las rosas deben entregarse primero.

Los aviones revolucionaron la industria del transporte de mercancías. Actualmente UPS lidera esta revolución dado que esta empresa moviliza 13,5 millones de paquetes diariamente. A continuación se detalla cómo nos puede ayudar un proceso automatizado. Escáneres leen y registran las especificaciones presentadas en el recibo de envío de cada paquete, la informática permite que un sistema de transportadoras computarizado traslade cada paquete a la siguiente área de carga donde los operadores lo colocan de nuevo en contenedores para continuar su viaje. La automatización es realmente singular. Básicamente los únicos seres humanos que manipulan los paquetes son quienes los cargan y descargan. Después de su introducción en las instalaciones “todo depende de una etiqueta inteligente la cual contiene información que se lee electrónicamente y que guía al paquete” (Uriza, 2011) a través del sistema y que lo traslada hacia el contenedor adecuado para su despacho.

Pero no siempre fue tan fácil, en los inicios del transporte aéreo toda la labor era manual, fundamentalmente la carga y la clasificación de los paquetes eran realizadas por personas que los colocaban en sacos de embalaje y los introducían en aviones de hélice que los llevaban a sus destinos. En 1911 el correo de Estados Unidos comenzó a prestar el servicio de correo aéreo y contrataba a compañías externas para el vuelo de los aviones. A finales de los años '20 la mayoría de los aviones utilizados eran los biplanos Douglas M2. Los contenedores ubicados entre el piloto y el motor contenían cerca de media tonelada de envíos, pero al igual que los demás aviones, la carga debía realizarse minuciosamente a mano. Durante los 15 años siguientes, los adelantos de la aviación permitieron el transporte de una mayor cantidad de carga con mayor eficiencia. Pero hubo que esperar hasta el estallido de la segunda guerra mundial para que el planeta fuera testigo de las verdaderas posibilidades del transporte aéreo. Durante la segunda guerra mundial fue necesario movilizar toneladas de carga para abastecer los frentes como por ejemplo: municiones, alimentos, ropa, era preciso transportar todo aquello que puede considerarse como necesario para apoyar al ejército.

El motor a reacción permitió que los aviones se convirtieran en eficientes transportes de carga. La Fuerza Aérea Estadounidense fue una de las primeras en reconocer las virtudes del transporte de Jets. A comienzos de los '60 el pentágono puso en servicio el C141, cuya zona de carga formaba parte del avión. Versiones rudimentarias de plataformas rodantes y

montacargas especialmente diseñadas para este avión permitían a los aviadores cargar con rapidez las paletas con armas y provisiones. Pero la rapidez y facilidad de cómo se realizaba la carga carecían del método de entrega. Los paracaídas se adaptaban a todo tipo de carga cuando el C141 volaba sobre el lugar de descarga, los aviadores arrojaban las mercancías por la parte trasera del avión. El área de recepción era cualquier lugar sobrevolado por el avión.

Hoy en día esta innovación militar se ha convertido en líder de esta actividad. La interacción de las modernas zonas de carga aéreas y la aviación implica que prácticamente todo pueda llegar al otro lado del planeta en cuestión de horas. A medida que aumenta la necesidad de producir mercancías de manera copiosa, la diferencia entre las zonas de carga y los depósitos es cada vez menor. Hasta hace poco los industriales construían los depósitos pensando en su tamaño y capacidad de almacenamiento, pero ahora la producción y su volumen son los factores relevantes.

Actualmente los diseñadores hacen uso de la más avanzada tecnología mucho antes de construirse el almacén. El diseño asistido por computadora o sistema CAD permite a los constructores de los almacenes adelantarse al futuro. Este sistema puede utilizarse para evaluar el diseño que se transfiere electrónicamente a un programa de simulación. Este programa permite observar el diseño en movimiento, de hecho, es posible observar la mercancía recorriendo las transportadoras. Los diseñadores utilizan las computadoras a fin de resolver cualquier falla pero estas máquinas no solo se utilizan durante el diseño ya que actualmente se desarrollan depósitos computarizados y sistemas de almacenaje y recuperación automatizados. En el interior de las instalaciones modernas, vehículos controlados por computadoras están ejecutando tareas antes realizadas por humanos. Los vehículos automatizados han revolucionado la tecnología en esta área, incluso se los puede considerar como unos robots ya que el hombre no interviene en esto. Son el equivalente de un vehículo capaz de memorizar la distribución del almacén. Los sensores del vehículo están conectados con señales fijas o con un software de la distribución del lugar a fin de garantizar que la máquina llegue a su destino.

El próximo reto de la industria es un centro controlado por máquinas. Como si se tratara de una novela de ciencia ficción, en los depósitos del futuro, máquinas y computadoras almacenarán, clasificarán, recuperarán y cargarán las mercancías sin la intervención del hombre (Beltrán, 2013).

El negocio del transporte de carga está en auge, “anualmente se desplazan por el mundo 36 millones de toneladas de mercancías de toda clase” (Beltrán, 2013), conectando sociedades, mercados y culturas. Los aviones de carga duplican su tamaño cada década. Con el paso del tiempo, la movilización de la carga vía aérea pasó de ser excepcional a algo corriente. Varias aerolíneas del mundo encuentran su principal fuente de ingresos en el traslado de paquetes, convirtiendo a los ingresos de las ventas de tickets aéreos en un ingreso secundario.

La empresa de bandera nacional Tame ofrece el servicio de transporte de carga y correo, el cual es seguro y eficiente con una amplia cobertura nacional e internacional. La empresa dispone de la capacidad de carga de su moderna flota, lo que le permite ofrecer el servicio diariamente a diferentes ciudades del país en una amplia gama de horarios, todo esto con el afán de satisfacer las necesidades de transporte de carga de sus clientes.

El transporte aéreo de mercancías es realizado por personal especializado y experto en la manipulación de todo tipo de encargos, sean estos perecibles, animales vivos, mercancías peligrosas y carga en general. El equipo de despachadores de aviones debe organizar la dificultosa logística para cumplir su tarea, no es tarea fácil, ya que los depósitos de los aeropuertos están colmados por un gran número de cajas y los operadores de carga tienen que asegurarse que las mercancías estén embaladas correctamente.

Por esta razón, Tame requiere que se agilicen los procesos que controlan el traslado de carga, ya que se tiene planificado una expansión de destinos tanto nacionales como internacionales, con el propósito de aumentar sus ingresos y demostrar a sus clientes que son una orgullosa empresa ecuatoriana suficientemente madura, capaz de proveer un servicio de transporte confiable.

Como dato referencial, para tener una idea del gran beneficio que podría causar una expansión de rutas internacionales, podemos apreciar que el total facturado en el año 2012 a nivel nacional fue de \$4.812.318,16 por el traslado de 8 511 956 paquetes.

2.1.2 El código de barras

En las últimas décadas, los códigos de barra se han convertido en un elemento esencial del comercio internacional. En la actualidad, es muy difícil encontrar un producto que no tenga un código de barras único asociado, que permite identificarlo en cualquier parte del planeta de manera ágil y sin errores.

Contrario a muchas creencias, el código de barras no contiene información del producto, meramente es la representación de un número que permite a los comercios identificar el producto y obtener y/o procesar la información necesaria.

2.1.2.1 Elementos



Figura 2-1 Ejemplo de código de barras

En la Figura 2.1, se puede evidenciar un ejemplo de código de barras, sus elementos son:

Tipo: Define el tipo de código de barras. Es el dígito en el extremo izquierdo del código de barras. En el ejemplo es el 1.

Fabricante: Identifica el fabricante del producto. Son los 5 primeros dígitos abajo del código de barras. En el ejemplo son los dígitos 38055.

Producto: Identifica el producto en sí, correspondiente al fabricante ya identificado. Son los 5 últimos dígitos abajo del código de barras. En el ejemplo son los dígitos 65154.

Dígito de verificación: Sirve como control para el dispositivo de lectura, después de escanear un código de barras, este dígito es enviado a la computadora quien revisa toda la información recibida y confirma al lector que toda la información se ha escaneado correctamente. Es el dígito en el extremo derecho del código de barras. En el ejemplo es el 7.

El código de barras en sí, está compuesto por líneas y espacios de distinto grosor, que permiten identificar cada uno de los dígitos y convertirlos en información inteligible (Ellen, s.f.).

2.1.3 El lector óptico

Es un dispositivo electrónico que nos permite la lectura de códigos de barras impresos. Como un lector de superficie plana, está conformado por un lente y un sensor liviano el cual

convierte impulsos ópticos en impulsos eléctricos. Adicionalmente, casi todos los lectores de códigos de barras contienen un circuito decodificador que analiza la imagen generada por el sensor y envía el contenido del código de barras al puerto de salida del dispositivo.

También son conocidos como *Scanners*, son los dispositivos de entrada de código de barras más populares y su función consiste en recuperar la información contenida en los mismos mediante un láser, convertirla en una señal eléctrica y por último transformarla en unos y ceros (lenguaje binario de las computadoras) (Anónimo, Escáner de código de barras, 2014).

El lector óptico permite la transmisión de datos a cualquiera de los siguientes elementos: un documento de Word, el block de notas, un archivo de Excel o cualquier campo de entrada de texto de una aplicación. Los datos son enviados al computador emulando un teclado en donde la lógica de la aplicación se encarga de procesarlos.

Un lector láser no necesita estar cerca del código de barras para hacer su trabajo, ya que puede leerlo a una distancia de 15 a 75 centímetros dependiendo de las características del modelo hasta un máximo de 9 metros de distancia.

Los precios de los lectores láser varían desde \$200 hasta \$2000 dólares y vienen en una gran variedad de modelos.

Los lectores de código de barras son muy importantes para realizar el seguimiento de algún producto o paquete a lo largo de todo el trayecto por el que circula el objeto.

2.1.3.1 Principales razones para utilizar un lector de códigos de barras en un negocio

- Seguridad para los inventarios
- Rapidez de atención al cliente
- Control de errores y fraudes
- Evita duplicación de datos
- Control de existencias
- Inversión muy baja (Desde \$50)
- Automatiza las ventas
- Automatiza las compras
- Controles adicionales
- Codificación mundial

2.1.3.2 Cómo se leen los códigos de barras

Los lectores ópticos iluminan el código con una luz roja. El sensor del scanner detecta la luz reflejada y genera una señal que representa la intensidad de la reflexión. Un decodificador recibe la señal, la interpreta y valida que el código de barras es legible, posteriormente lo convierte en texto ASCII, y lo envía al computador. En las Figuras 2-2 y 2-3 tenemos ejemplos de lectura correcta e incorrecta respectivamente.

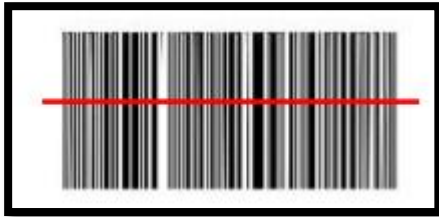


Figura 2-2: Forma correcta de lectura



Figura 2-3: Forma incorrecta de lectura

2.1.4 El Asistente Personal Digital (PDA)

La principal razón para la existencia del Asistente Personal Digital, es la de actuar como un organizador electrónico, concentrando todas las funcionalidades necesarias para que el usuario pueda manejar sus actividades diarias de manera eficiente.

Otra de las principales funcionalidades del Asistente Personal Digital, es la de poder sincronizarse con la computadora, lo que evidencia que el PDA nunca fue concebido como un reemplazo para la computadora, pero sí como una extensión de la misma.

Siguiendo la tendencia tecnológica normal, el Asistente Personal Digital ha sufrido una importante evolución, pasando de tener capacidades limitadas enfocadas a la organización personal, a convertirse en un dispositivo capaz de integrar tecnologías como 3G, 4G, Bluetooth y GPS por nombrar algunas. En lo referente a hardware también se puede evidenciar una evolución significativa, en la actualidad existen PDA's que integran cámaras fotográficas, lectores biométricos y escáneres de código de barras.

Los principales sistemas operativos disponibles para el Asistente Personal Digital, son PalmOS y Windows Mobile, el último incluye su gama de ofimática (Word, Excel, Outlook, etc.), lo que lo convierte en una herramienta poderosa y versátil que brinda la posibilidad de abrir y editar documentos sin la necesidad de estar en una computadora.

Con la creciente penetración de los celulares inteligentes y tablets en el mercado, el Asistente Personal Digital ha visto una reducción significativa en sus ventas, sin embargo existen ciertas funcionalidades (principalmente de trabajo) que lo hacen único, lo cual le garantizan una porción del mercado (Carmack & Freudenrich, n.d.).

2.1.5 La impresora térmica de etiquetas

La impresión termal es un proceso digital de impresión, el cual produce la impresión de una imagen por intermedio del calentamiento selectivo de un papel termal como es comúnmente conocido. Este papel pasa sobre el cabezal de impresión termal. El revestimiento del papel se torna negro en las áreas donde es calentado, produciendo una imagen. Existen impresoras que pueden imprimir en dos tonos de colores, el negro y el rojo, simplemente calentando la etiqueta a dos temperaturas diferentes.

Estas están dentro de la categoría de impresoras que no son de impacto.

Una impresora térmica se basa en una serie de agujas calientes que van recorriendo un papel especial (termosensible) que al contacto se vuelve de color negro. Son muy usadas en los cajeros y supermercados por su bajo coste.

Para lograr la impresión, el cabezal térmico especifica qué pins deben calentarse y a qué temperatura deben de hacerlo.

En este tipo de impresoras, los costos de impresión son bajos debido a que no consume más que el propio papel. Es posible que se logre una impresión en un único color que casi siempre es el negro.

La durabilidad de la impresión depende de la calidad de la etiqueta, aunque en la mayoría de los casos es baja debido a que muchas veces se pierde el texto impreso en la misma por las elevadas temperaturas.

La marca de la impresora a utilizar en nuestro proyecto será “Zebra” y el modelo será el “GK 420t”. Los modelos GK son de mejor valor para una impresora térmica de escritorio. Nos permite una impresión térmica directa a una velocidad de hasta 5 pulgadas por segundo.

2.1.5.1 Características

- Son económicas: sólo debemos preocuparnos por el papel ya que no utilizan tinta
- Son silenciosas y compactas
- Son fáciles de utilizar
- Diseño “Open Access” para simplificar la carga de papel
- Facilidad de recarga de cinta de etiquetas debido a su buen diseño
- Indicadores de color del panel de control
- Tiene un procesador de 32 bits con el cual se consigue un rendimiento mejorado de la impresora
- Velocidades de impresión rápidas
- Detección y configuración adaptativa automática del cable de puerto de serie para la integración plug-and-play
- La impresora viene con un CD en el cual podemos encontrar los instaladores, programas que nos permitirán diseñar etiquetas y manuales de usuario.
- Posibilidad de realizar el respaldo de la configuración de una impresora para posteriormente copiar la misma configuración en otras impresoras

- Genera informes sobre el mantenimiento del cabezal de impresión activado y personalizado por el usuario
- Distribución de etiquetas
- Servidor de impresión 10/100 interno e interfaz Ethernet

2.1.5.2 Desventajas

- Baja calidad de impresión
- Impresiones poco duraderas. Con el paso del tiempo las impresiones se van volviendo ilegibles, hasta que a veces terminan desapareciendo y con más razón si se le suman condiciones como la exposición al sol y la lluvia.

2.1.5.3 Diseño

La impresora termal comprende estos componentes clave:

- Cabezal termal: genera el calor e imprime sobre el papel
- Rodillo: alimenta de papel
- Resorte: aplica presión sobre el cabezal termal, generando el contacto con el papel termo sensible
- Tablero de control: controla el mecanismo

2.1.5.4 Aplicaciones

Las impresoras termales imprimen con menos vibración y son usualmente más rápidas que las impresoras de impacto (de matriz de puntos). Son más pequeñas, más ligeras y consumen menos energía, haciéndolas más portables. Los lugares en donde podemos encontrar este tipo de impresoras son: estaciones de gasolina, quioscos de información, puntos de ventas, bodegas, hospitales, etc.

2.2 Metodología y Técnicas

2.2.1 Metodología de desarrollo (Scrum y XP)

2.2.1.1 Scrum

Scrum es un marco de referencia de desarrollo de software ágil, iterativo e incremental. Se enfoca en la colaboración directa entre los miembros del equipo a fin de alcanzar un objetivo común. Una de las grandes ventajas de Scrum, es su adaptabilidad para cambios consecuencia de requerimientos de los clientes o por imprevistos presentados durante el desarrollo, justamente esta característica da a Scrum su nombre, el cual proviene del

deporte Rugby, donde Scrum significa reiniciar el juego posterior a una infracción accidental o cuando el balón ha salido de juego (Anónimo, Proyectos Agiles, s.f.).

2.2.1.1.1 Fases

En Scrum podemos identificar 3 fases claramente definidas:

2.2.1.1.1.1 Planificación del Sprint

La primera actividad en Scrum, es definir el Product Backlog, el cual agrupará todos los requerimientos definidos por el dueño del producto quien les asignará una prioridad. Este documento puede ser modificado en caso de ser necesario.

Una vez que el Product Backlog está creado, se procede a definir el Sprint Backlog, que selecciona ítems del Product Backlog para desarrollarlos en el Sprint, realizando la estimación de esfuerzo correspondiente.

2.2.1.1.1.2 Ejecución del Sprint

En la ejecución del Sprint, se empiezan a desarrollar los ítems del Sprint Backlog. La parte clave de esta fase son las reuniones diarias que se mantienen entre el Scrum Master y el equipo de desarrollo. En estas reuniones se abordan 3 temas principales para cada integrante del equipo:

- ¿Qué trabajo he realizado desde la reunión anterior?
- ¿Qué trabajo voy a realizar hasta la próxima reunión?
- ¿Qué problemas he encontrado que impiden realizar el trabajo según lo planificado?

2.2.1.1.1.3 Revisión del Sprint

Esta fase ocurre cuando el Sprint ha finalizado, consta de 2 partes:

Demostración: Consiste en la presentación del resultado del Sprint al dueño del producto y partes interesadas, quienes retroalimentarán al equipo sobre su percepción de los resultados alcanzados, lo que será tomado en cuenta para posteriores Sprints.

Retrospectiva: El equipo de desarrollo se reúne con el Scrum Master y se analiza a fondo el trabajo realizado, así como obstáculos y problemas surgidos, a fin de mejorar su productividad para posteriores Sprints.

2.2.1.1.2 Roles

2.2.1.1.2.1 Scrum Master

Es el líder del proyecto, también conocido como Facilitador. Sirve como nexo entre el equipo de desarrollo y el dueño del producto. Entre sus principales funciones tenemos:

- Liderar las reuniones diarias.
- Velar por el seguimiento de la metodología.

Facilitar el trabajo del equipo de desarrollo; siempre que un integrante del equipo encuentre un obstáculo será el Scrum Master el encargado de solventarlo, por lo que deberá tener una posición suficiente dentro de la organización que le permita realizar los escalamientos pertinentes en caso de ser necesario.

2.2.1.1.2.2 Dueño del producto

Es la persona que tiene un amplio conocimiento del negocio, encargada de obtener el mejor producto posible para el cliente. Su principal responsabilidad es la de mantener el Product Backlog actualizado con sus priorizaciones respectivas.

2.2.1.1.2.3 Equipo de desarrollo

Se recomienda que el equipo de desarrollo tenga entre 3 y 9 integrantes multidisciplinarios, de manera que no se dependa de un integrante para desarrollar una funcionalidad en particular. Los equipos de desarrollo tienen independencia para organizarse y dividir el trabajo según su criterio a fin de cumplir el Sprint (Anónimo, Dos Ideas, 2013).

2.2.1.1.3 Artefactos

Scrum define los siguientes artefactos en su proceso:

2.2.1.1.3.1 Product Backlog

Agrupar todas las funcionalidades que el producto debe tener. Es responsabilidad del Dueño del Producto mantenerlo actualizado con su priorización respectiva.

2.2.1.1.3.2 Sprint Backlog

Son los ítems seleccionados del Product Backlog para ser desarrollados en el Sprint Actual.

2.2.1.1.3.3 Incremento del Producto

Es el resultado de cada Sprint, es la funcionalidad adicional agregada al producto y que ha sido aceptada por el Dueño del Producto basado en los criterios establecidos.

2.2.1.1.4 Proceso

En la Figura 2-4 se aprecia el ciclo de vida de cada iteración o sprint.

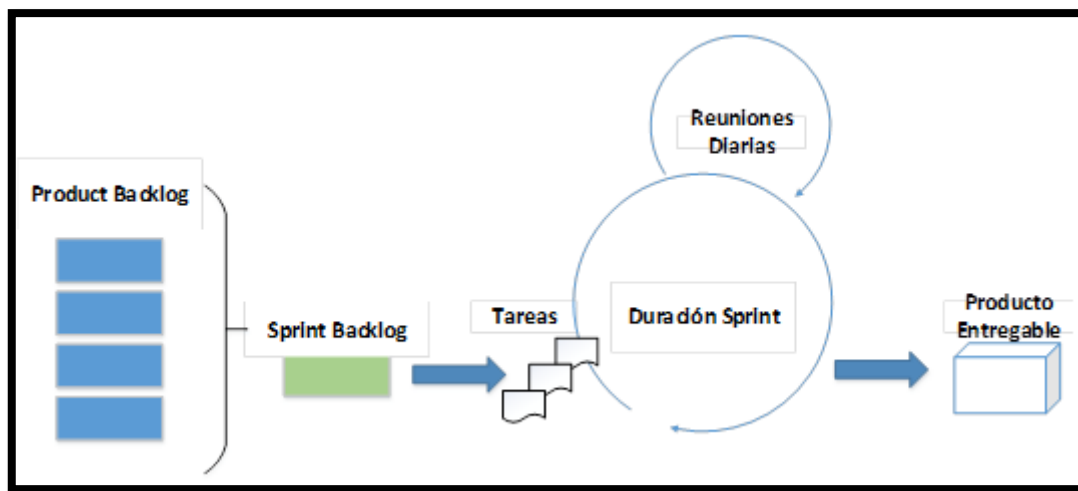


Figura 2-4: Proceso Scrum

2.2.1.2 Extreme Programming (XP)

Al igual que Scrum, Extreme Programming es un marco de referencia de desarrollo de software ágil. Su principal característica es la de enfocarse en la satisfacción del cliente. El concepto de Extreme Programming es el de entregar funcionalidades cuando el cliente las necesita, en lugar de acumular funcionalidades en un solo paquete que será entregado en una fecha específica. Extreme Programming es un marco de referencia adaptativo, es decir responde rápido a cambios en los requerimientos de los clientes, para lograr esto, empodera al equipo de desarrollo, dándoles poder de decisión y la capacidad para auto organizarse (Wells, 2013).

2.2.1.2.1 Principios

2.2.1.2.1.1 Comunicación

El desarrollador está en constante comunicación con el resto de equipo de desarrollo y con los clientes logrando detectar cualquier divergencia en los conceptos del producto antes de que maduren, lo que ayuda a que su solución sea menos costosa. Todos los integrantes del equipo estarán involucrados en cada fase del desarrollo, desde los requerimientos hasta la codificación.

2.2.1.2.1.2 *Simplicidad*

La simplicidad se evidencia en 2 aspectos. Primero en el ambiente de trabajo, manteniéndolo simple y productivo a la vez. Segundo en su diseño, haciéndolo lo más simple y limpio posible, de manera que cualquier integrante del equipo tenga claro cada aspecto. Cada funcionalidad será desarrollada de acuerdo a su concepto, sin excederlo ni complicarlo innecesariamente.

2.2.1.2.1.3 *Retroalimentación*

Se busca retroalimentación de todas las partes involucradas, para alcanzar esto se realizan pruebas constantemente de las funcionalidades nuevas y existentes. Cada iteración entrega software que funciona. Los procesos se adaptan el proyecto.

2.2.1.2.1.4 *Respeto*

Fundamental en cualquier trabajo en equipo. Todos los integrantes contribuyen de cualquier manera. Se valora el conocimiento de las contrapartes correspondientes (Cliente – Desarrollador) buscando el mejor resultado posible.

2.2.1.2.1.5 *Honestidad y Voluntad*

Es imprescindible ser completamente honesto en lo que respecta a estimaciones y progreso realizado. Los problemas encontrados no son del desarrollador, al contrario, involucran a todo el equipo quien se adaptará a cualquier cambio que se requiera.

2.2.1.2.2 *Fases*

Extreme Programming define las siguientes fases (Nayab, 2011):

2.2.1.2.2.1 *Planificación*

Es la primera fase de Extreme Programming donde la primera actividad es una reunión entre los clientes/usuarios y el equipo de desarrollo donde se definen historias de usuario¹. Una vez recopiladas, se asigna una prioridad a cada historia de usuario. De igual manera se establecen pruebas de aceptación de usuario basadas en las historias de usuario.

¹ Una historia de usuario es una herramienta de las metodologías de desarrollo de software ágiles. Consisten en expresar en lenguaje común un requerimiento de software. Se debe buscar que sean lo más independientes posibles.

2.2.1.2.2.2 Diseño

La fase de diseño es la primera fase iterativa en Extreme Programming. La primera actividad es seleccionar historias de usuario según su prioridad y agruparlas en la iteración actual, esta selección debe realizarse con los clientes/usuarios. En la primera iteración, se define la arquitectura de la aplicación, así como estándares de codificación como nomenclatura para las variables, métodos, estructuras, etc.

2.2.1.2.2.3 Codificación

Constituye la parte más importante de las iteraciones en Extreme Programming. Se deben seguir los estándares definidos en la etapa de diseño. Es importante que el usuario/cliente tenga disponibilidad absoluta para tomar decisiones en caso de que se evidencie la necesidad.

2.2.1.2.2.4 Pruebas

Las pruebas son definidas durante el desarrollo. Todo código implementado debe tener su prueba unitaria respectiva a fin de evitar bugs. Todas las pruebas unitarias deben pasarse antes de la liberación de una versión.

2.2.1.2.2.5 Retroalimentación

La fase de retroalimentación tiene lugar posteriormente a la fase de pruebas. Esta fase es muy importante para identificar los aspectos positivos que deben repotenciarse y los negativos que deben corregirse. Esta fase también es importante para los clientes/usuarios quienes tendrán un panorama mucho más claro respecto al producto que se está desarrollando y problemas y/o funcionalidades potenciales. Esta es la última fase iterativa en Extreme Programming.

2.2.1.2.2.6 Cierre del proyecto

El cierre del proyecto ocurre cuando todas las historias de usuario han sido exitosamente implementadas, lo cual indica que el producto está listo. Una vez llegado a este punto, se debe levantar toda la documentación pertinente, considerando que no existirán cambios en el futuro inmediato.

2.2.1.2.3 Roles

Extreme Programming define los siguientes roles en su metodología (Chromatic):

2.2.1.2.3.1 Cliente

Es el encargado de definir las historias de usuario, con su prioridad y prueba de aceptación respectivas. Tiene la idea clara del alcance del producto.

2.2.1.2.3.2 Desarrollador

Es el encargado de convertir las historias de usuario en código. Realiza estimaciones del trabajo requerido para completar cada historia de usuario. En caso de ser necesario se debe reunir con el cliente para que las 2 partes estén claras sobre lo que se quiere obtener y el esfuerzo implicado. Define las pruebas unitarias para el código desarrollado.

2.2.1.2.3.3 Tracker

Se encarga de dar seguimiento al calendario establecido. Determinando si está progresando o no de acuerdo a las estimaciones realizadas por el equipo de desarrollo.

2.2.1.2.3.4 Supervisor

Es la persona con más conocimiento de la metodología y/o experiencia. Se encarga de hacer cumplir la metodología por todas las partes involucradas. De igual manera es el primer nivel de escalamiento para el equipo de desarrollo.

2.2.1.2.3.5 Probador

Asiste al cliente en la escritura de las pruebas de aceptación. Se encarga de ejecutarlas regularmente basado en un calendario establecido con el equipo de desarrollo y comunica los resultados oportunamente.

2.2.1.2.4 Prácticas

Extreme Programming define las siguientes prácticas (Wells, 2013):

2.2.1.2.4.1 Liberaciones pequeñas o cortas

Extreme Programming se enfoca en constantes liberaciones de versiones del producto, manteniendo la simplicidad por encima de todo.

2.2.1.2.4.2 Propiedad colectiva

El código es de propiedad de todo el equipo. Todos los desarrolladores están en completa capacidad de dar mantenimiento a cualquier parte de del código evitando la dependencia de miembros en particular.

2.2.1.2.4.3 Refactorización continua del código

El código está sujeto a continuo análisis para mejorarlo, simplificarlo y evitar cualquier duplicidad innecesaria.

2.2.1.2.4.4 Pruebas unitarias antes de codificar

Se escriben pruebas unitarias para el código antes de implementarlo, de manera que se tenga plena conciencia de que el código funciona de acuerdo a su diseño.

2.2.1.2.4.5 Programación en pares

Una de las principales características de Extreme Programming, es la de programación en pares. Esta práctica dicta que un desarrollador se encuentra en el ordenador codificando, mientras otro se encuentra a su lado validando que la codificación sea correcta y enfocada hacia el objetivo final manteniendo los estándares acordados. Las parejas deben ir alternando, lo que garantiza la propiedad colectiva.

2.2.1.2.4.6 Integración continua

Todo el equipo de desarrollo debe trabajar sobre la última versión del producto, la cual debe ser actualizada diariamente con los cambios realizados por el equipo de desarrollo para evitar conflictos de versiones posteriores.

2.2.1.2.4.7 Metáfora del sistema

La metáfora del sistema, es básicamente la explicación del funcionamiento del sistema. Todas las partes involucradas deben estar al tanto y comprender esta metáfora. Es extremadamente útil para la comunicación del equipo y converger ideas y conceptos.

2.2.1.2.4.8 Carga de trabajo controlada

En lo posible no se debe exceder las 40 horas de trabajo semanales. En caso de que se utilicen 5 horas a la semana para reuniones, quedarán 35 horas para el desarrollo.

2.2.2 Arquitectura de la aplicación

2.2.2.1 ¿Qué es la arquitectura de Software?

Es un diseño, modelo para construir Software. La arquitectura de Software es la estructura de más elevado nivel de un sistema de software, la disciplina de crear una estructura de alto nivel y la documentación de aquella estructura. La Arquitectura de Software es una metáfora de la arquitectura de un edificio.

2.2.2.2 ¿Para qué sirve?

La arquitectura de Software sirve para elaborar una planificación de la estructura y funcionamiento de una aplicación. De este modo prever los puntos débiles y posibles problemas aún antes de que ocurran.

2.2.2.3 ¿Qué es un arquitecto de Software?

Es un gestor informático que realiza diseños de alto nivel y se basa en estándares técnicos como por ejemplo: estándares de código, plataformas y herramientas.

Los arquitectos de software aparecieron cuando empezó a popularizarse el desarrollo de aplicaciones multicapa y por ende, la programación orientada a objetos la cual permitió el desarrollo de aplicaciones más grandes y más complejas.

Entre las principales responsabilidades encontramos:

- Escoger una manera estándar para ejecutar el desarrollo de aplicaciones
- Definir un Framework para la aplicación
- Observar y entender el amplio ambiente del sistema
- Crear el diseño de los componentes
- Tener conocimiento sobre otras aplicaciones dentro de la empresa
- Sub dividir aplicaciones complejas durante la fase de diseño a piezas más pequeñas y manejables
- Comprender las funcionalidades de cada componente en una aplicación
- Entender las interacciones y dependencias entre los componentes
- Comunicar estos conceptos a los desarrolladores

2.2.2.4 Arquitecturas más comunes

2.2.2.4.1 Monolítica

Describe una aplicación de software de una sola capa en la cual, la interfaz de usuario y el código de acceso a datos son combinados desde una simple plataforma en un simple programa, en la Figura 2-5 podemos apreciar dicha arquitectura.

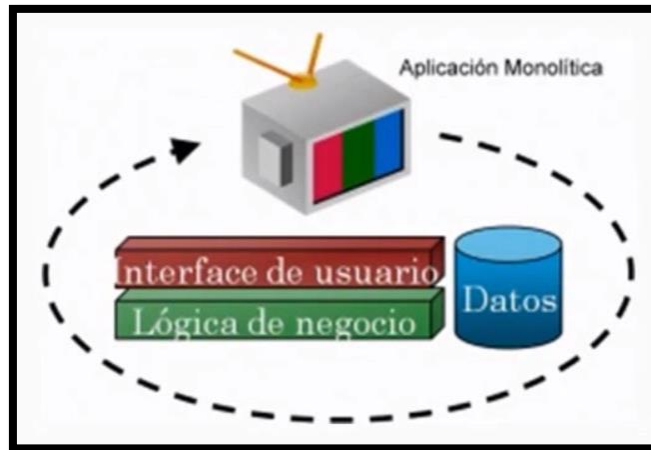


Figura 2-5: Arquitectura monolítica

2.2.2.4.2 Cliente-servidor

Es una estructura de aplicación distribuida que parte tareas o cargas de trabajo entre los proveedores de un recurso o servicio, llamados servidores y los solicitantes de servicio, llamados clientes.

Siempre, los clientes y servidores se comunican en la misma red pero en hardware separado, sin embargo, ambos pueden residir en el mismo sistema. El servidor ejecuta uno o más programas que son compartidos a los clientes. Un cliente no comparte ninguno de sus recursos pero a cambio realiza peticiones al servidor por lo que ellos inician sesiones de comunicación con los servidores, los cuales esperan por peticiones entrantes.

Como ejemplos de aplicaciones que usan este modelo tenemos: Email, impresión de red, y el World Wide Web. La Figura 2-6 describe la arquitectura de las aplicaciones cliente servidor.

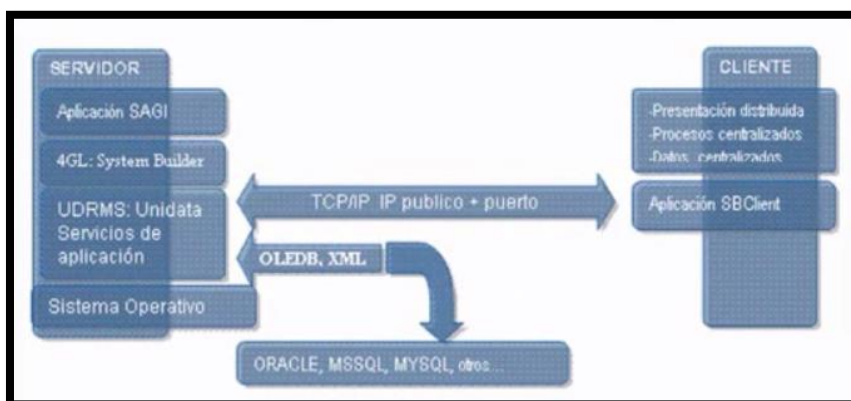


Figura 2-6: Arquitectura cliente servidor

2.2.2.4.3 MVC

“El patrón de arquitectura MVC (Modelo Vista Controlador) es un patrón que define la organización independiente del **Modelo** (Objetos de Negocio), la **Vista** (interfaz con el usuario u otro sistema) y el **Controlador** (controlador del workflow de la aplicación)” (Anónimo, ejemplosTIW, s.f.). Se ve frecuentemente en aplicaciones Web.

- **Modelo:** Notifica sus vistas asociadas y controladores cuando ha habido un cambio en su estado. Esta notificación permite a la vista producir salidas actualizadas, y los controladores pueden cambiar el grupo de comandos disponibles.
- **Vista:** Solicita información al modelo, la cual usa para generar una representación de salida para el usuario.
- **Controlador:** Puede enviar comandos al modelo para actualizar el estado del mismo, por ejemplo editando un documento. Además puede enviar comandos a su vista asociada para cambiar la presentación de la vista de un modelo, por ejemplo, cuando se utiliza el scroll por un documento.

En la Figura 2-7 se describe la arquitectura MVC.

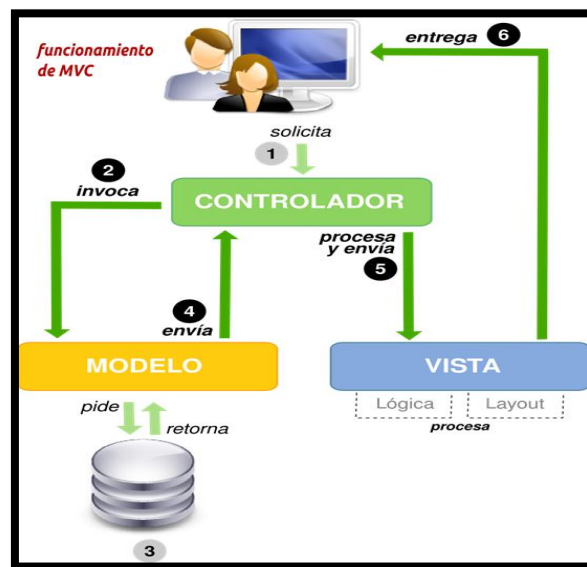


Figura 2-7: Arquitectura modelo vista controlador

2.2.2.4.4 De 3 niveles

Es una arquitectura en la cual la interfaz de usuario (presentación), lógica de procesos funcionales (reglas de negocio), almacenamiento de datos y el acceso a datos son

desarrollados y mantenidos como módulos independientes, frecuentemente sobre plataformas separadas (Cupi Veliz, 2014).

Destinada a permitir que cualquiera de las 3 capas pueda ser actualizada o reemplazada independientemente en respuesta a cambio de requerimientos o tecnología. Por ejemplo, un cambio de Sistema Operativo afectaría el código de la capa de presentación únicamente (Cupi Veliz, 2014). En la Figura 2-8 se aprecia la arquitectura de 3 niveles.

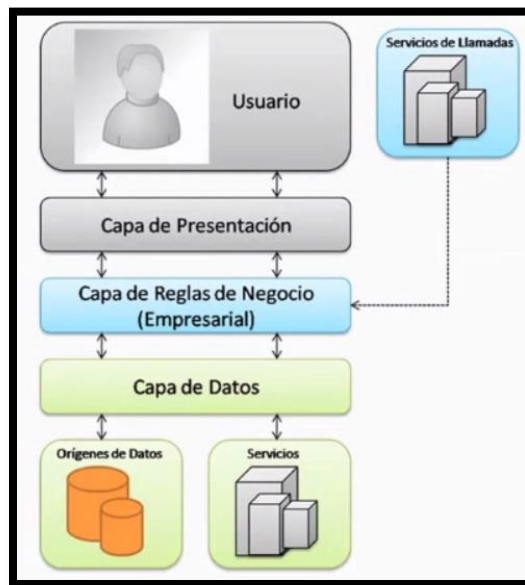


Figura 2-8: Arquitectura de 3 niveles

2.2.2.4.5 De n niveles

Arquitectura muy parecida a la de 3 niveles, con la diferencia en que las capas se pueden dividir en otras sub capas.

Esta arquitectura provee un modelo en el cual los desarrolladores pueden crear aplicaciones reusables y flexibles. Al realizar la separación en capas, los desarrolladores adquieren la opción de modificar o añadir una capa específica en vez de volver a empezar desde cero la programación de toda la aplicación. En la Figura 2-9 se evidencia la arquitectura de n niveles.

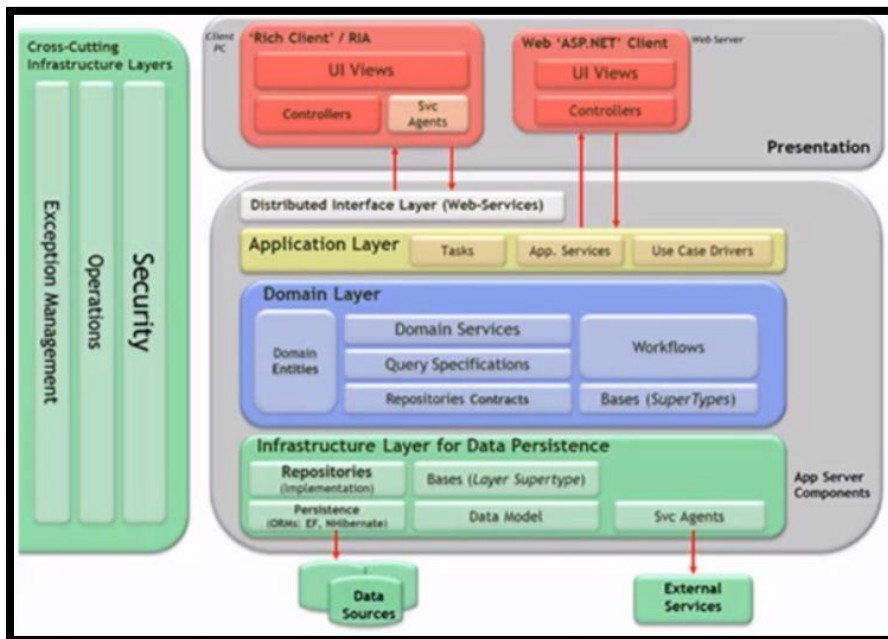


Figura 2-9: Arquitectura de n niveles

2.2.2.5 Modelos o Vistas

Es un Framework el cual define un conjunto coherente de vistas a ser usadas en la construcción de la arquitectura de un sistema, del Software de una arquitectura empresarial. Una vista es una representación de todo el sistema desde la perspectiva de un grupo de asuntos relacionados. Existen varios tipos de vistas pero las 3 fundamentales son:

- **Visión estática:** Detalla los componentes que posee la arquitectura
- **Visión funcional:** Especifica lo que realiza cada uno de los componentes
- **Visión dinámica:** Nos indica el comportamiento de cada componente en el tiempo

2.2.3 Lenguaje de Programación

Un lenguaje de programación es utilizado para escribir instrucciones interpretables por una computadora, para realizar esto, existen 2 opciones:

Compilador: Un compilador transforma un conjunto de instrucciones escritas en un lenguaje de programación de alto nivel, de código fuente a código de objeto, interpretable por una computadora (Anónimo, Compile: Webopedia, 2014).

Intérprete: Un intérprete es un programa que ejecuta instrucciones escritas en un lenguaje de programación de alto nivel. Generalmente este tipo de ejecución tarda más que el compilador (Anónimo, Interpreter: Webopedia, 2014).

La Figura 2-10, describe la estructura que define el orden de ejecución de las instrucciones de un lenguaje de programación.

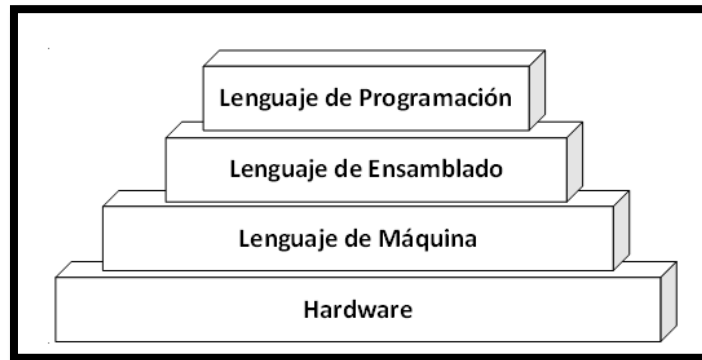


Figura 2-10: Orden de ejecución de un lenguaje de programación

2.2.3.1 Componentes

2.2.3.1.1 Sintaxis

La sintaxis de un lenguaje de programación es el conjunto de reglas que definen el significado de combinaciones específicas de caracteres y símbolos. La sintaxis puede ser textual o gráfica según la definición del lenguaje. Existen 3 niveles de sintaxis: palabras, frases y contexto.

2.2.3.1.2 Semántica

La semántica de un lenguaje de programación, se enfoca en el significado de las instrucciones del lenguaje, evaluando la estructura de expresiones sintácticamente correctas.

En la actualidad, existe una cantidad considerable de lenguajes de programación, cada uno con características, usos, fortalezas y debilidades específicas, por lo que en realidad depende del tipo de proyecto y las habilidades del programador escoger el lenguaje de programación más conveniente.

Entre los principales lenguajes de programación tenemos: Fortran, Perl, PHP, ASP, ASP.NET, Python, BASIC, Java, JavaScript, Ruby, C, C++, C#, Cobol, Delphi, Pascal.

2.2.4 Entorno de desarrollo integrado

Es un programa informático compuesto por un conjunto de herramientas de programación. También se los conoce como IDE por sus siglas en inglés (Integrated Development Environment). Es un editor de códigos que nos puede servir para depurar o para facilitar las diferentes tareas necesarias para la elaboración y desarrollo de cualquier tipo de aplicación. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

2.2.4.1 Componentes

- Un editor de texto
- Un compilador
- Un intérprete
- Un depurador
- Posibilidad de ofrecer un sistema de control de versiones
- Factibilidad para ayuda en la construcción de interfaces gráficas de usuario

Algunos IDEs ofrecen la posibilidad de programar en distintos lenguajes de programación como es el caso de Eclipse. Otros además permiten que se usen plugins para conseguir ciertas funcionalidades relacionadas a un lenguaje específico.

2.2.4.2 Características

- Diseñados para maximizar la productividad de los programadores mediante la unión de componentes con interfaz de usuario similares
- Típicamente nos proveen características como autoría, modificación, compilación y depuración.
- Utilizan teclas de función que ejecutan comandos usados con frecuencia

2.2.4.3 Ventajas

- Nos provee facilidades muy comprensivas
- Varios IDEs modernos ofrecen la posibilidad de autocompletar el código
- Reduce la necesidad de configurar y unir muchas utilidades de desarrollo
- Aviso instantáneo de errores de sintaxis generando un rápido aprendizaje del lenguaje y librerías asociadas
- Tienen herramientas que nos permiten refactorizar el código
- Posibilidad de ejecutar una aplicación Web dentro de un navegador nativo del IDE

El IDE es muy útil para la creación de programas, ya que cuenta con varios componentes para su fácil uso en diferentes entornos. Además es importante para el desarrollo de las organizaciones (García, 2013).

2.2.5 Versionador de Código

Versionamiento de código es el control de cambios realizados a documentos, software, sitios web, etc. Su principal utilidad se evidencia en equipos colaborativos de dos o más personas. Un Versionador de código es un repositorio donde se puede almacenar tanto los archivos de código fuente de un producto como la documentación del mismo, y al que tienen acceso todos los miembros del equipo de trabajo desde su computadora (Rouse, 2007).

Un Versionador de código es un software que puede ser levantado sobre un ambiente local, sobre una red privada o incluso sobre el Internet, lo que permite tener equipos de desarrollo que se encuentren geográficamente alejados, aumentando la productividad considerablemente.

Antes de la existencia de los versionadores de código, el trabajar en equipo era extremadamente complicado y restrictivo, haciendo que la integración de componentes represente una inversión importante tanto de esfuerzo como de tiempo, inevitablemente impactando en el costo del producto.

2.2.5.1 Conceptos en el Versionamiento

2.2.5.1.1 Repositorio

Lugar donde se almacenan los archivos y que centraliza el código sobre el cual el equipo de desarrollo trabajará.

2.2.5.1.2 Check Out

Es el proceso de obtener una copia de la versión que existe en el repositorio en la máquina local.

2.2.5.1.3 Check In

Actualiza los cambios realizados en la copia local, en el repositorio.

2.2.5.1.4 Revisión

Identificador de cada cambio que se realiza en el repositorio. Se recomienda incluir comentarios en cada revisión a fin de identificar claramente cada una de ellas.

2.2.5.1.5 Ramificar

Se crea una copia exacta paralela del repositorio.

2.2.5.1.6 Conflicto

Un conflicto sucede cuando el versionador de código encuentra varios cambios pendientes sobre una misma línea de código.

2.2.5.1.7 Resolver

Intervención manual de un usuario para solucionar un conflicto.

2.2.5.1.8 Fusión

Combina varios cambios en un mismo archivo.

2.3 Herramientas

2.3.1 Visual Studio 2010

Visual Studio es una potente solución de la gestión del ciclo de vida de las aplicaciones que permite a las organizaciones la entrega continua de valor con velocidad y calidad. Es una herramienta de desarrollo de Microsoft.

Este entorno integrado de desarrollo, ampliado y rico en funcionalidades, mejora la capacidad de entregar soluciones de alta calidad que exceden las expectativas de los clientes repetidamente.

Esta herramienta se integra con otras herramientas. Puede ser accedida por Office, Web Access, Share Point y se puede además integrar con otros entornos como Eclipse, Team Explorer Everywhere, Project Server, System Center, Windows Azure, SQL Server (Danysoft, 2012).

Dado que el mundo está cambiando, tanto empleados como clientes demandan herramientas que sean tan fáciles de usar como las aplicaciones que se descargan.

Las aplicaciones ahora deben satisfacer cinco requisitos: personales y contextuales, siempre conectadas, siempre accesibles, con un uso intensivo de los datos, siempre evolucionando. Aparte, deben seguir las tendencias tecnológicas actuales tales como: dispositivos conectados, servicios continuos, datos grandes, infraestructuras híbridas.

Los desarrolladores están buscando tres habilidades para ayudarles a entregar valor continuo:

- **Velocidad:** Acelerar la entrega de nuevas funcionalidades
- **Calidad:** Asegurar que la calidad está integrada
- **DevOps:** Integración de equipos de Dev (Desarrollos) y Ops (Operaciones)

Visual Studio 2010 proporciona estas capacidades para acelerar el desarrollo y entregar valor continuo

2.3.1.1 Licenciamiento

Visual Studio 2010 tiene varias licencias según la necesidad del usuario, a continuación se describen las principales características de cada una de ellas:

2.3.1.1.1 Professional

La licencia Professional entre sus principales beneficios permite el desarrollo de aplicaciones para Windows Store, así como la integración con plataformas de Microsoft como son Sharepoint y Office.

2.3.1.1.2 Test Professional

Como su nombre lo indica, la licencia Test Professional se enfoca en las pruebas que se deben realizar al código y software desarrollado. Proporciona un diagnóstico avanzado de datos que permite a los equipos de desarrollo tener más visibilidad de la calidad de su código. La licencia Test Professional permite la gestión de casos de pruebas, pruebas exploratorias e incluye el Gestor de Retroalimentación. Una herramienta extremadamente útil para el intercambio de información entre el equipo de testeo y el equipo de desarrollo.

2.3.1.1.3 Premium

La licencia Premium provee de métricas de código que facilitan el cumplimiento de estándares definidos por la organización o el equipo de desarrollo. Otra funcionalidad importante es la capacidad para realizar pruebas de interfaz codificadas, lo cual elimina la necesidad de tener una persona dedicada a estas pruebas.

2.3.1.1.4 Ultimate

La licencia Ultimate es la más completa de Visual Studio, incluye la característica IntelliTrace Anywhere, que permite depurar aplicaciones a nivel de producción. Otra funcionalidad importante que incluye es Microsoft Fakes, que permite simular objetos necesarios para las pruebas unitarias de código.

2.3.2 Internet Information Server (IIS)

Internet Information Server, es un servidor web² diseñado para sistemas operativos Windows. Si bien en un inicio IIS fue diseñado para Windows NT y Windows Server, con el transcurrir del tiempo se ha ido incorporando a los sistemas operativos Windows para computadoras de escritorio como Windows XP, Windows 7 y Windows 8.

Con el pasar de los años, Internet Information Server ha sufrido una importante evolución para mantenerse a la par de los vertiginosos avances del mercado. Actualmente, la última versión de Internet Information Server es 8.5 y viene incluido en Windows Server 2012 y Windows 8.1.

En la actualidad, Internet Information Server es el segundo servidor web más usado en el mundo, únicamente por detrás de Apache HTTP Server (Anónimo, Overview: IIS, 2014).

2.3.2.1 Características

Entre las principales características de Internet Information Server tenemos (Dodge, Lehto, & Weiner, 2008):

- Distintos tipos de autenticación (Anónima, Básica, Windows, Certificados, etc.).
- Filtro de requerimientos.
- Autorización de URLs.
- Módulos de compresión.
- Módulos de logging y diagnóstico.
- Soporte para centralización de certificados SSL.
- Soporte para protocolo WebSocket.

2.3.2.2 Arquitectura

En la Figura 2-11 se describe la arquitectura del Internet Information Server.

² Un servidor web es un software que ejecuta una petición de un cliente y retorna una respuesta que será interpretada por el cliente, permitiendo la comunicación y navegación.

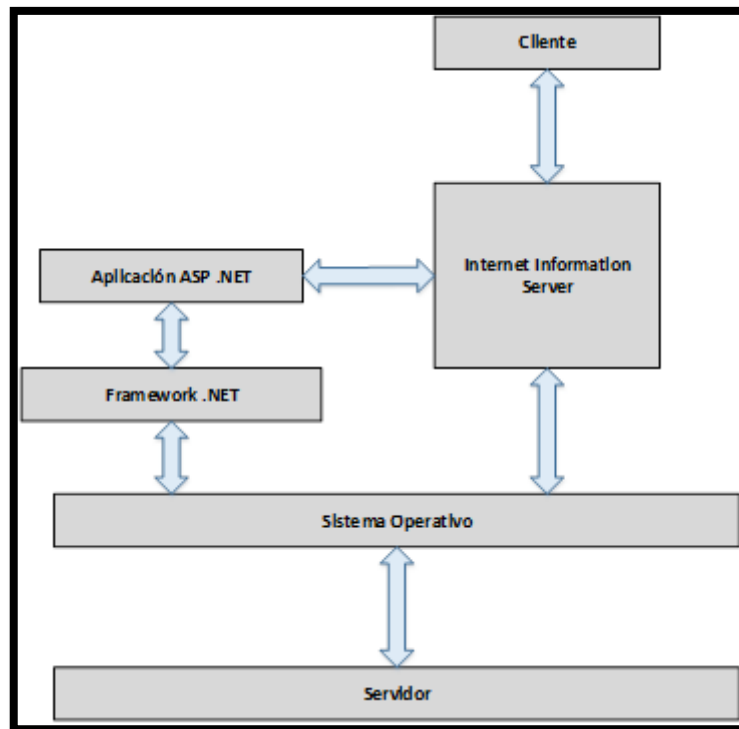


Figura 2-11: Arquitectura IIS

2.3.3 Gestor de base de datos DB2

DB2 es un motor de base de datos relacional con licencia comercial. Todo su manejo de la estructura de la base de datos es de tipo jerárquico. Almacena documentos completos y pertenece a la casa desarrolladora de IBM.

Puede hacer que sus documentos se conviertan en formato XML lo cual hace que su búsqueda relacional sea más rápida de una forma jerárquica reduciendo las necesidades de consumo.

Se puede trabajar sobre las plataformas de Linux, Unix y Windows, además permite la automatización de tareas por completo. Ahorra muchos procesos para los usuarios y los desarrolladores de los productos.

La compresión da soporte a los logros en rendimiento, esto tiene que ver con lo detallado sobre archivos XML. DB2 integra XML de manera nativa, lo que IBM ha denominado pureXML (Alice, 2009).

2.3.3.1 Características

- Permite el manejo de objetos grandes (hasta 2 GB)
- Definición de datos y funciones por parte del usuario
- Chequeo de integridad referencial

- SQL recursivo
- Soporte multimedia: texto, imágenes, video, audio
- Permite la ejecución de queries paralelos, commit de dos fases, backup/recuperación on-line y off-line
- Cuenta con un motor gráfico de alto rendimiento el cual posibilita observar el tiempo de ejecución de una sentencia SQL y corregir detalles para aumentar el rendimiento

2.3.3.2 Ventajas

- Permite agilizar el tiempo de respuesta de las consultas
- Recuperación utilizando accesos de sólo índices
- Predicados correlacionados
- Tablas de Resumen
- Tablas replicadas
- Uniones hash

2.3.3.3 Compatibilidad

Las plataformas que admite el producto son:

- OS/2
- Windows
- Unix
- HP-UX
- AIX
- Solaris
- Linux

2.3.4 Aqua Data Studio

AquaFold es una empresa desarrolladora de software, enfocada en soluciones de bases de datos, herramientas colaborativas y de análisis. Entre sus principales productos están:

- Aqua Data Studio
- Aqua Data Server
- AquaClusters

Aqua Data Studio es un entorno de desarrollo multiplataforma para diseño, administración y consultas de bases de datos. Se integra con los principales motores de base de datos como:

- DB2
- MySQL
- Oracle
- PostgreSQL
- SQLite
- SQL Server

Una de las principales herramientas de Aqua Data Studio es el Analizador de Consultas, que permite al usuario ejecutar comandos SQL ayudado por funcionalidades extra como:

- Autocompletar
- Delinear sintaxis
- Comentarios de código

Otro de los principales beneficios de Aqua Data Studio es su capacidad para exportar los resultados de una ejecución SQL a una variedad significativa de tipos de archivo, según la necesidad del usuario.

Aqua Data Studio también permite a sus usuarios automatizar las tareas más comunes y repetitivas que se realizan día a día, lo cual reduce significativamente el tiempo requerido para la creación de scripts, convirtiéndolo en una herramienta importante para el desarrollo de software.

La versión 1.0 de Aqua Data Studio fue liberada en el año 2002, a partir de ahí, ha sido una tendencia casi absoluta la liberación anual de nuevas versiones, incluso existiendo varias versiones en un mismo año, lo que demuestra el compromiso de AquaFold con la innovación. Actualmente la versión de Aqua Data Studio es la 14.0 liberada en el 2013 (Anónimo, Overview: Aquafold, 2014).

2.3.5 Herramienta de Modelamiento PowerDesigner

Es una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computador) líder en modelamiento que permite a las empresas visualizar, analizar y manipular metadatos basado en modelos. La herramienta fue producida por Sybase y está destinada a aumentar la productividad en el desarrollo de

Software, reduciendo el costo de las mismas en términos de tiempo y de dinero (Anónimo, Herramienta CASE: Wikipedia, 2014).

PowerDesigner se ejecuta sobre Microsoft Windows como una aplicación nativa y se ejecuta sobre Eclipse como un plugin (Anónimo, PowerDesigner: Wikipedia, 2014).

Durante todo el ciclo de vida del desarrollo de Software, esta herramienta nos facilitará las tareas, al permitirnos diseñar los distintos modelos.

Brinda también potentes técnicas de análisis, diseño y gestión. Combina varias técnicas de modelamiento y estándares:

- Modelamiento de aplicación a través de UML
- Técnicas de modelamiento de procesos empresariales
- Técnicas tradicionales de modelamiento en bases de datos.

2.3.5.1 Características generales

Pre visualización de comparación de modelos. El modelo orientado a objetos puede ser un Diagrama UML, Diagrama de Clase, Diagrama de caso de uso, Diagrama de secuencia, Diagrama de actividades o Diagrama de Componentes.

PowerDesigner es utilizado principalmente en los sistemas de información como los modelos de procesos del negocio.

2.3.5.2 Objetivos de la herramienta CASE

- Mejorar la productividad en el desarrollo y el mantenimiento del Software
- Aumentar la calidad del Software
- Reducir el tiempo y coste de desarrollo y mantenimiento de los Sistemas Informáticos
- Mejorar la planificación de un proyecto
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos
- La automatización de las actividades involucradas en el ciclo de vida de desarrollo de software como documentación, generación de código, etc.
- Gestión global en todas las fases de desarrollo de Software con la misma herramienta
- Emplear de manera más fácil las metodologías de desarrollo de Software.

- La generación automática de código a partir del diseño.

2.3.6 Team Foundation Server

Es un producto Microsoft creado para versionar código fuente, sin embargo ha ido evolucionando en la última década para convertirse en una herramienta que facilita la cooperación de equipos enteros para alcanzar la entrega de un producto o proyecto (Anónimo, Team Foundation Server: Visual Studio, 2014). Los principales servicios que brinda son:

- Versionamiento de código.
- Reportes.
- Manejo de requerimientos.
- Manejo de proyectos.
- Compilaciones automatizadas.
- Manejo de pruebas y publicaciones.

2.3.6.1 Capas de Team Foundation Server

Team Foundation Server está compuesto por las siguientes capas:

2.3.6.1.1 Capa de Aplicación

La capa de aplicación consiste principalmente de un conjunto de servicios web con los que la máquina del cliente se comunica utilizando un protocolo optimizado para este fin.

2.3.6.1.2 Capa de Datos

La capa de datos está compuesto por dos o más bases de datos SQL Server, que contienen la lógica que soporta Team Foundation Server.

2.3.6.2 Componentes de Team Foundation Server

Team Foundation Server tiene los siguientes componentes:

2.3.6.2.1 Servidor

Es la instancia de Team Foundation Server sobre la que se trabajará, puede instalarse en varias máquinas según la necesidad. Team Foundation Server puede soportar un gran número de usuarios concurrentes siempre y cuando el hardware del servidor lo permita.

2.3.6.2.2 Colección de Proyectos de Equipo

Es un contenedor de uno o más proyectos de equipos, es el principal nivel de aislamiento entre instancias en un servidor. Una colección de proyectos de equipo tiene una relación uno a uno con la instancia de la base de datos, por lo que se puede respaldar y restaurar a nivel de colección de proyectos.

2.3.6.2.3 Proyecto de Equipo

Es una colección de código, pruebas o compilaciones que agrupan todas las herramientas usadas en el ciclo de vida de un proyecto de desarrollo de software. La intención es que los proyectos de equipo representen la más larga unidad de trabajo en una organización o empresa.

2.3.6.2.4 Plantilla de Proceso

Un hecho conocido respecto a los proyectos de desarrollo de software, es que no existe un único proceso que se ajuste para todo tipo de proyectos y/o todo tipo de equipos. Por esta razón, Team Foundation Server fue diseñado para ser flexible y permitir a sus usuarios ajustar el proceso según sus necesidades.

2.3.6.2.5 Seguimiento de Trabajo

Los ítems de trabajo en Team Foundation Server incluyen requerimientos, bugs, problemas, y casos de prueba. El seguimiento de trabajo consiste en permitir al usuario definir qué ítems de trabajo seguir de una manera más cercana o con mayor prioridad, ayudando a mantener la visión del proyecto.

2.3.6.2.6 Versionamiento

Team Foundation Server incluye un sistema de versionamiento centralizado, que permite realizar protección de archivos atómica, ramificación, fusión, etc. (Anónimo, TFS: Microsoft, 2014). En la Figura 2-12 se puede apreciar la estructura de Team Foundation Server.

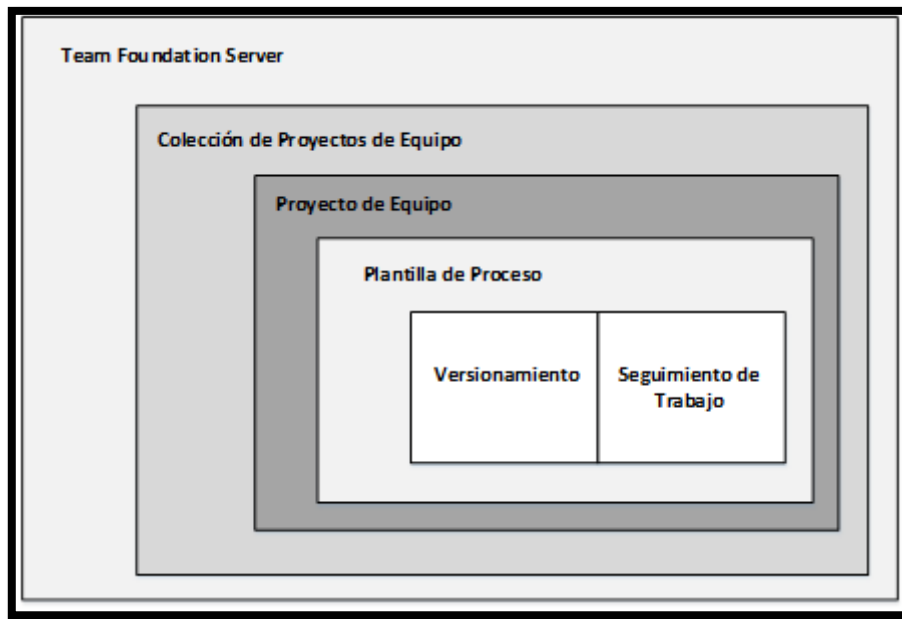


Figura 2-12: Estructura de Team Foundation Server

3 SELECCIÓN DE ELEMENTOS PARA EL PROYECTO

En este capítulo se describirán los elementos seleccionados para el desarrollo del proyecto, basados en la teoría expuesta en el capítulo 2.

3.1 Scrum y eXtreme Programming

Existe una gama considerable de metodologías para el desarrollo de software, cada una con características definidas por un enfoque particular. Una de las principales y más famosas metodologías usadas es el desarrollo en cascada³; el principal beneficio de este modelo es su relativamente corta y fácil curva de aprendizaje, sin embargo esto no siempre se traduce en facilidad de implementación de un proyecto. Uno de los principales problemas del desarrollo en cascada es su rigidez y poca adaptabilidad a cambios que se presentan durante el proyecto.

A medida que se iba evidenciando más la necesidad de modelos adaptativos y flexibles, surgió una tendencia de metodologías ágiles enfocadas en tiempos cortos de respuesta ante cambios, que permitieron afrontar de mejor manera las necesidades del negocio en el desarrollo de productos y servicios.

Un gran beneficio, es poder tomar los elementos que mejor se adapten al proyecto de una o más metodologías de desarrollo, que gracias a su estructura flexible, permiten combinarlos obteniendo un modelo altamente específico y adecuado para el equipo de desarrollo.

Una de las combinaciones de metodologías más comunes, es adaptar los procesos de Scrum, con prácticas y/o técnicas de Extreme Programming:

3.1.1 *Procesos de Scrum*

A continuación se enumeran los procesos seleccionados de Scrum para combinarlos con Extreme Programming, los mismos fueron descritos en la sección 2.2.1.1:

- Product Backlog
- Sprint
- Sprint Backlog
- Estimación del esfuerzo
- Reuniones diarias

³ Creada por Winston W. Royce en 1970.

3.1.2 Prácticas de eXtreme Programming

A continuación se enumeran las prácticas seleccionadas de Extreme Programming para combinarlas con Scrum, las mismas fueron descritas en la sección 2.2.1.2:

- Diseño simple
- Pruebas unitarias
- Refactorización continua del código
- Integración continua
- Programación en pares

3.2 Web Forms

Los ASP.Net Web Forms, los cuales fueron liberados por primera vez en el año 2002, están atravesando por su séptima edición. Ha sido la principal tecnología de desarrollo web usada sobre la plataforma de Microsoft por más de diez años. Los Formularios Web utilizan un modelo de programación que es muy similar a la programación de formularios “Windows Forms”.

Siguen un modelo donde un desarrollador puede diseñar una pantalla mediante el arrastre de controles como cuadros de texto y listas desplegables al área de diseño, para que después, mediante la ejecución de un doble clic al control, se pueda crear un manejador de eventos en una página que contiene el código por detrás. Por ejemplo, al hacer doble clic en un botón nos creará en evento “OnClick” donde podremos poner nuestro código que será ejecutado cuando el botón sea presionado.

Los principales beneficios de los Web Forms es que fueron fáciles de aprender por los programadores de Visual Basic quienes poco a poco se alejaron de la programación de clientes pesados para empezar a construir aplicaciones web. Los Web Forms ahorraron tiempo a los desarrolladores al utilizar características como controles de validación de formularios y seguridad de los sitios web (Santillana, 2008).

La desventaja que podemos observar hoy por hoy es que los Web Forms fueron diseñados para la web en el año 2002, cuando era aceptable para la manipulación de cada interfaz de usuario, el envío de un postback al servidor. Esto se arregló en el 2008 con la liberación de ASP.NET Ajax.

Los Web Forms pueden ser ideales todavía para equipos que necesitan poner en conjunto aplicaciones pequeñas que no necesitan una interfaz de usuario altamente sofisticada.

A continuación presentamos una visión rápida de una serie de conceptos clave en el desarrollo web tales como: HTTP, HTML, Scripts del lado del cliente y Postbacks.

En la Figura 3-1 se detalla el flujo de navegación web en un servidor IIS.

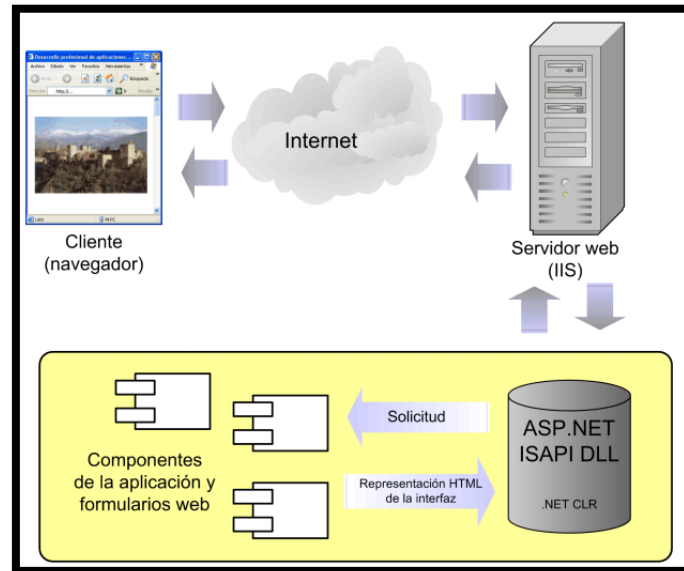


Figura 3-1: Ejecución de páginas ASP.NET usando el Internet Information Server como servidor HTTP, la librería ISAPI permite aprovechar los recursos de la plataforma .NET en el desarrollo de aplicaciones Web

3.2.1 El rol de HTTP

Las aplicaciones Web son muy diferentes a las aplicaciones de escritorio. La primera diferencia obvia es que una aplicación Web en un ambiente de producción involucra por lo menos dos máquinas interconectadas en la misma red, la una es la que hace de host para el sitio web y la otra es la que ve la información a través de un navegador web. Por supuesto es posible que durante el desarrollo se pueda tener una única máquina que tenga los dos roles, el cliente y el servidor. Dada la naturaleza de las aplicaciones Web, las máquinas conectadas en red en cuestión deben acordar un protocolo particular para determinar cómo enviar y recibir los datos. Este protocolo que conecta los computadores es el Protocolo de Transferencia de Hipertexto (HTTP).

3.2.2 El rol de HTML

Una aplicación Web es simplemente un conjunto de archivos que constituyen la funcionalidad del sitio. Muchos de estos archivos contendrán sentencias HTML. HTML es un lenguaje estándar de marcado usado para describir como un texto literal, imágenes, links externos y varios controles HTML van a ser renderizados en el navegador del cliente (Payne, 2002).

En estos días los IDEs modernos (incluyendo Visual Studio) y las plataformas de desarrollo web (como ASP.NET) generan automáticamente el código HTML por lo que a medida que se trabaje con ASP.NET se tendrá un conocimiento práctico de HTML.

3.2.3 El rol del Script en el lado del cliente

En adición a los elementos de la interfaz de usuarios, un archivo de tipo *.htm puede contener bloques de código de Script que serán procesados por el navegador que realiza la petición. A continuación se indican las dos principales razones por la que los Scripts son utilizados:

- Validación de ingreso de datos de los usuarios en el navegador antes de hacer postback al servidor web.
- Interacción con el Modelo de Objetos del Documento del navegador (DOM).

Con respecto al primer punto, debemos comprender que el mal inherente de una aplicación Web es la necesidad de hacer postbacks frecuentes al servidor para actualizar el código HTML que será renderizado en el navegador. Mientras los postbacks sean inevitables, debemos estar siempre conscientes de las maneras de minimizar el tráfico en la red. Una técnica que evita los postbacks es el uso de Scripts para validar el ingreso de datos de los usuarios antes de entregar los datos del formulario al servidor Web. Si se encuentra un error, como por ejemplo al no proveer datos en un campo requerido, podemos alertar el error al usuario sin incurrir en el costo de hacer un postback al servidor. Después de todo, nada es más molesto para los usuarios que hacer un postback en una conexión lenta, solo para recibir un mensaje de error.

Debemos ser conscientes que aun cuando hagamos una validación del lado del cliente (por conseguir una mejor respuesta de tiempo), la validación debería también darse el lado del servidor. Esto nos permitiría asegurarnos que los datos que circulan a través de la red no han sido manipulados. Los controles de validación de ASP.NET ejecutarán automáticamente validaciones por el lado del cliente y por el lado del servidor.

Con respecto al segundo punto, los Scripts del lado del cliente pueden además ser utilizados para interactuar con el DOM del navegador. La mayoría de navegadores comerciales exponen un conjunto de objetos que pueden ser apalancados para controlar cómo debe comportarse el navegador.

Cuando un navegador analiza una página HTML, construye un objeto de tipo árbol en la memoria, representando todos los contenidos de la página Web (forms, controles de ingreso, etc.). Los navegadores proveen una API llamada DOM que expone el árbol y nos permite modificar sus contenidos programáticamente. Por ejemplo, podemos escribir código JavaScript que se ejecuta en el navegador para obtener los valores de controles específicos, cambiar el color de un control, añadir dinámicamente nuevos controles a la página entre otras cosas (Creación de Sitios Web y su utilidad didáctica, 2014).

ASP.NET provee la propiedad `HttpRequest.Browser`, la cual permite determinar en tiempo de ejecución las capacidades del navegador y el dispositivo que envió la petición. Podemos utilizar esa información para estilizar como generar de regreso la respuesta HTTP de la manera más óptima. De todas maneras, raramente tendremos que preocuparnos por esto en casos de que estemos implementando controles personalizados, porque todos los controles Web estándares en ASP.NET conocen de manera automática como deben renderizarse apropiadamente ellos mismos basados en el tipo de navegador. Esta notable capacidad es conocida como “Renderización Adaptativa”, y es implementada para todos los controles estándar de ASP.NET.

Hay varios lenguajes de Script que pueden ser utilizados pero el más popular y por mucho es JavaScript. Es importante notar que de ninguna manera JavaScript es lo mismo que el lenguaje de programación Java. Mientras JavaScript y Java tienen alguna sintaxis similar, JavaScript es menos poderoso que Java. La buena noticia es que todos los navegadores modernos soportan JavaScript, el cual se convierte en un candidato natural para ejecutar la lógica de un Script en el lado del cliente.

3.2.4 Rol de los Postback al servidor

Los HTML simples ejecutan toda su funcionalidad dentro del navegador. Una página real necesita hacer postback a los recursos en el servidor Web, enviando toda la información de entrada al mismo tiempo. Una vez que el recurso del lado del servidor recibe los datos, el recurso puede usar esta información para construir una correcta respuesta HTTP generada dinámicamente (Tuttini, 2009).

El atributo “action” en la etiqueta de apertura “<form>” especifica el recipiente de los datos de entrada del formulario. Los posibles recipientes incluyen los servidores de correo, otros archivos HTML en el servidor web, un clásico archivo ASP (Active Server Page) y una página Web ASP.NET.

Más allá del atributo “action”, nos gustaría tener un botón “submit”, el cual, cuando sea dado clic, transmitirá los datos del formulario a la aplicación web vía una petición HTTP. En la Figura 3-2 podemos ver una actualización de un archivo, especificando el atributo “action” en la etiqueta “<form>” (Creación de Sitios Web y su utilidad didáctica, 2014)

```
<form id="defaultPage"
      action="http://localhost/Cars/ClassicAspPage.asp" method="GET">
  <input id="btnPostBack" type="submit" value="Post to Server!" />
  ...
</form>
```

Figura 3-2: Uso del atributo "action" en la etiqueta <form>

Cuando se da clic en el botón “submit” del formulario, los datos son enviados a una página ASP clásica en la URL especificada. Cuando especificamos el método “GET” como el modo de transmisión, los datos son apprehendidos a una cadena como un conjunto de parejas que tienen nombre/valor y que son separadas por ampersands (&). Alguna vez hemos de haber visto estos datos en nuestros navegadores como se puede ver en la Figura 3-3.

```
http://www.google.com/search?hl=en&source=hp&q=vikings&cts=1264370773666&aq=
f&aql=&aqi=g1g-z1g1g-z1g1g-z1g4&oq=
```

Figura 3-3: Datos transmitidos mediante el método "GET"

La otra opción de transmisión de datos al servidor web es haciendo uso del método “POST” como se puede ver en la Figura 3-4.

```
<form id="defaultPage"
      action="http://localhost/Cars/ClassicAspPage.asp" method = "POST">
  ...
</form>
```

Figura 3-4: Uso del método POST

En este caso, los datos del formulario no son adjuntados a la cadena. Al hacer uso de POST, los datos del formulario no son visibles directamente para el mundo exterior. Pero más importante los datos POST no tienen una limitación de longitud de caracteres, como lo tienen muchos navegadores que limitan las cadenas GET.

3.2.4.1 Postbacks de acuerdo a ASP.NET

Cuando estamos construyendo sitios Web usando Web Forms basados en ASP.NET, el framework tomará precaución de los mecanismos de posteo en beneficio nuestro. Uno de los varios beneficios de construir un sitio Web usando ASP.NET es que las capas del modelo de programación están en la cumbre del protocolo estándar de petición/respuesta HTTP de un sistema manejado por eventos. En consecuencia, en vez de configurar manualmente un atributo “action” y definir un botón HTML de tipo “submit”, podremos simplemente manejar eventos en los controles Web de ASP.NET usando la sintaxis estándar de C#.

Al utilizar este modelo manejado por eventos, podremos realizar postbacks al servidor Web muy fácilmente usando un gran número de controles. Si requerimos podremos hacer postbacks al servidor Web si el usuario da clic en un radio button, un ítem en una lista, una fecha en un calendario, etc. En cada caso, simplemente manejamos el evento correcto, y el tiempo de ejecución de ASP.NET emitirá automáticamente los datos HTML posteados.

3.3 Web Services

Las APIs actuales permiten el acceso a distintas funcionalidades provistas. Cuando necesitamos exponer los servicios de objetos remotos hacia cualquier sistema operativo y cualquier modelo de programación, los servicios Web proporcionan una sencilla manera de realizar esto (Payne, 2002).

A diferencia de las aplicaciones Web basadas en el uso de navegadores tradicionales, un servicio Web provee una manera de exponer las funcionalidades de componentes remotos haciendo uso de protocolos web estándares. Desde el estreno de .NET, los programadores han sido provistos de un soporte elevado para construir y consumir servicios Web XML con el namespace System.Web.Services. En muchos casos, construir un servicio Web completo no es más complicado que aplicar el atributo “WebMethod” a cada método público al que deseamos proveer acceso. Además, Visual Studio nos permite conectar a un servicio Web remoto haciendo clic en uno o dos botones.

Los servicios Web facilitan a los desarrolladores a construir ensamblados .NET conteniendo tipos que pueden ser accedidos usando el HTTP simple. A parte, un servicio Web codifica sus datos como un simple XML. Dado el hecho de que un servicio Web está basado en estándares abiertos de la industria (por ejemplo: HTTP, XML y SOAP) en

vez de sistemas propietarios y formatos propietarios cableados (como es el caso de DCOM o .NET remoting), ellos nos permiten un alto grado de interoperabilidad e intercambio de datos. En la Figura 3-5 se aprecia la interoperabilidad entre las aplicaciones.

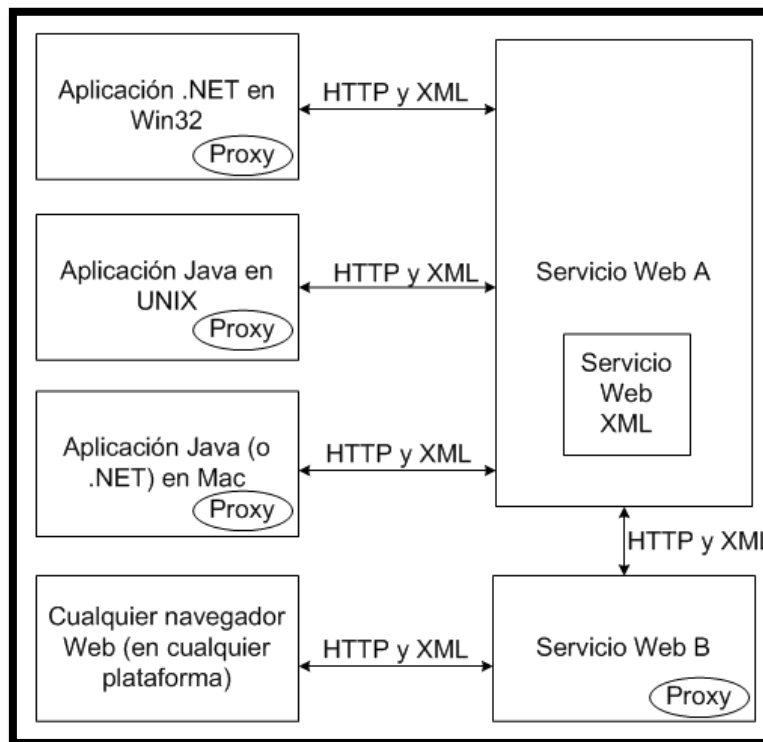


Figura 3-5: Los servicios Web XML permiten un alto grado de interoperabilidad

Claro que no todo es perfecto. Un inconveniente de los servicios Web es que puede padecer de problemas de rendimiento. Otro inconveniente es que no son una solución ideal para aplicaciones “in house”.

3.3.1 Estándares de los Servicios Web

Otro problema que los servicios Web enfrentaban antes fue que todos los involucrados de las grandes industrias (Microsoft, IBM, Sun Microsystems) crearon implementaciones de servicios Web que no fueron cien por ciento compatibles con otras implementaciones de servicios Web. Obviamente, esto fue un problema porque la gran meta de los servicios Web es lograr un alto grado de interoperabilidad entre las distintas plataformas y sistemas operativos (Web Services Standards, 2014).

Para asegurarse de la interoperabilidad de los servicios Web, grupos como el “World Wide Web Consortium” (W3C: www.w3.org) y la “Web Services Interoperability Organization” (WS-I: www.ws-i.org) empezaron a crear varias especificaciones que han

formulado cómo un vendedor de Software (IBM, Microsoft, etc.) debe construir librerías de Software de servicios Web para garantizar la compatibilidad.

Colectivamente, todas estas especificaciones son dadas el nombre genérico WS-*, y ellos cubren tales problemas como la seguridad, acoplamiento, la descripción de los servicios Web (usando el lenguaje de descripción de los servicios Web o WSDL), políticas, formatos SOAP (Simple Object Access Protocol) y una gran cantidad de otros detalles importantes (Besteiro & Rodríguez, 2014).

3.4 Visual C# 2010

En enero de 1999, Anders Hejlsberg⁴, Scott Wiltamuth y Peter Golde empezaron a construir un nuevo lenguaje orientado a objetos a partir de C, el resultado fue C#, cuya primera versión fue liberada en enero del 2002.

El principal propósito de C# es ser un lenguaje orientado a objetos moderno, simple y adaptativo.

3.4.1 Versiones

A lo largo de su historia, C# ha ido evolucionando notablemente, sus versiones son:

Versión	Fecha	.NET Framework
1.0	Enero 2002	1.0
1.2	Abril 2003	1.1
2.0	Noviembre 2005	2.0
3.0	Noviembre 2007	3.5
4.0	Abril 2010	4.0
5.0	Agosto 2012	4.5

Tabla 3-1: Versiones de C#

3.4.2 Características

Las principales características de C# son:

- Orientado a objetos
- Simple

⁴ Anders Hejlsberg fue pilar fundamental en el diseño de Turbo Pascal, Embarcadero Delphi y Visual J++

- Soporte de variables fuertemente tipadas
- Control de variables no inicializadas
- Control de límites en estructuras de datos
- Recolección automática de basura
- Robustez
- Seguridad

3.4.3 Tipos de proyecto

C# permite crear los siguientes tipos de proyectos:

- Aplicaciones de Consola

Aplicaciones sin interfaz gráfica, únicamente de texto. Se ejecutan en una ventana de DOS. Principalmente se usan para pruebas unitarias y demonios⁵.

- Aplicaciones ASP.NET

Aplicaciones web, que integran distintas tecnologías como HTML y Javascript. Existen 2 tipos de proyectos ASP.NET: Web Forms y MVC.

- Windows Forms

Aplicaciones de escritorio de Windows, consiste en clases que despliegan controles nativos de Windows.

- Windows Presentation Foundation (WPF)

Windows Presentation Foundation, es una tecnología relativamente nueva para crear aplicaciones de escritorio de Windows. WPF usa XAML⁶ para construir sus aplicaciones.

- Servicios Windows

Un servicio Windows es un programa diseñado para ejecutarse en segundo plano en los sistemas operativos basados en el kernel de Windows NT.

- Windows Communication Foundation (WCF)

⁵ Un demonio es un proceso que se ejecuta sin necesidad de que exista interacción del usuario.

⁶ XAML, acrónimo de eXtensible Application Markup Language, es una declaración XML utilizada para crear un formulario que representa todos los aspectos visuales y comportamiento de la aplicación.

Es una tecnología con importantes funcionalidades que brindan una importante gama de opciones de comunicación. WCF permite construir el servicio de comunicación una única vez, y exponerlo de varias maneras, incluso bajo distintos protocolos, tan solo modificando el archivo de configuración.

4 APLICACIÓN DE UN SISTEMA DE CÓDIGO DE BARRAS AL CASO DE ESTUDIO “TAME CARGO”

En este capítulo, se describe la motivación de realización del proyecto, así como las condiciones sobre las que se debe desarrollar.

4.1 Antecedentes

Uno de los proyectos planificados a corto plazo por la Gerencia Comercial para el 2010 fue la implementación de procesos dependientes con código de barras para Tame Cargo en Quito, Guayaquil y Cuenca. La unidad responsable de este proyecto es la Gerencia Comercial a través de la Jefatura de Carga Sistema. Las generalidades sobre el plan de acción, objetivo funcional, objetivo institucional y la justificación del proyecto se resumen en la Matriz de Proyecto o Acción a Corto Plazo de la Gerencia Comercial, la misma que fue aprobada en el 2009 y cuyo presupuesto fue incrementado en el 2010. Las autoridades de la Empresa, se encontraban conscientes de que la inversión en tecnología optimizaría los recursos materiales y humanos de la empresa, así como también mejoraría la calidad de servicio a nuestros clientes.

Mediante el oficio No. EY-I5-10-0041 del 18 de febrero del 2010, el Jefe de Carga Sistema solicita a la Gerencia de Tecnologías de Información la asignación de recursos técnicos para iniciar el desarrollo del Proyecto.

Actualmente, podemos encontrarnos con un proyecto dependiente de recurso humano cuyo horario laboral debe ajustarse a la demanda de los clientes para garantizarles el servicio, esta condición sumada a la naturaleza del ser humano, deriva en un servicio ineficiente por una sobrecarga de trabajo.

El único sistema que hoy en día gobierna al servicio TAME CARGA, es SIGETAME, creado en SNAP, un ambiente de trabajo integrado para el desarrollo y mantenimiento de sistemas sobre el servidor AS/400 de IBM.

Debido a la creciente demanda del servicio TAME CARGA, se ha puesto en evidencia la necesidad de adaptar una aplicación que permita agilizar los procesos involucrados desde que el cliente entrega un paquete, hasta que éste es entregado en su destino.

4.2 Necesidades

El crecimiento de la empresa en aspectos como creación de nuevas rutas, aumento de frecuencias en varias rutas, adquisición y arriendo de nuevas aeronaves, aumento de

centros de acopio de carga a nivel nacional ha derivado en la impetuosa necesidad de contratar más personal para poder satisfacer de cierto modo las demandas de los clientes.

Pero no basta del talento humano para garantizar un buen servicio al cliente ya que el ser humano no es una máquina y comete errores por varios motivos que de una u otra manera afectan a los servicios prestados.

Considerando este fenómeno fundamental, adaptando buenas prácticas de la competencia y comprendiendo que la tecnología es un buen aliado de los procesos, se plantea la posibilidad de rediseñar el sistema encargado de gestionar el negocio de envío de encargos, adaptándolo de herramientas tecnológicas móviles y estáticas las cuales serán una inversión dentro de nuestra empresa que busca incrementar sus ganancias.

Para poder adaptar dichas herramientas, después de un tiempo de inactividad del proyecto por cambio de la administración, en el último trimestre del año 2011 se retomó el proyecto y se continuó con la obtención de las siguientes necesidades que se reconocieron por parte de los empleados que trabajan a diario en las instalaciones de Tame Cargo y las cuales se detallan en la siguiente la tabla:

Necesidad	Expectativa	Solución Actual	Solución Propuesta
Evitar que se deba hacer dos colas en el área de Recepción de encomiendas	El cliente podrá realizar todo el trámite de envío realizando una sola cola	Los paquetes entregados por los clientes deben ser pesados y contados, esto se realiza en una cola independiente ya que solo existe una balanza. Posteriormente, el cliente debe ubicarse nuevamente en otra cola para cancelar el valor del envío de	Instalar una balanza digital y una impresora de etiquetas en cada computador. De este modo el peso registrado será transmitido a la aplicación y esta imprimirá una etiqueta con los datos para su posterior uso.

Necesidad	Expectativa	Solución Actual	Solución Propuesta
		acuerdo al peso y piezas registrados manualmente.	
Generar una etiqueta con registro único que contenga información de tipo, origen, destino, peso, código y número de piezas	Una vez que los paquetes sean pesados, se podrá imprimir y adherir a cada paquete una etiqueta que contenga los datos más importantes	El Agente de Carga cuenta la cantidad de paquetes, los pesa y apunta en un papel dichos datos. Por último, entrega el papel al cliente.	Un único Agente de Carga recepta la mercancía, pone el encargo en la balanza, abre la aplicación para que los datos se transmitan al computador, complementa la información y de esta manera el Agente puede imprimir la etiqueta
El Agente de Carga del área de Recepción desea reducir el tiempo de ingreso de datos del cliente al sistema antes de facturar	El Agente de Carga podrá utilizar un lector óptico para transmitir la información de la etiqueta y de la cédula de identidad de la persona que realiza el envío y así evitar digitar los datos	El Agente de Carga debe digitar todos los datos solicitados para la facturación	El Agente de Carga únicamente debe complementar los datos que no vienen con la etiqueta y también estará en la capacidad de obtener los datos principales como número de cédula y nombres completos de una cédula de identidad moderna

Necesidad	Expectativa	Solución Actual	Solución Propuesta
Se necesita poder visualizar los números de Taquillas en la Factura Guía	El usuario podrá visualizar en la factura guía los números de taquilla de todos sus paquetes para poder darles un mejor seguimiento	Dado que no existen taquillas, no hay números de seguimiento para cada paquete	Modificar el programa que permite la impresión de la factura guía para incluir los números de etiquetas
El Agente de Carga en el área de Despacho desea reducir el tiempo que se demora al llenar un Manifiesto de varias Guías	El Agente de Carga podrá seleccionar un Manifiesto creado anteriormente en procesos complementarios y hará uso del PDA para asignar todo paquete leído al manifiesto seleccionado	El Agente de Carga debe acceder al sistema complementario para ingresar manualmente los números de guía que serán despachados en determinado vuelo	El Agente de Carga dispondrá de una PDA en la cual mediante uso de la aplicación, podrá elegir el número de manifiesto al que quiere adjuntar los paquetes. Posteriormente los encargos serán leídos a través de su etiqueta con código de barras
El Agente de Carga en el área de Chequeo desea reducir el tiempo que se demora en asegurarse que los paquetes que están dentro del	Hacer uso de un PDA para poder seleccionar un Manifiesto e ir leyendo todas las etiquetas que se relacionan entre si	Se ingresa en el sistema complementario todos los números de guía de manera manual para ir descargando el	Mediante el uso de la aplicación se podrá seleccionar un Manifiesto del día, ver sus detalles como por ejemplo número de vuelo y comenzar a leer

Necesidad	Expectativa	Solución Actual	Solución Propuesta
Manifiesto en realidad llegaron		listado de los paquetes pendientes	todos los códigos de barras relacionados a ese manifiesto.
El Agente de Carga en el área de Chequeo desea poder dar una ubicación al paquete que arribó lo más pronto posible	La aplicación asignará una ubicación automática para encargos de tipo: sobres pequeños, sobres grandes, mínimos	La ubicación se la asigna de manera manual en sistema complementario	La aplicación asignará una ubicación automática dependiendo de cómo se indique y se genere la etiqueta en el área inicial de Recepción
El Agente de Carga en el área de Chequeo desea poder asignar un código de una ubicación a los paquetes que no son pequeños	Ingresar en el PDA cualquier cadena de texto que haga referencia al nombre de la bodega y que se guarde una vez que el encargo sea leído	La ubicación se la asigna de manera manual en sistema complementario	Antes de leer un conjunto de paquetes, el usuario selecciona el Manifiesto y el tipo de paquete que va a leer, ingresa una ubicación y continúa con la lectura de las taquillas. La ubicación se guarda en la base de datos
El Agente de Carga en el área de Entrega desea reducir el tiempo que se demora en	Se desea realizar un proceso en el que los paquetes sean entregados rápidamente	Una vez que el Agente que hace uso del sistema complementario da la orden de	El Agente de carga recepta la petición de un cliente que viene a retirar su encargo, da la orden

Necesidad	Expectativa	Solución Actual	Solución Propuesta
buscar un paquete y entregárselo al cliente	haciendo uso de un PDA tanto para visualizar los pendientes de entrega como para descargar el paquete leído. Evitar que se impriman duplicados de facturas guía solo por saber la ubicación del encargo ya que se podrá averiguar la ubicación en la PDA	entregar, se imprime un duplicado de la factura guía. El Agente que entrega físicamente los encargos debe estar alerta de escuchar el comienzo de la impresión para darse cuenta que se debe entregar un encargo ubicado en donde detalla la guía	que se entregue el paquete en el sistema complementario. Otro Agente que se encarga de la entrega física del paquete observa el PDA en busca de encomiendas por entregar, busca el paquete en la ubicación que dice el sistema y por último entrega el encargo al cliente

Tabla 4-1: Reconocimiento de necesidades

4.3 Requerimientos Funcionales – Casos de Uso

Considerando que es una herramienta opcional la inclusión de casos de uso en un modelado ágil, se optó por incluirlos ante la necesidad resolver temas relacionados a nuestro sistema relativamente grande, con el propósito de tener una visión más clara de los pasos que deben seguirse en cada proceso (West, 2009).

4.3.1 Recepción

- Impresión de Taquilla
- Impresión de Factura Guía / Guía de Despacho

CARGA-RECP 001	Impresión de Taquilla	
Objetivo	Imprimir automáticamente taquillas de carga de cada una de las piezas ingresadas para que el Auxiliar de marca adhiera a los paquetes	
Alcance y nivel	Sistema Carga: Auxiliar en marca/Agente de Venta	
Autor(es):	HEME, PSLC	
Fecha:	2012/05/11	
Revisores:	Patricio Paredes/Marcelo Vásquez	
Precondiciones	Solicitud de cliente, aplicación y equipos habilitados	
Condición final en caso de éxito	Impresión de taquillas con éxito	
Condición final en caso de falla	Reimprimir taquillas o marcado manual	
Actores primario y secundarios	Auxiliar de marca/aplicación de impresión de taquillas/Cliente	
Disparador	Auxiliar de marca	
DESCRIPCIÓN	Paso	Acción
	1	Cliente solicita envío de carga
	2	Auxiliar selecciona categoría de carga
		1) No perecible > 1kg.
		2) Perecible > 1kg.
		3) Sobre Pequeño < 1kg
		4) Sobre grande < 1kg
		5) Mínimos
	3	Auxiliar ingresa Destino y piezas
	4	Sistema captura peso y pide confirmar datos. La captura de peso sólo se realiza si el peso del encargo es mayor a 1 Kg, es decir, si pertenecen a la categoría 1 o 2.
	5	Auxiliar confirma datos
	6	Sistema imprime taquillas para cada pieza
	7	Auxiliar marca paquetes
	8	Realizar caso de uso CARGA-RECP 002
EXTENSIONES	Paso	
	6	En el caso de que el sistema no imprime
	6a	Auxiliar solicita al sistema último número de taquilla automática
	6b	Sistema muestra último número
	6c	Auxiliar de marca confirma y reimprime taquillas
	6d	Continuar con paso 7
SUB- VARIACIONES		

INFORMACION RELACIONADA	CARGA-RECP 001: Impresión de Taquilla
Prioridad:	Alta
Tiempo de respuesta	5 segundos
Frecuencia	2500 veces al día (UIO, GYE, CUE)
Canal a los actores	Interactivo/Archivo/Base de datos
CUESTIONES NO RESUELTAS	
Fecha de entrega	2012/06/18
Superiores	Ingreso de Catálogos
Subordinados	

CARGA-RECP 002	Imprimir Factura Guía/Guía de Despacho	
Objetivo	Imprimir Factura Guía y Guía de Despacho en un tiempo más corto ya que con la lectura del código de barras de la taquilla entregada en Recepción se evitará registrar los datos de: ciudad origen, destino, categoría guía, peso y piezas. Además se podrá leer una cédula moderna que posea código de barras para obtener los nombres del cliente y el número de cédula del mismo	
Alcance y nivel	Sistema Carga: Agente de Venta	
Autor(es):	HEME, PSLC	
Fecha:	2012/06/02	
Revisores:	Patricio Paredes/Marcelo Vásquez	
Precondiciones	Taquilla con código de barras, aplicación y equipos habilitados	
Condición final en caso de éxito	Impresión de factura guía y Guía de Despacho con éxito.	
Condición final en caso de falla	Reimprimir Factura Guía y Guía de Despacho en el sistema o llenar manualmente la guía en caso de caída del sistema.	
Actores primario y secundarios	Cliente/Agente de Venta/aplicación de carga.	
Disparador	Agente de Venta	
DESCRIPCIÓN	Paso	Acción
	1	Agente se cambia al sistema complementario
	2	Agente captura información (etiquetas y cédula) con lector
	3	Sistema verifica existencia de número de taquilla automática
	4	Agente ingresa código de tipo de guía y llena información excepto lo capturado en recepción.
	5	Agente confirma datos
	6	Agente ejecuta comando para imprimir Factura y Guía de Despacho.
	7	Sistema imprime documento
	8	Sistema registra número de guía en tabla de taquillas automáticas para hacer referencia cruzada.
EXTENSIONES	Paso	
	3	En el caso de que el sistema no encuentra número de taquilla en tablas
	3a	Ingresar manualmente el número de taquilla
	3b	Continuar con paso 4
SUB- VARIACIONES	Paso	Acción
	3	En el caso de que no se encuentre el número de taquilla en la tabla, puede tratarse de otra taquilla o que ya se utilizó con otra guía. Cuando no esté operativa la aplicación de impresión automática de CB, el campo del número de taquilla se dejará en blanco y se continuará normalmente con la emisión.

INFORMACION RELACIONADA	CARGA-RECP 002: Imprimir Factura Guía/Guía de Despacho
Prioridad:	Alta
Tiempo de respuesta	5 segundos
Frecuencia	2500 veces al día (UIO, GYE, CUE)
Canal a los actores	Interactivo/Archivo/Base de datos
CUESTIONES NO RESUELTAS	
Fecha de entrega	2012/06/18
Superiores	Ingreso de Catálogos/sistema automático de taquillas
Subordinados	

4.3.2 Despacho

- Despacho de carga con Código de Barras

CARGA-DES 001	Despacho de carga con código de barras	
Objetivo	Generar el conocimiento o manifiesto de embarque sólo con las guías que fueron leídas previamente su código de barras al momento de colocar en los coches. Se evitará el chequeo del manifiesto con la lista manual que elaboran los agentes al momento de colocar la carga en los coches según el destino.	
Alcance y nivel	Sistema Carga: Agente de Despacho	
Autor(es):	HEME, PSLC	
Fecha:	2012/06/10	
Revisores:	Patricio Paredes/Marcelo Vásquez	
Precondiciones	Taquilla con código de barras pegadas a los paquetes, cabeceras de conocimientos abiertos.	
Condición final en caso de éxito	Impresión del manifiesto con la carga chequeada (leída con código de barras) con éxito.	
Condición final en caso de falla	Encerar manifiesto, bajar carga de coches e ir leyendo los códigos de barras y finalmente reimprimir el manifiesto.	
Actores primario y secundarios	Agente de despacho/aplicación de Despacho	
Disparador	Agente de Despacho	
DESCRIPCIÓN	Paso	Acción
	1	Agente lleva carga de Recepción a coches etiquetados con el destino.
	2	Agente selecciona en PDA número de conocimiento
	3	Sistema muestra taquillas con campo "Piezas enviadas" menor que campo "número de piezas"
	4	Agente lee código de barras de cada paquete
	5	Sistema valida Destino de la taquilla
	6	Sistema asigna número de Conocimiento a la guía que está relacionada y hace cálculo del campo "Piezas enviadas" en tabla Taquillas
	7	Agente continúa con paso 4 hasta terminar los paquetes
	8	Agente informa que cerrará el conocimiento a todos los demás
	9	Agente cierra conocimiento
	10	Sistema asigna código de Manifiesto Cerrado para impedir la lectura de más taquillas.
	11	Agente imprime conocimiento
	12	Agente adjunta guías en orden secuencial y por cajero comparando con manifiesto (listado)
	13	Agente adjunta conocimiento más guías en valija para que lo lleven al avión.
EXTENSIONES	Paso	
	5	En el caso de que la taquilla no coincide con el destino del número de conocimiento seleccionado
	5a	Sistema emite un mensaje de error
	5b	Agente separa el paquete Continúa con el paso 4.
	6	En el caso de que el sistema ya encuentra un número de conocimiento en la guía

	6a	Si el número de “Piezas Enviadas” de la Tabla Taquillas es mayor que el campo Piezas de la última tabla, el sistema emite mensaje de que “hay piezas mayor de las recibidas” Continuar con paso 6
SUB- VARIACIONES	Paso	Acción

INFORMACION RELACIONADA	CARGA-DES 001: Despacho de carga con código de barras
Prioridad:	Alta
Tiempo de respuesta	< a 1 segundo por lectura
Frecuencia	2500 veces al día (UIO, GYE, CUE)
Canal a los actores	Interactivo/Archivo/Base de datos
CUESTIONES NO RESUELTAS	
Fecha de entrega	2012/06/18
Superiores	Sistema automático de taquillas/Sistema de Venta de guías.
Subordinados	

4.3.3 Chequeo

- Chequeo de Carga – No tránsito
- Chequeo de Carga – Guías en tránsito

CARGA-CHE 001	Chequeo de Carga – No tránsito	
Objetivo	Mejorar el tiempo de chequeo y ubicación de la carga para dejar listo para el proceso de Entrega. Esto se logrará con la ayuda del lector del código de barras para ubicar una guía.	
Alcance y nivel	Sistema Carga: Agente de Chequeo	
Autor(es):	HEME, PSLC	
Fecha:	2012/06/17	
Revisores:	Patricio Paredes/Marcelo Vásquez	
Precondiciones	Paquetes con taquillas automatizadas con código de barras pegadas, conocimientos de embarque generado en origen.	
Condición final en caso de éxito	Listado general de guías con ubicación con un 40% del tiempo más rápido.	
Condición final en caso de falla	Chequeo sin lector de código de barras	
Actores primario y secundarios	Agente de Chequeo/aplicación de Chequeo	
Disparador	Agente de Chequeo	
DESCRIPCIÓN	Paso	Acción
	1	Agente clasifica manualmente la carga en: Sobres pequeños, sobres grandes, valija interna, carga pesada.
	2	Agente grapa las copias de la guía en los paquetes y/o sobres
	3	Agente ingresa en el sistema número de Conocimiento y lugar de Entrega.
	4	Sistema despliega las taquillas automatizadas con su número de guía
	5	Agente por cada lugar de entrega clasifica en el sistema en categorías (sobres, valija, carga pesada)
	6	Sistema despliega taquillas clasificadas por categoría
	7	Agente ingresa ubicación en un campo para el caso de que los encargos sean mayores a 1Kg.

	8	Agente lee código de barras de taquillas automatizadas
	9	Sistema encuentra taquilla y registra de ubicación y calcula el campo 'piezas recibidas', cambia estados y muestra resultado
	10	Agente lee código de barras de los demás paquetes y/o sobres, paso 8.
	11	Agente imprime conocimiento con ubicación por lugar de entrega.
	12	Agente ubica paquetes en anaqueles y/o bodegas
	13	Archiva Manifiestos
EXTENSIONES	Paso	
	8	En el caso de que la taquilla no se encuentre en el manifiesto
	8a	Sistema emite un mensaje de error
	8b	Agente separa el paquete, para solucionar con Origen Continúa leyendo otra taquilla
SUB- VARIACIONES	Paso	Acción
	3	En el caso de que no lleguen taquillas automatizadas
	3a	El sistema dará la opción de chequear con la aplicación actual (Con taquillas Auto, Sin taquillas Automáticas) Agente chequea carga con el sistema actual
	3b	Continuar con el Paso 11.

INFORMACION RELACIONADA	CARGA-CHE 001: Chequeo de Carga – No tránsito
Prioridad:	Alta
Tiempo de respuesta	< a 1 segundo por lectura
Frecuencia	2500 veces al día entre UIO, GYE, CUE
Canal a los actores	Interactivo/Archivo/Base de datos
CUESTIONES NO RESUELTAS	
Fecha de entrega	2012/06/18
Superiores	Sistema automático de taquillas/Módulo de Despacho
Subordinados	Módulo de Entrega de Guías.

CARGA-CHE 002	Chequeo de Carga –Guías en tránsito	
Objetivo	Mejorar el tiempo de chequeo y ubicación de la carga para dejar listo para el proceso Despacho.	
Alcance y nivel	Sistema Carga: Agente de Chequeo/Agente de Despacho	
Autor(es):	HEME, PSLC	
Fecha:	2012/06/17	
Revisores:	Patricio Paredes/Marcelo Vásquez	
Precondiciones	Paquetes con taquillas automatizadas, Conocimientos de embarque generado en origen.	
Condición final en caso de éxito	Envío seguro de guías en tránsito hacia el destino con ayuda de las taquillas automatizadas con código de barra.	
Condición final en caso de falla	Chequeo sin lector de código de barras. (Con proceso actual)	
Actores primario y secundarios	Agentes de Chequeo y Despacho/aplicación de Chequeo/Despacho	
Disparador	Agente de Chequeo	
DESCRIPCIÓN	Paso	Acción
	1	Agente clasifica manualmente la carga en tránsito
	2	Agente valida manualmente carga con guías del Conocimiento en tránsito.

	3	Agente de Chequeo entrega a Agente de Despacho carga más guías
	4	Agente de Despacho realiza el proceso de Caso de Uso: CARGA-DES 001
EXTENSIONES	Paso	
	2	En el caso de encontrar diferencias entre las guías, listado y paquetes
	2a	Agente soluciona problema con agentes del origen de envío.
SUB- VARIACIONES	Paso	Acción

INFORMACION RELACIONADA	CARGA-CHE 002: Chequeo de Carga – Guías en tránsito
Prioridad:	Alta
Tiempo de respuesta	Manual – 5 minutos
Frecuencia	10 guías por Conocimiento
Canal a los actores	Interactivo/Archivo/Base de datos
CUESTIONES NO RESUELTAS	
Fecha de entrega	2012/06/18
Superiores	Sistema automático de taquillas/Módulo de Despacho
Subordinados	

4.3.4 Entrega

- Entrega de Guías - Sobres
- Entrega de Guías – Carga pesada en Bodegas

CARGA-ENT 001	Entrega de Guías - Sobres	
Objetivo	Mejorar el tiempo Entrega con seguridad y agilidad	
Alcance y nivel	Sistema Carga: Agente de Entrega	
Autor(es):	HEME, PSLC	
Fecha:	2012/06/18	
Revisores:	Patricio Paredes/Marcelo Vásquez	
Precondiciones	Guías chequeadas y con ubicación, sistema habilitado.	
Condición final en caso de éxito	Guía entregada al destinatario y descargada en el sistema (Estado=Entregado)	
Condición final en caso de falla	Entrega en base a listados de contingencia y luego ser descargadas en el sistema.	
Actores primario y secundarios	Agentes de Entrega /aplicación entrega	
Disparador	Agente de entrega.	
DESCRIPCIÓN	Paso	Acción
	1	Cliente solicita entrega de guía
	2	Agente ubica guía en el sistema en base a información suministrada por el cliente
	3	Sistema muestra guía con estado de Entrega y ubicación
	4	Agente ubica sobre en casilleros
	5	Agente solicita identificación a cliente o autorización
	6	Cliente firma guía de Recibí Conforme y entrega a Agente
	7	Agente valida firma y registra datos de quién retira en sistema
	8	Sistema guarda información y registra “Guía Entregada”
	9	Agente entrega sobre a cliente
	10	Agente archiva guía firmada por cliente

EXTENSIONES	Paso	
	2	En el caso de no encontrar guía
	2a 2b	Sistema informa la inexistencia Agente agota búsqueda con otros parámetros (número de guía, nombre remitente o destinatario)
SUB- VARIACIONES	Paso	Acción

INFORMACION RELACIONADA	CARGA-ENT 001: Entrega de Guías - Sobres
Prioridad:	Alta
Tiempo de respuesta	5 segundos en localizar guía
Frecuencia	200 por día (UIO, GYE, CUE)
Canal a los actores	Interactivo/Archivo/Base de datos
CUESTIONES NO RESUELTAS	
Fecha de entrega	2012/06/18
Superiores	Sistema automático chequeo
Subordinados	

CARGA-ENT 002	Entrega de Guías – Carga pesada en bodegas	
Objetivo	Mejorar el tiempo Entrega con seguridad, agilidad y ahorrando insumos (impresora y papel continuo)	
Alcance y nivel	Sistema Carga: Agente de Entrega	
Autor(es):	HEME, PSLC	
Fecha:	2012/06/18	
Revisores:	Patricio Paredes/Marcelo Vásquez	
Precondiciones	Guías chequeadas y con ubicación, sistema habilitado.	
Condición final en caso de éxito	Guía entregada al destinatario y descargada en el sistema en línea (Estado=Entregado)	
Condición final en caso de falla	Entrega en base a listados de contingencia y luego ser descargadas en el sistema.	
Actores primario y secundarios	Agentes de Entrega /aplicación entrega	
Disparador	Agente de entrega.	
DESCRIPCIÓN	Paso	Acción
	1	Cliente solicita entrega de guía
	2	Agente ubica guía en el sistema en base a información suministrada por el cliente
	3	Sistema muestra guía con estado de Entrega y ubicación
	4	Agente solicita identificación a cliente o autorización
	5	Agente de oficina de entrega registra datos de quién retira en sistema (no pone estado entregado)
	6	Sistema muestra guía por entregar en PDA con datos de la ubicación (estado=No entregado e identificación de quien retira)
	7	Agente en bodega localiza paquetes y coloca en carreta junto con otras
	8	Cliente en mostrador firma guía de ‘Recibí conforme’
	9	Agente lee código de barras de paquetes a entregarse
	10	Sistema asigna estado Entregado luego de que el número de piezas enviadas sea igual al número de piezas leídas en la entrega.
	11	Agente entrega carga a cliente
	12	Agente archiva guía firmada por cliente
EXTENSIONES	Paso	

	2	En el caso de no encontrar guía
	2a	Sistema informa la inexistencia
	2b	Agente agota búsqueda con otros parámetros (número de guía, nombre remitente o destinatario)
SUB- VARIACIONES	Paso	Acción

INFORMACION RELACIONADA	CARGA-ENT 002: Entrega de Guías - Carga pesada en bodegas
Prioridad:	Alta
Tiempo de respuesta	5 segundos en localizar guía en oficina
Frecuencia	200 por día (UIO, GYE, CUE)
Canal a los actores	Interactivo/Archivo/Base de datos
CUESTIONES NO RESUELTAS	
Fecha de entrega	2012/06/18
Superiores	Sistema automático chequeo
Subordinados	

4.4 Requerimientos no funcionales

- El sistema tendrá una alta disponibilidad.
- El sistema será fácilmente accesible desde los equipos de trabajo.
- El sistema tendrá una interfaz de usuario amigable.
- La aplicación responderá en tiempos adecuados menores a 5 segundos, permitiendo a los usuarios desempeñar sus labores normalmente sin que sufran perjuicio por latencia u otros factores propios tecnológicos.
- El sistema optimizará el uso de insumos y suministros.
- El sistema tendrá una arquitectura que permita su evolución continua.
- El sistema tendrá las seguridades necesarias para permitir el acceso únicamente a los usuarios registrados.

5 DESARROLLO DEL SISTEMA

En este capítulo, se detallará el desarrollo del proyecto según los requerimientos expuestos en el capítulo 4, describiendo cada proceso según los fundamentos teóricos del capítulo 2 y 3.

5.1 Definición del Product Backlog

El Product Backlog o pila del producto, es definido por el dueño del producto, quien es responsable de definir la prioridad para cada requerimiento que éste contenga, así como de mantenerlo actualizado.

Una vez que el dueño del producto ha definido todos los requerimientos, los cuales serán la “materia prima” para poder crear la pila del producto (Product Backlog), empezaremos con el desarrollo de nuestro sistema.

En la Figura 5-1 se puede observar que existen 19 requerimientos iniciales que abarcarán las diferentes actividades o tareas requeridas en el proyecto.

En este punto empezaremos con la definición de una lista de ítems a los cuales se les conoce también como “Historias de Usuarios”.

Cabe indicar que la columna “Valor Empresarial” corresponde a la “Prioridad” de la funcionalidad, la cual es asignada por el dueño del producto, y la columna “Trabajo” representa el esfuerzo estimado en horas hombre, este valor es calculado por el equipo de trabajo.

Team Foundation Server 2012 / ProyectoTAMECargo Paul Lopez Ayuda

INICIO CÓDIGO **TRABAJO** COMPILACIÓN PRUEBA

trabajo pendiente panel elementos de trabajo

Trabajo pendiente del prod...

Actual

Sprint 1

Futuro

Sprint 2

Sprint 3

Sprint 4

Trabajo pendiente del producto

elementos de trabajo pendiente panel

Crear consulta de trabajo pendiente | Opciones de columna

Tipo Elemento de trabajo pendiente d

Título Agregar

Ord...	Título	Estado De Tarea	Trabajo	Valor Empresarial
1	Reingeniería a base de datos del sistema	Nuevo	37	28
2	Selección y despliegue de información de manifiestos	Nuevo	27	25
3	Captura de peso marcado por balanza electrónica	Nuevo	20	13
4	Despliegue de ubicaciones de entrega de paquetes en ciudad de des..	Nuevo	10	13
5	Opción de regresar al menú inmediato superior desde cada ventana	Nuevo	3	8
6	Ingreso de información del paquete según categoría	Nuevo	30	30
7	Categorizar paquetes en módulo de chequeo	Nuevo	29	40
8	Control de usuarios con acceso al sistema	Nuevo	25	20
9	Despliegue de todos los destinos nacionales para envío de paquetes	Nuevo	8	12
10	Despliegue de información relacionada al usuario en cada ventana	Nuevo	5	10
11	Impresión de taquillas	Nuevo	40	50
12	Reimpresión de taquillas	Nuevo	25	23
13	Despliegue de menú de opciones para módulo de Recepción	Nuevo	17	16
14	Despliegue de menú de opciones	Nuevo	15	15
15	Leer código de barras de las taquillas en cada módulo	Nuevo	35	60
16	Visualizar encargos pendientes de procesar en cada módulo	Nuevo	28	29
17	Modificación de estado de taquilla	Nuevo	20	21
18	Ingreso de una ubicación para paquetes chequeados	Nuevo	13	11
19	Salir del sistema	Nuevo	2	6

Figura 5-1: Product Backlog

5.2 Sprints

5.2.1 Sprint 1

5.2.1.1 Fase de Planificación

5.2.1.1.1 Planificación del Sprint

En la jornada de trabajo previa al inicio del Sprint 1 se especificaron las historias de usuario que deberán estar concluidas al final de esta iteración y el objetivo de la iteración.

5.2.1.1.1.1 Historias de Usuario incluidas en el Sprint

Las historias de usuario incluidas en el presente Sprint se indican en la Figura 5-2.

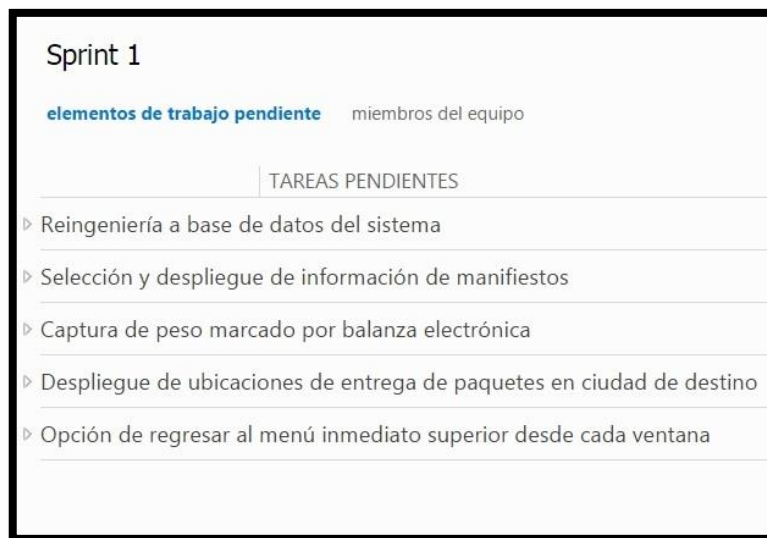


Figura 5-2: Historias de Usuario incluidas en el Sprint 1

5.2.1.1.1.2 Objetivo del Sprint

Demostrar la implementación de las nuevas funcionalidades de acuerdo a las historias de usuario el 13 de Julio de 2012.

5.2.1.1.2 Definición del Sprint Backlog

El Sprint Backlog se define seleccionando las historias de usuario con mayor importancia para el dueño del producto, el número de historias depende de la capacidad de trabajo del equipo de desarrollo, en este caso 10 horas diarias; debido a que las historias de usuario de mayor prioridad representaban una cantidad de trabajo superior a la mitad de la capacidad de trabajo del equipo de desarrollo, se repartieron entre los 4 Sprints. El dueño del producto estuvo de acuerdo con esto considerando que el objetivo era tener todo el

desarrollo terminado lo antes posible, y el avance se iría evaluando mediante demostraciones al finalizar cada sprint.

En la Figura 5-3 se detallan todas las tareas asociadas a las historias de usuario asignadas a la iteración 1.

Sprint 1							
elementos de trabajo pendiente		miembros del equipo					
		TAREAS PENDIENTES 97 h				EN CURSO	LISTO
Reingeniería a base de datos del sistema 37 h	+	Modificación de modelo conceptual de base de datos 4	Análisis modificación base de datos 6	Generación modelo físico base de datos 3	Creación de tablas y nuevas relaciones en SNAP 6		
		Validación integridad Base de Datos 8	Validar información registrada en Base de Datos 4	Pruebas de regresión 3	Creación de clase Utilitaria "BaseDatos" de conexión a base de datos 2		
		Creación de solución y estructura de proyectos 1					
Selección y despliegue de información de manifiestos 27 h	+	Agregar elementos gráficos para búsqueda de manifiestos a las 4	Codificación de capa de acceso a datos 10	Codificación GUI 3	Codificación de capa de lógica de negocio 4		
		Creación de clase "CargaBEL" 3	Creación de clase "UsuarioBEL" 1	Creación clase "CargaBLL" 1	Creación de clase "CargaDAL" 1		
Captura de peso marcado por balanza electrónica 20 h	+	Controlar que únicamente las categorías asignadas arrojen a 2	Integración de balanza electrónica con equipos 6	Codificación de módulo que permita la comunicación entre la antena al 4	Agregar elementos gráficos para desplegar el peso medido por la 3		
		Codificación GUI 5					
Despliegue de ubicaciones de entrega de paquetes en ciudad de destino 10 h	+	Agregar elementos gráficos para despliegue de ubicaciones por 2	Codificación de capa de acceso a datos 2	Codificación GUI 3	Creación de clase "EntregaBEL" 1		
		Creación de clase "EntregaBLL" 1	Creación de clase "EntregaDAL" 1				
Opción de regresar al menú inmediato superior desde cada ventana 3 h	+	Incluir botón regresar en todas las ventanas 1	Codificación de función que determine a que menú se debe 2				

Figura 5-3: Sprint 1 Backlog

5.2.1.1.3 Estimación del esfuerzo del Sprint

De la Figura 5-4 a la Figura 5-8, se aprecian las tareas necesarias para completar las historias de usuario del sprint actual.

Sprint 1			
elementos de trabajo pendiente miembros del equipo			
TAREAS PENDIENTES 97 h			
Reingeniería a base de datos del sistema 37 h	Modificación de modelo conceptual de base de datos 4	Análisis modificación base de datos 6	Generación modelo físico base de datos 3
	Creación de tablas y nuevas relaciones en SNAP 6	Validación integridad Base de Datos 8	Validar información registrada en Base de Datos 4
	Pruebas de regresión 3	Creación de clase utilitaria "BaseDatos" de conexión a base de datos 2	Creación de solución y estructura de proyectos 1
․ Selección y despliegue de información de manifiestos			
․ Captura de peso marcado por balanza electrónica			
․ Despliegue de ubicaciones de entrega de paquetes en ciudad de destino			
․ Opción de regresar al menú inmediato superior desde cada ventana			

Figura 5-4: Esfuerzo de cada tarea para la primera historia de usuario

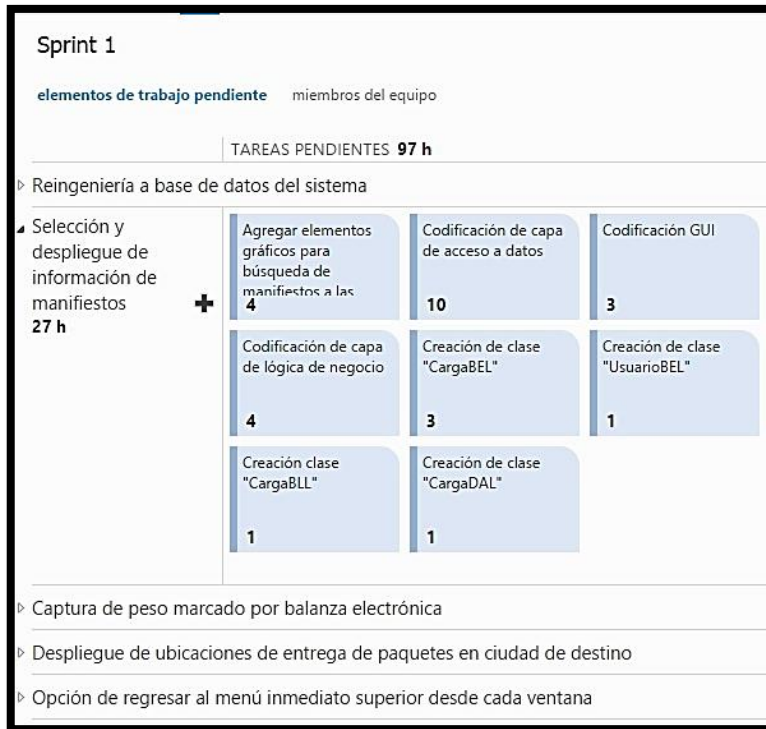


Figura 5-5: Esfuerzo de cada tarea para la segunda historia de usuario

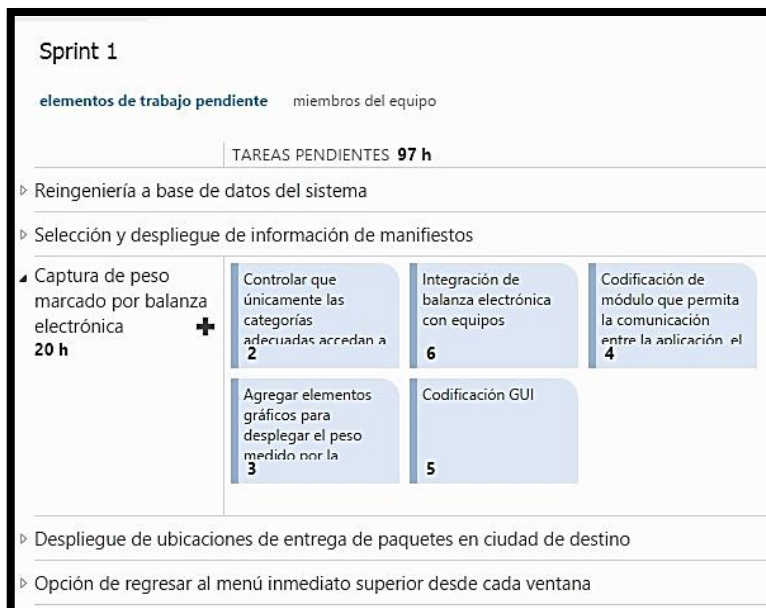


Figura 5-6: Esfuerzo de cada tarea para la tercera historia de usuario

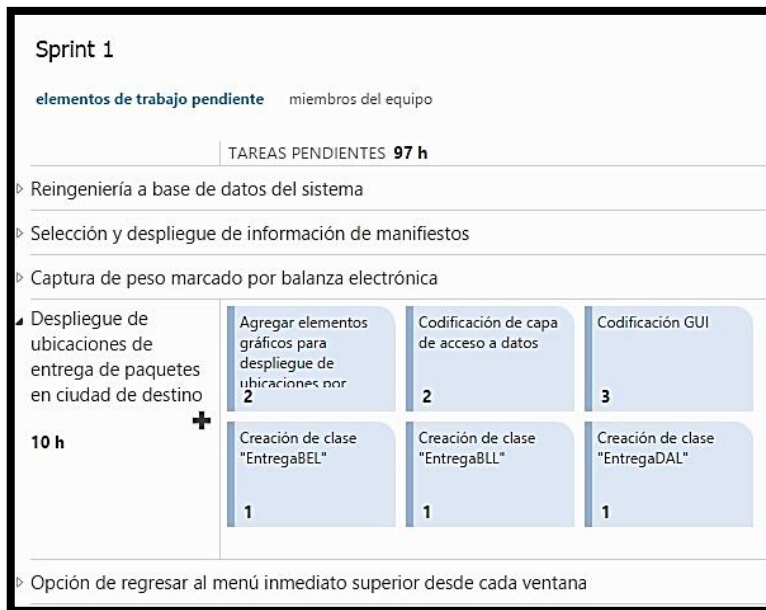


Figura 5-7: Esfuerzo de cada tarea para la cuarta historia de usuario

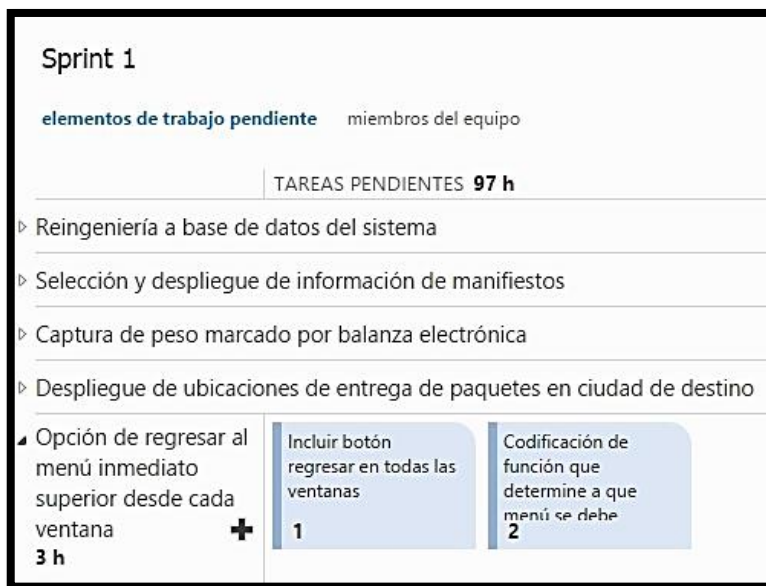


Figura 5-8: Esfuerzo de cada tarea para la quinta historia de usuario

5.2.1.2 Fase de Seguimiento

5.2.1.2.1 Desarrollo del Sprint

La reunión de seguimiento del Sprint se lleva a cabo diariamente a primera hora entre los integrantes del equipo de trabajo, dónde se comenta el trabajo realizado por cada integrante desde la anterior reunión, así como los problemas encontrados y el trabajo a realizar hasta la siguiente reunión.

En estas reuniones diarias, se da el seguimiento necesario para pronosticar si se cumplirá el trabajo comprometido en las fechas establecidas.

De la Figura 5-9 a la Figura 5-18, se detalla el avance diario realizado por el equipo de trabajo.

5.2.1.2.1.1 Día 1

Sprint 1		TAREAS PENDIENTES 83 h				EN CURSO 11 h	LISTO
Reingeniería a base de datos del sistema 37 h	Modificación de modelo conceptual de base de datos 4	Generación modelo físico base de datos 3	Creación de tablas y nuevas relaciones en SNAP 6	Validación integridad Base de Datos 8	Validación modificación base de datos 6		
	Validar información registrada en Base de Datos 4	Pruebas de regresión 3	Creación de clase utilitaria "BaseDatos" de conexión a base de datos 2	Creación de solución y estructura de proyectos 1			
Selección y despliegue de información de manifiestos 27 h	Agregar elementos gráficos para búsqueda de manifiestos a la aplicación 4	Codificación de capa de acceso a datos 10	Codificación GUI 3	Codificación de capa de lógica de negocio 4			
	Creación de clase "CargaBEL" 3	Creación de clase "UsuarioBEL" 1	Creación clase "CargaBLL" 1	Creación de clase "CargaDAL" 1			
Captura de peso marcado por balanza electrónica 17 h	Controlar que únicamente las categorías adecuadas arrojen a la aplicación 2	Integración de balanza electrónica con equipos 6	Codificación de módulo que permita la comunicación entre la aplicación y la balanza 4		Codificación GUI 5	Agregar elementos gráficos para desplegar el peso medido en la aplicación 1	
Despliegue de ubicaciones de entrega de paquetes en ciudad de destino 10 h	Agregar elementos gráficos para despliegue de ubicaciones en la aplicación 2	Codificación de capa de acceso a datos 2	Codificación GUI 3	Creación de clase "EntregaBEL" 1			
	Creación de clase "EntregaBLL" 1	Creación de clase "EntregaDAL" 1					
Opción de regresar al menú inmediato superior desde cada ventana 3 h	Incluir botón regresar en todas las ventanas 1	Codificación de función que determine a que menú se debe regresar 2					

Figura 5-9: Panel de Tareas al final del día 1

5.2.1.2.1.2 Día 2

Sprint 1						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 77 h				EN CURSO	LISTO
Reingeniería a base de datos del sistema 27 h	Generación modelo físico base de datos 3	Creación de tablas y nuevas relaciones en SNAP 6	Validación integridad Base de Datos 8	Validar información registrada en Base de Datos 4		Modificación de modelo conceptual de base de datos Análisis modificación base de datos
	Pruebas de regresión 3	Creación de clase utilitaria "BaseDatos" de conexión a base de datos 2	Creación de solución y estructura de proyectos 1			
Selección y despliegue de información de manifiestos 27 h	Agregar elementos gráficos para búsqueda de manifiestos a la 4	Codificación de capa de acceso a datos 10	Codificación GUI 3	Codificación de capa de lógica de negocio 4		
	Creación de clase "CargaBEL" 3	Creación de clase "UsuarioBEL" 1	Creación clase "CargaBLL" 1	Creación de clase "CargaDAL" 1		
Captura de peso marcado por balanza electrónica 10 h	Integración de balanza electrónica con equipos 6	Codificación de módulo que permita la comunicación entre la antirradiación al 4				Controlar que únicamente las categorías adecuadas arrojan a Agregar elementos gráficos para desplegar el peso medido en la Codificación GUI
Despliegue de ubicaciones de entrega de paquetes en ciudad de destino 10 h	Agregar elementos gráficos para despliegue de ubicaciones en 2	Codificación de capa de acceso a datos 2	Codificación GUI 3	Creación de clase "EntregaBEL" 1		
	Creación de clase "EntregaBLL" 1	Creación de clase "EntregaDAL" 1				
Opción de regresar al menú inmediato superior desde cada ventana 3 h	Incluir botón regresar en todas las ventanas 1	Codificación de función que determine a que menú se debe 2				

Figura 5-10: Panel de Tareas al final del día 2

5.2.1.2.1.3 Día 3

Sprint 1						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 62 h				EN CURSO 12 h	LISTO
Reingeniería a base de datos del sistema 24 h	Validación integridad Base de Datos 8 Creación de solución y estructura de proyectos 1	Validar información registrada en Base de Datos 4	Pruebas de regresión 3	Creación de clase utilitaria "BaseDatos" de conexión a base de datos 2	Creación de tablas y nuevas relaciones en SNAP 6	Modificación de modelo conceptual de base de datos Análisis modificación base de datos Generación modelo físico base de datos
Selección y despliegue de información de manifiestos 27 h	Agregar elementos gráficos para búsqueda de manifiestos a la x 4 Creación de clase "CargaBEL" 3	Codificación de capa de acceso a datos 10 Creación de clase "UsuarioBEL" 1	Codificación GUI 3 Creación clase "CargaBLL" 1	Codificación de capa de lógica de negocio 4 Creación de clase "CargaDAL" 1		
Captura de peso marcado por balanza electrónica 10 h	Codificación de módulo que permita la comunicación entre la aplicación al 4			Integración de balanza electrónica con equipos 6		Controlar que únicamente las categorías adecuadas arrojan a Agregar elementos gráficos para desplegar el peso medido en la Codificación GUI
Despliegue de ubicaciones de entrega de paquetes en ciudad de destino 10 h	Agregar elementos gráficos para despliegue de ubicaciones en 2 Creación de clase "EntregaBLL" 1	Codificación de capa de acceso a datos 2 Creación de clase "EntregaDAL" 1	Codificación GUI 3	Creación de clase "EntregaBEL" 1		
Opción de regresar al menú inmediato superior desde cada ventana 3 h	Incluir botón regresar en todas las ventanas 1	Codificación de función que determine a que menú se debe 2				

Figura 5-11: Panel de Tareas al final del día 3

5.2.1.2.1.4 Día 4

Sprint 1									
elementos de trabajo pendiente miembros del equipo									
	TAREAS PENDIENTES 50 h				EN CURSO 8 h	LISTO			
<ul style="list-style-type: none"> Reingeniería a base de datos del sistema 18 h 	<ul style="list-style-type: none"> Validar información registrada en Base de Datos 4 	<ul style="list-style-type: none"> Pruebas de regresión 3 	<ul style="list-style-type: none"> Creación de clase utilitaria "BaseDatos" de conexión a base de datos 2 	<ul style="list-style-type: none"> Creación de solución y estructura de proyectos 1 	<ul style="list-style-type: none"> Validación integridad Base de Datos 8 	<ul style="list-style-type: none"> Modificación de modelo conceptual de base de datos 	<ul style="list-style-type: none"> Análisis modificación base de datos 	<ul style="list-style-type: none"> Generación modelo físico base de datos 	<ul style="list-style-type: none"> Creación de tablas y nuevas relaciones en SNAP
<ul style="list-style-type: none"> Selección y despliegue de información de manifiestos 27 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para búsqueda de manifiestos a la 4 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 10 	<ul style="list-style-type: none"> Codificación GUI 3 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 					
<ul style="list-style-type: none"> Captura de peso marcado por balanza electrónica + 						<ul style="list-style-type: none"> Controlar que únicamente las categorías arduas arduas a 	<ul style="list-style-type: none"> Integración de balanza electrónica con equipos 	<ul style="list-style-type: none"> Codificación de módulo que permita la comunicación entre la aplicación y el 	<ul style="list-style-type: none"> Agregar elementos gráficos para desplegar el peso medido por la
<ul style="list-style-type: none"> Despliegue de ubicaciones de entrega de paquetes en ciudad de destino 10 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para despliegue de ubicaciones en 2 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 2 	<ul style="list-style-type: none"> Codificación GUI 3 	<ul style="list-style-type: none"> Creación de clase "EntregaBEL" 1 					
<ul style="list-style-type: none"> Opción de regresar al menú inmediato superior desde cada ventana 3 h 	<ul style="list-style-type: none"> Incluir botón regresar en todas las ventanas 1 	<ul style="list-style-type: none"> Codificación de función que determine a que menú se debe 2 							

Figura 5-12: Panel de Tareas al final del día 4

5.2.1.2.1.5 Día 5

Sprint 1									
elementos de trabajo pendiente miembros del equipo									
	TAREAS PENDIENTES 45 h				EN CURSO 8 h	LISTO			
<ul style="list-style-type: none"> Reingeniería a base de datos del sistema 18 h 	<ul style="list-style-type: none"> Validar información registrada en Base de Datos 4 	<ul style="list-style-type: none"> Pruebas de regresión 3 	<ul style="list-style-type: none"> Creación de clase utilitaria "BaseDatos" de conexión a base de datos 2 	<ul style="list-style-type: none"> Creación de solución y estructura de proyectos 1 	<ul style="list-style-type: none"> Validación integridad Base de Datos 8 	<ul style="list-style-type: none"> Modificación de modelo conceptual de base de datos 	<ul style="list-style-type: none"> Análisis modificación base de datos 	<ul style="list-style-type: none"> Generación modelo físico base de datos 	<ul style="list-style-type: none"> Creación de tablas y nuevas relaciones en SNAP
<ul style="list-style-type: none"> Selección y despliegue de información de manifiestos 27 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para búsqueda de manifiestos a la 4 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 10 	<ul style="list-style-type: none"> Codificación GUI 3 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 					
<ul style="list-style-type: none"> Creación de clase "CargaSEL" 3 	<ul style="list-style-type: none"> Creación de clase "UsuarioSEL" 1 	<ul style="list-style-type: none"> Creación clase "CargaBLL" 1 	<ul style="list-style-type: none"> Creación de clase "CargaDAL" 1 						
<ul style="list-style-type: none"> Captura de peso marcado por balanza electrónica 						<ul style="list-style-type: none"> Controlar que únicamente las categorías avanzadas arrojen a 	<ul style="list-style-type: none"> Integración de balanza electrónica con equipos 	<ul style="list-style-type: none"> Codificación de módulo que permita la comunicación entre la aplicación y el 	<ul style="list-style-type: none"> Agregar elementos gráficos para desplegar el peso medido por la
						<ul style="list-style-type: none"> Codificación GUI 			
<ul style="list-style-type: none"> Despliegue de ubicaciones de entrega de paquetes en ciudad de destino 5 h 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 2 	<ul style="list-style-type: none"> Creación de clase "EntregaSEL" 1 	<ul style="list-style-type: none"> Creación de clase "EntregaBLL" 1 	<ul style="list-style-type: none"> Creación de clase "EntregaDAL" 1 		<ul style="list-style-type: none"> Agregar elementos gráficos para despliegue de ubicaciones por 	<ul style="list-style-type: none"> Codificación GUI 		
<ul style="list-style-type: none"> Opción de regresar al menú inmediato superior desde cada ventana 3 h 	<ul style="list-style-type: none"> Incluir botón regresar en todas las ventanas 1 	<ul style="list-style-type: none"> Codificación de función que determine a que menú se debe 2 							

Figura 5-13: Panel de Tareas al final del día 5

5.2.1.2.1.6 Día 6

TAREAS PENDIENTES 37 h		EN CURSO				LISTO					
Reingeniería a base de datos del sistema 7 h	Validar información registrada en Base de Datos 4	Pruebas de regresión 3					Modificación de modelo conceptual de base de datos	Análisis modificación base de datos	Generación modelo físico base de datos	Creación de tablas y nuevas relaciones en SNAP	Validación integridad Base de Datos
Selección y despliegue de información de manifiestos 27 h	Agregar elementos gráficos para búsqueda de manifiestos 4	Codificación de capa de acceso a datos 10	Codificación GUI 3	Codificación de capa de lógica de negocio 4	Creación de clase "CargaBEL" 3						
	Creación de clase "UsuarioBEL" 1	Creación clase "CargaBL" 1	Creación de clase "CargaDAL" 1								
Captura de peso marcado por balanza electrónica +							Controlar que unicamente las categorías seleccionadas	Integración de balanza electrónica con equipos	Codificación de módulo que permita la comunicación entre la aplicación al	Agregar elementos gráficos para desplegar el peso medido por la	Codificación GUI
Despliegue de ubicaciones de entrega de paquetes en ciudad de destino +							Agregar elementos gráficos para despliegue de ubicaciones por	Codificación de capa de acceso a datos	Codificación GUI	Creación de clase "EntregaBEL"	Creación de clase "EntregaBL"
							Creación de clase "EntregaDAL"				
Opción de regresar al menú inmediato superior desde cada ventana 3 h	Incluir botón regresar en todas las ventanas 1	Codificación de función que determine a que menú se debe									

Figura 5-14: Panel de Tareas al final del día 6

5.2.1.2.1.7 Día 7

Sprint 1						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 23 h				EN CURSO 6 h	LISTO
<ul style="list-style-type: none"> Reingeniería a base de datos del sistema 3 h 					<ul style="list-style-type: none"> Pruebas de regresión 3 	<ul style="list-style-type: none"> Modificación de modelo conceptual de base de datos Análisis modificación base de datos Generación modelo físico base de datos Creación de tablas y nuevas relaciones en SNAP Validación integridad Base de Datos Validar información registrada en Base de Datos Creación de clase utilitaria "BaseDatos" de conexión a base de datos Creación de solución y estructura de proyectos
<ul style="list-style-type: none"> Selección y despliegue de información de manifiestos 23 h 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 10 Creación clase "CargaBLL" 1 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 Creación de clase "CargaDAL" 1 	<ul style="list-style-type: none"> Creación de clase "CargaBEL" 3 	<ul style="list-style-type: none"> Creación de clase "UsuarioBEL" 1 	<ul style="list-style-type: none"> Codificación GUI 3 	<ul style="list-style-type: none"> Agregar elementos gráficos para búsqueda de manifiestos a la
<ul style="list-style-type: none"> Captura de peso marcado por balanza electrónica 3 h 						<ul style="list-style-type: none"> Controlar que únicamente las categorías adecuadas arrastren a Integración de balanza electrónica con equipos Codificación de módulo que permita la comunicación entre la antirradiación al Agregar elementos gráficos para desplegar el peso medido por la Codificación GUI
<ul style="list-style-type: none"> Despliegue de ubicaciones de entrega de paquetes en ciudad de destino 3 h 						<ul style="list-style-type: none"> Agregar elementos gráficos para despliegue de ubicaciones por Codificación de capa de acceso a datos Codificación GUI Creación de clase "EntregaBEL" Creación de clase "EntregaDAL" Creación de clase "EntregaBEL"
<ul style="list-style-type: none"> Opción de regresar al menú inmediato superior desde cada ventana 3 h 	<ul style="list-style-type: none"> Incluir botón regresar en todas las ventanas 1 	<ul style="list-style-type: none"> Codificación de función que determine a que menú se dirige 2 				

Figura 5-101. Plan de trabajo de Sprint de Día 7

5.2.1.2.1.8 Día 8

Sprint 1											
elementos de trabajo pendiente miembros del equipo											
	TAREAS PENDIENTES 17 h					EN CURSO	LISTO				
<ul style="list-style-type: none"> Reingeniería a base de datos del sistema + 							<ul style="list-style-type: none"> Modificación de modelo conceptual de base de datos Análisis modificación base de datos Generación modelo físico base de datos Creación de tablas y nuevas relaciones en SNAP Validación integridad Base de Datos 				
<ul style="list-style-type: none"> Selección y despliegue de información de manifiestos 17 h + 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 10 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 	<ul style="list-style-type: none"> Creación de clase "UsuarioBÉL" 1 	<ul style="list-style-type: none"> Creación clase "CargaBÉL" 1 	<ul style="list-style-type: none"> Creación de clase "CargaDAL" 1 		<ul style="list-style-type: none"> Agregar elementos gráficos para búsqueda de manifiestos a la Codificación GUI Creación de clase "CargaBÉL" 				
<ul style="list-style-type: none"> Captura de peso marcado por balanza electrónica + 							<ul style="list-style-type: none"> Controlar que únicamente las categorías seleccionadas sean Integración de balanza electrónica con equipos Codificación de módulo que permita la comunicación entre la aplicación al Agregar elementos gráficos para desplegar el peso marcado por la Codificación GUI 				
<ul style="list-style-type: none"> Despliegue de ubicaciones de entrega de paquetes en ciudad de destino + 							<ul style="list-style-type: none"> Agregar elementos gráficos para despliegue de ubicaciones para Codificación de capa de acceso a datos Codificación GUI Creación de clase "EntregaBÉL" Creación de clase "EntregaDAL" 				
<ul style="list-style-type: none"> Opción de regresar al menú inmediato superior desde cada ventana + 							<ul style="list-style-type: none"> Incluir botón regresar en todas las ventanas Codificación de función que determine a que menú va debe 				

Figura 5-16: Panel de Tareas al final del día 8

5.2.1.2.1.9 Día 9

Sprint 1								
elementos de trabajo pendiente miembros del equipo								
	TAREAS PENDIENTES 2 h		EN CURSO 10 h	LISTO				
<ul style="list-style-type: none"> Reingeniería a base de datos del sistema + 				<ul style="list-style-type: none"> Modificación de modelo conceptual de base de datos Validar información registrada en Base de Datos 	<ul style="list-style-type: none"> Análisis modificación base de datos Pruebas de regresión 	<ul style="list-style-type: none"> Generación modelo físico base de datos Creación de clase utilitaria "BaseDatos" de conexión a base de datos 	<ul style="list-style-type: none"> Creación de tablas y nuevas relaciones en SNAP Creación de solución y estructura de proyectos 	<ul style="list-style-type: none"> Validación integridad Base de Datos
<ul style="list-style-type: none"> Selección y despliegue de información de manifiestos 12 h + 	<ul style="list-style-type: none"> Creación clase "CargaBELL" 1 	<ul style="list-style-type: none"> Creación de clase "CargaDAL" 1 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 10 	<ul style="list-style-type: none"> Agregar elementos gráficos para búsqueda de manifiestos a la 	<ul style="list-style-type: none"> Codificación GUI 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 	<ul style="list-style-type: none"> Creación de clase "CargaBELL" 	<ul style="list-style-type: none"> Creación de clase "UsuarioBELL"
<ul style="list-style-type: none"> Captura de peso marcado por balanza electrónica + 				<ul style="list-style-type: none"> Controlar que únicamente las categorías 	<ul style="list-style-type: none"> Integración de balanza electrónica con equipos 	<ul style="list-style-type: none"> Codificación de módulo que permita la comunicación entre la aplicación y el 	<ul style="list-style-type: none"> Agregar elementos gráficos para desplegar el peso marcado por la 	<ul style="list-style-type: none"> Codificación GUI
<ul style="list-style-type: none"> Despliegue de ubicaciones de entrega de paquetes en ciudad de destino + 				<ul style="list-style-type: none"> Agregar elementos gráficos para despliegue de ubicaciones de entrega de paquetes Creación de clase "EntregaDAL" 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 	<ul style="list-style-type: none"> Codificación GUI 	<ul style="list-style-type: none"> Creación de clase "EntregaBELL" 	<ul style="list-style-type: none"> Creación de clase "EntregaBELL"
<ul style="list-style-type: none"> Opción de regresar al menú inmediato superior desde cada ventana + 				<ul style="list-style-type: none"> Incluir botón regresar en todas las ventanas 	<ul style="list-style-type: none"> Codificación de función que determine a que menú se debe 			

Figura 5-17: Panel de Tareas al final del día 9

5.2.1.2.1.10 Día 10

Sprint 1					
elementos de trabajo pendiente miembros del equipo					
	TAREAS PENDIENTES	EN CURSO	LISTO		
<ul style="list-style-type: none"> Reingeniería a base de datos del sistema + 			<ul style="list-style-type: none"> Modificación de modelo conceptual de base de datos Validar información registrada en Base de Datos 	<ul style="list-style-type: none"> Análisis modificación base de datos Pruebas de regresión 	<ul style="list-style-type: none"> Generación modelo físico base de datos Creación de clase utilitaria "BaseDatos" de conexión a base de datos Creación de tablas y nuevas relaciones en SNAP Creación de solución y estructura de proyectos
<ul style="list-style-type: none"> Selección y despliegue de información de manifiestos + 			<ul style="list-style-type: none"> Agregar elementos gráficos para búsqueda de manifiestos a la Creación de clase "UsuarioBEL" 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos Creación clase "CargaBEL" 	<ul style="list-style-type: none"> Codificación GUI Creación de clase "CargaDAL" Codificación de capa de lógica de negocio Creación de clase "CargaBEL"
<ul style="list-style-type: none"> Captura de peso marcado por balanza electrónica + 			<ul style="list-style-type: none"> Controlar que únicamente las categorías seleccionadas se capturen 	<ul style="list-style-type: none"> Integración de balanza electrónica con equipos Creación de clase "EntregaDAL" 	<ul style="list-style-type: none"> Codificación de modulo que permita la comunicación entre la balanza y el sistema Agregar elementos gráficos para desplegar el peso medido por la balanza Codificación GUI Creación de clase "EntregaBEL" Creación de clase "EntregaBEL"
<ul style="list-style-type: none"> Despliegue de ubicaciones de entrega de paquetes en ciudad de destino + 			<ul style="list-style-type: none"> Agregar elementos gráficos para despliegue de ubicaciones por Creación de clase "EntregaDAL" 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 	<ul style="list-style-type: none"> Codificación GUI Creación de clase "EntregaBEL" Creación de clase "EntregaBEL"
<ul style="list-style-type: none"> Opción de regresar al menú inmediato superior desde cada ventana + 			<ul style="list-style-type: none"> Incluir botón regresar en todas las ventanas 	<ul style="list-style-type: none"> Codificación de función que determine a que menú se debe 	

Figura 5-18: Panel de Tareas al final del día 10

5.2.1.2.2 Burn-Down chart

En la Figura 5-19, se puede apreciar el Burn Down Chart, este gráfico evidencia el avance real realizado diariamente por el equipo de trabajo en oposición al avance recomendado.

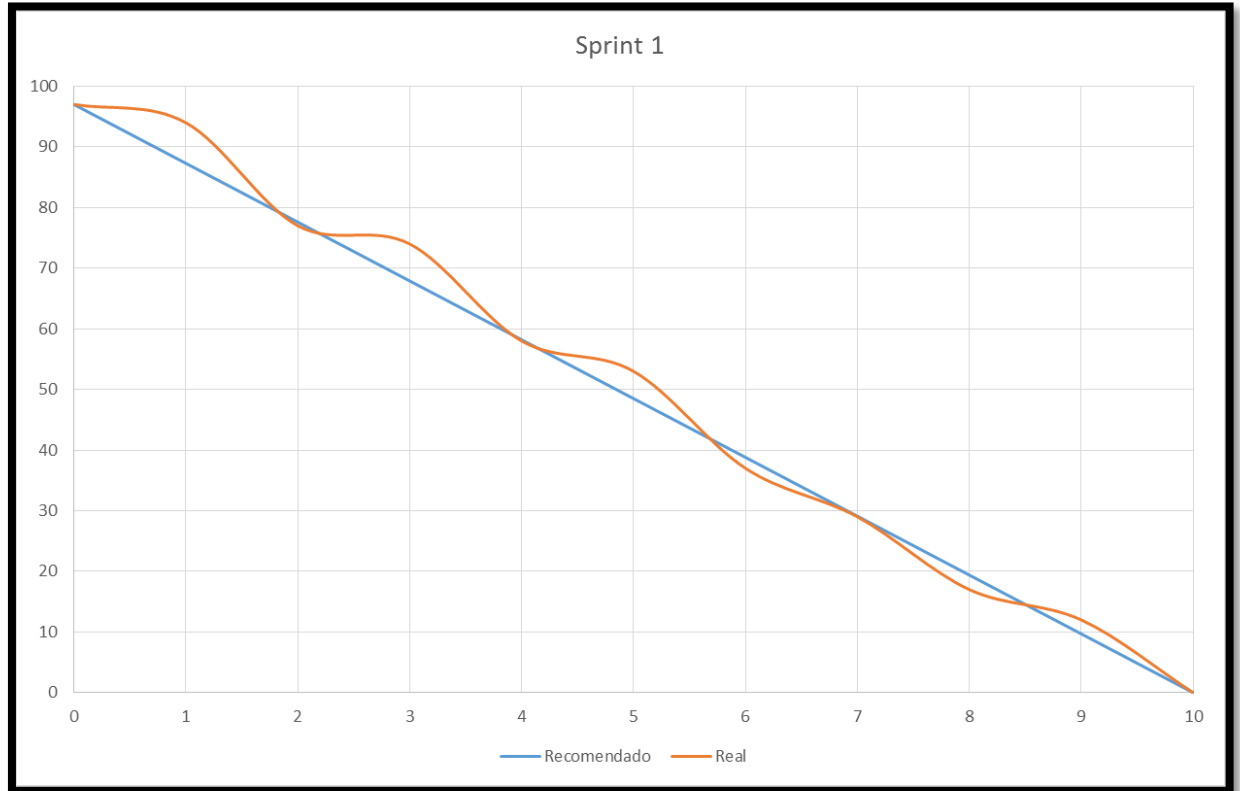


Figura 5-19: Burn down chart

5.2.1.2.3 Reporte de defectos

Al finalizar el sprint, se realizaron las pruebas respectivas del nuevo desarrollo. La Figura 5-20 muestra los defectos reportados.

The screenshot shows a web interface for "Consulta de defectos" (Defect Query). It displays a table with 2 selected items. The table has columns for ID, Tipo De El... (Type of Element), Título (Title), Asignado A (Assigned To), Estado De... (Status), and Etiquetas (Tags). The interface includes a search bar, a "Guardar consulta" (Save query) button, and various utility icons.

ID	Tipo De El...	Título	Asignado A	Estado De ...	Etiquetas
134	Error	Defecto: Error desplegando la lista con las ubicaciones		Nuevo	
136	Error	Defecto: Cuando el peso medido por la balanza electrónica tiene decimales no es ...		Nuevo	

Figura 5-20: Reporte de defectos inicial

El equipo de desarrollo procedió a realizar las verificaciones y correcciones respectivas, al finalizar ésto, se realizaron nuevamente las pruebas haciendo énfasis sobre los defectos previamente encontrados. La Figura 5-21 muestra los resultados de los defectos corregidos.

ID	Tipo De El.	Título	Asignado A	Estado De...	Etiquetas
134	Error	Defecto: Error desplegando la lista con las ubicaciones	Paul Lopez	Listo	
136	Error	Defecto: Cuando el peso medido por la balanza electrónica tiene decimales no es ...	Eduardo...	Listo	

Figura 5-21: Reporte de defectos final

5.2.1.2.4 Documento de la reunión para la revisión del Sprint

Después de la ejecución del Sprint, el equipo mantuvo una reunión en la cual se revisó el Sprint con el propósito de demostrar un incremento del producto trabajado al dueño del producto y a las personas interesadas.

En esta reunión se presentó una demostración en vivo mas no un reporte, dado que así lo indica la metodología escogida.

Después de la demostración, el dueño del producto revisó los compromisos hechos durante la planificación del Sprint y declaró los ítems considerados como “Hechos”.

Los ítems demostrados y aprobados en este Sprint fueron:

- Reingeniería a base de datos del sistema
- Selección y despliegue de información de manifiestos
- Captura de peso marcado por balanza electrónica
- Despliegue de ubicaciones de entrega de paquetes en ciudad de destino
- Opción de regresar al menú inmediato superior desde cada ventana

En la sección “5.2.1.3.1 Producto terminado” se puede observar imágenes que demuestran las funcionalidades del sistema entregadas en el presente Sprint.

5.2.1.2.5 Documento de la reunión de la retrospectiva del equipo

Al final del sprint, el equipo de desarrollo se reunió para evaluar el desempeño de las partes involucradas durante el desarrollo, el resultado se refleja en la Tabla 5-1.

Pregunta	Completamente de acuerdo	De acuerdo	En desacuerdo	Completamente en desacuerdo	NA
¿Los objetivos y metas estuvieron claros?	XX				
¿Las partes interesadas se involucraron en el proyecto?		X	X		
¿Existió una correcta retroalimentación por parte del cliente?			XX		
¿Las funcionalidades se definieron claramente?	X	X			
¿El equipo respondió adecuadamente a problemas surgidos durante el desarrollo?	XX				
¿Se planificaron correctamente las actividades de acuerdo al calendario?		XX			
¿El producto obtenido fue de acorde a las expectativas del cliente?	XX				

Tabla 5-1: Encuesta de la reunión de retrospectiva del equipo del sprint 1

5.2.1.3 Fase de Revisión

5.2.1.3.1 Producto terminado

De la Figura 5-22 a la Figura 5-26 se muestran muestras de capturas de pantallas de las funcionalidades logradas en la presente iteración.

CCTNUM	CCTNUJ1	CCTCIJ	CCTCI1	CCTCI2	CCTCAT	CCTPIE
3	1	UJO	CUE		3	0
6	2	UJO	CUE		3	0
8	2	UJO	SCY		2	0
9	1	UJO	OCC		1	0
22	1	UJO	LGQ		2	0
25	3	UJO	GYE		2	1
27	5	UJO	GYE		2	1
28	9	UJO	GYE		2	1
29	0	UJO	GYE		4	1
30	0	UJO	CUE		4	0
31	0	UJO	OCC		3	0
32	0	UJO	GYE		3	1
33	4	UJO	GYE		2	1
34	0	UJO	GYE		4	1
35	4	UJO	ESM		2	0
36	0	UJO	CUE		3	0
37	34	UJO	GYE		2	1
38	0	UJO	GYE		4	0
39	1	UJO	GYE		2	0
40	2	UJO	GYE		2	0
41	0	UJO	LOH		3	0
42	17	UJO	CUE		2	0
43	10	UJO	OCC		2	0
44	4	UJO	GYE		2	0
45	1	UJO	LOH		2	0
46	47	UJO	GPS		2	0
47	1	UJO	OCC		2	0
48	12	UJO	GYE		2	0
49	0	UJO	GYE		4	0
50	4	UJO	GYE		2	0
51	0	UJO	GYE		4	0
52	1	UJO	GYE		2	0
53	0	UJO	GYE		4	0
54	1	UJO	GYE		2	0
55	0	UJO	GYE		3	0
56	0	UJO	GYE		3	0
57	0	UJO	CUE		4	0
58	1	UJO	GYE		2	0
59	1	UJO	GYE		2	0
60	0	UJO	GYE		4	0
61	5	UJO	GYE		2	0

Figura 5-22: Extracto de una de las nuevas tablas integradas "Taquilla" y posibilidad de realizar el CRUD

Manifiesto	Ubicación	Origen	Destino	Fecha	Vuelo
523725	AEROPUERTO	GYE	UIO	2014100	304

Figura 5-23: Selección y despliegue de información de Manifiestos

SISTEMA DE CARGA TAME
RECEPCION

USUARIO
TSCA34

tame CARGO

Categoría: No Perecibles

Origen: UIO Destino: GUAYAQUIL

Peso: 4

Piezas: 1

Figura 5-24: Captura automática de peso marcado por la balanza

Ubicación

- AEROPUERTO
- VALIJA INTERNA
- OFICINAS TAME
- AEROPUERTO
- TAME EL EJIDO
- TAME TUMBACO
- TAME EL RECREO
- TAME OLIMPICO**
- AMERICA
- DOMICILIO URBANO
- DOMICILIO PERIFERICO
- GSA SAN RAFAEL

Figura 5-25: Despliegue de ubicaciones de entrega de paquetes en ciudad de destino

Manifiesto	Ubicación	Origen	Destino	Fecha	Vuelo
523730	VALIJA INTERNA	LGQ	UIO	201410	202

MENU

Figura 5-26: Botón que permite regresar al menú inmediato superior

5.2.1.3.2 Notas de la versión

Funcionalidades implementadas:

- Reingeniería a base de datos del sistema
- Selección y despliegue de información de manifiestos
- Captura de peso marcado por balanza electrónica
- Despliegue de ubicaciones de entrega de paquetes en ciudad de destino

- Opción de regresar al menú inmediato superior desde cada ventana

Defectos corregidos:

- Error desplegando la lista con las ubicaciones de los manifiestos
- Error en el registro del peso marcado por la balanza electrónica al contener decimales

5.2.1.3.3 *Manual de Usuario*

El Manual de Usuario final integrado se lo puede encontrar en el Anexo A.

5.2.1.3.4 *Programa de mejora*

En la Tabla 5-2 se establecen las acciones a implementar para corregir las oportunidades de mejora identificadas.

Programa de mejora			
Acciones de mejora	Condiciones de Ejecución	Tiempo de Ejecución	Recursos Necesarios
Mantener actualizado el gestor de base de datos DB2	Asegurarse que el driver de conexión sea compatible con el framework	Mediano plazo	Humanos, financieros y materiales
Adaptar código para compatibilidad con HTML5	Capacitación a desarrolladores	Mediano plazo	Humanos y financieros
Ejecución de un mantenimiento regular preventivo de la balanza electrónica	Inclusión del mantenimiento en partidas presupuestarias anuales	Mediano plazo	Humanos y financieros

Tabla 5-2: Programa de mejora del sprint 1

5.2.2 *Sprint 2*

5.2.2.1 *Fase de Planificación*

5.2.2.1.1 *Planificación del Sprint*

En la jornada de trabajo previa al inicio del Sprint 2 se especificaron las historias de usuario que deberán estar concluidas al final de esta iteración y el objetivo de la iteración.

5.2.2.1.1.1 *Historias de Usuario incluidas en el Sprint*

Las historias de usuario incluidas en el presente Sprint se indican en la Figura 5-27.

Sprint 2	
elementos de trabajo pendiente	miembros del equipo
TAREAS PENDIENTES	
▸ Ingreso de información del paquete según categoría	
▸ Categorizar paquetes en módulo de chequeo	
▸ Control de usuarios con acceso al sistema	
▸ Despliegue de todos los destinos nacionales para envío de paquetes	
▸ Despliegue de información relacionada al usuario en cada ventana	

Figura 5-27: Historias de Usuario incluidas en el Sprint 2

5.2.2.1.1.2 Objetivo del Sprint

Demostrar la implementación de las nuevas funcionalidades de acuerdo a las historias de usuario el 27 de Julio de 2012.

5.2.2.1.2 Definición del Sprint Backlog

El propósito de este proceso se definió en la sección 5.2.1.1.2.

En la Figura 5-28 se detallan todas las tareas asociadas a las historias de usuario asignadas a la iteración 2.

Sprint 2							
elementos de trabajo pendiente miembros del equipo							
	TAREAS PENDIENTES 97 h					EN CURSO	LISTO
Ingreso de información del paquete según categoría 30 h +	Agregar elementos gráficos para ingreso de información del paquete según categoría 5	Codificación capa de acceso a datos 8	Personalización de campos requeridos del paquete según la categoría 7	Codificación GUI 5	Codificación de capa de lógica de negocio 5		
Categorizar paquetes en módulo de chequeo 29 h +	Agregar elementos gráficos para filtrar paquetes por categoría 6	Codificación GUI 8	Codificación de capa de lógica de negocio 6	Codificación de capa de acceso a datos 9			
Control de usuarios con acceso al sistema 25 h +	Agregar elementos gráficos para ventana de inicio de sesión 4 Creación de clase "UsuarioBLL" 1	Codificación de capa de acceso a datos 8	Codificación de capa de lógica de negocio 8	Codificación de eventos para elementos de interfaz gráfica 3	Creación de clase "UsuarioDAL" 1		
Despliegue de todos los destinos nacionales para envío de paquetes 8 h +	Creación de lista desplegable para destinos 2	Incluir lista desplegable en ventanas del módulo de reserva 2	Cargar lista desplegable con destinos posibles según región 2	Codificación de capa de lógica de negocio 2			
Despliegue de información relacionada al usuario en cada ventana 5 h +	Codificación de capa de acceso a datos 2	Diseño de elementos de interfaz gráfica para desplegar información del 2	Codificación de eventos para elementos de interfaz gráfica 1				

Figura 5-28: Sprint 2 Backlog

5.2.2.1.3 Estimación del esfuerzo del Sprint

De la Figura 5-29 a la Figura 5-33 se detallan las tareas necesarias para desarrollar la funcionalidad requerida por las historias de usuario.

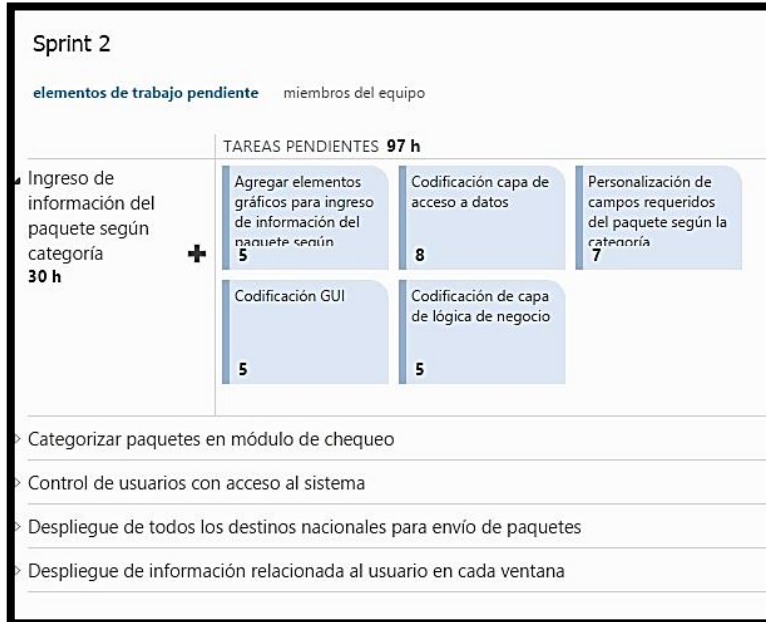


Figura 5-29: Esfuerzo de cada tarea para la primera historia de usuario

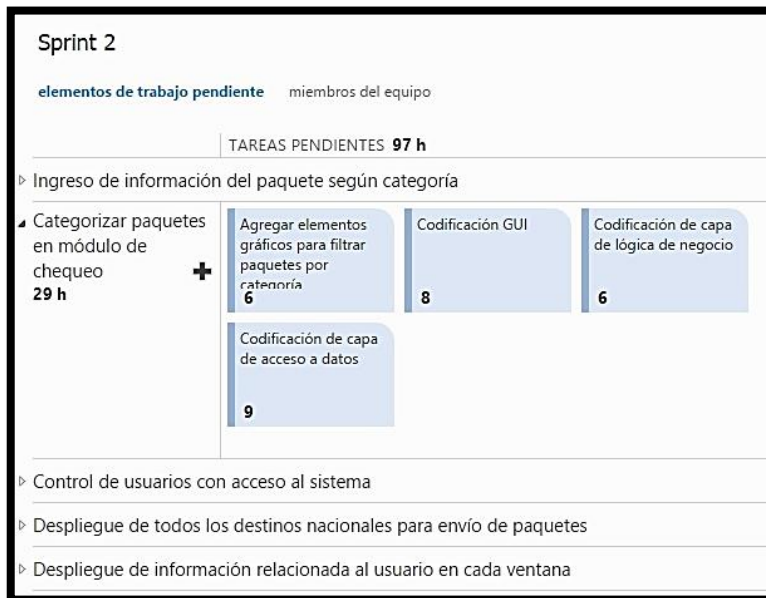


Figura 5-30: Esfuerzo de cada tarea para la segunda historia de usuario

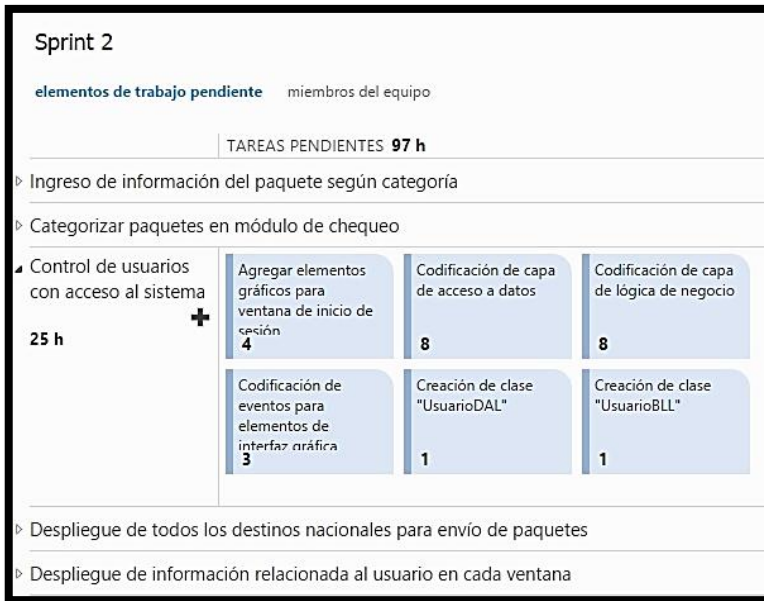


Figura 5-31: Esfuerzo de cada tarea para la tercera historia de usuario

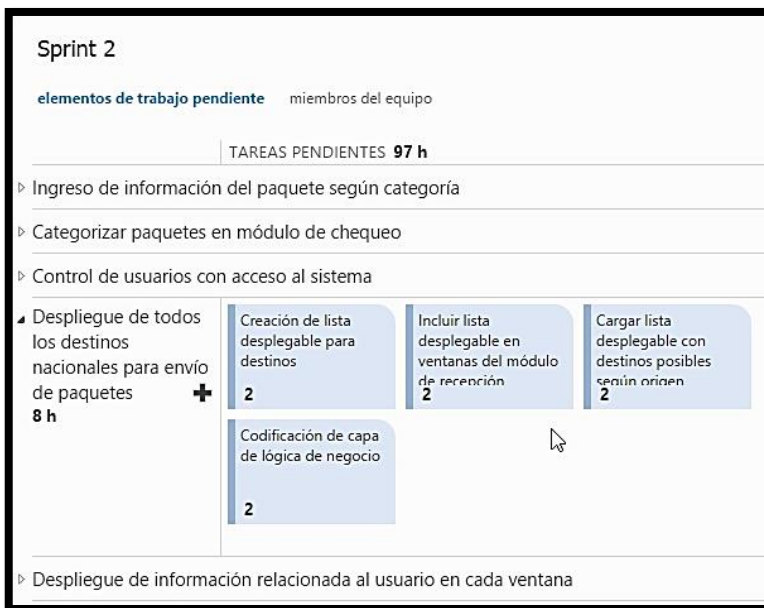


Figura 5-32: Esfuerzo de cada tarea para la cuarta historia de usuario

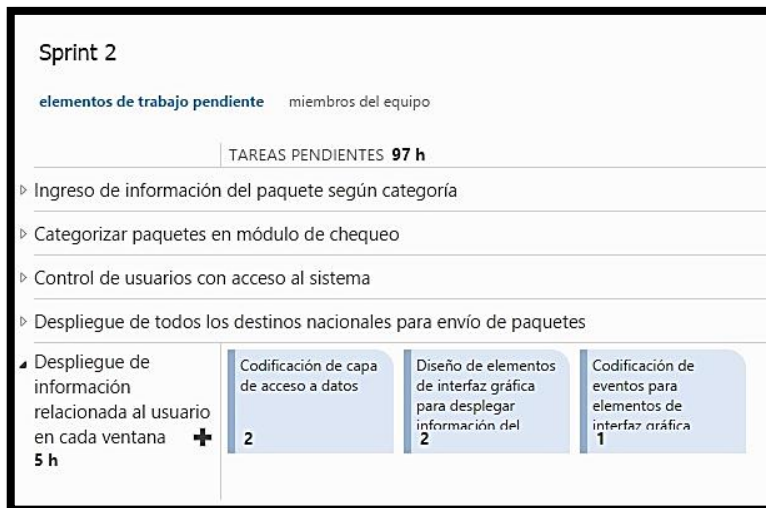


Figura 5-33: Esfuerzo de cada tarea para la quinta historia de usuario

5.2.2.2 Fase de Seguimiento

5.2.2.2.1 Desarrollo del Sprint

La reunión de seguimiento del sprint se lleva a cabo diariamente a primera hora entre los integrantes del equipo de trabajo, dónde se comenta el trabajo realizado por cada integrante desde la anterior reunión, así como los problemas encontrados y el trabajo a realizar hasta la siguiente reunión.

En estas reuniones diarias, se da el seguimiento necesario para pronosticar si se cumplirá el trabajo comprometido en las fechas establecidas.

De la Figura 5-34 a la Figura 5-43 se describe el avance realizado cada día por el equipo de trabajo.

5.2.2.2.1.1 Día 11

Sprint 2						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 85 h				EN CURSO 3 h	LISTO
Ingreso de información del paquete según categoría 25 h +	Codificación capa de acceso a datos 8	Personalización de campos requeridos del paquete según la categoría 7	Codificación GUI 5	Codificación de capa de lógica de negocio 5		Agregar elementos gráficos para ingreso de información del paquete según
Categorizar paquetes en módulo de chequeo 29 h +	Agregar elementos gráficos para filtrar paquetes por categoría 6	Codificación GUI 8	Codificación de capa de lógica de negocio 6	Codificación de capa de acceso a datos 9		
Control de usuarios con acceso al sistema 21 h +	Codificación de capa de acceso a datos 8	Codificación de capa de lógica de negocio 8	Creación de clase "UsuarioDAL" 1	Creación de clase "UsuarioBLL" 1	Codificación GUI 3	Agregar elementos gráficos para ventana de inicio de sesión
Despliegue de todos los destinos nacionales para envío de paquetes 8 h +	Creación de lista desplegable para destinos 2	Incluir lista desplegable en ventanas del módulo de recepción 2	Cargar lista desplegable con destinos posibles según origen 2	Codificación de capa de lógica de negocio 2		
Despliegue de información relacionada al usuario en cada ventana 5 h +	Codificación de capa de acceso a datos 2	Diseño de elementos de interfaz gráfica para desplegar información del 2	Codificación de eventos para elementos de interfaz gráfica 1			

Figura 5-34: Panel de Tareas al final del día 11

5.2.2.2.1.2 Día 12

Sprint 2						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 71 h			EN CURSO 8 h	LISTO	
Ingreso de información del paquete según categoría 20 h	Codificación capa de acceso a datos 8	Personalización de campos requeridos del paquete según la categoría 7	Codificación de capa de lógica de negocio 5		Agregar elementos gráficos para ingreso de información del paquete según categoría	Codificación GUI
Categorizar paquetes en módulo de chequeo 29 h	Agregar elementos gráficos para filtrar paquetes por categoría 6	Codificación GUI 8	Codificación de capa de lógica de negocio 6	Codificación de capa de acceso a datos 9		
Control de usuarios con acceso al sistema 17 h	Codificación de capa de lógica de negocio 8	Creación de clase "UsuarioBLL" 1		Codificación de capa de acceso a datos 8	Agregar elementos gráficos para ventana de inicio de sesión	Codificación GUI Creación de clase "UsuarioDAL"
Despliegue de todos los destinos nacionales para envío de paquetes 8 h	Creación de lista desplegable para destinos 2	Incluir lista desplegable en ventanas del módulo de creación 2	Cargar lista desplegable con destinos posibles según origen 2	Codificación de capa de lógica de negocio 2		
Despliegue de información relacionada al usuario en cada ventana 5 h	Codificación de capa de acceso a datos 2	Diseño de elementos de interfaz gráfica para desplegar información del usuario 2	Codificación de eventos para elementos de interfaz gráfica 1			

Figura 5-35: Panel de Tareas al final del día 12

5.2.2.2.1.3 Día 13

Sprint 2						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 64 h				EN CURSO 15 h	LISTO
<ul style="list-style-type: none"> Ingreso de información del paquete según categoría 20 h + 	<ul style="list-style-type: none"> Codificación capa de acceso a datos 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 5 			<ul style="list-style-type: none"> Personalización de campos requeridos del paquete según la categoría 7 	<ul style="list-style-type: none"> Agregar elementos gráficos para ingreso de información del paquete según categoría Codificación GUI
<ul style="list-style-type: none"> Categorizar paquetes en módulo de chequeo 29 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para filtrar paquetes por categoría 6 	<ul style="list-style-type: none"> Codificación GUI 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 6 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 9 		
<ul style="list-style-type: none"> Control de usuarios con acceso al sistema 17 h + 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Creación de clase "UsuarioBLL" 1 			<ul style="list-style-type: none"> Codificación de capa de acceso a datos 8 	<ul style="list-style-type: none"> Agregar elementos gráficos para ventana de inicio de sesión Codificación GUI Creación de clase "UsuarioDAL"
<ul style="list-style-type: none"> Despliegue de todos los destinos nacionales para envío de paquetes 8 h + 	<ul style="list-style-type: none"> Creación de lista desplegable para destinos 2 	<ul style="list-style-type: none"> Incluir lista desplegable en ventanas del módulo de selección 2 	<ul style="list-style-type: none"> Cargar lista desplegable con destinos posibles según origen 2 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 2 		
<ul style="list-style-type: none"> Despliegue de información relacionada al usuario en cada ventana 5 h + 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 2 	<ul style="list-style-type: none"> Diseño de elementos de interfaz gráfica para desplegar información del 2 	<ul style="list-style-type: none"> Codificación de eventos para elementos de interfaz gráfica 1 			

Figura 5-36: Panel de Tareas al final del día 13

5.2.2.2.1.4 Día 14

Sprint 2						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 50 h				EN CURSO 13 h	LISTO
<ul style="list-style-type: none"> Ingreso de información del paquete según categoría 13 h + 	<ul style="list-style-type: none"> Codificación capa de acceso a datos 8 				<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 5 	<ul style="list-style-type: none"> Agregar elementos gráficos para ingreso de información del paquete según categoría Personalización de campos requeridos del paquete según la categoría Codificación GUI
<ul style="list-style-type: none"> Categorizar paquetes en módulo de chequeo 29 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para filtrar paquetes por categoría 6 	<ul style="list-style-type: none"> Codificación GUI 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 6 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 9 		
<ul style="list-style-type: none"> Control de usuarios con acceso al sistema 8 h + 					<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Agregar elementos gráficos para ventana de inicio de sesión Codificación de capa de acceso a datos Codificación GUI Creación de clase "UsuarioDAL" Creación de clase "UsuarioBILL"
<ul style="list-style-type: none"> Despliegue de todos los destinos nacionales para envío de paquetes 8 h + 	<ul style="list-style-type: none"> Creación de lista desplegable para destinos 2 	<ul style="list-style-type: none"> Incluir lista desplegable en ventanas del módulo de recepción 2 	<ul style="list-style-type: none"> Cargar lista desplegable con destinos posibles según origen 2 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 2 		
<ul style="list-style-type: none"> Despliegue de información relacionada al usuario en cada ventana 5 h + 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 2 	<ul style="list-style-type: none"> Diseño de elementos de interfaz gráfica para desplegar información del 2 	<ul style="list-style-type: none"> Codificación de eventos para elementos de interfaz gráfica 1 			

Figura 5-37: Panel de Tareas al final del día 14

5.2.2.2.1.5 Día 15

Sprint 2									
elementos de trabajo pendiente miembros del equipo									
	TAREAS PENDIENTES 42 h				EN CURSO 8 h	LISTO			
<ul style="list-style-type: none"> Ingreso de información del paquete según categoría 8 h + 					<ul style="list-style-type: none"> Codificación capa de acceso a datos 8 	<ul style="list-style-type: none"> Agregar elementos gráficos para ingreso de información del paquete según la categoría 	<ul style="list-style-type: none"> Personalización de campos requeridos del paquete según la categoría 	<ul style="list-style-type: none"> Codificación GUI 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Categorizar paquetes en módulo de chequeo 29 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para filtrar paquetes por categoría 6 	<ul style="list-style-type: none"> Codificación GUI 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 6 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 9 					
<ul style="list-style-type: none"> Control de usuarios con acceso al sistema + 						<ul style="list-style-type: none"> Agregar elementos gráficos para ventana de inicio de sesión 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 	<ul style="list-style-type: none"> Codificación GUI
						<ul style="list-style-type: none"> Creación de clase "UsuarioDAL" 	<ul style="list-style-type: none"> Creación de clase "UsuarioBL" 		
<ul style="list-style-type: none"> Despliegue de todos los destinos nacionales para envío de paquetes 8 h + 	<ul style="list-style-type: none"> Creación de lista desplegable para destinos 2 	<ul style="list-style-type: none"> Incluir lista desplegable en ventanas del módulo de recepción 2 	<ul style="list-style-type: none"> Cargar lista desplegable con destinos posibles según origen 2 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 2 					
<ul style="list-style-type: none"> Despliegue de información relacionada al usuario en cada ventana 5 h + 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 2 	<ul style="list-style-type: none"> Diseño de elementos de interfaz gráfica para desplegar información del 2 	<ul style="list-style-type: none"> Codificación de eventos para elementos de interfaz gráfica 1 						

Figura 5-38: Panel de Tareas al final del día 15

5.2.2.2.1.6 Día 16

Sprint 2					
elementos de trabajo pendiente miembros del equipo					
	TAREAS PENDIENTES 36 h			EN CURSO 6 h	LISTO
<ul style="list-style-type: none"> Ingreso de información del paquete según categoría + 					<ul style="list-style-type: none"> Agregar elementos gráficos para ingreso de información del paquete según categoría Codificación capa de acceso a datos Personalización de campos requeridos del paquete según la categoría Codificación GUI Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Categorizar paquetes en módulo de chequeo 29 h + 	<ul style="list-style-type: none"> Codificación GUI 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 6 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 9 	<ul style="list-style-type: none"> Agregar elementos gráficos para filtrar paquetes por categoría 6 	
<ul style="list-style-type: none"> Control de usuarios con acceso al sistema + 					<ul style="list-style-type: none"> Agregar elementos gráficos para ventana de inicio de sesión Codificación de capa de acceso a datos Codificación de capa de lógica de negocio Codificación GUI Creación de clase "UsuarioDAL" Creación de clase "UsuarioBLL"
<ul style="list-style-type: none"> Despliegue de todos los destinos nacionales para envío de paquetes 8 h + 	<ul style="list-style-type: none"> Creación de lista desplegable para destinos 2 	<ul style="list-style-type: none"> Incluir lista desplegable en ventanas del módulo de selección 2 	<ul style="list-style-type: none"> Cargar lista desplegable con destinos posibles según origen 2 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 2 	
<ul style="list-style-type: none"> Despliegue de información relacionada al usuario en cada ventana 5 h + 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 2 	<ul style="list-style-type: none"> Diseño de elementos de interfaz gráfica para desplegar información del 2 	<ul style="list-style-type: none"> Codificación de eventos para elementos de interfaz gráfica 1 		

Figura 5-39: Panel de Tareas al final del día 16

5.2.2.2.1.7 Día 17

Sprint 2						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 23 h				EN CURSO 8 h	LISTO
Ingreso de información del paquete según categoría +						Agregar elementos gráficos para ingreso de información del paquete según categoría Codificación capa de acceso a datos Personalización de campos requeridos del paquete según la categoría Codificación GUI Codificación de capa de lógica de negocio
Categorizar paquetes en módulo de chequeo 23 h +	Codificación de capa de lógica de negocio 6	Codificación de capa de acceso a datos 9			Codificación GUI 8	Agregar elementos gráficos para filtrar paquetes por categoría
Control de usuarios con acceso al sistema +						Agregar elementos gráficos para ventana de inicio de sesión Codificación de capa de acceso a datos Codificación de capa de lógica de negocio Codificación GUI Creación de clase "UsuarioDAL"
Despliegue de todos los destinos nacionales para envío de paquetes 8 h +	Creación de lista desplegable para destinos 2	Incluir lista desplegable en ventanas del módulo de reservación 2	Cargar lista desplegable con destinos posibles según región 2	Codificación de capa de lógica de negocio 2		
Despliegue de información relacionada al usuario en cada ventana +						Codificación de capa de acceso a datos Diseño de elementos de interfaz gráfica para desplegar información del usuario Codificación de eventos para elementos de interfaz gráfica

Figura 5-40: Panel de Tareas al final del día 17

5.2.2.2.1.8 Día 18

Sprint 2							
elementos de trabajo pendiente miembros del equipo							
	TAREAS PENDIENTES 8 h	EN CURSO 11 h	LISTO				
Ingreso de información del paquete según categoría +			Agregar elementos gráficos para ingreso de información del paquete según categoría	Codificación capa de acceso a datos	Personalización de campos requeridos del paquete según la categoría	Codificación GUI	Codificación de capa de lógica de negocio
Categorizar paquetes en módulo de chequeo 15 h +	Codificación de capa de lógica de negocio 6	Codificación de capa de acceso a datos 9	Agregar elementos gráficos para filtrar paquetes por categoría	Codificación GUI			
Control de usuarios con acceso al sistema +			Agregar elementos gráficos para ventana de inicio de sesión	Codificación de capa de acceso a datos	Codificación de capa de lógica de negocio	Codificación GUI	Creación de clase "UsuarioDAL"
Despliegue de todos los destinos nacionales para envío de paquetes 4 h +	Codificación de capa de lógica de negocio 2	Cargar lista desplegable con destinos posibles 2	Creación de clase "UsuarioBLL"	Creación de lista desplegable para destinos	Incluir lista desplegable en ventanas del módulo de envío		
Despliegue de información relacionada al usuario en cada ventana +			Codificación de capa de acceso a datos	Diseño de elementos de interfaz gráfica para desplegar información del usuario	Codificación de eventos para elementos de interfaz gráfica		

Figura 5-41: Panel de Tareas al final del día 18

5.2.2.2.1.9 Día 19

Sprint 2								
elementos de trabajo pendiente miembros del equipo								
	TAREAS PENDIENTES	EN CURSO 15 h		LISTO				
• Ingreso de información del paquete según categoría +				Agregar elementos gráficos para ingreso de información del paquete según la categoría.	Codificación capa de acceso a datos	Personalización de campos requeridos del paquete según la categoría.	Codificación GUI	Codificación de capa de lógica de negocio
• Categorizar paquetes en módulo de chequeo 15 h +		Codificación de capa de lógica de negocio 6	Codificación de capa de acceso a datos 9	Agregar elementos gráficos para filtrar paquetes por categoría.	Codificación GUI			
• Control de usuarios con acceso al sistema +				Agregar elementos gráficos para ventana de inicio de sesión.	Codificación de capa de acceso a datos	Codificación de capa de lógica de negocio	Codificación GUI	Creación de clase "UsuarioDAL"
• Despliegue de todos los destinos nacionales para envío de paquetes +				Creación de clase "UsuarioBLL"				
• Despliegue de información relacionada al usuario en cada ventana +				Creación de lista desplegable para destinos.	Incluir lista desplegable en ventanas del módulo de reserva.	Cargar lista desplegable con destinos posibles según región.	Codificación de capa de lógica de negocio	
				Codificación de capa de acceso a datos	Diseño de elementos de interfaz gráfica para desplegar información del usuario.	Codificación de eventos para elementos de interfaz gráfica.		

Figura 5-42: Panel de Tareas al final del día 19

5.2.2.2.1.10 Día 20

Sprint 2			
elementos de trabajo pendiente		miembros del equipo	
	TAREAS PENDIENTES	EN CURSO	LISTO
<ul style="list-style-type: none"> Ingreso de información del paquete según categoría + 			<ul style="list-style-type: none"> Agregar elementos gráficos para ingreso de información del paquete según categoría Codificación capa de acceso a datos Personalización de campos requeridos del paquete según la categoría Codificación GUI Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Categorizar paquetes en módulo de chequeo + 			<ul style="list-style-type: none"> Agregar elementos gráficos para filtrar paquetes por categoría Codificación GUI Codificación de capa de lógica de negocio Codificación de capa de acceso a datos
<ul style="list-style-type: none"> Control de usuarios con acceso al sistema + 			<ul style="list-style-type: none"> Agregar elementos gráficos para ventana de inicio de sesión Codificación de capa de acceso a datos Codificación de capa de lógica de negocio Codificación GUI Creación de clase "UsuarioDAL" Creación de clase "UsuarioBLL"
<ul style="list-style-type: none"> Despliegue de todos los destinos nacionales para envío de paquetes + 			<ul style="list-style-type: none"> Creación de lista desplegable para destinos Incluir lista desplegable en ventanas del módulo de recepción Cargar lista desplegable con destinos posibles según origen Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Despliegue de información relacionada al usuario en cada ventana + 			<ul style="list-style-type: none"> Codificación de capa de acceso a datos Diseño de elementos de interfaz gráfica para desplegar información del Codificación de eventos para elementos de interfaz gráfica

Figura 5-43: Panel de Tareas al final del día 20

5.2.2.2 Burn-Down chart

El significado de este artefacto se describe en la sección 5.2.1.2.2.

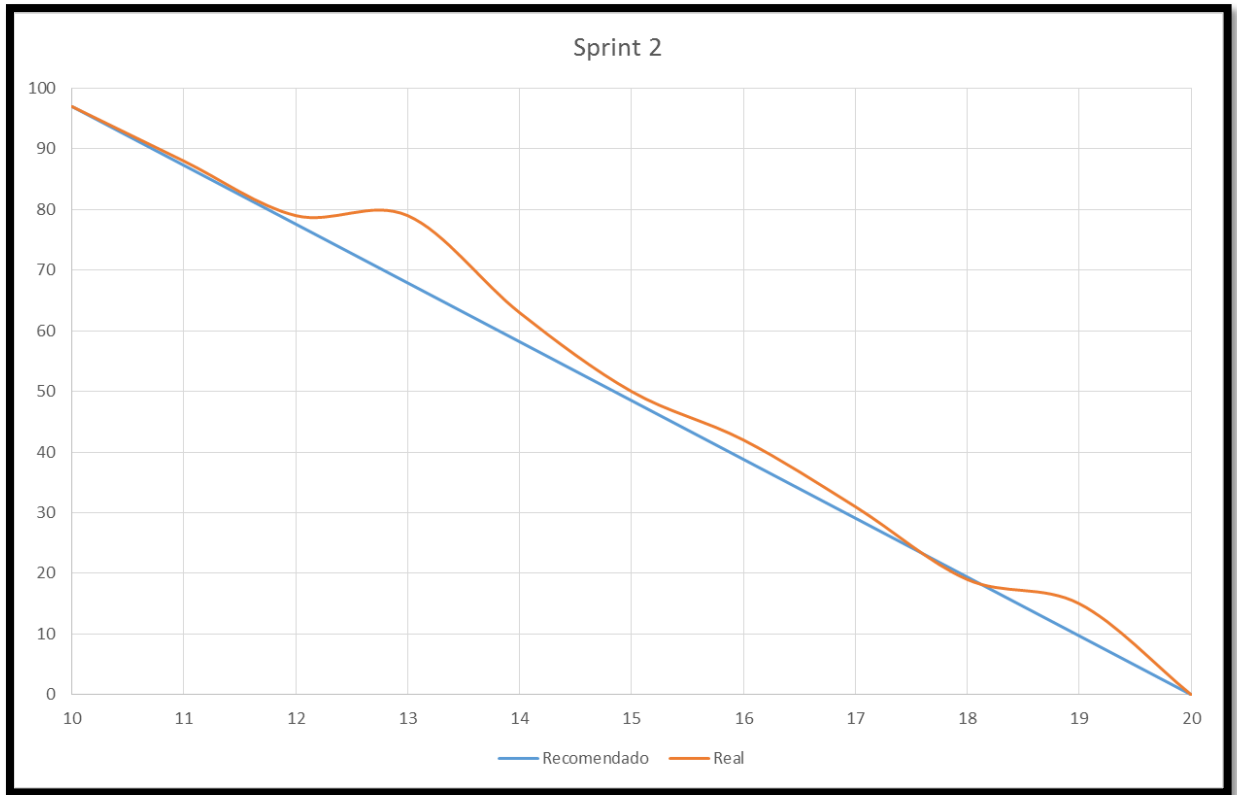


Figura 5-44: Burn down chart

5.2.2.3 Reporte de defectos

Al finalizar el sprint, se realizaron las pruebas respectivas del nuevo desarrollo. La Figura 5-45 muestra los defectos reportados.

The screenshot shows a web interface for "Consulta de defectos" (Defect Query). It displays a table with 3 elements of work, 1 of which is selected. The table has columns for ID, Tipo De El. (Error), Título, Asignado A, Estado De..., and Etiquetas. The selected row (ID 134) has the title "Defecto: Error desplegando la lista con las ubicaciones" and is assigned to Paul Lopez, with a status of "Listo".

ID	Tipo De El...	Título	Asignado A	Estado De ...	Etiquetas
134	Error	Defecto: Error desplegando la lista con las ubicaciones	Paul Lopez	Listo	
136	Error	Defecto: Cuando el peso medido por la balanza electrónica tiene decimales no es ...	Eduardo...	Listo	
137	Error	Defecto: No se despliegan todos los destinos nacionales en la lista		Nuevo	

Figura 5-45: Reporte de defectos inicial

El equipo de desarrollo procedió a realizar las verificaciones y correcciones respectivas, al finalizar esto, se realizaron nuevamente las pruebas haciendo énfasis sobre los defectos previamente encontrados. La Figura 5-46 muestra los resultados de los defectos corregidos.

ID	Tipo De El...	Título	Asignado A	Estado De ...	Etiquetas
134	Error	Defecto: Error desplegando la lista con las ubicaciones	Paul Lopez	Listo	
136	Error	Defecto: Cuando el peso medido por la balanza electrónica tiene decimales no es ...	Eduardo...	Listo	
137	Error	Defecto: No se despliegan todos los destinos nacionales en la lista	Eduardo...	Listo	

Figura 5-46: Reporte de defectos final

5.2.2.2.4 Documento de la reunión para la revisión del Sprint

El propósito de este proceso se encuentra detallado en la sección 5.2.1.2.4.

Los ítems demostrados y aprobados en este Sprint fueron:

- Ingreso de información del paquete según categoría
- Categorizar paquetes en módulo de chequeo
- Control de usuarios con acceso al sistema
- Despliegue de todos los destinos nacionales para envío de paquetes
- Despliegue de información relacionada al usuario en cada ventana

En la sección “5.2.2.3.1 Producto terminado” se puede observar imágenes que demuestran las funcionalidades del sistema entregadas en el presente Sprint.

5.2.2.2.5 Documento de la reunión de la retrospectiva del equipo

Al final del sprint, el equipo de desarrollo se reunió para evaluar el desempeño de las partes involucradas durante el desarrollo, el resultado se refleja en la Tabla 5-3.

Pregunta	Completamente de acuerdo	De acuerdo	En desacuerdo	Completamente en desacuerdo	NA
¿Los objetivos y metas estuvieron claros?	X	X			
¿Las partes interesadas se involucraron en el proyecto?		XX			
¿Existió una correcta retroalimentación por parte del cliente?	X	X			
¿Las funcionalidades se definieron claramente?		XX			
¿El equipo respondió adecuadamente a problemas surgidos durante el desarrollo?		X	X		
¿Se planificaron correctamente las actividades de acuerdo al calendario?	X		X		
¿El producto obtenido fue de acorde a las expectativas del cliente?		XX			

Tabla 5-3: Encuesta de la reunión de retrospectiva del equipo del sprint 2

5.2.2.3 Fase de Revisión

5.2.2.3.1 Producto terminado

De la Figura 5-47 a la Figura 5-51, se presentan capturas de pantallas de las funcionalidades logradas en la presente iteración.

The screenshot shows a web interface for 'SISTEMA DE CARGA TAME RECEPCION'. At the top left is a logo with a world map and the text 'tame'. To the right, it says 'USUARIO TSCA34'. Below the logo is a 'CARGO' section with the heading 'Categoría: Perecibles'. The form includes fields for 'Origen: UIO', 'Destino: GUAYAQUIL', 'Peso:', 'Piezas:', and 'Cubicaje: 0 0 0'. There is also a 'Tipo Cubicaje:' dropdown menu. At the bottom, there are three blue buttons: 'Imprimir', 'Cancelar', and 'Regresar'.

Figura 5-47: Ingreso de información del paquete según categoría

The screenshot shows a form with several dropdown menus and buttons. The 'Manifiesto' dropdown is set to '523730'. The 'Ubicación' dropdown is set to 'VALIJA INTERNA'. The 'Origen' dropdown is set to 'LGQ', 'Destino' to 'UIO', 'Fecha' to '2014100', and 'Vuelo' to '202'. Below these are five buttons: 'S. Pequeño', 'S. Grande', 'Mínimos', '>=1KG', and 'MENU'. A red rectangle highlights the first four buttons.

Figura 5-48: Categorización de paquetes en módulo de Chequeo

The screenshot shows a login interface titled 'ACCESO AL SISTEMA'. It has two input fields: 'USUARIO:' and 'CONTRASEÑA:'. Below the fields are two buttons: 'Ingresar' and 'Salir'.

Figura 5-49: Control de usuarios con acceso al sistema

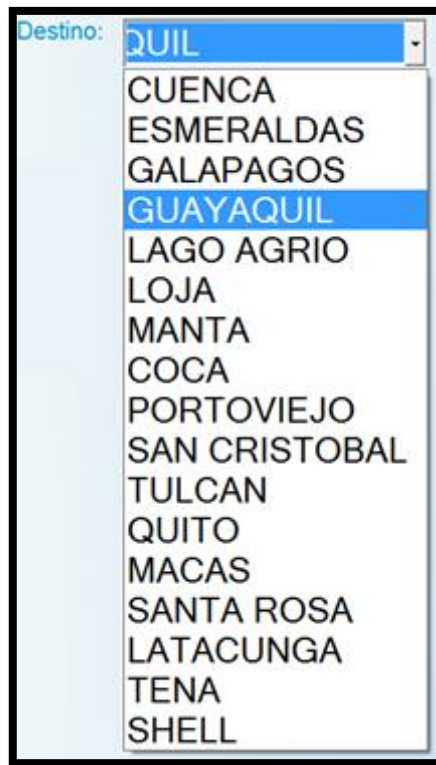


Figura 5-50: Despliegue de todos los destinos nacionales para envío de paquetes

DESPACHO		
Usuario	Ciudad	Oficina
TSCA34	UIO	79990035

Figura 5-51: Despliegue de información relacionada al usuario en ventana despacho

5.2.2.3.2 Notas de la versión

Funcionalidades implementadas:

- Ingreso de información del paquete según categoría
- Categorizar paquetes en módulo de chequeo
- Control de usuarios con acceso al sistema
- Despliegue de todos los destinos nacionales para envío de paquetes
- Despliegue de información relacionada al usuario en cada ventana

Defectos corregidos:

- No se despliegan todos los destinos nacionales en la lista de selección de destino

5.2.2.3.3 Manual de Usuario

El Manual de Usuario final integrado se lo puede encontrar en el Anexo A.

5.2.2.3.4 Programa de mejora

Programa de mejora			
Acciones de mejora	Condiciones de Ejecución	Tiempo de Ejecución	Recursos Necesarios
Realizar talleres de capacitación a los agentes de carga regularmente	2 horas por 3 días	Corto plazo	Humanos y materiales
Monitoreo regular de los canales de comunicación por parte del departamento de redes	Dar prioridad al monitoreo en horas pico de trabajo y a las ciudades principales	Corto plazo	Humanos y materiales

Tabla 5-4: Programa de mejora del sprint 2

5.2.3 Sprint 3

5.2.3.1 Fase de Planificación

5.2.3.1.1 Planificación del Sprint

En la jornada de trabajo previa al inicio del Sprint 3 se especificaron las historias de usuario que deberán estar concluidas al final de esta iteración y el objetivo de la iteración.

5.2.3.1.1.1 Historias de Usuario incluidas en el Sprint

Las historias de usuario incluidas en el presente Sprint se indican en la Figura 5-52.

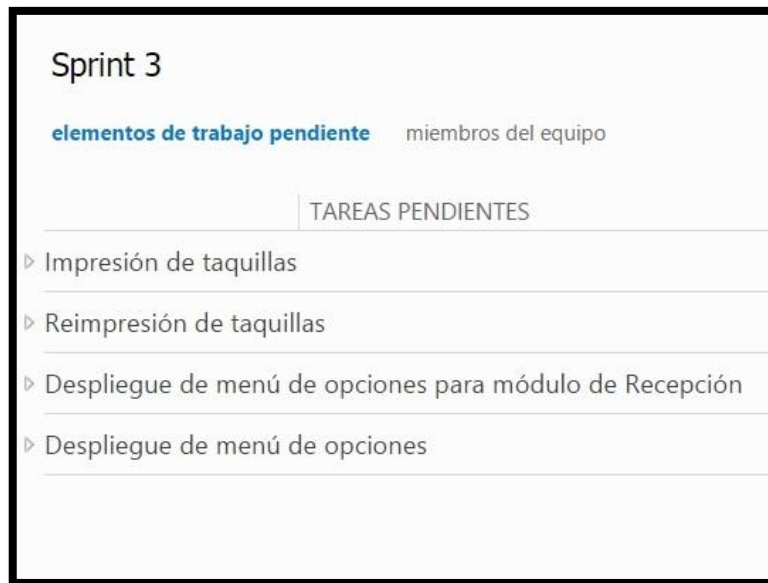


Figura 5-52: Historias de Usuario incluidas en el Sprint 3

5.2.3.1.1.2 Objetivo del Sprint

Demostrar la implementación de las nuevas funcionalidades de acuerdo a las historias de usuario el 10 de Agosto de 2012.

5.2.3.1.2 Definición del Sprint Backlog

En la Figura 5-53 se detallan todas las tareas asociadas a las historias de usuario asignadas a la iteración 3.

TAREAS PENDIENTES 97 h		EN CURSO	LISTO
Impresión de taquillas 40 h	Creación de clase "Constantes" que guarde los paths donde se guardarán 1 Integración de impresora de taquillas con equipo 13	Agregar elementos gráficos para impresión de taquilla 1 Codificación de capa de lógica de negocio 4 Codificación GUI 4	Codificación de capa de acceso a datos 11 Creación de servicios web para acceso a datos 6
Reimpresión de taquillas 25 h	Agregar elementos gráficos para reimpresión de taquillas 5 Codificación de capa de lógica de negocio 4	Codificación GUI 5 Codificación de capa de acceso de datos 6	Creación de servicio web para consulta de taquillas 5
Despliegue de menú de opciones para módulo de Recepción 17 h	Agregar elementos gráficos al menú de módulo Recepción 8	Codificación GUI 9	
Despliegue de menú de opciones 15 h	Agregar elementos gráficos al menú principal 4	Codificación GUI 5 Codificación de capa de lógica de negocio 6	

Figura 5-53: Sprint 3 Backlog

5.2.3.1.3 Estimación del esfuerzo del Sprint

De la Figura 5-54 a la Figura 5-57 se detallan todas las tareas necesarias para desarrollar las funcionalidades requeridas por las historias de usuario.

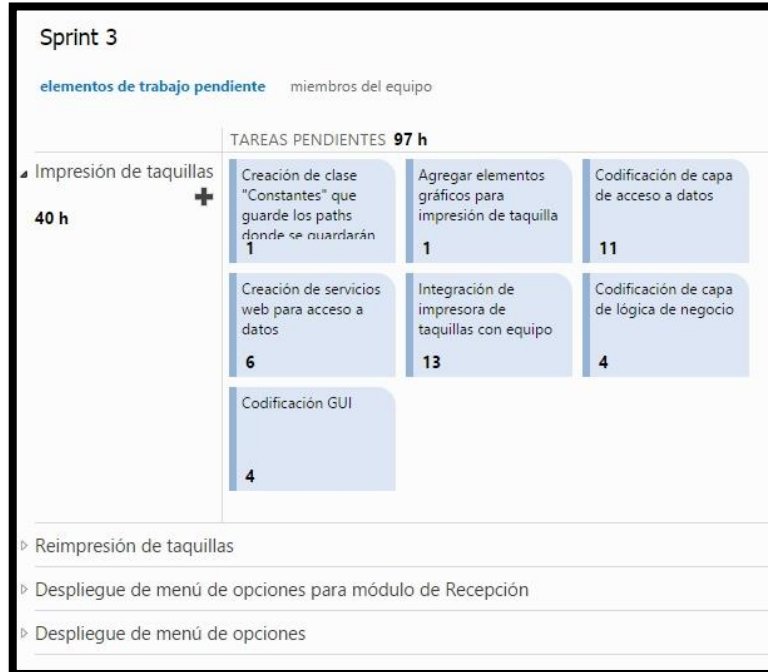


Figura 5-54: Esfuerzo de cada tarea para la primera historia de usuario

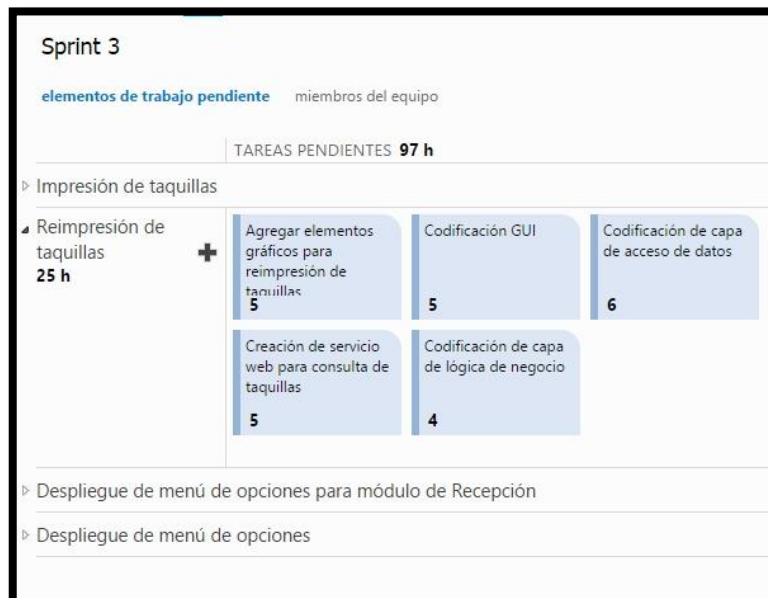


Figura 5-55: Esfuerzo de cada tarea para la segunda historia de usuario

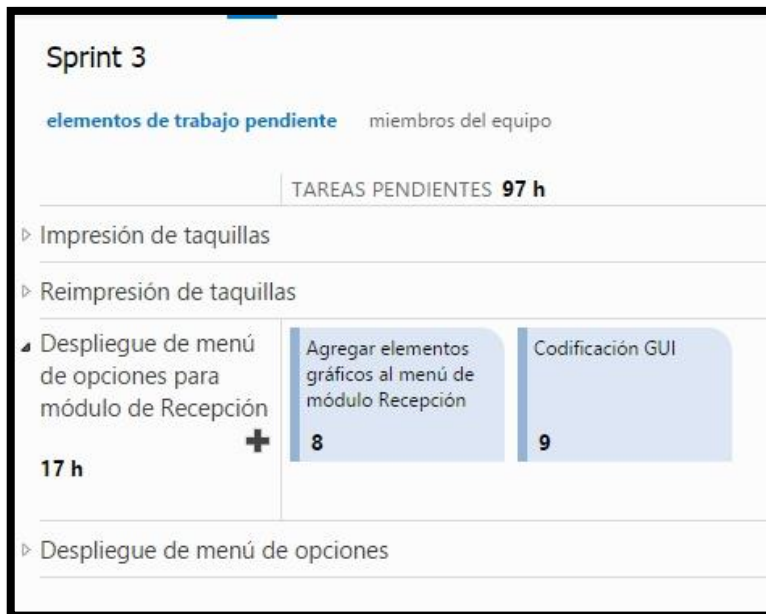


Figura 5-56: Esfuerzo de cada tarea para la tercera historia de usuario



Figura 5-57: Esfuerzo de cada tarea para la cuarta historia de usuario

5.2.3.2 Fase de Seguimiento

5.2.3.2.1 Desarrollo del Sprint

El propósito de este proceso, se detalla en la sección 5.2.1.2.1.

En estas reuniones diarias, se da el seguimiento necesario para pronosticar si se cumplirá el trabajo comprometido en las fechas establecidas.

De la Figura 5.58 a la Figura 5.67 se detalla el avance diario realizado por el equipo de trabajo.

5.2.3.2.1.1 Día 21

Sprint 3								
elementos de trabajo pendiente miembros del equipo								
		TAREAS PENDIENTES 83 h				EN CURSO 12 h	LISTO	
Impresión de taquillas 38 h	+	Codificación de capa de acceso a datos 11	Creación de servicios web para acceso a datos 6	Integración de impresora de taquillas con equipo 13	Codificación de capa de lógica de negocio 4	Codificación GUI 4	Creación de clase "Constantes" que guarde los paths donde se guardarán	Agregar elementos gráficos para impresión de taquilla
Reimpresión de taquillas 25 h	+	Agregar elementos gráficos para reimpresión de taquillas 5	Codificación GUI 5	Codificación de capa de acceso de datos 6	Creación de servicio web para consulta de taquillas 5			
		Codificación de capa de lógica de negocio 4						
Despliegue de menú de opciones para módulo de Recepción 17 h	+	Codificación GUI 9				Agregar elementos gráficos al menú de módulo Recepción 8		
Despliegue de menú de opciones 15 h	+	Agregar elementos gráficos al menú principal 4	Codificación GUI 5	Codificación de capa de lógica de negocio 6				

Figura 5-58: Panel de Tareas al final del día 21

5.2.3.2.1.2 Día 22

Sprint 3						
elementos de trabajo pendiente miembros del equipo						
	TAREAS PENDIENTES 63 h			EN CURSO 20 h	LISTO	
<ul style="list-style-type: none"> Impresión de taquillas 34 h 	<ul style="list-style-type: none"> Creación de servicios web para acceso a datos 6 	<ul style="list-style-type: none"> Integración de impresora de taquillas con equipo 13 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 11 	<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se guardarán 	<ul style="list-style-type: none"> Agregar elementos gráficos para impresión de taquilla Codificación GUI
<ul style="list-style-type: none"> Reimpresión de taquillas 25 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquillas 5 Creación de servicio web para consulta de taquillas 5 	<ul style="list-style-type: none"> Codificación GUI 5 Codificación de capa de lógica de negocio 4 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 6 			
<ul style="list-style-type: none"> Despliegue de menú de opciones para módulo de Recepción 9 h 				<ul style="list-style-type: none"> Codificación GUI 9 	<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción 	
<ul style="list-style-type: none"> Despliegue de menú de opciones 15 h 	<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal 4 	<ul style="list-style-type: none"> Codificación GUI 5 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 6 			

Figura 5-59: Panel de Tareas al final del día 22

5.2.3.2.1.3 Día 23

Sprint 3					
elementos de trabajo pendiente miembros del equipo					
	TAREAS PENDIENTES 63 h			EN CURSO 20 h	LISTO
<ul style="list-style-type: none"> Impresión de taquillas 34 h 	<ul style="list-style-type: none"> Creación de servicios web para acceso a datos 6 	<ul style="list-style-type: none"> Integración de impresora de taquillas con equipo 13 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 11 	<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se guardarán Agregar elementos gráficos para impresión de taquilla Codificación GUI
<ul style="list-style-type: none"> Reimpresión de taquillas 25 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquillas 5 Creación de servicio web para consulta de taquillas 5 	<ul style="list-style-type: none"> Codificación GUI 5 Codificación de capa de lógica de negocio 4 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 6 		
<ul style="list-style-type: none"> Despliegue de menú de opciones para módulo de Recepción 9 h 				<ul style="list-style-type: none"> Codificación GUI 9 	<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción
<ul style="list-style-type: none"> Despliegue de menú de opciones 15 h 	<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal 4 	<ul style="list-style-type: none"> Codificación GUI 5 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 6 		

Figura 5-60: Panel de Tareas al final del día 23

5.2.3.2.1.4 Día 24

Sprint 3			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES 48 h	EN CURSO 11 h	LISTO
Impresión de taquillas 23 h	<ul style="list-style-type: none"> Integración de impresora de taquillas con equipo 13 Codificación de capa de lógica de negocio 4 	<ul style="list-style-type: none"> Creación de servicios web para acceso a datos 6 	<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se guardarán Agregar elementos gráficos para impresión de taquilla Codificación de capa de acceso a datos Codificación GUI
Reimpresión de taquillas 25 h	<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquilla 5 Codificación GUI 5 Codificación de capa de acceso de datos 6 Creación de servicio web para consulta de taquillas 5 Codificación de capa de lógica de negocio 4 		
Despliegue de menú de opciones para módulo de Recepción			<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción Codificación GUI
Despliegue de menú de opciones 11 h	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 6 	<ul style="list-style-type: none"> Codificación GUI 5 	<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal

Figura 5-61: Panel de Tareas al final del día 24

5.2.3.2.1.5 Día 25

Sprint 3			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES 38 h	EN CURSO 10 h	LISTO
<ul style="list-style-type: none"> Impresión de taquillas 17 h 	<ul style="list-style-type: none"> Integración de impresora de taquillas con equipo 13 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 	<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se guardarán Agregar elementos gráficos para impresión de taquilla Codificación de capa de acceso a datos Creación de servicios web para acceso a datos Codificación GUI
<ul style="list-style-type: none"> Reimpresión de taquillas 25 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquillas 5 Creación de servicio web para consulta de taquillas 5 Codificación GUI 5 Codificación de capa de acceso de datos 6 Codificación de capa de lógica de negocio 4 		
<ul style="list-style-type: none"> Despliegue de menú de opciones para módulo de Recepción 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción Codificación GUI
<ul style="list-style-type: none"> Despliegue de menú de opciones 6 h 		<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 6 	<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal Codificación GUI

Figura 5-62: Panel de Tareas al final del día 25

5.2.3.2.1.6 Día 26

Sprint 3			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES 25 h	EN CURSO 13 h	LISTO
<ul style="list-style-type: none"> Impresión de taquillas 13 h 		<ul style="list-style-type: none"> Integración de impresora de taquillas con equipo 13 	<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se guardarán Agregar elementos gráficos para impresión de taquilla Codificación de capa de acceso a datos Creación de servicios web para acceso a datos Codificación de capa de lógica de negocio Codificación GUI
<ul style="list-style-type: none"> Reimpresión de taquillas 25 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquillas 5 Codificación GUI 5 Codificación de capa de acceso de datos 6 Creación de servicio web para consulta de taquillas 5 Codificación de capa de lógica de negocio 4 		
<ul style="list-style-type: none"> Despliegue de menú de opciones para módulo de Recepción 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción Codificación GUI
<ul style="list-style-type: none"> Despliegue de menú de opciones 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal Codificación GUI Codificación de capa de lógica de negocio

Figura 5-63: Panel de Tareas al final del día 26

5.2.3.2.1.7 Día 27

Sprint 3			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES 20 h	EN CURSO 13 h	LISTO
<ul style="list-style-type: none"> Impresión de taquillas 13 h 		<ul style="list-style-type: none"> Integración de impresora de taquillas con equipo 13 	<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se guardarán Agregar elementos gráficos para impresión de taquilla Codificación de capa de acceso a datos Creación de servicios web para acceso a datos Codificación de capa de lógica de negocio Codificación GUI
<ul style="list-style-type: none"> Reimpresión de taquillas 20 h 	<ul style="list-style-type: none"> Codificación GUI 5 Codificación de capa de acceso de datos 6 Creación de servicio web para consulta de taquillas 5 Codificación de capa de lógica de negocio 4 		<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquillas
<ul style="list-style-type: none"> Despliegue de menú de opciones para módulo de Recepción 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción Codificación GUI
<ul style="list-style-type: none"> Despliegue de menú de opciones 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal Codificación GUI Codificación de capa de lógica de negocio

Figura 5-64: Panel de Tareas al final del día 27

5.2.3.2.1.8 Día 28

Sprint 3			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES 15 h	EN CURSO 13 h	LISTO
<ul style="list-style-type: none"> Impresión de taquillas 13 h 		<ul style="list-style-type: none"> Integración de impresora de taquillas con equipo 13 	<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se guardarán Agregar elementos gráficos para impresión de taquilla Codificación de capa de acceso a datos Creación de servicios web para acceso a datos Codificación de capa de lógica de negocio Codificación GUI
<ul style="list-style-type: none"> Reimpresión de taquillas 15 h 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 6 Creación de servicio web para consulta de taquillas 5 Codificación de capa de lógica de negocio 4 		<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquillas Codificación GUI
<ul style="list-style-type: none"> Despliegue de menú de opciones para módulo de Recepción 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción Codificación GUI
<ul style="list-style-type: none"> Despliegue de menú de opciones 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal Codificación GUI Codificación de capa de lógica de negocio

Figura 5-65: Panel de Tareas al final del día 28

5.2.3.2.1.9 Día 29

Sprint 3			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES 5 h	EN CURSO 6 h	LISTO
<ul style="list-style-type: none"> Impresión de taquillas + 			<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se encuentran Agregar elementos gráficos para impresión de taquilla Codificación de capa de acceso a datos Creación de servicios web para acceso a datos Integración de impresora de taquillas con equipo Codificación de capa de lógica de negocio Codificación GUI
<ul style="list-style-type: none"> Reimpresión de taquillas 11 h + 	<ul style="list-style-type: none"> Creación de servicio web para consulta de taquillas 5 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 6 	<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquillas Codificación GUI Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Despliegue de menú de opciones para módulo de Recepción + 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción Codificación GUI
<ul style="list-style-type: none"> Despliegue de menú de opciones + 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal Codificación GUI Codificación de capa de lógica de negocio

Figura 5-66: Panel de Tareas al final del día 29

5.2.3.2.1.10 Día 30

Sprint 3			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES	EN CURSO	LISTO
<ul style="list-style-type: none"> Impresión de taquillas + 			<ul style="list-style-type: none"> Creación de clase "Constantes" que guarde los paths donde se guardarán Agregar elementos gráficos para impresión de taquilla Codificación de capa de acceso a datos Creación de servicios web para acceso a datos Integración de impresora de taquillas con equipo Codificación de capa de lógica de negocio Codificación GUI
<ul style="list-style-type: none"> Reimpresión de taquillas + 			<ul style="list-style-type: none"> Agregar elementos gráficos para reimpresión de taquillas Codificación GUI Codificación de capa de acceso a datos Creación de servicio web para consulta de taquillas Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Despliegue de menú de opciones para módulo de Recepción + 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú de módulo Recepción Codificación GUI
<ul style="list-style-type: none"> Despliegue de menú de opciones + 			<ul style="list-style-type: none"> Agregar elementos gráficos al menú principal Codificación GUI Codificación de capa de lógica de negocio

Figura 5-67: Panel de Tareas al final del día 30

5.2.3.2.2 *Burn-Down chart*

El significado de este artefacto se detalla en la sección 5.2.1.2.2

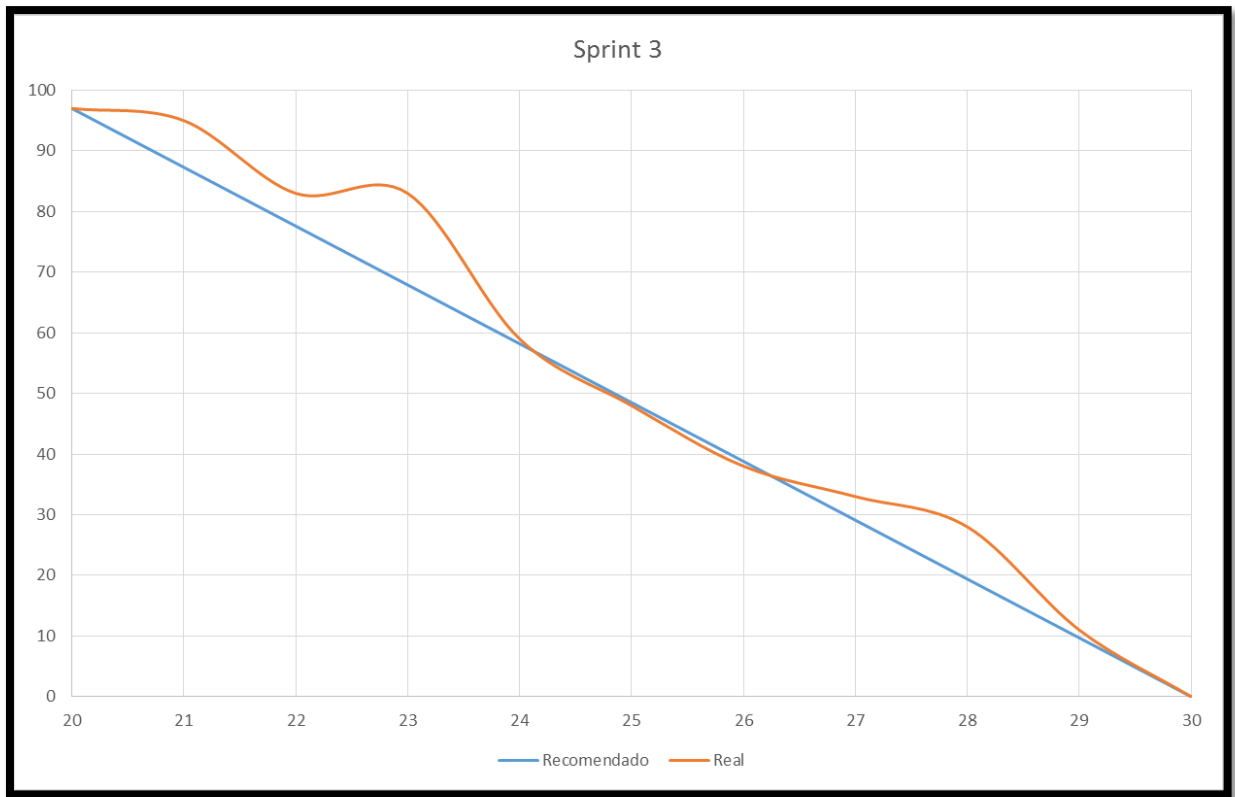


Figura 5-68: Burn down chart

5.2.3.2.3 *Reporte de defectos*

Al finalizar el sprint, se realizaron las pruebas respectivas del nuevo desarrollo. La Figura 5-69 muestra los defectos reportados.

The screenshot shows a web interface titled "Consulta de defectos" (Defect Query) with 6 selected items. The interface includes a toolbar with icons for saving, refreshing, and other actions. Below the toolbar is a table with the following data:

ID	Tipo De EL...	Título	Asignado A	Estado De ...	Etiquetas
134	Error	Defecto: Error desplegando la lista con las ubicaciones	Paul Lopez	Listo	
136	Error	Defecto: Cuando el peso medido por la balanza electrónica tiene decimales no es ...	Eduardo...	Listo	
137	Error	Defecto: No se despliegan todos los destinos nacionales en la lista	Eduardo...	Listo	
138	Error	Defecto: La reimpresión de taquillas se detiene en la 3era reimpresión		Nuevo	
139	Error	Defecto: La consulta de taquillas para su reimpresión no busca adecuadamente la...		Nuevo	
140	Error	Defecto: La interfaz para la impresión de taquillas no solicita todos los campos ne...		Nuevo	

Figura 5-69: Reporte de defectos inicial

El equipo de desarrollo procedió a realizar las verificaciones y correcciones respectivas, al finalizar esto, se realizaron nuevamente las pruebas haciendo énfasis sobre los defectos previamente encontrados. La Figura 5-70 muestra los resultados de los defectos

ID	Tipo De El.	Título	Asignado A	Estado De ...	Etiquetas
134	Error	Defecto: Error desplegando la lista con las ubicaciones	Paul Lopez	Listo	
136	Error	Defecto: Cuando el peso medido por la balanza electrónica tiene decimales no es ...	Eduardo...	Listo	
137	Error	Defecto: No se despliegan todos los destinos nacionales en la lista	Eduardo...	Listo	
138	Error	Defecto: La reimpresión de taquillas se detiene en la 3era reimpresión	Paul Lopez	Listo	
139	Error	Defecto: La consulta de taquillas para su reimpresión no busca adecuadamente la...	Paul Lopez	Listo	
140	Error	Defecto: La interfaz para la impresión de taquillas no solicita todos los campos ne...	Eduardo...	Listo	

Figura 5-70: Reporte de defectos final

corregidos.

5.2.3.2.4 Documento de la reunión para la revisión del Sprint

El propósito de este proceso es detallado en la sección 5.2.1.2.4

Los ítems demostrados y aprobados en este Sprint fueron:

- Impresión de taquillas
- Reimpresión de taquillas
- Despliegue de menú de opciones para módulo de recepción
- Despliegue de menú de opciones

En la sección “5.2.3.3.1 Producto terminado” se puede observar imágenes que demuestran las funcionalidades del sistema entregadas en el presente Sprint.

5.2.3.2.5 Documento de la reunión de la retrospectiva del equipo

Al final del sprint, el equipo de desarrollo se reunió para evaluar el desempeño de las partes involucradas durante el desarrollo, el resultado se refleja en la Tabla 5-5.

Pregunta	Completamente de acuerdo	De acuerdo	En desacuerdo	Completamente en desacuerdo	NA
¿Los objetivos y metas estuvieron claros?	X	X			
¿Las partes interesadas se involucraron en el proyecto?	XX				
¿Existió una correcta retroalimentación por parte del cliente?	XX				
¿Las funcionalidades se definieron claramente?	XX				
¿El equipo respondió adecuadamente a problemas surgidos durante el desarrollo?		X	X		
¿Se planificaron correctamente las actividades de acuerdo al calendario?		X	X		
¿El producto obtenido fue de acorde a las expectativas del cliente?	XX				

Tabla 5-5: Encuesta de la reunión de retrospectiva del equipo del sprint 3

5.2.3.3 Fase de Revisión

5.2.3.3.1 Producto terminado

De la Figura 5-71 a la Figura 5-75 se ilustran capturas de pantallas de las funcionalidades logradas en la presente iteración.



Figura 5-71: Botón que permite la impresión de la Taquilla



Figura 5-72: Modelo de taquilla impresa o reimpressa

SISTEMA DE CARGA TAME
RECEPCION

USUARIO
TSCA34

tame **CARGO**

Número Taquilla
250000 **Buscar**

Datos Taquilla

Origen: GYE Destino: UIO
 Peso: 0 Estado: 1
 Piezas: 1 Reimprimir desde Taquilla: Hasta:
 Cubicaje: 0 0 0

Reimprimir **Cancelar** **Regresar**

Figura 5-73: Ventana que permite la reimpresión de taquillas

SISTEMA DE CARGA TAME
RECEPCION

ORIGEN
UIO

tame **CARGO** USUARIO
TSCA34

Sobre Pequeño **Sobre Grande** **Mínimos** **Percible**

No Percible **Reimprimir Taquilla** **Modificar Taquilla** **Salir**

Figura 5-74: Despliegue de menú de opciones para módulo de recepción

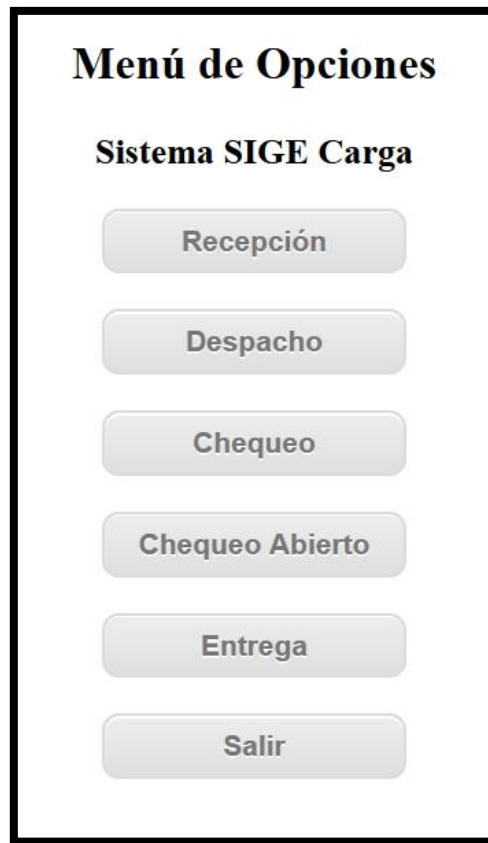


Figura 5-75: Despliegue de menú principal de opciones

5.2.3.3.2 *Notas de la versión*

Funcionalidades implementadas:

- Impresión de taquillas
- Reimpresión de taquillas
- Despliegue de menú de opciones para módulo de recepción
- Despliegue de menú de opciones

Defectos corregidos:

- La reimpresión de taquillas se detiene en la tercera reimpresión
- La consulta de taquillas para su reimpresión no busca adecuadamente la información completa de la taquilla
- La interfaz para la impresión de taquillas no solicita todos los campos necesarios

5.2.3.3.3 *Manual de Usuario*

El Manual de Usuario final integrado se lo puede encontrar en el Anexo A.

5.2.3.3.4 Programa de mejora

En la Tabla 5-6, se definen acciones a implementar para la mejora continua.

Programa de mejora			
Acciones de mejora	Condiciones de Ejecución	Tiempo de Ejecución	Recursos Necesarios
Ejecución de un mantenimiento regular preventivo de la impresora de taquillas	Inclusión del mantenimiento en partidas presupuestarias anuales	Mediano plazo	Humanos y financieros
Compra de etiquetas resistentes al agua	Aumento del presupuesto destinado a la compra de etiquetas	Corto plazo	Humanos y financieros

Tabla 5-6: Programa de mejora del sprint 3

5.2.4 Sprint 4

5.2.4.1 Fase de Planificación

5.2.4.1.1 Planificación del Sprint

En la jornada de trabajo previa al inicio del Sprint 4 se especificaron las historias de usuario que deberán estar concluidas al final de esta iteración y el objetivo de la iteración.

5.2.4.1.1.1 Historias de Usuario incluidas en el Sprint

Las historias de usuario incluidas en el presente Sprint se indican en la Figura 5-76.

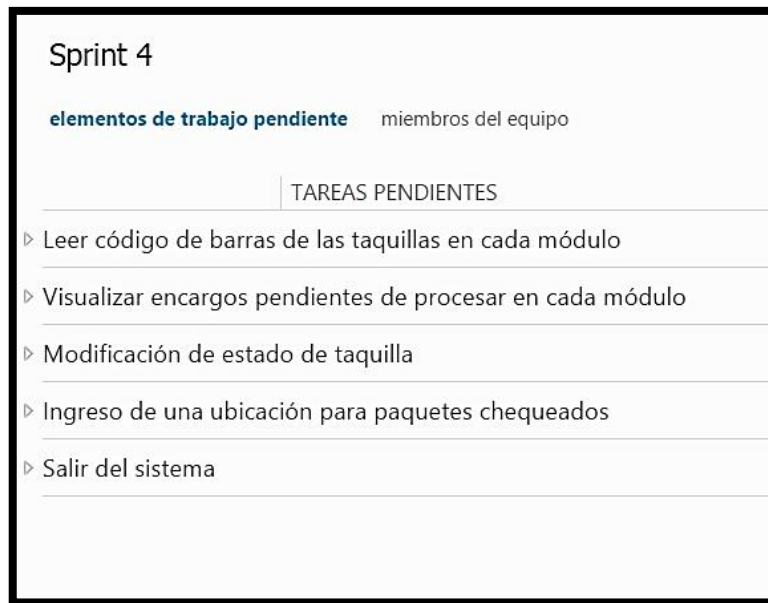


Figura 5-76: Historias de Usuario incluidas en el Sprint 4

5.2.4.1.1.2 Objetivo del Sprint

Demostrar la implementación de las nuevas funcionalidades de acuerdo a las historias de usuario el 24 de Agosto de 2012.

5.2.4.1.2 Definición del Sprint Backlog

En la Figura 5-77 se detallan todas las tareas asociadas a las historias de usuario asignadas a la iteración 4.

TAREAS PENDIENTES 98 h					EN CURSO	LISTO
<ul style="list-style-type: none"> Leer código de barras de las taquillas en cada módulo 35 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos de búsqueda de taquilla 6 	<ul style="list-style-type: none"> Codificación GUI 6 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 15 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 		
<ul style="list-style-type: none"> Visualizar encargos pendientes de procesar en cada módulo 28 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos de presentación de encargos pendientes 6 	<ul style="list-style-type: none"> Codificación GUI 6 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 		
<ul style="list-style-type: none"> Modificación de estado de taquilla 20 h + 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Validación de estado modificado de taquilla 4 			
<ul style="list-style-type: none"> Ingreso de una ubicación para paquetes chequeados 13 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para el ingreso alfanumérico del nombre una 3 	<ul style="list-style-type: none"> Codificación GUI 2 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 4 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 		
<ul style="list-style-type: none"> Salir del sistema 2 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para salir del sistema 1 	<ul style="list-style-type: none"> Codificación GUI 1 				

Figura 5-77: Sprint 4 Backlog

5.2.4.1.3 Estimación del esfuerzo del Sprint

De la Figura 5-78 a la Figura 5-82 se detallan las tareas necesarias para desarrollar las funcionalidades requeridas por las historias de usuario.

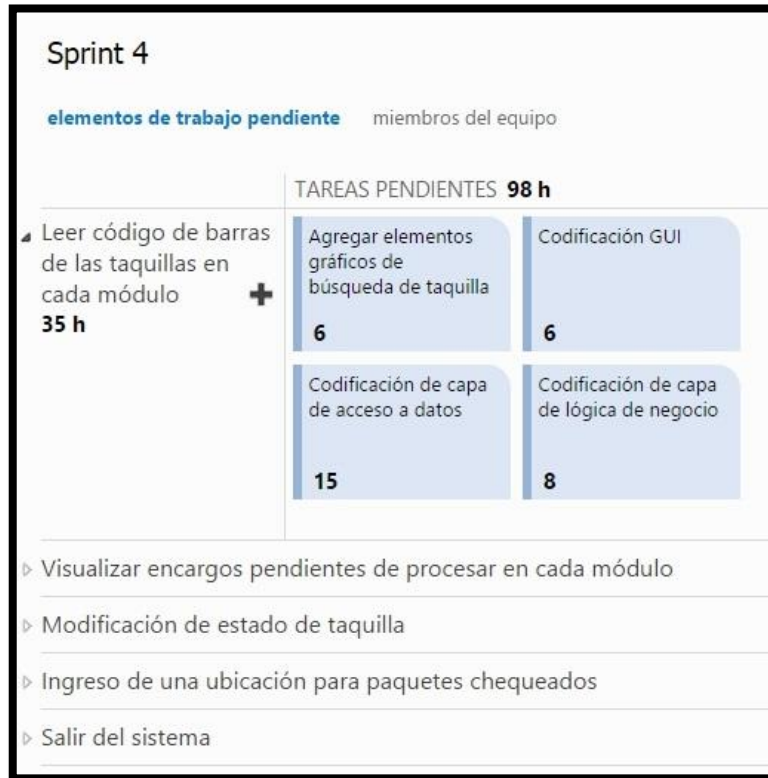


Figura 5-78: Esfuerzo de cada tarea para la primera historia de usuario

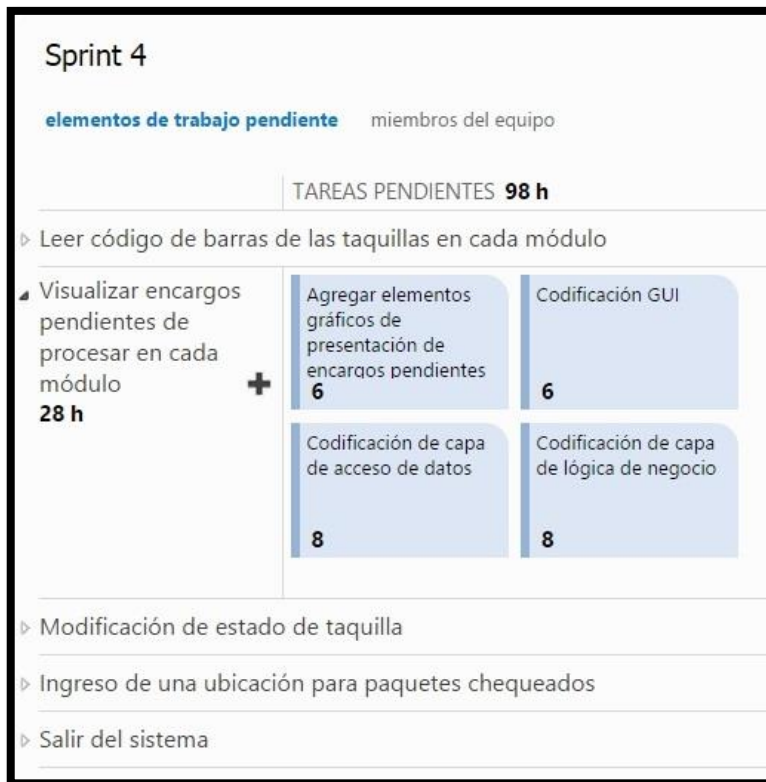


Figura 5-79: Esfuerzo de cada tarea para la segunda historia de usuario

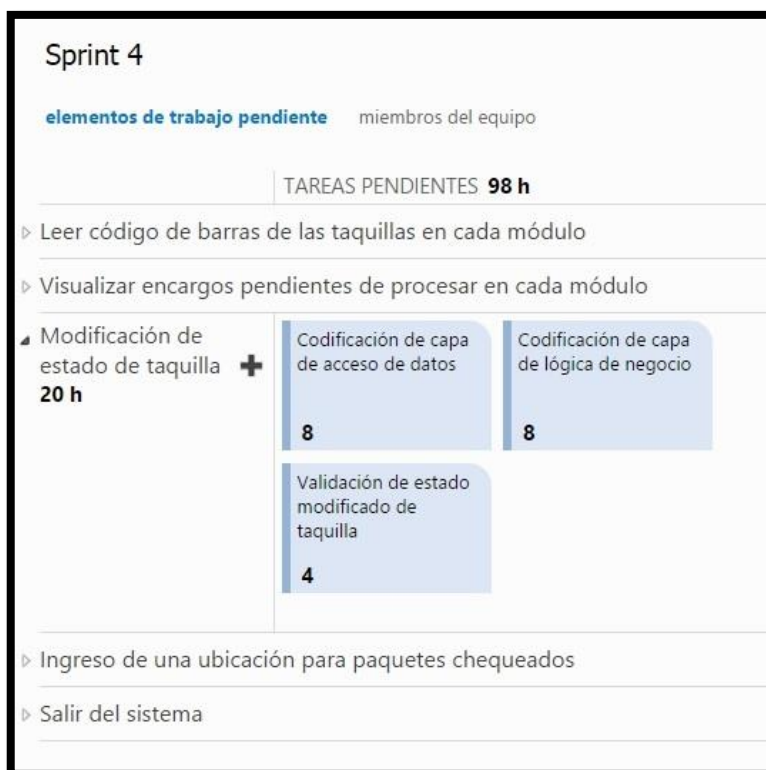


Figura 5-80: Esfuerzo de cada tarea para la tercera historia de usuario

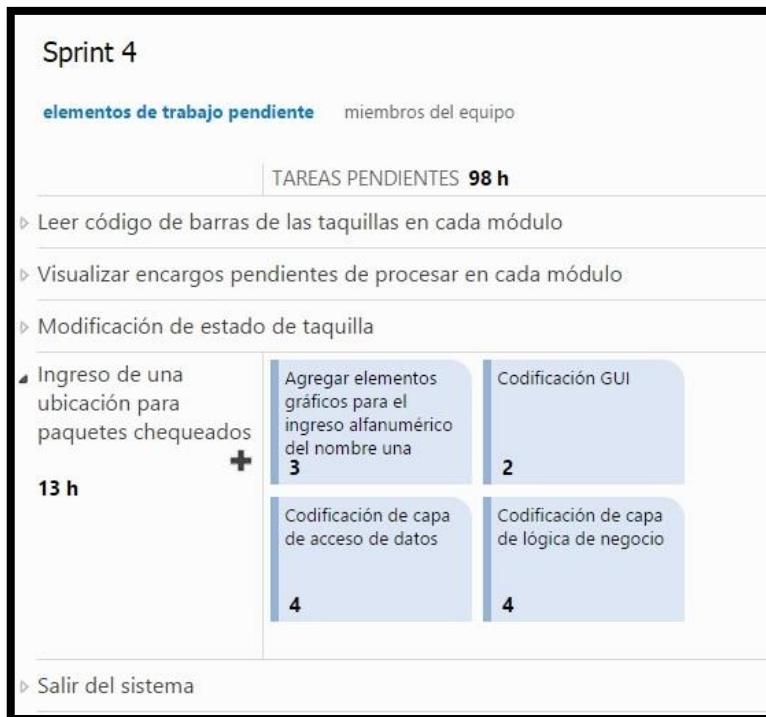


Figura 5-81: Esfuerzo de cada tarea para la cuarta historia de usuario

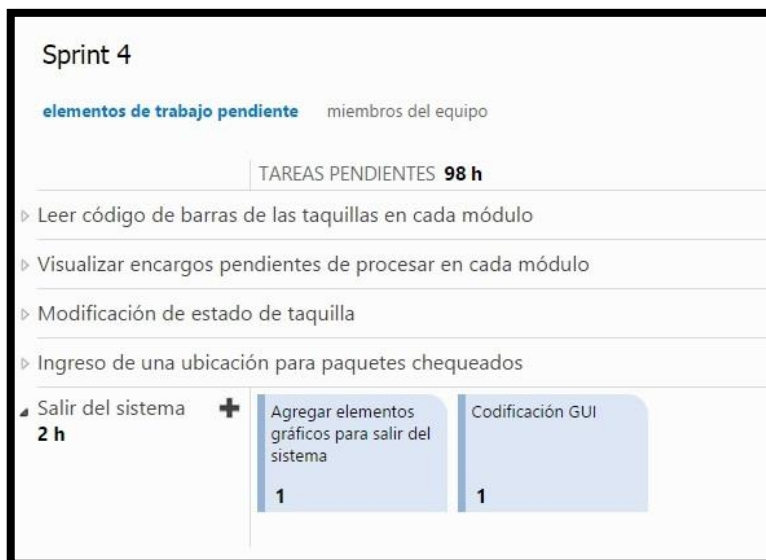


Figura 5-82: Esfuerzo de cada tarea para la quinta historia de usuario

5.2.4.2 Fase de Seguimiento

5.2.4.2.1 Desarrollo del Sprint

El propósito de este proceso se detalla en la sección 5.2.1.2.1.

De la Figura 5-83 a la Figura 5-92 se detalla el avance diario realizado por el equipo de trabajo.

5.2.4.2.1.1 Día 31

Sprint 4					
elementos de trabajo pendiente miembros del equipo					
	TAREAS PENDIENTES 86 h			EN CURSO 12 h	LISTO
Leer código de barras de las taquillas en cada módulo 35 h	Codificación GUI 6	Codificación de capa de acceso a datos 15	Codificación de capa de lógica de negocio 8	Agregar elementos gráficos de búsqueda de taquilla 6	
Visualizar encargos pendientes de procesar en cada módulo 28 h	Codificación GUI 6	Codificación de capa de acceso de datos 8	Codificación de capa de lógica de negocio 8	Agregar elementos gráficos de presentación de encargos pendientes 6	
Modificación de estado de taquilla 20 h	Codificación de capa de acceso de datos 8	Codificación de capa de lógica de negocio 8	Validación de estado modificado de taquilla 4		
Ingreso de una ubicación para paquetes chequeados 13 h	Agregar elementos gráficos para el ingreso alfanumérico del nombre una 3	Codificación GUI 2	Codificación de capa de acceso de datos 4	Codificación de capa de lógica de negocio 4	
Salir del sistema 2 h	Agregar elementos gráficos para salir del sistema 1	Codificación GUI 1			

Figura 5-83: Panel de Tareas al final del día 31

5.2.4.2.1.2 Día 32

Sprint 4					
elementos de trabajo pendiente miembros del equipo					
	TAREAS PENDIENTES 74 h			EN CURSO 12 h	LISTO
<ul style="list-style-type: none"> Leer código de barras de las taquillas en cada módulo + 29 h 	<ul style="list-style-type: none"> Codificación de capa de acceso a datos 15 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 		<ul style="list-style-type: none"> Codificación GUI 6 	<ul style="list-style-type: none"> Agregar elementos gráficos de búsqueda de taquilla
<ul style="list-style-type: none"> Visualizar encargos pendientes de procesar en cada módulo + 22 h 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 		<ul style="list-style-type: none"> Codificación GUI 6 	<ul style="list-style-type: none"> Agregar elementos gráficos de presentación de encargos pendientes
<ul style="list-style-type: none"> Modificación de estado de taquilla + 20 h 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Validación de estado modificado de taquilla 4 		
<ul style="list-style-type: none"> Ingreso de una ubicación para paquetes chequeados + 13 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para el ingreso alfanumérico del nombre una 3 	<ul style="list-style-type: none"> Codificación GUI 2 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 4 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 	
<ul style="list-style-type: none"> Salir del sistema + 2 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para salir del sistema 1 	<ul style="list-style-type: none"> Codificación GUI 1 			

Figura 5-84: Panel de Tareas al final del día 32

5.2.4.2.1.3 Día 33

Sprint 4					
elementos de trabajo pendiente miembros del equipo					
	TAREAS PENDIENTES 51 h			EN CURSO 23 h	LISTO
<ul style="list-style-type: none"> Leer código de barras de las taquillas en cada módulo 23 h + 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 			<ul style="list-style-type: none"> Codificación de capa de acceso a datos 15 	<ul style="list-style-type: none"> Agregar elementos gráficos de búsqueda de taquilla Codificación GUI
<ul style="list-style-type: none"> Visualizar encargos pendientes de procesar en cada módulo 16 h + 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 			<ul style="list-style-type: none"> Codificación de capa de acceso a datos 8 	<ul style="list-style-type: none"> Agregar elementos gráficos de presentación de encargos pendientes Codificación GUI
<ul style="list-style-type: none"> Modificación de estado de taquilla 20 h + 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Validación de estado modificado de taquilla 4 		
<ul style="list-style-type: none"> Ingreso de una ubicación para paquetes chequeados 13 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para el ingreso alfanumérico del nombre una 3 	<ul style="list-style-type: none"> Codificación GUI 2 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 4 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 	
<ul style="list-style-type: none"> Salir del sistema 2 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para salir del sistema 1 	<ul style="list-style-type: none"> Codificación GUI 1 			

Figura 5-85: Panel de Tareas al final del día 33

5.2.4.2.1.4 Día 34

Sprint 4					
elementos de trabajo pendiente miembros del equipo					
	TAREAS PENDIENTES 43 h			EN CURSO 23 h	LISTO
<ul style="list-style-type: none"> Leer código de barras de las taquillas en cada módulo 23 h 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 			<ul style="list-style-type: none"> Codificación de capa de acceso a datos 15 	<ul style="list-style-type: none"> Agregar elementos gráficos de búsqueda de taquilla Codificación GUI
<ul style="list-style-type: none"> Visualizar encargos pendientes de procesar en cada módulo 8 h 				<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Agregar elementos gráficos de presentación de encargos pendientes Codificación GUI Codificación de capa de acceso de datos
<ul style="list-style-type: none"> Modificación de estado de taquilla 20 h 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 8 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Validación de estado modificado de taquilla 4 		
<ul style="list-style-type: none"> Ingreso de una ubicación para paquetes chequeados 13 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para el ingreso alfanumérico del nombre una 3 	<ul style="list-style-type: none"> Codificación GUI 2 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 4 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 4 	
<ul style="list-style-type: none"> Salir del sistema 2 h 	<ul style="list-style-type: none"> Agregar elementos gráficos para salir del sistema 1 	<ul style="list-style-type: none"> Codificación GUI 1 			

Figura 5-86: Panel de Tareas al final del día 34

5.2.4.2.1.5 Día 35

Sprint 4					
elementos de trabajo pendiente miembros del equipo					
	TAREAS PENDIENTES 43 h			EN CURSO 15 h	LISTO
<ul style="list-style-type: none"> Leer código de barras de las taquillas en cada módulo 23 h + 	Codificación de capa de lógica de negocio 8			Codificación de capa de acceso a datos 15	Agregar elementos gráficos de búsqueda de taquilla Codificación GUI
<ul style="list-style-type: none"> Visualizar encargos pendientes de procesar en cada módulo + 					Agregar elementos gráficos de presentación de encargos pendientes Codificación GUI Codificación de capa de acceso de datos Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Modificación de estado de taquilla 20 h + 	Codificación de capa de acceso de datos 8	Codificación de capa de lógica de negocio 8	Validación de estado modificado de taquilla 4		
<ul style="list-style-type: none"> Ingreso de una ubicación para paquetes chequeados 13 h + 	Agregar elementos gráficos para el ingreso alfanumérico del nombre una 3	Codificación GUI 2	Codificación de capa de acceso de datos 4	Codificación de capa de lógica de negocio 4	
<ul style="list-style-type: none"> Salir del sistema 2 h + 	Agregar elementos gráficos para salir del sistema 1	Codificación GUI 1			

Figura 5-87: Panel de Tareas al final del día 35

5.2.4.2.1.6 Día 36

Sprint 4				
elementos de trabajo pendiente miembros del equipo				
	TAREAS PENDIENTES 27 h		EN CURSO 16 h	LISTO
<ul style="list-style-type: none"> Leer código de barras de las taquillas en cada módulo 8 h + 			<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Agregar elementos gráficos de búsqueda de taquilla Codificación GUI Codificación de capa de acceso a datos
<ul style="list-style-type: none"> Visualizar encargos pendientes de procesar en cada módulo + 				<ul style="list-style-type: none"> Agregar elementos gráficos de presentación de encargos pendientes Codificación GUI Codificación de capa de acceso a datos Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Modificación de estado de taquilla 20 h + 	<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Validación de estado modificado de taquilla 4 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 8 	
<ul style="list-style-type: none"> Ingreso de una ubicación para paquetes chequeados 13 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para el ingreso alfanumérico del nombre una 3 	<ul style="list-style-type: none"> Codificación GUI 2 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos 4 Codificación de capa de lógica de negocio 4 	
<ul style="list-style-type: none"> Salir del sistema 2 h + 	<ul style="list-style-type: none"> Agregar elementos gráficos para salir del sistema 1 	<ul style="list-style-type: none"> Codificación GUI 1 		

Figura 5-88: Panel de Tareas al final del día 36

5.2.4.2.1.7 Día 37

Sprint 4				
elementos de trabajo pendiente miembros del equipo				
	TAREAS PENDIENTES 18 h		EN CURSO 6 h	LISTO
• Leer código de barras de las taquillas en cada módulo +				<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Agregar elementos gráficos de búsqueda de taquilla</div> <div style="width: 20%;">Codificación GUI</div> <div style="width: 20%;">Codificación de capa de acceso a datos</div> <div style="width: 20%;">Codificación de capa de lógica de negocio</div> </div>
• Visualizar encargos pendientes de procesar en cada módulo +				<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Agregar elementos gráficos de presentación de encargos pendientes</div> <div style="width: 20%;">Codificación GUI</div> <div style="width: 20%;">Codificación de capa de acceso a datos</div> <div style="width: 20%;">Codificación de capa de lógica de negocio</div> </div>
• Modificación de estado de taquilla 12 h +	<div style="border: 1px solid #ccc; padding: 5px; width: 100%;">Codificación de capa de lógica de negocio</div> <div style="border: 1px solid #ccc; padding: 5px; width: 100%; background-color: #e0e0e0;">8</div>		<div style="border: 1px solid #ccc; padding: 5px; width: 100%;">Validación de estado modificado de taquilla</div> <div style="border: 1px solid #ccc; padding: 5px; width: 100%; background-color: #e0e0e0;">4</div>	<div style="border: 1px solid #ccc; padding: 5px; width: 100%;">Codificación de capa de acceso de datos</div>
• Ingreso de una ubicación para paquetes chequeados 10 h +	<div style="border: 1px solid #ccc; padding: 5px; width: 50%;">Codificación de capa de acceso de datos</div> <div style="border: 1px solid #ccc; padding: 5px; width: 50%;">Codificación de capa de lógica de negocio</div> <div style="border: 1px solid #ccc; padding: 5px; width: 50%; background-color: #e0e0e0;">4</div> <div style="border: 1px solid #ccc; padding: 5px; width: 50%; background-color: #e0e0e0;">4</div>		<div style="border: 1px solid #ccc; padding: 5px; width: 100%;">Codificación GUI</div> <div style="border: 1px solid #ccc; padding: 5px; width: 100%; background-color: #e0e0e0;">2</div>	<div style="border: 1px solid #ccc; padding: 5px; width: 100%;">Agregar elementos gráficos para el ingreso alfanumérico del nombre una</div>
• Salir del sistema 2 h +	<div style="border: 1px solid #ccc; padding: 5px; width: 50%;">Agregar elementos gráficos para salir del sistema</div> <div style="border: 1px solid #ccc; padding: 5px; width: 50%;">Codificación GUI</div> <div style="border: 1px solid #ccc; padding: 5px; width: 50%; background-color: #e0e0e0;">1</div> <div style="border: 1px solid #ccc; padding: 5px; width: 50%; background-color: #e0e0e0;">1</div>			

Figura 5-89: Panel de Tareas al final del día 37

5.2.4.2.1.8 Día 38

Sprint 4				
elementos de trabajo pendiente miembros del equipo				
	TAREAS PENDIENTES 6 h	EN CURSO 8 h	LISTO	
• Leer código de barras de las taquillas en cada módulo +			Agregar elementos gráficos de búsqueda de taquilla	Codificación GUI Codificación de capa de acceso a datos Codificación de capa de lógica de negocio
• Visualizar encargos pendientes de procesar en cada módulo +			Agregar elementos gráficos de presentación de encargos pendientes	Codificación GUI Codificación de capa de acceso a datos Codificación de capa de lógica de negocio
• Modificación de estado de taquilla 8 h +		Codificación de capa de lógica de negocio 8	Codificación de capa de acceso a datos	Validación de estado modificado de taquilla
• Ingreso de una ubicación para paquetes chequeados 4 h +	Codificación de capa de lógica de negocio 4		Agregar elementos gráficos para el ingreso alfanumérico del nombre una	Codificación GUI Codificación de capa de acceso a datos
• Salir del sistema 2 h +	Agregar elementos gráficos para salir del sistema 1	Codificación GUI 1		

Figura 5-90: Panel de Tareas al final del día 38

5.2.4.2.1.9 Día 39

Sprint 4			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES 1 h	EN CURSO 8 h	LISTO
<ul style="list-style-type: none"> Leer código de barras de las taquillas en cada módulo + 			<ul style="list-style-type: none"> Agregar elementos gráficos de búsqueda de taquilla Codificación GUI Codificación de capa de acceso a datos Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Visualizar encargos pendientes de procesar en cada módulo + 			<ul style="list-style-type: none"> Agregar elementos gráficos de presentación de encargos pendientes Codificación GUI Codificación de capa de acceso a datos Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Modificación de estado de taquilla 8 h + 		<ul style="list-style-type: none"> Codificación de capa de lógica de negocio 8 	<ul style="list-style-type: none"> Codificación de capa de acceso de datos Validación de estado modificado de taquilla
<ul style="list-style-type: none"> Ingreso de una ubicación para paquetes chequeados + 			<ul style="list-style-type: none"> Agregar elementos gráficos para el ingreso alfanumérico del nombre una Codificación GUI Codificación de capa de acceso de datos Codificación de capa de lógica de negocio
<ul style="list-style-type: none"> Salir del sistema 1 h + 	<ul style="list-style-type: none"> Codificación GUI 1 		<ul style="list-style-type: none"> Agregar elementos gráficos para salir del sistema

Figura 5-91: Panel de Tareas al final del día 39

5.2.4.2.1.10Día 40

Sprint 4			
elementos de trabajo pendiente miembros del equipo			
	TAREAS PENDIENTES	EN CURSO	LISTO
• Leer código de barras de las taquillas en cada módulo +			<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Agregar elementos gráficos de búsqueda de taquilla</div> <div style="width: 20%;">Codificación GUI</div> <div style="width: 20%;">Codificación de capa de acceso a datos</div> <div style="width: 20%;">Codificación de capa de lógica de negocio</div> </div>
• Visualizar encargos pendientes de procesar en cada módulo +			<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Agregar elementos gráficos de presentación de encargos pendientes</div> <div style="width: 20%;">Codificación GUI</div> <div style="width: 20%;">Codificación de capa de acceso a datos</div> <div style="width: 20%;">Codificación de capa de lógica de negocio</div> </div>
• Modificación de estado de taquilla +			<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Codificación de capa de acceso de datos</div> <div style="width: 20%;">Codificación de capa de lógica de negocio</div> <div style="width: 20%;">Validación de estado modificado de taquilla</div> </div>
• Ingreso de una ubicación para paquetes chequeados +			<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Agregar elementos gráficos para el ingreso alfanumérico del nombre una</div> <div style="width: 20%;">Codificación GUI</div> <div style="width: 20%;">Codificación de capa de acceso de datos</div> <div style="width: 20%;">Codificación de capa de lógica de negocio</div> </div>
• Salir del sistema +			<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Agregar elementos gráficos para salir del sistema</div> <div style="width: 20%;">Codificación GUI</div> </div>

Figura 5-92: Panel de Tareas al final del día 40

5.2.4.2.2 Burn-Down chart

El significado de este artefacto se detalla en la sección 5.2.1.2.2.

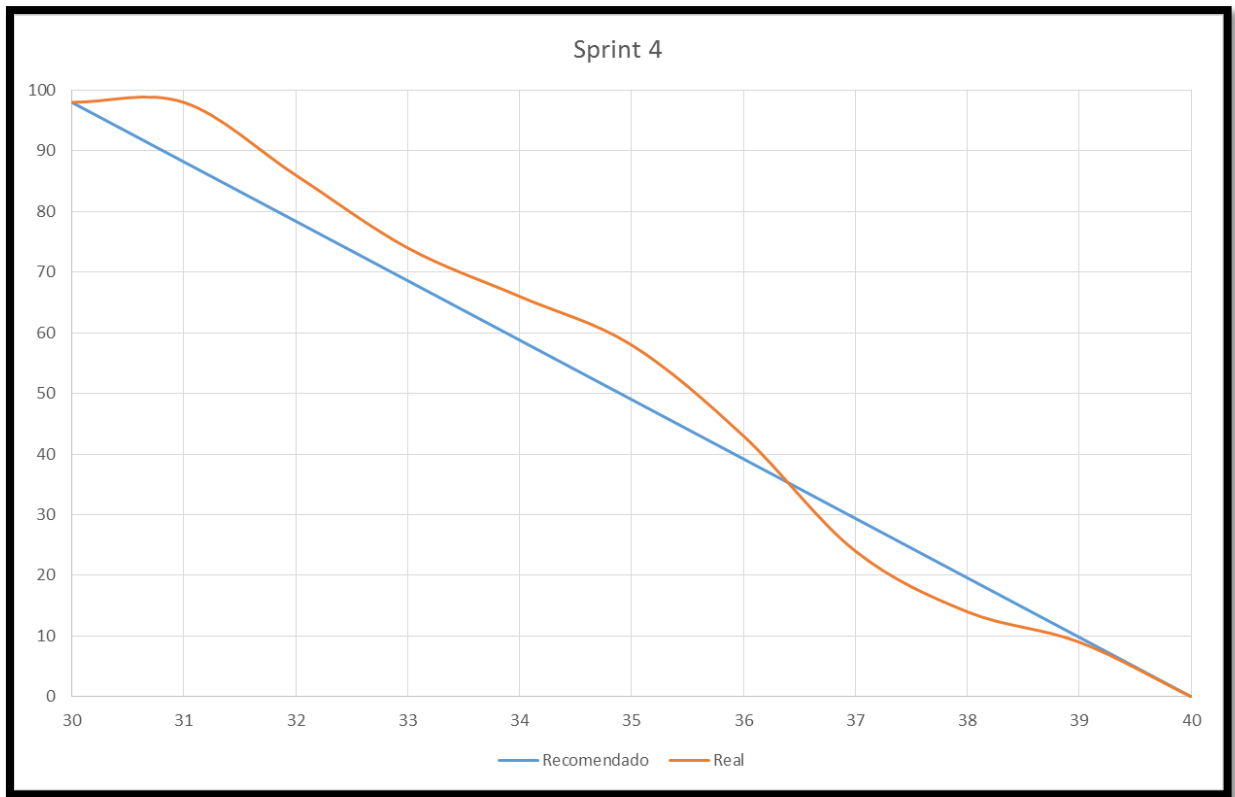


Figura 5-93: Burn down chart

5.2.4.2.3 Reporte de defectos

Al finalizar el sprint, se realizaron las pruebas respectivas del nuevo desarrollo. La Figura 5-94 muestra los defectos reportados.

The screenshot shows a web interface titled "Consulta de defectos" with a sub-header "7 elementos de trabajo (1 seleccionado)". Below the title are tabs for "resultados" and "editor", and a "panel de elementos ... inferior" link. A toolbar contains icons for "Guardar consulta", refresh, undo, redo, print, and email, along with "Opciones de columna". The main content is a table with the following data:

ID	Tipo De El...	Título	Asignado A	Estado De ...	Etiquetas
134	Error	Defecto: Error desplegando la lista con las ubicaciones	Paul Lopez	Listo	
136	Error	Defecto: Cuando el peso medido por la balanza electrónica tiene decimales no es ...	Eduardo...	Listo	
137	Error	Defecto: No se despliegan todos los destinos nacionales en la lista	Eduardo...	Listo	
138	Error	Defecto: La reimpresión de taquillas se detiene en la 3era reimpresión	Paul Lopez	Listo	
139	Error	Defecto: La consulta de taquillas para su reimpresión no busca adecuadamente la...	Paul Lopez	Listo	
140	Error	Defecto: La interfaz para la impresión de taquillas no solicita todos los campos ne...	Eduardo...	Listo	
141	Error	Defecto: El estado de una taquilla no se modifica correctamente		Nuevo	

Figura 5-94: Reporte de defectos inicial

El equipo de desarrollo procedió a realizar las verificaciones y correcciones respectivas, al finalizar esto, se realizaron nuevamente las pruebas haciendo énfasis sobre los defectos previamente encontrados. La Figura 5-95 muestra los resultados de los defectos corregidos.

ID	Tipo De El...	Título	Asignado A	Estado De ...	Etiquetas
134	Error	Defecto: Error desplegando la lista con las ubicaciones	Paul Lopez	Listo	
136	Error	Defecto: Cuando el peso medido por la balanza electrónica tiene decimales no es ...	Eduardo...	Listo	
137	Error	Defecto: No se despliegan todos los destinos nacionales en la lista	Eduardo...	Listo	
138	Error	Defecto: La reimpresión de taquillas se detiene en la 3era reimpresión	Paul Lopez	Listo	
139	Error	Defecto: La consulta de taquillas para su reimpresión no busca adecuadamente la...	Paul Lopez	Listo	
140	Error	Defecto: La interfaz para la impresión de taquillas no solicita todos los campos ne...	Eduardo...	Listo	
141	Error	Defecto: El estado de una taquilla no se modifica correctamente	Eduardo...	Listo	

Figura 5-95: Reporte de defectos final

5.2.4.2.4 Documento de la reunión para la revisión del Sprint

El propósito de este proceso se detalla en la sección 5.2.1.2.4.

Los ítems demostrados y aprobados en este Sprint fueron:

- Leer código de barras de las taquillas en cada módulo
- Visualizar encargos pendientes de procesar en cada módulo
- Modificación de estado de taquilla
- Ingreso de una ubicación para paquetes chequeados
- Salir del Sistema

En la sección “5.2.4.3.1 Producto terminado” se puede observar imágenes que demuestran las funcionalidades del sistema entregadas en el presente Sprint.

5.2.4.2.1 Documento de la reunión de la retrospectiva del equipo

Al final del sprint, el equipo de desarrollo se reunió para evaluar el desempeño de las partes involucradas durante el desarrollo, el resultado se refleja en la Tabla 5-7.

Pregunta	Completamente de acuerdo	De acuerdo	En desacuerdo	Completamente en desacuerdo	NA
¿Los objetivos y metas estuvieron claros?	X	X			
¿Las partes interesadas se involucraron en el proyecto?		XX			
¿Existió una correcta retroalimentación por parte del cliente?	X	X			
¿Las funcionalidades se definieron claramente?	XX				
¿El equipo respondió adecuadamente a problemas surgidos durante el desarrollo?	XX				
¿Se planificaron correctamente las actividades de acuerdo al calendario?	XX				
¿El producto obtenido fue de acorde a las expectativas del cliente?	XX				

Tabla 5-7: Encuesta de la reunión de retrospectiva del equipo del sprint 4

5.2.4.3 Fase de Revisión

5.2.4.3.1 Producto terminado

De la Figura 5-96 a la Figura 5-100, se ilustran capturas de pantallas de las funcionalidades logradas en la presente iteración.

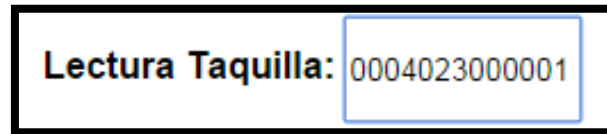


Figura 5-96: Lectura de código de barras de la taquilla

NumeroTaquilla	Peso	Piezas	NumeroGuia	NumeroConocimiento
422215	2	1	64893	511012
422989	0	1	64922	511101
423294	1	1	61391	511101
423971	1	1	61404	511272
425155	0	1	65042	511355
425372	0	1	61448	511355
427231	0	1	61544	511638
427428	0	1	61557	511638
431230	0	1	65311	512195
433411	1	1	65391	512460
435861	0	1	65478	512846
437242	3	1	65526	513010
437948	16	1	62114	513132
440238	2	1	5005123	1
440615	21	1	62247	513449
446440	4	1	62536	514259
446928	0	1	62581	514259
450648	0	1	62715	514770
451237	0	1	62754	514887
451693	0	1	66137	514887

Figura 5-97: Visualización de encargos pendientes de procesar

SISTEMA DE CARGA TAME
RECEPCION

USUARIO: TSCA34

tame CARGO

Número Taquilla: 150008 **Buscar**

Datos Taquilla

Origen: UIO Destino: GYE

Peso: 0 Estado: ACTIVO

Piezas: 1

Cubicaje: 0 0 0

Modificar Cancelar Grabar Regresar

Figura 5-98: Pantalla que permite la modificación del estado de una taquilla

CHEQUEO

Usuario: TSCA34 Ciudad Oficina: UIO 79990035

Manifiesto	Ubicación	Origen	Destino	Fecha	Vuelc
430483	AEROPUERTO	CUE	UIO	2013012	1172

S. PEQUEÑO S. GRANDE

Ubicación: Ingrese Ubicación..

Taquilla: MENU

Numero Taquilla	Peso	Piezas	Numero Guia	Numero Conocimiento
12194	24	1	33511	430483
12198	1	1	33513	430483
12199	6	1	33514	430483
12200	5	1	33515	430483
12200	5	2	33515	430483
12208	1	1	33523	430483
12213	16	1	33528	430483

Figura 5-99: Ingreso de ubicación para paquetes chequeados



Figura 5-100: Opción para salir del Sistema desde el menú principal

5.2.4.3.2 Notas de la versión

Funcionalidades implementadas:

- Leer código de barras de las taquillas en cada módulo
- Visualizar encargos pendientes de procesar en cada módulo
- Modificación de estado de taquilla
- Ingreso de una ubicación para paquetes chequeados
- Salir del sistema

Defectos corregidos:

- El estado de una taquilla no se modifica correctamente

5.2.4.3.3 Manual de Usuario

El Manual de Usuario final integrado se lo puede encontrar en el Anexo A.

5.2.4.3.4 Programa de mejora

La Tabla 5-8 detalla las acciones a implementar para la mejora continua.

Programa de mejora			
Acciones de mejora	Condiciones de Ejecución	Tiempo de Ejecución	Recursos Necesarios
Ejecución de un mantenimiento regular preventivo PDA's y lectores ópticos	Inclusión del mantenimiento en partidas presupuestarias anuales	Mediano plazo	Humanos y financieros
Adquisición de equipos tecnológicos más potentes y resistentes compatibles con los últimos estándares en software	Aumento del presupuesto destinado a la compra de equipos tecnológicos	Mediano plazo	Humanos y financieros
Elaboración de un módulo que sea capaz de informar al usuario vía mensajes de texto la ubicación del paquete	Capacitación a los desarrolladores sobre temas relacionados	Largo plazo	Humanos y financieros

Tabla 5-8: Programa de mejora del sprint 4

6 COMENTARIOS, CONCLUSIONES Y RECOMENDACIONES

6.1 Comentarios

En todo desarrollo de software es crucial que el usuario esté involucrado durante todo el proceso para que el equipo de desarrollo pueda cristalizar su visión de lo que debería ser el producto terminado, y dedicar menos tiempo a corregir defectos por mala interpretación de los programadores.

Un sistema no depende únicamente del código desarrollado, es fundamental garantizar la infraestructura que lo soportará y realizar los mantenimientos respectivos.

Es importante tener herramientas que se acoplen entre sí y eviten que el equipo de desarrollo desperdicie tiempo integrándolas, en nuestro caso, Visual Studio y Team Foundation Server, por su diseño conceptual, se integran inmediatamente y permiten que distintos procesos sean administrados desde una misma interfaz.

Una gran ventaja fue la incorporación de la plantilla de Scrum en Team Foundation Server, si bien es imprescindible tener conocimiento sobre esta metodología, el tener una herramienta que automatiza sus procesos ayuda significativamente para su ejecución.

Un levantamiento adecuado de requerimientos es crucial para tener un claro panorama de la visión del usuario para un desarrollo.

El usar herramientas existentes dedicadas para un determinado proceso ayuda a todo el equipo a enfocarse en el proyecto como tal, minimizando el desperdicio de recursos en tareas que no tienen impacto directo en el producto.

6.2 Conclusiones

- Durante el desarrollo del presente trabajo se escogieron 2 metodologías de desarrollo, Scrum y Extreme Programming, las cuales se acoplaban al presente tipo de proyecto y al equipo de desarrollo.
- Scrum y Extreme Programming son 2 metodologías complementarias como se pudo confirmar durante el desarrollo del presente proyecto. La primera abarca las fases del ciclo de vida del desarrollo, mientras que la segunda se enfoca en las prácticas de programación.

- Emplear una herramienta de versionamiento de código ayudó a mejorar la productividad del equipo de desarrollo.
- Para cumplir con el desarrollo del presente trabajo, se debió prestar importante atención a:
 - Que el usuario esté involucrado y disponible durante el desarrollo, de manera que se puedan hacer revisiones exhaustivas del incremento del producto y cualquier defecto sea solventado.
 - Definir claramente los criterios de aceptación para cada funcionalidad, de manera que se evidencie qué es lo que el usuario desea de un desarrollo.
 - Comunicación constante entre todo el equipo.
 - Abstracción en capas, así el código se hizo más comprensible y posibilitará su mantenimiento.

6.3 Recomendaciones

- Identificar claramente el rol y responsabilidades de cada integrante del equipo.
- Hacer un estudio previo de la(s) metodología(s) de desarrollo de software que se aplicará(n) en la construcción de una aplicación. Cada metodología tiene sus ventajas para ciertos proyectos. Esto influirá en la forma de trabajo y en los resultados finales.
- No iniciar ningún desarrollo hasta que todo el equipo tenga completamente claro el resultado que se desea obtener.
- Mantener las reuniones de seguimiento diarias involucrando únicamente al equipo de desarrollo y al Scrum Master, es importante que exista un ambiente seguro donde se puedan compartir todos los avances y problemas encontrados sin presión externa.
- Mantener las reuniones de retrospectiva al finalizar cada sprint y exponer los puntos de vista directa y honestamente a fin de mejorar procesos.
- El equipo de desarrollo debe estandarizar convenciones para nombres de proyectos, funciones y variables en el código.
- El ambiente de desarrollo donde se codifique, debe ser lo más similar al ambiente de producción, a fin de minimizar impacto al liberar el producto.
- Combinar las metodologías ágiles Scrum y Extreme Programming ya que proporciona buenos resultados en equipos de desarrollo pequeños.

- La pila del producto debe ser revisada y actualizada constantemente por el dueño del producto, el equipo de desarrollo siempre se referirá a ésta para implementar las funcionalidades.
- A la PUCE, recomendamos que como parte de la malla curricular se dicten clases acerca de metodologías ágiles de desarrollo con sus respectivos talleres. En la actualidad los desarrolladores buscan mayor flexibilidad mayor tiempo para centrarse en lo que verdaderamente importa que es el producto terminado.

7 BIBLIOGRAFÍA

- Alice. (1 de Septiembre de 2009). *DB2: YouTube*. Recuperado el 25 de Junio de 2014
- Andrade, J. (6 de Noviembre de 2013). *Arquitectura de Software: YouTube*. Recuperado el 22 de 6 de 2014
- Anónimo. (9 de Mayo de 2013). *Dos Ideas*. Recuperado el 25 de Junio de 2014
- Anónimo. (2014). *Compile: Webopedia*. Retrieved 06 23, 2014
- Anónimo. (29 de Abril de 2014). *Escáner de código de barras*. Obtenido de Wikipedia: http://es.wikipedia.org/wiki/Esc%C3%A1ner_de_c%C3%B3digo_de_barras
- Anónimo. (4 de Junio de 2014). *Herramienta CASE: Wikipedia*. Recuperado el 1 de Julio de 2014
- Anónimo. (2014). *Interpreter: Webopedia*. Recuperado el 23 de 6 de 2014
- Anónimo. (2014). *Overview: Aquafold*. Retrieved Junio 25, 2014
- Anónimo. (2014). *Overview: IIS*. Retrieved Junio 24, 2014
- Anónimo. (2014, Mayo 6). *PowerDesigner: Wikipedia*. Retrieved 6 25, 2014
- Anónimo. (2014). *Team Foundation Server: Visual Studio*. Retrieved Junio 25, 2014
- Anónimo. (2014). *TFS: Microsoft*. Recuperado el 25 de Junio de 2014
- Anónimo. (s.f.). *ejemplosTIW*. Recuperado el 30 de Junio de 2014
- Anónimo. (s.f.). *Impresoras*. Recuperado el 28 de Junio de 2014
- Anónimo. (s.f.). *LAS IMPRESORAS TÉRMICAS*. Obtenido de Informática Moderna: http://www.informaticamoderna.com/Impresora_termica.htm
- Anónimo. (s.f.). *Proyectos Agiles*. Recuperado el 28 de Junio de 2014
- Arquitectura de Software: Wikipedia*. (1 de Abril de 2014). Recuperado el 1 de Julio de 2014
- Barrera, L. (26 de Febrero de 2014). *Arquitectura de hardware y software 2014: slideshare*. Recuperado el 1 de Julio de 2014
- Beltrán, A. (Dirección). (2013). *Logística y Transporte de Mercancías* [Película].
- Besteiro, M., & Rodríguez, M. (2014). *Servicios Web*. Obtenido de Universidad del País Vasco: <http://www.ehu.es/mrodriguez/archivos/csharp/pdf/ServiciosWeb/WebServices.pdf>

- Carmack, C., & Freudenrich, C. (n.d.). *How PDAs Work: howstuffworks*. Retrieved Julio 1, 2014, from howstuffworks: <http://electronics.howstuffworks.com/gadgets/travel/pda.htm>
- Chromatic. (n.d.). *Extreme Programming. Pocket Guide*. O'Reilly.
- Creación de Sitios Web y su utilidad didáctica*. (2014). Obtenido de Universidad de Sevilla: <http://www.sav.us.es/formaciononline/creaciondesitiosweb/apartados/apartado3-1.asp>
- Cruz Orellana, R. E. (21 de Julio de 2006). *Herramientas Case: Galeon*. Recuperado el 1 de Julio de 2014, de Galeon.
- Cupi Veliz, J. (9 de Marzo de 2014). *Aplicación en 3 capas utilizando ASP: Scribd*. Obtenido de Scribd: <http://es.scribd.com/doc/211417205/Aplicacion-en-3-capas-utilizando-ASP-pdf>
- Danysoft. (2 de Septiembre de 2012). *Microsoft Visual Studio 2010: YouTube*. Recuperado el 24 de Junio de 2014
- Dodge, K., Lehto, B., & Weiner, M. (2008, Octubre). *IIS: Search Windows Server*. Retrieved Junio 24, 2014
- Echeverría, C. (20 de Octubre de 2012). *Arquitectura de Software: Youtube*. Recuperado el 22 de Junio de 2014
- Ellen, S. (s.f.). *Computación y electrónica*. Obtenido de eHow en Español: http://www.ehowenespanol.com/partes-codigo-barras-lista_83859/
- Escáner de código de barras: Wikipedia*. (29 de Abril de 2014). Recuperado el 1 de Julio de 2014, de ingeniatic: <http://ingeniatic.net/index.php/tecnologias/item/492-lector-de-c%C3%B3digo-de-barras>
- fergarcia. (25 de Enero de 2013). *Entorno de Desarrollo Integrado (IDE): wordpress*. Recuperado el 1 de Julio de 2014
- fmorazan. (8 de Junio de 2010). *Ingeniería de Software 1: blogspot*. Recuperado el 1 de Julio de 2014
- Gallego Herrera, E. (22 de Noviembre de 2012). *Arquitectura cliente-servidor: blogspot*. Recuperado el 1 de Julio de 2014
- García, F. (25 de Enero de 2013). *Entorno de Desarrollo Integrado: Wordpress*. Recuperado el 23 de 6 de 2014
- Gómez, A. (s.f.). *slideshare*. Recuperado el 30 de Junio de 2014
- Landeros, A. (9 de Febrero de 2011). *Herramientas Case: Youtube*. Recuperado el 25 de Junio de 2014

- Las Impresoras Térmicas: informaticamoderna.* (s.f.). Recuperado el 1 de Julio de 2014
- Lector de código de barras: ingeniatic.* (s.f.). Recuperado el 30 de Junio de 2014
- López Castillo, R. F. (16 de Enero de 2014). *Ingeniería de software - Descripción, características, modelos: slideshare.* Recuperado el 1 de Julio de 2014
- Nagel, C., Glynn, J., & Skinner, M. (2014). *Professional C# 5.0 and .NET 4.5.1.* Indianapolis: John Wiley & Sons, Inc.
- Nayab, N. (2011, Mayo 5). *The Extreme Programming Life Cycle: Bright Hub PM.* Retrieved 06 22, 2014
- Payne, C. (2002). *Books: Google.* Obtenido de Google: http://books.google.com.ec/books?id=iXxI7pB8ntkC&pg=PA791&lpg=PA791&dq=formularios+web+asp.net&source=bl&ots=oTIHQQP0WX&sig=vwuF742XV2q6jnY-udoDokQPJGg&hl=es-419&sa=X&ei=0N_TU8X6N4zIsATayYGABQ&ved=0CDsQ6AEwBDgK#v=onepage&q=formularios%20web%20asp.net&f=fa
- Pries, K. H., & Quigley, J. M. (2011). *Scrum Project Management.* Boca Raton: CRC Press.
- Rouse, M. (2007, Abril). *Versioning: Search Software Quality.* Retrieved Junio 24, 2014
- Sánchez, A. (25 de Enero de 2013). *ENTORNO DE DESARROLLO INTEGRADO(IDE): wordpress.* Recuperado el 1 de Julio de 2014
- Santillana, F. (15 de Junio de 2008). Obtenido de Wikispaces: <http://inf411.wikispaces.com/file/view/Elaboraci%C3%B3n+de+un+Formulario+Web+utilizando+ASP.pdf>
- Tuttini, L. (Octubre de 2009). *Developer Network.* Obtenido de Microsoft: <http://social.msdn.microsoft.com/Forums/es-ES/0d428d56-1425-45bb-9896-1bbf0211f8e6/definicion-de-postback?forum=netfxwebes>
- Uriza, T. (26 de Julio de 2011). *Taller cargue y descargue: slideshare.* Recuperado el 1 de Julio de 2014
- Web Services Standards.* (2014). Retrieved from tutorialspoint: http://www.tutorialspoint.com/webservices/web_services_standard.htm
- Wells, D. (2013, Octubre 8). *Extreme Programming.* Retrieved Junio 20, 2014

West, D. (17 de Febrero de 2009). *Use Cases Considered Valuable (but Optional) For Lean/Agile Requirements Capture*. Obtenido de InfoQ: <http://www.infoq.com/news/2009/02/Use-Cases-Valuable-But-Optional>