

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

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

TEMA:

IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA LA AUTOMATIZACIÓN DE LOS PROCESOS DE DONACIONES, REGISTRO DE CAJA, CONTROL DE PACIENTES E INVENTARIO, UTILIZANDO VISUAL STUDIO EXPRESS PARA EL ALBERGUE FUDIS.

AUTORES:

GUAMÁN GUAÑUNA CARLOS ANDRÉS

RECALDE MOSQUERA JORGE MANUEL

QUITO – 2013

Declaratoria de responsabilidad

Los conceptos desarrollados, análisis realizado y las conclusiones del presente trabajo, son de exclusiva responsabilidad de los autores.

Carlos Andrés Guamán Guañuna

Jorge Manuel Recalde Mosquera

DEDICATORIA

A mi madre Mercedes.

Por haber estado conmigo en todo momento y circunstancia, por sus consejos, sus reprimendas y valores, por haberme enseñado a valorar cada acción que se presenta en mi vida, por recordarme siempre que para conseguir un objetivo se debe luchar duro, por siempre asegurarme mi estabilidad ante sus necesidades y por brindarme su amor y ternura cada vez que lo necesitaba.

A mi padre Andrés.

Por brindarme su ejemplo de perseverancia y firmeza, por ser la persona quien sin necesidad de estar siempre a mi lado, creía en mis pensamientos e ilusiones, que sin dudar ponía mis esperanzas delante de sus aspiraciones, porque me ha mostrado que en el camino hay que superarse cada día y que con esfuerzo se combina ideologías, anhelos y proyectos.

A mis hermanos.

A mi hermana Nancy por ayudarme a nunca rendirme ante la frustración, a mi hermano Juan por ser mi ejemplo de superación, a mi hermano William por ser mi apoyo en momentos difíciles, a ellos porque sin dudar harian lo imposible para socorrerme cuando he caido.

A mi novia

A ella, a la cual no le importo las horas de espera, a ella quién fue mi aliciente en momentos duros y de pesar, a ella quien creyó en mí en toda ocasión y me apoyó en todas mis decisiones.

A mis amigos.

A ellos que sin su apoyo incondicional no habría llenado el espacio de sabiduría que me faltaba y porque sin ellos habría sido una larga y aburrida carrera en el camino de la universidad.

Att.

Carlos Andrés Guamán Guañuna.

DEDICATORIA

A mi familia.

Por estar a mi lado y darme su apoyo incondicional, por mostrarme el valor de la amistad y la familia, por haber depositado en mí su amor y sus anhelos.

A mi madre Patricia.

Por estar siempre conmigo en todo momento. Gracias por la paciencia que has tenido para enseñarme, por el amor que me das, por tus cuidados, por los regaños, y por apoyarme siempre en todas las decisiones que he tomado en la vida. Te dedico este trabajo de tesis por ser el pilar fundamental de mi vida.

A padre Jorge.

Por estar siempre en los momentos importantes de mi vida, por ser el ejemplo para salir adelante y por los consejos que fueron de gran ayuda para forjar mis principios, mi carácter, mi empeño, mi perseverancia y mi coraje para seguir adelante. Esta tesis es el resultado de lo que me has enseñado en la vida, ya que siempre has sido una persona honesta y entregada a tu trabajo, pero más que todo eso, una gran persona que siempre ha podido salir adelante. Es por ello que hoy te dedico este trabajo de tesis.

A mi abuela Carmen.

Por enseñarme el camino de la vida, gracias por tus consejos, por el amor que me has dado y por tu apoyo incondicional en mi vida.

A mi tía Luz y mi tío Guillermo.

A ustedes a quien quiero como madre y padre, por compartir momentos significativos conmigo y por siempre estar dispuestos a escucharme y ayudarme en cualquier momento.

A mi amiga Lynda.

A veces llegan personas a nuestras vidas para no quedarse por mucho tiempo, sin embargo, tú fuiste quién nuna perdió la fé en mi y supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a encarar las adversidades.

Att.

Jorge Manuel Recalde Mosquera.

AGRADECIMIENTOS

Antes que nada agradezco a Dios por cuidarme y protegerme en cada instante de mi vida y por permitirme estar al lado de las personas que amo.

A la Pontificia Universidad Católica del Ecuador, porque gracias a esta pude aprender a valorar cada oportunidad que se presenta en la vida, en la cual forme mis valores, y aptitudes y en la cual mostre todas mis actitudes durante mi larga estadia.

A mis padres Andrecito y Michita por apoyarme siempre y cada vez que lo necesité con sus consejos, con sus expresiones de cariño y sobre todo anteponiendo sus deseos ante mis necesidades.

A mis hermanos y hermana por darme una palabra de aliento, y ante todas las cosas por confiar en la tenacidad de mis propósitos.

A mis amigos de clase que me dieron su ánimo y carisma cuando estaba cansado y afligido.

A mis maestros que impartieron su conocimiento y a quienes me enseñaron a buscar soluciones óptimas para llegar a resolver problemas, y de manera especial a Alfredo Calderón, Oswaldo Espinosa y Anita Urgilés, quienes me ayudarón a terminar esta tesis de una manera simple y directa.

De igual manera a mi novia Bato que participa directa o indirectamente en toda decisión que realizo y es mi apoyo indondicional.

¡Gracias a ustedes!

Att.

Carlos Andrés Guamán Guañuna.

AGRADECIMIENTOS

A la Pontificia Universidad Católica del Ecuador or darme la oportunidad de estudiar y ser un profesional.

A mis padres, quienes permanentemente me apoyaron con espíritu alentador, contribuyendo incondicionalmente a lograr las metas y objetivos propuestos.

A mis compañeros qienes me acompañaron en ese viaje de aprendizaje y conocimiento.

A los docentes que me han acompañado durante el largo camino, brindándome siempre su orientación con profesionalismo ético en la adquisición de conocimientos.

A mi y tutor y correctores quienes me han orientado en todo momento en la realización de este gran proyecto.

Muchas gracias

Att.

Jorge Manuel Recalde Mosquera.

ÍNDICE GENERAL

DECLARATORIA DE RESPONSABILIDAD 2

DEDICATORIA 3

DEDICATORIA 4

AGRADECIMIENTOS 5

AGRADECIMIENTOS 6

ÍNDICE GENERAL 7

ÍNDICE DE TABLAS 13

1 CAPÍTULO 1: SITUACIÓN ACTUAL 14

1.1 Introducción. 14
1.2 Misión y visión de la fundación 14
1.2.1 Misión: *14*
1.2.2 Visión: *15*
1.3 Debilidades y fortalezas de la fundación 15
1.3.1 Fortalezas *15*
1.3.2 Debilidades *15*
1.4 Servicios que ofrece la fundación 16
1.5 Antecedentes del problema 16

2 CAPÍTULO 2: FUNDAMENTOS TEÓRICOS 18

2.1 Técnicas de desarrollo de aplicaciones 18
2.1.1 CASE (Ingeniería de Software Asistida por Computadora). *18*
2.1.2 I-CASE (CASE integrado). *19*
2.1.3 Programación Visual. *19*
2.1.4 Ingeniería de la Información. *19*
2.1.5 Bases de datos orientados a objetos. *20*
2.1.6 Lenguajes no procedimentales. *20*
2.1.7 Motor de Inferencias. *20*
2.1.8 Tecnología Cliente-Servidor. *21*
2.1.9 Biblioteca de clases. *21*
2.1.10 Análisis y diseño orientado a objetos. *21*
2.2 Metodologías Ágiles de aplicaciones 22
2.2.1 Extreme Programming (XP) *22*
2.2.2 ¿Qué es XP? *23*
2.2.2.1 Iteraciones cortas 23
2.2.2.2 Planificación de desarrollo flexible 23
2.2.2.3 Pruebas automatizadas 24
2.2.2.4 Diseño evolutivo 24
2.2.2.5 Comunicación Oral 24
2.2.2.6 Trabajo colaborativo 24
2.3 RAD (Diseño Rápido De Aplicaciones) 25
2.3.1 Definición De RAD *25*
2.3.2 Principios de la existencia de la definición *25*
2.3.3 Problemas abordados por RAD *25*

2.3.4 ¿Por qué utilizar RAD? *26*
2.3.5 CALENDARIO frente ECONOMÍA frente CALIDAD DEL PRODUCTO *26*
2.3.6 Historia abreviada de RAD *27*
2.3.7 Características del RAD *28*
2.3.8 Limitaciones importantes de RAD *30*
2.3.9 Cuando trabaja RAD *30*
2.3.10 Evaluación de RAD *31*
2.4 SELECCIÓN DE HERRAMIENTAS DE DESARROLLO 33
2.4.1 Estudio de herramientas de ayuda de desarrollo *33*
2.4.2 Consideraciones en el para la elección de la herramienta de desarrollo *34*
2.5 BASES DE DATOS 35
2.5.1 SQL Server Compact *35*
2.5.1.1 Integración con Visual Studio 35
2.5.1.2 Explorador de servidores y Diseñador de consultas 35
2.5.1.3 Editor de Transact-SQL 36
2.5.1.4 ClickOnce Implementación para aplicaciones administradas 36
2.6 LENGUAJES DE PROGRAMACIÓN 36
2.6.1 Definición *36*
2.6.2 Visual Studio 2010 Como Herramienta CASE *37*
2.6.3 Características De Visual Studio *37*
2.6.3.1 Entorno de desarrollo integrado 37
2.6.3.2 Compatibilidad con la plataforma de desarrollo 37
2.6.3.3 Herramientas de prueba 37
2.6.3.4 Ayuda en Línea de Microsoft MSDN 38
2.6.3.5 Ayuda de Colaboración SharePoint 38
2.6.3.6 Cree aplicaciones de RIA y WPF sin complicaciones 38
2.6.3.7 Simplifique la implementación de aplicaciones web 38
2.6.3.8 REQUISITOS DE SOFTWARE 39
2.6.3.9 REQUISITOS DE HARDWARE 39
2.7 SERVIDOR DE APLICACIONES 40
2.7.1 Three-Tier Client / Server *40*
2.7.2 Servidores de Aplicaciones y Servidores Web *41*
2.7.3 Java EE Application Server *42*
2.8 ARQUITECTURA DE APLICACIONES 43
2.8.1 Definición: *43*
2.8.2 Tipos de Arquitecturas *43*
2.8.2.1 Arquitectura Centralizada 43
..... 43
2.8.2.2 Arquitectura Distribuida 43
2.8.2.3 Cliente Servidor 44
2.8.2.3.1 Definición 44
2.8.2.3.2 Funcionamiento del modelo cliente/servidor 44
2.8.2.3.3 Características del modelo cliente/servidor 45
2.8.2.4 Ventajas y desventajas del modelo cliente/servidor 2 Capas 45
2.9 POR CAPAS 46
õ *Capa de Datos* *46*
õ *Capa de negocio* *46*
õ *Capa de presentación* *47*
3. CAPÍTULO 3: CONSTRUCCIÓN DEL PRODUCTO **48**

3.1 ESTUDIO DE FACTIBILIDAD 48
3.1.1 FACTIBILIDAD OPERACIONAL *48*
3.1.2 FACTIBILIDAD TÉCNICA *48*
3.2 REQUERIMIENTOS DEL SISTEMA 50
3.2.1 REQUERIMIENTOS FUNCIONALES *52*
3.2.2 REQUERIMIENTOS NO FUNCIONALES *53*
3.3 LEVANTAMIENTO DE LA INFORMACIÓN 55
3.3.1 ENTREVISTA *55*
3.3.2 OBSERVACIÓN *55*
3.4 ANÁLISIS DEL SISTEMA 56
3.4.1 FASE DE INSPECCIÓN *56*
3.4.2 FASE DE ESTUDIO *57*
3.4.3 FASE DE DEFINICIÓN *58*
3.5 DISEÑO DEL SISTEMA 58
3.5.1 DISEÑO DE LOS DATOS *58*
3.5.2 DISEÑO ARQUITECTÓNICO *60*
3.5.3 DISEÑO DE LA INTERFAZ *60*
3.5.4 DISEÑO DE PROCEDIMIENTOS *61*
3.5.4.1 MAPA DE PROCESOS *63*
3.5.4.2 SIPOC *67*
3.6 IMPLEMENTACIÓN 70
3.6.1 PREPARACIÓN DEL ENTORNO OPERATIVO *70*
3.6.2 DEFINICIÓN DE POLÍTICAS DE BACKUP *71*
3.6.3 INSTALACIÓN DE ENTORNO DE DESARROLLO Y DE PRUEBA *72*
3.6.4 PARAMETRIZACIÓN PRELIMINAR *74*
3.6.5 DESARROLLO DE CUSTOMIZACIONES E INTERFACES *75*
3.6.6 DESARROLLO DE PROGRAMAS PARA MIGRACIÓN DE ARCHIVOS MAESTROS Y PRUEBAS DE MIGRACIÓN *77*
3.6.7 SIMULACIÓN PRELIMINAR DEL ENLATADO EN PRODUCCIÓN *78*
3.6.8 CAPACITACIÓN DE USUARIOS FINALES *80*
3.6.9 AJUSTES DE PARAMETRIZACIÓN, CUSTOMIZACIONES E INFORMES *81*
3.6.10 SIMULACIÓN FINAL DE ENTORNO DE PRODUCCIÓN *82*
3.6.11 HABILITACIÓN DEL ENTORNO DE PRODUCCIÓN *84*
3.7 PRUEBAS 84
3.7.1 PRUEBAS ALFA *84*
3.7.2 PRUEBAS BETA *85*

4. CAPÍTULO 4: MANUAL TÉCNICO 87

4.1 ESTRUCTURA DE BASE DE DATOS 87
4.2 DETALLE DE LAS TABLAS (CAMPOS, TIPO DE DATOS) 89
4.3 EXPLICACIÓN DE LA RELACIÓN ENTRE TABLAS 96
4.4 CÓDIGO DE FUENTES DOCUMENTADOS EN MEDIO MAGNÉTICO 97
4.5 DOCUMENTOS DE DEFINICIÓN INICIAL Y TODAS LAS MODIFICACIONES SOLICITADAS Y REALIZADAS 97

CONCLUSIONES: 99

RECOMENDACIONES: 100

ANEXOS 102

ANEXO 1: PRUEBAS ALFA 102
ANEXO 2: PRUEBAS BETA 108
ANEXO 3: CÓDIGO FUENTE DOCUMENTADO 110
BASE DE DATOS *110*
CREACION DE TABLAS *110*
CREACION DE PROCEDIMIENTOS ALMACENADOS *126*
CREACION DE VISTAS *140*
FORMULARIOS *157*
LOGIN *157*
PRINCIPAL *159*
ACOMPAÑANTE *164*
AGREGARARTICULODONACION *166*
ARTICULO *172*
CAMA *175*
DIARIO *179*
DONACIÓNARTÍCULO *179*
DONACIÓNDINERO *182*
DONADOR *186*
HABITACIÓN *189*
LUGAR *191*
OPCIÓNIMPRIMIR *192*
PACIENTE *194*
PAGO *197*
TIPO CAMA *200*
TRANSACCIÓN *202*
VER ACOMPAÑANTE *202*
VER ARTICULO *203*
VERARTÍCULODONACIÓN *205*
VER CAMA *206*
VERCAMADISPONIBLE *206*
VERDEVOLUCIONPAGO *207*
VERDONACIÓNARTÍCULO *208*
VERDONACIÓNDINERO *209*
VERDONADOR *211*
VER HABITACION *212*
VER LUGAR *213*
VER PACIENTE *214*
VERPACIENTEDISPONIBLE *215*
VER PAGO *216*
VER TIPOCAMA *219*
VERUSUARIO *220*
VISTAPREVIA *221*
CLASES *224*
CONTROLUSUARIO *225*
IMPRIMIRDGV *233*
LISTAVISTAIMAGEN *240*
VISTAPREVIACONTROL *242*
ANEXO 4: MANUAL DE USUARIO 251

MANUAL DE USUARIO *251*
INGRESO AL SISTEMA *251*
Ingreso de nombre de usuario y contraseña *251*
Ingreso como ADMINISTRADOR *253*
Ingreso como TÉCNICO *254*
Ingreso como USUARIO *256*
ADMINISTRACIONES *257*
Búsqueda de donador *257*
Creación de Donador *258*
Edición de Donador *262*
Eliminación de Donador *264*
Búsqueda de Usuario *266*
Creación de Usuario *267*
Edición de Usuario *269*
Eliminación de Usuario *270*
PROCESOS *272*
Búsqueda de Donaciones de dinero *272*
Crear Donación de Dinero a FUDIS *274*
Crear Donación de dinero a otras instituciones *281*
Crear Donación de Artículos a FUDIS *284*
Crear Donación de Artículos a otras Instituciones *295*
Editar donaciones *301*
Eliminar donaciones *305*
Diario *308*
Impresión Diario *309*
Inventario *313*

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Servidor de Aplicaciones 40
Ilustración 2 WebSphere Application Server 41
Ilustración 3 Servidores de Aplicaciones y Servidores Web 42
Ilustración 4 Arquitectura Centralizada 43
Ilustración 5 Arquitectura Cliente/Servidor 44
Ilustración 6 Modelo Cliente Servidor 45
Ilustración 7 Identificación de actores 50
Ilustración 8 Identificación de Casos de Uso 52
Ilustración 9 Diseño de Datos 59
Ilustración 10 Diseño Arquitectónico 61
Ilustración 11 Cadena de Valor 63
Ilustración 12 Mapa de Procesos Donación 64
Ilustración 13 Mapa de Procesos Ingreso de Pacientes 65
Ilustración 14 Mapa de Proceso Devolución 66
Ilustración 15 Diagrama SIPOC Ingreso Diario de Pacientes 67
Ilustración 16 Diagrama SIPOC Donación 68
Ilustración 17 Diagrama SIPOC Devolución 69
Ilustración 18 Entorno de Desarrollo 72
Ilustración 19 Instalación IDE 73
Ilustración 20 Instalación IDE 1 73
Ilustración 21 Instalación BDD 74
Ilustración 22Interface Principal 75
Ilustración 23 Interface Secundaria 76
Ilustración 24 Interface Elaborada 76
Ilustración 25 Pruebas de Migración 78
Ilustración 26 Diagrama conceptual 87
Ilustración 27 Diagrama Físico 88

ÍNDICE DE TABLAS

Tabla 1 Selección de Herramientas de Desarrollo 35
Tabla 2 Factibilidad Técnica 49
Tabla 3 Identificación de escenarios 51
Tabla 4 Requerimientos no funcionales 54
Tabla 5 Fase de Inspección 58
Tabla 6 Simulación Preliminar 80
Tabla 7 Capacitación de Usuarios Finales 80
Tabla 8 Ajustes de Parametrización 82
Tabla 9 Simulación Final 84
Tabla 10 Lista de Tablas 89
Tabla 11 Lista de Columnas ACOMPAÑANTE 89
Tabla 12 Lista de Columnas ARTICULO 90
Tabla 13 Lista de Columnas CAMA 90
Tabla 14 Lista de Columnas DONACION 91
Tabla 15 Lista de Columnas DONADOR 91
Tabla 16 Lista de Columnas HABITACION 92
Tabla 17 Lista de Columnas LUGAR 92
Tabla 18 Lista de Columnas PACIENTE 93
Tabla 19 Lista de Columnas TIPOCAMA 94
Tabla 20 Lista de Columnas TIPODONACION 94
Tabla 21 Lista de Columnas TIPOTRANSACCION 94
Tabla 22 Lista de Columnas TRANSACCION 95
Tabla 23 Relación entre Tablas 96
Tabla 24 Prueba alfa ADMINISTRADOR 1 103
Tabla 25 Prueba alfa ADMINISTRADOR 2 104
Tabla 26 Prueba alfa TECNICO 106
Tabla 27 Prueba alfa USUARIO 107
Tabla 28 Pruebas beta Todos los Actores 110

1 Capítulo 1: Situación Actual

1.1 Introducción.

FUDIS- Fundación Ecuatoriana Para el Desarrollo Integral y Solidario con el Discapacitado y su programa Amigos del Mundo Frente al Cáncer, entidad sin fines de lucro con Personería Jurídica No. 492 de 1995, registrada en el Consejo Nacional de Discapacidades CONADIS, MIEMBRO DE LA Red Ibero Americana de Cooperación Técnica, afiliada a la Federación de ONG's núcleo de Pichincha y registrada en el SRI con RUC 1791776445001.

A partir de enero de 1999, basados en el conocimiento de las difíciles circunstancias por las que deben atravesar los pacientes provenientes de provincia para el tratamiento de una enfermedad de tan elevada incidencia y difíciles secuelas, como es el cáncer, y amparados por los Estatutos, iniciamos el Programa AMIGOS DEL MUNDO FRENTE AL CÁNCER, brindando albergue, alimentación, apoyo psicológico, asistencia espiritual, prevención y seguimiento a pacientes con cáncer procedentes de provincia, que deben trasladarse hasta Quito para realizarse tratamientos en situaciones de alto riesgo, vulnerabilidad y extrema pobreza.

1.2 Misión y visión de la fundación

1.2.1 Misión:

Solidarizarnos con el paciente discapacitado y/o oncológico de gran vulnerabilidad y elevado riesgo, así como con su familia, a través de nuestro programa Amigos del Mundo Frente al Cáncer brindándoles servicios básicos de calidad y a bajo costo que les ayuden a sobrellevar la difícil situación de salud durante el tiempo que requieran desplazarse para tratamiento médico especializado y que dure la separación de su grupo familiar.

1.2.2 Visión:

Nos proponemos contribuir de manera sostenible al desarrollo integral y solidario de nuestra población beneficiaria, garantizando servicios oportunos, necesarios, de alta calidad, en unas instalaciones físicas cómodas que brinden bienestar al paciente durante su permanencia, a fin de promover el mejor resultado en su tratamiento médico especializado, constituyéndonos como una institución de respuesta permanente ante la elevada incidencia del cáncer y sus secuelas en mujeres, hombres, jóvenes y niños con discapacidades por extirpación o amputación parcial o total de miembros y órganos, deficiencias visuales y renales, parálisis, disminución de la capacidad mental, entre otras secuelas y sus efectos negativos en su entorno familiar y social.

1.3 Debilidades y fortalezas de la fundación

1.3.1 Fortalezas

- Si bien el área de influencia de la fundación es la ciudad de Quito, todas las acciones y objetivos van encaminadas a atender las necesidades en términos de alojamiento, alimentación y apoyo psico-social, terapéutico y espiritual de los habitantes aquejados por diagnóstico de cáncer o discapacidad, procedentes de las diversas provincias del país, con gran frecuencia estamos acogiendo pacientes de regiones como La Joya de los Sachas, de Sucumbíos, Napo, Tena, Lago Agrio, Puyo, El oro, Bolívar, Riobamba, Tungurahua, Carchi, Loja, Imbabura, Manabí, Guayas, Esmeraldas, Santo Domingo y de la zona rural de la provincia de Pichincha.
- Las personas que se alojan en el albergue tienen derecho a un trato digno, igualitario, fraterno, sin distinción de condición económica, social, cultural, étnica, sexo, procedencia, religión y edad. Además tienen derecho a una cama cómoda con sábanas y cobertores limpios, y agua caliente. Además tienen derecho a una cama cómoda con sábanas y cobertores limpios, agua caliente

1.3.2 Debilidades

- Actualmente en el Ecuador y de acuerdo con los datos del INEC, existe una alta incidencia de pacientes con diagnóstico de cáncer, de los cuales un alto porcentaje

reside fuera de la ciudad de Quito; constituyéndose en una demanda potencial, de la cual un 10% está siendo atendida actualmente por FUDIS, demanda ante la cual resulta escaso nuestro espacio físico en comparación a la cantidad de solicitudes.

- El hospedaje en el albergue tiene un costo diario al momento de USD 4,00 el cual se irá ajustando de acuerdo a la realidad económica del país, el costo es por persona en cuartos compartidos. Hay habitaciones de mayor costo en caso de alojamiento a enfermos sujetos a tratamientos de yodo radioactivo.

1.4 Servicios que ofrece la fundación

Albergue y protección
Recuperación nutricional
Servicios administrativos y logísticos
Servicios básicos en salud y Pastoral de la Salud
Banco de medicamentos
Apoyo psicológico
Apoyo espiritual
Terapias ocupacionales (labores manuales varias, recreación)
Capacitación para el voluntariado
Ropero Amigo

1.5 Antecedentes del problema

Conscientes del problema que se presentan muchas veces en la mayoría de las Fundaciones similares o afines de la Ciudad en cuanto a la optimización, control y administración de sus ingresos y bienes nosotros como estudiantes de Ingeniería en Sistemas de la PUCE, hemos hecho énfasis especialmente en esta fundación, dedicada a diversos aspectos como lo son: *Albergue y protección, Recuperación nutricional, Servicios administrativos y logísticos, Servicios básicos en salud y Pastoral de la Salud, Banco de medicamentos entre otras,* cuyo nombre es, ***FUNDACIÓN FUDIS – ALBERGUE DE AMIGOS DEL MUNDO FRENTE AL CÁNCER***. Dentro de la cual hemos podido evidenciar que existe la pérdida de algunos productos que están en al servicio del albergue, algo que notamos que les pasa muy frecuente que por no llevar un control de los artículos de uso del albergue esto causa pérdida de recursos de la Fundación.

Adicionalmente la Fundación lleva el inventario de mercancía manualmente ya sea esta donada o comprada, capturando y archivando los recibos o facturas de aquellos productos que de utilización para los pacientes del albergue.

Además el registro diario de pacientes que ingresan al albergue se los hace de igual manera que el registro de inventario. Y los servicios que se realiza en el albergue tampoco tienen un tipo específico de control es por ello que hemos decidido realizar un análisis y brindar una propuesta en la construcción de un sistema automatizado de donaciones, registro diario de pacientes e inventario para que así tengan un mejor aprovechamiento de aquellos servicios brindados y puedan llevar un control de los artículos que poseen.

Para los problemas evidenciamos hemos realizado una investigación encaminada al desarrollo de la solución más óptima al problema, dentro del cual tenemos el reto que forma parte de nuestro desarrollo profesional que es llevar a cabo el desarrollo de un proyecto de análisis de un sistema donaciones, registro pacientes e inventario, que en estos momentos se hace de forma manual en la Fundación FUDIS.

El desarrollo y análisis de esta investigación y desarrollo enriquecen de un modo u otro al logro de muchos objetivos, los cuales representan aportes significativos a la Fundación, la comunidad y cada uno de nosotros como analistas. Aportes tales como la automatización de los procesos, contempla diversos aspectos como lo son la rapidez, eficiencia y mejor administración de los recursos disponibles en la Fundación conllevan sin duda al logro de muchas metas establecidas y a un mejor servicio brindado por la fundación, de modo que el usuario se sienta satisfecho por una excelente atención, lo cual es sin duda un aporte del sistema a la Fundación.

2 Capítulo 2: Fundamentos Teóricos

2.1 Técnicas de desarrollo de aplicaciones

Las técnicas de desarrollo de aplicaciones nos proporciona una combinación sinérgica de tecnologías trae una revolución en el software de aplicaciones. Esto necesitará un equipo de trabajo, educación y una buena administración, junto con la cooperación de las compañías de software. Los estándares actuales ayudarán a esta cooperación.

a. CASE o I-CASE.
b. Programación Visual.
c. Generadores de Código.
d. Repositorios y Coordinadores de Repositorios.
e. Metodologías basadas en Repositorios.
f. Ingeniería de la Información.
g. Bases de Datos Orientadas a Objetos.
h. Lenguajes no procedimentales.
i. Motores de inferencias.
j. Tecnologías Cliente-Servidor.
k. Biblioteca de clases que maximicen la reutilización.
l. Análisis y Diseño orientado a objetos.

2.1.1 CASE (Ingeniería de Software Asistida por Computadora).

El crecimiento de la industria CASE reviste una particular importancia para el desarrollo de aplicaciones. Las herramientas CASE son el equivalente para la industria del software de las herramientas CAD (diseño asistido por computadora) para actividades como la organización de un circuito y el diseño de chips.

Las herramientas CASE utilizan representaciones gráficas en la pantalla para ayudar a automatizar la planeación, el análisis, el diseño y la generación de aplicaciones.

2.1.2 I-CASE (CASE integrado).

Una capacidad importante de las herramientas CASE es el generador de códigos. Los generadores de código deben ser de lo más poderoso y de lo más eficiente posible. El término I-CASE se utiliza para indicar el CASE integrado, en el cual se ligan entre sí las herramientas de todas las etapas del ciclo vital y controlan un generador de códigos.

I-CASE se refiere a un conjunto de herramientas CASE que soportan todas las etapas del ciclo vital, incluyendo toda la generación de códigos, con un único repositorio lógicamente consistente

2.1.3 Programación Visual.

La programación visual es una forma de CASE que expresa el diseño de programas con gráficos, colores y hasta sonido. Los objetos se representan en forma visual y se pueden considerar como máquinas físicas que pasan de un estado a otro.

La Programación Visual permite a los diseñadores de aplicaciones introducir, comprender, reflexionar, hacer pruebas y controlar programas en la máquina mediante notaciones pictóricas.

2.1.4 Ingeniería de la Información.

La ingeniería de la información aplica a toda una empresa el desarrollo basado en repositorios, para integrar la planeación, el diseño y la construcción de sistemas que necesiten interactuar a través de dicha empresa. Crea un modelo de ésta e intenta rediseñar los sistemas de información de la misma, de modo que sean lo más eficientes posible.

La Ingeniería de la Información aplica las técnicas de modelamiento y diseño integral aplicadas a diversas empresas como un todo (o a una gran parte de la empresa) en vez de solamente a un proyecto.

2.1.5 Bases de datos orientados a objetos.

Una base de datos orientado a objetos es una base de datos inteligente. Soporta el paradigma orientado a objetos almacenando datos y métodos, y no sólo datos. Está diseñada para ser eficaz, desde el punto de vista físico, para almacenar objetos complejos. Evita el acceso a los datos si esto no se realiza mediante los métodos almacenados en ella.

Una base de datos orientada a objetos está diseñada para almacenar datos y métodos objeto mediante técnicas adecuadas para el procesamiento orientado a objetos

2.1.6 Lenguajes no procedimentales.

Con los lenguajes procedimentales (como COBOL o C) indicamos a la computadora hasta que secuencia de operaciones debe ejecutar. Con los lenguajes no procedimentales, definimos los resultados necesarios. Un intérprete o compilador genera el código.

Los lenguajes no procedimentales definen lo que se desea y no la forma como se programe. Muchos de los lenguajes de cuarta generación (como Focus o Natural) contienen elementos procedimentales como no procedimentales.

2.1.7 Motor de Inferencias.

Un motor de inferencias actúa como un conjunto de reglas relativas a un área de conocimiento. Selecciona reglas y las concatena de manera eficaz para llevar a cabo un razonamiento inferencial. Puede utilizar la concatenación hacia adelante (razonamiento controlado por la entrada), hacia atrás (controlado por los objetivos) o ambos. Permite que una computadora realice deducciones complejas sin un programa de aplicación. Esta es la principal técnica utilizada en el software de inteligencia artificial.

Un motor de inferencias es un software que hace deducciones a partir de hechos y reglas mediante las técnicas de inferencia lógica.

2.1.8 Tecnología Cliente-Servidor.

El mundo de las computadoras requiere cada vez más un software que se puede ejecutar en varias computadoras, como los sistemas integrados por redes, sistemas cooperativos, cómputo distribuido y computadoras paralelas. El cómputo cliente-servidor se puede describir mediante una relación del software, del modo siguiente:

i. Un cliente es un módulo en software que solicita una operación.
ii. Un servidor es un módulo en software que responde a dicha solicitud.
iii. Los clientes y los servidores se pueden ejecutar en máquinas independientes ordenadas mediante una configuración arbitraria.

2.1.9 Biblioteca de clases.

Una biblioteca de clases contiene la implantación reutilizable de tipos de objetos. Su intención es la de lograr el máximo grado de reutilización en el desarrollo de software. El software para las bibliotecas de clases debe ayudar a los diseñadores a encontrar, adaptar y utilizar las clases que necesiten.

Una clase es la implantación de un tipo de objeto. Una clase puede tener muchas subclases.

2.1.10 Análisis y diseño orientado a objetos.

El analista orientado a objetos ve el mundo como objetos (con estructuras de datos y métodos) y eventos que activan operaciones, el cual modifican el estado de los objetos. Las operaciones aparecen como objetos que hacen solicitudes a otros objetos. El analista crea diagramas de la estructura de los objetos y de los eventos que los modifican. El modelo del diseñador es similar al modelo del analista, pero se toma con el detalle suficiente como para crear el código.

El análisis y diseño orientado a objetos intenta lograr la reutilización masiva de las clases de objetos.

El análisis orientado a objetos modela al mundo en términos de objetos que tienen prioridades y comportamiento, y eventos que activan operaciones que modifican el estado de los objetos. Los objetos interactúan de manera formal con otros objetos.

El modelado y el diseño orientado a objetos es un paradigma de integración, que entrelazará todas las herramientas y las técnicas poderosas para la creación de software para el desarrollo de aplicaciones.[1]

2.2 Metodologías Ágiles de aplicaciones

2.2.1 Extreme Programming (XP)

Extreme Programming es una metodología creada por Kent Beck. Su origen se fija en el momento en el que se publica el libro "*Extreme Programming Explained"* (Octubre de 1999).

No es correcto decir que es una metodología que se basa en el Manifiesto Ágil puesto que su aparición es anterior al mismo. A pesar de ello XP contempla todos los principios y valores ágiles.

En marzo de 1996, se llamó a Kent Beck para mejorar el rendimiento del proyecto C3:"Payroll Project" de la compañía Chrysler Comprehensive Compensation System.

Fruto de este trabajo surgió la metodología XP. Kent Beck introdujo nuevos métodos y refinó otros, mejorando los resultados obtenidos. A pesar de ello, el proyecto fue cancelado en febrero de 2000, lo que constituyó el principal motivo que esgrimieron los detractores de XP para justificar su escepticismo sobre la misma. Según el propio Kent Beck las razones de su cancelación fueron dos: la tecnología y los fondos. Era un sistema escrito en Smalltalk, en el año 2000. El lenguaje dominante ya era Java y Chrysler no deseaba mantener indefinidamente un proyecto en un lenguaje poco usado. Además, la empresa pidió financiación para sustentar posteriores fases del proyecto. El financiamiento fue denegado y por lo tanto el proyecto fue cancelado.

No obstante, Kent Beck asegura que el resultado de este proyecto piloto fue un sistema confiable, barato, fácil de mantener y escalable. Opina que la arquitectura se mantuvo fluida y ajustada a las necesidades. El nivel de defectos fue increíblemente bajo comparado con proyectos similares. En definitiva, afirma que desde el punto de vista técnico el proyecto fue un rotundo éxito.

[1] (ALIPSO, 2000)

Desde sus orígenes hasta la actualidad XP ha evolucionado para adaptarse mejor a las necesidades de los proyectos de hoy en día, hasta tal punto que es una de las metodologías ágiles más utilizadas en la industria.

2.2.2 ¿Qué es XP?

Parafraseando a Kent, XP implica un cambio social. Es necesario bajar las defensas que nos protegen pero que al mismo tiempo interfieren en nuestra productividad. Se tiene presente que las buenas relaciones conducen a buenos negocios. Se ponen al mismo nivel las técnicas empleadas y las buenas relaciones ya que ambas, en conjunción, conducen al éxito.

En XP se pretende reconciliar el aspecto humano del desarrollo del software con la productividad. No por la componente humana en sí misma, sino porque cuanto mejor sea el trato entre personas de un mismo equipo, cuanto mejor sea el ambiente de trabajo, mayores cotas de productividad serán alcanzadas.

XP se basa en 5 valores que todo equipo debe tratar de potenciar al aplicar esta metodología: comunicación, retroalimentación, simplicidad, valor, y respeto.

Además XP define un conjunto de principios o técnicas intelectuales mediante las cuales llevar los valores a la práctica.

Desde el punto de vista técnico eXtreme Programming se caracteriza por:

2.2.2.1 Iteraciones cortas

Al final de cada ciclo se debe tener una versión del sistema que ha de ser sometida a la aceptación del cliente. De esta manera, si el sistema no es aceptado, los recursos invertidos desde la última versión validada son menores.

2.2.2.2 Planificación de desarrollo flexible

La planificación flexible del desarrollo permite responder mejor a los cambios en las necesidades del cliente.

2.2.2.3 Pruebas automatizadas

La realización de pruebas ha de ser un proceso automático (código probando código).

De tal manera que cada nueva versión debe pasar toda la batería de pruebas definida hasta el momento. Es recomendable que estén escritas las pruebas antes que se implemente la funcionalidad, puesto que agiliza su desarrollo.

2.2.2.4 Diseño evolutivo

XP pretende entregar valor al cliente de forma rápida y continua. Por lo tanto no se ha de invertir tiempo diseñando para necesidades que aún no se tienen. Puesto que existe la posibilidad de que nunca lleguen a necesitarse (filosofía YAGNI).

2.2.2.5 Comunicación Oral

Con una conversación cara a cara se transmite mucha mayor información y con mayor velocidad que con un documento escrito. Por lo que se dinamiza el ritmo de un proyecto.

2.2.2.6 Trabajo colaborativo

En un entorno en el que las necesidades del cliente son vagas o cambiantes se hace muy importante la presencia/disponibilidad del cliente/usuario con capacidad para la toma de decisiones. Por otra parte, el trabajo colaborativo facilita que los miembros del equipo realicen o asistan aquellas tareas que se adapten mejor a sus cualidades.

En la primera edición del libro *"eXtreme Programming Explained: Embrace Change",* Kent Beck define XP como: "Una metodología ligera para equipos de desarrollo software de tamaño pequeño-mediano2, capaz de afrontar requisitos cambiantes o vagos". En la segunda edición del libro (Noviembre de 2004), sostiene que los valores y principios que están detrás de XP son aplicables a cualquier escala, por lo tanto, equipos de cualquier tamaño pueden trabajar con esta metodología. Si bien es cierto, hoy por hoy, se cuenta con estudios empíricos que dicen lo contrario.

Se ha de tener presente que XP es una metodología que aborda aspectos relacionados con el desarrollo de software. Por lo tanto XP no considera directamente temas como: gestión de documentos, justificaciones de gastos, operaciones, marketing, ventas, etc.

En definitiva XP se define como una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.[2]

2.3 RAD (Diseño Rápido De Aplicaciones)

2.3.1 Definición De RAD

Es un proceso de desarrollo de software que permite a los sistemas utilizables que se construirá en tan sólo 60 a 90 días, a menudo con algunos compromisos

2.3.2 Principios de la existencia de la definición

a. En ciertas situaciones, una solución utilizable 80% se puede producir en20% de las veces que se habría requerido para producir un total de solución.
b. En ciertas situaciones, los requisitos de negocio para un sistema pueden ser completamente satisfechas, aunque algunas de sus necesidades operativas no están satisfechos.
c. En ciertas situaciones, la aceptabilidad de un sistema puede evaluarse contra el útil conjunto mínimo acordado de requisitos en lugar de todos los requisitos.

2.3.3 *Problemas abordados por RAD*

a. Con los métodos convencionales, hay una larga demora antes de que el cliente llega a ver ningún resultado.

[2] (Castillo, 2008)

b. Con los métodos convencionales, el desarrollo puede tomar tanto tiempo que el negocio del cliente ha cambiado de manera fundamental por el tiempo que el sistema está listo para su uso.
c. Con los métodos convencionales, no hay nada hasta el 100% del proceso está terminado, entonces el 100% del software se entrega.

2.3.4 *¿Por qué utilizar RAD?*

a. Malas razones para usar RAD

i. Para evitar los excesos de costes (RAD necesita un equipo ya disciplinado en la gestión de costes).
ii. Para evitar que los horarios de fugitivos (RAD necesita un equipo ya disciplinado en la gestión del tiempo).

b. Buenas razones para usar RAD

i. A converger hacia principios de un diseño aceptable para el cliente y factible para los desarrolladores.
ii. Limitar la exposición de un proyecto para las fuerzas del cambio.
iii. Para ahorrar tiempo de desarrollo, posiblemente a expensas de la economía o la calidad del producto.

2.3.5 *CALENDARIO frente ECONOMÍA frente CALIDAD DEL PRODUCTO*

a. Compensaciones determinan el ritmo de desarrollo.

i. *Desarrollo Eficiente equilibra la economía, la programación y la calidad*
   1. Programa - más rápido que el promedio.
   2. Economía - cuesta menos que la media.
   3. Producto - mejor que la calidad media.
ii. *Sensible RAD se inclina lejos de la economía y la calidad hacia el programa más rápido*
   1. Programa - mucho más rápido que el promedio.
   2. Economía - cuesta un poco menos que el promedio.
   3. Producto - un poco mejor que la calidad media.
iii. *Todo Salida RAD "código como el infierno"*
   1. Programa - más rápida posible.
   2. Economía - cuesta más que la media.

3. Producto - peor que la calidad media.

b. Para RAD, algo distinto de lo planificado debe ser negociable.

i. *RAD tiene una buena oportunidad de éxito si el cliente va a negociar ya sea la economía o de la calidad.*

ii. *RAD tiene más posibilidades de éxito si el cliente va a negociar la economía y la calidad.*

iii. *Negociar la calidad no significa aceptar una mayor tasa de efectos. Significa aceptar un producto que es menos útil, menos totalmente equipado, o menos eficiente.*

c. Por lo tanto, con RAD, uno o más de los siguientes objetivos puede ser inalcanzable.

i. *El menor número de defectos posibles (ya que los desarrolladores no pueden tener el derecho legal de modificar el origen de algunos componentes enchufables).*

ii. *El nivel más alto posible de satisfacción del cliente (porque los requisitos secundarios pueden ser sacrificados para permanecer en horario).*

iii. *Los costes de desarrollo más bajos (debido a la compra de componentes reutilizables puede costar más que la construcción).*

2.3.6 *Historia abreviada de RAD*

a. Barry Boehm (modelo espiral)
b. Tom Gilb (ciclo de vida evolutivo)
c. De Scott Shultz (RIPP, rápido iterativo de prototipos productiva)
d. James Martin (RAD, alrededor del año 1991)

2.3.7 *Características del RAD*

a. RAD usos equipos híbridos

i. Los equipos deben consistir en cerca de 6 personas, entre ellas dos desarrolladores y usuarios de tiempo completo del sistema, además de cualquier otra persona que tenga un interés en los requisitos.

ii. Desarrolladores elegidos para los equipos de RAD deben ser polifacéticos pueblo "renacimiento" que son analistas, diseñadores y programadores de todo en uno.

b. RAD utiliza herramientas especializadas que soporta

i. Desarrollo "visual".

ii. Creación de prototipos falsos (simulaciones puros).

iii. Creación de prototipos de trabajo.

iv. Múltiples idiomas.

v. Programación de equipo.

vi. Trabajo en equipo y la colaboración.

vii. El uso de componentes reutilizables.

viii. El uso de las API estándar.

ix. Control de versiones (porque muchas versiones se generarán).

c. RAD utiliza "Timeboxing"

Funciones secundarias se dejan caer como sea necesario para mantenerse en una fecha prevista.

d. RAD usos iterativos, prototipos evolutivos

i. *JAD (desarrollo conjunto de aplicaciones) Reunión de los usuarios finales y los diseñadores de alto nivel se reúnen en un brainstorming sesión para generar una lista aproximada de los requisitos iniciales.*

1. Desarrolladores hablar y escuchar.
2. Los clientes hablan y escuchan.

ii. *ITERATE hasta que esté hecho*

1. Los desarrolladores construyen / evolucionan prototipo basado en actuales requisitos.
2. Diseñadores revisar el prototipo.
3. Los clientes probar el prototipo, evolucionan sus necesidades.
4. ENFOQUE reunión del grupo de usuarios y desarrolladores se reúnen para revisar productos juntos, refinar los requisitos, generar solicitudes de cambio.
    a. Desarrolladores escuchan.
    b. Los clientes hablan.
5. Requisitos y solicitudes de cambio son "duración fija".
    a. Los cambios que no pueden ser acomodados dentro existentes time boxes son eliminados.
    b. Si es necesario para estar "en la caja", los requisitos secundarios se eliminan.

iii. *NOTAS*

1. Iteraciones requieren entre 1 día y 3 semanas.
2. En algún momento, los prototipos de exploración pueden convertirse en prototipos operativos.
3. Sesiones en grupos focales
    a. Duran alrededor de 2 horas.
    b. Son guiados por un facilitador experimentado, que mantiene el grupo "en foco".
        i. Por haber objetivos claros en cuanto al tipo de información que necesita ser despertado.
        ii. Mediante la elaboración de una agenda de temas concretos antes de la reunión.
        iii. Al asegurar que la discusión adecuada se dirige hacia cada tema.
        iv. Garantizando todo el mundo tiene una oportunidad adecuada para participar.
    c. Van seguidos de un informe del facilitador.

2.3.8 Limitaciones importantes de RAD

a. "Aptitud para un propósito comercial" debe ser el criterio de aceptación de los entregables.
b. Todos los grupos que pueden afectar a los requisitos de aplicación deben tener representación en el equipo de desarrollo de todo el proceso.
c. Los clientes, desarrolladores y la gestión deben aceptar informales entregables.
    i. *Prototipos de papel en vez de sistemas a gran escala.*
    ii. *Notas de los talleres de usuario en lugar de los requisitos formales de los documentos.*
    iii. *Notas de las reuniones de los diseñadores en lugar de diseño formales documentos.*
    iv. *PRINCIPIO: Crear la documentación mínima necesaria para facilitar el desarrollo y el mantenimiento futuro.*
d. Los equipos de desarrollo deben tener la facultad de tomar decisiones tradicionalmente dejadas a la gerencia.
e. De extremo a extremo escala de tiempo debe ser de 6 meses o menos.
f. Iteración debe ser utilizado de tal manera que el proceso de desarrollo converge hacia una solución de negocio aceptable.
g. Prototipos debe incorporar las nuevas necesidades rápidamente, en tiempo real, y obtener un consenso antes de tiempo.
h. Tiene que haber un sesgo de "comprar antes de construir".

2.3.9 *Cuando trabaja RAD*

a. RAD tiende a trabajar cuando:

i. La aplicación se ejecutará independiente.
ii. Mayor uso se puede hacer de bibliotecas de clases preexistentes (API).
iii. El rendimiento no es crítico.
iv. La distribución del producto será estrecha (interna o verticales de mercado).
v. El alcance del proyecto (macro-horario) se ve limitado.
vi. La fiabilidad no es crítica.

vii. El sistema puede ser dividido en varios módulos independientes.
viii. El producto está dirigido a una alta especialización IS (información de mercado de los sistemas).
ix. El proyecto tiene fuertes restricciones micro-programación (time boxes).
x. La tecnología requerida es de más de un año de edad.

b. RAD tiende a fallar cuando:

i. La aplicación debe interoperar con los programas existentes.
ii. Pocos componentes enchufables están disponibles.
iii. Se requiere un rendimiento óptimo.
iv. El desarrollo de productos no puede aprovechar la gama alta es herramientas (por ejemplo, 4GLs).
v. Distribución de productos será amplia (mercado horizontal o masa).
vi. RAD convierte QADAD (desarrollo de aplicaciones rápido y sucio).
vii. Se utilizan métodos RAD para construir sistemas operativos (fiabilidad objetivo demasiado alto para RAD), los juegos de ordenador (objetivo de rendimiento muy alto para RAD).
viii. Los riesgos técnicos son altos debido al uso del borde "sangrado" de la tecnología.
ix. El producto es una misión de vida o crítico.
x. El sistema no puede ser modular (derrotas paralelismo)

2.3.10 *Evaluación de RAD*

a. Ventajas de RAD

i. Compra puede ahorrar dinero en comparación con la construcción.
ii. Entregas a veces más fácil al puerto (porque hacen un mayor uso de las abstracciones de alto nivel, scripts, código intermedio).
iii. Desarrollo lleva a cabo a un nivel más alto de abstracción (porque las herramientas RAD operan a ese nivel).
iv. Visibilidad temprana (a causa de prototipos).
v. Mayor flexibilidad (ya que los desarrolladores pueden rediseñar casi a voluntad).

vi. Reduce en gran medida la codificación manual (a causa de asistentes, generadores de código, reutilización de código).

vii. Mayor participación de los usuarios (ya que están representados en el equipo en todo momento).

viii. Es posible que un menor número de defectos (porque las herramientas de CASE pueden generar gran parte del código).

ix. Posiblemente reducido costo (porque el tiempo es dinero, también a causa de reutilización).

x. Ciclos de desarrollo más cortos (porque el desarrollo se inclina hacia la programación y lejos de economía y calidad).

xi. Mirada estandarizada y la sensación (porque APIs y otros componentes reutilizables ofrecen una consistente apariencia).

b. Desventajas

i. Buying no puede ahorrar dinero en comparación con la construcción.

ii. Costo del conjunto de herramientas integrado y hardware para ejecutarlo.

iii. Más difícil de medir el progreso (porque no hay hitos clásicos).

iv. Menos eficiente (porque el código no está hecho a mano).

v. La pérdida de la precisión científica (porque no se utilizan métodos formales).

vi. Puede querer potenciar el retorno a las prácticas no controladas de los primeros días de desarrollo de software más defectos (debido al síndrome de "code-like-hell").

vii. Prototype no puede escalar, un gran problema.

viii. Características reducidas (por Timeboxing, reutilización de software).

ix. La dependencia de los componentes de terceros puede:

1. Sacrificar funcionalidad necesaria.
2. Agregar funcionalidad innecesaria.
3. Crear problemas legales.

x. Los requisitos pueden no converger (debido a que los intereses de los clientes y los desarrolladores pueden divergir de una iteración a la siguiente).

xi. Mirada estandarizada y la sensación (mediocre, mediocre aparición).

xii. Los esfuerzos exitosos difícil repetir (no hay dos proyectos se desarrollan de la misma manera).

xiii. Características no deseadas (a través de la reutilización de Componentes existentes).[3]

2.4 Selección De Herramientas De Desarrollo

2.4.1 Estudio de herramientas de ayuda de desarrollo

El análisis de las herramientas de ayuda para el desarrollo surge para poder satisfacer las necesidades y requisitos impuestos por el usuario final de los sistemas de información, requisitos tanto de calidad como de coste (mejora de la productividad).

La primera etapa que debe abordarse de modo sistemático, dentro de este proceso, es el análisis de las necesidades existentes, que deberán ser satisfechas a través de la implantación de la herramienta que se va a adquirir. Por ejemplo:

a. Los principales requisitos funcionales que debe cumplir la herramienta.
b. El tipo de facilidades de uso que deben prestar.
c. Las limitaciones y restricciones que se derivan del entorno de operación previsto.

En función de los requisitos funcionales se podrá deducir qué tipo de herramienta es la más adecuada.

Algunos factores a tener en cuenta y que son comunes a todas estas herramientas son:

a. Tipo(s) de plataforma(s) sobre las que deberá funcionar la herramienta, tanto desde el punto de vista del equipamiento lógico como del equipamiento físico.
b. Requisitos físicos (espacio en disco, memoria RAM, UCP, etc.).
c. Necesidad de integración con otras herramientas de ayuda al desarrollo ya existentes.
d. Necesidad de acceso simultáneo para diferentes usuarios. Esto puede enfocar la elección hacia una herramienta que permita accesos compartidos a los datos y que cuente con una definición de perfiles de usuario para la protección de información.

[3] (Manner, 1997)

e. Necesidad de compartir datos con aplicaciones externas. Se valorará más a aquella aplicación que permita exportar sus datos o que almacene la información en un formato de fácil acceso para otra aplicación.

2.4.2 Consideraciones en el para la elección de la herramienta de desarrollo

Nombre	Características	Área donde actúa
Estrategia de implantación	Se empieza a utilizar al principio del desarrollo de un proyecto piloto	Herramienta CASE
Requisitos físicos	Expresado en el Modelo de Tecnología de Arquitectura y características del puesto de desarrollo y características del puesto de producción	Procesador, memoria RAM y espacio en disco
Requisitos lógicos	Expresado en el Modelo de Tecnología que analiza con atención la necesidad de otros módulos, no incluidos en el producto ofertado por el vendedor	Compiladores, módulos para trabajo en grupo, etc.
Prueba en condiciones reales	La prueba se debe realizar en las condiciones más parecidas a las reales que se puedan conseguir e intentando simular el acceso de un número de usuarios, parecido al esperado.	Plataforma física y lógica y las necesidades funcionales
Dependencia del proveedor	Las herramientas llevan integradas partes de la plataforma operativa	La asesoría técnica, la capacitación y la información técnica.
Coste límite de elección	Oferta en cuanto a disponer de licencias individuales, grupos de licencia o licencias corporativas	Función del tipo de licencia
Coste de instalación	Instalación de un módulo propiedad del suministrador (Runtime)	Módulo distribuido entre los usuarios
Capacidad de integración	Plataformas diferentes en que va a ser instalada la herramienta en cuestión, su tipología (fabricante, modelo y sistema operativo	Elección de herramienta CASE y 4GL
Portabilidad	Instalar la aplicación generada en	Código generado según

	entornos técnicos diferentes: sistemas operativos, plataformas físicas, interfaces gráficas y protocolos de red	especificaciones del diseño

Tabla 1 Selección de Herramientas de Desarrollo

Fuente: Instituto Nacional de estadística e informática
Elaborado: Carlos Guamán y Jorge Recalde

2.5 BASES DE DATOS

2.5.1 SQL Server Compact

SQL Server Compact 4.0 es una base de datos compacta que se puede implementar en equipos e insertarse en aplicaciones web ASP.NET. SQL Server Compact 4.0 admite la misma sintaxis de SQL y los modelos de programación ADO.NET que otras ediciones de SQL Server. SQL Server Compact 4.0 se puede implementar en equipos mediante el instalador (SSCERuntime_x86-ENU.exe y SSCERuntime_x64-ENU.exe) o a través de Microsoft Visual Studio 2010 SP1.

La compatibilidad de desarrollo de SQL Server Compact 4.0 la proporciona Visual Studio 2010 SP1 o Microsoft Visual Web Developer 2010 SP1 Express para las aplicaciones de escritorio y web. Es posible obtener acceso a las bases de datos SQL Server Compact almacenadas en un equipo de escritorio mediante Visual Studio 2010 SP1 o Microsoft Visual Web Developer 2010 SP1 Express.

Algunas de las características distintivas de SQL Server Compact son las siguientes:

2.5.1.1 Integración con Visual Studio

Todos los componentes de SQL Server Compact que tienen que instalarse para permitir que SQL Server Compact 4.0 funcione con Visual Studio 2010 SP1 se pueden instalar con un clic mediante el instalador de plataformas web de Microsoft 3. Los diferentes diseñadores y sistemas de proyectos que pueden usarse para desarrollar aplicaciones para SQL Server Compact se detallan a continuación:

2.5.1.2 Explorador de servidores y Diseñador de consultas

Tras instalar los componentes de SQL Server Compact 4.0 para Visual Studio 2010 SP1, hay una nueva opción para conectarse a las bases de datos de SQL Server Compact 4.0 en el cuadro de diálogo Cambiar origen de datos. Esta opción se suma a la existente para conectarse a las bases de datos de SQL Server Compact 3.5. Tras

conectarse a la base de datos de SQL Server Compact 4.0, puede usar el Explorador de servidores para navegar a través de los objetos en el Explorador de servidores y el Diseñador de consultas para ejecutar las consultas.

2.5.1.3 Editor de Transact-SQL

SQL Server Compact 4.0 admite un variado subconjunto de la sintaxis de Transact-SQL que se podía usar en las demás ediciones de SQL Server. Puede acceder al editor de Transact-SQL en el menú Datos de Visual Studio 2010 y conectarse a una base de datos de SQL Server Compact 4.0 en el cuadro de diálogo Conectar con el servidor. Pueden agregarse campos al área de diseño especificando la ruta de acceso al archivo de base de datos o seleccionando la opción Nueva base de datos en el campo desplegable Archivo de base de datos.

2.5.1.4 ClickOnce Implementación para aplicaciones administradas

ClickOnce es una tecnología de instalación de software que SQL Server Compact admite para implementar las aplicaciones de escritorio administradas en los equipos. Para los administradores, la implementación o la actualización de una aplicación consisten en actualizar archivos en un servidor de implementación. No es necesario actualizar cada cliente por separado. Microsoft Visual Studio 2010 SP1 ofrece una compatibilidad completa para publicar y actualizar las aplicaciones que se implementan con ClickOnce. La implementación de ClickOnce está disponible para los proyectos de escritorio de Windows que se crean con Visual Basic, Visual C# y Visual J#, pero no para Visual C++.[4]

2.6 Lenguajes de Programación

2.6.1 Definición

Los lenguajes de programación son lenguajes creado por el ser humano para poder comunicarse con las computadoras. Así lenguaje de programación es el conjunto de símbolos y palabras que permiten al usuario de una computadora darle instrucciones y órdenes para que la computadora los ejecute.

[4] (Microsoft, 2013)

2.6.2 Visual Studio 2010 Como Herramienta CASE

Microsoft Visual Studio 2010 es la herramienta esencial para aquellas personas que realicen tareas básicas de desarrollo. Simplifica la creación, depuración e implementación de aplicaciones en diversas plataformas, incluidas SharePoint y Cloud. Visual Studio 2010 incorpora soporte integrado para el desarrollo mediante pruebas, así como herramientas de depuración que ayudan a garantizar la creación de soluciones de gran calidad.

2.6.3 Características De Visual Studio

Microsoft Visual Studio 2010 incluye una variedad de características que permite a los desarrolladores crear, depurar, realizar pruebas de unidad e implementar aplicaciones de alta calidad en diferentes plataformas, desde Windows, Internet, Cloud, Office y SharePoint, entre otros muchos.

2.6.3.1 Entorno de desarrollo integrado

Visual Studio 2010 proporciona compatibilidad con varios monitores, de modo que pueda organizar y administrar su trabajo como quiera. También puede dar rienda suelta a su creatividad utilizando los diseñadores visuales para mejorar las últimas plataformas, incluido Windows 7.

2.6.3.2 Compatibilidad con la plataforma de desarrollo

Tanto si un proyecto es sencillo como complejo, puede usar Visual Studio 2010 para plasmar su visión de la vida en una serie de plataformas que incluyen Windows, Windows Server, Web, Cloud, Office y SharePoint, entre otras.

2.6.3.3 Herramientas de prueba

Visual Studio 2010 Professional incluye capacidades para pruebas de unidades dentro del entorno de desarrollo integrado que pueden generar todos los componentes de métodos necesarios para compilar pruebas de unidades, lo que ayuda a garantizar que cada unidad de código funcione correctamente.

2.6.3.4 Ayuda en Línea de Microsoft MSDN

Visual Studio 2010 con MSDN Essentials es una suscripción de prueba de un año de duración que proporciona acceso a las últimas plataformas básicas de Microsoft (Windows 7 Ultimate, Windows Server 2008 R2 Enterprise y Microsoft SQL Server 2008 R2 Datacenter), además de proporcionar una mejor asistencia a través de la comunidad profesional. Visual Studio 2010 con MSDN Essentials se puede actualizar a Visual Studio 2010 Professional con MSDN al término del período de prueba.

Visual Studio 2010 Professional con MSDN es una suscripción que proporciona acceso tanto a las versiones actuales como a las anteriores de las principales plataformas de Microsoft. Esta suscripción completa incluye Visual Studio Team Foundation Server 2010, horas de utilización de Windows Azure, cursos de e-learning y mucho más.

2.6.3.5 Ayuda de Colaboración SharePoint

Introduzca herramientas de colaboración personalizada en su empresa aprovechando el nuevo soporte para el desarrollo de SharePoint, lo que incluye herramientas para partes web, listas, flujos de trabajo, eventos y mucho más.

2.6.3.6 Cree aplicaciones de RIA y WPF sin complicaciones

La nueva característica de conexión de datos mediante su arrastre y colocación disponible en diseñadores de Windows Presentation Foundation (WPF) y Silverlight hacen que la creación de aplicaciones enriquecidas de Internet (RIA) y Windows resulte fácil y rápida, tanto para diseñadores como para desarrolladores.

2.6.3.7 Simplifique la implementación de aplicaciones web

Traslade sus aplicaciones web a un entorno de producción con un único clic. Visual Studio 2010 Professional transfiere el código, la configuración de Internet Information Server (IIS) y el esquema de bases de datos al servidor de destino.

2.6.3.8 REQUISITOS DE SOFTWARE

Visual Studio 2010 puede instalarse en los sistemas operativos siguientes:

- Todas las ediciones de Windows XP (x86) con Service Pack 3, excepto Starter Edition
- Todas las ediciones de Windows XP (x64) con Service Pack 2, excepto Starter Edition
- Todas las ediciones de Windows Vista (x86 y x64) con Service Pack 1, excepto Starter Edition
- Windows 7 (x86 y x64)
- Windows Server 2003 (x86 y x64) con Service Pack 2
- Windows Server 2003 R2 (x86 y x64)
- Windows Server 2008 (x86 y x64) con Service Pack 2
- Windows Server 2008 R2 (x64)

Arquitecturas compatibles:

32 bits (x86)

64 bits (x64)

2.6.3.9 REQUISITOS DE HARDWARE

Equipo con un procesador de al menos 1,6 GHz

1024 MB de RAM

Espacio disponible en disco duro de 3 GB

Unidad de disco duro de 5.400 RPM

Tarjeta de vídeo compatible con DirectX 9 con una resolución de 1280 x 1024 o superior

Unidad de DVD-ROM[5]

[5] (Danysoft, 2010)

2.7 Servidor de aplicaciones

a. Antes de la Web, el término se refiere a un equipo en un entorno cliente / servidor que realiza la lógica de negocio (el procesamiento de datos). En un entorno cliente / servidor de dos niveles, que es más común, la máquina del usuario realiza la lógica de negocio, así como la interfaz de usuario y el servidor proporciona el procesamiento de base de datos. En un entorno de tres niveles, un equipo independiente (servidor de aplicaciones) realiza la lógica de negocio, aunque alguna parte todavía puede ser manejada por el equipo del usuario. Después de la Web explotó en la década de 1990, los servidores de aplicaciones Web se convirtió en la base (véase la siguiente definición).

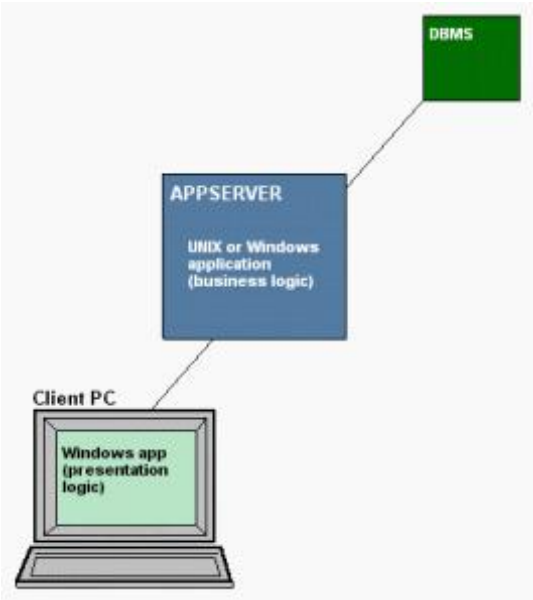


Ilustración 1 Servidor de Aplicaciones
Fuente: Computer Desktop Encyplopedia,
Elaborado: Computer Desktop Encyplopedia

2.7.1 Three-Tier Client / Server

Un servidor de aplicaciones en un entorno cliente / servidor de tres niveles proporciona un procesamiento de etapa intermedia entre el equipo del usuario y el sistema de gestión de base de datos (DBMS).

Desde el advenimiento de la Web, el término a menudo se refiere al software en un entorno de intranet / Internet que aloja una variedad de sistemas de lenguaje utilizado para consultar bases de datos y / o llevar a cabo el procesamiento de negocios en

general. Estos scripts y servicios, tales como Active Server Pages (ASP), Java Server Pages (JSP) y los servlets de Java, por lo general acceden a una base de datos para recuperar hasta al día los datos que se presentan a los usuarios a través de sus navegadores o aplicaciones cliente.

El servidor de aplicaciones puede residir en el mismo equipo que el servidor Web (servidor HTTP) o estar en un equipo independiente. En los grandes sitios, múltiples servidores de aplicaciones y múltiples servidores web (servidores HTTP) se utilizan. Los ejemplos de servidores de aplicaciones Web incluyen BEA Weblogic Server y de IBM WebSphere Application Server.

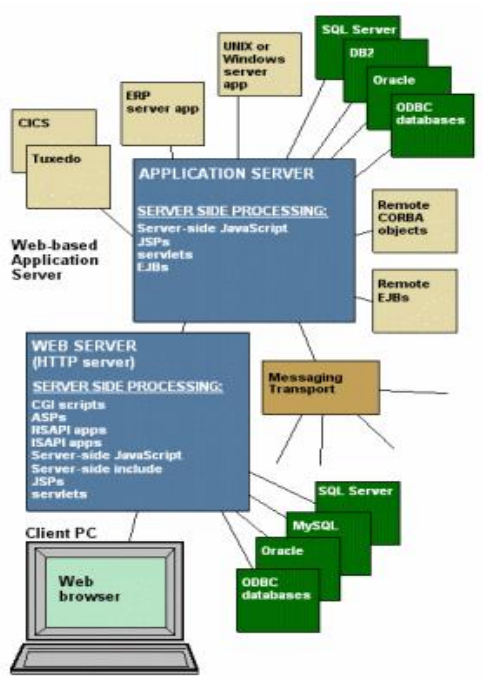


Ilustración 2 WebSphere Application Server

Fuente: Computer Desktop Encyplopedia,
Elaborado: Computer Desktop Encyplopedia

2.7.2 Servidores de Aplicaciones y Servidores Web

Existe una superposición entre un servidor de aplicaciones y un servidor Web, ya que ambos pueden realizar tareas similares. El servidor Web (servidor HTTP) puede invocar una variedad de secuencias de comandos y servicios para consultar bases de datos y llevar a cabo procesos de negocio, y los servidores de aplicaciones a menudo vienen con su propio servidor HTTP que suministra páginas Web en el navegador.

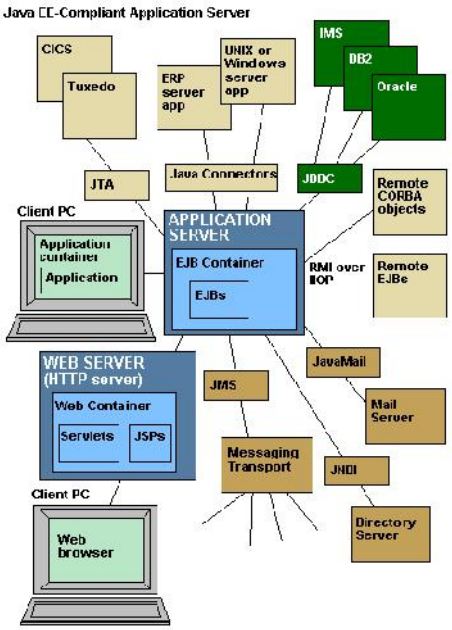


Ilustración 3 Servidores de Aplicaciones y Servidores Web

Fuente: Computer Desktop Encyplopedia,
Elaborado: Computer Desktop Encyplopedia

2.7.3 Java EE Application Server

Los servidores de aplicaciones se han convertido en el middleware para la empresa, ya que cada vez ofrecen más ganchos en las aplicaciones heredadas. Se trata de un servidor de aplicaciones Java EE con RoHS correr sólo Java y el uso de Enterprise JavaBeans (EJB) para la lógica de negocio.[6]

[6] (McGraw-Hill, 2003)

2.8 Arquitectura de aplicaciones

2.8.1 Definición:

La manera en la que es diseñada tanto física como lógicamente una aplicación.

2.8.2 Tipos de Arquitecturas

2.8.2.1 Arquitectura Centralizada

Se basa en la existencia de una máquina que almacena los datos y aplicaciones a las cuales se conectan las terminales.



Ilustración 4 Arquitectura Centralizada

Fuente: (Chio, 2012),
Elaborado: Chiona Chio

2.8.2.2 Arquitectura Distribuida

- **Peer to Peer:** El sistema es un conjunto de objetos que interaccionan, y cuya localización no interesa. No hay distinción entre un proveedor de servicios y el usuario de los mismos.

- **Cliente – Servidor:** Existen dos aplicaciones ejecutándose de forma independiente,

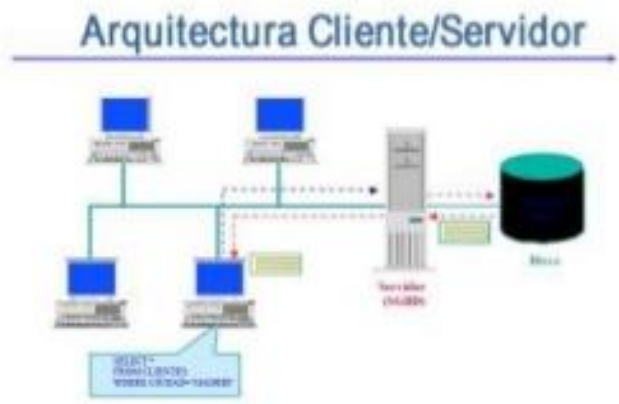


Ilustración 5 Arquitectura Cliente/Servidor

una actúa como servidora y la otra como cliente.[7]

Fuente: (Chio, 2012),
Elaborado: Chiona Chio

2.8.2.3 Cliente Servidor

2.8.2.3.1 Definición

El modelo cliente-servidor (client-server), describe el proceso de interacción entre la computadora local (el cliente) y la remota (el servidor).

2.8.2.3.2 Funcionamiento del modelo cliente/servidor

El cliente le hace peticiones (requests, solicitudes, requerimientos) al servidor, el cual procesa dicho requerimiento y retorna los resultados al cliente apropiado.

Por lo general, los clientes y los servidores se comunican entre sí a través de una red, pero también pueden residir ambos en un mismo sistema (el mismo hardware). [8]

[7] (Chio, 2012)
[8] (Freddy, 2009)

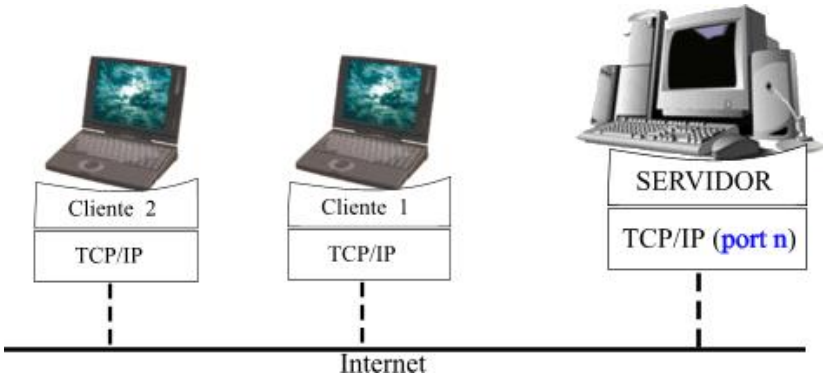


Ilustración 6 Modelo Cliente Servidor

Fuente: (Freddy, 2009)
Elaborado: John Freddy

Siguen este modelo de cliente/servidor servicios como el intercambio de emails, el acceso a webs, el acceso a bases de datos, y muchos otros protocolos de internet se basan en esta idea (HTTP, SMTP, Telnet, DNS), etc.

2.8.2.3.3 Características del modelo cliente/servidor

La máquina que sirve como servidor (host) es la que tiene en ejecución programas de servidor que contestan los requerimientos de los clientes. Por lo general los clientes inician la sesión de comunicación, mientras que los servidores esperan la llegada de solicitudes.

La interacción entre cliente y servidor es generalmente representada empleando diagramas de secuencia (estandarizados en UML).

En contraste, la arquitectura P2P (peer-to-peer), cada host o instancia de programa puede ser simultáneamente tanto cliente como servidor, y cada uno tener sus propias responsabilidades y estado.

2.8.2.4 Ventajas y desventajas del modelo cliente/servidor 2 Capas

a. Ventajas

   i. Centralización del control de los recursos, datos y accesos.

ii. Facilidad de mantenimiento y actualización del lado del servidor: Esto es porque el lado del servidor se puede mantener o actualizar fácilmente. Por ejemplo, una actualización se aplica a un único servidor, pero los beneficios los obtienen múltiples clientes generalmente sin necesidad de que éstos actualicen nada.

iii. Toda la información es almacenada en el lado del servidor, que suele tener mayor seguridad que los clientes.

iv. Hay muchas herramientas cliente-servidor probado, seguro y amigable para usar.

b. Desventajas

i. Si el número de clientes simultáneos es elevado, el servidor puede saturarse. Esto sucede con menor frecuencia en las redes P2P.

ii. Frente a fallas del lado del servidor, el servicio queda paralizado para los clientes. Algo que no sucede en una red P2P.

2.9 Por capas

Arquitectura de tres capas

- **Capa de Datos**

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

  - Base de datos
  - Tablas
  - Procedimientos almacenados
  - Componentes de datos

- **Capa de negocio**

  Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se

denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

- Reglas del negocio
- Validaciones
- Cálculos
- Flujos y procesos

- **Capa de presentación**

  Es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.

  - Formularios
  - Informes
  - Respuestas al usuario[9]

[9] (Chio, 2012)

3. CAPÍTULO 3: CONSTRUCCIÓN DEL PRODUCTO

3.1 ESTUDIO DE FACTIBILIDAD

3.1.1 FACTIBILIDAD OPERACIONAL

La Factibilidad Operativa permite predecir, si se pondrá en marcha el sistema propuesto, aprovechando los beneficios que ofrece, a todos los usuarios involucrados con el mismo, ya sean los que interactuarán en forma directa con este, como también aquellos que recibirán información producida por el sistema. Por otra parte, el correcto funcionamiento del sistema en cuestión, siempre estará supeditado a la capacidad de las personas encargadas de dicha tarea.

La necesidad y deseo de un cambio con el método de registro actual, expresada por los usuarios, llevó a la aceptación de un nuevo sistema, que de una manera más sencilla y amigable, cubra todos sus requerimientos, expectativas y proporciona la información en forma oportuna y confiable. Basándose en las entrevistas y conversaciones sostenidas con las personas encargadas se demostró que estos no representan ninguna oposición al cambio, por lo que el sistema es factible operacionalmente.

En el proceso de adiestramiento se detallaron los aspectos de actualización de conocimientos y nuevas formas en el procesamiento de transacciones que representan el manejo del nuevo sistema.

Con la finalidad de garantizar el buen funcionamiento del sistema y que este impactará en forma positiva a los usuarios, el mismo será desarrollado con una interfaz amigable al usuario, lo que se traduce en una herramienta de fácil manejo y comprensión, tanto las pantallas como los reportes serán familiares a los operadores, contando con la opinión de los mismos para cualquier modificación del sistema

3.1.2 FACTIBILIDAD TÉCNICA

La Factibilidad Técnica consistió en realizar una evaluación de la tecnología existente en la fundación, este estudio estará destinado a recolectar información sobre los componentes técnicos que posee la organización y la posibilidad de hacer uso de los mismos en el desarrollo

e implementación del sistema propuesto y de ser necesario, los requerimientos tecnológicos que deben ser adquiridos para el desarrollo y puesta en marcha del sistema en cuestión.

De acuerdo a la tecnología necesaria para la implantación del sistema dentro de la información, se evaluará bajo dos enfoques:

HARDWARE	**Computador 1**	**Computador 2**	**Computador3**
Nombre	**Sv1**	**Xtratech**	**DELL ONE**
Motherboard:	Intel D845GVFN	Intel A123FGS	Intel DQ1231ER
CPU	Intel Pentium Celeron 2.4Ghz	Intel Pentium Celeron 2.4Ghz	Core-2.20Ghz
Memoria	512MB	2 GB	4gb
Disco Duro	40GB IDE	160 GB	500 GB
OS	Win XP SP3	Win XP SP3	Win 7
CDRW	52X	52X	52X
Puertos Frontales	Dos USB frontales. Mic y Audífono frontal.	Dos USB frontales. Mic y Audífono frontal.	
Puertos Posteriores	2 USB, Mic, Entrada Audio, Salida Audio, Paralelo, Serial, dos PS/2, Monitor, Ethernet.	4 USB, Mic, Entrada Audio, Salida Audio, Monitor, Ethernet.	6 USB, Mic, salida audio, Ethernet
Tarjeta Gráfica	256 MB	512MB	1GB
SOFTWARE			
Ofimática	Office 2003 SP3	Office 2007 SP1	Office 2010
Antivirus	Avast Free Antivirus 8.0.1489	Avast Free Antivirus 8.0.1489	Avast Free Antivirus 8.0.1489
NET.Framework	3.0	NO	3.5 Y 4
Microsoft Visual C++	NO	2008	2008, 2010
Visual Basic Runtime 6.0	NO	SI	SI
Microsoft SyncToy	NO	NO	SI
DropBox	NO	SI	SI

Tabla 2 Factibilidad Técnica

Fuente: Fundación FUDIS
Elaborado: Carlos Guamán y Jorge Recalde

3.2 REQUERIMIENTOS DEL SISTEMA

Dentro del levantamiento de requerimientos podemos utilizar dos conceptos muy importantes los cuales deben ser bien descritos en un lenguaje natural para que puedan ser entendidos, estos son:

- Escenarios: Estos ayudan a describir un ejemplo del uso del sistema en términos de una serie de interacciones entre el usuario y el sistema.
- Casos de uso: Es una abstracción en la que se describe una clase de escenarios.

Para cubrir el levantamiento de requerimientos es necesario cumplir algunas actividades como por ejemplo:

Identificación de Actores

Dentro del sistema hemos identificado dos actores principales que interactúan directamente.

Usuarios: Las personas involucradas dentro de los procesos de registro de pacientes y donación de artículos.

Administrador: La persona encargada de utilizar la administración de procesos involucrados con los usuarios.

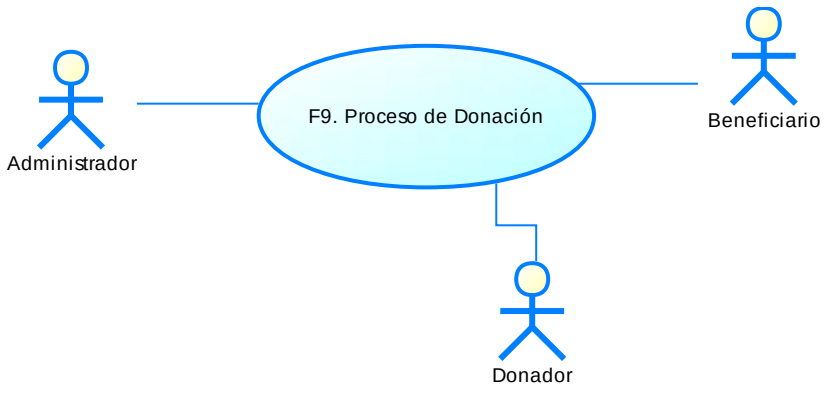


Ilustración 7 Identificación de actores

Fuente: Requerimientos Funcionales y No Funcionales (Pablo Quiroga)
Elaborado: Carlos Guamán – Jorge Recalde

Identificación de escenarios

Nombre del escenario	DONACIÓN
Instancias de los usuarios participantes	ADMINISTRADOR:DONADOR
Flujo de eventos	1. Donador ingresa al sistema indicando sus datos en este caso la cédula o documento similar. 2. El sistema indica un menú dando cada una de las posibilidades del sistema. 3. Donador indica que quiere donar dinero para la fundación. 4. El sistema solicita ingresar la cantidad que desea donar, y si existe empresa que representa. 5. Donador da el valor a ser donado e indica que no tiene empresa a representar. 6. El sistema registra la información correspondiente a la donación efectuada.

Tabla 3 Identificación de escenarios

Fuente: Requerimientos Funcionales y No Funcionales (Pablo Quiroga)
Elaborado: Carlos Guamán – Jorge Recalde

Identificación de casos de uso

Para esta identificación hemos podido identificar mediante un diagrama de casos de uso general donde podemos observar las administraciones y procesos involucrados en el sistema.

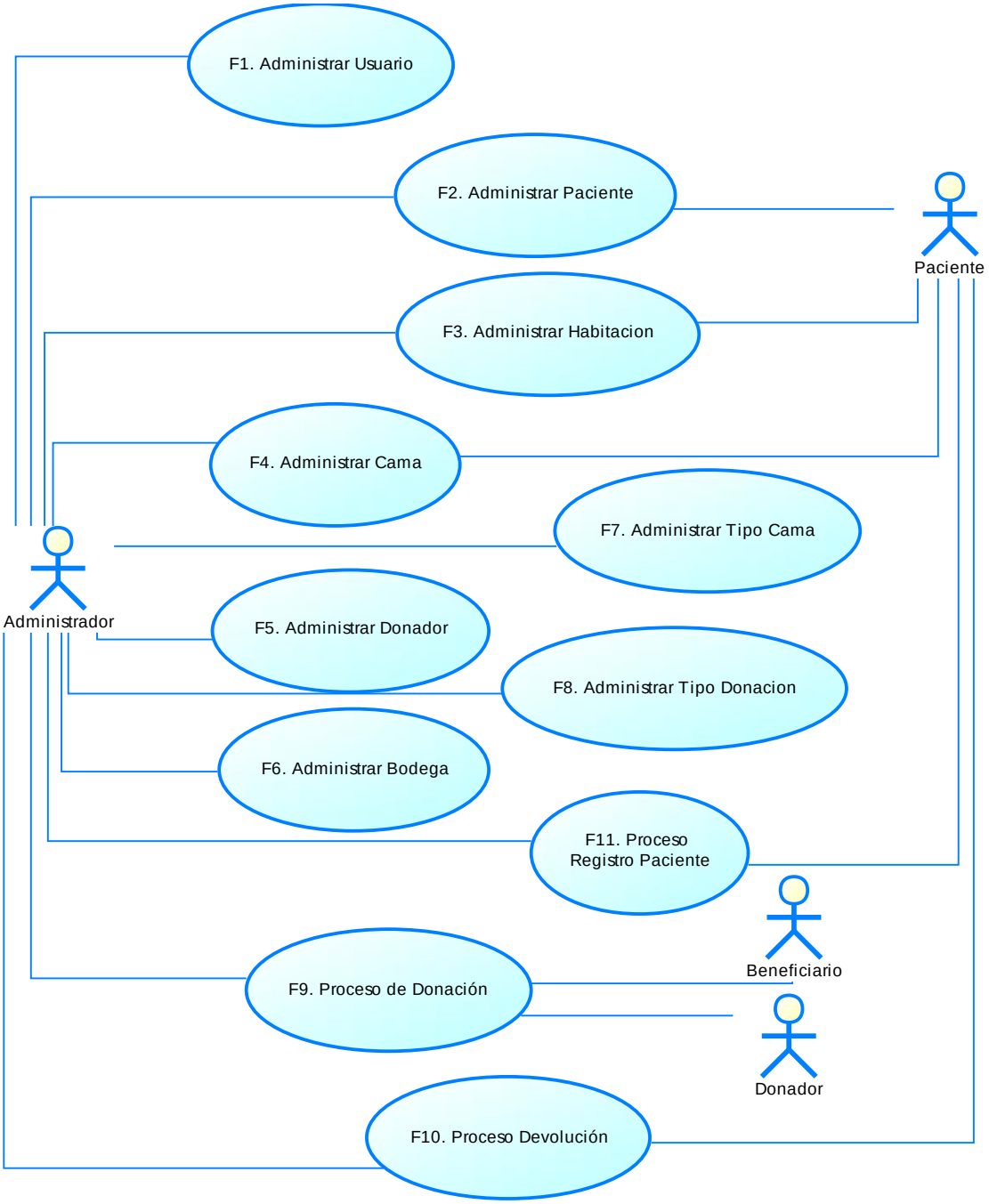


Ilustración 8 Identificación de Casos de Uso

Fuente: Requerimientos Funcionales y No Funcionales (Pablo Quiroga)

Elaborado: Carlos Guamán – Jorge Recalde

3.2.1 REQUERIMIENTOS FUNCIONALES

- El Sistema administrará un registro de habitaciones y camas disponibles dentro del albergue en base a un esquema del mismo e información almacenada en la base de datos.
- El Sistema permitirá registrar a todos los pacientes mientras exista camas disponibles.

- El Sistema permitirá llevar un completo registro de utilización del albergue en el cual constarán datos como el nombre y apellido del paciente, fecha de ingreso, fecha de salida, su valor a pagar por estancia y la cama asignada.
- El Sistema imprimirá un reporte de actividades diarias con la información de los pacientes ingresados, fecha de entrada, fecha de salida, cama asignada y valor a pagar, el mismo que servirá como respaldo para la encargada de la contabilidad a verificar los valores ingresados.
- El Sistema permitirá el desembolso de dinero si fuere el caso que un paciente se quiera ir antes de la fecha de salida acordada.
- El sistema permitirá llevar un completo registro de donaciones para el albergue en el cual constarán los datos del Donador, el detalle de la donación, la cantidad de los artículos, el lugar de almacenamiento y la fecha en la cual se realiza la donación.
- El sistema permitirá llevar un completo registro de donaciones que realiza la Fundación hacia otras entidades similares, en la cual constarán los datos del detalle de la donación, la cantidad de los artículos, el beneficiario, el lugar de almacenamiento del cual se envía la donación y la fecha en la cual se realiza la donación.
- El sistema imprimirá un reporte de las donaciones por mes para que sea respaldo de los encargados de los lugares donde se almacenen estas donaciones.

3.2.2 REQUERIMIENTOS NO FUNCIONALES

Tecnología

Un acceso a internet es importante porque tenemos un respaldo de la base de datos en una herramienta como Dropbox que es un servicio online de almacenamiento en la nube.

Conexión a internet: 256Kbps:

Licenciamiento

El sistema está elaborado bajo las siguientes especificaciones.

Lenguaje De Programación: Visual Basic Express Edition.

Base De Datos: SQL SERVER 2008 Express Edition.

Con este detalle podemos involucrar un licenciamiento libre del tipo software de dominio público sometido a uso, copia, modificación o redistribución con o sin fines de lucro. Para este caso utilizaremos la licencia **GPL versión 3**, donde se puede tener libertad de usar el programa, libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades, libertad de distribuir copias, libertad de mejorar el programa y hacer públicas las mejoras.

Requerimientos:

Hardware	Sistemas Operativos	Software
Computador • **Memoria RAM:** 256 MB • **Espacio en Disco:** 1 GB • **Procesador:** Núcleo Simple	Windows XP Professional Windows Vista Professional Windows 7 Ultimate o Inferior Windows 8 Home Edition Windows Server 2000/2008	• Microsoft .NET Framework Redistributable 3.5 • Visual Basic Runtime 6.0 • Microsoft Visual C++ Redistributable 2005, 2008, 2010 • Microsoft SyncToy

Tabla 4 Requerimientos no funcionales

Fuente: Requerimientos Funcionales y No Funcionales (Pablo Quiroga)
Elaborado: Carlos Guamán – Jorge Recalde

3.3 LEVANTAMIENTO DE LA INFORMACIÓN

3.3.1 ENTREVISTA

Durante el levantamiento de la información se realizó varias entrevistas con los directivos y las personas encargadas de la Fundación las cuales nos ayudaron con la información de cómo se manejaba las actividades dentro del albergue.

Primero se habló directamente con la directora de la Fundación la Sra. Amparo de Páez la cual nos manifestó que tiene varios pacientes que se registran durante un tiempo determinado y que la fundación recibe un valor por cada noche de hospedaje, además nos indicó las normas que rigen en el albergue y como se opera con cada una de las personas encargadas.

Segundo hablamos con la Gerente de la fundación la cual nos manifestó a más detalle las actividades que estaban realizando hasta esos momentos por lo que nos manifestó que la fundación recibe donaciones de diferentes personas o empresas y que también realizaban donaciones como Fundación hacia otras entidades.

Tercero hablamos con la Secretaria de la Fundación la cual nos detalló cada una de las actividades que ella realizaba para el ingreso diario de un paciente y como se realiza la donación y que para todo esto se lo realizaba a mano o con la utilización de herramientas de ofimática, además nos manifestó que ella tenía problemas cuando los usuarios del albergue tenían que realizar una devolución, y que ese problema descuadraba los valores de caja cuando se los cuadraba.

3.3.2 OBSERVACIÓN

Durante el tiempo que estuvimos en las diferentes entrevistas también nos proporcionaron las formas en las cuales se realizaba dichos registros y pudimos observar que tenían varias maneras y lugares donde guardaban los registros que había un diseño único pero varios archivos y documentos con mucha información desordenada.

Era necesario hacer un sistema que proporcione una base de datos donde la información este en un solo lugar y así l facilite las actividades a los encargados de la fundación. También en la estancia en el albergue pudimos observar de manera directa varios registros de pacientes, que

documentos eran necesarios para ese registro y que elementos adicionales eran obligatorios para que tenga un funcionamiento correcto cada actividad.

3.4 ANÁLISIS DEL SISTEMA

En el análisis del sistema que implementaremos debemos presentar de una manera que el usuario final entienda y domine la información de los problemas que vamos a detallar, definiremos las funciones que realizará el software y que utilidades podrían darle a la herramienta dentro del albergue para que así refleje un comportamiento con el software a consecuencias y acontecimientos externos de parte de quien lo utiliza. Además dividiremos de forma jerárquica los modelos que representa la información obtenida en las funciones, procesos, actores y comportamientos

3.4.1 FASE DE INSPECCIÓN

Dentro de la lista de defectos en lo que se concierne al registro de datos tanto de los pacientes como de los artículos que poseen, se pudo observar y discutir con las autoridades de la Fundación todo acerca de tres puntos muy importantes que se desarrollaran de la siguiente manera:

- Registro Diario de Pacientes

  El ingreso Diario de Paciente hace referencia a la actividad de registrar el ingreso de cada paciente asignando una cama y una habitación por la cual se deberá cancelar un valor por día o un total al finalizar la estadía del paciente en el Albergue.

- Registro de Donaciones

  El registro de donaciones tiene que ver tanto con los artículos o dinero que recibe la Fundación y el inventario que se debería llevar para mantener un control de los bienes materiales. Además los registros de donaciones deben tener un remitente o más claro un donador que realiza la donación, para esto debemos tener registrado la fecha que realiza dicha donación para que así se puedan enviar los agradecimientos a los donadores. También la Fundación realiza donaciones a otras

entidades que también deben estar registradas y actualizadas en el inventario.

- Registro de Devoluciones

  El problema que persiste en las devoluciones es más bien de aspecto técnico porque existen pacientes que realizan un registro y pagan por varios días por su estadía, pero existen veces que ellos deciden irse antes del tiempo acordado y se debe calcular el valor a devolver.

3.4.2 FASE DE ESTUDIO

En esta fase podemos explicar los defectos encontrados dentro de la Fundación y aclaramos algunos puntos relevantes de cómo estos defectos perjudican a FUDIS en el funcionamiento diario de sus actividades ya que si se realiza un proyecto de software ayudará en la optimización de los recursos y un seguro mantenimiento de información.

Proceso	Registro Diario de Pacientes	Donaciones	Devoluciones
	Ingreso de pacientes por nombre y apellido.	Ingreso de artículos no se registra.	Cuando un paciente se retira antes del tiempo asignado realiza una devolución.
	No existe un registro único de todos los pacientes.	Ingreso de dinero no se registra.	Las devoluciones se realizan manualmente o con ayuda de herramientas de ofimática.
	La asignación de habitaciones depende de la observación de los encargados.	Los donadores no están bien identificados.	No existe el registro de dichas devoluciones.
	La asignación de las camas depende de la observación de los encargados.	No se registra que donaciones realiza la fundación.	Existe confusión cuando hay alguna devolución.
	No existe un control de los acompañantes de cada uno de los pacientes.	Los artículos no están ubicados dentro de un inventario.	

	Calculo de tarifas individuales y totales se realizan a mano o con ayuda de herramientas de ofimática.	Las transacciones con dinero no tienen un fondo común de registro.	
	Registro de ingreso con fechas exactas no existen.	No existe un inventario donde se pueda observar los artículos de la fundación.	

Tabla 5 Fase de Inspección

Fuente: Información recopilada en FUDIS
Elaborado: Carlos Guamán – Jorge Recalde

3.4.3 FASE DE DEFINICIÓN

Una vez realizada las fases de inspección y estudio debemos planificar y definir los objetivos a realizarse e identificar los participantes y los criterios de entrada a cada uno de sus procesos.

Para esto vamos a implementar de un sistema informático para la automatización de los procesos de donaciones, registro diario de Pacientes, devoluciones e inventario, utilizando herramientas para la mejor optimización de los recursos dentro de la Fundación.

Dentro del desarrollo o implementación de dicho sistema informático iremos haciendo mejoras de los procesos observados y actores identificados en las anteriores fases.

3.5 DISEÑO DEL SISTEMA

3.5.1 DISEÑO DE LOS DATOS

Para el diseño de los datos hemos optado por recopilar la información necesaria a través de los datos proporcionados en la Fundación, para lo cual cada una de las tablas es evidencia de documentos físicos que se utilizaba antes de la sistematización de los procesos y las administraciones.

Podemos observar a detalle el Diseño de los Datos (véase Ilustración 9) que cada una de las tablas tiene un identificador el cual será incrementado en cada ingreso de un nuevo registro.

Cada dato tienen la relación con el nombre de su tabla que los contiene, por ejemplo en la tabla donador podemos observar que tiene:

IDDONADOR, CIDONADOR, APELLIDODONADOR, TELEFONODONADOR, EMPRESAREPRESENTANTEDONADOR Y EMAILDONADOR, todos estos pertenecen a la tabla DONADOR.

Las relaciones entre tablas que existen en el Diseño de Datos son importantes para el desarrollo del aplicativo, por ejemplo en las tablas DONADOR Y DONACION hay una relación de un donador puede hacer muchas donaciones y solo puede existir un único donador que no se repite.

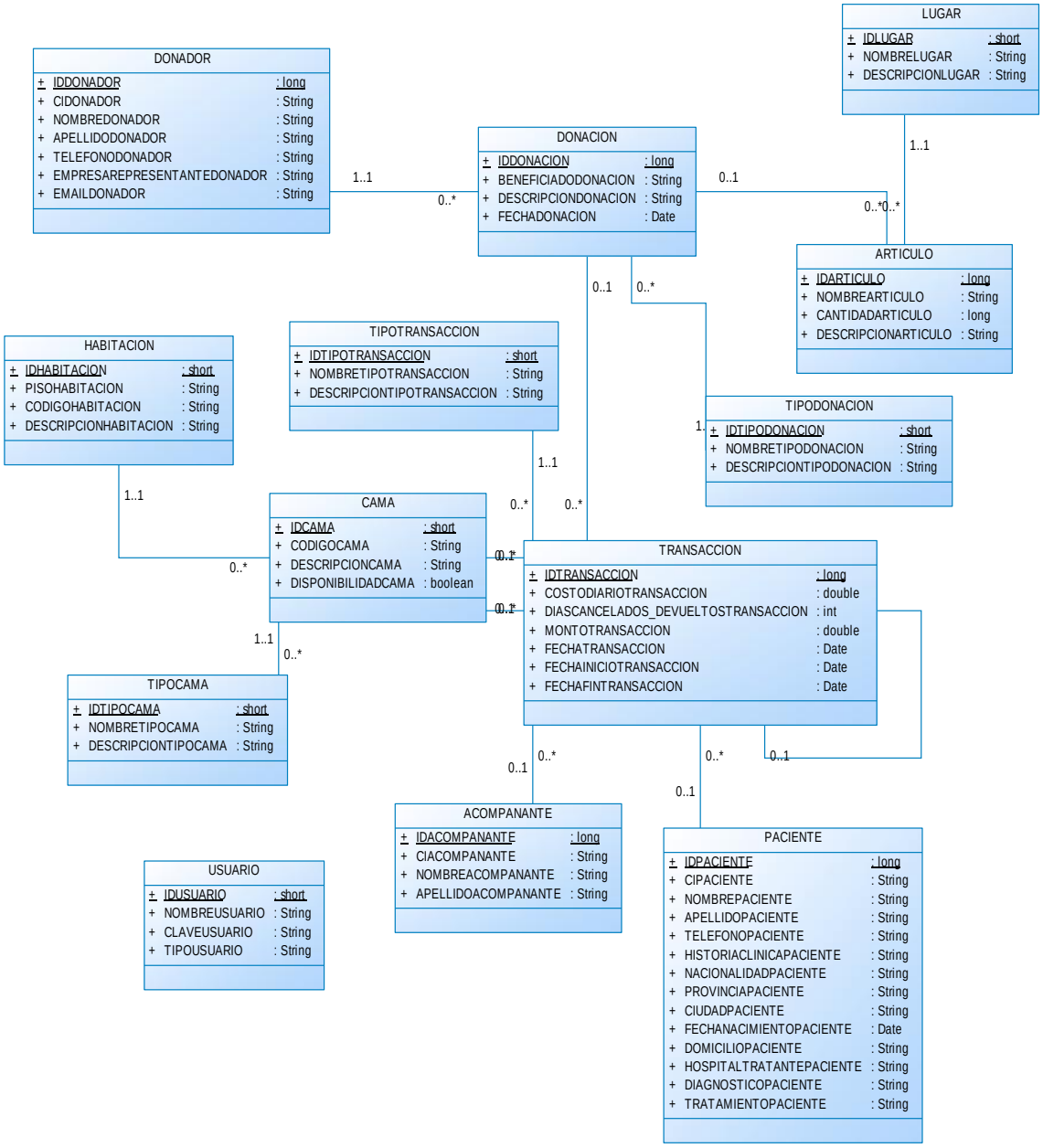


Ilustración 9 Diseño de Datos

Fuente: Bases de Datos Felipe Duarte Camacho
Elaborado: Carlos Guamán – Jorge Recalde

3.5.2 DISEÑO ARQUITECTÓNICO

Arquitectura centrada en los datos

Para esta arquitectura tenemos un almacén de datos, el cual es accedido de manera frecuente por los componentes que actualizan, añaden, borran o modifican dichos almacenes.

El software cliente accede a un repositorio central dependiendo el tipo la categoría de administrador, este repositorio lo llamamos Principal.

Arquitectura centrada en los flujos de datos.-

Utilizamos Combobox que nos ayudan a que filtren componentes independientes situados antes o después de cada conjunto de datos determinados utilizados en los Frameworks y con esto tenemos como resultado otros datos de salida en un formato específico.

Arquitectura llamada y respuesta (call and return).

El Programa principal/subprograma.- Descompone las funciones en una jerarquía de control donde el programa principal invoca a los otros programas subordinados, los cuales pueden a su vez invocar otros.

3.5.3 DISEÑO DE LA INTERFAZ

La creación de escenarios de usuarios nos ayudan a mantener un lenguaje simple entre el usuario y el sistema donde el cual el obtendrá el mayor beneficio de la interfaz desarrollada

En la generación de formatos de pantallas, utilizamos los formatos por defecto en Visual Basic Express Edition y modificadas de acuerdo a la utilidad ya sea una administración o un proceso, cada una de estas tiene diferente modalidad de ingreso, por ejemplo si es un proceso las administraciones le ayudan al proceso para ser completado con el ingreso de nuevos registros.

Con esto nos lleva a un desarrollo y modificación iterativa de prototipos para que sean utilizados dependiendo de la estructura que se desea.

Entre tanto el diseño de la interfaz entre los componentes utilizados ayudan al diseño de otras interfaces entre el software y otros productores y consumidores de información no externas dentro del programa, como resultado obtenemos un diseño de interfaz entre el administrador y la interacción de la máquina.

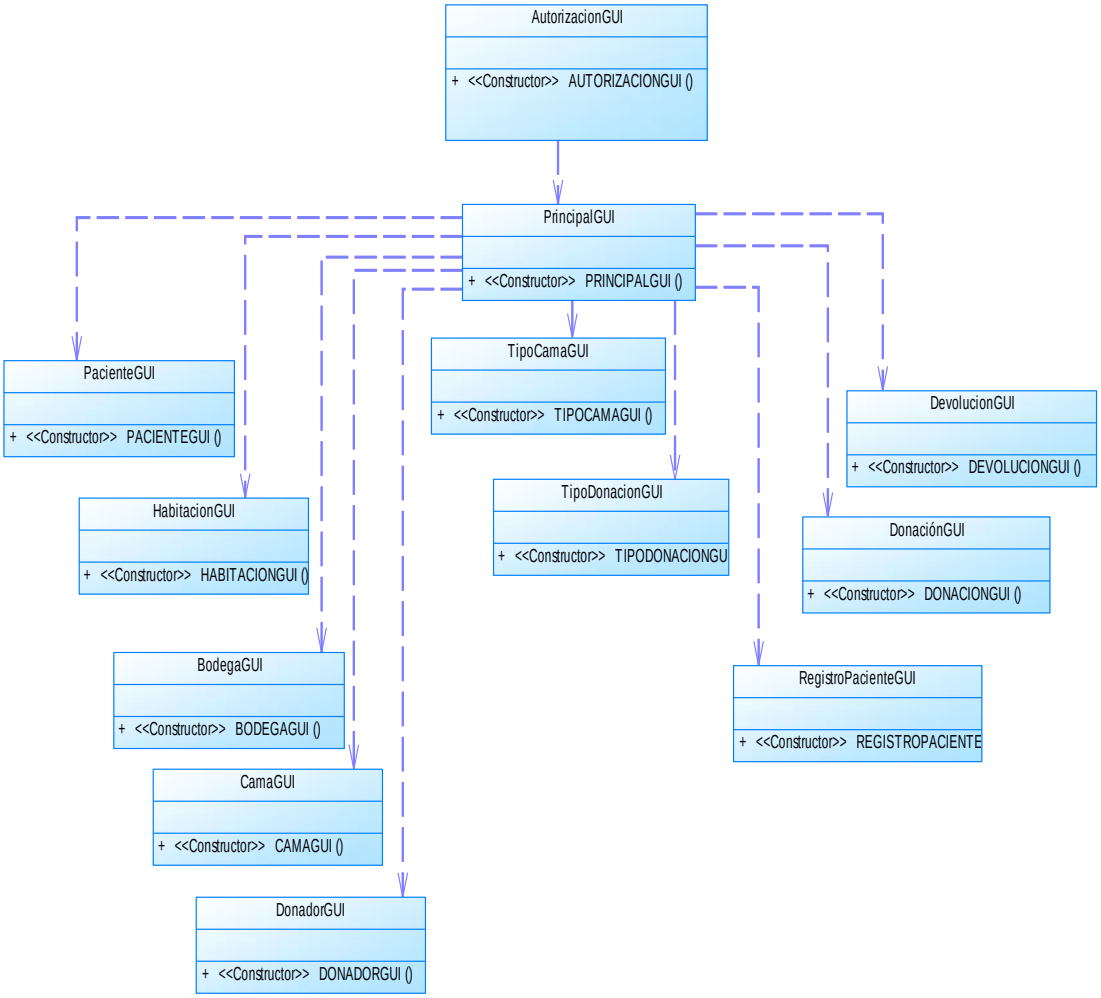


Ilustración 10 Diseño Arquitectónico

Fuente: https://docs.google.com/document/d/1C7cPJSR2mjAAuTYRfSR4N1Xb5BzS4czb0llPirSAYrU/edit
Elaborado: Carlos Guamán – Jorge Recalde

3.5.4 DISEÑO DE PROCEDIMIENTOS

Descripción:

La administración de un albergue involucra a varios actores, procesos y escenarios, los mismos que vendrán descritos dentro de las diferentes diagramaciones y explicaciones de actividades en este capítulo.

Actores:

- Usuarios
- Administrador

Procesos Involucrados:

Dentro de todo el manejo en el albergue se ha identificado los siguientes macro procesos y subprocesos

- Registro Diario de Pacientes
    - Registrar Paciente
    - Calcular Precio a Pagar
    - Asignar Código al Paciente
    - Registrar Habitación
    - Registrar Cama
    - Identificar Disponibilidad
    - Registrar Ingreso

- Donación
    - Validar Donador
    - Registrar Donador
    - Identificar Lugar de Donación
    - Registrar Beneficiario
    - Registrar en Inventario
    - Asignar Código a la Donación
    - Registrar Donación

- Devolución
    - Validar Paciente
    - Verificar Fecha de salida
    - Calcular monto a devolver
    - Registrar Salida

Cadena de Valor

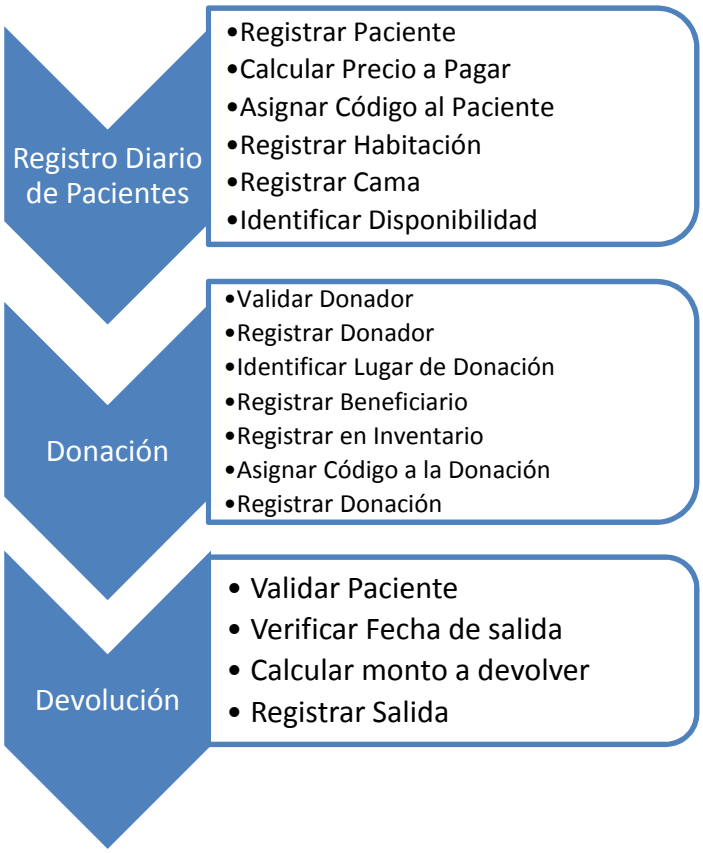


Ilustración 11 Cadena de Valor

Fuente: http://gestiondeempresas.org/diseno-procedimientos/
Elaborado: Carlos Guamán – Jorge Recalde

3.5.4.1 MAPA DE PROCESOS

Para el Mapa de Proceso de Donación hemos tomado en cuenta a los siguientes actores fundamentales como son el Donador y los encargados de FUDIS

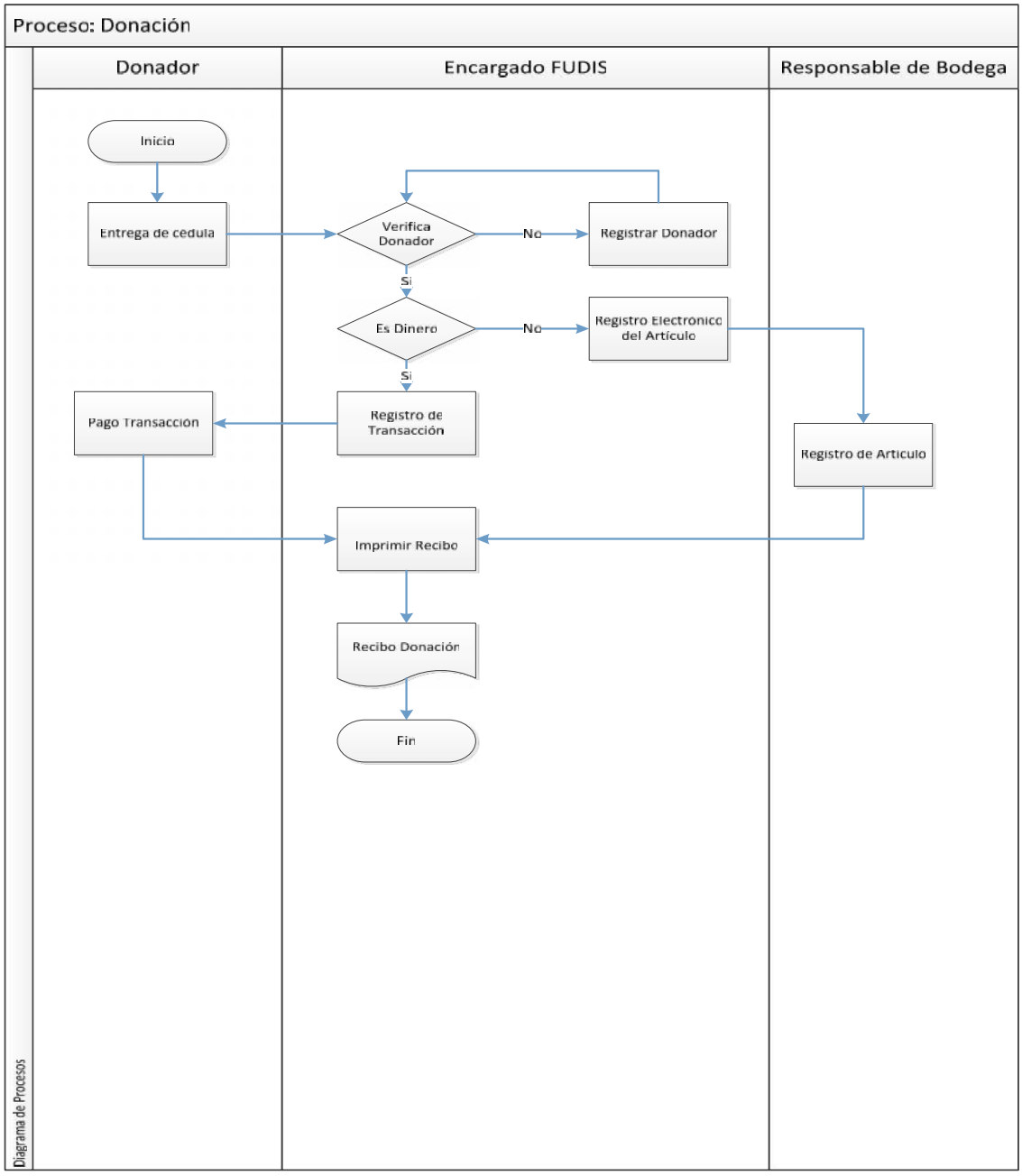


Ilustración 12 Mapa de Procesos Donación

Fuente:
Elaborado: Carlos Guamán – Jorge Recalde

Para el Mapa de Proceso de Ingreso Diario de pacientes hemos tomado en cuenta a los siguientes actores fundamentales como son el Paciente, Acompañante y Encargado de FUDIS

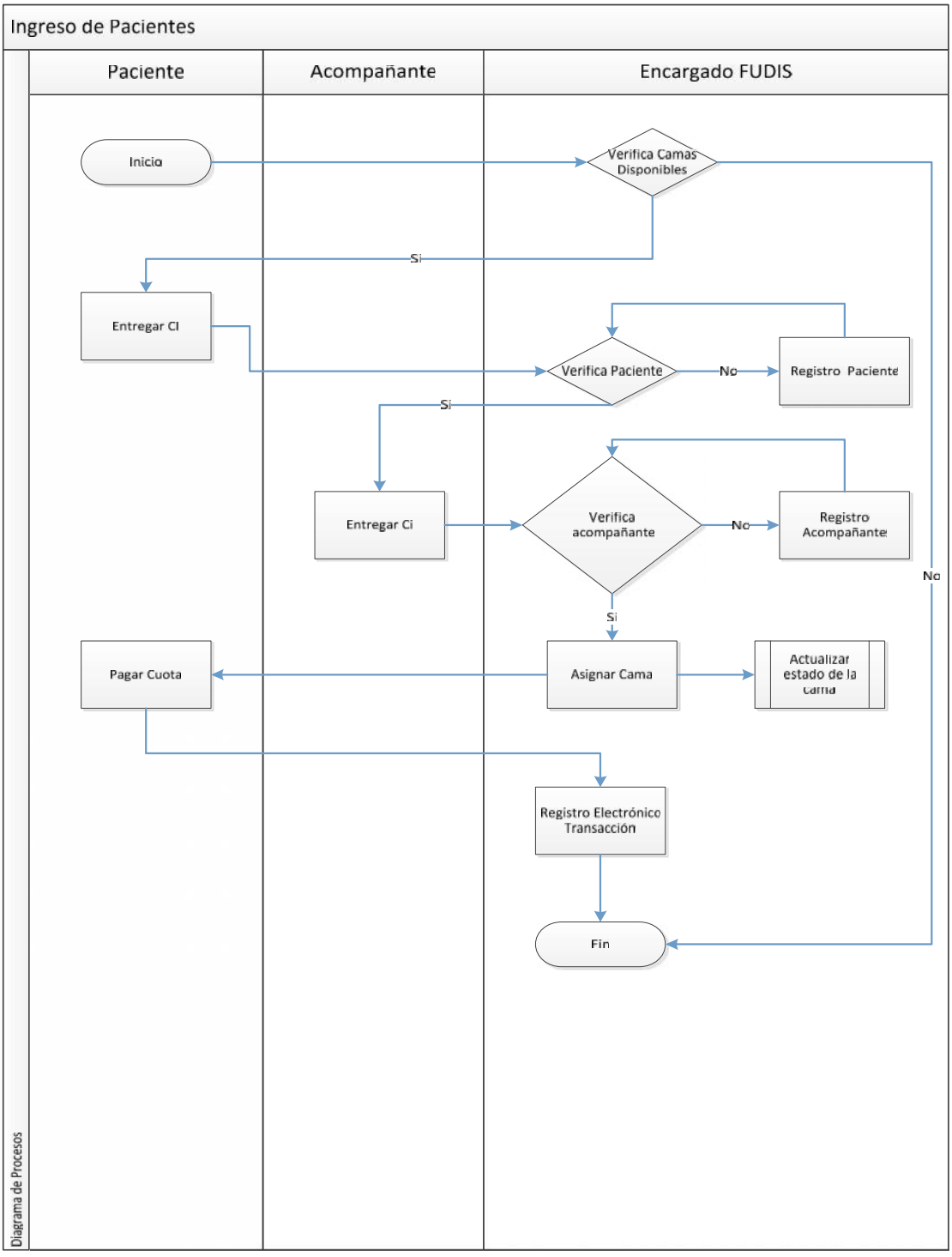


Ilustración 13 Mapa de Procesos Ingreso de Pacientes

Fuente:
Elaborado: Carlos Guamán – Jorge Recalde

Para el Mapa de Proceso de Devolución hemos tomado en cuenta a los siguientes actores fundamentales como son el Paciente y Encargado de FUDIS

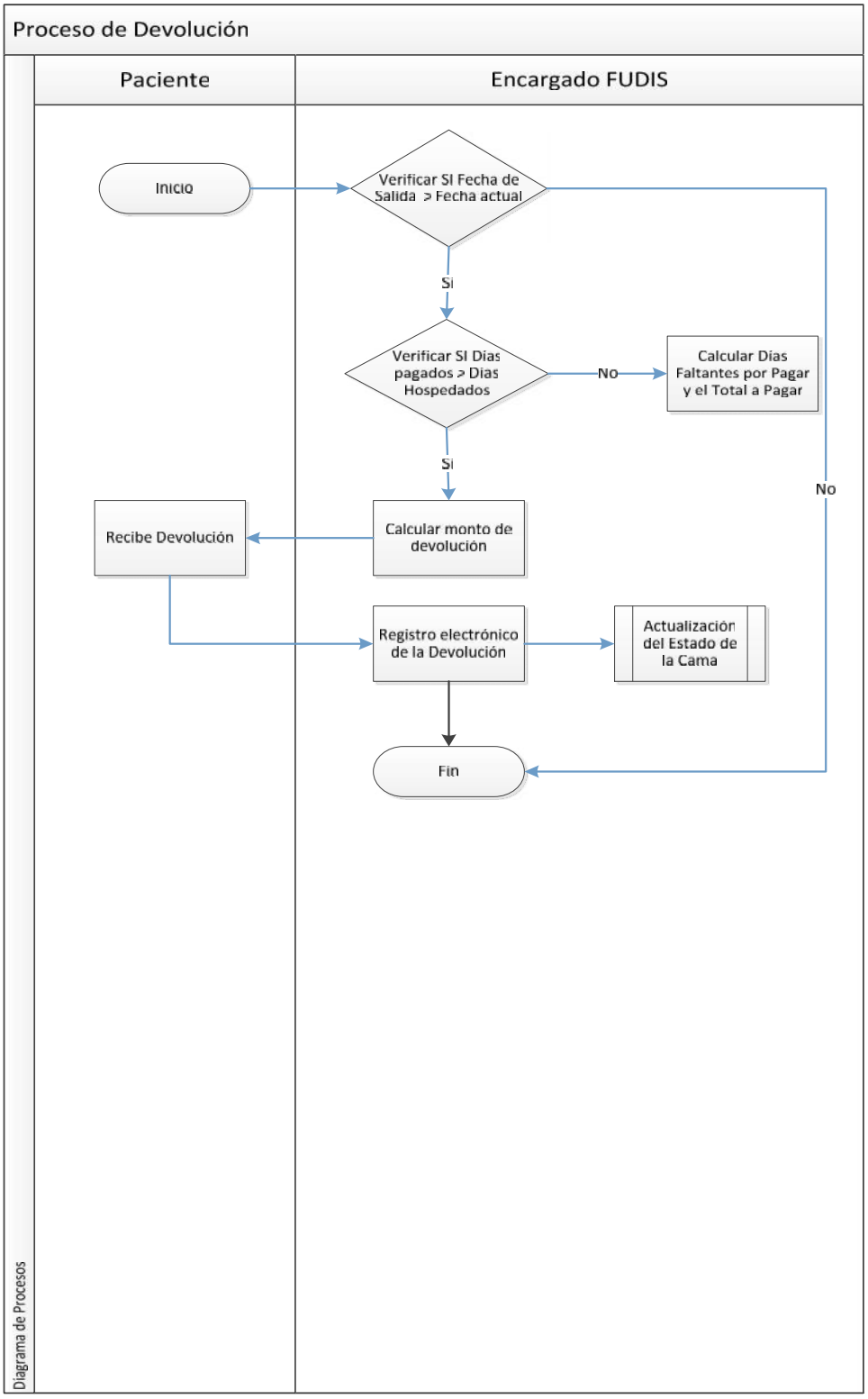


Ilustración 14 Mapa de Proceso Devolución

Fuente:
Elaborado: Carlos Guamán – Jorge Recalde

3.5.4.2 SIPOC

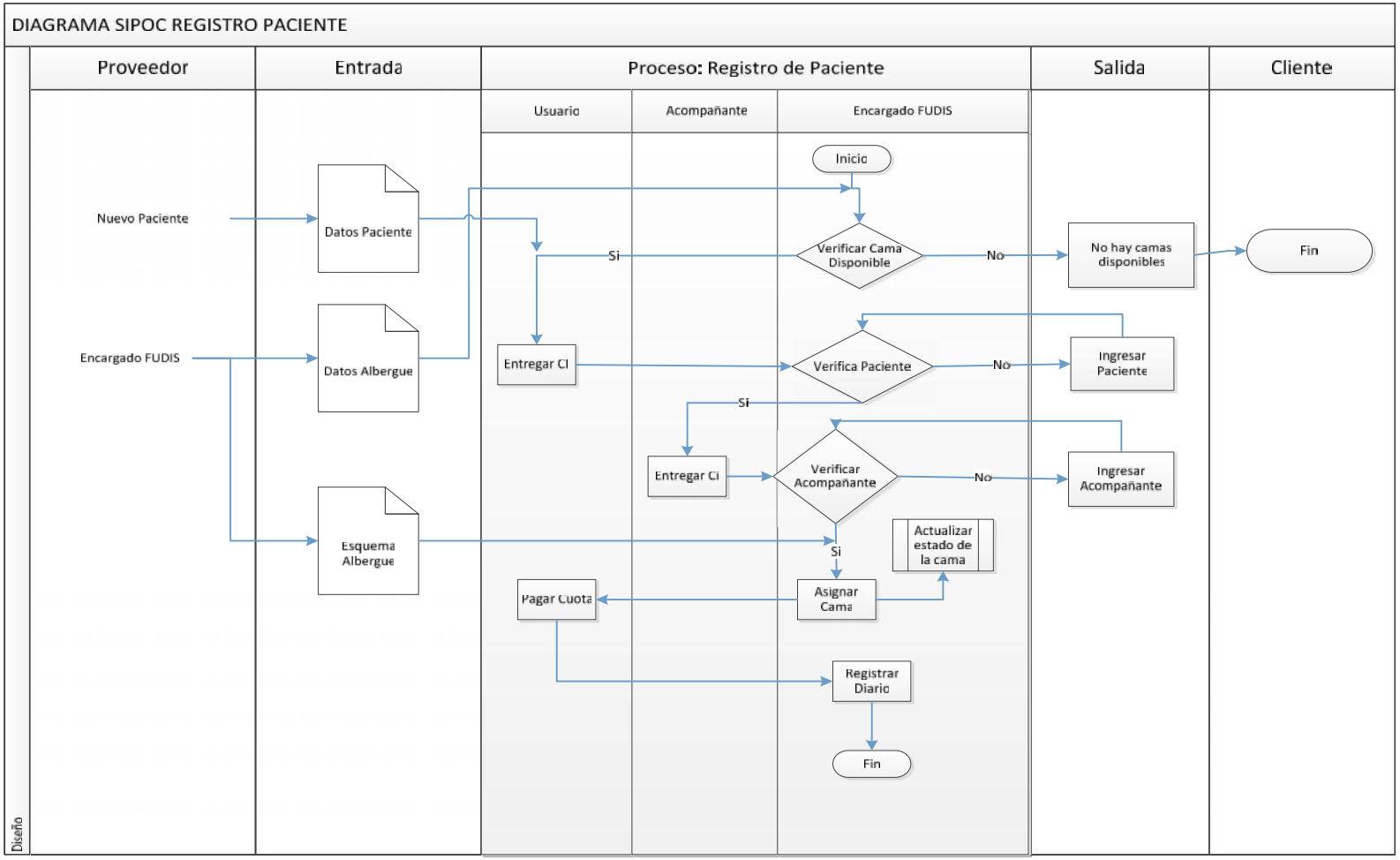


Ilustración 15 Diagrama SIPOC Ingreso Diario de Pacientes

Fuente:
Elaborado: Carlos Guamán – Jorge Recalde

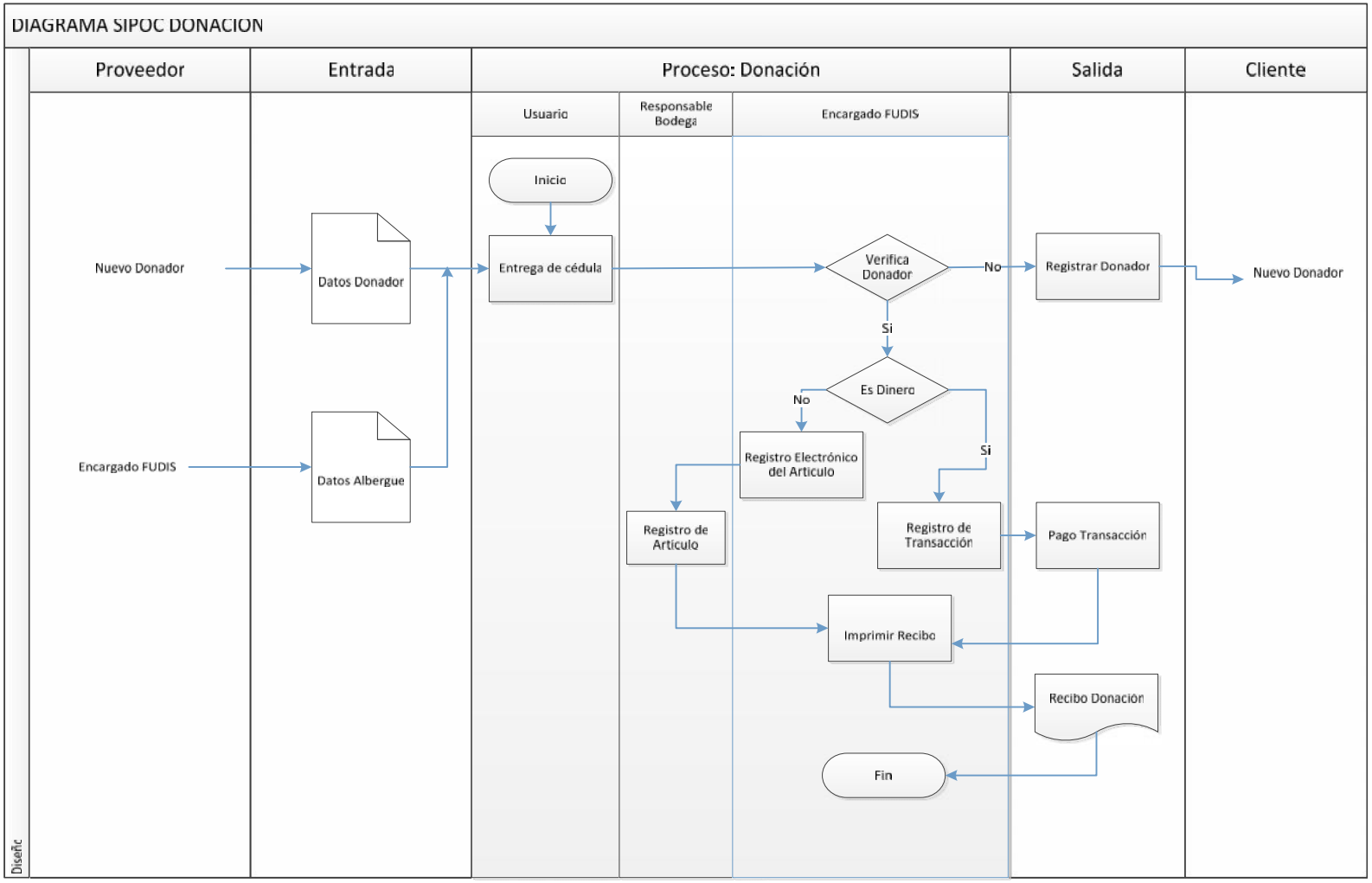


Ilustración 16 Diagrama SIPOC Donación

Fuente:
Elaborado: Carlos Guamán – Jorge Recalde

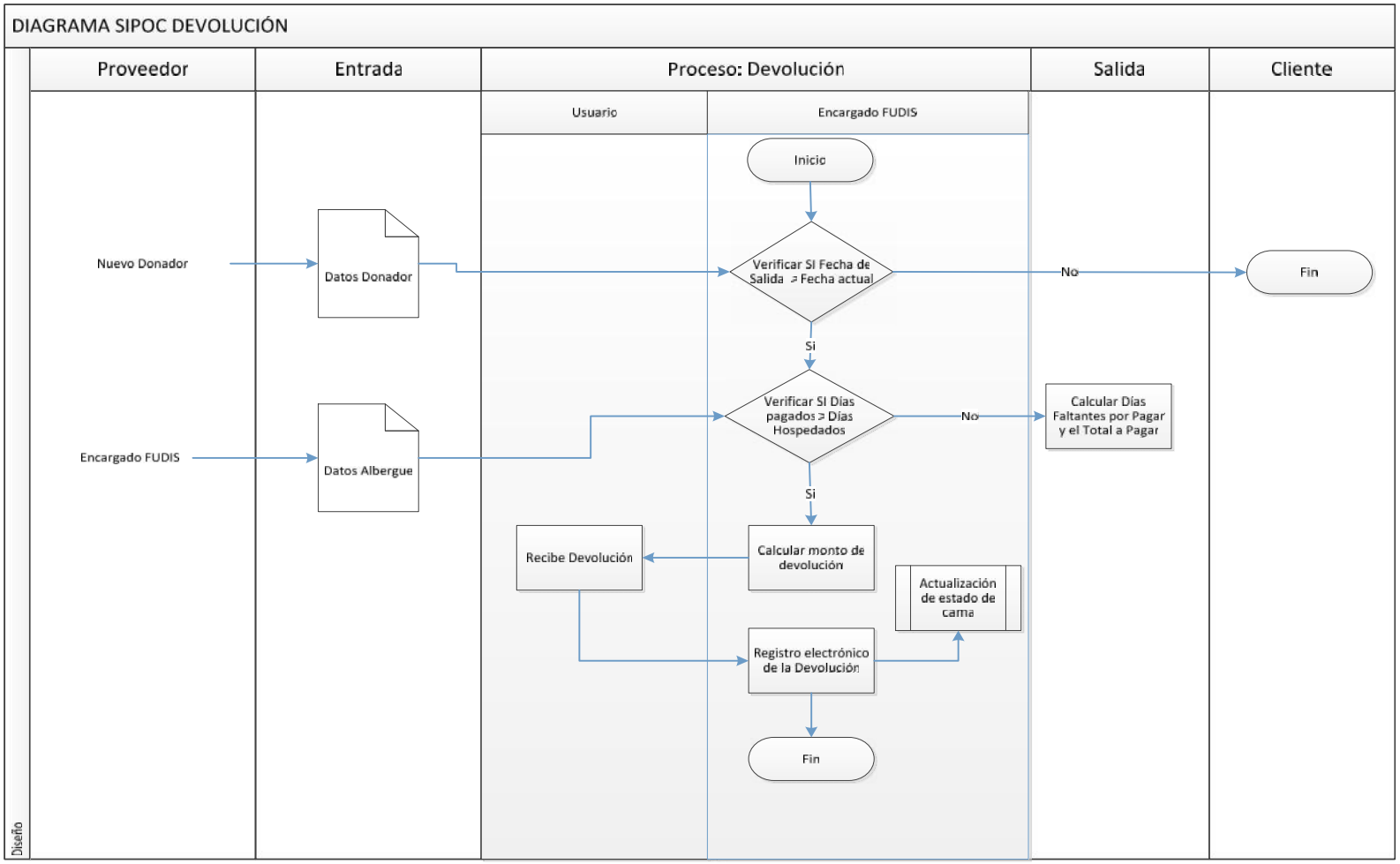


Ilustración 17 Diagrama SIPOC Devolución

Fuente:
Elaborado: Carlos Guamán – Jorge Recalde

3.6 IMPLEMENTACIÓN

3.6.1 PREPARACIÓN DEL ENTORNO OPERATIVO

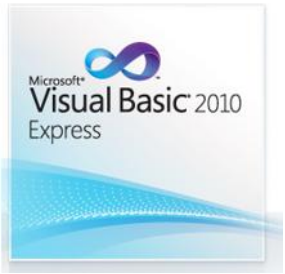
Cuando se está desarrollando el código de una aplicación, es muy conveniente disponer de un entorno de desarrollo local en el que podamos trabajar sin afectar al servidor en producción. Por ejemplo.

- Un sistema operativo Windows
- Un servidor de aplicaciones
- Una base de datos SQL SERVER Express Edition
- Un software escrito en el lenguaje de programación Visual Basic

Aspectos básicos

El desarrollo de clientes en Windows implica tres modelos principales: C++ nativo para programar directamente sobre las API de Windows, código administrado .NET con Win Forms o Windows Presentation Foundation (WPF) y código administrado .NET con Silverlight para el desarrollo rápido de aplicaciones. Puede escribir para cada uno de estos entornos de programación y otros más con Visual Studio, el entorno de desarrollo integrado (IDE) de Microsoft. El vídeo explicará cómo elegir el entorno de programación adecuado para cada situación.

Descarga



Modo de Programación

Win32 es una interfaz de programación de aplicaciones (API) utilizada para crear todo tipo de aplicaciones Windows. Win32 proporciona servicios y elementos de interfaz de usuario a las aplicaciones. Las

aplicaciones que están escritas en Win32 tienen acceso al conjunto más amplio de características de Windows.

3.6.2 DEFINICIÓN DE POLÍTICAS DE BACKUP

Una política de Backup debe comprender en mayor medida todos los aspectos de una copia de seguridad, entre los que cabe mencionar:

- Ámbito de aplicación establece el nivel de protección que hay que asignar a las copias realizadas. Dicha especificación del influir en los siguientes apartados de datos de usuarios, copia del sistema operativo o copia de aplicaciones. Periodicidad de las copias que se van a realizar.
- Elección del soporte para el backup.
- Método empleado (absoluto, incremental o combinaciones de ambos).
- Ubicación final de las copias.
- Responsables y responsabilidades.
- Definiciones pertinentes.

1. El sistema del que se pretende realizar copias de seguridad está formado por dos puestos de trabajo. Todos ellos pertenecen a una subred privada, obteniendo acceso al exterior a través de un Router con NAT y mejorando de este modo la seguridad interna.
2. Las copias se van a realizar en dos soportes distintos: Disco duro, y la nube. Los dos están en un servidor local que después se conectan a una red LAN y actualizan el servidor de backup. El servidor de backup consta de 1 disco SATA de 500Gb.
3. Los puestos de trabajo, comparten (previa autenticación del usuario local o del usuario de copia remoto) el directorio de documentos ubicado en una partición de 1.5Gb en DropBox. En principio dicho directorio no contempla la presencia de datos sensibles. El backup de estos queda bajo la responsabilidad del usuario del puesto de trabajo.
4. En lo que respecta al servidor de backup, las copias se almacenarán de modo jerárquico y estructurado en subdirectorios por su estación de trabajo.

5. La periodicidad con la que se realicen las copias dependerá del soporte: la copia a disco se hará semanal, automatizándola para que comience cada viernes a partir de las 22:00.
6. Todos los Backups serán absolutos.
7. El control de la integridad de las copias se realizará a través del empleo de checksums escogiendo un porcentaje de archivos aleatorios, y comparándolos con los originales.

3.6.3 INSTALACIÓN DE ENTORNO DE DESARROLLO Y DE PRUEBA

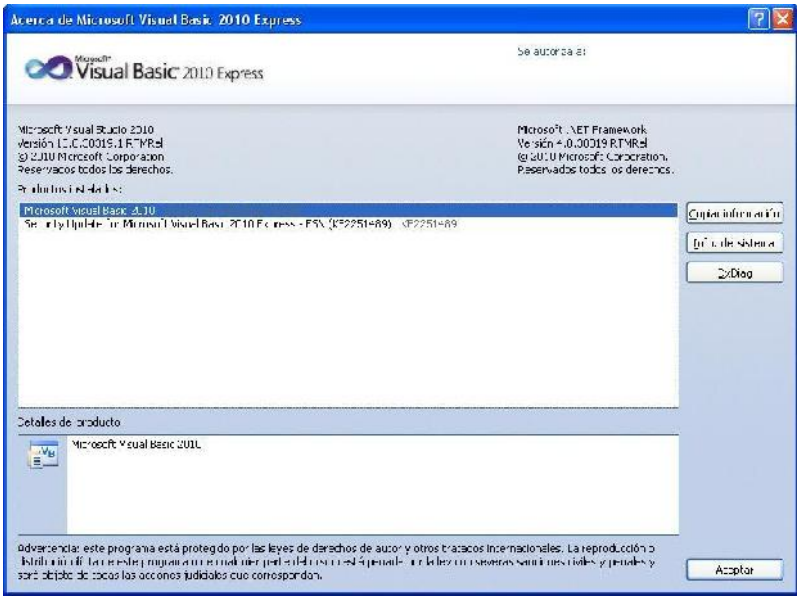


Ilustración 18 Entorno de Desarrollo

Fuente: http://www.historiasdeuncazique.com/articulo/48/descargar-gratis-visual-basic-2010-express-edition.html
Elaborado:

Se iniciará el asistente para la instalación d Microsoft Visual Studio 2010 (versión de evaluación). Pulsaremos en "Instalar Microsoft Visual Studio 2010":



Ilustración 19 Instalación IDE

Fuente: http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=517
Elaborado:

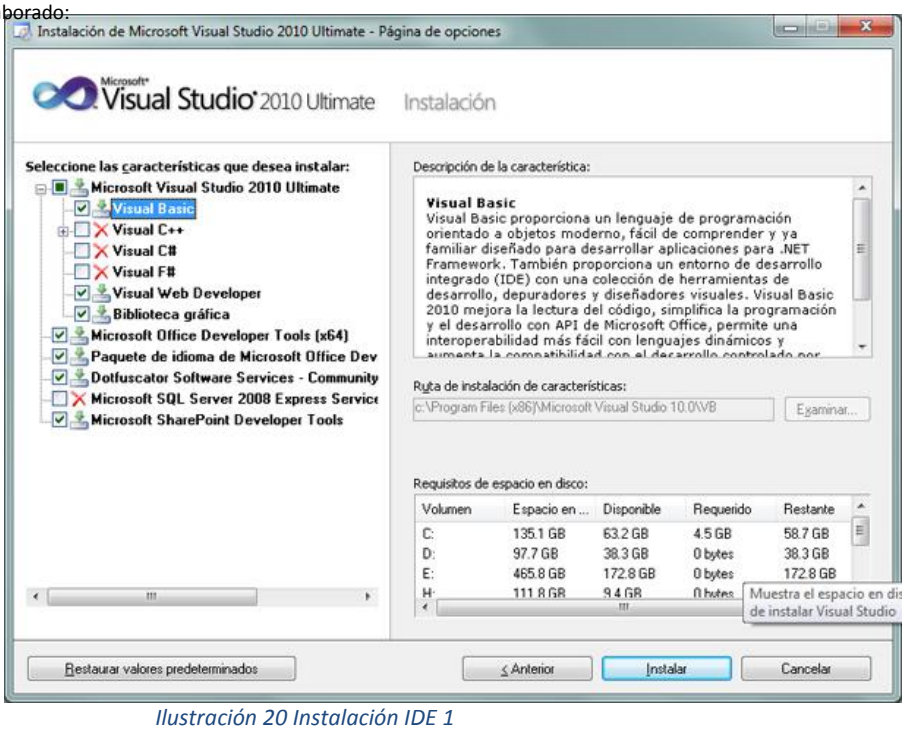


Ilustración 20 Instalación IDE 1

Fuente: http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=517
Elaborado:

Para más detalle de la instalación de Visual Studio Express Edition (Véase http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=517)

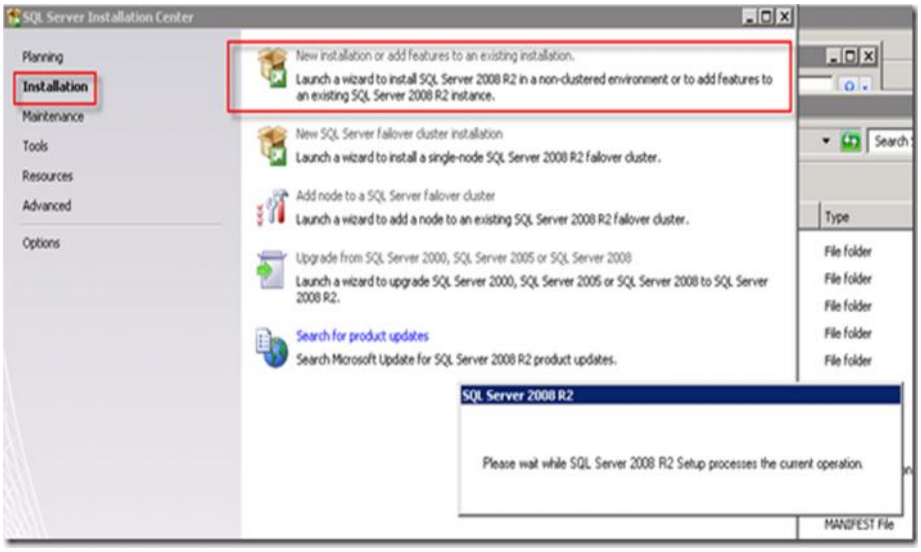


Ilustración 21 Instalación BDD

Fuente: http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=517
Elaborado:

Para más detalle de la instalación de SQL Server 2008R Express Edition (Véase http://www.elblogdedynamicscrm.com/post/2011/03/30/Instalando-Microsoft-Dynamics-CRM-2011-Parte-II%E2%80%93Instalacion.aspx)

3.6.4 PARAMETRIZACIÓN PRELIMINAR

- Nombre Aplicación: PROGRAMA DISERTACION DE GRADO.
- Nombre del Servidor: Data Source =.\SQLEXPRESS;AttachDbFilename =|DataDirectory|\FUDIS.mdf;Integrated Security=True; User Instance=True".
- Carpeta compartida: C:\Users\...\Dropbox\Tesis FUDIS\VISUAL BASIC TESIS\PROGRAMA DISERTACION DE GRADO
- Inicialización de IDs: Inicializar los IDs de cada una de las tablas en el DataSet
- Tipo vistas: Todas las tablas tienen una relación de vistas
- Numero Dígitos: Estos deberán estar de acuerdo a la Base de Datos

- Artículos en stock: Muestra artículos que hay en inventario
- Mostrar Mensajes de Texto: Cada vez que se realice una acción en cualquier formulario de la aplicación
- Botón guardar: Guarda los datos implicados en el formulario

3.6.5 DESARROLLO DE CUSTOMIZACIONES E INTERFACES

Con la ayuda de Visual Studio podremos desarrollar interfaces que ya vienen por defecto y crear posibles diseños de interfaces. Pero con la ayuda de los procedures creados en SQL SERVER 2008R y los asistentes podremos elaborar Interfaces elaboradas al gusto de los desarrolladores. Por ejemplo:

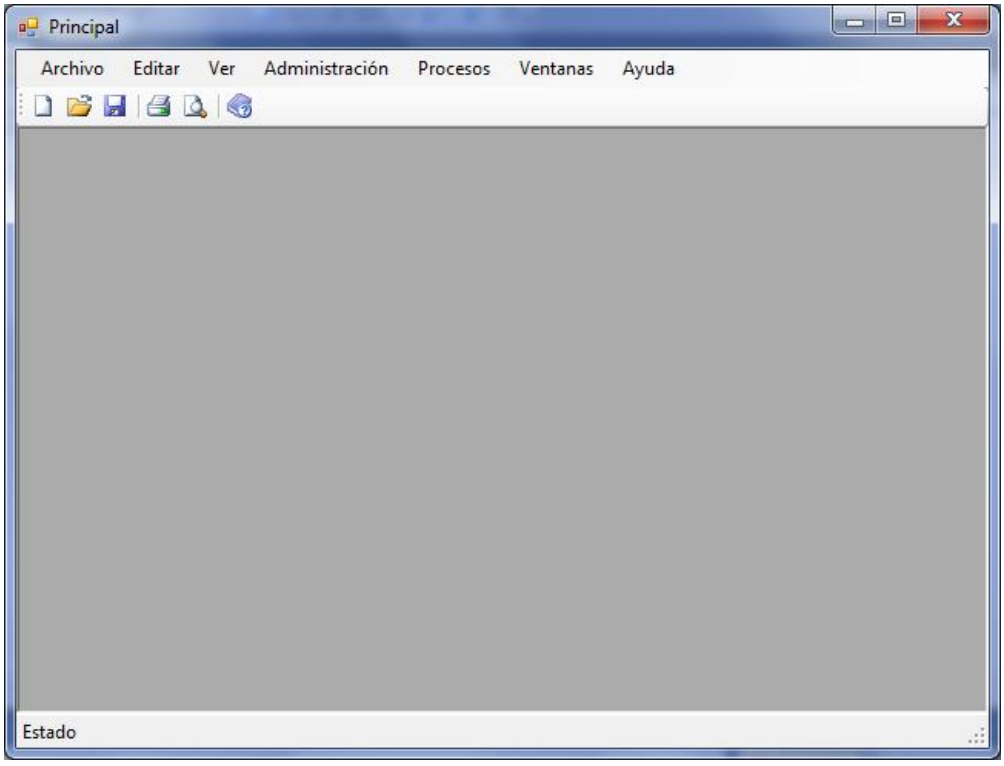


Ilustración 22Interface Principal

Fuente: Visual Studio 2010 Express Edition
Elaborado: Carlos Guamán – Jorge Recalde

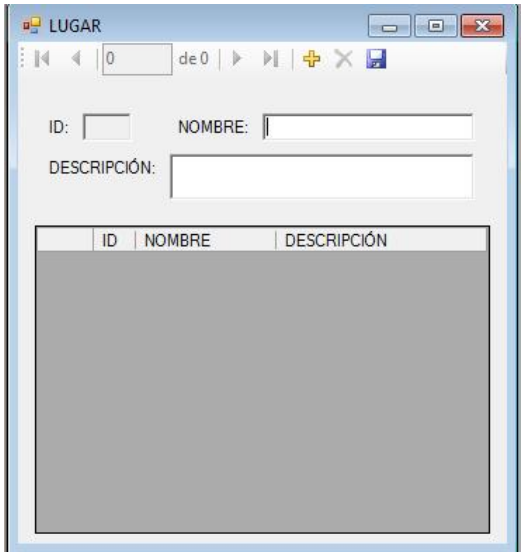


Ilustración 23 Interface Secundaria

Fuente: Visual Studio 2010 Express Edition
Elaborado: Carlos Guamán – Jorge Recalde

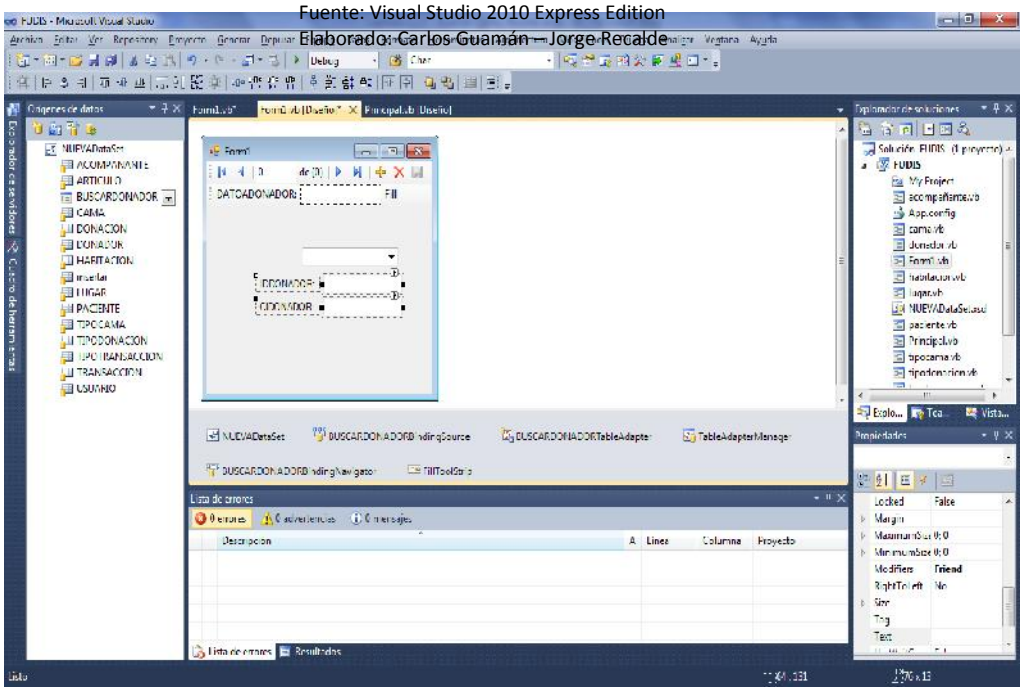


Ilustración 24 Interface Elaborada

Fuente: Visual Studio 2010 Express Edition
Elaborado: Carlos Guamán – Jorge Recalde

Además se añade un código de programación extra en el cual damos a conocer la colaboración del procedure dentro de la interface

```
  Private Sub FillToolStripButton_Click_1(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
FillToolStripButton.Click
        Try
Me.BUSCARDONADORTableAdapter.Fill(Me.NUEVADataSet.BUSCARDONADO
R, DATOADONADORToolStripTextBox.Text)
```

```
            Catch ex As System.Exception
                System.Windows.Forms.MessageBox.Show(ex.Message)
            End Try
        End Sub
```

3.6.6 DESARROLLO DE PROGRAMAS PARA MIGRACIÓN DE ARCHIVOS MAESTROS Y PRUEBAS DE MIGRACIÓN

Una vez detallado los requerimientos con el usuario hemos desarrollado un prototipo de la aplicación se hace pruebas de transferencia del programa a la base de datos de acuerdo a las circunstancias descritas en las entrevistas, para esto se han realizado algunas pruebas con el prototipo con datos aleatorios e interactuando con cada una de las tablas a ver si se están trasladando todos los datos ingresados.

Podemos decir que en todas las tablas que interactúa las administraciones no hay ninguna falencia, sin embargo en los procesos hay algunas parámetros que deben efectuarse de mejor manera, a mayor detalle hemos podido localizar que cuando se realiza una operación de devolución no se puede registrar bien el traspaso de información entre el prototipo y la base de datos.

Hemos constatado que todos los datos se han ingresado exitosamente sin problema alguno, sin poner mayor énfasis en el proceso de devolución ya que pudimos establecer cuál era el error y hemos corregido el mismo para que actúe bien el prototipo en la transferencia de todos los datos. Para muestra un ejemplo de datos ingresados exitosamente.

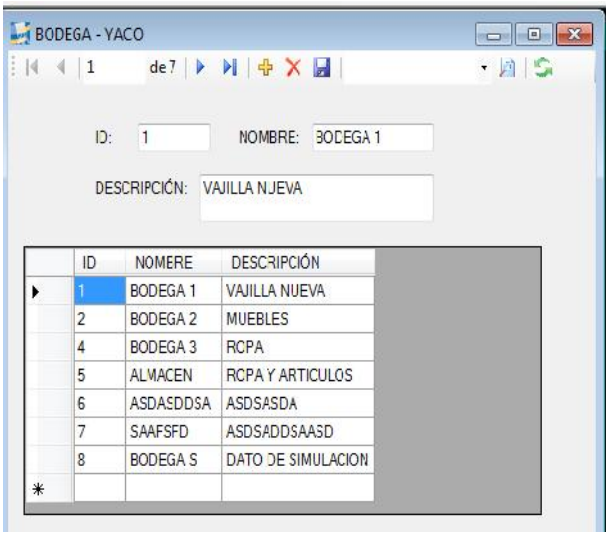


Ilustración 25 Pruebas de Migración
Fuente: Visual Studio 2010 Express Edition
Elaborado: Carlos Guamán – Jorge Recalde

3.6.7 SIMULACIÓN PRELIMINAR DEL ENLATADO EN PRODUCCIÓN

Operaciones	Variantes	Complejidad	Cant	Resultado
Ingreso al sistema	Usuario y clave	Baja	4	Ingreso existoso al sistema
Ingreso a Administraciones	Elección de administración	Baja	4	Ingreso a los diferentes formularios
Ingreso a Bodega	Nombre y descripción de la bodega	Intermedia	3	Registro en la BDD con los datos proporcionados
Ingreso a Cama	Habitación, tipo de cama, combre de cama disponibilidad y descripción	Alta	5	Registro en la BDD con los datos proporcionados
Ingreso a Paciente	Cedula, nombres, apellidos, provincia, ciudad fecha de naciemiento, hopistal tratante, hihstoria clinica, medico trantante, tratamiento, cedula acompañante y nombre del acompañante	Muy Alta	3	Ingreso de los carctres en cada uno de los cuadros de texto. Registro en la BDD con los datos proporcionados
Ingreso a Registro Diario	Cedula, cama monto, fecha actual, fecha de inicio de transacion y fecha fin de	Muy Alta	4	Calculo de Dias de hospedaje. Calculo del valor a pagar.

	transaccion			Registro en la BDD con los datos proporcionados
Ingreso a Donación Entrada	Nombre donador, bodega, fecha de entrada, tipo de entrada, nombre del articulo, cantidad y descripción	Muy Alta	5	Reconocimiento del donador al momento de ingresar su cedula. Suma de inventario de acuerdo al nombre del articulo. Ubicación del articulo en el lugar adecuado. Registro en la BDD con los datos proporcionados
Ingreso a Donación Salida	Beneficiario, bodega , fecha de salida, tipo de salida, nombre del articulo, cantidad y descripcion	Muy Alta	5	Resta de inventario de acuerdo al nombre del articulo. Ubicación del articulo en el lugar adecuado. Registro en la BDD con los datos proporcionados.
Impresión	Nombres, dirección, fecha, CI o RUC, telefono ciudad, detalle, cantidad y valor unitario	Alta	2	Impresión de la factura con los datos proporcionados. Calculo del valor total. Calculo del subtotal sin iva Calculo del valor total con iva.
Ingreso Guía Telefónica	Nombres, apellidos, empresa	Intermedia	12	Búsqueda de datos por parámetros. Mostrar los datos proporcionados e identificarlos.
Ingreso a Inventario	Nombre, bodega	Baja	6	Búsqueda de datos por parámetros. Mostrar los datos

				proporcionados e identificarlos.

Tabla 6 Simulación Preliminar

Fuente: http://www.escribimos.com.ar/www/soft.htm
Elaborado: Carlos Guamán – Jorge Recalde

3.6.8 CAPACITACIÓN DE USUARIOS FINALES

Durante las reuniones con la representante de FUDIS nos supo manifestar que no tenían conocimiento en manejo de computadoras y si les podríamos impartir una capacitación para que los usuarios comprendan el uso correcto del software que se les hará entrega al finalizar esta aplicación elaborada para la disertación de grado.

Fecha	ACTIVIDADES
27 de Mayo de 2013	Instalación del sistema para la simulación preliminar, además de los instrumentos y software necesario para que pueda funcionar correctamente el sistema
29 de Mayo de 2013	Este día se alisto una laptop la cual estaba nueva y sin software previos que permitirían al usuario trabajar de manera efectiva, se procedió a instalar las herramientas de ofimática necesarias así como un antivirus para proteger a los archivos de cualquier código malicioso o peligro externo.
30 de Mayo de 2013	Se realizó la instalación del servicio DropBox, como forma de compartir archivos y se los mantenga como un respaldo en la red; el resto de la tarde se capacitó al usuario en el uso de DropBox y compartición de archivos, se ejecutaron las pruebas necesarias en las dos estaciones de trabajo que posee la fundación, de tal manera que el usuario pueda trabajar desde casa y desde la oficina, facilitando así su trabajo
31 de Mayo de 2013	La capacitación de uso del programa con algunos inconsistencias en el sistema por la falta de utilización de herramientas informáticas
2 de Junio de 2013	La capacitación de todo el sistema de simulación Preliminar llega a su fin con mejores resultados que el día anterior.

Tabla 7 Capacitación de Usuarios Finales

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

3.6.9 AJUSTES DE PARAMETRIZACIÓN, CUSTOMIZACIONES E INFORMES

Una vez finalizada la capacitación y entregada la simulación preliminar del producto podemos identificar los ajustes necesarios para la modificación del producto desarrollado.

Operaciones	Variantes	Correcciones	Modificaciones
Ingreso al sistema	Usuario y clave	Ingreso solo mayúsculas	La comprobación se realizará con esos parámetros
Ingreso a Administraciones	Elección de administración	No	No
Ingreso a Bodega	Nombre y descripción de la bodega	Ingreso solo mayúsculas	El registro de usuario se realizara solo con mayúsculas
Ingreso a Cama	Habitación, tipo de cama, combre de cama disponibilidad y descripción	Ingreso solo mayúsculas	El registro de usuario se realizara solo con mayúsculas
Ingreso a Paciente	Cedula, nombres, apellidos, provincia, ciudad fecha de naciemiento, hopistal tratante, hihstoria clinica, medico trantante, tratamiento, cedula acompañante y nombre del acompañante	Búsqueda por CI. Ingreso de caracteres en mayúsculas	La selección de fecha se realiza mediante un datetimepicker y que este en la fecha actual.
Ingreso a Registro Diario	Cedula, cama monto, fecha actual, fecha de inicio de transacion y fecha fin de transaccion	Búsqueda por CI. Elección de fecha final. Ingreso de caracteres en mayúsculas	La selección de fecha se realiza mediante un datetimepicker y que este en la fecha actual.
Ingreso a Donación Entrada	Nombre donador, bodega, fecha de entrada, tipo de entrada, nombre del articulo, cantidad y descripción	Búsqueda por CI. Ingreso de caracteres en mayúsculas. Búsqueda de productos por lugar	Se mostrará diferentes tipos de formularios para la elección de los datos
Ingreso a Donación Salida	Beneficiario, bodega , fecha de salida,	Búsqueda articulo Ingreso de	Se mostrará diferentes tipos de

	tipo de salida, nombre del articulo, cantidad y descripcion	caracteres en mayúsculas.	formularios para la elección de los datos
Impresión	Nombres, dirección, fecha, CI o RUC, telefono ciudad, detalle, cantidad y valor unitario	No	No
Ingreso Guía Telefónica	Nombres, apellidos, empresa	No	No
Ingreso a Inventario	Nombre, bodega	No	No
Ingreso a Devolución	Nombre paciente, fecha de fin de transacción	Búsqueda por CI. Selección por datetimepicker la fecha fin de la transacción	Se mostrará diferentes tipos de formularios para la elección de los datos

Tabla 8 Ajustes de Parametrización

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

3.6.10 SIMULACIÓN FINAL DE ENTORNO DE PRODUCCIÓN

En la simulación preliminar del producto se realizó con datos reales, ahora en este punto, con los usuarios, realizaremos una práctica con las correcciones y/ o modificaciones especificadas en el ajuste de la parametrización.

Operaciones	**Variantes**	**Complejidad**	**Resultado**	**Modificaciones**
Ingreso al sistema	Usuario y clave	Baja	Ingreso existoso al sistema	La comprobación se realizará con esos parámetros
Ingreso a Administraciones	Elección de administración	Baja	Ingreso a los diferentes formularios	No
Ingreso a Bodega	Nombre y descripción de la bodega	Intermedia	Registro en la BDD con los datos proporcionados	El registro de usuario se realizara solo con mayúsculas
Ingreso a Cama	Habitación, tipo de cama, combre de cama disponibilidad y descripción	Alta	Registro en la BDD con los datos proporcionados	El registro de usuario se realizara solo con

				mayúsculas
Ingreso a Paciente	Cedula, nombres, apellidos, provincia, ciudad fecha de naciemiento, hopistal tratante, hihstoria clinica, medico trantante, tratamiento, cedula acompañante y nombre del acompañante	Muy Alta	Ingreso de los carctres en cada uno de los cuadros de texto. Registro en la BDD con los datos proporcionados	La selección de fecha se realiza mediante un datetimepicker y que este en la fecha actual.
Ingreso a Registro Diario	Cedula, cama monto, fecha actual, fecha de inicio de transacion y fecha fin de transaccion	Muy Alta	Cálculo de Dias de hospedaje. Calculo del valor a pagar. Registro en la BDD con los datos proporcionados	La selección de fecha se realiza mediante un datetimepicker y que este en la fecha actual.
Ingreso a Donación Entrada	Nombre donador, bodega, fecha de entrada, tipo de entrada, nombre del articulo, cantidad y descripción	Muy Alta	Reconocimiento del donador al momento de ingresar su cedula. Suma de inventario de aceurdo al nombre del articulo. Ubicación del articulo en el lugar adecuado. Registro en la BDD con los datos proporcionados	Se mostrará diferentes tipos de formularios para la elección de los datos
Ingreso a Donación Salida	Beneficiario, bodega , fecha de salida, tipo de salida, nombre del articulo, cantidad y descripcion	Muy Alta	Resta de inventario de aceurdo al nombre del articulo. Ubicación del articulo en el lugar adecuado. Registro en la BDD con los datos proporcionados.	Se mostrará diferentes tipos de formularios para la elección de los datos
Impresión	Nombres, dirección, fecha, CI o RUC, telefono ciudad,	Alta	Impresión de la factura con los datos	Imprime en hoja de papel o un archivo el

	detalle, cantidad y valor unitario		proporcionados. Calculo del valor total. Calculo del subtotal sin iva Calculo del valor total con iva.	cual se puede utilizar como respaldo
Ingreso Guía Telefónica	Nombres, apellidos, empresa	Intermedia	Búsqueda de datos por parámetros. Mostrar los datos proporcionados e identificarlos.	Muestra los datos de los donadores y funcionarios del albergue
Ingreso a Inventario	Nombre, bodega	Baja	Búsqueda de datos por parámetros. Mostrar los datos proporcionados e identificarlos.	Muestra los datos del inventario y muestra la bodega que tiene
Ingreso a Devolución	Nombre paciente, fecha de fin de transacción	Muy Alta	Búsqueda por CI. Selección por datetimepicker la fecha fin de la transacción	Se mostrará diferentes tipos de formularios para la elección de los datos

Tabla 9 Simulación Final
Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

3.6.11 HABILITACIÓN DEL ENTORNO DE PRODUCCIÓN

En la habilitación del producto hemos permitido el acceso a los usuarios en este caso a los encargados en operar con el sistema desarrollado para que empiecen a funcionar los controles de la aplicación.

Además toda la información transferida está controlada a consideración del prototipo realizado anteriormente y contra la documentación de respaldo. Se prueban los informes en este caso serán los Data Grid View donde se observa la información de cada una de las tablas o vistas utilizadas en el desarrollo de la aplicación y las impresiones que cada uno de los formularios tiene de acuerdo a la funcionalidad que desee.

3.7 PRUEBAS

3.7.1 PRUEBAS ALFA

Para las pruebas alfa hemos desarrollado un formato donde se puede evaluar al sistema implementado durante su ejecución, esta evaluación será realizada siempre a través de los desarrolladores como

evaluadores mientras el usuario final prueba a detalle cada una de las funcionalidades.

Formulario	Controles de ingreso de texto	Validación	Mensajes de aviso	Conexión a otros formularios	CRUD
LOGIN					
PRINCIPAL					
CAMA					
DONADOR					
HABITACION					
LUGAR					
PACIENTE					
TIPO DE CAMA					
USUARIO					
ACOMPAÑANTE					
DONACION DINERO					
DONACION ARTICULO					
DIARIO					
INVENTARIO					
PAGO					

Para ver los resultados del formato antes mensionado véase Anexos PRUEBAS ALFA, aquí en contraremos a detalle el desarrollo de las pruebas con los diferentes actores que intervienen en el sistema.

3.7.2 PRUEBAS BETA

Para las pruebas beta los usuarios finales reconocidos como actores serán capaces de probar el sistema sin la ayuda de los desarrolladores que les brinden ayuda en el momento que encuentren fallos o errores durante la sesión de pruebas. El actor encontrará el formulario donde ocurre algún error y notificará a detalle que sucede. Para el ejemplo tenemos este formato para el usuario final.

ACTOR INVOLUCRADO	FORMULARIO	DETALLE
Administrador 1		
Administrador 2		
Tecnico		
Usuario		

Para ver los resultados del formato antes mensionado véase Anexos PRUEBAS BETA, aquí en contraremos a detalle el desarrollo de las pruebas con los diferentes actores que intervienen en el sistema, detallado con la forma y palabras del usuario final.

4. CAPÍTULO 4: MANUAL TÉCNICO

4.1 ESTRUCTURA DE BASE DE DATOS

Diagrama conceptual

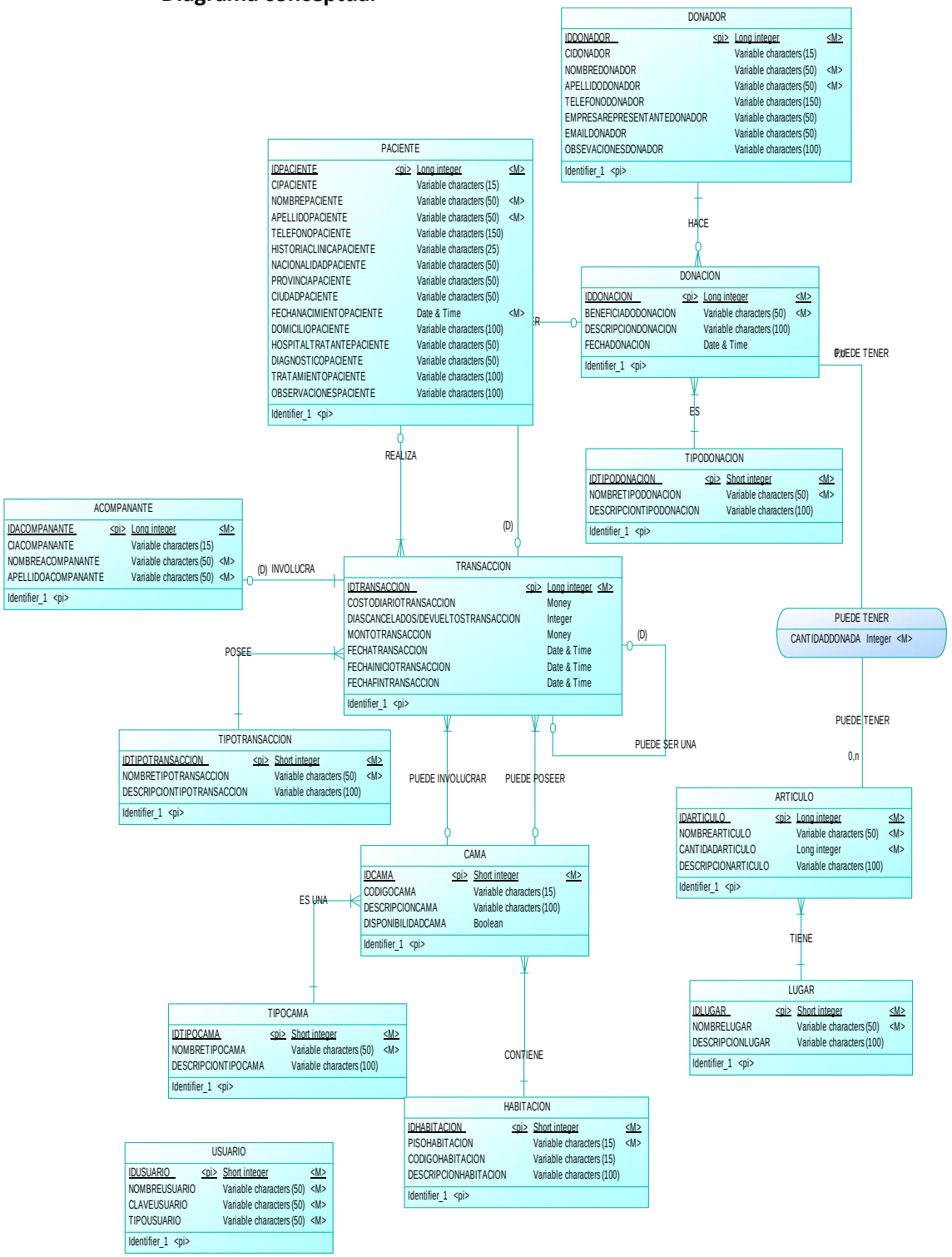


Ilustración 2626 Diagrama conceptual

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Diagrama Físico

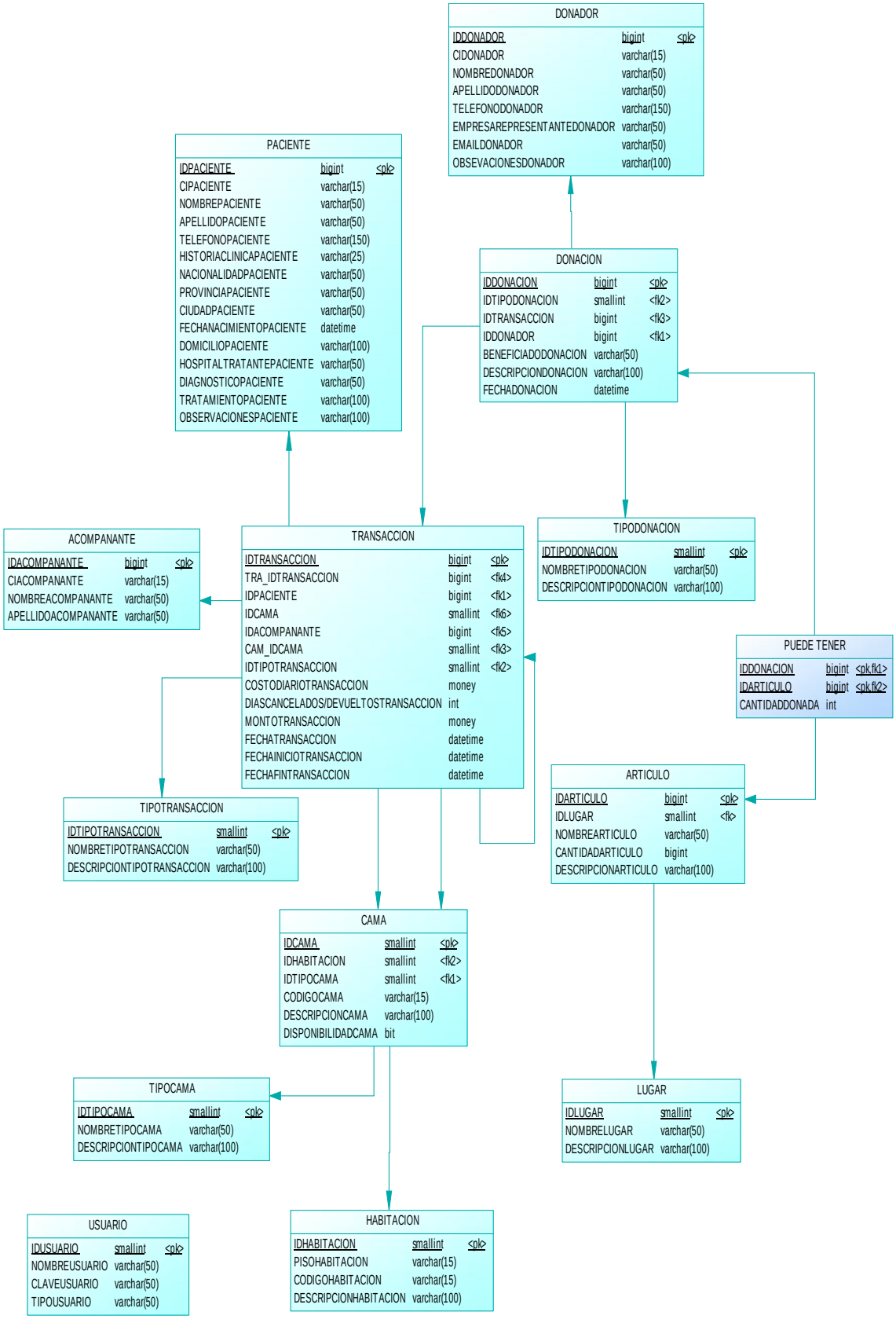


Ilustración 2727 Diagrama Físico

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

4.2 DETALLE DE LAS TABLAS (CAMPOS, TIPO DE DATOS)

Nombre	Código
ACOMPANANTE	ACOMPANANTE
ARTICULO	ARTICULO
CAMA	CAMA
DONACION	DONACION
DONADOR	DONADOR
HABITACION	HABITACION
LUGAR	LUGAR
PACIENTE	PACIENTE
TIPOCAMA	TIPOCAMA
TIPODONACION	TIPODONACION
TIPOTRANSACCION	TIPOTRANSACCION
TRANSACCION	TRANSACCION
USUARIO	USUARIO

Tabla 10 Lista de Tablas
Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "ACOMPANANTE"

Nombre	Código	Tipo de Datos
IDACOMPANANTE	IDACOMPANANTE	bigint
CIACOMPANANTE	CIACOMPANANTE	Varchar(15)
NOMBREACOMPANANTE	NOMBREACOMPANANTE	Varchar(50)
APELLIDOACOMPANANTE	APELLIDOACOMPANANTE	Varchar(50)

Tabla 11 Lista de Columnas ACOMPAÑANTE
Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "ARTICULO"

Nombre	Código	Tipo de Datos
IDARTICULO	IDARTICULO	Bigint
IDLUGAR	IDLUGAR	Smallint
NOMBREARTICULO	NOMBREARTICULO	Varchar(50)
CANTIDADARTICULO	CANTIDADARTICULO	Bigint
DESCRIPCIONARTICULO	DESCRIPCIONARTICULO	Varchar(100)

Tabla 12 Lista de Columnas ARTICULO

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "CAMA"

Nombre	Código	Tipo de Datos
IDCAMA	IDCAMA	Smallint
IDHABITACION	IDHABITACION	Smallint
IDTIPOCAMA	IDTIPOCAMA	Smallint
CODIGOCAMA	CODIGOCAMA	Varchar(15)
DESCRIPCIONCAMA	DESCRIPCIONCAMA	Varchar(100)
DISPONIBILIDAD	DISPONIBILIDAD	Bit

Tabla 13 Lista de Columnas CAMA

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "DONACION"

Nombre	Código	Tipo de Datos
IDDONACION	IDDONACION	Bigint
IDTIPODONACION	IDTIPODONACION	Smallint
IDTRANSACCION	IDTRANSACCION	Bigint
IDDONADOR	IDDONADOR	Bigint
BENEFICIADODONACION	BENEFICIADODONACION	Varchar(50)
DESCRIPCIONDONACION	DESCRIPCIONDONACION	Varchar(100)
FECHADONACION	FECHADONACION	Datetime

Tabla 14 Lista de Columnas DONACION
Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "DONADOR"

Nombre	Código	Tipo de Datos
IDDONADOR	IDDONADOR	Bigint
CIDONADOR	CIDONADOR	Varchar(15)
NOMBREDONADOR	NOMBREDONADOR	Varchar(50)
APELLIDODONADOR	APELLIDODONADOR	Varchar(50)
TELEFONODONADOR	TELEFONODONADOR	Varchar(150)
EMPRESAREPRESENTANTEDONADOR	EMPRESAREPRESENTANTEDONADOR	Varchar(50)
EMAILDONADOR	EMAILDONADOR	Varchar(50)
OBSERAVCIONDONADOR	OBSERAVCIONDONADOR	Varchar(100)

Tabla 15 Lista de Columnas DONADOR

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "HABITACION"

Nombre	Código	Tipo de Dato
IDHABITACION	IDHABITACION	Smallint
PISOHABITACION	PISOHABITACION	Varchar(15)
CODIGOHABITACION	CODIGOHABITACION	Varchar(15)
DESCRIPCIONHABITACION	DESCRIPCIONHABITACION	Varchar(100)

Tabla 16 Lista de Columnas HABITACION
Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "LUGAR"

Nombre	Código	Tipo de Dato
IDLUGAR	IDLUGAR	smallint
NOMBRELUGAR	NOMBRELUGAR	Varchar(50)
DESCRIPCIONLUGAR	DESCRIPCIONLUGAR	Varchar(100)

Tabla 17 Lista de Columnas LUGAR
Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "PACIENTE"

Nombre	Código	Tipo de Dato
IDPACIENTE	IDPACIENTE	Bigint
CIPACIENTE	CIPACIENTE	Varchar(15)
NOMBREPACIENTE	NOMBREPACIENTE	Varchar(50)
APELLIDOPACIENTE	APELLIDOPACIENTE	Varchar(50)
TELEFONOPACIENTE	TELEFONOPACIENTE	Varchar(150)
HISTORIACLINICAPACIENTE	HISTORIACLINICAPACIENTE	Varchar(25)
NACIONALIDADPACIENTE	NACIONALIDADPACIENTE	Varchar(50)
PROVINCIAPACIENTE	PROVINCIAPACIENTE	Varchar(50)
CIUDADPACIENTE	CIUDADPACIENTE	Varchar(50)
FECHANACIMIENTOPACIENTE	FECHANACIMIENTOPACIENTE	Datetime
DOMICILIOPACIENTE	DOMICILIOPACIENTE	Varchar(100)
HOSPITALTRATANTEPACIENTE	HOSPITALTRATANTEPACIENTE	Varchar(50)
DIAGNOSTICOPACIENTE	DIAGNOSTICOPACIENTE	Varchar(50)
TRATAMIENTOPACIENTE	TRATAMIENTOPACIENTE	Varchar(100)
OBSERVACIONESPACIENTE	OBSERVACIONESPACIENTE	Varchar(100)

Tabla 18 Lista de Columnas PACIENTE

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "TIPOCAMA"

Nombre	Código	Tipo de Datos
IDTIPOCAMA	IDTIPOCAMA	Smallint
NOMBRETIPOCAMA	NOMBRETIPOCAMA	Varchar(50)
DESCRIPCIONTIPOCAMA	DESCRIPCIONTIPOCAMA	Varchar(100)

Tabla 19 Lista de Columnas TIPOCAMA

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "TIPODONACION"

Nombre	Código	Tipo de Datos
IDTIPODONACION	IDTIPODONACION	Smallint
NOMBRETIPODONACION	NOMBRETIPODONACION	Varchar(50)
DESCRIPCIONTIPODONACION	DESCRIPCIONTIPODONACION	Varchar(100)

Tabla 20 Lista de Columnas TIPODONACION

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla "TIPOTRANSACCION"

Nombre	Código	Tipo de Datos
IDTIPOTRANSACCION	IDTIPOTRANSACCION	Smallint
NOMBRETIPOTRANSACCION	NOMBRETIPOTRANSACCION	Varchar(50)
DESCRIPCIONTIPOTRANSACCION	DESCRIPCIONTIPOTRANSACCION	Varchar(100)

Tabla 21 Lista de Columnas TIPOTRANSACCION

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

Lista de columnas de la tabla “TRANSACCION”

Nombre	Código	Tipo de Datos
IDTRANSACCION	IDTRANSACCION	Bigint
TRA_IDTRANSACCION	TRA_IDTRANSACCION	Bigint
IDACOMPANANTE	IDACOMPANANTE	Bigint
IDCAMA	IDCAMA	Smallint
IDPACIENTE	IDPACIENTE	Bigint
CAM_IDCAMA	CAM_IDCAMA	Smallint
IDTIPOTRANSACCION	IDTIPOTRANSACCION	Smallint
COSTODIARIO	COSTODIARIO	Money
DIASCANCELADOS_DEVUELTOSTRANSACCION	DIASCANCELADOS_DEVUELTOSTRANSACCION	Int
MONTOTRANSACCION	MONTOTRANSACCION	Money
FECHATRANSACCION	FECHATRANSACCION	Datetime
FECHAINICIOTRANSACCION	FECHAINICIOTRANSACCION	Datetime
FECHAFINTRANSACCION	FECHAFINTRANSACCION	Datetime

Tabla 22 Lista de Columnas TRANSACCION

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

4.3 EXPLICACIÓN DE LA RELACIÓN ENTRE TABLAS

NOMBRE	CÓDIGO	TABLA PADRE	TABLA HIJA
CONTIENE	CONTIENE	HABITACION	CAMA
ES	ES	TIPODONACION	DONACION
ES UNA	ES_UNA	TIPOCAMA	CAMA
HACE	HACE	DONADOR	DONACION
POSEE	POSEE	TIPOTRANSACCION	TRANSACCION
PUEDE INVOLUCRAR	PUEDE_INVOLUCRAR	TRANSACCION	CAMA
PUEDE SER	PUEDE_SER	DONACION	TRANSACCION
PUEDE TENER	PUEDE_TENER	DONACION	ARTICULO
REALIZA	REALIZA	PACIENTE	TRANSACCION
TIENE	TIENE	LUGAR	ARTICULO
VIENE CON	VIENE_CON	PACIENTE	ACOMPANANTE

Tabla 23 Relación entre Tablas

Fuente: Carlos Guamán – Jorge Recalde
Elaborado: Carlos Guamán – Jorge Recalde

4.4 CÓDIGO DE FUENTES DOCUMENTADOS EN MEDIO MAGNÉTICO

Para la explicación y documentación de los códigos fuentes empleados durante el desarrollo de la aplicación empezaremos con el código de cración de la base de datos, los procedimientos almacenados y vistas utilizadas.

También detallaremos los formularios utilizados cabe recalcar que solo utilizaremos el código fuente donde esta la interacción de los programadores más nolo que genera el lenguaje de programación.

Además esá detallado las clases utilizadas para los diferentes usos aplicadosdurante el desarrollo de formularios vistas y reportes.

Para mayor detalle del código fuente *Véase ANEXO 4 MANUAL DE USUARIO.*

4.5 DOCUMENTOS DE DEFINICIÓN INICIAL Y TODAS LAS MODIFICACIONES SOLICITADAS Y REALIZADAS

DEFINICIÓN INICIAL

- Analizar, diseñar, y desarrollar un sistema informático para el manejo de los procesos de donaciones, registro de caja, control de pacientes e inventario para el albergue FUDIS.
- Identificar los procesos que se van a automatizar.
- Analizar la situación actual de cada proceso a automatizar.
- Diseñar la mejor alternativa de solución para la automatización del proceso.
- Implementar una aplicación de fácil uso que sea accesible para los responsables de la administración de la información de los pacientes que se hospedan en el albergue FUDIS
- Proveer a la institución de un sistema informático para el correcto manejo de los datos de su personal.
- Usar una metodología de desarrollo para el correcto funcionamiento de la aplicación, de esta manera servir a la comunidad en general.
- Capacitar al personal indicado en la utilización del aplicativo.

- Desarrollar los manuales técnicos y de usuarios.
- Investigación de las herramientas que se usarán para el desarrollo del sistema.

MODIFICACIONES SOLICITADAS

- Ingreso de pacientes deben realizar por medio de nombre y apellido o por cédula.
- El acceso a lugar de almacenamiento debe realizar de mejor manera para la elección de la bodega y una mejor opción de selección.
- Solo debe escoger una fecha para el Registro Diario para que se pueda calcular el monto a pagar
- Botones más grandes y con letras para mejorar la efectividad de la aplicación
- Mayor detalle de cada una de las vistas que se muestran en los formularios.
- Que tenga en cada uno de las cajas de texto una indicación que datos pueden ingresarse.
- Presentar mensajes de notificación para la interacción de la aplicación con el usuario o encargado de la Fundación

CONCLUSIONES:

- Las necesidades del proyecto se apegaron a la realidad del manejo de la metodología RAD ya que sus características siempre fueron orientadas a sistemas de agiles y de fácil desarrollo y en periodos cortos de tiempo.
- El desarrollo tanto de los diagramas SIPOC como Mapa de Procesos ayuda a identificar los actores involucrados y facilita la interacción de los escenarios difíciles de identificar.
- Para una mejor visualización de las actividades, escenarios y actores en las cuales se desarrolla cada proceso es necesario realizar la Cadena de Valor.
- Poner todas las actividades que se realizarán en un lenguaje natural el cual el cliente y el desarrollador puedan entenderse es de mucha importancia en el momento de la entrega del producto.
- Los reportes realizados con este sistema facilitará la administración del albergue, para conocer la situación de cada una de sus actividades desarrolladas por la misma
- El desarrollo de un ambiente de interface amigable hacia el usuario del sistema ayuda a la no equivocación de la administración de los datos proporcionados por los usuarios del sistema.
- Establecer reglas claras de diseño y desarrollo ayuda a la óptima realización de un producto confiable y seguro que ayude a la Fundación FUDIS.
- Con el manejo sistematizado del manejo diario de pacientes se puede obtener un mejor reporte diario, semanal y mensual ya que toda la información está centralizada.
- El registro de las donaciones realizadas por los amigos o donadores de FUDIS, ayuda a mantener una estrecha relación entre las partes invlucradas para realizar nuevas donaciones.
- Mantener un registro de inventario ayuda de manera eficiente al conocimiento actual de bienes que posee la Fundación, ya que si existe perdidas o faltantes se puede saber en que lugar estaba y a quién se pudo haber enviado dicho artículo.

RECOMENDACIONES:

.

- Es necesario realizar un backup de la base de datos cada semana ya que con esto podremos mantener los datos del albergue asegurándolos desde un principio.
- El saber levantar bien la información a través de un diagramado de procesos es esencial para el desenvolvimiento de la información antes, durante y después de la entrega del producto desarrollado.
- La implementación de una auditoria de todas las actividades realizadas dentro de la base de datos, todas estas, asociadas al usuario que se conecta a dicho momento.
- Se recomienda el uso de un lector de barras para que así puedan ingresar con mayor rapidez los datos donde se requiere la lectura de datos electrónicos.
- Mediante el uso del sistema en fundaciones similares o afines a FUDIS ayudará al manejo sistematizado de las actividades desarrolladas a diario.
- Es recomendable el uso de Visual Studio 2010 y Sql Server Express porque se puede desarrollar aplicaciones amigables al usuario donde se obtienen mejores reultados en el manejo sistematizado de la fundación ya que se tiene un ambiente gráfico familiar a las aplicaciones que están en el Sistema Operativo que se estáutilizando.
- Es prudente utilizar el sistema continuamente ya que con la interacción de los encargados de parte de la Fundación FUDIS con el sistema ayudará a que no exista inconvenientes para el ingreso o administración de datos de los usuarios.
- Para la segunda versión de la aplicación desarrollada puede ser de mayor utilidad que se utilice una base de datos externa.
- Con la habilidad que se adquiere durante el uso del aplicativo se puede desarrollar una contabilidad integrada donde se tenga en cuenta todos los aspectos económicos de la Fundación para tener un sistema centralizado.
- Con la administración de los pacientes se puede manejar todos los diferentes niveles de discapacidad que tiene cada uno de los usuarios de

la Fundación ya que asi se tendrá un análisis de los casos más tratados en el Albergue.

- Es necesario que la Pontifica Universidad Católica del Ecuador, a través de sus autoridades, presten real atención a las relaciones humanas que se generan durante el desarrollo de una aplicación de esta índole ya que alguna mala interpretación con los directivos que manejan estas empresas o fundaciones pueden generar disconformidad con el trato entre las dos partes involucradas.
- La Facultad de Ingeniería debería ayudar a sus estudiantes de manera más directa a conseguir alternativas para el desarrollo de aplicaciones tanto de escritorio, web o móvil para que sean aplicadas en distintas organizaciones y no se queden como un trabajo más realizado.
- La Escuela de Sistemas debería ser participe en el conseguir cursos o seminarios para que los alumnos puedan adquirir conocimiento y experiencia en el desaroollo de aplicaciones, para que la calidad del producto entregado sea satisfactorio, tanto para el usuario final como para el desarrolladr del producto.

ANEXOS

ANEXO 1: PRUEBAS ALFA

Formulario	Controles de ingreso de texto	Validación	Mensajes de aviso	Conexión a otros formularios	CRUD
LOGIN	SI	SI	SI	Principal	No
PRINCIPAL	No	No	SI	Cama, donador, habitación, lugar, paciente, tipo de cama, donación, diario, inventario, pago	No
CAMA	SI	SI	SI	Ingreso cama, impresión	D,U
INGRESO CAMA	SI	SI	SI	cama	C,R,U,D
DONADOR	SI	SI	SI	Ingreso donador, impresión	D,U
INGRESO DONADOR	SI	SI	SI	Donador	C,R,U,D
HABITACION	SI	SI	SI	Ingreso habitación, impresión	D,U
CREAR HABITACIÓN	SI	SI	SI	Habitación	C,R,U,D
LUGAR	SI	SI	SI	Ingreso lugar, impresión	D,U
CREAR LUGAR	SI	SI	SI	Lugar	C,R,U,D
PACIENTE	SI	SI	SI	Ingreso lugar, impresión	D,U
CREAR PACIENTE	SI	SI	SI	Paciente	C,R,U,D
TIPO DE CAMA	SI	SI	SI	Ingreso tipo de cama, impresión	D,U
INGRESO TIPO DE CAMA	SI	SI	SI	Tipo de cama	C,R,U,D
USUARIO	SI	SI	SI	No	C,R,U,D
ACOMPAÑANTE	SI	SI	SI	Ingreso acompañante, impresión	D,U
CREAR ACOMPAÑANTE	SI	SI	SI	Acompañante	C,R,U,D
DONACION	SI	SI	SI	Ingreso	D,U

DINERO				donación dinero, impresión	
CREACION DONACION DINERO	SI	SI	SI	Donación dinero	C,R,U,D
DONACION ARTICULO	SI	SI	SI	Ingreso donación dinero, impresión	D,U
CREACIÓN DONACIÓN ARTICULO	SI	SI	SI	Donación dinero	C,R,U,D
DIARIO	SI	SI	No	No	R
INVENTARIO	SI	SI	SI	Ingreso articulo, impresión	D,U
CREACIÓN ARTICULO	SI	SI	SI	Inventario	C,R,U,D
PAGO	SI	SI	SI	Ingreso pago, impresión	D,U
CREACIÓN PAGO	SI	SI	SI	Pago	C,R,U,D

Tabla 24 Prueba alfa ADMINISTRADOR 1

Fuente: Fundación FUDIS

Elaborado: Carlos Guamán y Jorge Recalde

Formulario	**Controles de ingreso de texto**	**Validación**	**Mensajes de aviso**	**Conexión a otros formularios**	**CRUD**
LOGIN	SI	SI	SI	Principal	No
PRINCIPAL	No	No	SI	Cama, donador, habitación, lugar, paciente, tipo de cama, donación, diario, inventario, pago	No
CAMA	SI	SI	SI	Ingreso cama, impresión	No
INGRESO CAMA	SI	SI	SI	cama	SI
DONADOR	SI	SI	SI	Ingreso donador, impresión	No
INGRESO DONADOR	SI	SI	SI	Donador	SI
HABITACION	SI	SI	SI	Ingreso	No

				habitación, impresión	
CREAR HABITACIÓN	SI	SI	SI	Habitación	SI
LUGAR	SI	SI	SI	Ingreso lugar, impresión	No
CREAR LUGAR	SI	SI	SI	Lugar	SI
PACIENTE	SI	SI	SI	Ingreso lugar, impresión	No
CREAR PACIENTE	SI	SI	SI	Paciente	No
TIPO DE CAMA	SI	SI	SI	Ingreso tipo de cama, impresión	No
INGRESO TIPO DE CAMA	SI	SI	SI	Tipo de cama	No
USUARIO	SI	SI	SI	No	SI
ACOMPAÑANTE	SI	SI	SI	Ingreso acompañante, impresión	No
CREAR ACOMPAÑANTE	SI	SI	SI	Acompañante	No
DONACION DINERO	SI	SI	SI	Ingreso donación dinero, impresión	SI
CREACION DONACION DINERO	SI	SI	SI	Donación dinero	SI
DONACION ARTICULO	SI	SI	SI	Ingreso donación dinero, impresión	SI
CREACIÓN DONACIÓN ARTICULO	SI	SI	SI	Donación dinero	SI
DIARIO	SI	SI	No	No	SI
INVENTARIO	SI	SI	SI	Ingreso articulo, impresión	SI
CREACIÓN ARTICULO	SI	SI	SI	Inventario	SI
PAGO	SI	SI	SI	Ingreso pago, impresión	SI
CREACIÓN PAGO	SI	SI	SI	Pago	SI

Tabla 25 Prueba alfa ADMINISTRADOR 2

Fuente: Fundación FUDIS

Elaborado: Carlos Guamán y Jorge Recalde

Formulario	Controles de ingreso de texto	Validación	Mensajes de aviso	Conexión a otros formularios	CRUD
LOGIN	SI	SI	SI	Principal	No
PRINCIPAL	No	No	SI	Cama, donador, habitación, lugar, paciente, tipo de cama, donación, diario, inventario, pago	No
CAMA	SI	SI	SI	Ingreso cama, impresión	D,U
INGRESO CAMA	SI	SI	SI	cama	C,R,U,D
DONADOR	SI	SI	SI	Ingreso donador, impresión	D,U
INGRESO DONADOR	SI	SI	SI	Donador	C,R,U,D
HABITACION	SI	SI	SI	Ingreso habitación, impresión	D,U
CREAR HABITACIÓN	SI	SI	SI	Habitación	C,R,U,D
LUGAR	SI	SI	SI	Ingreso lugar, impresión	D,U
CREAR LUGAR	SI	SI	SI	Lugar	C,R,U,D
PACIENTE	SI	SI	SI	Ingreso lugar, impresión	D,U
CREAR PACIENTE	SI	SI	SI	Paciente	C,R,U,D
TIPO DE CAMA	SI	SI	SI	Ingreso tipo de cama, impresión	D,U
INGRESO TIPO DE CAMA	SI	SI	SI	Tipo de cama	C,R,U,D
ACOMPAÑANTE	SI	SI	SI	Ingreso acompañante, impresión	D,U
CREAR ACOMPAÑANTE	SI	SI	SI	Acompañante	C,R,U,D
DONACION DINERO	SI	SI	SI	Ingreso donación dinero, impresión	D,U
CREACION	SI	SI	SI	Donación	C,R,U,D

DONACION DINERO				dinero	
DONACION ARTICULO	SI	SI	SI	Ingreso donación dinero, impresión	D,U
CREACIÓN DONACIÓN ARTICULO	SI	SI	SI	Donación dinero	C,R,U,D
DIARIO	SI	SI	No	No	R
INVENTARIO	SI	SI	SI	Ingreso articulo, impresión	D,U
CREACIÓN ARTICULO	SI	SI	SI	Inventario	C,R,U,D
PAGO	SI	SI	SI	Ingreso pago, impresión	D,U
CREACIÓN PAGO	SI	SI	SI	Pago	C,R,U,D

Tabla 26 Prueba alfa TECNICO

Fuente: Fundación FUDIS
Elaborado: Carlos Guamán y Jorge Recalde

Formulario	**Controles de ingreso de texto**	**Validación**	**Mensajes de aviso**	**Conexión a otros formularios**	**CRUD**
LOGIN	SI	SI	SI	Principal	No
PRINCIPAL	No	No	SI	Cama, donador, habitación, lugar, paciente, tipo de cama, donación, diario, inventario, pago	No
DONADOR	SI	SI	SI	Ingreso donador, impresión	D,U
INGRESO DONADOR	SI	SI	SI	Donador	C,R,U,D
CREAR PACIENTE	SI	SI	SI	Paciente	C,R,U,D
ACOMPAÑANTE	SI	SI	SI	Ingreso acompañante, impresión	D,U
CREAR	SI	SI	SI	Acompañante	C,R,U,D

ACOMPAÑANTE					
DONACION DINERO	SI	SI	SI	Ingreso donación dinero, impresión	D,U
CREACION DONACION DINERO	SI	SI	SI	Donación dinero	C,R,U,D
DONACION ARTICULO	SI	SI	SI	Ingreso donación dinero, impresión	D,U
CREACIÓN DONACIÓN ARTICULO	SI	SI	SI	Donación dinero	C,R,U,D
DIARIO	SI	SI	No	No	R
INVENTARIO	SI	SI	SI	Ingreso articulo, impresión	D,U
CREACIÓN ARTICULO	SI	SI	SI	Inventario	C,R,U,D
PAGO	SI	SI	SI	Ingreso pago, impresión	D,U
CREACIÓN PAGO	SI	SI	SI	Pago	C,R,U,D

Tabla 27 Prueba alfa USUARIO

Fuente: Fundación FUDIS
Elaborado: Carlos Guamán y Jorge Recalde

ANEXO 2: PRUEBAS BETA

ACTOR INVOLUCRADO	FORMULARIO	DETALLE
Administrador 1	LOGIN	No muestra a detalle que están mal ingresado, usuario o contraseña
	PRINCIPAL	No esta en funcionamiento, no se puede visualizar nada
	CAMA	No exsite una ayuda para la búsqueda
	DONADOR	No exsite una ayuda para la búsqueda
	HABITACION	No exsite una ayuda para la búsqueda
	LUGAR	No exsite una ayuda para la búsqueda
	PACIENTE	No exsite una ayuda para la búsqueda
	TIPO DE CAMA	No exsite una ayuda para la búsqueda
	USUARIO	Esta bien
	ACOMPAÑANTE	Esta bien
	DONACION DINERO	Esta bien
	DONACION ARTICULO	No exsite una ayuda para la búsqueda
	DIARIO	La selección de fechas no es asignado automáticamente
	INVENTARIO	No exsite una ayuda para la búsqueda
	PAGO	El ingreso o devolución de pago no esta detallado claramente
Administrador 2	LOGIN	Esta bien
	PRINCIPAL	Esta bien
	CAMA	No funciona correctamenete no se ve ningún dato
	DONADOR	Esta bien
	HABITACION	Esta bien
	LUGAR	Esta bien
	PACIENTE	Hay muchos datos que llenar en este formulario
	TIPO DE CAMA	Esta bien
	USUARIO	No existe una explicación de porque se utiliza

		categorias
	ACOMPAÑANTE	Esta bien
	DONACION DINERO	Esta bien
	DONACION ARTICULO	Esta bien
	DIARIO	EL detalle del diario tiene varios datos
	INVENTARIO	La visualizacion esta correcta
	PAGO	No hay un detalle claro de que entra y que sale
Tecnico	LOGIN	Esta bien
	PRINCIPAL	Esta bien
	CAMA	No funciona este formulario
	DONADOR	Esta bien
	HABITACION	Esta bien
	LUGAR	Esta bien
	PACIENTE	Esta bien
	TIPO DE CAMA	Esta bien
	USUARIO	No puedo ver a este formulario
	ACOMPAÑANTE	Esta bien
	DONACION DINERO	Esta bien
	DONACION ARTICULO	Esta bien
	DIARIO	La elección de datos es por día o por semana
	INVENTARIO	Esta bien
	PAGO	No hay detalle de ingreso o salida
Usuario	LOGIN	Esta bien
	PRINCIPAL	Esta bien
	CAMA	No puedo ver este campo
	DONADOR	Esta bien
	HABITACION	No puedo ver este campo
	LUGAR	No puedo ver este campo
	PACIENTE	Esta bien
	TIPO DE CAMA	No puedo ver este campo
	USUARIO	No puedo ver este campo
	ACOMPAÑANTE	Cuando se ingresa no debería haber un paciente asociado
	DONACION DINERO	Esta bien

	DONACION ARTICULO	Esta bien
	DIARIO	Cuando se escoje la fecha muestra muchos detalles de ese movimiento
	INVENTARIO	Tiene el detalle de todas las bodegas
	PAGO	El análisis del ingreso o devolución tiene algunas falencias al momento del ingreso y registro

Tabla 28 Pruebas beta Todos los Actores

Fuente: Fundación FUDIS
Elaborado: Carlos Guamán y Jorge Recalde

ANEXO 3: CÓDIGO FUENTE DOCUMENTADO

BASE DE DATOS

CREACION DE TABLAS

```
/*==============================================================
*/
/* DBMS name:      Microsoft SQL Server 2008                    */
/* Created on:     ...                                          */
/*==============================================================
*/

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
   where r.fkeyid = object_id('ARTICULO') and o.name =
'FK_ARTICULO_TIENE_LUGAR')
alter table ARTICULO
   drop constraint FK_ARTICULO_TIENE_LUGAR
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
   where r.fkeyid = object_id('CAMA') and o.name =
'FK_CAMA_CONTIENE_HABITACI')
alter table CAMA
   drop constraint FK_CAMA_CONTIENE_HABITACI
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
   where r.fkeyid = object_id('CAMA') and o.name =
'FK_CAMA_ES_UNA_TIPOCAMA')
alter table CAMA
   drop constraint FK_CAMA_ES_UNA_TIPOCAMA
go
```

```
if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
  where r.fkeyid = object_id('DONACION') and o.name =
'FK_DONACION_ES_TIPODONA')
alter table DONACION
  drop constraint FK_DONACION_ES_TIPODONA
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
  where r.fkeyid = object_id('DONACION') and o.name =
'FK_DONACION_HACE_DONADOR')
alter table DONACION
  drop constraint FK_DONACION_HACE_DONADOR
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
  where r.fkeyid = object_id('DONACION') and o.name =
'FK_DONACION_PUEDE_SER_TRANSACC')
alter table DONACION
  drop constraint FK_DONACION_PUEDE_SER_TRANSACC
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
  where r.fkeyid = object_id('PUEDE_TENER') and o.name =
'FK_PUEDE_TE_PUEDE_TEN_DONACION')
alter table PUEDE_TENER
  drop constraint FK_PUEDE_TE_PUEDE_TEN_DONACION
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
  where r.fkeyid = object_id('PUEDE_TENER') and o.name =
'FK_PUEDE_TE_PUEDE_TEN_ARTICULO')
alter table PUEDE_TENER
  drop constraint FK_PUEDE_TE_PUEDE_TEN_ARTICULO
go

if exists (select 1
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
  where r.fkeyid = object_id('TRANSACCION') and o.name =
```

```
'FK_TRANSACC_INVOLUCRA_ACOMPANA')
alter table TRANSACCION
   drop constraint FK_TRANSACC_INVOLUCRA_ACOMPANA
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
   where r.fkeyid = object_id('TRANSACCION') and o.name =
'FK_TRANSACC_POSEE_TIPOTRAN')
alter table TRANSACCION
   drop constraint FK_TRANSACC_POSEE_TIPOTRAN
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
   where r.fkeyid = object_id('TRANSACCION') and o.name =
'FK_TRANSACC_PUEDE_INV_CAMA')
alter table TRANSACCION
   drop constraint FK_TRANSACC_PUEDE_INV_CAMA
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
   where r.fkeyid = object_id('TRANSACCION') and o.name =
'FK_TRANSACC_PUEDE_POS_CAMA')
alter table TRANSACCION
   drop constraint FK_TRANSACC_PUEDE_POS_CAMA
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
   where r.fkeyid = object_id('TRANSACCION') and o.name =
'FK_TRANSACC_PUEDE_SER_TRANSACC')
alter table TRANSACCION
   drop constraint FK_TRANSACC_PUEDE_SER_TRANSACC
go

if exists (select 1
   from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type
= 'F')
   where r.fkeyid = object_id('TRANSACCION') and o.name =
'FK_TRANSACC_REALIZA_PACIENTE')
alter table TRANSACCION
   drop constraint FK_TRANSACC_REALIZA_PACIENTE
go
```

```
if exists (select 1
          from  sysobjects
          where  id = object_id('ACOMPANANTE')
          and   type = 'U')
  drop table ACOMPANANTE
go

if exists (select 1
          from  sysindexes
          where  id    = object_id('ARTICULO')
          and   name  = 'TIENE_FK'
          and   indid > 0
          and   indid < 255)
  drop index ARTICULO.TIENE_FK
go

if exists (select 1
          from  sysobjects
          where  id = object_id('ARTICULO')
          and   type = 'U')
  drop table ARTICULO
go

if exists (select 1
          from  sysindexes
          where  id    = object_id('CAMA')
          and   name  = 'CONTIENE_FK'
          and   indid > 0
          and   indid < 255)
  drop index CAMA.CONTIENE_FK
go

if exists (select 1
          from  sysindexes
          where  id    = object_id('CAMA')
          and   name  = 'ES_UNA_FK'
          and   indid > 0
          and   indid < 255)
  drop index CAMA.ES_UNA_FK
go

if exists (select 1
          from  sysobjects
          where  id = object_id('CAMA')
          and   type = 'U')
  drop table CAMA
go

if exists (select 1
          from  sysindexes
```

```
        where  id    = object_id('DONACION')
         and   name  = 'PUEDE_SER_FK'
         and   indid > 0
         and   indid < 255)
  drop index DONACION.PUEDE_SER_FK
go

if exists (select 1
         from  sysindexes
        where  id    = object_id('DONACION')
         and   name  = 'ES_FK'
         and   indid > 0
         and   indid < 255)
  drop index DONACION.ES_FK
go

if exists (select 1
         from  sysindexes
        where  id    = object_id('DONACION')
         and   name  = 'HACE_FK'
         and   indid > 0
         and   indid < 255)
  drop index DONACION.HACE_FK
go

if exists (select 1
         from  sysobjects
        where  id = object_id('DONACION')
         and   type = 'U')
  drop table DONACION
go

if exists (select 1
         from  sysobjects
        where  id = object_id('DONADOR')
         and   type = 'U')
  drop table DONADOR
go

if exists (select 1
         from  sysobjects
        where  id = object_id('HABITACION')
         and   type = 'U')
  drop table HABITACION
go

if exists (select 1
         from  sysobjects
        where  id = object_id('LUGAR')
         and   type = 'U')
```

```
  drop table LUGAR
go

if exists (select 1
          from  sysobjects
          where  id = object_id('PACIENTE')
          and   type = 'U')
  drop table PACIENTE
go

if exists (select 1
          from  sysindexes
          where  id    = object_id('PUEDE_TENER')
          and   name  = 'PUEDE_TENER2_FK'
          and   indid > 0
          and   indid < 255)
  drop index PUEDE_TENER.PUEDE_TENER2_FK
go

if exists (select 1
          from  sysindexes
          where  id    = object_id('PUEDE_TENER')
          and   name  = 'PUEDE_TENER_FK'
          and   indid > 0
          and   indid < 255)
  drop index PUEDE_TENER.PUEDE_TENER_FK
go

if exists (select 1
          from  sysobjects
          where  id = object_id('PUEDE_TENER')
          and   type = 'U')
  drop table PUEDE_TENER
go

if exists (select 1
          from  sysobjects
          where  id = object_id('TIPOCAMA')
          and   type = 'U')
  drop table TIPOCAMA
go

if exists (select 1
          from  sysobjects
          where  id = object_id('TIPODONACION')
          and   type = 'U')
  drop table TIPODONACION
go

if exists (select 1
```

```
        from  sysobjects
       where  id = object_id('TIPOTRANSACCION')
        and   type = 'U')
  drop table TIPOTRANSACCION
go

if exists (select 1
        from  sysindexes
       where  id    = object_id('TRANSACCION')
        and   name  = 'PUEDE_POSEER_FK'
        and   indid > 0
        and   indid < 255)
  drop index TRANSACCION.PUEDE_POSEER_FK
go

if exists (select 1
        from  sysindexes
       where  id    = object_id('TRANSACCION')
        and   name  = 'INVOLUCRA_FK'
        and   indid > 0
        and   indid < 255)
  drop index TRANSACCION.INVOLUCRA_FK
go

if exists (select 1
        from  sysindexes
       where  id    = object_id('TRANSACCION')
        and   name  = 'PUEDE_SER_UNA_FK'
        and   indid > 0
        and   indid < 255)
  drop index TRANSACCION.PUEDE_SER_UNA_FK
go

if exists (select 1
        from  sysindexes
       where  id    = object_id('TRANSACCION')
        and   name  = 'PUEDE_INVOLUCRAR_FK'
        and   indid > 0
        and   indid < 255)
  drop index TRANSACCION.PUEDE_INVOLUCRAR_FK
go

if exists (select 1
        from  sysindexes
       where  id    = object_id('TRANSACCION')
        and   name  = 'POSEE_FK'
        and   indid > 0
        and   indid < 255)
  drop index TRANSACCION.POSEE_FK
go
```

```
if exists (select 1
          from  sysindexes
          where  id    = object_id('TRANSACCION')
          and   name  = 'REALIZA_FK'
          and   indid > 0
          and   indid < 255)
  drop index TRANSACCION.REALIZA_FK
go

if exists (select 1
          from  sysobjects
          where  id = object_id('TRANSACCION')
          and   type = 'U')
  drop table TRANSACCION
go

if exists (select 1
          from  sysobjects
          where  id = object_id('USUARIO')
          and   type = 'U')
  drop table USUARIO
go

/*==============================================================
*/
/* Table: ACOMPANANTE                                           */
/*==============================================================
*/
create table ACOMPANANTE (
  IDACOMPANANTE       bigint            not null,
  CIACOMPANANTE       varchar(15)       null,
  NOMBREACOMPANANTE   varchar(50)       not null,
  APELLIDOACOMPANANTE  varchar(50)      not null,
  constraint PK_ACOMPANANTE primary key nonclustered
(IDACOMPANANTE)
)
go

/*==============================================================
*/
/* Table: ARTICULO                                              */
/*==============================================================
*/
create table ARTICULO (
  IDARTICULO          bigint            not null,
  IDLUGAR             smallint          not null,
  NOMBREARTICULO      varchar(50)       not null,
  CANTIDADARTICULO    bigint            not null,
  DESCRIPCIONARTICULO  varchar(100)     null,
```

```
  constraint PK_ARTICULO primary key nonclustered (IDARTICULO)
)
go

/*==============================================================
*/
/* Index: TIENE_FK                                */
/*==============================================================
*/
create index TIENE_FK on ARTICULO (
IDLUGAR ASC
)
go

/*==============================================================
*/
/* Table: CAMA                                    */
/*==============================================================
*/
create table CAMA (
  IDCAMA            smallint          not null,
  IDHABITACION      smallint          not null,
  IDTIPOCAMA        smallint          not null,
  CODIGOCAMA        varchar(15)       null,
  DESCRIPCIONCAMA   varchar(100)      null,
  DISPONIBILIDADCAMA  bit             null,
  constraint PK_CAMA primary key nonclustered (IDCAMA)
)
go

/*==============================================================
*/
/* Index: ES_UNA_FK                               */
/*==============================================================
*/
create index ES_UNA_FK on CAMA (
IDTIPOCAMA ASC
)
go

/*==============================================================
*/
/* Index: CONTIENE_FK                             */
/*==============================================================
*/
create index CONTIENE_FK on CAMA (
IDHABITACION ASC
)
go
```

```
/*==============================================================
*/
/* Table: DONACION                                */
/*==============================================================
*/
create table DONACION (
  IDDONACION          bigint              not null,
  IDTIPODONACION      smallint            not null,
  IDTRANSACCION       bigint              null,
  IDDONADOR           bigint              not null,
  BENEFICIADODONACION  varchar(50)        not null,
  DESCRIPCIONDONACION  varchar(100)       null,
  FECHADONACION       datetime            null,
  constraint PK_DONACION primary key nonclustered (IDDONACION)
)
go

/*==============================================================
*/
/* Index: HACE_FK                                 */
/*==============================================================
*/
create index HACE_FK on DONACION (
IDDONADOR ASC
)
go

/*==============================================================
*/
/* Index: ES_FK                                   */
/*==============================================================
*/
create index ES_FK on DONACION (
IDTIPODONACION ASC
)
go

/*==============================================================
*/
/* Index: PUEDE_SER_FK                            */
/*==============================================================
*/
create index PUEDE_SER_FK on DONACION (
IDTRANSACCION ASC
)
go

/*==============================================================
*/
/* Table: DONADOR                                 */
```

```
/*==============================================================
*/
create table DONADOR (
  IDDONADOR          bigint           not null,
  CIDONADOR          varchar(15)      null,
  NOMBREDONADOR      varchar(50)      not null,
  APELLIDODONADOR    varchar(50)      not null,
  TELEFONODONADOR    varchar(150)     null,
  EMPRESAREPRESENTANTEDONADOR varchar(50)  null,
  EMAILDONADOR       varchar(50)      null,
  OBSEVACIONESDONADOR varchar(100)    null,
  constraint PK_DONADOR primary key nonclustered (IDDONADOR)
)
go

/*==============================================================
*/
/* Table: HABITACION                                    */
/*==============================================================
*/
create table HABITACION (
  IDHABITACION       smallint         not null,
  PISOHABITACION     varchar(15)      not null,
  CODIGOHABITACION   varchar(15)      null,
  DESCRIPCIONHABITACION varchar(100)  null,
  constraint PK_HABITACION primary key nonclustered (IDHABITACION)
)
go

/*==============================================================
*/
/* Table: LUGAR                                         */
/*==============================================================
*/
create table LUGAR (
  IDLUGAR            smallint         not null,
  NOMBRELUGAR        varchar(50)      not null,
  DESCRIPCIONLUGAR   varchar(100)     null,
  constraint PK_LUGAR primary key nonclustered (IDLUGAR)
)
go

/*==============================================================
*/
/* Table: PACIENTE                                      */
/*==============================================================
*/
create table PACIENTE (
  IDPACIENTE         bigint           not null,
  CIPACIENTE         varchar(15)      null,
```

```
  NOMBREPACIENTE      varchar(50)       not null,
  APELLIDOPACIENTE    varchar(50)       not null,
  TELEFONOPACIENTE    varchar(150)      null,
  HISTORIACLINICAPACIENTE varchar(25)       null,
  NACIONALIDADPACIENTE varchar(50)       null,
  PROVINCIAPACIENTE   varchar(50)       null,
  CIUDADPACIENTE      varchar(50)       null,
  FECHANACIMIENTOPACIENTE datetime          not null,
  DOMICILIOPACIENTE   varchar(100)      null,
  HOSPITALTRATANTEPACIENTE varchar(50)       null,
  DIAGNOSTICOPACIENTE  varchar(50)       null,
  TRATAMIENTOPACIENTE  varchar(100)      null,
  OBSERVACIONESPACIENTE varchar(100)      null,
  constraint PK_PACIENTE primary key nonclustered (IDPACIENTE)
)
go

/*==============================================================
*/
/* Table: PUEDE_TENER                                  */
/*==============================================================
*/
create table PUEDE_TENER (
  IDDONACION        bigint            not null,
  IDARTICULO        bigint            not null,
  CANTIDADDONADA    int               not null,
  constraint PK_PUEDE_TENER primary key (IDDONACION, IDARTICULO)
)
go

/*==============================================================
*/
/* Index: PUEDE_TENER_FK                               */
/*==============================================================
*/
create index PUEDE_TENER_FK on PUEDE_TENER (
IDDONACION ASC
)
go

/*==============================================================
*/
/* Index: PUEDE_TENER2_FK                              */
/*==============================================================
*/
create index PUEDE_TENER2_FK on PUEDE_TENER (
IDARTICULO ASC
)
go
```

```
/*==============================================================
*/
/* Table: TIPOCAMA                                */
/*==============================================================
*/
create table TIPOCAMA (
  IDTIPOCAMA          smallint          not null,
  NOMBRETIPOCAMA      varchar(50)       not null,
  DESCRIPCIONTIPOCAMA  varchar(100)      null,
  constraint PK_TIPOCAMA primary key nonclustered (IDTIPOCAMA)
)
go

/*==============================================================
*/
/* Table: TIPODONACION                            */
/*==============================================================
*/
create table TIPODONACION (
  IDTIPODONACION      smallint          not null,
  NOMBRETIPODONACION   varchar(50)       not null,
  DESCRIPCIONTIPODONACION varchar(100)   null,
  constraint PK_TIPODONACION primary key nonclustered (IDTIPODONACION)
)
go

/*==============================================================
*/
/* Table: TIPOTRANSACCION                         */
/*==============================================================
*/
create table TIPOTRANSACCION (
  IDTIPOTRANSACCION    smallint          not null,
  NOMBRETIPOTRANSACCION varchar(50)      not null,
  DESCRIPCIONTIPOTRANSACCION varchar(100)  null,
  constraint PK_TIPOTRANSACCION primary key nonclustered
(IDTIPOTRANSACCION)
)
go

/*==============================================================
*/
/* Table: TRANSACCION                             */
/*==============================================================
*/
create table TRANSACCION (
  IDTRANSACCION       bigint            not null,
  TRA_IDTRANSACCION   bigint            null,
  IDPACIENTE          bigint            null,
  IDCAMA              smallint          null,
```

```
  IDACOMPANANTE       bigint          null,
  CAM_IDCAMA          smallint        null,
  IDTIPOTRANSACCION   smallint        not null,
  COSTODIARIOTRANSACCION money        null,
  DIASCANCELADOS_DEVUELTOSTRANSACCION int          null,
  MONTOTRANSACCION    money           null,
  FECHATRANSACCION    datetime        null,
  FECHAINICIOTRANSACCION datetime     null,
  FECHAFINTRANSACCION  datetime       null,
  constraint PK_TRANSACCION primary key nonclustered (IDTRANSACCION)
)
go

/*==============================================================
*/
/* Index: REALIZA_FK                                            */
/*==============================================================
*/
create index REALIZA_FK on TRANSACCION (
IDPACIENTE ASC
)
go

/*==============================================================
*/
/* Index: POSEE_FK                                              */
/*==============================================================
*/
create index POSEE_FK on TRANSACCION (
IDTIPOTRANSACCION ASC
)
go

/*==============================================================
*/
/* Index: PUEDE_INVOLUCRAR_FK                                   */
/*==============================================================
*/
create index PUEDE_INVOLUCRAR_FK on TRANSACCION (
CAM_IDCAMA ASC
)
go

/*==============================================================
*/
/* Index: PUEDE_SER_UNA_FK                                      */
/*==============================================================
*/
create index PUEDE_SER_UNA_FK on TRANSACCION (
TRA_IDTRANSACCION ASC
```

```
)
go

/*==============================================================
*/
/* Index: INVOLUCRA_FK                                  */
/*==============================================================
*/
create index INVOLUCRA_FK on TRANSACCION (
IDACOMPANANTE ASC
)
go

/*==============================================================
*/
/* Index: PUEDE_POSEER_FK                               */
/*==============================================================
*/
create index PUEDE_POSEER_FK on TRANSACCION (
IDCAMA ASC
)
go

/*==============================================================
*/
/* Table: USUARIO                                       */
/*==============================================================
*/
create table USUARIO (
  IDUSUARIO          smallint          not null,
  NOMBREUSUARIO      varchar(50)       not null,
  CLAVEUSUARIO       varchar(50)       not null,
  TIPOUSUARIO        varchar(50)       not null,
  constraint PK_USUARIO primary key nonclustered (IDUSUARIO)
)
go

alter table ARTICULO
  add constraint FK_ARTICULO_TIENE_LUGAR foreign key (IDLUGAR)
    references LUGAR (IDLUGAR)
      on update cascade on delete cascade
go

alter table CAMA
  add constraint FK_CAMA_CONTIENE_HABITACI foreign key (IDHABITACION)
    references HABITACION (IDHABITACION)
      on update cascade on delete cascade
go

alter table CAMA
```

```
  add constraint FK_CAMA_ES_UNA_TIPOCAMA foreign key (IDTIPOCAMA)
    references TIPOCAMA (IDTIPOCAMA)
      on update cascade on delete cascade
go

alter table DONACION
  add constraint FK_DONACION_ES_TIPODONA foreign key
(IDTIPODONACION)
    references TIPODONACION (IDTIPODONACION)
      on update cascade on delete cascade
go

alter table DONACION
  add constraint FK_DONACION_HACE_DONADOR foreign key (IDDONADOR)
    references DONADOR (IDDONADOR)
      on update cascade on delete cascade
go

alter table DONACION
  add constraint FK_DONACION_PUEDE_SER_TRANSACC foreign key
(IDTRANSACCION)
    references TRANSACCION (IDTRANSACCION)
      on update cascade on delete cascade
go

alter table PUEDE_TENER
  add constraint FK_PUEDE_TE_PUEDE_TEN_DONACION foreign key
(IDDONACION)
    references DONACION (IDDONACION)
      on update cascade on delete cascade
go

alter table PUEDE_TENER
  add constraint FK_PUEDE_TE_PUEDE_TEN_ARTICULO foreign key
(IDARTICULO)
    references ARTICULO (IDARTICULO)
      on update cascade on delete cascade
go

alter table TRANSACCION
  add constraint FK_TRANSACC_INVOLUCRA_ACOMPANA foreign key
(IDACOMPANANTE)
    references ACOMPANANTE (IDACOMPANANTE)
      on update cascade on delete cascade
go

alter table TRANSACCION
  add constraint FK_TRANSACC_POSEE_TIPOTRAN foreign key
(IDTIPOTRANSACCION)
    references TIPOTRANSACCION (IDTIPOTRANSACCION)
```

```
      on update cascade on delete cascade
go

alter table TRANSACCION
  add constraint FK_TRANSACC_PUEDE_INV_CAMA foreign key
(CAM_IDCAMA)
    references CAMA (IDCAMA)
go

alter table TRANSACCION
  add constraint FK_TRANSACC_PUEDE_POS_CAMA foreign key (IDCAMA)
    references CAMA (IDCAMA)
      on update cascade on delete cascade
go

alter table TRANSACCION
  add constraint FK_TRANSACC_PUEDE_SER_TRANSACC foreign key
(TRA_IDTRANSACCION)
    references TRANSACCION (IDTRANSACCION)
go

alter table TRANSACCION
  add constraint FK_TRANSACC_REALIZA_PACIENTE foreign key (IDPACIENTE)
    references PACIENTE (IDPACIENTE)
      on update cascade on delete cascade
go
```

CREACION DE PROCEDIMIENTOS ALMACENADOS

```
USE [FUNDACIÓN FUDIS DISERTACIÓN DE GRADO]
GO
/****** Object:  StoredProcedure [dbo].[BUSCARPACIENTE]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARPACIENTE]

   @CI varchar(15),
   @NOMBRE varchar(50),
   @APELLIDO varchar(50),
   @NACIONALIDAD varchar(50),
   @PROVINCIA varchar(50),
   @CIUDAD varchar(50),
   @HOSPITALTRATANTE varchar(50),
   @DIAGNOSTICO varchar(50),
   @TRATAMIENTO varchar(100),
   @EDADINI INT,
```

```
   @EDADFIN INT

AS
BEGIN

        SELECT  IDPACIENTE ,
        CIPACIENTE ,
        NOMBREPACIENTE ,
        APELLIDOPACIENTE ,
        TELEFONOPACIENTE ,
        HISTORIACLINICAPACIENTE ,
        NACIONALIDADPACIENTE ,
        PROVINCIAPACIENTE ,
   CIUDADPACIENTE,
        (((365* year(getdate()))-(365*(year(FECHANACIMIENTOPACIENTE))))+
(month(getdate())-month(FECHANACIMIENTOPACIENTE))*30 +(day(getdate())
- day(FECHANACIMIENTOPACIENTE)))/365 AS EDADPACIENTE ,
        DOMICILIOPACIENTE ,
        HOSPITALTRATANTEPACIENTE ,
        DIAGNOSTICOPACIENTE ,
        TRATAMIENTOPACIENTE
        FROM dbo.PACIENTE
        WHERE (CIPACIENTE LIKE '%'+ @CI +'%' AND
        NOMBREPACIENTE LIKE '%'+@NOMBRE+'%' AND
        APELLIDOPACIENTE LIKE '%'+@APELLIDO+'%' AND
        NACIONALIDADPACIENTE LIKE '%'+@NACIONALIDAD+'%' AND
        PROVINCIAPACIENTE LIKE '%'+@PROVINCIA+'%' AND
        HOSPITALTRATANTEPACIENTE LIKE '%'+@HOSPITALTRATANTE+'%'
AND
        CIUDADPACIENTE LIKE '%'+@CIUDAD+'%' AND
        DIAGNOSTICOPACIENTE LIKE '%'+@DIAGNOSTICO+'%' AND
        TRATAMIENTOPACIENTE LIKE '%'+@TRATAMIENTO+'%' AND
        IDPACIENTE <> -1 AND
        ((((365* year(getdate()))-(365*(year(FECHANACIMIENTOPACIENTE))))+
(month(getdate())-month(FECHANACIMIENTOPACIENTE))*30 +(day(getdate())
- day(FECHANACIMIENTOPACIENTE)))/365) BETWEEN @EDADINI AND
@EDADFIN
        )

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARLUGAR]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARLUGAR]
   -- Add the parameters for the stored procedure here
   @NOMBRELUGAR varchar(50),
```

```
    @DESCRIPCIONLUGAR varchar(100)

AS
BEGIN

        SELECT   IDLUGAR,
         NOMBRELUGAR,
        DESCRIPCIONLUGAR

        FROM  dbo.LUGAR
        WHERE (NOMBRELUGAR LIKE '%' +@NOMBRELUGAR +'%' AND
        DESCRIPCIONLUGAR LIKE '%' +@DESCRIPCIONLUGAR+'%' AND
        IDLUGAR <> -1)

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARHABITACION]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARHABITACION]
   -- Add the parameters for the stored procedure here
   @PISOHABITACION varchar(15),
   @CODIGOHABITACION varchar(15),
   @DESCRIPCIONHABITACION varchar(100)

AS
BEGIN

        SELECT  IDHABITACION ,
        PISOHABITACION ,
        CODIGOHABITACION ,
        DESCRIPCIONHABITACION

        FROM  dbo.HABITACION
        WHERE (PISOHABITACION LIKE '%' +@PISOHABITACION +'%' AND
        CODIGOHABITACION LIKE '%' +@CODIGOHABITACION +'%' AND
        DESCRIPCIONHABITACION LIKE '%' +@DESCRIPCIONHABITACION +'%'
        AND IDHABITACION <> -1)

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARDONADOR]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [dbo].[BUSCARDONADOR]
   -- Add the parameters for the stored procedure here
   @CIDONADOR varchar(15),
   @NOMBREDONADOR varchar(50),
   @APELLIDODONADOR varchar(50),
   @EMPRESAREPRESENTANTEDONADOR varchar(50),
   @DESCRIPCIONDONADOR VARCHAR(100)


AS
BEGIN

        SELECT IDDONADOR,
        CIDONADOR ,
        NOMBREDONADOR ,
        APELLIDODONADOR ,
        TELEFONODONADOR ,
        EMPRESAREPRESENTANTEDONADOR ,
        EMAILDONADOR,
        OBSEVACIONESDONADOR
        FROM dbo.DONADOR
        WHERE (CIDONADOR LIKE '%'+@CIDONADOR+'%' AND
        NOMBREDONADOR LIKE '%'+@NOMBREDONADOR+'%' AND
        APELLIDODONADOR LIKE '%'+@APELLIDODONADOR+'%' AND
        EMPRESAREPRESENTANTEDONADOR LIKE
'%'+@EMPRESAREPRESENTANTEDONADOR+'%' AND
        OBSEVACIONESDONADOR LIKE '%'+@DESCRIPCIONDONADOR +'%'
AND
        IDDONADOR<>-1)

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARACOMPANANTE]    Script
Date: 06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARACOMPANANTE]
   -- Add the parameters for the stored procedure here
   @CI VARCHAR (15),
   @NOMBRE varchar(50),
   @APELLIDO varchar(50)

AS
BEGIN

        SELECT   IDACOMPANANTE ,
        CIACOMPANANTE ,
        NOMBREACOMPANANTE ,
```

```
        APELLIDOACOMPANANTE

        FROM  dbo.ACOMPANANTE
        WHERE CIACOMPANANTE LIKE '%' + @CI + '%'
        AND NOMBREACOMPANANTE LIKE '%' + @NOMBRE + '%'
        AND APELLIDOACOMPANANTE LIKE '%' + @APELLIDO + '%'
        AND IDACOMPANANTE <> -1

END
GO
/****** Object:  StoredProcedure [dbo].[DESCRIPCIONLUGAR]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DESCRIPCIONLUGAR]
(
        @IDLUGAR smallint
)
AS
        SET NOCOUNT ON;
SELECT DESCRIPCIONLUGAR FROM LUGAR WHERE IDLUGAR=@IDLUGAR
GO
/****** Object:  StoredProcedure [dbo].[BUSCARTIPOCAMA]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARTIPOCAMA]
   -- Add the parameters for the stored procedure here
   @NOMBRETIPOCAMA varchar(50),
   @DESCRIPCIONTIPOCAMA varchar(100)

AS
BEGIN

        SELECT   IDTIPOCAMA,
        NOMBRETIPOCAMA,
        DESCRIPCIONTIPOCAMA

        FROM  dbo.TIPOCAMA
        WHERE (NOMBRETIPOCAMA LIKE '%' +@NOMBRETIPOCAMA +'%' AND
        DESCRIPCIONTIPOCAMA LIKE '%' +@DESCRIPCIONTIPOCAMA AND
        IDTIPOCAMA <> -1)

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARPACIENTEDISPONIBLE]
```

```
Script Date: 06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARPACIENTEDISPONIBLE]

   @CI varchar(15),
   @NOMBRE varchar(50),
   @APELLIDO varchar(50),
   @NACIONALIDAD varchar(50),
   @PROVINCIA varchar(50),
   @CIUDAD varchar(50),
   @HOSPITALTRATANTE varchar(50),
   @DIAGNOSTICO varchar(50),
   @TRATAMIENTO varchar(100),
   @EDADINI INT,
   @EDADFIN INT

AS
BEGIN

         SELECT  IDPACIENTE ,
         CIPACIENTE ,
         NOMBREPACIENTE ,
         APELLIDOPACIENTE ,
         TELEFONOPACIENTE ,
         HISTORIACLINICAPACIENTE ,
         NACIONALIDADPACIENTE ,
         PROVINCIAPACIENTE ,
   CIUDADPACIENTE,
         (((365* year(getdate()))-(365*(year(FECHANACIMIENTOPACIENTE))))+
(month(getdate())-month(FECHANACIMIENTOPACIENTE))*30 +(day(getdate())
- day(FECHANACIMIENTOPACIENTE)))/365 AS EDADPACIENTE ,
         DOMICILIOPACIENTE ,
         HOSPITALTRATANTEPACIENTE ,
         DIAGNOSTICOPACIENTE ,
         TRATAMIENTOPACIENTE
         FROM dbo.PACIENTE
         WHERE (CIPACIENTE LIKE '%'+ @CI +'%' AND
         NOMBREPACIENTE LIKE '%'+@NOMBRE+'%' AND
         APELLIDOPACIENTE LIKE '%'+@APELLIDO+'%' AND
         NACIONALIDADPACIENTE LIKE '%'+@NACIONALIDAD+'%' AND
         PROVINCIAPACIENTE LIKE '%'+@PROVINCIA+'%' AND
         HOSPITALTRATANTEPACIENTE LIKE '%'+@HOSPITALTRATANTE+'%'
AND
         CIUDADPACIENTE LIKE '%'+@CIUDAD+'%' AND
         DIAGNOSTICOPACIENTE LIKE '%'+@DIAGNOSTICO+'%' AND
         TRATAMIENTOPACIENTE LIKE '%'+@TRATAMIENTO+'%' AND
         IDPACIENTE NOT IN (SELECT IDPACIENTE FROM TRANSACCION WHERE
```

```
(GETDATE() BETWEEN FECHAINICIOTRANSACCION AND
FECHAFINTRANSACCION) ) AND
        IDPACIENTE <> -1 AND
        ((((365* year(getdate()))-(365*(year(FECHANACIMIENTOPACIENTE))))+
(month(getdate())-month(FECHANACIMIENTOPACIENTE))*30 +(day(getdate())
- day(FECHANACIMIENTOPACIENTE)))/365) BETWEEN @EDADINI AND
@EDADFIN)

END
GO
/****** Object:  StoredProcedure [dbo].[ACTUALIZARCAMAS]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[ACTUALIZARCAMAS]

AS
BEGIN

        update dbo.CAMA
        set DISPONIBILIDADCAMA='FALSE'
        where IDCAMA = (SELECT IDCAMA FROM dbo.TRANSACCION
        where IDTRANSACCION NOT IN (SELECT TRA_IDTRANSACCION FROM
TRANSACCION)
        and IDTIPOTRANSACCION = 1 and (GETDATE() between
FECHAINICIOTRANSACCION and FECHAFINTRANSACCION))

        update dbo.CAMA
        set DISPONIBILIDADCAMA='TRUE'
        where IDCAMA NOT IN (SELECT IDCAMA FROM dbo.TRANSACCION
        where IDTRANSACCION NOT IN (SELECT TRA_IDTRANSACCION FROM
TRANSACCION)
        and IDTIPOTRANSACCION = 1 and (GETDATE() between
FECHAINICIOTRANSACCION and FECHAFINTRANSACCION))


END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARCAMADISPONIBLE]    Script
Date: 06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARCAMADISPONIBLE]
   -- Add the parameters for the stored procedure here
   @CODIGO varchar(15),
   @DESCRIPCION varchar(100),
```

```
    @NOMBRETIPOCAMA varchar(50),
    @PISOHABITACION varchar(15),
    @CODIGOHABITACION varchar(15),
    @IDCAMA smallint

AS
BEGIN

        SELECT  IDCAMA ,
        CODIGOCAMA ,
        DESCRIPCIONCAMA ,
        DISPONIBILIDADCAMA ,
        NOMBRETIPOCAMA ,
        PISOHABITACION,
        CODIGOHABITACION,
        DIASDISPONIBLESCAMA
        FROM  dbo.VISTACAMA
        WHERE (CODIGOCAMA LIKE '%' +@CODIGO +'%' AND
        DESCRIPCIONCAMA LIKE '%' +@DESCRIPCION +'%' AND
        NOMBRETIPOCAMA LIKE '%' +@NOMBRETIPOCAMA +'%' AND
        PISOHABITACION LIKE '%' +@PISOHABITACION +'%' AND
        CODIGOHABITACION LIKE '%' +@CODIGOHABITACION +'%' AND
        IDCAMA <> @IDCAMA AND
        DISPONIBILIDADCAMA = 'TRUE')
END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARCAMA]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARCAMA]
    -- Add the parameters for the stored procedure here
    @CODIGO varchar(15),
    @DESCRIPCION varchar(100),
    @NOMBRETIPOCAMA varchar(50),
    @PISOHABITACION varchar(15),
    @CODIGOHABITACION varchar(15)


AS
BEGIN

        SELECT  IDCAMA ,
        CODIGOCAMA ,
        DESCRIPCIONCAMA ,
        DISPONIBILIDADCAMA ,
        NOMBRETIPOCAMA ,
        PISOHABITACION,
```

```
        CODIGOHABITACION
        FROM  dbo.VISTACAMA
        WHERE (CODIGOCAMA LIKE '%' +@CODIGO +'%' AND
        DESCRIPCIONCAMA LIKE '%' +@DESCRIPCION +'%' AND
        NOMBRETIPOCAMA LIKE '%' +@NOMBRETIPOCAMA +'%' AND
        PISOHABITACION LIKE '%' +@PISOHABITACION +'%' AND
        CODIGOHABITACION LIKE '%' +@CODIGOHABITACION +'%')

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARARTICULO]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARARTICULO]
   -- Add the parameters for the stored procedure here

   @NOMBRE varchar(50),
   @LUGAR VARCHAR (50),
   @DESCRIPCION varchar(100)
AS
BEGIN

        SELECT IDARTICULO ,
        NOMBREARTICULO ,
        CANTIDADARTICULO,
        DESCRIPCIONARTICULO,
        NOMBRELUGAR
        FROM  VISTAARTICULO
        WHERE (NOMBREARTICULO LIKE '%'+@NOMBRE+'%' AND
        DESCRIPCIONARTICULO LIKE '%'+@DESCRIPCION+'%' AND
        NOMBRELUGAR LIKE '%'+@LUGAR+'%')

END
GO
/****** Object:  StoredProcedure [dbo].[DIARIO]    Script Date: 06/19/2013
23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DIARIO]
   -- Add the parameters for the stored procedure here

   @FECHA AS DATE
AS
BEGIN
```

```
        SELECT IDTRANSACCION,
        CIPACIENTE,
        NOMBREPACIENTE,
        APELLIDOPACIENTE,
        CIACOMPANANTE,
        NOMBREACOMPANANTE,
        APELLIDOACOMPANANTE,
        MONTOTRANSACCION,
        FECHATRANSACCION,
        NOMBRETIPOTRANSACCION
        FROM  dbo.VISTATRANSACCION
        WHERE (convert(DATE,FECHATRANSACCION ,126)) = @FECHA AND
        IDTRANSACCION <> -1

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARDEVOLUCIONPAGO]    Script
Date: 06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARDEVOLUCIONPAGO]
  -- Add the parameters for the stored procedure here

  @CIPACIENTE varchar (15),
        @NOMBREPACIENTE varchar(50),
        @APELLIDOPACIENTE varchar(50),
        @CIACOMPANANTE varchar(15),
        @NOMBREACOMPANANTE varchar(50),
        @APELLIDOACOMPANANTE varchar(50),
        @FECHAINICIO DATE,
        @FECHAFIN DATE
AS
BEGIN

        SELECT IDTRANSACCION,
        TRA_IDTRANSACCION,
        CIPACIENTE,
        NOMBREPACIENTE,
        APELLIDOPACIENTE,
        IDCAMA,
        CIACOMPANANTE,
        NOMBREACOMPANANTE,
        APELLIDOACOMPANANTE,
        CAM_IDCAMA,
        COSTODIARIOTRANSACCION,
        MONTOTRANSACCION,
        FECHATRANSACCION
        FROM  dbo.VISTATRANSACCION
```

```
        WHERE (CIPACIENTE LIKE'%'+@CIPACIENTE+'%' AND
        NOMBREPACIENTE LIKE'%'+@NOMBREPACIENTE+'%' AND
        APELLIDOPACIENTE LIKE'%'+@APELLIDOPACIENTE+'%' AND
        CIACOMPANANTE LIKE'%'+@CIACOMPANANTE+'%' AND
        NOMBREACOMPANANTE LIKE'%'+@NOMBREACOMPANANTE+'%' AND
        APELLIDOACOMPANANTE LIKE'%'+@APELLIDOACOMPANANTE+'%'
AND
        NOMBRETIPOTRANSACCION = 'DEVOLUCIÓN PAGO' AND
        (convert(DATE,FECHATRANSACCION ,126)) BETWEEN @FECHAINICIO
AND @FECHAFIN)

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARPAGO]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARPAGO]
  -- Add the parameters for the stored procedure here

  @CIPACIENTE varchar (15),
        @NOMBREPACIENTE varchar(50),
        @APELLIDOPACIENTE varchar(50),
        @CIACOMPANANTE varchar(15),
        @NOMBREACOMPANANTE varchar(50),
        @APELLIDOACOMPANANTE varchar(50),
        @FECHAINICIO DATE,
        @FECHAFIN DATE
AS
BEGIN

        SELECT IDTRANSACCION,
        CIPACIENTE,
        NOMBREPACIENTE,
        APELLIDOPACIENTE,
        IDCAMA,
        CIACOMPANANTE,
        NOMBREACOMPANANTE,
        APELLIDOACOMPANANTE,
        CAM_IDCAMA,
        COSTODIARIOTRANSACCION,
        MONTOTRANSACCION,
        FECHATRANSACCION,
        FECHAINICIOTRANSACCION,
        FECHAFINTRANSACCION
        FROM  dbo.VISTATRANSACCION
        WHERE (CIPACIENTE LIKE'%'+@CIPACIENTE+'%' AND
        NOMBREPACIENTE LIKE'%'+@NOMBREPACIENTE+'%' AND
```

```
        APELLIDOPACIENTE LIKE'%'+@APELLIDOPACIENTE+'%' AND
        CIACOMPANANTE LIKE'%'+@CIACOMPANANTE+'%' AND
        NOMBREACOMPANANTE LIKE'%'+@NOMBREACOMPANANTE+'%' AND
        APELLIDOACOMPANANTE LIKE'%'+@APELLIDOACOMPANANTE+'%'
AND
        NOMBRETIPOTRANSACCION = 'PAGO' AND
        IDTRANSACCION NOT IN (SELECT TRA_IDTRANSACCION FROM
TRANSACCION) AND
        (convert(DATE,FECHATRANSACCION ,126)) BETWEEN @FECHAINICIO
AND @FECHAFIN)

END
GO
/****** Object:  StoredProcedure [dbo].[BORRARARTICULOS]    Script Date:
06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BORRARARTICULOS]

  @IDDONACION bigint,
  @BENEFICIADODONACION varchar(50)

 AS
BEGIN

        WHILE (SELECT COUNT(*) FROM dbo.PUEDE_TENER WHERE
IDDONACION = @IDDONACION) > 0
        BEGIN
                DECLARE @IDPUEDETENER  bigint
                SET @IDPUEDETENER = (SELECT TOP 1 IDPUEDETENER FROM
dbo.PUEDE_TENER WHERE IDDONACION = @IDDONACION)
                DECLARE @CANTIDAD  int
                SET @CANTIDAD = (SELECT CANTIDADDONADA FROM
dbo.PUEDE_TENER WHERE IDPUEDETENER=@IDPUEDETENER)
                DECLARE @IDARTICULO  int
                SET @IDARTICULO = (SELECT IDARTICULO FROM
dbo.PUEDE_TENER WHERE IDPUEDETENER=@IDPUEDETENER)

                DELETE FROM dbo.PUEDE_TENER
                WHERE IDPUEDETENER = @IDPUEDETENER

                IF  @BENEFICIADODONACION = 'FUDIS'
                        Update ARTICULO
                        set CANTIDADARTICULO = CANTIDADARTICULO -
@CANTIDAD
                        where IDARTICULO = @IDARTICULO
```

```
                ELSE
                        Update ARTICULO
                        set CANTIDADARTICULO = CANTIDADARTICULO +
@CANTIDAD
                        where IDARTICULO = @IDARTICULO
                END



END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARDONACIONDINERO]    Script
Date: 06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARDONACIONDINERO]
  -- Add the parameters for the stored procedure here

        @CIDONADOR varchar(15),
        @NOMBREDONADOR varchar(50),
        @APELLIDODONADOR varchar(50),
        @EMPRESAREPRESENTANTEDONADOR varchar(50),
        @OBSEVACIONESDONADOR varchar(100),
        @BENEFICIADODONACION varchar(50),
        @DESCRIPCIONDONACION varchar(100),
        @FECHAINICIO DATE,
        @FECHAFIN DATE
AS
BEGIN

        SELECT IDDONACION,
        CIDONADOR,
        NOMBREDONADOR,
        APELLIDODONADOR,
        EMPRESAREPRESENTANTEDONADOR,
        OBSEVACIONESDONADOR,
        BENEFICIADODONACION,
        DESCRIPCIONDONACION,
        FECHADONACION,
        IDTRANSACCION,
        MONTOTRANSACCION
        FROM  dbo.VISTADONACION
        WHERE (CIDONADOR LIKE'%'+@CIDONADOR+'%' AND
        NOMBREDONADOR LIKE'%'+@NOMBREDONADOR+'%' AND
        APELLIDODONADOR LIKE'%'+@APELLIDODONADOR+'%' AND
        EMPRESAREPRESENTANTEDONADOR
LIKE'%'+@EMPRESAREPRESENTANTEDONADOR+'%' AND
        OBSEVACIONESDONADOR LIKE'%'+@OBSEVACIONESDONADOR+'%'
```

```
AND
        BENEFICIADODONACION LIKE'%'+@BENEFICIADODONACION+'%' AND
        DESCRIPCIONDONACION LIKE'%'+@DESCRIPCIONDONACION+'%' AND
        IDTIPODONACION = 1 AND
        (convert(DATE, FECHADONACION,126)) BETWEEN @FECHAINICIO
AND @FECHAFIN)

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARDONACIONARTICULO]
Script Date: 06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARDONACIONARTICULO]
  -- Add the parameters for the stored procedure here

        @CIDONADOR varchar(15),
        @NOMBREDONADOR varchar(50),
        @APELLIDODONADOR varchar(50),
        @EMPRESAREPRESENTANTEDONADOR varchar(50),
        @OBSEVACIONESDONADOR varchar(100),
        @BENEFICIADODONACION varchar(50),
        @DESCRIPCIONDONACION varchar(100),
        @FECHAINICIO DATE,
        @FECHAFIN DATE
AS
BEGIN

        SELECT IDDONACION,
        CIDONADOR,
        NOMBREDONADOR,
        APELLIDODONADOR,
        EMPRESAREPRESENTANTEDONADOR,
        OBSEVACIONESDONADOR,
        BENEFICIADODONACION,
        DESCRIPCIONDONACION,
        FECHADONACION

        FROM  dbo.VISTADONACION
        WHERE (CIDONADOR LIKE'%'+@CIDONADOR+'%' AND
        NOMBREDONADOR LIKE'%'+@NOMBREDONADOR+'%' AND
        APELLIDODONADOR LIKE'%'+@APELLIDODONADOR+'%' AND
        EMPRESAREPRESENTANTEDONADOR
LIKE'%'+@EMPRESAREPRESENTANTEDONADOR+'%' AND
        OBSEVACIONESDONADOR LIKE'%'+@OBSEVACIONESDONADOR+'%'
AND
        BENEFICIADODONACION LIKE'%'+@BENEFICIADODONACION+'%' AND
        DESCRIPCIONDONACION LIKE'%'+@DESCRIPCIONDONACION+'%' AND
```

```
        IDTIPODONACION = 2 AND
        (convert(DATE, FECHADONACION,126)) BETWEEN @FECHAINICIO
AND @FECHAFIN)

END
GO
/****** Object:  StoredProcedure [dbo].[BUSCARARTICULODONACION]
Script Date: 06/19/2013 23:21:16 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[BUSCARARTICULODONACION]
   -- Add the parameters for the stored procedure here

   @NOMBRE varchar(50),
   @LUGAR VARCHAR (50),
   @DESCRIPCION varchar(100),
   @IDDONACION bigint
AS
BEGIN

        SELECT IDPUEDETENER,
        IDARTICULO ,
        NOMBREARTICULO ,
        DESCRIPCIONARTICULO,
        NOMBRELUGAR,
        CANTIDADDONADA
        FROM  dbo.VISTAARTICULODONACION
        WHERE (NOMBREARTICULO LIKE '%'+@NOMBRE+'%' AND
        DESCRIPCIONARTICULO LIKE '%'+@DESCRIPCION+'%' AND
        NOMBRELUGAR LIKE '%'+@LUGAR+'%' AND
        IDARTICULO IN (SELECT IDARTICULO FROM dbo.PUEDE_TENER WHERE
IDDONACION = @IDDONACION))

END
GO
CREACION DE VISTAS
        USE [FUNDACIÓN FUDIS DISERTACIÓN DE GRADO]
GO
/****** Object:  View [dbo].[VISTAARTICULO]    Script Date: 06/19/2013
23:22:12 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[VISTAARTICULO]
AS
SELECT     dbo.ARTICULO.IDARTICULO, dbo.LUGAR.NOMBRELUGAR,
dbo.ARTICULO.NOMBREARTICULO, dbo.ARTICULO.CANTIDADARTICULO,
```

```
dbo.ARTICULO.DESCRIPCIONARTICULO
FROM       dbo.ARTICULO INNER JOIN
                dbo.LUGAR ON dbo.ARTICULO.IDLUGAR = dbo.LUGAR.IDLUGAR
WHERE    (dbo.ARTICULO.IDARTICULO <> - 1)
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1',
@value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
  Begin PaneConfigurations =
    Begin PaneConfiguration = 0
      NumPanes = 4
      Configuration = "(H (1[40] 4[20] 2[20] 3) )"
    End
    Begin PaneConfiguration = 1
      NumPanes = 3
      Configuration = "(H (1 [50] 4 [25] 3))"
    End
    Begin PaneConfiguration = 2
      NumPanes = 3
      Configuration = "(H (1 [50] 2 [25] 3))"
    End
    Begin PaneConfiguration = 3
      NumPanes = 3
      Configuration = "(H (4 [30] 2 [40] 3))"
    End
    Begin PaneConfiguration = 4
      NumPanes = 2
      Configuration = "(H (1 [56] 3))"
    End
    Begin PaneConfiguration = 5
      NumPanes = 2
      Configuration = "(H (2 [66] 3))"
    End
    Begin PaneConfiguration = 6
      NumPanes = 2
      Configuration = "(H (4 [50] 3))"
    End
    Begin PaneConfiguration = 7
      NumPanes = 1
      Configuration = "(V (3))"
    End
    Begin PaneConfiguration = 8
      NumPanes = 3
      Configuration = "(H (1[56] 4[18] 2) )"
    End
    Begin PaneConfiguration = 9
      NumPanes = 2
      Configuration = "(H (1 [75] 4))"
    End
    Begin PaneConfiguration = 10
```

```
      NumPanes = 2
      Configuration = "(H (1[66] 2) )"
    End
    Begin PaneConfiguration = 11
      NumPanes = 2
      Configuration = "(H (4 [60] 2))"
    End
    Begin PaneConfiguration = 12
      NumPanes = 1
      Configuration = "(H (1) )"
    End
    Begin PaneConfiguration = 13
      NumPanes = 1
      Configuration = "(V (4))"
    End
    Begin PaneConfiguration = 14
      NumPanes = 1
      Configuration = "(V (2))"
    End
    ActivePaneConfig = 0
  End
  Begin DiagramPane =
    Begin Origin =
      Top = 0
      Left = 0
    End
    Begin Tables =
      Begin Table = "ARTICULO"
        Begin Extent =
          Top = 6
          Left = 38
          Bottom = 126
          Right = 247
        End
        DisplayFlags = 280
        TopColumn = 0
      End
      Begin Table = "LUGAR"
        Begin Extent =
          Top = 6
          Left = 285
          Bottom = 111
          Right = 476
        End
        DisplayFlags = 280
        TopColumn = 0
      End
    End
  End
  Begin SQLPane =
```

```
    End
    Begin DataPane =
      Begin ParameterDefaults = ""
      End
      Begin ColumnWidths = 9
        Width = 284
        Width = 1500
        Width = 1500
        Width = 1500
        Width = 1500
        Width = 1500
        Width = 1500
        Width = 1500
        Width = 1500
      End
    End
    Begin CriteriaPane =
      Begin ColumnWidths = 11
        Column = 1440
        Alias = 900
        Table = 1170
        Output = 720
        Append = 1400
        NewValue = 1170
        SortType = 1350
        SortOrder = 1410
        GroupBy = 1350
        Filter = 1350
        Or = 1350
        Or = 1350
        Or = 1350
      End
    End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTAARTICULO'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount',
@value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTAARTICULO'
GO
/****** Object:  View [dbo].[VISTACAMA]    Script Date: 06/19/2013 23:22:12
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[VISTACAMA]
AS
SELECT     dbo.CAMA.IDCAMA, dbo.HABITACION.PISOHABITACION,
```

```
dbo.HABITACION.CODIGOHABITACION, dbo.TIPOCAMA.NOMBRETIPOCAMA,
dbo.CAMA.CODIGOCAMA, dbo.CAMA.DESCRIPCIONCAMA,
                  dbo.CAMA.DISPONIBILIDADCAMA,
dbo.CAMA.DIASDISPONIBLESCAMA
FROM        dbo.CAMA INNER JOIN
                  dbo.HABITACION ON dbo.CAMA.IDHABITACION =
dbo.HABITACION.IDHABITACION INNER JOIN
                  dbo.TIPOCAMA ON dbo.CAMA.IDTIPOCAMA =
dbo.TIPOCAMA.IDTIPOCAMA
WHERE     (dbo.CAMA.IDCAMA <> - 1) AND (dbo.CAMA.IDCAMA <> - 2)
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1',
@value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
  Begin PaneConfigurations =
    Begin PaneConfiguration = 0
      NumPanes = 4
      Configuration = "(H (1[40] 4[20] 2[20] 3) )"
    End
    Begin PaneConfiguration = 1
      NumPanes = 3
      Configuration = "(H (1 [50] 4 [25] 3))"
    End
    Begin PaneConfiguration = 2
      NumPanes = 3
      Configuration = "(H (1 [50] 2 [25] 3))"
    End
    Begin PaneConfiguration = 3
      NumPanes = 3
      Configuration = "(H (4 [30] 2 [40] 3))"
    End
    Begin PaneConfiguration = 4
      NumPanes = 2
      Configuration = "(H (1 [56] 3))"
    End
    Begin PaneConfiguration = 5
      NumPanes = 2
      Configuration = "(H (2 [66] 3))"
    End
    Begin PaneConfiguration = 6
      NumPanes = 2
      Configuration = "(H (4 [50] 3))"
    End
    Begin PaneConfiguration = 7
      NumPanes = 1
      Configuration = "(V (3))"
    End
    Begin PaneConfiguration = 8
      NumPanes = 3
      Configuration = "(H (1[56] 4[18] 2) )"
```

```
    End
    Begin PaneConfiguration = 9
      NumPanes = 2
      Configuration = "(H (1 [75] 4))"
    End
    Begin PaneConfiguration = 10
      NumPanes = 2
      Configuration = "(H (1[66] 2) )"
    End
    Begin PaneConfiguration = 11
      NumPanes = 2
      Configuration = "(H (4 [60] 2))"
    End
    Begin PaneConfiguration = 12
      NumPanes = 1
      Configuration = "(H (1) )"
    End
    Begin PaneConfiguration = 13
      NumPanes = 1
      Configuration = "(V (4))"
    End
    Begin PaneConfiguration = 14
      NumPanes = 1
      Configuration = "(V (2))"
    End
    ActivePaneConfig = 0
  End
  Begin DiagramPane =
    Begin Origin =
      Top = 0
      Left = 0
    End
    Begin Tables =
      Begin Table = "CAMA"
        Begin Extent =
          Top = 6
          Left = 38
          Bottom = 126
          Right = 238
        End
        DisplayFlags = 280
        TopColumn = 3
      End
      Begin Table = "HABITACION"
        Begin Extent =
          Top = 6
          Left = 276
          Bottom = 126
          Right = 497
        End
```

```
            DisplayFlags = 280
            TopColumn = 0
          End
          Begin Table = "TIPOCAMA"
            Begin Extent =
              Top = 6
              Left = 535
              Bottom = 111
              Right = 746
            End
            DisplayFlags = 280
            TopColumn = 0
          End
        End
      End
      Begin SQLPane =
      End
      Begin DataPane =
        Begin ParameterDefaults = ""
        End
        Begin ColumnWidths = 9
          Width = 284
          Width = 1500
          Width = 1500
          Width = 1500
          Width = 1500
          Width = 1500
          Width = 1500
          Width = 1500
          Width = 1500
        End
      End
      Begin CriteriaPane =
        Begin ColumnWidths = 11
          Column = 1440
          Alias = 900
          Table = 1170
          Output = 720
          Append = 1400
          NewValue = 1170
          SortType = 1350
          SortOrder = 1410
          GroupBy = 1350
          Filter = 1350
          Or = 1350
          Or = 1350
          Or = 1350
        End
      End
    End
```

```
' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTACAMA'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount',
@value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTACAMA'
GO
/****** Object:  View [dbo].[VISTATRANSACCION]    Script Date: 06/19/2013
23:22:12 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[VISTATRANSACCION]
AS
SELECT     dbo.TRANSACCION.IDTRANSACCION,
dbo.TRANSACCION.TRA_IDTRANSACCION, dbo.PACIENTE.CIPACIENTE,
dbo.PACIENTE.NOMBREPACIENTE, dbo.PACIENTE.APELLIDOPACIENTE,
                      dbo.TRANSACCION.IDCAMA,
dbo.ACOMPANANTE.CIACOMPANANTE,
dbo.ACOMPANANTE.NOMBREACOMPANANTE,
dbo.ACOMPANANTE.APELLIDOACOMPANANTE,
dbo.TRANSACCION.CAM_IDCAMA,
                      dbo.TRANSACCION.COSTODIARIOTRANSACCION,
dbo.TRANSACCION.MONTOTRANSACCION,
dbo.TRANSACCION.FECHATRANSACCION,
dbo.TRANSACCION.FECHAINICIOTRANSACCION,
                      dbo.TRANSACCION.FECHAFINTRANSACCION,
dbo.TIPOTRANSACCION.NOMBRETIPOTRANSACCION
FROM       dbo.ACOMPANANTE INNER JOIN
                      dbo.TRANSACCION ON dbo.ACOMPANANTE.IDACOMPANANTE =
dbo.TRANSACCION.IDACOMPANANTE INNER JOIN
                      dbo.PACIENTE ON dbo.TRANSACCION.IDPACIENTE =
dbo.PACIENTE.IDPACIENTE INNER JOIN
                      dbo.TIPOTRANSACCION ON
dbo.TRANSACCION.IDTIPOTRANSACCION =
dbo.TIPOTRANSACCION.IDTIPOTRANSACCION
WHERE    (dbo.TRANSACCION.IDTRANSACCION <> - 1)
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1',
@value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
  Begin PaneConfigurations =
    Begin PaneConfiguration = 0
      NumPanes = 4
      Configuration = "(H (1[40] 4[20] 2[20] 3) )"
    End
    Begin PaneConfiguration = 1
      NumPanes = 3
      Configuration = "(H (1 [50] 4 [25] 3))"
```

```
End
Begin PaneConfiguration = 2
  NumPanes = 3
  Configuration = "(H (1 [50] 2 [25] 3))"
End
Begin PaneConfiguration = 3
  NumPanes = 3
  Configuration = "(H (4 [30] 2 [40] 3))"
End
Begin PaneConfiguration = 4
  NumPanes = 2
  Configuration = "(H (1 [56] 3))"
End
Begin PaneConfiguration = 5
  NumPanes = 2
  Configuration = "(H (2 [66] 3))"
End
Begin PaneConfiguration = 6
  NumPanes = 2
  Configuration = "(H (4 [50] 3))"
End
Begin PaneConfiguration = 7
  NumPanes = 1
  Configuration = "(V (3))"
End
Begin PaneConfiguration = 8
  NumPanes = 3
  Configuration = "(H (1[56] 4[18] 2) )"
End
Begin PaneConfiguration = 9
  NumPanes = 2
  Configuration = "(H (1 [75] 4))"
End
Begin PaneConfiguration = 10
  NumPanes = 2
  Configuration = "(H (1[66] 2) )"
End
Begin PaneConfiguration = 11
  NumPanes = 2
  Configuration = "(H (4 [60] 2))"
End
Begin PaneConfiguration = 12
  NumPanes = 1
  Configuration = "(H (1) )"
End
Begin PaneConfiguration = 13
  NumPanes = 1
  Configuration = "(V (4))"
End
Begin PaneConfiguration = 14
```

```
      NumPanes = 1
      Configuration = "(V (2))"
    End
    ActivePaneConfig = 0
  End
  Begin DiagramPane =
    Begin Origin =
      Top = 0
      Left = 0
    End
    Begin Tables =
      Begin Table = "ACOMPANANTE"
        Begin Extent =
          Top = 6
          Left = 38
          Bottom = 126
          Right = 251
        End
        DisplayFlags = 280
        TopColumn = 0
      End
      Begin Table = "TRANSACCION"
        Begin Extent =
          Top = 11
          Left = 493
          Bottom = 131
          Right = 809
        End
        DisplayFlags = 280
        TopColumn = 0
      End
      Begin Table = "PACIENTE"
        Begin Extent =
          Top = 6
          Left = 289
          Bottom = 126
          Right = 528
        End
        DisplayFlags = 280
        TopColumn = 0
      End
      Begin Table = "TIPOTRANSACCION"
        Begin Extent =
          Top = 41
          Left = 241
          Bottom = 146
          Right = 496
        End
        DisplayFlags = 280
        TopColumn = 0
```

```
      End
    End
  End
  Begin SQLPane =
  End
  Begin DataPane =
    Begin ParameterDefaults = ""
    End
    Begin ColumnWidths = 9
      Width = 284
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
    End
  End
  Begin CriteriaPane =
    Begin ColumnWidths = 11
      Column = 1440
      Alias = 900
      Table = 1170
      Output = 720
      Append = 1400
      NewValue = 1170
      SortType = 1350
      SortOrder = 1410
      GroupBy = 1350
      Filter = 1350
      Or = 1350
      Or = 1350
      Or = 135' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTATRANSACCION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane2',
@value=N'0
    End
  End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTATRANSACCION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount',
@value=2 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTATRANSACCION'
GO
/****** Object:  View [dbo].[VISTADONACION]    Script Date: 06/19/2013
```

```
23:22:12 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[VISTADONACION]
AS
SELECT    dbo.DONACION.IDDONACION, dbo.DONADOR.CIDONADOR,
dbo.DONADOR.NOMBREDONADOR, dbo.DONADOR.APELLIDODONADOR,
dbo.DONADOR.EMPRESAREPRESENTANTEDONADOR,
                dbo.DONADOR.OBSEVACIONESDONADOR,
dbo.DONACION.BENEFICIADODONACION, dbo.DONACION.FECHADONACION,
dbo.DONACION.DESCRIPCIONDONACION, dbo.DONACION.IDTRANSACCION,
                dbo.DONACION.IDTIPODONACION,
dbo.TRANSACCION.MONTOTRANSACCION
FROM       dbo.DONACION INNER JOIN
                dbo.DONADOR ON dbo.DONACION.IDDONADOR =
dbo.DONADOR.IDDONADOR INNER JOIN
                dbo.TRANSACCION ON dbo.DONACION.IDTRANSACCION =
dbo.TRANSACCION.IDTRANSACCION
WHERE    (dbo.DONACION.IDTIPODONACION <> - 1)
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1',
@value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
  Begin PaneConfigurations =
    Begin PaneConfiguration = 0
      NumPanes = 4
      Configuration = "(H (1[40] 4[20] 2[20] 3) )"
    End
    Begin PaneConfiguration = 1
      NumPanes = 3
      Configuration = "(H (1 [50] 4 [25] 3))"
    End
    Begin PaneConfiguration = 2
      NumPanes = 3
      Configuration = "(H (1 [50] 2 [25] 3))"
    End
    Begin PaneConfiguration = 3
      NumPanes = 3
      Configuration = "(H (4 [30] 2 [40] 3))"
    End
    Begin PaneConfiguration = 4
      NumPanes = 2
      Configuration = "(H (1 [56] 3))"
    End
    Begin PaneConfiguration = 5
      NumPanes = 2
      Configuration = "(H (2 [66] 3))"
    End
```

```
    Begin PaneConfiguration = 6
      NumPanes = 2
      Configuration = "(H (4 [50] 3))"
    End
    Begin PaneConfiguration = 7
      NumPanes = 1
      Configuration = "(V (3))"
    End
    Begin PaneConfiguration = 8
      NumPanes = 3
      Configuration = "(H (1[56] 4[18] 2) )"
    End
    Begin PaneConfiguration = 9
      NumPanes = 2
      Configuration = "(H (1 [75] 4))"
    End
    Begin PaneConfiguration = 10
      NumPanes = 2
      Configuration = "(H (1[66] 2) )"
    End
    Begin PaneConfiguration = 11
      NumPanes = 2
      Configuration = "(H (4 [60] 2))"
    End
    Begin PaneConfiguration = 12
      NumPanes = 1
      Configuration = "(H (1) )"
    End
    Begin PaneConfiguration = 13
      NumPanes = 1
      Configuration = "(V (4))"
    End
    Begin PaneConfiguration = 14
      NumPanes = 1
      Configuration = "(V (2))"
    End
    ActivePaneConfig = 0
  End
  Begin DiagramPane =
    Begin Origin =
      Top = 0
      Left = 0
    End
    Begin Tables =
      Begin Table = "DONACION"
        Begin Extent =
          Top = 6
          Left = 38
          Bottom = 126
          Right = 251
```

```
        End
        DisplayFlags = 280
        TopColumn = 3
      End
      Begin Table = "DONADOR"
        Begin Extent =
          Top = 6
          Left = 289
          Bottom = 126
          Right = 558
        End
        DisplayFlags = 280
        TopColumn = 0
      End
      Begin Table = "TRANSACCION"
        Begin Extent =
          Top = 6
          Left = 596
          Bottom = 126
          Right = 830
        End
        DisplayFlags = 280
        TopColumn = 1
      End
    End
  End
  Begin SQLPane =
  End
  Begin DataPane =
    Begin ParameterDefaults = ""
    End
    Begin ColumnWidths = 9
      Width = 284
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
    End
  End
  Begin CriteriaPane =
    Begin ColumnWidths = 11
      Column = 1440
      Alias = 900
      Table = 1170
      Output = 720
      Append = 1400
```

```
      NewValue = 1170
      SortType = 1350
      SortOrder = 1410
      GroupBy = 1350
      Filter = 1350
      Or = 1350
      Or = 1350
      Or = 1350
    End
  End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTADONACION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount',
@value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTADONACION'
GO
/****** Object:  View [dbo].[VISTAARTICULODONACION]    Script Date:
06/19/2013 23:22:12 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[VISTAARTICULODONACION]
AS
SELECT     dbo.ARTICULO.IDARTICULO, dbo.ARTICULO.NOMBREARTICULO,
dbo.ARTICULO.DESCRIPCIONARTICULO, dbo.LUGAR.NOMBRELUGAR,
dbo.PUEDE_TENER.CANTIDADDONADA,
                      dbo.PUEDE_TENER.IDDONACION,
dbo.PUEDE_TENER.IDPUEDETENER
FROM         dbo.ARTICULO INNER JOIN
                      dbo.LUGAR ON dbo.ARTICULO.IDLUGAR = dbo.LUGAR.IDLUGAR
INNER JOIN
                      dbo.PUEDE_TENER ON dbo.ARTICULO.IDARTICULO =
dbo.PUEDE_TENER.IDARTICULO
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1',
@value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
  Begin PaneConfigurations =
    Begin PaneConfiguration = 0
      NumPanes = 4
      Configuration = "(H (1[40] 4[20] 2[20] 3) )"
    End
    Begin PaneConfiguration = 1
      NumPanes = 3
      Configuration = "(H (1 [50] 4 [25] 3))"
    End
    Begin PaneConfiguration = 2
```

```
    NumPanes = 3
    Configuration = "(H (1 [50] 2 [25] 3))"
End
Begin PaneConfiguration = 3
    NumPanes = 3
    Configuration = "(H (4 [30] 2 [40] 3))"
End
Begin PaneConfiguration = 4
    NumPanes = 2
    Configuration = "(H (1 [56] 3))"
End
Begin PaneConfiguration = 5
    NumPanes = 2
    Configuration = "(H (2 [66] 3))"
End
Begin PaneConfiguration = 6
    NumPanes = 2
    Configuration = "(H (4 [50] 3))"
End
Begin PaneConfiguration = 7
    NumPanes = 1
    Configuration = "(V (3))"
End
Begin PaneConfiguration = 8
    NumPanes = 3
    Configuration = "(H (1[56] 4[18] 2) )"
End
Begin PaneConfiguration = 9
    NumPanes = 2
    Configuration = "(H (1 [75] 4))"
End
Begin PaneConfiguration = 10
    NumPanes = 2
    Configuration = "(H (1[66] 2) )"
End
Begin PaneConfiguration = 11
    NumPanes = 2
    Configuration = "(H (4 [60] 2))"
End
Begin PaneConfiguration = 12
    NumPanes = 1
    Configuration = "(H (1) )"
End
Begin PaneConfiguration = 13
    NumPanes = 1
    Configuration = "(V (4))"
End
Begin PaneConfiguration = 14
    NumPanes = 1
    Configuration = "(V (2))"
```

```
    End
    ActivePaneConfig = 0
  End
  Begin DiagramPane =
    Begin Origin =
      Top = 0
      Left = 0
    End
    Begin Tables =
      Begin Table = "ARTICULO"
        Begin Extent =
          Top = 6
          Left = 38
          Bottom = 126
          Right = 247
        End
        DisplayFlags = 280
        TopColumn = 0
      End
      Begin Table = "LUGAR"
        Begin Extent =
          Top = 6
          Left = 285
          Bottom = 111
          Right = 476
        End
        DisplayFlags = 280
        TopColumn = 0
      End
      Begin Table = "PUEDE_TENER"
        Begin Extent =
          Top = 6
          Left = 514
          Bottom = 111
          Right = 698
        End
        DisplayFlags = 280
        TopColumn = 0
      End
    End
  End
  Begin SQLPane =
  End
  Begin DataPane =
    Begin ParameterDefaults = ""
    End
    Begin ColumnWidths = 9
      Width = 284
      Width = 1500
      Width = 1500
```

```
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
    End
  End
  Begin CriteriaPane =
    Begin ColumnWidths = 11
      Column = 1440
      Alias = 900
      Table = 1170
      Output = 720
      Append = 1400
      NewValue = 1170
      SortType = 1350
      SortOrder = 1410
      GroupBy = 1350
      Filter = 1350
      Or = 1350
      Or = 1350
      Or = 1350
    End
  End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTAARTICULODONACION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount',
@value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'VISTAARTICULODONACION'
GO
```

A continuación indicaremos el código fuente del programa

FORMULARIOS

LOGIN

```
Public Class login
   Public Shared catego As String
      Private Sub OK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OK.Click
```

```
        'declaramos una variable nombre para obtener un valor cuando se
ingresa usuario y contraseña en el formulario de login
    Dim nombre = Me.USUARIOTableAdapter.loguin(UsernameTextBox.Text,
PasswordTextBox.Text)
                'declaramos una variable tipo para obtener la categoria del
usuario ingresado
    Dim tipo = Me.USUARIOTableAdapter.tipo(UsernameTextBox.Text,
PasswordTextBox.Text)
                'si la variable no tiene nada q mostrar se indicara el mensaje
que los datos son incorrectos
    If nombre Is Nothing Then
      MsgBox("Lo sentimos, los datos no han sidos encontrados, por favor
intentelo nuevamente.", MsgBoxStyle.Critical, "Aviso del Sistema")
        UsernameTextBox.Clear()
        PasswordTextBox.Clear()
        UsernameTextBox.Focus()
    Else
                'aqui a travez de la variable tipo comprobaremos si existe las
tres distintas categorias y si mostrará un mensaje de ingreso exitoso,
                'ademas se pasará el nombre y categoria a la pagina principal
        If tipo = "Administrador" Or tipo = "ADMINISTRADOR" Or tipo =
"administrador" Then
          MsgBox("Ingreso Exitoso, los datos han sido comprobados
correctamente.", MsgBoxStyle.Information, "Aviso del Sistema")
          Principal.Show()
          Principal.ToolStripLabel2.Text = UsernameTextBox.Text
          Principal.ToolStripLabel4.Text = tipo
          'catego = tipo
        ElseIf tipo = "Tecnico" Or tipo = "TECNICO" Or tipo = "tecnico" Then
          MsgBox("Ingreso Exitoso, los datos han sido comprobados
correctamente.", MsgBoxStyle.Information, "Aviso del Sistema")
          Principal.Show()
          Principal.ToolStripLabel2.Text = UsernameTextBox.Text
          Principal.ToolStripLabel4.Text = tipo
          Principal.UsuarioToolStripMenuItem.Visible = False
        ElseIf tipo = "Usuario" Or tipo = "usuario" Or tipo = "USUARIO" Then
          MsgBox("Ingreso Exitoso, los datos han sido comprobados
correctamente.", MsgBoxStyle.Information, "Aviso del Sistema")
          Principal.ToolStripLabel2.Text = UsernameTextBox.Text
          Principal.ToolStripLabel4.Text = tipo
        Else
          MsgBox("Lo sentimos, los datos no han sidos encontrados, por favor
intentelo nuevamente", MsgBoxStyle.Critical, "Aviso del Sistema")
          UsernameTextBox.Clear()
          PasswordTextBox.Clear()
          UsernameTextBox.Focus()
        End If
        Me.Close()
    End If
  End Sub
```

```
    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Cancel.Click
        'boton para cerrar la aplicación
        Me.Close()
    End Sub
End Class
```

PRINCIPAL

```
Imports System.Windows.Forms

Public Class Principal

    Private Sub ShowNewForm(ByVal sender As Object, ByVal e As EventArgs)
Handles NewToolStripMenuItem.Click, NewWindowToolStripMenuItem.Click
        ' Cree una nueva instancia del formulario secundario.
        Dim ChildForm As New System.Windows.Forms.Form
        ' Conviértalo en un elemento secundario de este formulario MDI antes de
mostrarlo.
        ChildForm.MdiParent = Me

        m_ChildFormNumber += 1
        ChildForm.Text = "Ventana " & m_ChildFormNumber

        ChildForm.Show()
    End Sub

    Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs) Handles
OpenToolStripMenuItem.Click
        Dim OpenFileDialog As New OpenFileDialog
        OpenFileDialog.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments
        OpenFileDialog.Filter = "Archivos de texto (*.txt)|*.txt|Todos los archivos
(*.*)|*.*"
        If (OpenFileDialog.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then
            Dim FileName As String = OpenFileDialog.FileName
            ' TODO: agregue código aquí para abrir el archivo.
        End If
    End Sub

    Private Sub SaveAsToolStripMenuItem_Click(ByVal sender As Object, ByVal e
As EventArgs) Handles SaveAsToolStripMenuItem.Click
        Dim SaveFileDialog As New SaveFileDialog
        SaveFileDialog.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments
        SaveFileDialog.Filter = "Archivos de texto (*.txt)|*.txt|Todos los archivos
(*.*)|*.*"
```

```
        If (SaveFileDialog.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then
            Dim FileName As String = SaveFileDialog.FileName
            ' TODO: agregue código aquí para guardar el contenido actual del
formulario en un archivo.
        End If
    End Sub


    Private Sub ExitToolsStripMenuItem_Click(ByVal sender As Object, ByVal e
As EventArgs) Handles ExitToolStripMenuItem.Click
        Me.Close()
    End Sub

    Private Sub CutToolStripMenuItem_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles CutToolStripMenuItem.Click
        ' Utilice My.Computer.Clipboard para insertar el texto o las imágenes
seleccionadas en el Portapapeles
    End Sub

    Private Sub CopyToolStripMenuItem_Click(ByVal sender As Object, ByVal e
As EventArgs) Handles CopyToolStripMenuItem.Click
        ' Utilice My.Computer.Clipboard para insertar el texto o las imágenes
seleccionadas en el Portapapeles
    End Sub

    Private Sub PasteToolStripMenuItem_Click(ByVal sender As Object, ByVal e
As EventArgs) Handles PasteToolStripMenuItem.Click
        'Utilice My.Computer.Clipboard.GetText() o
My.Computer.Clipboard.GetData para recuperar la información del
Portapapeles.
    End Sub

    Private Sub ToolBarToolStripMenuItem_Click(ByVal sender As Object, ByVal
e As EventArgs) Handles ToolBarToolStripMenuItem.Click
        Me.ToolStrip.Visible = Me.ToolBarToolStripMenuItem.Checked
    End Sub

    Private Sub StatusBarToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As EventArgs) Handles StatusBarToolStripMenuItem.Click
        Me.StatusStrip.Visible = Me.StatusBarToolStripMenuItem.Checked
    End Sub

    Private Sub CascadeToolStripMenuItem_Click(ByVal sender As Object, ByVal
e As EventArgs) Handles CascadeToolStripMenuItem.Click
        Me.LayoutMdi(MdiLayout.Cascade)
    End Sub

    Private Sub TileVerticalToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As EventArgs) Handles TileVerticalToolStripMenuItem.Click
```

```
        Me.LayoutMdi(MdiLayout.TileVertical)
    End Sub

    Private Sub TileHorizontalToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As EventArgs) Handles TileHorizontalToolStripMenuItem.Click
        Me.LayoutMdi(MdiLayout.TileHorizontal)
    End Sub

    Private Sub ArrangeIconsToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As EventArgs) Handles ArrangeIconsToolStripMenuItem.Click
        Me.LayoutMdi(MdiLayout.ArrangeIcons)
    End Sub

    Private Sub CloseAllToolStripMenuItem_Click(ByVal sender As Object, ByVal
e As EventArgs) Handles CloseAllToolStripMenuItem.Click
        ' Cierre todos los formularios secundarios del principal.
        For Each ChildForm As Form In Me.MdiChildren
            ChildForm.Close()
        Next
    End Sub

    Private m_ChildFormNumber As Integer


    Private Sub LugarToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles LugarToolStripMenuItem.Click
        Dim frm As New VERLUGAR 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        'VERLUGAR.Show()
    End Sub

    Private Sub DonadorToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DonadorToolStripMenuItem.Click
        Dim frm As New VERDONADOR 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        'VERDONADOR.Show()
    End Sub

    Private Sub OptionsToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
OptionsToolStripMenuItem.Click
        Dim frm As New VERHABITACION 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
```

```
        ' VERHABITACION.Show()
    End Sub

    Private Sub PacienteToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
PacienteToolStripMenuItem.Click
        Dim frm As New VERPACIENTE 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        'VERPACIENTE.Show()
    End Sub

    Private Sub TipoDeCamaToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TipoDeCamaToolStripMenuItem.Click
        Dim frm As New VERTIPOCAMA 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        'VERTIPOCAMA.Show()
    End Sub

    Private Sub CamaToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CamaToolStripMenuItem.Click
        Dim frm As New VERCAMA 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        ' VERCAMA.Show()
    End Sub

    Private Sub UsuarioToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
UsuarioToolStripMenuItem.Click
        Dim frm As New VERUSUARIO 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
    End Sub

    Private Sub AcompañanteToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AcompañanteToolStripMenuItem.Click
        Dim frm As New VERACOMPANANTE 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        'VERACOMPANANTE.Show()
    End Sub
```

```
    Private Sub TipoDeDonaciónToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub TipoDeTransacciónToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub PagoToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles PagoToolStripMenuItem.Click
        Dim frm As New VERPAGO 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        ' VERPAGO.Show()
    End Sub

    Private Sub DineroToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles DineroToolStripMenuItem.Click
        Dim frm As New VERDONACIONDINERO 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        'VERDONACIONDINERO.Show()
    End Sub

    Private Sub ArtículoToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles ArtículoToolStripMenuItem.Click
        Dim frm As New VERDONACIÓNARTÍCULO 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        'VERDONACIÓNARTÍCULO.Show()
    End Sub

    Private Sub InventarioToolStripMenuItem_Click(sender As System.Object, e
As System.EventArgs) Handles InventarioToolStripMenuItem.Click
        Dim frm As New VERARTÍCULO 'abrir varias ventanas
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text
        'VERARTÍCULO.Show()
    End Sub

    Private Sub DiarioToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles DiarioToolStripMenuItem.Click
        Dim frm As New DIARIO 'abrir varias ventanas
```

```
        frm.MdiParent = Me 'abrir en MDI
        frm.Show()   'Mostrar el frame
        ToolStripStatusLabel.Text = frm.Text '
        'DIARIO.Show()
        DIARIO.FECHADateTimePicker.Value = Today
    End Sub

    Private Sub Principal_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load

CAMA.CAMATableAdapter.ACTUALIZARTODASLASCAMAS(VERCAMA.FUNDACI
ÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.CAMA)
    End Sub

End Class
```

ACOMPAÑANTE

```
Public Class ACOMPANANTE

    Private Sub ACOMPANANTEBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
ACOMPANANTEBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.ACOMPANANTEBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.FUDIS_BASEDataSet)

    End Sub

    Private Sub ACOMPANANTE_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUDIS_BASEDataSet.ACOMPANANTE' Puede moverla o quitarla según sea
necesario.

Me.ACOMPANANTETableAdapter.Fill(Me.FUDIS_BASEDataSet.ACOMPANANTE
)

    End Sub

    Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub

    Private Sub GUARDARACOMPANANTEButton_Click(sender As
System.Object, e As System.EventArgs) Handles
GUARDARACOMPANANTEButton.Click
```

```
        If NOMBREACOMPANANTETextBox.Text = "" Or
APELLIDOACOMPANANTETextBox.Text = "" Then
            If NOMBREACOMPANANTETextBox.Text = "" Then
                NOMBREOBLLabel.Text = "Campo Obligatorio"
            Else
                NOMBREOBLLabel.Text = "*"
            End If
            If APELLIDOACOMPANANTETextBox.Text = "" Then
                APELLIDOOBLLabel.Text = "Campo Obligatorio"
            Else
                APELLIDOOBLLabel.Text = "*"
            End If
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        Else
            If CIACOMPANANTETextBox.Text = "" Then
                CIACOMPANANTETextBox.Text = " "
            End If
            ACOMPANANTEBindingNavigatorSaveItem.PerformClick()
            MsgBox("ACOMPAÑANTE GUARDADO")
            Me.Dispose()
            VERACOMPANANTE.CIToolStripTextBox.Text = "1"
            VERACOMPANANTE.CIToolStripTextBox.Text = ""
        End If
    End Sub

    Private Sub ELIMINARACOMPANANTEButton_Click(sender As System.Object,
e As System.EventArgs) Handles ELIMINARACOMPANANTEButton.Click
        If (MessageBox.Show("¿Sí borra el acompañante """ +
NOMBREACOMPANANTETextBox.Text + " " +
APELLIDOACOMPANANTETextBox.Text + """, también se borraran las
transacciones asociadas al mismo. ¿Desea continuar?", "Notificación",
MessageBoxButtons.OKCancel) = DialogResult.OK) Then
            BindingNavigatorDeleteItem.PerformClick()
            ACOMPANANTEBindingNavigatorSaveItem.PerformClick()
            MsgBox("ACOMPAÑANTE ELIMINADO")
            Me.Dispose()
            VERACOMPANANTE.CIToolStripTextBox.Text = "1"
            VERACOMPANANTE.CIToolStripTextBox.Text = ""
        End If
    End Sub

    Private Sub CIACOMPANANTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
CIACOMPANANTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(CILabel,
CIACOMPANANTETextBox)
    End Sub
```

```
    Private Sub NOMBREACOMPANANTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
NOMBREACOMPANANTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(NOMBRELabel,
NOMBREACOMPANANTETextBox)
    End Sub

    Private Sub APELLIDOACOMPANANTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
APELLIDOACOMPANANTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(APELLIDOLabel,
APELLIDOACOMPANANTETextBox)

    End Sub

    Private Sub CIACOMPANANTETextBox_KeyPress(sender As System.Object, e
As System.Windows.Forms.KeyPressEventArgs) Handles
CIACOMPANANTETextBox.KeyPress
        If Char.IsNumber(e.KeyChar) Then
            e.Handled = False
        ElseIf Char.IsLetter(e.KeyChar) Then
            e.Handled = True
        ElseIf Char.IsControl(e.KeyChar) Then
            e.Handled = False
        ElseIf Char.IsSeparator(e.KeyChar) Then
            e.Handled = True
        End If
    End Sub
End Class
```

AGREGARARTICULODONACION

```
Public Class AGREGARARTÍCULODONACION

    Private Sub PUEDE_TENERBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
PUEDE_TENERBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PUEDE_TENERBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_D
E_GRADODataSet)

    End Sub

    Private Sub AGREGARARTÍCULODONACION_Load(sender As System.Object,
e As System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.PUEDE_TENER' Puede
moverla o quitarla según sea necesario.
```

```
Me.PUEDE_TENERTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE
_GRADODataSet.PUEDE_TENER)

    End Sub

    Private Sub ARTICULOButton_Click(sender As System.Object, e As
System.EventArgs) Handles ARTICULOButton.Click
        VERARTÍCULO.Show()
        VERARTÍCULO.BUSCARARTICULODataGridView.Columns(1).Visible = True
        VERARTÍCULO.Text = "VER ARTÍCULO DONACIÓN "
        If BENEFICIADOButton.Text <> "FUDIS" Then
            VERARTÍCULO.ToolStripLabel1.Visible = False
            VERARTÍCULO.CREARToolStripButton.Visible = False
            VERARTÍCULO.ToolStripButtonCREAR.Visible = False
        End If
    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlusuariocampoobligatorio(BENEFICIADOOBLLabel,
IDDONACIONTextBox, ARTICULOOBLLabel, IDARTICULOTextBox,
CANTIDADOBLLabel, CANTIDADDONADATextBox) Then


            If TARTÍCULOLabel.Text = "NUEVO" Then
                If BENEFICIADOButton.Text <> "FUDIS" Then
                    If Convert.ToInt32(CANTIDADARTICULONUEVOLabel.Text) <
Convert.ToInt32(CANTIDADDONADATextBox.Text) Then
                        MsgBox("NO SE PUEDE DONAR MÁS DE LO QUE SE POSEE")
                    Else

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO((Convert.
ToInt32(CANTIDADARTICULONUEVOLabel.Text) -
Convert.ToInt32(CANTIDADDONADATextBox.Text)),
Convert.ToInt64(IDARTICULOTextBox.Text))
                        PUEDE_TENERBindingNavigatorSaveItem.PerformClick()
                        MsgBox("DONACIÓN DE ARTÍCULO GUARDADA") 'GUARDA LOS
DATOS
                        Me.Dispose()
                    End If
                Else

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO((Convert.
ToInt32(CANTIDADARTICULONUEVOLabel.Text) +
Convert.ToInt32(CANTIDADDONADATextBox.Text)),
Convert.ToInt64(IDARTICULOTextBox.Text))
                    PUEDE_TENERBindingNavigatorSaveItem.PerformClick()
                    MsgBox("DONACIÓN DE ARTÍCULO GUARDADA") 'GUARDA LOS
DATOS
```

```
            Me.Dispose()
        End If
    Else
        Dim CANTIDADANTERIOR, IDARTICULOANTERIOR, CANTIDADNUEVA,
IDARTICULONUEVO, CANTIDADDONADA As Int32
        Try



            CANTIDADANTERIOR =
Convert.ToInt32(CANTIDADARTICULOVIEJOLabel.Text)
            IDARTICULOANTERIOR =
Convert.ToInt32(IDARTICULOANTERIORLabel.Text)
            CANTIDADNUEVA =
Convert.ToInt32(CANTIDADARTICULONUEVOLabel.Text)
            IDARTICULONUEVO = Convert.ToInt32(IDARTICULOTextBox.Text)
            CANTIDADDONADA =
Convert.ToInt32(CANTIDADDONADATextBox.Text)
        Catch ex As Exception
            Me.Dispose()
        End Try
        If BENEFICIADOButton.Text <> "FUDIS" Then
            If CANTIDADNUEVA < CANTIDADDONADA Then
                MsgBox("NO SE PUEDE DONAR MÁS DE LO QUE SE POSEE")
            Else

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO(CANTIDA
DANTERIOR + CANTIDADDONADA, IDARTICULOANTERIOR)

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO(CANTIDA
DNUEVA - CANTIDADDONADA, IDARTICULONUEVO)
                PUEDE_TENERBindingNavigatorSaveItem.PerformClick()
                MsgBox("DONACIÓN DE ARTÍCULO GUARDADA") 'GUARDA LOS
DATOS
                Me.Dispose()
            End If
        Else
            If CANTIDADANTERIOR < CANTIDADDONADA Then
                MsgBox("NO SE PUEDE DONAR MÁS DE LO QUE SE POSEE")
            Else

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO(CANTIDA
DANTERIOR - CANTIDADDONADA, IDARTICULOANTERIOR)

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO(CANTIDA
DNUEVA + CANTIDADDONADA, IDARTICULONUEVO)
                PUEDE_TENERBindingNavigatorSaveItem.PerformClick()
                MsgBox("DONACIÓN DE ARTÍCULO GUARDADA") 'GUARDA LOS
DATOS
                Me.Dispose()
```

```
            End If
        End If


    End If

    Try

        VERARTÍCULODONACIÓN.NOMBRETextBox.Text = "1"
        VERARTÍCULODONACIÓN.NOMBRETextBox.Text = "" 'SE USA
PARA RECARGAR
    Catch ex As Exception
        'EN CASO DE SER INGRESO DIRECTO NO RECARGA NADA
    End Try
  Else
    MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
  End If

End Sub

Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
  If (MessageBox.Show("¿DESEA ELIMINAR EL ARTÍCULO """ +
ARTICULOButton.Text + """?", "Notificación", MessageBoxButtons.OKCancel) =
DialogResult.OK) Then


    Dim CANTIDADANTERIOR, IDARTICULOANTERIOR, CANTIDADNUEVA,
IDARTICULONUEVO, CANTIDADDONADA As Int32
    CANTIDADANTERIOR =
Convert.ToInt32(CANTIDADARTICULOVIEJOLabel.Text)
    IDARTICULOANTERIOR =
Convert.ToInt32(IDARTICULOANTERIORLabel.Text)
    CANTIDADDONADA = Convert.ToInt32(CANTIDADDONADATextBox.Text)
    If BENEFICIADOButton.Text <> "FUDIS" Then
        If CANTIDADNUEVA < CANTIDADDONADA Then
            MsgBox("NO SE PUEDE DONAR MÁS DE LO QUE SE POSEE")
        Else

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO(CANTIDA
DANTERIOR + CANTIDADDONADA, IDARTICULOANTERIOR)

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO(CANTIDA
DNUEVA - CANTIDADDONADA, IDARTICULONUEVO)
            PUEDE_TENERBindingNavigatorSaveItem.PerformClick()
            MsgBox("DONACIÓN DE ARTÍCULO GUARDADA") 'GUARDA LOS
DATOS
            Me.Dispose()
        End If
    Else
```

```
            If CANTIDADANTERIOR < CANTIDADDONADA Then
                MsgBox("NO SE PUEDE DONAR MÁS DE LO QUE SE POSEE")
            Else

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO(CANTIDA
DANTERIOR - CANTIDADDONADA, IDARTICULOANTERIOR)

ARTÍCULO.ARTICULOTableAdapter.ACTUALIZARCANTIDADARTÍCULO(CANTIDA
DNUEVA + CANTIDADDONADA, IDARTICULONUEVO)
                PUEDE_TENERBindingNavigatorSaveItem.PerformClick()
                MsgBox("DONACIÓN DE ARTÍCULO GUARDADA") 'GUARDA LOS
DATOS
                Me.Dispose()
            End If
        End If



        Try

            VERARTÍCULODONACIÓN.NOMBRETOolStripTextBox.Text = "1"
            VERARTÍCULODONACIÓN.NOMBRETOolStripTextBox.Text = "" 'SE USA
PARA RECARGAR
        Catch ex As Exception
            'EN CASO DE SER INGRESO DIRECTO NO RECARGA NADA
        End Try



        BindingNavigatorDeleteItem.PerformClick()
        PUEDE_TENERBindingNavigatorSaveItem.PerformClick()
        MsgBox("ARTICULO ELIMINADO DE LA DONACIÓN")
        Me.Dispose()
        Try
            VERARTÍCULODONACIÓN.NOMBRETOolStripTextBox.Text = "1"
            VERARTÍCULODONACIÓN.NOMBRETOolStripTextBox.Text = "" 'SE USA
PARA RECARGAR LA VISTA
        Catch ex As Exception

        End Try

    End If
  End Sub

  Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
    Me.Dispose()
  End Sub
```

```
    Private Sub BENEFICIADOButton_MouseHover(sender As System.Object, e
As System.EventArgs) Handles BENEFICIADOButton.MouseHover
        Try
            Dim DESCRIPCIÓNDONACIÓN As String

            DESCRIPCIÓNDONACIÓN =
DONACIÓNARTÍCULO.DONACIONTableAdapter.DESCRIPCIONDONACION(IDDO
NACIONTextBox.Text)
            DESCRIPCIÓNDONACIÓN =
controlUsuario.añadirenter(DESCRIPCIÓNDONACIÓN)
            DONACIONLabel.Text = DESCRIPCIÓNDONACIÓN
        Catch ex As Exception

        End Try
    End Sub

    Private Sub BENEFICIADOButton_MouseLeave(sender As System.Object, e
As System.EventArgs) Handles BENEFICIADOButton.MouseLeave
        DONACIONLabel.Text = ""
    End Sub

    Private Sub ARTICULOButton_MouseHover(sender As System.Object, e As
System.EventArgs) Handles ARTICULOButton.MouseHover
        Try
            Dim DESCRIPCIÓNARTÍCULO As String

            DESCRIPCIÓNARTÍCULO =
ARTÍCULO.ARTICULOTableAdapter.DESCRIPCIONARTICULO(IDARTICULOTextBo
x.Text)
            DESCRIPCIÓNARTÍCULO =
controlUsuario.añadirenter(DESCRIPCIÓNARTÍCULO)
            ARTICULOLabel.Text = DESCRIPCIÓNARTÍCULO
        Catch ex As Exception

        End Try
    End Sub

    Private Sub ARTICULOButton_MouseLeave(sender As System.Object, e As
System.EventArgs) Handles ARTICULOButton.MouseLeave
        ARTICULOLabel.Text = ""
    End Sub

    Private Sub IDDONACIONTextBox_TextChanged(sender As System.Object, e
As System.EventArgs) Handles IDDONACIONTextBox.TextChanged
        Try
            Dim BENEFICIARIODONACION As String

            BENEFICIARIODONACION =
DONACIÓNARTÍCULO.DONACIONTableAdapter.BENEFICIADODONACION(IDDO
NACIONTextBox.Text)
```

```
            BENEFICIADOButton.Text = BENEFICIARIODONACION
        Catch ex As Exception
            BENEFICIADOButton.Text = "BENEFICIADO"
        End Try
    End Sub

    Private Sub IDARTICULOTextBox_TextChanged(sender As System.Object, e
As System.EventArgs) Handles IDARTICULOTextBox.TextChanged
        Try
            Dim NOMBREARTICULO As String

            NOMBREARTICULO =
ARTÍCULO.ARTICULOTableAdapter.NOMBREARTICULO(IDARTICULOTextBox.Te
xt)

            ARTICULOButton.Text = NOMBREARTICULO
        Catch ex As Exception
            ARTICULOButton.Text = "ARTÍCULO"
        End Try
    End Sub
End Class
```

ARTICULO

```
Public Class ARTÍCULO

    Private Sub ARTICULOBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
ARTICULOBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.ARTICULOBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_D
E_GRADODataSet)

    End Sub

    Private Sub ARTÍCULOvb_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.ARTICULO' Puede
moverla o quitarla según sea necesario.

Me.ARTICULOTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GR
ADODataSet.ARTICULO)

    End Sub
```

```
    Private Sub DESCRIPCIONARTICULOTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
DESCRIPCIONARTICULOTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(DESCRIPCIONLabel,
DESCRIPCIONARTICULOTextBox)
    End Sub



    Private Sub NOMBREARTICULOTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
NOMBREARTICULOTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(NOMBRELabel,
NOMBREARTICULOTextBox)
    End Sub

    Private Sub IDLUGARTextBox_TextChanged(sender As System.Object, e As
System.EventArgs) Handles IDLUGARTextBox.TextChanged
        Try
            Dim NOMBRELUGAR As String

            NOMBRELUGAR =
LUGAR.LUGARTableAdapter.NOMBRELUGAR(IDLUGARTextBox.Text)

            LUGARButton.Text = NOMBRELUGAR
        Catch ex As Exception
            LUGARButton.Text = "LUGAR"
        End Try


    End Sub

    Private Sub LUGARButton_MouseHover(sender As System.Object, e As
System.EventArgs) Handles LUGARButton.MouseHover
        Try
            Dim DESCRIPCIONLUGAR As String

            DESCRIPCIONLUGAR =
LUGAR.LUGARTableAdapter.DESCRIPCIONLUGAR(IDLUGARTextBox.Text)
            DESCRIPCIONLUGAR = controlUsuario.añadirenter(DESCRIPCIONLUGAR)
            LUGARLabel.Text = DESCRIPCIONLUGAR
        Catch ex As Exception

        End Try

    End Sub

    Private Sub LUGARButton_MouseLeave(sender As System.Object, e As
System.EventArgs) Handles LUGARButton.MouseLeave
        LUGARLabel.Text = ""
```

```
    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlusuariocampoobligatorio(LUGAROBLLabel,
IDLUGARTextBox, NOMBREOBLLabel, NOMBREARTICULOTextBox,
CANTIDADOBLLabel, CANTIDADARTICULOTextBox) Then
            controlUsuario.controlusuarionononull(DESCRIPCIONARTICULOTextBox)
            ARTICULOBindingNavigatorSaveItem.PerformClick()
            MsgBox("ARTÍULO GUARDADO") 'GUARDA LOS DATOS
            Me.Dispose()
            Try
                VERARTÍCULO.NOMBREToolStripTextBox.Text = "1"
                VERARTÍCULO.NOMBREToolStripTextBox.Text = "" 'SE USA PARA
RECARGAR LA VISTA DE ARTÍCULOS
            Catch ex As Exception
                'EN CASO DE SER INGRESO DIRECTO NO RECARGA NADA
            End Try
        Else
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        End If
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("SÍ ELIMINA EL ARTÍCULO """ +
NOMBREARTICULOTextBox.Text + """, TAMBIÉN SE ELIMINARÁN LAS
DONACIONES RELACIONADAS CON ESTE. ¿DESEA CONTINUAR?",
"Notificación", MessageBoxButtons.OKCancel) = DialogResult.OK) Then
            BindingNavigatorDeleteItem.PerformClick()
            ARTICULOBindingNavigatorSaveItem.PerformClick()
            MsgBox("LUGAR ELIMINADO")
            Me.Dispose()
            Try
                VERARTÍCULO.NOMBREToolStripTextBox.Text = "1"
                VERARTÍCULO.NOMBREToolStripTextBox.Text = "" 'SE USA PARA
RECARGAR LA VISTA DE ARTÍCULOS
            Catch ex As Exception

            End Try

        End If
    End Sub

    Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub
```

```
    Private Sub LUGARButton_Click(sender As System.Object, e As
System.EventArgs) Handles LUGARButton.Click
        VERLUGAR.Show()
        VERLUGAR.BUSCARLUGARDataGridView.Columns(1).Visible = True
    End Sub

    Private Sub CANTIDADARTICULOTextBox_Leave(sender As System.Object, e
As System.EventArgs) Handles CANTIDADARTICULOTextBox.Leave
        Try
            Convert.ToInt32(CANTIDADARTICULOTextBox.Text)
            CANTIDADLabel.ForeColor = Color.Gray
            CANTIDADLabel.Text = ""
        Catch ex As Exception
            CANTIDADARTICULOTextBox.Text = 0
            CANTIDADLabel.ForeColor = Color.Red
            CANTIDADLabel.Text = "Ingrese una cantidad válida"
        End Try

    End Sub
End Class
```

CAMA

```
Public Class CAMA

    Private Sub CAMABindingNavigatorSaveItem_Click(sender As System.Object,
e As System.EventArgs) Handles CAMABindingNavigatorSaveItem.Click
        Me.Validate()
        Me.CAMABindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.FUDIS_BASEDataSet)

    End Sub

    Private Sub CAMA_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUDIS_BASEDataSet.CAMA' Puede moverla o quitarla según sea necesario.
        Me.CAMATableAdapter.Fill(Me.FUDIS_BASEDataSet.CAMA)

    End Sub

    Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub

    Private Sub CODIGOCAMATextBox_TextChanged(sender As System.Object, e
As System.EventArgs) Handles CODIGOCAMATextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(CODIGOLabel,
CODIGOCAMATextBox)
```

```
    End Sub

    Private Sub DESCRIPCIONCAMATextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
DESCRIPCIONCAMATextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(DESCRIPCIONLabel,
DESCRIPCIONCAMATextBox)
    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If IDHABITACIONTextBox.Text = "" Or IDTIPOCAMATextBox.Text = "" Or
CODIGOCAMATextBox.Text = "" Then
            If IDHABITACIONTextBox.Text = "" Then
                HABITACIONOBLLabel.Text = "Campo Obligatorio"
            Else
                HABITACIONOBLLabel.Text = "*"
            End If
            If IDTIPOCAMATextBox.Text = "" Then
                TIPOCAMAOBLLabel.Text = "Campo Obligatorio"
            Else
                TIPOCAMAOBLLabel.Text = "*"
            End If
            If CODIGOCAMATextBox.Text = "" Then
                CODIGOOBLLabel.Text = "Campo Obligatorio"
            Else
                CODIGOOBLLabel.Text = "*"
            End If
            MsgBox("Campos obligatorios incompletos")
        Else
            CAMABindingNavigatorSaveItem.PerformClick()
            MsgBox("Cama guardada")
            Me.Dispose()
            Try
                VERCAMA.CODIGOToolStripTextBox.Text = "1"
                VERCAMA.CODIGOToolStripTextBox.Text = ""
                VERCAMADISPONIBLE.CODIGOToolStripTextBox.Text = "1"
                VERCAMADISPONIBLE.CODIGOToolStripTextBox.Text = ""
            Catch ex As Exception

            End Try

        End If
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("¿Desea eliminar la cama " +
CODIGOCAMATextBox.Text + "?", "Notificación",
MessageBoxButtons.OKCancel) = DialogResult.OK) Then
```

```
            BindingNavigatorDeleteItem.PerformClick()
            CAMABindingNavigatorSaveItem.PerformClick()
            MsgBox("Cama eliminada")
            Me.Dispose()
            Try
                VERCAMA.CODIGOToolStripTextBox.Text = "1"
                VERCAMA.CODIGOToolStripTextBox.Text = ""
                VERCAMADISPONIBLE.CODIGOToolStripTextBox.Text = "1"
                VERCAMADISPONIBLE.CODIGOToolStripTextBox.Text = ""
            Catch ex As Exception

            End Try
        End If
    End Sub

    Private Sub HABITACIONButton_Click(sender As System.Object, e As
System.EventArgs) Handles HABITACIONButton.Click
        VERHABITACION.Show()

VERHABITACION.BUSCARHABITACIONDataGridView.Columns("AGREGAR").Visi
ble = True
    End Sub

    Private Sub TIPOCAMAButton_Click(sender As System.Object, e As
System.EventArgs) Handles TIPOCAMAButton.Click
        VERTIPOCAMA.Show()

VERTIPOCAMA.BUSCARTIPOCAMADataGridView.Columns("AGREGAR").Visible
= True
    End Sub

    Private Sub IDHABITACIONTextBox_TextChanged(sender As System.Object, e
As System.EventArgs) Handles IDHABITACIONTextBox.TextChanged
        Try
            HABITACIONButton.Text =
HABITACION.HABITACIONTableAdapter.CODIGOHABITACION(IDHABITACIONTe
xtBox.Text)

        Catch
            HABITACIONButton.Text = "HABITACIÓN"
        End Try
    End Sub

    Private Sub HABITACIONButton_MouseEnter(sender As System.Object, e As
System.EventArgs) Handles HABITACIONButton.MouseEnter
        Try
            HABITACIONLabel.Text =
HABITACION.HABITACIONTableAdapter.DESCRIPCIONHABITACION(IDHABITACI
ONTextBox.Text)
```

```
        Catch
            HABITACIONLabel.Text = ""
        End Try
    End Sub

    Private Sub HABITACIONButton_MouseLeave(sender As System.Object, e As
System.EventArgs) Handles HABITACIONButton.MouseLeave
        Try
            HABITACIONLabel.Text = ""

        Catch
            HABITACIONLabel.Text = ""
        End Try
    End Sub

    Private Sub IDTIPOCAMATextBox_TextChanged(sender As System.Object, e
As System.EventArgs) Handles IDTIPOCAMATextBox.TextChanged
        Try
            TIPOCAMAButton.Text =
TIPOCAMA.TIPOCAMATableAdapter.NOMBRETIPOCAMA(IDTIPOCAMATextBox
.Text)

        Catch
            TIPOCAMAButton.Text = "TIPO CAMA"
        End Try
    End Sub

    Private Sub TIPOCAMAButton_MouseEnter(sender As System.Object, e As
System.EventArgs) Handles TIPOCAMAButton.MouseEnter
        Try
            TIPOCAMALabel.Text =
TIPOCAMA.TIPOCAMATableAdapter.DESCRIPCIONTIPOCAMA(IDTIPOCAMATex
tBox.Text)

        Catch
            TIPOCAMALabel.Text = ""
        End Try
    End Sub

    Private Sub TIPOCAMAButton_MouseLeave(sender As System.Object, e As
System.EventArgs) Handles TIPOCAMAButton.MouseLeave
        Try
            TIPOCAMALabel.Text = ""

        Catch
            TIPOCAMALabel.Text = ""
        End Try
    End Sub
End Class
```

DIARIO

```
Public Class DIARIO

    Private Sub FillToolStripButton_Click(sender As System.Object, e As
System.EventArgs) Handles FillToolStripButton.Click, MyBase.Load,
FECHAToolStripTextBox.TextChanged
        Try

Me.DIARIOTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRA
DODataSet.DIARIO, New System.Nullable(Of
Date)(CType(FECHAToolStripTextBox.Text, Date)))

        Catch ex As System.Exception

        End Try

    End Sub

    Private Sub FECHADateTimePicker_ValueChanged(sender As
System.Object, e As System.EventArgs) Handles
FECHADateTimePicker.ValueChanged
        FECHAToolStripTextBox.Text = FECHADateTimePicker.Value
    End Sub

    Private Sub Imprimir_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Imprimir.Click
        ImprimirDGV.Print_DataGridView(DIARIODataGridView)
    End Sub
End Class
```

DONACIÓNARTÍCULO

```
Public Class DONACIÓNARTÍCULO

    Private Sub DONACIONBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
DONACIONBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.DONACIONBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_D
E_GRADODataSet)

    End Sub

    Private Sub DONACIÓNARTÍCULO_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.DONACION' Puede
moverla o quitarla según sea necesario.
```

```
Me.DONACIONTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.DONACION)

    End Sub

    Private Sub CANCELARButton_Click(sender As System.Object, e As System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("SÍ ELIMINA LA DONACIÓN DE """ + DONADORButton.Text + """ A " + BENEFICIADODONACIONTextBox.Text + ", TAMBIÉN SE ELIMINARÁN LOS ARTÍCULOS RELACIONADOS CON LA DONACIÓN. ¿DESEA CONTINUAR?", "Notificación", MessageBoxButtons.OKCancel) = DialogResult.OK) Then
            Dim beneficiado, iddonacion As String
            iddonacion =IDDONACIONTextBox.Text
            beneficiado = Me.DONACIONTableAdapter.BENEFICIADODONACION(iddonacion)

DONACIONTableAdapter.BORRARARTICULOS(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.DONACION, IDDONACIONTextBox.Text, beneficiado)


DONACIONTableAdapter.BUSCARDONACIONID(FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.DONACION, iddonacion)
            BindingNavigatorDeleteItem.PerformClick()




            DONACIONBindingNavigatorSaveItem.PerformClick()
            MsgBox("DONACIÓN ELIMINADA")
            Me.Dispose()
            Try
                VERDONACIÓNARTÍCULO.NOMBREDONADORToolStripTextBox.Text = "1"
                VERDONACIÓNARTÍCULO.NOMBREDONADORToolStripTextBox.Text = ""  'SE USA PARA RECARGAR LA VISTA
            Catch ex As Exception

            End Try

        End If
    End Sub
```

```
    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlusuariocampoobligatorio(DONADOROBLLabel,
IDDONADORTextBox, BENEFICIADOOBLLabel,
BENEFICIADODONACIONTextBox) Then
            controlUsuario.controlusuarionononull(DESCRIPCIONDONACIONTextBox)

            DONACIONBindingNavigatorSaveItem.PerformClick()
            MsgBox("DONACION GUARDADA") 'GUARDA LOS DATOS

            Me.Dispose()
            Try
                VERDONACIÓNARTÍCULO.NOMBREDONADORToolStripTextBox.Text =
"1"
                VERDONACIÓNARTÍCULO.NOMBREDONADORToolStripTextBox.Text =
"" 'SE USA PARA RECARGAR LA VISTA DE DONACIONES
            Catch ex As Exception
                'EN CASO DE SER INGRESO DIRECTO NO RECARGA NADA
            End Try
            If Me.Text = "CREAR DONACIÓN ARTÍCULO" Then
                VERARTÍCULODONACIÓN.Show()
                VERARTÍCULODONACIÓN.IDDONACIONToolStripTextBox.Text =
IDDONACIONTextBox.Text
            End If
        Else
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        End If
    End Sub

    Private Sub IDDONADORTextBox_TextChanged(sender As System.Object, e
As System.EventArgs) Handles IDDONADORTextBox.TextChanged
        Try
            Dim NOMBREDONADOR As String

            NOMBREDONADOR =
DONADOR.DONADORTableAdapter.NOMBRECOMPLETODONADOR(IDDONAD
ORTextBox.Text)
            DONADORButton.Text = NOMBREDONADOR
        Catch ex As Exception
            DONADORButton.Text = "DONADOR"
        End Try
    End Sub

    Private Sub DONADORButton_MouseHover(sender As System.Object, e As
System.EventArgs) Handles DONADORButton.MouseHover
        Try
            Dim EMPRESAREPRESENTANTEDONADOR As String
```

```
        EMPRESAREPRESENTANTEDONADOR =
DONADOR.DONADORTableAdapter.EMPRESAREPRESENTANTEDONADOR(IDD
ONADORTextBox.Text)
        EMPRESAREPRESENTANTEDONADOR =
controlUsuario.añadirenter(EMPRESAREPRESENTANTEDONADOR)
        DONADORLabel.Text = EMPRESAREPRESENTANTEDONADOR
      Catch ex As Exception
        DONADORLabel.Text = ""
      End Try
    End Sub

    Private Sub DONADORButton_MouseLeave(sender As System.Object, e As
System.EventArgs) Handles DONADORButton.MouseLeave
      DONADORLabel.Text = ""
    End Sub

    Private Sub DONADORButton_Click(sender As System.Object, e As
System.EventArgs) Handles DONADORButton.Click
      VERDONADOR.Show()
      VERDONADOR.BUSCARDONADORDataGridView.Columns(1).Visible = True
    End Sub

    Private Sub BENEFICIADODONACIONTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
BENEFICIADODONACIONTextBox.TextChanged
      controlUsuario.controlusuariocontarcaracteres(BENEFICIADOLabel,
BENEFICIADODONACIONTextBox)
    End Sub

    Private Sub DESCRIPCIONDONACIONTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
DESCRIPCIONDONACIONTextBox.TextChanged
      controlUsuario.controlusuariocontarcaracteres(DESCRIPCIONLabel,
DESCRIPCIONDONACIONTextBox)
    End Sub


End Class
```

DONACIÓNDINERO

```
Public Class DONACIONDINERO

    Private Sub DONACIONBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
DONACIONBindingNavigatorSaveItem.Click
      Me.Validate()
      Me.DONACIONBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_D
E_GRADODataSet)
```

```
    End Sub

    Private Sub DONACIONDINERO_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.DONACION' Puede
moverla o quitarla según sea necesario.

Me.DONACIONTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_G
RADODataSet.DONACION)

    End Sub

    Private Sub DONADORButton_Click(sender As System.Object, e As
System.EventArgs) Handles DONADORButton.Click
        VERDONADOR.Show()
        VERDONADOR.BUSCARDONADORDataGridView.Columns(1).Visible = True
    End Sub

    Private Sub IDDONADORTextBox_TextChanged(sender As System.Object, e
As System.EventArgs) Handles IDDONADORTextBox.TextChanged
        Try
            Dim NOMBREDONADOR As String

            NOMBREDONADOR =
DONADOR.DONADORTableAdapter.NOMBRECOMPLETODONADOR(IDDONAD
ORTextBox.Text)
            DONADORButton.Text = NOMBREDONADOR
        Catch ex As Exception
            DONADORButton.Text = "DONADOR"
        End Try
    End Sub

    Private Sub DONADORButton_MouseHover(sender As System.Object, e As
System.EventArgs) Handles DONADORButton.MouseHover
        Try
            Dim EMPRESAREPRESENTANTEDONADOR As String

            EMPRESAREPRESENTANTEDONADOR =
DONADOR.DONADORTableAdapter.EMPRESAREPRESENTANTEDONADOR(IDD
ONADORTextBox.Text)
            EMPRESAREPRESENTANTEDONADOR =
controlUsuario.añadirenter(EMPRESAREPRESENTANTEDONADOR)
            DONADORLabel.Text = EMPRESAREPRESENTANTEDONADOR
        Catch ex As Exception
            DONADORLabel.Text = ""
        End Try
    End Sub
```

```
    Private Sub DONADORButton_MouseLeave(sender As System.Object, e As
System.EventArgs) Handles DONADORButton.MouseLeave
        DONADORLabel.Text = ""
    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlusuariocampoobligatorio(DONADOROBLLabel,
IDDONADORTextBox, BENEFICIADOOBLLabel,
BENEFICIADODONACIONTextBox, MONTOOBLLabel, MONTOTextBox) Then
            controlUsuario.controlusuarionononull(DESCRIPCIONDONACIONTextBox)
            If BENEFICIADODONACIONTextBox.Text <> "FUDIS" Then
                MONTOTextBox.Text = (Convert.ToDouble(MONTOTextBox.Text) * -
1).ToString()
            End If

            If IDTRANSACCIONTextBox.Text = "" Then
                TRANSACCION.Show()
                TRANSACCION.BindingNavigatorAddNewItem.PerformClick()
                TRANSACCION.TRA_IDTRANSACCIONTextBox.Text = -1
                TRANSACCION.IDPACIENTETextBox.Text = -1
                TRANSACCION.IDCAMATextBox.Text = -1
                TRANSACCION.IDACOMPANANTETextBox.Text = -1
                TRANSACCION.CAM_IDCAMATextBox.Text = -2
                TRANSACCION.IDTIPOTRANSACCIONTextBox.Text = 3
                TRANSACCION.COSTODIARIOTRANSACCIONTextBox.Text = 0
                TRANSACCION.MONTOTRANSACCIONTextBox.Text =
MONTOTextBox.Text
                TRANSACCION.FECHATRANSACCIONDateTimePicker.Value = Today
                TRANSACCION.FECHAINICIOTRANSACCIONDateTimePicker.Value =
Today
                TRANSACCION.FECHAFINTRANSACCIONDateTimePicker.Value = Today
                IDTRANSACCIONTextBox.Text =
TRANSACCION.IDTRANSACCIONTextBox.Text

TRANSACCION.TRANSACCIONBindingNavigatorSaveItem.PerformClick()
                TRANSACCION.Dispose()

            Else
                TRANSACCION.Show()

TRANSACCION.TRANSACCIONTableAdapter.BUSCARTRANSACCIONID(TRANSAC
CION.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.TRANSACCION,
IDTRANSACCIONTextBox.Text)
                TRANSACCION.MONTOTRANSACCIONTextBox.Text =
MONTOTextBox.Text

TRANSACCION.TRANSACCIONBindingNavigatorSaveItem.PerformClick()
                TRANSACCION.Dispose()
            End If
```

```
            DONACIONBindingNavigatorSaveItem.PerformClick()
            MsgBox("DONACION GUARDADA") 'GUARDA LOS DATOS

            Me.Dispose()
            Try
                VERDONACIONDINERO.NOMBREDONADORToolStripTextBox.Text =
"1"
                VERDONACIONDINERO.NOMBREDONADORToolStripTextBox.Text = ""
'SE USA PARA RECARGAR LA VISTA DE DONACIONES
            Catch ex As Exception
                'EN CASO DE SER INGRESO DIRECTO NO RECARGA NADA
            End Try

        Else
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        End If
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("SÍ ELIMINA LA DONACIÓN DE """ +
DONADORButton.Text + """ A " + BENEFICIADODONACIONTextBox.Text + ",
TAMBIÉN SE ELIMINARÁN LOS ARTÍCULOS RELACIONADOS CON LA DONACIÓN.
¿DESEA CONTINUAR?", "Notificación", MessageBoxButtons.OKCancel) =
DialogResult.OK) Then
            Dim idtransaccion As String
            idtransaccion = IDTRANSACCIONTextBox.Text

            BindingNavigatorDeleteItem.PerformClick()
            DONACIONBindingNavigatorSaveItem.PerformClick()


            TRANSACCION.Show()

TRANSACCION.TRANSACCIONTableAdapter.BUSCARTRANSACCIONID(TRANSAC
CION.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.TRANSACCION,
idtransaccion)
            TRANSACCION.BindingNavigatorDeleteItem.PerformClick()
            TRANSACCION.TRANSACCIONBindingNavigatorSaveItem.PerformClick()
            TRANSACCION.Dispose()

            MsgBox("DONACIÓN ELIMINADA")


            Me.Dispose()
            Try
                VERDONACIONDINERO.NOMBREDONADORToolStripTextBox.Text =
"1"
```

```
            VERDONACIONDINERO.NOMBREDONADORToolStripTextBox.Text = ""
'SE USA PARA RECARGAR LA VISTA
        Catch ex As Exception

        End Try

    End If
  End Sub

  Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
    Me.Dispose()
  End Sub
End Class
```

DONADOR

```
Public Class DONADOR

  Private Sub DONADORBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
DONADORBindingNavigatorSaveItem.Click
    Me.Validate()
    Me.DONADORBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_D
E_GRADODataSet)

  End Sub

  Private Sub DONADOR_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
    'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.DONADOR' Puede
moverla o quitarla según sea necesario.

Me.DONADORTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GR
ADODataSet.DONADOR)

  End Sub

  Private Sub CIDONADORTextBox_TextChanged(sender As System.Object, e
As System.EventArgs)
    controlUsuario.controlusuariocontarcaracteres(CILabel,
CIDONADORTextBox)
  End Sub

  Private Sub NOMBREDONADORTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
NOMBREDONADORTextBox.TextChanged
```

```
        controlUsuario.controlusuariocontarcaracteres(NOMBRELabel,
NOMBREDONADORTextBox)
    End Sub

    Private Sub APELLIDODONADORTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
APELLIDODONADORTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(APELLIDOLabel,
APELLIDODONADORTextBox)
    End Sub

    Private Sub TELEFONODONADORTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
TELEFONODONADORTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(TELÉFONOLabel,
TELEFONODONADORTextBox)
    End Sub

    Private Sub
EMPRESAREPRESENTANTEDONADORTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
EMPRESAREPRESENTANTEDONADORTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(EMPRESALabel,
EMPRESAREPRESENTANTEDONADORTextBox)
    End Sub

    Private Sub EMAILDONADORTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
EMAILDONADORTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(EMAILLabel,
EMAILDONADORTextBox)
    End Sub

    Private Sub OBSEVACIONESDONADORTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
OBSEVACIONESDONADORTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(OBSERVACIONESLabel,
OBSEVACIONESDONADORTextBox)
    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlusuariocampoobligatorio(NOMBREOBLLabel,
NOMBREDONADORTextBox, APELLIDOOBLLabel, APELLIDODONADORTextBox)
Then
            controlUsuario.controlusuarionononull(CIDONADORTextBox,
TELEFONODONADORTextBox, EMPRESAREPRESENTANTEDONADORTextBox,
EMAILDONADORTextBox, OBSEVACIONESDONADORTextBox)
            DONADORBindingNavigatorSaveItem.PerformClick()
            MsgBox("DONADOR GUARDADO") 'GUARDA LOS DATOS
```

```
            Me.Dispose()
            Try
                VERDONADOR.NOMBREDONADORToolStripTextBox.Text = "1"
                VERDONADOR.NOMBREDONADORToolStripTextBox.Text = "" 'SE USA
PARA RECARGAR LA VISTA DE DONADORES
            Catch ex As Exception
                'EN CASO DE SER INGRESO DIRECTO NO RECARGA NADA
            End Try
        Else
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        End If
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("SÍ ELIMINA EL DONADOR """ +
NOMBREDONADORTextBox.Text + " " + APELLIDODONADORTextBox.Text + """,
TAMBIÉN SE ELIMINARÁN LAS DONACIONES RELACIONADAS CON ESTE.
¿DESEA CONTINUAR?", "Notificación", MessageBoxButtons.OKCancel) =
DialogResult.OK) Then
            BindingNavigatorDeleteItem.PerformClick()
            DONADORBindingNavigatorSaveItem.PerformClick()
            MsgBox("DONADOR ELIMINADO")
            Me.Dispose()
            Try
                VERDONADOR.NOMBREDONADORToolStripTextBox.Text = "1"
                VERDONADOR.NOMBREDONADORToolStripTextBox.Text = "" 'SE USA
PARA RECARGAR LA VISTA DE DONADORES
            Catch ex As Exception

            End Try

        End If
    End Sub

    Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub



    Private Sub CIDONADORTextBox_KeyPress(sender As System.Object, e As
System.Windows.Forms.KeyPressEventArgs) Handles
CIDONADORTextBox.KeyPress
        If Char.IsNumber(e.KeyChar) Then
            e.Handled = False
        ElseIf Char.IsLetter(e.KeyChar) Then
            e.Handled = True
        ElseIf Char.IsControl(e.KeyChar) Then
```

```
            e.Handled = False
        ElseIf Char.IsSeparator(e.KeyChar) Then
            e.Handled = True
        End If
    End Sub

    Private Sub CIDONADORTextBox_TextChanged_1(sender As System.Object,
e As System.EventArgs) Handles CIDONADORTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(CILabel,
CIDONADORTextBox)
    End Sub
End Class
```

HABITACIÓN

```
Public Class HABITACION

    Private Sub HABITACIONBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
HABITACIONBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.HABITACIONBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.FUDIS_BASEDataSet)

    End Sub

    Private Sub HABITACION_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUDIS_BASEDataSet.HABITACION' Puede moverla o quitarla según sea
necesario.
        Me.HABITACIONTableAdapter.Fill(Me.FUDIS_BASEDataSet.HABITACION)

    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If PISOHABITACIONTextBox.Text = "" Or CODIGOHABITACIONTextBox.Text
= "" Then
            If PISOHABITACIONTextBox.Text = "" Then
                PISOOBLLabel.Text = "Campo Obligatorio"
            Else
                PISOOBLLabel.Text = "*"
            End If
            If CODIGOHABITACIONTextBox.Text = "" Then
                CODIGOOBLLabel.Text = "Campo Obligatorio"
            Else
                CODIGOOBLLabel.Text = "*"
            End If
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        Else
```

```
            If DESCRIPCIONHABITACIONTextBox.Text = "" Then
                DESCRIPCIONHABITACIONTextBox.Text = " "
            End If
            HABITACIONBindingNavigatorSaveItem.PerformClick()
            MsgBox("HABITACIÓN GUARDADA")
            Me.Dispose()
            VERACOMPANANTE.CIToolStripTextBox.Text = "1"
            VERACOMPANANTE.CIToolStripTextBox.Text = ""
        End If
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("¿Desea eliminar la habitación " +
CODIGOHABITACIONTextBox.Text + "?", "Notificación",
MessageBoxButtons.OKCancel) = DialogResult.OK) Then
            BindingNavigatorDeleteItem.PerformClick()
            HABITACIONBindingNavigatorSaveItem.PerformClick()
            MsgBox("HABITACIÓN ELIMINADA")
            Me.Dispose()
            VERHABITACION.CODIGOHABITACIONToolStripTextBox.Text = "1"
            VERHABITACION.CODIGOHABITACIONToolStripTextBox.Text = ""
        End If
    End Sub

    Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub

    Private Sub DESCRIPCIONHABITACIONTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
DESCRIPCIONHABITACIONTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(DESCRIPCIONLabel,
DESCRIPCIONHABITACIONTextBox)
    End Sub

    Private Sub CODIGOHABITACIONTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
CODIGOHABITACIONTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(CODIGOLabel,
CODIGOHABITACIONTextBox)
    End Sub

    Private Sub PISOHABITACIONTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
PISOHABITACIONTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(PISOLabel,
PISOHABITACIONTextBox)
    End Sub
```

```
End Class
```

LUGAR

```
Public Class LUGAR

    Private Sub LUGARBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
LUGARBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.LUGARBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_D
E_GRADODataSet)

    End Sub

    Private Sub LUGAR_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.LUGAR' Puede moverla
o quitarla según sea necesario.

Me.LUGARTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRAD
ODataSet.LUGAR)

    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlusuariocampoobligatorio(NOMBREOBLLabel,
NOMBRELUGARTextBox) Then
            controlUsuario.controlusuarioononull(DESCRIPCIONLUGARTextBox)
            LUGARBindingNavigatorSaveItem.PerformClick()
            MsgBox("LUGAR GUARDADO") 'GUARDA LOS DATOS
            Me.Dispose()
            Try
                VERLUGAR.NOMBRELUGARToolStripTextBox.Text = "1"
                VERLUGAR.NOMBRELUGARToolStripTextBox.Text = "" 'SE USA PARA
RECARGAR LA VISTA DE LUGARES
            Catch ex As Exception
                'EN CASO DE SER INGRESO DIRECTO NO RECARGA NADA
            End Try
        Else
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        End If
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
```

```
        If (MessageBox.Show("SÍ ELIMINA EL LUGAR """ +
NOMBRELUGARTextBox.Text + """, TAMBIÉN SE ELIMINARÁN LOS ARTÍCULOS
Y LAS DONACIONES RELACIONADAS CON ESTE. ¿DESEA CONTINUAR?",
"Notificación", MessageBoxButtons.OKCancel) = DialogResult.OK) Then
            BindingNavigatorDeleteItem.PerformClick()
            LUGARBindingNavigatorSaveItem.PerformClick()
            MsgBox("LUGAR ELIMINADO")
            Me.Dispose()
            Try
                VERLUGAR.NOMBRELUGARToolStripTextBox.Text = "1"
                VERLUGAR.NOMBRELUGARToolStripTextBox.Text = "" 'SE USA PARA
RECARGAR LA VISTA DE LUGARES
            Catch ex As Exception

            End Try

        End If
    End Sub

    Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub

    Private Sub NOMBRELUGARTextBox_TextChanged(sender As System.Object,
e As System.EventArgs) Handles NOMBRELUGARTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(NOMBRELabel,
NOMBRELUGARTextBox)
    End Sub

    Private Sub DESCRIPCIONLUGARTextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
DESCRIPCIONLUGARTextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(DESCRIPCIONLabel,
DESCRIPCIONLUGARTextBox)
    End Sub
End Class
```

OPCIÓNIMPRIMIR

```
Public Class OpcioneImpresion
    Public Sub New(ByVal availableFields As List(Of String))

        InitializeComponent()

        For Each field As String In availableFields
            chklst.Items.Add(field, True)
        Next
    End Sub
```

```
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        Me.DialogResult = Windows.Forms.DialogResult.Cancel
        Me.Close()
    End Sub

    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click

        '  Using dlg As CoolPrintPreviewDialog = New CoolPrintPreviewDialog
        'dlg.Document = PrintDGV1.PrintDoc
        ' dlg.ShowDialog(Me)
        Me.DialogResult = Windows.Forms.DialogResult.OK
        Me.Close()
        ' End Using
    End Sub


    Private Sub nuevo_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        rdoAllRows.Checked = True
        chkFitToPageWidth.Checked = True
    End Sub
    Public Function GetSelectedColumns() As List(Of String)
        Dim lst As New List(Of String)
        For Each item As Object In chklst.CheckedItems
            lst.Add(item.ToString)
        Next
        Return lst
    End Function

    Public ReadOnly Property PrintTitle() As String
        Get
            Return txtTitle.Text
        End Get
    End Property

    Public ReadOnly Property PrintAllRows() As Boolean
        Get
            Return rdoAllRows.Checked
        End Get
    End Property

    Public ReadOnly Property FitToPageWidth() As Boolean
        Get
            Return chkFitToPageWidth.Checked
        End Get
    End Property
End Class
```

PACIENTE

```
Public Class PACIENTE

    Private Sub PACIENTEBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
PACIENTEBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PACIENTEBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.FUDIS_BASEDataSet)

    End Sub

    Private Sub PACIENTE_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUDIS_BASEDataSet.PACIENTE' Puede moverla o quitarla según sea necesario.
        Me.PACIENTETableAdapter.Fill(Me.FUDIS_BASEDataSet.PACIENTE)

    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlusuariocampoobligatorio(NOMBREOBLLabel,
NOMBREPACIENTETextBox, APELLIDOOBLLabel, APELLIDOPACIENTETextBox)
Then
            controlUsuario.controlusuarionononull(CIPACIENTETextBox,
TELEFONOPACIENTETextBox, HISTORIACLINICAPACIENTETextBox,
NACIONALIDADPACIENTETextBox, OBSERVACIONESPACIENTETextBox,
PROVINCIAPACIENTETextBox, CIUDADPACIENTETextBox,
DOMICILIOPACIENTETextBox, HOSPITALTRATANTEPACIENTETextBox,
DIAGNOSTICOPACIENTETextBox, TRATAMIENTOPACIENTETextBox)
            PACIENTEBindingNavigatorSaveItem.PerformClick()
            MsgBox("PACIENTE GUARDADO") 'GUARDA LOS DATOS
            Me.Dispose()
            Try
                VERPACIENTE.NOMBRETOolStripTextBox.Text = "1"
                VERPACIENTE.NOMBRETOolStripTextBox.Text = "" 'SE USA PARA
RECARGAR LA VISTA
                VERPACIENTESDISPONIBLES.NOMBRETOolStripTextBox.Text = "1"
                VERPACIENTESDISPONIBLES.NOMBRETOolStripTextBox.Text = ""
            Catch ex As Exception


            End Try
        Else
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        End If
    End Sub
```

```
    Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("¿Desea eliminar el paciente " +
NOMBREPACIENTETextBox.Text + " " + APELLIDOPACIENTETextBox.Text + "?",
"Notificación", MessageBoxButtons.OKCancel) = DialogResult.OK) Then
            BindingNavigatorDeleteItem.PerformClick()
            PACIENTEBindingNavigatorSaveItem.PerformClick()
            MsgBox("PACIENTE ELIMINADO")
            Me.Dispose()
            Try
                VERPACIENTE.CIToolStripTextBox.Text = "1"
                VERPACIENTE.CIToolStripTextBox.Text = ""
                VERPACIENTESDISPONIBLES.NOMBREToolStripTextBox.Text = "1"
                VERPACIENTESDISPONIBLES.NOMBREToolStripTextBox.Text = ""
            Catch

            End Try
        End If
    End Sub


    Private Sub CIPACIENTETextBox_TextChanged(sender As System.Object, e As
System.EventArgs) Handles CIPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(CILabel,
CIPACIENTETextBox)
    End Sub

    Private Sub NOMBREPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
NOMBREPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(NOMBRELabel,
NOMBREPACIENTETextBox)
    End Sub

    Private Sub APELLIDOPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
APELLIDOPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(APELLIDOLabel,
APELLIDOPACIENTETextBox)
    End Sub

    Private Sub TELEFONOPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
TELEFONOPACIENTETextBox.TextChanged
```

```
        controlUsuario.controlusuariocontarcaracteres(TELEFONOLabel,
TELEFONOPACIENTETextBox)
    End Sub

    Private Sub HISTORIACLINICAPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
HISTORIACLINICAPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(HISTORIALabel,
HISTORIACLINICAPACIENTETextBox)
    End Sub

    Private Sub NACIONALIDADPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
NACIONALIDADPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(NACIONALIDADLabel,
NACIONALIDADPACIENTETextBox)
    End Sub

    Private Sub PROVINCIAPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
PROVINCIAPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(PROVINCIALabel,
PROVINCIAPACIENTETextBox)
    End Sub

    Private Sub CIUDADPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
CIUDADPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(CIUDADLabel,
CIUDADPACIENTETextBox)
    End Sub

    Private Sub DOMICILIOPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
DOMICILIOPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(DOMICILIOLabel,
DOMICILIOPACIENTETextBox)
    End Sub

    Private Sub HOSPITALTRATANTEPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
HOSPITALTRATANTEPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(HOSPITALLabel,
HOSPITALTRATANTEPACIENTETextBox)
    End Sub

    Private Sub DIAGNOSTICOPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
DIAGNOSTICOPACIENTETextBox.TextChanged
```

```
        controlUsuario.controlusuariocontarcaracteres(DIAGNOSTICOLabel,
DIAGNOSTICOPACIENTETextBox)
    End Sub

    Private Sub TRATAMIENTOPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
TRATAMIENTOPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(TRATAMIENTOLabel,
TRATAMIENTOPACIENTETextBox)
    End Sub

    Private Sub OBSERVACIONESPACIENTETextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
OBSERVACIONESPACIENTETextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(OBSERVACIONESLabel,
OBSERVACIONESPACIENTETextBox)
    End Sub
End Class
```

PAGO

```
Public Class PAGO

    Private Sub TRANSACCIONBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
TRANSACCIONBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.TRANSACCIONBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_D
E_GRADODataSet)

    End Sub

    Private Sub PAGO_Load(sender As System.Object, e As System.EventArgs)
Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.TRANSACCION' Puede
moverla o quitarla según sea necesario.

Me.TRANSACCIONTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE
_GRADODataSet.TRANSACCION)

    End Sub

    Private Sub PACIENTEButton_Click(sender As System.Object, e As
System.EventArgs) Handles PACIENTEButton.Click
        VERPACIENTESDISPONIBLES.Show()
```

```
VERPACIENTESDISPONIBLES.BUSCARPACIENTEDISPONIBLEDataGridView.Colu
mns(1).Visible = True
    End Sub

    Private Sub CAMAPAButton_Click(sender As System.Object, e As
System.EventArgs) Handles CAMAPAButton.Click
        VERCAMADISPONIBLE.Show()
        VERCAMADISPONIBLE.Text = "CAMAS DISPONIBLES PARA PACIENTE"
        If IDCAMATextBox.Text = "" And CAM_IDCAMATextBox.Text = "" Then
            VERCAMADISPONIBLE.IDCAMAToolStripTextBox.Text = "-1"
        ElseIf CAM_IDCAMATextBox.Text <> "" Then
            VERCAMADISPONIBLE.IDCAMAToolStripTextBox.Text =
CAM_IDCAMATextBox.Text
        End If

VERCAMADISPONIBLE.BUSCARCAMADISPONIBLEDataGridView.Columns(1).Visi
ble = True
    End Sub

    Private Sub CAMAACButton_Click(sender As System.Object, e As
System.EventArgs) Handles CAMAACButton.Click
        VERCAMADISPONIBLE.Show()
        VERCAMADISPONIBLE.Text = "CAMAS DISPONIBLES PARA
ACOMPAÑANTE"
        If IDCAMATextBox.Text = "" And CAM_IDCAMATextBox.Text = "" Then
            VERCAMADISPONIBLE.IDCAMAToolStripTextBox.Text = "-1"
        ElseIf IDCAMATextBox.Text <> "" Then
            VERCAMADISPONIBLE.IDCAMAToolStripTextBox.Text =
IDCAMATextBox.Text
        End If

VERCAMADISPONIBLE.BUSCARCAMADISPONIBLEDataGridView.Columns(1).Visi
ble = True
    End Sub

    Private Sub ACOMPANANTEButton_Click(sender As System.Object, e As
System.EventArgs) Handles ACOMPANANTEButton.Click
        VERACOMPANANTE.Show()

VERACOMPANANTE.BUSCARACOMPANANTEDataGridView.Columns(1).Visible
= True
    End Sub

    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlusuariocampoobligatorio(PACIENTEOBLLabel,
IDPACIENTETextBox, CAMAPAOBLLabel, CAM_IDCAMATextBox,
COSTOOBLLabel, COSTODIARIOTRANSACCIONTextBox, MONTOOBLLabel,
MONTOTRANSACCIONTextBox) Then
```

```
            TRANSACCIONBindingNavigatorSaveItem.PerformClick()

            If FECHAINICIOTRANSACCIONDateTimePicker.Value >= Today And Today
<= FECHAFINTRANSACCIONDateTimePicker.Value Then
                CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(False,
IDCAMATextBox.Text)
                CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(False,
CAM_IDCAMATextBox.Text)
            End If
            MsgBox("PAGO GUARDADO") 'GUARDA LOS DATOS
            Me.Dispose()
            Try
                VERPAGO.NOMBREACOMPANANTEToolStripTextBox.Text = "1"
                VERPAGO.NOMBREACOMPANANTEToolStripTextBox.Text = "" 'SE USA
PARA RECARGAR LA VISTA

            Catch ex As Exception


            End Try
        Else
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        End If
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("¿Desea eliminar el pago?", "Notificación",
MessageBoxButtons.OKCancel) = DialogResult.OK) Then

            If FECHAINICIOTRANSACCIONDateTimePicker.Value >= Today And Today
<= FECHAFINTRANSACCIONDateTimePicker.Value Then
                CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(True,
IDCAMATextBox.Text)
                CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(True,
CAM_IDCAMATextBox.Text)
            End If

            BindingNavigatorDeleteItem.PerformClick()
            TRANSACCIONBindingNavigatorSaveItem.PerformClick()
            MsgBox("PAGO ELIMINADO")
            Me.Dispose()
            Try
                VERPAGO.CIPACIENTEToolStripTextBox.Text = "1"
                VERPAGO.CIPACIENTEToolStripTextBox.Text = ""
            Catch

            End Try
        End If
```

```
    End Sub

    Private Sub IDCAMATextBox_TextChanged(sender As System.Object, e As
System.EventArgs) Handles IDCAMATextBox.TextChanged
        Try
            If FECHAINICIOTRANSACCIONDateTimePicker.Value <= Today And Today
<= FECHAFINTRANSACCIONDateTimePicker.Value Then
                CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(True,
CPAUX.Text)
                CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(False,
IDCAMATextBox.Text)
                CPAUX.Text = IDCAMATextBox.Text

            End If
        Catch ex As Exception

        End Try
    End Sub

    Private Sub CAM_IDCAMATextBox_TextChanged(sender As System.Object, e
As System.EventArgs) Handles CAM_IDCAMATextBox.TextChanged
        Try
            If FECHAINICIOTRANSACCIONDateTimePicker.Value <= Today
And Today <= FECHAFINTRANSACCIONDateTimePicker.Value Then

CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(True, CAAUX.Text)

CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(False,
CAM_IDCAMATextBox.Text)
                CAAUX.Text = CAM_IDCAMATextBox.Text
            End If
        Catch ex As Exception

        End Try
    End Sub
End Class
```

TIPO CAMA

```
Public Class TIPOCAMA
 Private Sub TIPOCAMABindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
TIPOCAMABindingNavigatorSaveItem.Click
        Me.Validate()
        Me.TIPOCAMABindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.FUDIS_BASEDataSet)

    End Sub

    Private Sub TIPOCAMA_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUDIS_BASEDataSet.TIPOCAMA' Puede moverla o quitarla según sea
necesario.
        Me.TIPOCAMATableAdapter.Fill(Me.FUDIS_BASEDataSet.TIPOCAMA)
```

```
    End Sub

    Private Sub NOMBRETIPOCAMATextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
NOMBRETIPOCAMATextBox.TextChanged
        controlUsuario.controlusuariocontarcaracteres(NOMBRELabel,
NOMBRETIPOCAMATextBox)
    End Sub

    Private Sub DESCRIPCIONTIPOCAMATextBox_TextChanged(sender As
System.Object, e As System.EventArgs) Handles
DESCRIPCIONTIPOCAMATextBox.TextChanged

controlUsuario.controlusuariocontarcaracteres(DESCRIPCIONLabel,
DESCRIPCIONTIPOCAMATextBox)
    End Sub


    Private Sub GUARDARButton_Click(sender As System.Object, e As
System.EventArgs) Handles GUARDARButton.Click
        If NOMBRETIPOCAMATextBox.Text = "" Then
            If NOMBRETIPOCAMATextBox.Text = "" Then
                NOMBREOBLLabel.Text = "Campo Obligatorio"
            Else
                NOMBREOBLLabel.Text = "*"
            End If

            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS")
        Else
            If DESCRIPCIONTIPOCAMATextBox.Text = "" Then
                DESCRIPCIONTIPOCAMATextBox.Text = " "
            End If
            TIPOCAMABindingNavigatorSaveItem.PerformClick()
            MsgBox("TIPO DE CAMA GUARDADA")
            Me.Dispose()
            VERTIPOCAMA.NOMBRETIPOCAMAToolStripTextBox.Text = "1"
            VERTIPOCAMA.NOMBRETIPOCAMAToolStripTextBox.Text = ""
        End If
    End Sub

    Private Sub ELIMINARButton_Click(sender As System.Object, e As
System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("¿Desea eliminar el tipo de cama " +
NOMBRETIPOCAMATextBox.Text + "?", "Notificación",
MessageBoxButtons.OKCancel) = DialogResult.OK) Then
            BindingNavigatorDeleteItem.PerformClick()
            TIPOCAMABindingNavigatorSaveItem.PerformClick()
            MsgBox("TIPO DE CAMA ELIMINADA")
            Me.Dispose()
            VERTIPOCAMA.NOMBRETIPOCAMAToolStripTextBox.Text = "1"
            VERTIPOCAMA.NOMBRETIPOCAMAToolStripTextBox.Text = ""
        End If
    End Sub

    Private Sub CANCELARButton_Click(sender As System.Object, e As
System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub
End Class
```

TRANSACCIÓN

```
Public Class TRANSACCION

    Private Sub TRANSACCIONBindingNavigatorSaveItem_Click(sender As
System.Object, e As System.EventArgs) Handles
TRANSACCIONBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.TRANSACCIONBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_
GRADODataSet)

    End Sub

    Private Sub TRANSACCION_Load(sender As System.Object, e As
System.EventArgs) Handles MyBase.Load
        'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.TRANSACCION' Puede
moverla o quitarla según sea necesario.

Me.TRANSACCIONTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_G
RADODataSet.TRANSACCION)

    End Sub
End Class
```

VER ACOMPAÑANTE

```
Public Class VERACOMPANANTE

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
NOMBREToolStripTextBox.TextChanged, CIToolStripTextBox.TextChanged,
APELLIDOToolStripTextBox.TextChanged, MyBase.Load
        Try

Me.BUSCARACOMPANANTETableAdapter.Fill(Me.FUDIS_BASEDataSet.BUSCARAC
OMPANANTE, CIToolStripTextBox.Text, NOMBREToolStripTextBox.Text,
APELLIDOToolStripTextBox.Text)
        Catch ex As System.Exception

        End Try

    End Sub

    Private Sub CREARToolStripButton1_Click(sender As
System.Object, e As System.EventArgs) Handles
CREARToolStripButton1.Click
        ACOMPANANTE.Show()
        ACOMPANANTE.BindingNavigatorAddNewItem.PerformClick()
        ACOMPANANTE.ELIMINARACOMPANANTEButton.Enabled = False
    End Sub

    Private Sub
BUSCARACOMPANANTEDataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARACOMPANANTEDataGridView.CellContentClick
        Try
            If e.ColumnIndex = 5 Then
                ACOMPANANTE.Show()
```

```
ACOMPANANTE.ACOMPANANTETableAdapter.BUSCARACOMPANANTEID(ACOMPANANTE
.FUDIS_BASEDataSet.ACOMPANANTE,
BUSCARACOMPANANTEDataGridView.Rows(e.RowIndex).Cells(0).Value())
                ACOMPANANTE.ELIMINARACOMPANANTEButton.Enabled =
True
            ElseIf e.ColumnIndex = 1 Then
                PAGO.IDACOMPANANTETextBox.Text =
BUSCARACOMPANANTEDataGridView.Rows(e.RowIndex).Cells(0).Value()
                MsgBox("ACOMPAÑANTE AÑADIDO")
                Try
                    VERCAMA.CODIGOToolStripTextBox.Text = "1"
                    VERCAMA.CODIGOToolStripTextBox.Text = ""
                    VERCAMADISPONIBLE.CODIGOToolStripTextBox.Text =
"1"
                    VERCAMADISPONIBLE.CODIGOToolStripTextBox.Text =
""
                Catch ex As Exception

                End Try
                Me.Dispose()
            End If

        Catch
            ACOMPANANTE.Dispose()
            'MsgBox("No se puede ordenar por este campo")
        End Try


    End Sub

    Private Sub
BUSCARACOMPANANTEBindingNavigator_RefreshItems(sender As
System.Object, e As System.EventArgs) Handles
BUSCARACOMPANANTEBindingNavigator.RefreshItems

    End Sub

    Private Sub CIToolStripTextBox_KeyPress(sender As
System.Object, e As System.Windows.Forms.KeyPressEventArgs) Handles
CIToolStripTextBox.KeyPress
        If Char.IsNumber(e.KeyChar) Then
            e.Handled = False
        ElseIf Char.IsLetter(e.KeyChar) Then
            e.Handled = True
        ElseIf Char.IsControl(e.KeyChar) Then
            e.Handled = False
        ElseIf Char.IsSeparator(e.KeyChar) Then
            e.Handled = True
        End If
    End Sub
End Class
```

VER ARTICULO

```
Public Class VERARTÍCULO

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
```

```
NOMBRETOOLSTRIPTEXTBOX.TEXTCHANGED, MYBASE.LOAD,
LUGARTOOLSTRIPTEXTBOX.TEXTCHANGED,
DESCRIPCIONTOOLSTRIPTEXTBOX.TEXTCHANGED
        Try

Me.BUSCARARTICULOTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_D
E_GRADODataSet.BUSCARARTICULO, NOMBRETOOLSTRIPTEXTBOX.TEXT,
LUGARTOOLSTRIPTEXTBOX.TEXT, DESCRIPCIONTOOLSTRIPTEXTBOX.TEXT)
        Catch ex As System.Exception

        End Try

    End Sub

    Private Sub BUSCARARTICULODataGridView_CellContentClick(sender
As System.Object, e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles
BUSCARARTICULODataGridView.CellContentClick
        Try
            Dim IDARTICULO, CANTIDADARTICULO As Int32

            IDARTICULO =
BUSCARARTICULODataGridView.Rows(e.RowIndex).Cells(0).Value()
            CANTIDADARTICULO =
BUSCARARTICULODataGridView.Rows(e.RowIndex).Cells(3).Value()

            If e.ColumnIndex = 6 Then
                ARTÍCULO.Show()
                ARTÍCULO.ELIMINARButton.Enabled = True

ARTÍCULO.ARTICULOTableAdapter.BUSCARARTICULOID(ARTÍCULO.FUNDACIÓN_F
UDIS_DISERTACIÓN_DE_GRADODataSet.ARTICULO, IDARTICULO)
                If Me.Text = "VER ARTÍCULO DONACIÓN " Then
                    ARTÍCULO.CANTIDADARTICULOTextBox.Enabled =
False
                End If

controlUsuario.quitarblanco(ARTÍCULO.DESCRIPCIONARTICULOTextBox)
            ElseIf e.ColumnIndex = 1 Then
                AGREGARARTÍCULODONACION.IDARTICULOTextBox.Text =
IDARTICULO.ToString()

AGREGARARTÍCULODONACION.CANTIDADARTICULONUEVOLabel.Text =
CANTIDADARTICULO
                MsgBox("ARTÍCULO AGREGADO")
                Me.Dispose()
            End If
        Catch
            ARTÍCULO.Dispose()
        End Try
    End Sub

    Private Sub ToolStripButtonCREAR_Click(sender As System.Object,
e As System.EventArgs) Handles ToolStripButtonCREAR.Click
        ARTÍCULO.Show()
        ARTÍCULO.BindingNavigatorAddNewItem.PerformClick()
        If Me.Text = "VER ARTÍCULO DONACIÓN " Then
            ARTÍCULO.CANTIDADARTICULOTextBox.Text = 0
            ARTÍCULO.CANTIDADARTICULOTextBox.Enabled = False
        End If
```

```
    End Sub

    Private Sub Imprimir_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Imprimir.Click
        ImprimirDGV.Print_DataGridView(BUSCARARTICULODataGridView)
    End Sub
End Class
```

VERARTÍCULODONACIÓN

```
Public Class VERARTÍCULODONACIÓN

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
NOMBRETextBox.TextChanged,
LUGARToolStripTextBox.TextChanged,
IDDONACIONToolStripTextBox.TextChanged,
DESCRIPCIONToolStripTextBox.TextChanged
        Try

Me.BUSCARARTICULODONACIONTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISER
TACIÓN_DE_GRADODataSet.BUSCARARTICULODONACION,
NOMBRETextBox.Text, LUGARTextBox.Text,
DESCRIPCIONTextBox.Text, New System.Nullable(Of
Long)(CType(IDDONACIONTextBox.Text, Long)))
        Catch ex As System.Exception
            System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try

    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        AGREGARARTÍCULODONACION.Show()

AGREGARARTÍCULODONACION.BindingNavigatorAddNewItem.PerformClick()
        AGREGARARTÍCULODONACION.IDDONACIONTextBox.Text =
IDDONACIONTextBox.Text
        AGREGARARTÍCULODONACION.TARTÍCULOLabel.Text = "NUEVO"


    End Sub

    Private Sub
BUSCARARTICULODONACIONDataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARARTICULODONACIONDataGridView.CellContentClick
        Try
            Dim IDPUEDETENER, IDARTICULO As Int32
            IDPUEDETENER =
BUSCARARTICULODONACIONDataGridView.Rows(e.RowIndex).Cells(0).Value(
)
            'IDARTICULO =
BUSCARARTICULODONACIONDataGridView.Rows(e.RowIndex).Cells(1).Value(
)
            If e.ColumnIndex = 5 Then
                AGREGARARTÍCULODONACION.Show()
                AGREGARARTÍCULODONACION.ELIMINARButton.Enabled =
True
```

```
AGREGARARTÍCULODONACION.PUEDE_TENERTableAdapter.BUSCARPUEDETENERID(
AGREGARARTÍCULODONACION.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet
.PUEDE_TENER, IDPUEDETENER)

'AGREGARARTICULODONACION.IDARTICULOANTERIORLabel.Text =
AGREGARARTICULODONACION.IDARTICULOTextBox.Text

'AGREGARARTICULODONACION.CANTIDADARTICULOVIEJOLabel.Text =
ARTÍCULO.ARTICULOTableAdapter.CANTIDADARTICULO(IDARTICULO)


            End If

        Catch
            AGREGARARTÍCULODONACION.Dispose()
        End Try
    End Sub

End Class
```

VER CAMA

```
Public Class VERCAMA

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click
        Try

Me.BUSCARCAMATableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GR
ADODataSet.BUSCARCAMA, CODIGOToolStripTextBox.Text,
DESCRIPCIONToolStripTextBox.Text,
NOMBRETIPOCAMAToolStripTextBox.Text,
PISOHABITACIONToolStripTextBox.Text,
CODIGOHABITACIONToolStripTextBox.Text)
        Catch ex As System.Exception
            System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try

    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton1.Click
        ImprimirDGV.Print_DataGridView(BUSCARCAMADataGridView)
    End Sub


End Class
```

VERCAMADISPONIBLE

```
Public Class VERCAMADISPONIBLE

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
PISOHABITACIONToolStripTextBox.TextChanged,
NOMBRETIPOCAMAToolStripTextBox.TextChanged, MyBase.Load,
IDCAMAToolStripTextBox.TextChanged,
DESCRIPCIONToolStripTextBox.TextChanged,
```

```
CODIGOToolStripTextBox.TextChanged,
CODIGOHABITACIONToolStripTextBox.TextChanged
        Try

Me.BUSCARCAMADISPONIBLETableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTA
CIÓN_DE_GRADODataSet.BUSCARCAMADISPONIBLE,
CODIGOToolStripTextBox.Text, DESCRIPCIONToolStripTextBox.Text,
NOMBRETIPOCAMAToolStripTextBox.Text,
PISOHABITACIONToolStripTextBox.Text,
CODIGOHABITACIONToolStripTextBox.Text, New System.Nullable(Of
Short)(CType(IDCAMAToolStripTextBox.Text, Short)))
        Catch ex As System.Exception

        End Try

    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        CAMA.Show()
        CAMA.BindingNavigatorAddNewItem.PerformClick()
    End Sub

    Private Sub
BUSCARCAMADISPONIBLEDataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARCAMADISPONIBLEDataGridView.CellContentClick
        Try
            If e.ColumnIndex = 9 Then
                CAMA.Show()

CAMA.CAMATableAdapter.BUSCARCAMAID(CAMA.FUDIS_BASEDataSet.CAMA,
BUSCARCAMADISPONIBLEDataGridView.Rows(e.RowIndex).Cells(0).Value())
                CAMA.ELIMINARButton.Enabled = True
            ElseIf e.ColumnIndex = 1 Then
                If Me.Text = "CAMAS DISPONIBLES PARA PACIENTE" Then
                    PAGO.IDCAMATextBox.Text =
BUSCARCAMADISPONIBLEDataGridView.Rows(e.RowIndex).Cells(0).Value()
                    MsgBox("CAMA PACIENTE AÑADIDA")
                    Me.Dispose()
                ElseIf Me.Text = "CAMAS DISPONIBLES PARA
ACOMPAÑANTE" Then
                    PAGO.CAM_IDCAMATextBox.Text =
BUSCARCAMADISPONIBLEDataGridView.Rows(e.RowIndex).Cells(0).Value()
                    MsgBox("CAMA ACOMPAÑANTE AÑADIDA")
                    Me.Dispose()
                End If
            End If
        Catch
            CAMA.Dispose()
        End Try
    End Sub
End Class
```

VERDEVOLUCIONPAGO

```
Public Class VERDEVOLUCIONPAGO

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
NOMBREPACIENTEToolStripTextBox.TextChanged,
```

```
NOMBREACOMPANANTEToolStripTextBox.TextChanged, MyBase.Load,
FECHAINICIOToolStripTextBox.TextChanged,
FECHAFINToolStripTextBox.TextChanged,
CIPACIENTEToolStripTextBox.TextChanged,
CIACOMPANANTEToolStripTextBox.TextChanged,
APELLIDOPACIENTEToolStripTextBox.TextChanged,
APELLIDOACOMPANANTEToolStripTextBox.TextChanged
        Try

Me.BUSCARDEVOLUCIONPAGOTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTA
CIÓN_DE_GRADODataSet.BUSCARDEVOLUCIONPAGO,
CIPACIENTEToolStripTextBox.Text,
NOMBREPACIENTEToolStripTextBox.Text,
APELLIDOPACIENTEToolStripTextBox.Text,
CIACOMPANANTEToolStripTextBox.Text,
NOMBREACOMPANANTEToolStripTextBox.Text,
APELLIDOACOMPANANTEToolStripTextBox.Text, New System.Nullable(Of
Date)(CType(FECHAINICIOToolStripTextBox.Text, Date)), New
System.Nullable(Of Date)(CType(FECHAFINToolStripTextBox.Text,
Date)))
        Catch ex As System.Exception

        End Try

    End Sub
End Class
```

VERDONACIÓNARTÍCULO

```
Public Class VERDONACIÓNARTÍCULO

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
OBSEVACIONESDONADORToolStripTextBox.TextChanged,
NOMBREDONADORToolStripTextBox.TextChanged, MyBase.Load,
FECHAINICIOToolStripTextBox.TextChanged,
FECHAFINToolStripTextBox.TextChanged,
EMPRESAREPRESENTANTEDONADORToolStripTextBox.TextChanged,
DESCRIPCIONDONACIONToolStripTextBox.TextChanged,
CIDONADORToolStripTextBox.TextChanged,
BENEFICIADODONACIONToolStripTextBox.TextChanged,
APELLIDODONADORToolStripTextBox.TextChanged
        Try

Me.BUSCARDONACIONARTICULOTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISER
TACIÓN_DE_GRADODataSet.BUSCARDONACIONARTICULO,
CIDONADORToolStripTextBox.Text, NOMBREDONADORToolStripTextBox.Text,
APELLIDODONADORToolStripTextBox.Text,
EMPRESAREPRESENTANTEDONADORToolStripTextBox.Text,
OBSEVACIONESDONADORToolStripTextBox.Text,
BENEFICIADODONACIONToolStripTextBox.Text,
DESCRIPCIONDONACIONToolStripTextBox.Text, New System.Nullable(Of
Date)(CType(FECHAINICIOToolStripTextBox.Text, Date)), New
System.Nullable(Of Date)(CType(FECHAFINToolStripTextBox.Text,
Date)))
        Catch ex As System.Exception

        End Try

    End Sub
```

```
    Private Sub
BUSCARDONACIONARTICULODataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARDONACIONARTICULODataGridView.CellContentClick
        Try
            Dim IDDONACION As Int32
            IDDONACION =
BUSCARDONACIONARTICULODataGridView.Rows(e.RowIndex).Cells(0).Value(
)
            If e.ColumnIndex = 9 Then
                DONACIÓNARTÍCULO.Show()
                DONACIÓNARTÍCULO.ELIMINARButton.Enabled = True

DONACIÓNARTÍCULO.DONACIONTableAdapter.BUSCARDONACIONID(DONACIÓNARTÍ
CULO.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.DONACION,
IDDONACION)
                DONACIÓNARTÍCULO.Text = "EDITAR DONACIÓN ARTÍCULO"

DONACIÓNARTÍCULO.FECHADONACIONDateTimePicker.Enabled = True

controlUsuario.quitarblanco(DONACIÓNARTÍCULO.DESCRIPCIONDONACIONTex
tBox)
            ElseIf e.ColumnIndex = 10 Then
                VERARTÍCULODONACIÓN.Show()
                VERARTÍCULODONACIÓN.IDDONACIONToolStripTextBox.Text
= IDDONACION
            End If
        Catch
            DONACIÓNARTÍCULO.Dispose()
            VERARTÍCULODONACIÓN.Dispose()
        End Try
    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        DONACIÓNARTÍCULO.Show()
        DONACIÓNARTÍCULO.Text = "CREAR DONACIÓN ARTÍCULO"
        DONACIÓNARTÍCULO.BindingNavigatorAddNewItem.PerformClick()
        DONACIÓNARTÍCULO.IDTRANSACCIONTextBox.Text = "-1"
        DONACIÓNARTÍCULO.IDTIPODONACIONTextBox.Text = "2"
        DONACIÓNARTÍCULO.FECHADONACIONDateTimePicker.Value = Today
    End Sub

    Private Sub Imprimir_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Imprimir.Click

ImprimirDGV.Print_DataGridView(BUSCARDONACIONARTICULODataGridView)
    End Sub
End Class
```

VERDONACIÓNDINERO

```
Public Class VERDONACIONDINERO

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
OBSEVACIONESDONADORToolStripTextBox.TextChanged,
NOMBREDONADORToolStripTextBox.TextChanged, MyBase.Load,
FECHAINICIOToolStripTextBox.TextChanged,
FECHAFINToolStripTextBox.TextChanged,
```

```
EMPRESAREPRESENTANTEDONADORToolStripTextBox.TextChanged,
DESCRIPCIONDONACIONToolStripTextBox.TextChanged,
CIDONADORToolStripTextBox.TextChanged,
BENEFICIADODONACIONToolStripTextBox.TextChanged,
APELLIDODONADORToolStripTextBox.TextChanged
        Try

Me.BUSCARDONACIONDINEROTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTA
CIÓN_DE_GRADODataSet.BUSCARDONACIONDINERO,
CIDONADORToolStripTextBox.Text, NOMBREDONADORToolStripTextBox.Text,
APELLIDODONADORToolStripTextBox.Text,
EMPRESAREPRESENTANTEDONADORToolStripTextBox.Text,
OBSEVACIONESDONADORToolStripTextBox.Text,
BENEFICIADODONACIONToolStripTextBox.Text,
DESCRIPCIONDONACIONToolStripTextBox.Text, New System.Nullable(Of
Date)(CType(FECHAINICIOToolStripTextBox.Text, Date)), New
System.Nullable(Of Date)(CType(FECHAFINToolStripTextBox.Text,
Date)))
        Catch ex As System.Exception

        End Try

    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        DONACIONDINERO.Show()
        DONACIONDINERO.BindingNavigatorAddNewItem.PerformClick()
        DONACIONDINERO.Text = "NUEVA DONACIÓN DINERO"
        DONACIONDINERO.IDTIPODONACIONTextBox.Text = 1
        DONACIONDINERO.FECHADONACIONDateTimePicker.Value = Today
        DONACIONDINERO.MONTOTextBox.Text = ""
    End Sub

    Private Sub
BUSCARDONACIONDINERODataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARDONACIONDINERODataGridView.CellContentClick
        Try
            Dim IDDONACION As Int32
            IDDONACION =
BUSCARDONACIONDINERODataGridView.Rows(e.RowIndex).Cells(0).Value()
            If e.ColumnIndex = 11 Then
                DONACIONDINERO.Show()
                DONACIONDINERO.ELIMINARButton.Enabled = True

DONACIONDINERO.DONACIONTableAdapter.BUSCARDONACIONID(DONACIONDINERO
.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.DONACION, IDDONACION)
                DONACIONDINERO.Text = "EDITAR DONACIÓN DINERO"
                DONACIONDINERO.FECHADONACIONDateTimePicker.Enabled
= True
                DONACIONDINERO.MONTOTextBox.Text =
BUSCARDONACIONDINERODataGridView.Rows(e.RowIndex).Cells(10).Value()
                DONACIONDINERO.IDTRANSACCIONTextBox.Text =
BUSCARDONACIONDINERODataGridView.Rows(e.RowIndex).Cells(1).Value()

controlUsuario.quitarblanco(DONACIONDINERO.DESCRIPCIONDONACIONTextB
ox)

            End If
```

```
        Catch
            DONACIONDINERO.Dispose()

        End Try
    End Sub

    Private Sub Imprimir_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Imprimir.Click

ImprimirDGV.Print_DataGridView(BUSCARDONACIONDINERODataGridView)
    End Sub
End Class
```

VERDONADOR

```
Public Class VERDONADOR

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
NOMBREDONADORToolStripTextBox.TextChanged, MyBase.Load,
EMPRESAREPRESENTANTEDONADORToolStripTextBox.TextChanged,
DESCRIPCIONDONADORToolStripTextBox.TextChanged,
CIDONADORToolStripTextBox.TextChanged,
APELLIDODONADORToolStripTextBox.TextChanged
        Try

Me.BUSCARDONADORTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE
_GRADODataSet.BUSCARDONADOR, CIDONADORToolStripTextBox.Text,
NOMBREDONADORToolStripTextBox.Text,
APELLIDODONADORToolStripTextBox.Text,
EMPRESAREPRESENTANTEDONADORToolStripTextBox.Text,
DESCRIPCIONDONADORToolStripTextBox.Text)
        Catch ex As System.Exception

        End Try

    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        DONADOR.Show()
        DONADOR.BindingNavigatorAddNewItem.PerformClick()

    End Sub

    Private Sub BUSCARDONADORDataGridView_CellContentClick(sender
As System.Object, e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles
BUSCARDONADORDataGridView.CellContentClick
        Try
            Dim IDDONADOR As Int32
            IDDONADOR =
BUSCARDONADORDataGridView.Rows(e.RowIndex).Cells(0).Value()
            If e.ColumnIndex = 9 Then
                DONADOR.Show()
                DONADOR.ELIMINARButton.Enabled = True

DONADOR.DONADORTableAdapter.BUSCARDONADORID(DONADOR.FUNDACIÓN_FUDIS
_DISERTACIÓN_DE_GRADODataSet.DONADOR, IDDONADOR)
```

```
controlUsuario.quitarblanco(DONADOR.CIDONADORTextBox,
DONADOR.TELEFONODONADORTextBox,
DONADOR.EMPRESAREPRESENTANTEDONADORTextBox,
DONADOR.EMAILDONADORTextBox, DONADOR.OBSEVACIONESDONADORTextBox)
            ElseIf e.ColumnIndex = 1 Then
                Try
                    DONACIÓNARTÍCULO.IDDONADORTextBox.Text =
IDDONADOR.ToString()
                    DONACIONDINERO.IDDONADORTextBox.Text =
IDDONADOR.ToString()
                Catch ex As Exception

                End Try
                MsgBox("DONADOR AGREGADO")
                Me.Dispose()
            End If
        Catch
            ''DONADOR.Dispose()
        End Try
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton1.Click
        ImprimirDGV.Print_DataGridView(BUSCARDONADORDataGridView)
    End Sub
End Class
```

VER HABITACION

```
Public Class VERHABITACION

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
PISOHABITACIONToolStripTextBox.TextChanged, MyBase.Load,
DESCRIPCIONHABITACIONToolStripTextBox.TextChanged,
CODIGOHABITACIONToolStripTextBox.TextChanged
        Try

Me.BUSCARHABITACIONTableAdapter.Fill(Me.FUDIS_BASEDataSet.BUSCARHAB
ITACION, PISOHABITACIONToolStripTextBox.Text,
CODIGOHABITACIONToolStripTextBox.Text,
DESCRIPCIONHABITACIONToolStripTextBox.Text)
        Catch ex As System.Exception
            System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try

    End Sub

    Private Sub
BUSCARHABITACIONDataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARHABITACIONDataGridView.CellContentClick
        Try
            If e.ColumnIndex = 2 Then
                HABITACION.Show()

HABITACION.HABITACIONTableAdapter.BUSCARHABITACIONID(HABITACION.FUD
IS_BASEDataSet.HABITACION,
BUSCARHABITACIONDataGridView.Rows(e.RowIndex).Cells(0).Value())
```

```
                If
BUSCARHABITACIONDataGridView.Columns("AGREGAR").Visible = True Then
                    HABITACION.ELIMINARButton.Visible = False
                End If
            ElseIf e.ColumnIndex = 1 Then
                CAMA.IDHABITACIONTextBox.Text =
BUSCARHABITACIONDataGridView.Rows(e.RowIndex).Cells(0).Value()
                MsgBox("HABITACIÓN AGREGADA")
                Me.Dispose()
            End If
        Catch
            HABITACION.Dispose()
            'MsgBox("No se puede ordenar por este campo")
        End Try
    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        HABITACION.Show()
        HABITACION.BindingNavigatorAddNewItem.PerformClick()
        HABITACION.ELIMINARButton.Visible = False
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton1.Click

ImprimirDGV.Print_DataGridView(BUSCARHABITACIONDataGridView)
    End Sub
End Class
```

VER LUGAR

```
Public Class VERLUGAR

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
NOMBRELUGARToolStripTextBox.TextChanged, MyBase.Load,
DESCRIPCIONLUGARToolStripTextBox.TextChanged
        Try

Me.BUSCARLUGARTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_G
RADODataSet.BUSCARLUGAR, NOMBRELUGARToolStripTextBox.Text,
DESCRIPCIONLUGARToolStripTextBox.Text)
        Catch ex As System.Exception

        End Try

    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        LUGAR.Show()
        LUGAR.BindingNavigatorAddNewItem.PerformClick()
    End Sub

    Private Sub BUSCARLUGARDataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARLUGARDataGridView.CellContentClick
        Try
            Dim IDLUGAR As Int32
```

```
            IDLUGAR =
BUSCARLUGARDataGridView.Rows(e.RowIndex).Cells(0).Value()
            If e.ColumnIndex = 4 Then
                LUGAR.Show()
                LUGAR.ELIMINARButton.Enabled = True

LUGAR.LUGARTableAdapter.BUSCARLUGARID(LUGAR.FUNDACIÓN_FUDIS_DISERTA
CIÓN_DE_GRADODataSet.LUGAR, IDLUGAR)

controlUsuario.quitarblanco(LUGAR.DESCRIPCIONLUGARTextBox)
            ElseIf e.ColumnIndex = 1 Then
                ARTÍCULO.IDLUGARTextBox.Text = IDLUGAR.ToString()
                MsgBox("LUGAR AGREGADO")
                Me.Dispose()
            End If
        Catch
            LUGAR.Dispose()
        End Try
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton1.Click
        ImprimirDGV.Print_DataGridView(BUSCARLUGARDataGridView)
    End Sub
End Class
```

VER PACIENTE

```
Public Class VERPACIENTE

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
TRATAMIENTOToolStripTextBox.TextChanged,
PROVINCIAToolStripTextBox.TextChanged,
NOMBREToolStripTextBox.TextChanged,
NACIONALIDADToolStripTextBox.TextChanged,
HOSPITALTRATANTEToolStripTextBox.TextChanged,
EDADINIToolStripTextBox.TextChanged,
EDADFINToolStripTextBox.TextChanged,
DIAGNOSTICOToolStripTextBox.TextChanged,
CIUDADToolStripTextBox.TextChanged, CIToolStripTextBox.TextChanged,
APELLIDOToolStripTextBox.TextChanged, MyBase.Load
        Try

Me.BUSCARPACIENTETableAdapter.Fill(Me.FUDIS_BASEDataSet.BUSCARPACIE
NTE, CIToolStripTextBox.Text, NOMBREToolStripTextBox.Text,
APELLIDOToolStripTextBox.Text, NACIONALIDADToolStripTextBox.Text,
PROVINCIAToolStripTextBox.Text, CIUDADToolStripTextBox.Text,
HOSPITALTRATANTEToolStripTextBox.Text,
DIAGNOSTICOToolStripTextBox.Text, TRATAMIENTOToolStripTextBox.Text,
New System.Nullable(Of Integer)(CType(EDADINIToolStripTextBox.Text,
Integer)), New System.Nullable(Of
Integer)(CType(EDADFINToolStripTextBox.Text, Integer)))
        Catch ex As System.Exception
            ''System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try

    End Sub

    Private Sub BUSCARPACIENTEDataGridView_CellContentClick(sender
As System.Object, e As
```

```
System.Windows.Forms.DataGridViewCellEventArgs) Handles
BUSCARPACIENTEDataGridView.CellContentClick
        Try

            If e.ColumnIndex = 1 Then
                PACIENTE.Show()

PACIENTE.PACIENTETableAdapter.BUSCARPACIENTEID(PACIENTE.FUDIS_BASED
ataSet.PACIENTE,
BUSCARPACIENTEDataGridView.Rows(e.RowIndex).Cells(0).Value())
            End If
        Catch
            PACIENTE.Dispose()

        End Try

    End Sub

    Private Sub PACIENTE_Load(sender As System.Object, e As
System.EventArgs)

    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        PACIENTE.Show()
        PACIENTE.BindingNavigatorAddNewItem.PerformClick()
        PACIENTE.ELIMINARButton.Visible = False
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolStripButton1.Click
        ImprimirDGV.Print_DataGridView(BUSCARPACIENTEDataGridView)
    End Sub
End Class
```

VERPACIENTEDISPONIBLE

```
Public Class VERPACIENTESDISPONIBLES

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
TRATAMIENTOToolStripTextBox.TextChanged,
PROVINCIAToolStripTextBox.TextChanged,
NOMBREToolStripTextBox.TextChanged,
NACIONALIDADToolStripTextBox.TextChanged, MyBase.Load,
HOSPITALTRATANTEToolStripTextBox.TextChanged,
EDADINIToolStripTextBox.TextChanged,
EDADFINToolStripTextBox.TextChanged,
DIAGNOSTICOToolStripTextBox.TextChanged,
CIUDADToolStripTextBox.TextChanged, CIToolStripTextBox.TextChanged,
APELLIDOToolStripTextBox.TextChanged
        Try

Me.BUSCARPACIENTEDISPONIBLETableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DIS
ERTACIÓN_DE_GRADODataSet.BUSCARPACIENTEDISPONIBLE,
CIToolStripTextBox.Text, NOMBREToolStripTextBox.Text,
APELLIDOToolStripTextBox.Text, NACIONALIDADToolStripTextBox.Text,
PROVINCIAToolStripTextBox.Text, CIUDADToolStripTextBox.Text,
HOSPITALTRATANTEToolStripTextBox.Text,
```

```
DIAGNOSTICOToolStripTextBox.Text, TRATAMIENTOToolStripTextBox.Text,
New System.Nullable(Of Integer)(CType(EDADINIToolStripTextBox.Text,
Integer)), New System.Nullable(Of
Integer)(CType(EDADFINToolStripTextBox.Text, Integer)))
        Catch ex As System.Exception

        End Try

    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        PACIENTE.Show()
        PACIENTE.BindingNavigatorAddNewItem.PerformClick()
        PACIENTE.FECHANACIMIENTOPACIENTEDateTimePicker.Value =
Today
    End Sub

    Private Sub
BUSCARPACIENTEDISPONIBLEDataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARPACIENTEDISPONIBLEDataGridView.CellContentClick
        Try
            Dim IDPACIENTE As Long
            IDPACIENTE =
BUSCARPACIENTEDISPONIBLEDataGridView.Rows(e.RowIndex).Cells(0).Valu
e()
            If e.ColumnIndex = 15 Then
                PACIENTE.Show()

PACIENTE.PACIENTETableAdapter.BUSCARPACIENTEID(PACIENTE.FUDIS_BASED
ataSet.PACIENTE, IDPACIENTE)
                PACIENTE.ELIMINARButton.Enabled = True
            ElseIf e.ColumnIndex = 1 Then
                PAGO.IDPACIENTETextBox.Text = IDPACIENTE
                MsgBox("PACIENTE AGREGADO")
                Me.Dispose()
            End If
        Catch
            PACIENTE.Dispose()

        End Try
    End Sub
End Class
```

VER PAGO

```
Public Class VERPAGO

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
NOMBREPACIENTEToolStripTextBox.TextChanged,
NOMBREACOMPANANTEToolStripTextBox.TextChanged, MyBase.Load,
FECHAINICIOToolStripTextBox.TextChanged,
CIPACIENTEToolStripTextBox.TextChanged,
CIACOMPANANTEToolStripTextBox.TextChanged,
APELLIDOPACIENTEToolStripTextBox.TextChanged,
APELLIDOACOMPANANTEToolStripTextBox.TextChanged
        Try

Me.BUSCARPAGOTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GR
```

```
ADODataSet.BUSCARPAGO, CIPACIENTEToolStripTextBox.Text,
NOMBREPACIENTEToolStripTextBox.Text,
APELLIDOPACIENTEToolStripTextBox.Text,
CIACOMPANANTEToolStripTextBox.Text,
NOMBREACOMPANANTEToolStripTextBox.Text,
APELLIDOACOMPANANTEToolStripTextBox.Text, New System.Nullable(Of
Date)(CType(FECHAINICIOToolStripTextBox.Text, Date)), New
System.Nullable(Of Date)(CType(FECHAFINToolStripTextBox.Text,
Date)))
        Catch ex As System.Exception

        End Try

    End Sub


    Private Sub ToolStripButton1_Click(sender As System.Object, e
As System.EventArgs) Handles ToolStripButton1.Click
        PAGO.Show()
        PAGO.BindingNavigatorAddNewItem.PerformClick()
        PAGO.IDTIPOTRANSACCIONTextBox.Text = "1"
        PAGO.FECHATRANSACCIONDateTimePicker.Value = Today
        PAGO.FECHAINICIOTRANSACCIONDateTimePicker.Value = Today
        PAGO.FECHAFINTRANSACCIONDateTimePicker.Value = Today
        PAGO.TRA_IDTRANSACCIONTextBox.Text = "-1"
    End Sub

    Private Sub BUSCARPAGODataGridView_CellContentClick(sender As
System.Object, e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles BUSCARPAGODataGridView.CellContentClick
        Try
            If e.ColumnIndex = 15 Then
                PAGO.Show()

PAGO.TRANSACCIONTableAdapter.BUSCARTRANSACCIONID(PAGO.FUNDACIÓN_FUD
ISIS_DISERTACIÓN_DE_GRADODataSet.TRANSACCION,
BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(0).Value())
                PAGO.ELIMINARButton.Enabled = True
                PAGO.CPAUX.Text = PAGO.IDCAMATextBox.Text
                PAGO.CAAUX.Text = PAGO.CAM_IDCAMATextBox.Text
                PAGO.FECHATRANSACCIONDateTimePicker.Enabled = True

            ElseIf e.ColumnIndex = 1 Then
                Try
                    Dim tiempoTranscurrido As TimeSpan
                    Dim diasparahospedar, diashospedados As Int32
                    Dim montohospedado, montopagado, devolucion As
Double

                    tiempoTranscurrido =
BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(14).Value().Subtract(
BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(13).Value())
                    diasparahospedar = tiempoTranscurrido.Days + 1
                    tiempoTranscurrido =
Today.Subtract(BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(13).Va
lue())
                    diashospedados = tiempoTranscurrido.Days
                    If diashospedados >= 0 And
System.DateTime.Now.ToShortTimeString >= "12:00" Then
                        diashospedados = diashospedados + 1
```

```
                    ElseIf diashospedados < 0 Then
                        diashospedados = 0
                    End If

                    montohospedado =
BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(10).Value() *
diashospedados
                    montopagado =
BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(11).Value()
                    devolucion = montopagado - montohospedado
                    If devolucion >= 0 Then
                        MsgBox("La cantidad a devolver es: " +
devolucion.ToString() + " $")
                    Else
                        MsgBox("El paciente " +
BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(3).Value() + " " +
BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(4).Value() + "
todvaía debe " + (devolucion * -1).ToString() + " $")
                        devolucion = 0
                    End If

                    Dim TRA, PA, CA, AC, CAAC, TIPO, COSTO As
String

                    TRANSACCION.Show()

TRANSACCION.TRANSACCIONTableAdapter.BUSCARTRANSACCIONID(TRANSACCION
.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.TRANSACCION,
BUSCARPAGODataGridView.Rows(e.RowIndex).Cells(0).Value())
                    TRA = TRANSACCION.IDTRANSACCIONTextBox.Text
                    PA = TRANSACCION.IDPACIENTETextBox.Text
                    CA = TRANSACCION.IDCAMATextBox.Text
                    AC = TRANSACCION.IDACOMPANANTETextBox.Text
                    CAAC = TRANSACCION.CAM_IDCAMATextBox.Text
                    TIPO =
TRANSACCION.IDTIPOTRANSACCIONTextBox.Text
                    COSTO =
TRANSACCION.COSTODIARIOTRANSACCIONTextBox.Text


TRANSACCION.BindingNavigatorAddNewItem.PerformClick()
                    TRANSACCION.TRA_IDTRANSACCIONTextBox.Text = TRA
                    TRANSACCION.IDPACIENTETextBox.Text = PA
                    TRANSACCION.IDCAMATextBox.Text = -1
                    TRANSACCION.IDACOMPANANTETextBox.Text = AC
                    TRANSACCION.CAM_IDCAMATextBox.Text = -1
                    TRANSACCION.IDTIPOTRANSACCIONTextBox.Text = 2
                    TRANSACCION.COSTODIARIOTRANSACCIONTextBox.Text
= 0
                    TRANSACCION.MONTOTRANSACCIONTextBox.Text =
devolucion

TRANSACCION.FECHATRANSACCIONDateTimePicker.Value = Today

TRANSACCION.FECHAINICIOTRANSACCIONDateTimePicker.Value = Today

TRANSACCION.FECHAFINTRANSACCIONDateTimePicker.Value = Today


TRANSACCION.TRANSACCIONBindingNavigatorSaveItem.PerformClick()
```

```
                    TRANSACCION.Dispose()
                    If diashospedados >= 0 Then

CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(True, CA)

CAMA.CAMATableAdapter.ACTUALIZARDISPONIBILIDAD(True, CAAC)
                    End If

                    MsgBox("DEVOLUCIÓN REGISTRADA")

                Catch ex As Exception

                End Try
                Me.Dispose()
                VERDEVOLUCIONPAGO.Show()
            End If

        Catch
            PAGO.Dispose()
            'MsgBox("No se puede ordenar por este campo")
        End Try
    End Sub

    Private Sub ToolStripButton2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles Imprimir.Click
        ImprimirDGV.Print_DataGridView(BUSCARPAGODataGridView)
    End Sub
End Class
```

VER TIPOCAMA

```
Public Class VERTIPOCAMA

    Private Sub FillToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles FillToolStripButton.Click,
NOMBRETIPOCAMAToolStripTextBox.TextChanged, MyBase.Load,
DESCRIPCIONTIPOCAMAToolStripTextBox.TextChanged
        Try

Me.BUSCARTIPOCAMATableAdapter.Fill(Me.FUDIS_BASEDataSet.BUSCARTIPOC
AMA, NOMBRETIPOCAMAToolStripTextBox.Text,
DESCRIPCIONTIPOCAMAToolStripTextBox.Text)
        Catch ex As System.Exception
            System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try

    End Sub

    Private Sub BUSCARTIPOCAMADataGridView_CellContentClick(sender
As System.Object, e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles
BUSCARTIPOCAMADataGridView.CellContentClick
        Try
            If e.ColumnIndex = 2 Then
                TIPOCAMA.Show()

TIPOCAMA.TIPOCAMATableAdapter.BUSCARTIPOCAMAID(TIPOCAMA.FUDIS_BASED
ataSet.TIPOCAMA,
BUSCARTIPOCAMADataGridView.Rows(e.RowIndex).Cells(0).Value())
                If
BUSCARTIPOCAMADataGridView.Columns("AGREGAR").Visible = True Then
```

```
                    TIPOCAMA.ELIMINARButton.Visible = False
                End If
            ElseIf e.ColumnIndex = 1 Then
                CAMA.IDTIPOCAMATextBox.Text =
BUSCARTIPOCAMADataGridView.Rows(e.RowIndex).Cells(0).Value()
                MsgBox("TIPO DE CAMA AGREGADA")
                Me.Dispose()
            End If
        Catch
            TIPOCAMA.Dispose()
            'MsgBox("No se puede ordenar por este campo")
        End Try
    End Sub

    Private Sub CREARToolStripButton_Click(sender As System.Object,
e As System.EventArgs) Handles CREARToolStripButton.Click
        TIPOCAMA.Show()
        TIPOCAMA.BindingNavigatorAddNewItem.PerformClick()
        TIPOCAMA.ELIMINARButton.Visible = False
    End Sub
End Class
```

VERUSUARIO

```
Public Class VERUSUARIO

    Private Sub USUARIOBindingNavigatorSaveItem_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
USUARIOBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.USUARIOBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_
GRADODataSet)

    End Sub

    Private Sub VERUSUARIO_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        TIPOUSUARIOComboBox.SelectedItem = 0
        'TODO: esta línea de código carga datos en la tabla
'FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADODataSet.USUARIO' Puede moverla
o quitarla según sea necesario.

Me.USUARIOTableAdapter.Fill(Me.FUNDACIÓN_FUDIS_DISERTACIÓN_DE_GRADO
DataSet.USUARIO)

    End Sub

    Private Sub CANCELARButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CANCELARButton.Click
        Me.Dispose()
    End Sub

    Private Sub GUARDARButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles GUARDARButton.Click
        If controlUsuario.controlingresousuario(USUARIOOBLLabel,
NOMBREUSUARIOTextBox, CLAVEOBLabel, CLAVEUSUARIOTextBox,
CATEGORIAOBLabel, TIPOUSUARIOComboBox) Then
            USUARIOBindingNavigatorSaveItem.PerformClick()
```

```
            MsgBox("USUARIO GUARDADO", MsgBoxStyle.Information,
"Aviso del Sistema") 'GUARDA LOS DATOS


        Else
            MsgBox("CAMPOS OBLIGATORIOS INCOMPLETOS",
MsgBoxStyle.Information, "Aviso del Sistema")
        End If
    End Sub


    Private Sub ELIMINARButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ELIMINARButton.Click
        If (MessageBox.Show("SÍ ELIMINA A: """ +
NOMBREUSUARIOTextBox.Text + """, NO TENDRÁ ACCESO A LA APLICACIÓN
CON ESTE USUARIO. ¿DESEA CONTINUAR?", "Aviso del Sistema",
MessageBoxButtons.OKCancel) = DialogResult.OK) Then
            BindingNavigatorDeleteItem.PerformClick()
            USUARIOBindingNavigatorSaveItem.PerformClick()
            MsgBox("USUARIO ELIMINADO", MsgBoxStyle.Information,
"Aviso del Sistema")

        End If
    End Sub
End Class
```

VISTAPREVIA

```
Imports System.Drawing.Printing
Imports System.ComponentModel


Public Class VistaPrevia

    ' ** llenado
    Private _doc As PrintDocument

    ' ** actores
    Public Sub New()
        Me.New(Nothing)
    End Sub

    Public Sub New(ByVal parentForm As Control)
        Me.components = Nothing
        Me.InitializeComponent()
        If (Not parentForm Is Nothing) Then
            MyBase.Size = parentForm.Size
        End If
    End Sub

    ' ** Sobrecarga
    Protected Overrides Sub OnFormClosing(ByVal e As
FormClosingEventArgs)
        MyBase.OnFormClosing(e)
        If Not (Not Me._preview.IsRendering OrElse e.Cancel) Then
            Me._preview.Cancel()
        End If
    End Sub

    Protected Overrides Sub OnShown(ByVal e As EventArgs)
        MyBase.OnShown(e)
```

```
            Me._preview.Document = Me.Document
        End Sub

        ' ** propiedades
        Public Property Document() As PrintDocument
            Get
                Return Me._doc
            End Get
            Set(ByVal value As PrintDocument)
                If (Not Me._doc Is Nothing) Then
                    RemoveHandler Me._doc.BeginPrint, New
PrintEventHandler(AddressOf Me._doc_BeginPrint)
                    RemoveHandler Me._doc.EndPrint, New
PrintEventHandler(AddressOf Me._doc_EndPrint)
                End If
                Me._doc = value
                If (Not Me._doc Is Nothing) Then
                    AddHandler Me._doc.BeginPrint, New
PrintEventHandler(AddressOf Me._doc_BeginPrint)
                    AddHandler Me._doc.EndPrint, New
PrintEventHandler(AddressOf Me._doc_EndPrint)
                End If
                If MyBase.Visible Then
                    Me._preview.Document = Me.Document
                End If
            End Set
        End Property

        ' ** eventos de handlers
        Private Sub _btnCancel_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles _btnCancel.Click
            If Me._preview.IsRendering Then
                Me._preview.Cancel()
            Else
                MyBase.Close()
            End If
        End Sub

        Private Sub _btnFirst_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles _btnFirst.Click
            Me._preview.StartPage = 0
        End Sub

        Private Sub _btnLast_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles _btnLast.Click
            Me._preview.StartPage = (Me._preview.PageCount - 1)
        End Sub

        Private Sub _btnNext_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles _btnNext.Click
            Me._preview.StartPage += 1
        End Sub

        Private Sub _btnPrev_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles _btnPrev.Click
            Me._preview.StartPage -= 1
        End Sub

        Private Sub _btnPageSetup_Click(ByVal sender As Object, ByVal e
As EventArgs) Handles _btnPageSetup.Click
```

```
            Using dlg As PageSetupDialog = New PageSetupDialog
                dlg.Document = Me.Document
                If (dlg.ShowDialog(Me) = DialogResult.OK) Then
                    Me._preview.RefreshPreview()
                End If
            End Using
        End Sub

        Private Sub _btnPrint_Click(ByVal sender As Object, ByVal e As
EventArgs) Handles _btnPrint.Click
            Using dlg As PrintDialog = New PrintDialog
                dlg.AllowSomePages = True
                dlg.AllowSelection = True
                dlg.UseEXDialog = True
                dlg.Document = Me.Document
                Dim ps As PrinterSettings = dlg.PrinterSettings
                ps.MinimumPage = 1
                ps.MaximumPage = Me._preview.PageCount
                ps.FromPage = 1
                ps.ToPage = Me._preview.PageCount
                If (dlg.ShowDialog(Me) = DialogResult.OK) Then
                    Me._preview.Print()
                End If
            End Using
        End Sub

        Private Sub _btnZoom_ButtonClick(ByVal sender As Object, ByVal
e As EventArgs) Handles _btnZoom.ButtonClick
            Me._preview.ZoomMode = IIf((Me._preview.ZoomMode =
ZoomMode.ActualSize), ZoomMode.FullPage, ZoomMode.ActualSize)
        End Sub

        Private Sub _btnZoom_DropDownItemClicked(ByVal sender As
Object, ByVal e As ToolStripItemClickedEventArgs) Handles
_btnZoom.DropDownItemClicked
            If (e.ClickedItem Is Me._itemActualSize) Then
                Me._preview.ZoomMode = ZoomMode.ActualSize
            ElseIf (e.ClickedItem Is Me._itemFullPage) Then
                Me._preview.ZoomMode = ZoomMode.FullPage
            ElseIf (e.ClickedItem Is Me._itemPageWidth) Then
                Me._preview.ZoomMode = ZoomMode.PageWidth
            ElseIf (e.ClickedItem Is Me._itemTwoPages) Then
                Me._preview.ZoomMode = ZoomMode.TwoPages
            End If
            If (e.ClickedItem Is Me._item10) Then
                Me._preview.Zoom = 0.1
            ElseIf (e.ClickedItem Is Me._item100) Then
                Me._preview.Zoom = 1
            ElseIf (e.ClickedItem Is Me._item150) Then
                Me._preview.Zoom = 1.5
            ElseIf (e.ClickedItem Is Me._item200) Then
                Me._preview.Zoom = 2
            ElseIf (e.ClickedItem Is Me._item25) Then
                Me._preview.Zoom = 0.25
            ElseIf (e.ClickedItem Is Me._item50) Then
                Me._preview.Zoom = 0.5
            ElseIf (e.ClickedItem Is Me._item500) Then
                Me._preview.Zoom = 5
            ElseIf (e.ClickedItem Is Me._item75) Then
                Me._preview.Zoom = 0.75
```

```
            End If
        End Sub

        Private Sub _doc_BeginPrint(ByVal sender As Object, ByVal e As
PrintEventArgs)
            Me._btnCancel.Text = "&Cancel"
            Me._btnPrint.Enabled = Me._btnPageSetup.Enabled = False
        End Sub

        Private Sub _doc_EndPrint(ByVal sender As Object, ByVal e As
PrintEventArgs)
            Me._btnCancel.Text = "&Cerrar"
            Me._btnPrint.Enabled = Me._btnPageSetup.Enabled = True
        End Sub

        Private Sub _preview_PageCountChanged(ByVal sender As Object,
ByVal e As EventArgs) Handles _preview.PageCountChanged
            MyBase.Update()
            Application.DoEvents()
            Me._lblPageCount.Text = String.Format("de {0}",
Me._preview.PageCount)
        End Sub

        Private Sub _preview_StartPageChanged(ByVal sender As Object,
ByVal e As EventArgs) Handles _preview.StartPageChanged
            Me._txtStartPage.Text = (Me._preview.StartPage +
1).ToString
        End Sub

        Private Sub _txtStartPage_Enter(ByVal sender As Object, ByVal e
As EventArgs) Handles _txtStartPage.Enter
            Me._txtStartPage.SelectAll()
        End Sub

        Private Sub _txtStartPage_KeyPress(ByVal sender As Object,
ByVal e As KeyPressEventArgs) Handles _txtStartPage.KeyPress
            Dim c As Char = e.KeyChar
            If (c = ChrW(13)) Then
                Me.CommitPageNumber()
                e.Handled = True
            ElseIf Not ((c <= " "c) OrElse Char.IsDigit(c)) Then
                e.Handled = True
            End If
        End Sub

        Private Sub _txtStartPage_Validating(ByVal sender As Object,
ByVal e As CancelEventArgs) Handles _txtStartPage.Validating
            Me.CommitPageNumber()
        End Sub

        Private Sub CommitPageNumber()
            Dim page As Integer
            If Integer.TryParse(Me._txtStartPage.Text, page) Then
                Me._preview.StartPage = (page - 1)
            End If
        End Sub

    End Class
```

CLASES

CONTROLUSUARIO

```
Public Class controlUsuario
    Public Shared Sub controlusuariocontarcaracteres(label As
Label, textbox As TextBox)
        label.Text = "Caracteres disponibles: " +
(textbox.MaxLength - textbox.Text.Length()).ToString()
    End Sub

    Public Shared Sub controlusuariomoneda(textbox1 As TextBox,
label1 As Label)
        Try
            textbox1.Text =
Convert.ToDecimal(textbox1.Text).ToString()
            label1.Text = "*"
        Catch ex As Exception
            label1.Text = "Ingrese un valor monetario válido"
        End Try

    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox, textbox4 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
        If textbox4.Text = "" Then
            textbox4.Text = " "
        End If
    End Sub
```

```
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox, textbox4 As TextBox,
textbox5 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
        If textbox4.Text = "" Then
            textbox4.Text = " "
        End If
        If textbox5.Text = "" Then
            textbox5.Text = " "
        End If
    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox, textbox4 As TextBox,
textbox5 As TextBox, textbox6 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
        If textbox4.Text = "" Then
            textbox4.Text = " "
        End If
        If textbox5.Text = "" Then
            textbox5.Text = " "
        End If
    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox, textbox4 As TextBox,
textbox5 As TextBox, textbox6 As TextBox, textbox7 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
        If textbox4.Text = "" Then
            textbox4.Text = " "
        End If
        If textbox5.Text = "" Then
            textbox5.Text = " "
        End If
        If textbox6.Text = "" Then
            textbox6.Text = " "
        End If
```

```
    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox, textbox4 As TextBox,
textbox5 As TextBox, textbox6 As TextBox, textbox7 As TextBox,
textbox8 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
        If textbox4.Text = "" Then
            textbox4.Text = " "
        End If
        If textbox5.Text = "" Then
            textbox5.Text = " "
        End If
        If textbox6.Text = "" Then
            textbox6.Text = " "
        End If
        If textbox7.Text = "" Then
            textbox7.Text = " "
        End If

    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox, textbox4 As TextBox,
textbox5 As TextBox, textbox6 As TextBox, textbox7 As TextBox,
textbox8 As TextBox, textbox9 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
        If textbox4.Text = "" Then
            textbox4.Text = " "
        End If
        If textbox5.Text = "" Then
            textbox5.Text = " "
        End If
        If textbox6.Text = "" Then
            textbox6.Text = " "
        End If
        If textbox7.Text = "" Then
            textbox7.Text = " "
        End If
        If textbox8.Text = "" Then
            textbox8.Text = " "
        End If
        If textbox9.Text = "" Then
            textbox9.Text = " "
        End If
```

```
    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox, textbox4 As TextBox,
textbox5 As TextBox, textbox6 As TextBox, textbox7 As TextBox,
textbox8 As TextBox, textbox9 As TextBox, textbox10 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
        If textbox4.Text = "" Then
            textbox4.Text = " "
        End If
        If textbox5.Text = "" Then
            textbox5.Text = " "
        End If
        If textbox6.Text = "" Then
            textbox6.Text = " "
        End If
        If textbox7.Text = "" Then
            textbox7.Text = " "
        End If
        If textbox8.Text = "" Then
            textbox8.Text = " "
        End If
        If textbox9.Text = "" Then
            textbox9.Text = " "
        End If
        If textbox10.Text = "" Then
            textbox10.Text = " "
        End If

    End Sub
    Public Shared Sub controlusuariononull(textbox1 As TextBox,
textbox2 As TextBox, textbox3 As TextBox, textbox4 As TextBox,
textbox5 As TextBox, textbox6 As TextBox, textbox7 As TextBox,
textbox8 As TextBox, textbox9 As TextBox, textbox10 As TextBox,
textbox11 As TextBox)
        If textbox1.Text = "" Then
            textbox1.Text = " "
        End If
        If textbox2.Text = "" Then
            textbox2.Text = " "
        End If
        If textbox3.Text = "" Then
            textbox3.Text = " "
        End If
        If textbox4.Text = "" Then
            textbox4.Text = " "
        End If
        If textbox5.Text = "" Then
            textbox5.Text = " "
        End If
        If textbox6.Text = "" Then
```

```
                textbox6.Text = " "
            End If
            If textbox7.Text = "" Then
                textbox7.Text = " "
            End If
            If textbox8.Text = "" Then
                textbox8.Text = " "
            End If
            If textbox9.Text = "" Then
                textbox9.Text = " "
            End If
            If textbox10.Text = "" Then
                textbox10.Text = " "
            End If
            If textbox11.Text = "" Then
                textbox11.Text = " "
            End If
        End Sub
        Public Shared Function controlusuariocampoobligatorio(label1 As
Label, textbox1 As TextBox) As Boolean
            Dim retorno As Boolean
            retorno = True
            If textbox1.Text = "" Then
                label1.Text = "Campo Obligatorio"
                retorno = False
            Else
                label1.Text = "*"
            End If
            Return retorno
        End Function

        Public Shared Function controlingresousuario(ByVal label1 As
Label, ByVal textbox1 As TextBox, ByVal label2 As Label, ByVal
textbox2 As TextBox, ByVal label3 As Label, ByVal combobox1 As
ComboBox) As Boolean
            Dim retorno As Boolean
            retorno = True
            If textbox1.Text = "" Then
                label1.Text = "Campo Obligatorio"
                retorno = False
            Else
                label1.Text = "*"
            End If
            If textbox2.Text = "" Then
                label2.Text = "Campo Obligatorio"
                retorno = False
            Else
                label2.Text = "*"
            End If
            If combobox1.Text = "" Then
                label3.Text = "Seleccione una opción"
                retorno = False
            Else
                label3.Text = "*"
            End If
            Return retorno
        End Function
        Public Shared Function controlusuariocampoobligatorio(label1 As
Label, textbox1 As TextBox, label2 As Label, textbox2 As TextBox)
As Boolean
```

```
            Dim retorno As Boolean
            retorno = True
            If textbox1.Text = "" Then
                label1.Text = "Campo Obligatorio"
                retorno = False
            Else
                label1.Text = "*"
            End If
            If textbox2.Text = "" Then
                label2.Text = "Campo Obligatorio"
                retorno = False
            Else
                label2.Text = "*"
            End If
            Return retorno
        End Function
        Public Shared Function controlusuariocampoobligatorio(label1 As
Label, textbox1 As TextBox, label2 As Label, textbox2 As TextBox,
label3 As Label, textbox3 As TextBox) As Boolean
            Dim retorno As Boolean
            retorno = True
            If textbox1.Text = "" Then
                label1.Text = "Campo Obligatorio"
                retorno = False
            Else
                label1.Text = "*"
            End If
            If textbox2.Text = "" Then
                label2.Text = "Campo Obligatorio"
                retorno = False
            Else
                label2.Text = "*"
            End If
            If textbox3.Text = "" Then
                label3.Text = "Campo Obligatorio"
                retorno = False
            Else
                label3.Text = "*"
            End If
            Return retorno
        End Function
        Public Shared Function controlusuariocampoobligatorio(label1 As
Label, textbox1 As TextBox, label2 As Label, textbox2 As TextBox,
label3 As Label, textbox3 As TextBox, label4 As Label, textbox4 As
TextBox) As Boolean
            Dim retorno As Boolean
            retorno = True
            If textbox1.Text = "" Then
                label1.Text = "Campo Obligatorio"
                retorno = False
            Else
                label1.Text = "*"
            End If
            If textbox2.Text = "" Then
                label2.Text = "Campo Obligatorio"
                retorno = False
            Else
                label2.Text = "*"
            End If
            If textbox3.Text = "" Then
```

```
                label3.Text = "Campo Obligatorio"
                retorno = False
            Else
                label3.Text = "*"
            End If
            If textbox4.Text = "" Then
                label4.Text = "Campo Obligatorio"
                retorno = False
            Else
                label4.Text = "*"
            End If
            Return retorno
        End Function
        Public Shared Function controlusuariocampoobligatorio(label1 As
Label, textbox1 As TextBox, label2 As Label, textbox2 As TextBox,
label3 As Label, textbox3 As TextBox, label4 As Label, textbox4 As
TextBox, label5 As Label, textbox5 As TextBox) As Boolean
            Dim retorno As Boolean
            retorno = True
            If textbox1.Text = "" Then
                label1.Text = "Campo Obligatorio"
                retorno = False
            Else
                label1.Text = "*"
            End If
            If textbox2.Text = "" Then
                label2.Text = "Campo Obligatorio"
                retorno = False
            Else
                label2.Text = "*"
            End If
            If textbox3.Text = "" Then
                label3.Text = "Campo Obligatorio"
                retorno = False
            Else
                label3.Text = "*"
            End If
            If textbox4.Text = "" Then
                label4.Text = "Campo Obligatorio"
                retorno = False
            Else
                label4.Text = "*"
            End If
            If textbox5.Text = "" Then
                label5.Text = "Campo Obligatorio"
                retorno = False
            Else
                label5.Text = "*"
            End If
            Return retorno
        End Function
        Public Shared Sub quitarblanco(textbox1 As TextBox)
            If textbox1.Text = " " Then
                textbox1.Text = ""
            End If
        End Sub
        Public Shared Sub quitarblanco(textbox1 As TextBox, textbox2 As
TextBox)
            If textbox1.Text = " " Then
                textbox1.Text = ""
```

```
            End If
            If textbox2.Text = " " Then
                textbox2.Text = ""
            End If
        End Sub
        Public Shared Sub quitarblanco(textbox1 As TextBox, textbox2 As
TextBox, textbox3 As TextBox)
            If textbox1.Text = " " Then
                textbox1.Text = ""
            End If
            If textbox2.Text = " " Then
                textbox2.Text = ""
            End If
            If textbox3.Text = " " Then
                textbox3.Text = ""
            End If
        End Sub
        Public Shared Sub quitarblanco(textbox1 As TextBox, textbox2 As
TextBox, textbox3 As TextBox, textbox4 As TextBox)
            If textbox1.Text = " " Then
                textbox1.Text = ""
            End If
            If textbox2.Text = " " Then
                textbox2.Text = ""
            End If
            If textbox3.Text = " " Then
                textbox3.Text = ""
            End If
            If textbox4.Text = " " Then
                textbox4.Text = ""
            End If
        End Sub
        Public Shared Sub quitarblanco(textbox1 As TextBox, textbox2 As
TextBox, textbox3 As TextBox, textbox4 As TextBox, textbox5 As
TextBox)
            If textbox1.Text = " " Then
                textbox1.Text = ""
            End If
            If textbox2.Text = " " Then
                textbox2.Text = ""
            End If
            If textbox3.Text = " " Then
                textbox3.Text = ""
            End If
            If textbox4.Text = " " Then
                textbox4.Text = ""
            End If
            If textbox5.Text = " " Then
                textbox5.Text = ""
            End If
        End Sub
        Public Shared Function añadirenter(cadena As String) As String
            Dim i, cont As Integer
            cont = 0
            For i = 0 To (cadena.Length() - 1)
                If cadena.Substring(i, 1) = " " Then
                    cont = cont + 1
                    If cont Mod 3 = 0 Then
                        cadena = cadena.Substring(0, i - 1) + vbCrLf +
cadena.Substring(i + 1)
```

```
                End If
            End If
        Next




        Return cadena
    End Function
End Class
```

IMPRIMIRDGV

```
Imports System.Collections.Generic


Public Class ImprimirDGV
    Private Shared StrFormat As StringFormat         ' Mantiene
contenido de una celda de caja de texto por dibujar
    Private Shared StrFormatComboBox As StringFormat ' Mantiene
contenio de una celda booleana para dibujar imagenes
    Private Shared CellButton As Button          ' Mantiene el
contenido de Button Cell
    Private Shared CellCheckBox As CheckBox      ' Mantiene el
contenido de CheckBox Cell
    Private Shared CellComboBox As ComboBox      ' Mantiene el
contenido de ComboBox Cell
    Private Shared TotalWidth As Int16           ' Suma de columnas
    Private Shared RowPos As Int16               ' Posicion actual
de la celda de impresión
    Private Shared NewPage As Boolean            ' Indica si
necesita una nueva pagina
    Private Shared PageNo As Int16               ' Numero de
paginas a imprimir
    Private Shared ColumnLefts As New ArrayList  ' Coordina
columnas a la izquierda
    Private Shared ColumnWidths As New ArrayList ' Ancho de
columnas
    Private Shared ColumnTypes As New ArrayList  ' Tipo de datos de
columnas
    Private Shared CellHeight As Int16           ' Alto de las
celdas de DataGrid
    Private Shared RowsPerPage As Int16          ' Número de dilas
por Página
    Public Shared WithEvents PrintDoc As New
System.Drawing.Printing.PrintDocument ' Imprime el objeto usado por
la impresión
    Private Shared PrintTitle As String = ""               '
Cabecera de paginas
    Private Shared dgv As DataGridView                     '
Mantiene el objeto del DataGrid para imprimir el contenido
    Private Shared SelectedColumns As New List(Of String)  ' Las
columnas seleccionadas por el usuraio para imprimir
    Private Shared AvailableColumns As New List(Of String) ' Todas
las columnas habilitadas en el DataGrid
    Private Shared PrintAllRows As Boolean = True          ' True =
imprimir tdas las filas,  False = imprimir filas seleccionadas
```

```
    Private Shared FitToPageWidth As Boolean = True        ' True =
filas seleccionadas ,  False = imprimir columnas mostradas
    Private Shared HeaderHeight As Int16 = 0



    Public Shared Sub Print_DataGridView(ByVal dgv1 As
DataGridView)

        Using dlg1 As VistaPrevia = New VistaPrevia
            Try

                ' Conseguir los objetos del DataGridView para
imprimir
                dgv = dgv1
                ' Conseguir todas los nombres de las columnas en el
DataGridView
                AvailableColumns.Clear()
                For Each c As DataGridViewColumn In dgv.Columns
                    If Not c.Visible Then Continue For
                    AvailableColumns.Add(c.HeaderText)
                Next
                ' Mostrar el formulario de Opciones de impresión
                Dim dlg As New OpcioneImpresion(AvailableColumns)
                If dlg.ShowDialog() <> DialogResult.OK Then Exit
Sub
                ' Guardar algunos atributos de impresiones
                PrintTitle = dlg.PrintTitle
                PrintAllRows = dlg.PrintAllRows
                FitToPageWidth = dlg.FitToPageWidth
                SelectedColumns = dlg.GetSelectedColumns
                RowsPerPage = 0
                dlg1.Document = PrintDoc
                'Mostrar la Vista Previa de impresión
                If dlg1.ShowDialog() <> DialogResult.OK Then Exit
Sub
                'Imprimir el documento
                PrintDoc.Print()

            Catch ex As Exception
                MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
            Finally

            End Try
        End Using
    End Sub


    Private Shared Sub PrintDoc_BeginPrint(ByVal sender As Object,
_
                ByVal e As System.Drawing.Printing.PrintEventArgs)
Handles PrintDoc.BeginPrint

        Try
            ' Formatear el contenido de las celdas de texto a
imprimir
            StrFormat = New StringFormat
            StrFormat.Alignment = StringAlignment.Near
            StrFormat.LineAlignment = StringAlignment.Center
```

```
            StrFormat.Trimming = StringTrimming.EllipsisCharacter

            ' Formatear el contenido de las celdas de combobox a
imprimir
            StrFormatComboBox = New StringFormat
            StrFormatComboBox.LineAlignment =
StringAlignment.Center
            StrFormatComboBox.FormatFlags =
StringFormatFlags.NoWrap
            StrFormatComboBox.Trimming =
StringTrimming.EllipsisCharacter

            ColumnLefts.Clear()
            ColumnWidths.Clear()
            ColumnTypes.Clear()
            CellHeight = 0
            RowsPerPage = 0

            ' Diseño para Varios tipos de columnas
            CellButton = New Button
            CellCheckBox = New CheckBox
            CellComboBox = New ComboBox

            TotalWidth = 0
            For Each GridCol As DataGridViewColumn In dgv.Columns
                If Not GridCol.Visible Then Continue For
                If Not SelectedColumns.Contains(GridCol.HeaderText)
Then Continue For
                TotalWidth += GridCol.Width
            Next
            PageNo = 1
            NewPage = True
            RowPos = 0
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        Finally

        End Try
    End Sub

    Private Shared Sub PrintDoc_PrintPage(ByVal sender As Object, _
            ByVal e As System.Drawing.Printing.PrintPageEventArgs)
Handles PrintDoc.PrintPage


        Dim tmpWidth As Int16, i As Int16
        Dim tmpTop As Int16 = e.MarginBounds.Top
        Dim tmpLeft As Int16 = e.MarginBounds.Left

        Try
            ' antes de empezar la primera pagina, guarda el ancho y
alto de la cabecera y el tipo de columna
            If PageNo = 1 Then
                For Each GridCol As DataGridViewColumn In
dgv.Columns
                    If Not GridCol.Visible Then Continue For
                    If Not
SelectedColumns.Contains(GridCol.HeaderText) Then
                        Continue For
```

```
                    End If

                    ' determina si las columnas se dispone a la
página o no.
                    If FitToPageWidth Then
                        tmpWidth = CType(Math.Floor(GridCol.Width /
TotalWidth * _
                                        TotalWidth *
(e.MarginBounds.Width / TotalWidth)), Int16)
                    Else
                        tmpWidth = GridCol.Width
                    End If
                    HeaderHeight =
e.Graphics.MeasureString(GridCol.HeaderText, _
                                        GridCol.InheritedStyle.Font,
tmpWidth).Height + 11

                    ColumnLefts.Add(tmpLeft)
                    ColumnWidths.Add(tmpWidth)
                    ColumnTypes.Add(GridCol.GetType)
                    tmpLeft += tmpWidth
                Next
            End If

            ' Imprimir pagina actual fila por fila
            Do While RowPos <= dgv.Rows.Count - 1
                Dim GridRow As DataGridViewRow = dgv.Rows(RowPos)
                If GridRow.IsNewRow OrElse (Not PrintAllRows
AndAlso Not GridRow.Selected) Then
                    RowPos += 1 : Continue Do
                End If

                CellHeight = GridRow.Height

                If tmpTop + CellHeight >= e.MarginBounds.Height +
e.MarginBounds.Top Then
                    DrawFooter(e, RowsPerPage)
                    NewPage = True
                    PageNo += 1
                    e.HasMorePages = True
                    Exit Sub
                Else
                    If NewPage Then
                        ' Dibujar cabecera
                        e.Graphics.DrawString(PrintTitle, New
Font(dgv.Font, FontStyle.Bold), _
                                        Brushes.Black, e.MarginBounds.Left,
e.MarginBounds.Top - _
                        e.Graphics.MeasureString(PrintTitle, New
Font(dgv.Font, _
                                        FontStyle.Bold),
e.MarginBounds.Width).Height - 13)

                        Dim s As String = "Fecha: " +
Now.ToLongDateString + " " + Now.ToShortTimeString

                        e.Graphics.DrawString(s, New Font(dgv.Font,
FontStyle.Bold), _
                           Brushes.Black, e.MarginBounds.Left +
(e.MarginBounds.Width - _
```

```
                            e.Graphics.MeasureString(s, New
Font(dgv.Font, FontStyle.Bold), _
                            e.MarginBounds.Width).Width),
e.MarginBounds.Top - _
                            e.Graphics.MeasureString(PrintTitle, _
                            New Font(New Font(dgv.Font,
FontStyle.Bold), FontStyle.Bold), _
                            e.MarginBounds.Width).Height - 13)

                        ' Dibujar Columnas
                        tmpTop = e.MarginBounds.Top
                        i = 0
                        For Each GridCol As DataGridViewColumn In
dgv.Columns
                            If Not GridCol.Visible Then Continue
For
                            If Not
SelectedColumns.Contains(GridCol.HeaderText) Then
                                Continue For
                            End If

                            e.Graphics.FillRectangle(New
SolidBrush(Drawing.Color.LightGray), _
                                    New Rectangle(ColumnLefts(i),
tmpTop, ColumnWidths(i), HeaderHeight))

                            e.Graphics.DrawRectangle(Pens.Black,
New Rectangle(ColumnLefts(i), _
                                    tmpTop, ColumnWidths(i),
HeaderHeight))


e.Graphics.DrawString(GridCol.HeaderText,
GridCol.InheritedStyle.Font, _
                                    New
SolidBrush(GridCol.InheritedStyle.ForeColor), _
                                    New RectangleF(ColumnLefts(i),
tmpTop, ColumnWidths(i), _
                                    HeaderHeight), StrFormat)
                            i += 1
                        Next
                        NewPage = False

                        tmpTop += HeaderHeight
                    End If

                    i = 0
                    For Each Cel As DataGridViewCell In
GridRow.Cells
                        If Not Cel.OwningColumn.Visible Then
Continue For
                        If Not
SelectedColumns.Contains(Cel.OwningColumn.HeaderText) Then
                            Continue For
                        End If

                        ' para las columnas de textbox
                        If ColumnTypes(i) Is
GetType(DataGridViewTextBoxColumn) OrElse _
```

```
                                ColumnTypes(i) Is
GetType(DataGridViewLinkColumn) Then


e.Graphics.DrawString(Cel.Value.ToString, Cel.InheritedStyle.Font,
_
                                        New
SolidBrush(Cel.InheritedStyle.ForeColor), _
                                        New RectangleF(ColumnLefts(i),
tmpTop, ColumnWidths(i), _
                                        CellHeight), StrFormat)

                                ' para los botones en la columna
                            ElseIf ColumnTypes(i) Is
GetType(DataGridViewButtonColumn) Then

                                CellButton.Text = Cel.Value.ToString
                                CellButton.Size = New
Size(ColumnWidths(i), CellHeight)
                                Dim bmp As New Bitmap(CellButton.Width,
CellButton.Height)
                                CellButton.DrawToBitmap(bmp, New
Rectangle(0, 0, _
                                        bmp.Width, bmp.Height))
                                e.Graphics.DrawImage(bmp, New
Point(ColumnLefts(i), tmpTop))

                                ' para los checkbox en la columna
                            ElseIf ColumnTypes(i) Is
GetType(DataGridViewCheckBoxColumn) Then

                                CellCheckBox.Size = New Size(14, 14)
                                CellCheckBox.Checked = CType(Cel.Value,
Boolean)
                                Dim bmp As New Bitmap(ColumnWidths(i),
CellHeight)
                                Dim tmpGraphics As Graphics =
Graphics.FromImage(bmp)

tmpGraphics.FillRectangle(Brushes.White, New Rectangle(0, 0, _
                                        bmp.Width, bmp.Height))
                                CellCheckBox.DrawToBitmap(bmp, New
Rectangle(CType((bmp.Width - _
                                        CellCheckBox.Width) / 2,
Int32), CType((bmp.Height - _
                                        CellCheckBox.Height) / 2,
Int32), CellCheckBox.Width, _
                                        CellCheckBox.Height))
                                e.Graphics.DrawImage(bmp, New
Point(ColumnLefts(i), tmpTop))

                                ' Para los combobox en la columna
                            ElseIf ColumnTypes(i) Is
GetType(DataGridViewComboBoxColumn) Then

                                CellComboBox.Size = New
Size(ColumnWidths(i), CellHeight)
                                Dim bmp As New
Bitmap(CellComboBox.Width, CellComboBox.Height)
```

```
                                CellComboBox.DrawToBitmap(bmp, New
Rectangle(0, 0, _
                                        bmp.Width, bmp.Height))
                                e.Graphics.DrawImage(bmp, New
Point(ColumnLefts(i), tmpTop))

e.Graphics.DrawString(Cel.Value.ToString, Cel.InheritedStyle.Font,
_
                                        New
SolidBrush(Cel.InheritedStyle.ForeColor), _
                                        New RectangleF(ColumnLefts(i) +
1, tmpTop, ColumnWidths(i) _
                                        - 16, CellHeight),
StrFormatComboBox)

                                ' Para imagenes en la columna
                            ElseIf ColumnTypes(i) Is
GetType(DataGridViewImageColumn) Then

                                Dim CelSize As Rectangle = New
Rectangle(ColumnLefts(i), _
                                        tmpTop, ColumnWidths(i),
CellHeight)
                                Dim ImgSize As Size =
CType(Cel.FormattedValue, Image).Size

e.Graphics.DrawImage(Cel.FormattedValue, New
Rectangle(ColumnLefts(i) _
                                        + CType(((CelSize.Width -
ImgSize.Width) / 2), Int32), _
                                        tmpTop + CType(((CelSize.Height
- ImgSize.Height) / 2),  _
                                        Int32),
CType(Cel.FormattedValue, Image).Width, CType(Cel.FormattedValue,
_
                                        Image).Height))

                            End If

                            ' dibujando los bordes de las celdas
                            e.Graphics.DrawRectangle(Pens.Black, New
Rectangle(ColumnLefts(i), _
                                    tmpTop, ColumnWidths(i),
CellHeight))

                            i += 1

                        Next
                        tmpTop += CellHeight

                    End If

                    RowPos += 1
                    ' Para la primera pagina calula filas por paginas
                    If PageNo = 1 Then
                        RowsPerPage += 1
                    End If
                Loop

                If RowsPerPage = 0 Then Exit Sub
```

```
            ' escribe el pie de pagina (numero de pagina)
            DrawFooter(e, RowsPerPage)

            e.HasMorePages = False
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        Finally

        End Try
    End Sub

    Private Shared Sub DrawFooter(ByVal e As
System.Drawing.Printing.PrintPageEventArgs, ByVal RowsPerPage As
Int32)
        Dim cnt As Integer

        ' Determina numero de filas para imprimir
        If PrintAllRows Then
            If dgv.Rows(dgv.Rows.Count - 1).IsNewRow Then
                ' Cuando el DataGridView no esta permitido no
adiciona filas
                cnt = dgv.Rows.Count - 2
            Else
                ' Cuando el DataGridView apermite adiciona filas
                cnt = dgv.Rows.Count - 1
            End If
        Else
            cnt = dgv.SelectedRows.Count
        End If

        ' Escrbiendo el numero de pagina on la parte inferior de la
página
        Dim PageNum As String = PageNo.ToString + " de " + _
                    Math.Ceiling(cnt / RowsPerPage).ToString
        e.Graphics.DrawString(PageNum, dgv.Font, Brushes.Black, _
                    e.MarginBounds.Left + (e.MarginBounds.Width - _
                    e.Graphics.MeasureString(PageNum, dgv.Font, _
                    e.MarginBounds.Width).Width) / 2,
e.MarginBounds.Top + _
                    e.MarginBounds.Height + 31)

    End Sub

End Class
```

LISTAVISTAIMAGEN

```
Imports System.IO
Imports System.ComponentModel
Imports System.Drawing.Imaging

#If False Then


    ' ** Llenado de opciones en tool strip
    Private _list As List(Of Byte()) = New List(Of Byte())

    ' ** Propiedades
```

```
        Public ReadOnly Property Count() As Integer
            Get
                Return Me._list.Count
            End Get
        End Property

        Default Public Property Item(ByVal index As Integer) As Image
            Get
                Return Me.GetImage(Me._list.Item(index))
            End Get
            Set(ByVal value As Image)
                Me._list.Item(index) = Me.GetBytes(value)
            End Set
        End Property

        ' ** Métodos
        Public Sub Add(ByVal img As Image)
            Me._list.Add(Me.GetBytes(img))
            img.Dispose()
        End Sub

        Public Sub Clear()
            Me._list.Clear()
        End Sub

        ' ** implementación
        Function GetBytes(ByVal img As Image) As Byte()
            Dim mf As Metafile = TryCast(img, Metafile)
            Dim enhMetafileHandle As Integer =
mf.GetHenhmetafile.ToInt32
            Dim bufferSize As Integer =
PageImageList.GetEnhMetaFileBits(enhMetafileHandle, 0, Nothing)
            Dim buffer As Byte() = New Byte(bufferSize - 1) {}
            PageImageList.GetEnhMetaFileBits(enhMetafileHandle,
bufferSize, buffer)
            Return buffer
        End Function

        <System.Runtime.InteropServices.DllImport("gdi32")> _
        Shared Function GetEnhMetaFileBits(ByVal hemf As Integer, ByVal
cbBuffer As Integer, ByVal lpbBuffer As Byte()) As Integer
        End Function

        Function GetImage(ByVal data As Byte()) As Image
            Dim ms As New MemoryStream(data)
            Return Image.FromStream(ms)
        End Function

    End Class

    #Else

    ' Uso regular de lista genérica
    Friend Class ListaVistaImagen
        Inherits List(Of Image)
    End Class

    #End If
```

VISTAPREVIACONTROL

```
Imports System.Drawing.Imaging
Imports System.Drawing.Printing
Imports System.ComponentModel

Friend Enum ZoomMode
    ActualSize = 0
    Custom = 4
    FullPage = 1
    PageWidth = 2
    TwoPages = 3
End Enum

Friend Class VistaPreviaControl

    Inherits UserControl

    ' Eventos
    Public Event PageCountChanged As EventHandler
    Public Event StartPageChanged As EventHandler
    Public Event ZoomModeChanged As EventHandler

    ' Metodos
    Public Sub New()
        Me.BackColor = SystemColors.AppWorkspace
        Me.ZoomMode = ZoomMode.FullPage
        Me.StartPage = 0
        MyBase.SetStyle(ControlStyles.OptimizedDoubleBuffer, True)
    End Sub

    Private Sub _doc_EndPrint(ByVal sender As Object, ByVal e As
PrintEventArgs)
        Me.SyncPageImages(True)
    End Sub

    Private Sub _doc_PrintPage(ByVal sender As Object, ByVal e As
PrintPageEventArgs)
        Me.SyncPageImages(False)
        If Me._cancel Then
            e.Cancel = True
        End If
    End Sub

    Public Sub Cancel()
        Me._cancel = True
    End Sub

    Private Function GetImage(ByVal page As Integer) As Image
        If (page > -1) AndAlso (page < Me.PageCount) Then
            Return Me._img.Item(page)
        Else
            Return Nothing
        End If
    End Function

    Private Function GetImageRectangle(ByVal img As Image) As
Rectangle

        Dim sz As Size = Me.GetImageSizeInPixels(img)
        Dim rc As New Rectangle(0, 0, sz.Width, sz.Height)
```

```
            Dim rcCli As Rectangle = MyBase.ClientRectangle
            Select Case Me._zoomMode
                Case ZoomMode.ActualSize
                    Me._zoom = 1
                    GoTo Label_00FF
                Case ZoomMode.FullPage
                    Exit Select
                Case ZoomMode.PageWidth
                    Me._zoom = IIf((rc.Width > 0), (CDbl(rcCli.Width) /
CDbl(rc.Width)), 0)
                    GoTo Label_00FF
                Case ZoomMode.TwoPages
                    rc.Width = (rc.Width * 2)
                    Exit Select
                Case Else
                    GoTo Label_00FF
            End Select
            Dim zoomX As Double = IIf((rc.Width > 0),
(CDbl(rcCli.Width) / CDbl(rc.Width)), 0)
            Dim zoomY As Double = IIf((rc.Height > 0),
(CDbl(rcCli.Height) / CDbl(rc.Height)), 0)
            Me._zoom = Math.Min(zoomX, zoomY)
Label_00FF:
            rc.Width = CInt((rc.Width * Me._zoom))
            rc.Height = CInt((rc.Height * Me._zoom))
            Dim dx As Integer = ((rcCli.Width - rc.Width) / 2)
            If (dx > 0) Then
                rc.X = (rc.X + dx)
            End If
            Dim dy As Integer = ((rcCli.Height - rc.Height) / 2)
            If (dy > 0) Then
                rc.Y = (rc.Y + dy)
            End If
            rc.Inflate(-4, -4)
            If (Me._zoomMode = ZoomMode.TwoPages) Then
                rc.Inflate(-2, 0)
            End If
            Return rc
        End Function

        Private Function GetImageSizeInPixels(ByVal img As Image) As
Size
            Dim szf As SizeF = img.PhysicalDimension
            If TypeOf img Is Metafile Then
                If (Me._himm2pix.X < 0.0!) Then
                    Using g As Graphics = MyBase.CreateGraphics
                        Me._himm2pix.X = (g.DpiX / 2540.0!)
                        Me._himm2pix.Y = (g.DpiY / 2540.0!)
                    End Using
                End If
                szf.Width = (szf.Width * Me._himm2pix.X)
                szf.Height = (szf.Height * Me._himm2pix.Y)
            End If
            Return Size.Truncate(szf)
        End Function

        Protected Overrides Function IsInputKey(ByVal keyData As Keys)
As Boolean
            Select Case keyData
```

```
            Case Keys.Prior, Keys.Next, Keys.End, Keys.Home,
Keys.Left, Keys.Up, Keys.Right, Keys.Down
                Return True
        End Select
        Return MyBase.IsInputKey(keyData)
    End Function

    Protected Overrides Sub OnKeyDown(ByVal e As KeyEventArgs)
        MyBase.OnKeyDown(e)
        If e.Handled Then Return

        Select Case e.KeyCode

            ' permite deslizarse por la vista previa de acerdo al
zoom
            Case Keys.Left, Keys.Up, Keys.Right, Keys.Down

                ' buscardor
                If ZoomMode = ZoomMode.FullPage OrElse ZoomMode =
ZoomMode.TwoPages Then
                    Select Case e.KeyCode
                        Case Keys.Left, Keys.Up
                            StartPage -= 1

                        Case Keys.Right, Keys.Down
                            StartPage += 1
                    End Select
                End If

                ' scroll
                Dim pt As Point = AutoScrollPosition
                Select Case e.KeyCode
                    Case Keys.Left
                        pt.X += 20
                    Case Keys.Right
                        pt.X -= 20
                    Case Keys.Up
                        pt.Y += 20
                    Case Keys.Down
                        pt.Y -= 20
                End Select
                AutoScrollPosition = New Point(-pt.X, -pt.Y)

                ' buscador por paginas
            Case Keys.PageUp
                StartPage -= 1
            Case Keys.PageDown
                StartPage += 1

                ' inicio/fin
            Case Keys.Home
                AutoScrollPosition = Point.Empty
                StartPage = 0
            Case Keys.End
                AutoScrollPosition = Point.Empty
                StartPage = PageCount - 1

            Case Else
                Return
        End Select
```

```
            'si cnseguimos estar aqui los eventos son enviados
            e.Handled = True
        End Sub

        Protected Overrides Sub OnMouseDown(ByVal e As MouseEventArgs)
            MyBase.OnMouseDown(e)
            If ((e.Button = MouseButtons.Left) AndAlso
(MyBase.AutoScrollMinSize <> Size.Empty)) Then
                Me.Cursor = Cursors.NoMove2D
                Me._ptLast = New Point(e.X, e.Y)
            End If
        End Sub

        Protected Overrides Sub OnMouseMove(ByVal e As MouseEventArgs)
            MyBase.OnMouseMove(e)
            If (Me.Cursor Is Cursors.NoMove2D) Then
                Dim dx As Integer = (e.X - Me._ptLast.X)
                Dim dy As Integer = (e.Y - Me._ptLast.Y)
                If ((dx <> 0) OrElse (dy <> 0)) Then
                    Dim pt As Point = MyBase.AutoScrollPosition
                    MyBase.AutoScrollPosition = New Point(-(pt.X + dx),
-(pt.Y + dy))
                    Me._ptLast = New Point(e.X, e.Y)
                End If
            End If
        End Sub

        Protected Overrides Sub OnMouseUp(ByVal e As MouseEventArgs)
            MyBase.OnMouseUp(e)
            If ((e.Button = MouseButtons.Left) AndAlso (Me.Cursor Is
Cursors.NoMove2D)) Then
                Me.Cursor = Cursors.Default
            End If
        End Sub

        Protected Sub OnPageCountChanged(ByVal e As EventArgs)
            RaiseEvent PageCountChanged(Me, e)
        End Sub

        Protected Overrides Sub OnPaint(ByVal e As PaintEventArgs)
            Dim img As Image = Me.GetImage(Me.StartPage)
            If (Not img Is Nothing) Then
                Dim rc As Rectangle = Me.GetImageRectangle(img)
                If ((rc.Width > 2) AndAlso (rc.Height > 2)) Then
                    rc.Offset(MyBase.AutoScrollPosition)
                    If (Me._zoomMode <> ZoomMode.TwoPages) Then
                        Me.RenderPage(e.Graphics, img, rc)
                    Else
                        rc.Width = ((rc.Width - 4) / 2)
                        Me.RenderPage(e.Graphics, img, rc)
                        img = Me.GetImage((Me.StartPage + 1))
                        If (Not img Is Nothing) Then
                            rc = Me.GetImageRectangle(img)
                            rc.Width = ((rc.Width - 4) / 2)
                            rc.Offset((rc.Width + 4), 0)
                            Me.RenderPage(e.Graphics, img, rc)
                        End If
                    End If
                End If
```

```
            End If
            e.Graphics.FillRectangle(Me._backBrush,
MyBase.ClientRectangle)
        End Sub

        Protected Overrides Sub OnPaintBackground(ByVal e As
PaintEventArgs)
        End Sub

        Protected Overrides Sub OnSizeChanged(ByVal e As EventArgs)
            Me.UpdateScrollBars()
            MyBase.OnSizeChanged(e)
        End Sub

        Protected Sub OnStartPageChanged(ByVal e As EventArgs)
            RaiseEvent StartPageChanged(Me, e)
        End Sub

        Protected Sub OnZoomModeChanged(ByVal e As EventArgs)
            RaiseEvent ZoomModeChanged(Me, e)
        End Sub

        Public Sub Print()
            Dim ps As PrinterSettings = Me._doc.PrinterSettings
            Dim first As Integer = (ps.MinimumPage - 1)
            Dim last As Integer = (ps.MaximumPage - 1)
            Select Case ps.PrintRange
                Case PrintRange.AllPages
                    Me.Document.Print()
                    Return
                Case PrintRange.Selection
                    first = Me.StartPage
                    last = Me.StartPage
                    If (Me.ZoomMode = ZoomMode.TwoPages) Then
                        last = Math.Min(CInt((first + 1)),
CInt((Me.PageCount - 1)))
                    End If
                    Exit Select
                Case PrintRange.SomePages
                    first = (ps.FromPage - 1)
                    last = (ps.ToPage - 1)
                    Exit Select
                Case PrintRange.CurrentPage
                    first = Me.StartPage
                    last = Me.StartPage
                    Exit Select
            End Select

            Dim dp = New DocumentPrinter(Me, first, last)
            dp.Print()
        End Sub

        Public Sub RefreshPreview()
            If (Not Me._doc Is Nothing) Then
                Me._img.Clear()
                Dim savePC As PrintController = Me._doc.PrintController
                Try
                    Me._cancel = False
                    Me._rendering = True
```

```
                Me._doc.PrintController = New
PreviewPrintController
                AddHandler Me._doc.PrintPage, New
PrintPageEventHandler(AddressOf Me._doc_PrintPage)
                AddHandler Me._doc.EndPrint, New
PrintEventHandler(AddressOf Me._doc_EndPrint)
                Me._doc.Print()
            Finally
                Me._cancel = False
                Me._rendering = False
                RemoveHandler Me._doc.PrintPage, New
PrintPageEventHandler(AddressOf Me._doc_PrintPage)
                RemoveHandler Me._doc.EndPrint, New
PrintEventHandler(AddressOf Me._doc_EndPrint)
                Me._doc.PrintController = savePC
            End Try
        End If
        Me.OnPageCountChanged(EventArgs.Empty)
        Me.UpdatePreview()
        Me.UpdateScrollBars()
    End Sub

    Private Sub RenderPage(ByVal g As Graphics, ByVal img As Image,
ByVal rc As Rectangle)
        rc.Offset(1, 1)
        g.DrawRectangle(Pens.Black, rc)
        rc.Offset(-1, -1)
        g.FillRectangle(Brushes.White, rc)
        g.DrawImage(img, rc)
        g.DrawRectangle(Pens.Black, rc)
        rc.Width += 1
        rc.Height += 1
        g.ExcludeClip(rc)
        rc.Offset(1, 1)
        g.ExcludeClip(rc)
    End Sub

    Private Sub SyncPageImages(ByVal lastPageReady As Boolean)
        Dim pv As PreviewPrintController =
DirectCast(Me._doc.PrintController, PreviewPrintController)
        If (Not pv Is Nothing) Then
            Dim pageInfo As PreviewPageInfo() =
pv.GetPreviewPageInfo
            Dim count As Integer = IIf(lastPageReady,
pageInfo.Length, (pageInfo.Length - 1))
            Dim i As Integer
            For i = Me._img.Count To count - 1
                Dim img As Image = pageInfo(i).Image
                Me._img.Add(img)
                Me.OnPageCountChanged(EventArgs.Empty)
                If (Me.StartPage < 0) Then
                    Me.StartPage = 0
                End If
                If ((i = Me.StartPage) OrElse (i = (Me.StartPage +
1))) Then
                    Me.Refresh()
                End If
                Application.DoEvents()
            Next i
        End If
```

```
        End Sub

        Private Sub UpdatePreview()
            If (Me._startPage < 0) Then
                Me._startPage = 0
            End If
            If (Me._startPage > (Me.PageCount - 1)) Then
                Me._startPage = (Me.PageCount - 1)
            End If
            MyBase.Invalidate()
        End Sub

        Private Sub UpdateScrollBars()
            Dim rc As Rectangle = Rectangle.Empty
            Dim img As Image = Me.GetImage(Me.StartPage)
            If (Not img Is Nothing) Then
                rc = Me.GetImageRectangle(img)
            End If
            Dim scrollSize As New Size(0, 0)
            Select Case Me._zoomMode
                Case ZoomMode.ActualSize, ZoomMode.Custom
                    scrollSize = New Size((rc.Width + 8), (rc.Height +
8))
                    Exit Select
                Case ZoomMode.PageWidth
                    scrollSize = New Size(0, (rc.Height + 8))
                    Exit Select
            End Select
            If (scrollSize <> MyBase.AutoScrollMinSize) Then
                MyBase.AutoScrollMinSize = scrollSize
            End If
            Me.UpdatePreview()
        End Sub


        ' Propiedades
        <DefaultValue(GetType(Color), "AppWorkspace")> _
        Public Overrides Property BackColor() As Color
            Get
                Return MyBase.BackColor
            End Get
            Set(ByVal value As Color)
                MyBase.BackColor = value
                Me._backBrush = New SolidBrush(value)
            End Set
        End Property

        Public Property Document() As PrintDocument
            Get
                Return Me._doc
            End Get
            Set(ByVal value As PrintDocument)
                If (Not value Is Me._doc) Then
                    Me._doc = value
                    Me.RefreshPreview()
                End If
            End Set
        End Property
```

```
<DesignerSerializationVisibility(DesignerSerializationVisibility.Hi
dden), Browsable(False)> _
    Public ReadOnly Property IsRendering() As Boolean
        Get
            Return Me._rendering
        End Get
    End Property

    <Browsable(False),
DesignerSerializationVisibility(DesignerSerializationVisibility.Hid
den)> _
    Public ReadOnly Property PageCount() As Integer
        Get
            Return Me._img.Count
        End Get
    End Property

    <Browsable(False)> _
    Public ReadOnly Property PageImages() As ListaVistaImagen
        Get
            Return Me._img
        End Get
    End Property

    <Browsable(False),
DesignerSerializationVisibility(DesignerSerializationVisibility.Hid
den)> _
    Public Property StartPage() As Integer
        Get
            Return Me._startPage
        End Get
        Set(ByVal value As Integer)
            If (value > (Me.PageCount - 1)) Then
                value = (Me.PageCount - 1)
            End If
            If (value < 0) Then
                value = 0
            End If
            If (value <> Me._startPage) Then
                Me._startPage = value
                Me.UpdateScrollBars()
                Me.OnStartPageChanged(EventArgs.Empty)
            End If
        End Set
    End Property

    <Browsable(False),
DesignerSerializationVisibility(DesignerSerializationVisibility.Hid
den)> _
    Public Property Zoom() As Double
        Get
            Return Me._zoom
        End Get
        Set(ByVal value As Double)
            If ((value <> Me._zoom) OrElse (Me.ZoomMode <>
ZoomMode.Custom)) Then
                Me.ZoomMode = ZoomMode.Custom
                Me._zoom = value
                Me.UpdateScrollBars()
```

```
                Me.OnZoomModeChanged(EventArgs.Empty)
            End If
        End Set
    End Property

    <DefaultValue(1)> _
    Public Property ZoomMode() As ZoomMode
        Get
            Return Me._zoomMode
        End Get
        Set(ByVal value As ZoomMode)
            If (value <> Me._zoomMode) Then
                Me._zoomMode = value
                Me.UpdateScrollBars()
                Me.OnZoomModeChanged(EventArgs.Empty)
            End If
        End Set
    End Property


    ' Llenado
    Private _backBrush As Brush
    Private _cancel As Boolean
    Private _doc As PrintDocument
    Private _himm2pix As PointF = New PointF(-1.0!, -1.0!)
    Private _img As ListaVistaImagen = New ListaVistaImagen
    Private _ptLast As Point
    Private _rendering As Boolean
    Private _startPage As Integer
    Private _zoom As Double
    Private _zoomMode As ZoomMode
    Private Const MARGIN As Integer = 4

    ' Tipos de documentos
    Friend Class DocumentPrinter
        Inherits PrintDocument
        ' Metodos
        Public Sub New(ByVal preview As VistaPreviaControl, ByVal
first As Integer, ByVal last As Integer)
            Me._first = first
            Me._last = last
            Me._imgList = preview.PageImages
            MyBase.DefaultPageSettings =
preview.Document.DefaultPageSettings
            MyBase.PrinterSettings =
preview.Document.PrinterSettings
        End Sub

        Protected Overrides Sub OnBeginPrint(ByVal e As
PrintEventArgs)
            Me._index = Me._first
        End Sub

        Protected Overrides Sub OnPrintPage(ByVal e As
PrintPageEventArgs)
            e.Graphics.PageUnit = GraphicsUnit.Display
            e.Graphics.DrawImage(Me._imgList.Item(Me._index),
e.PageBounds)
            Me._index = Me._index + 1
            e.HasMorePages = (Me._index <= Me._last)
```

```
            End Sub


            ' Llenado
            Private _first As Integer
            Private _imgList As ListaVistaImagen
            Private _index As Integer
            Private _last As Integer
        End Class


        Private Sub InitializeComponent()
            Me.SuspendLayout()
            '
            'VistaPreviaControl
            '
            Me.Name = "VistaPreviaControl"
            Me.ResumeLayout(False)

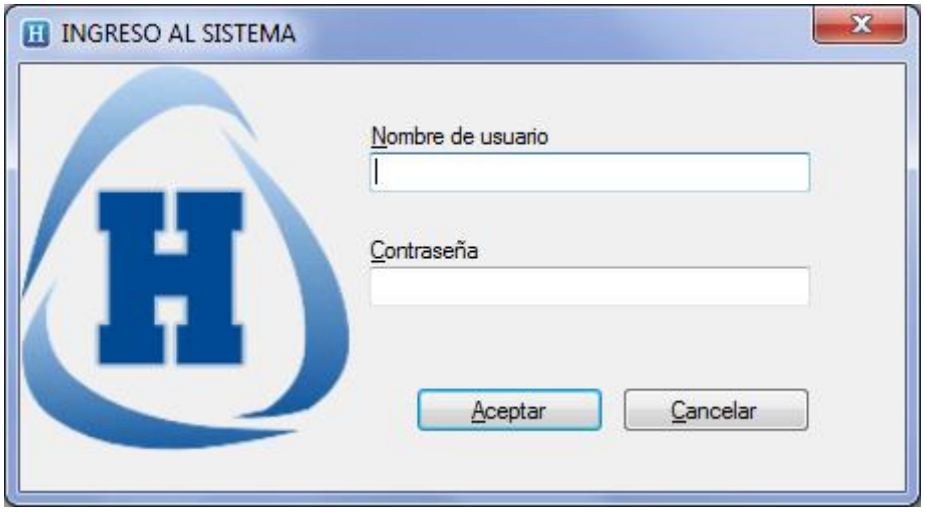
        End Sub
    End Class
```

ANEXO 4: MANUAL DE USUARIO

MANUAL DE USUARIO

INGRESO AL SISTEMA

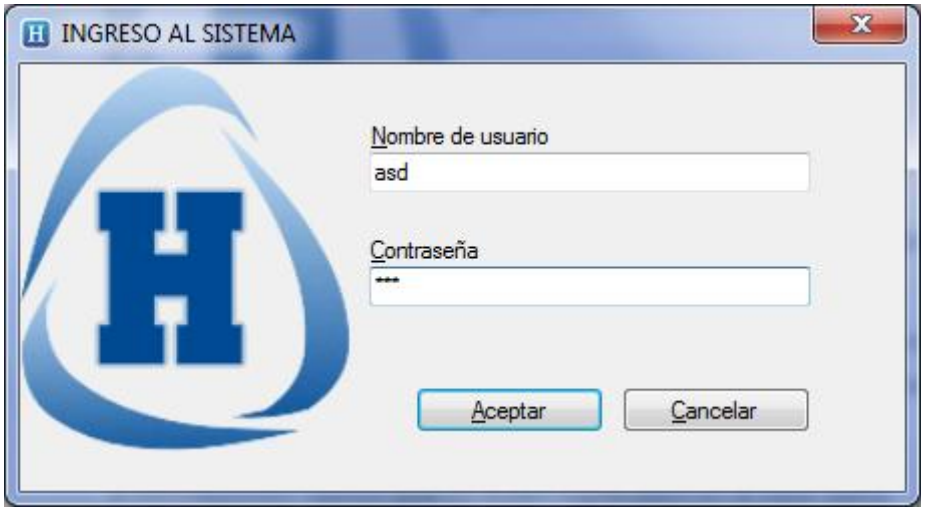
Para el ingreso al sistema debemos proporcionar un usuario y clave que esté debidamente registrado en la base de datos, en la cual tendrá tres categorías donde se puede visualizar un formulario Principal donde tendrá diferentes tipos de accesos y habilidades que puede realizar el usuario.



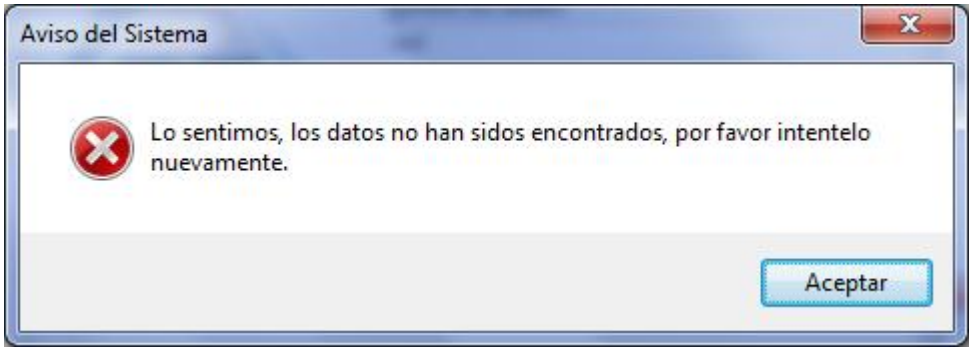
Ingreso de nombre de usuario y contraseña

Para el ingreso de usuario y contraseña se lo puede realizar tanto en escritura en minúsculas, mayúsculas o combinadas, lo cual la verificación se lo realizará en cualquier tipo de estas opciones.

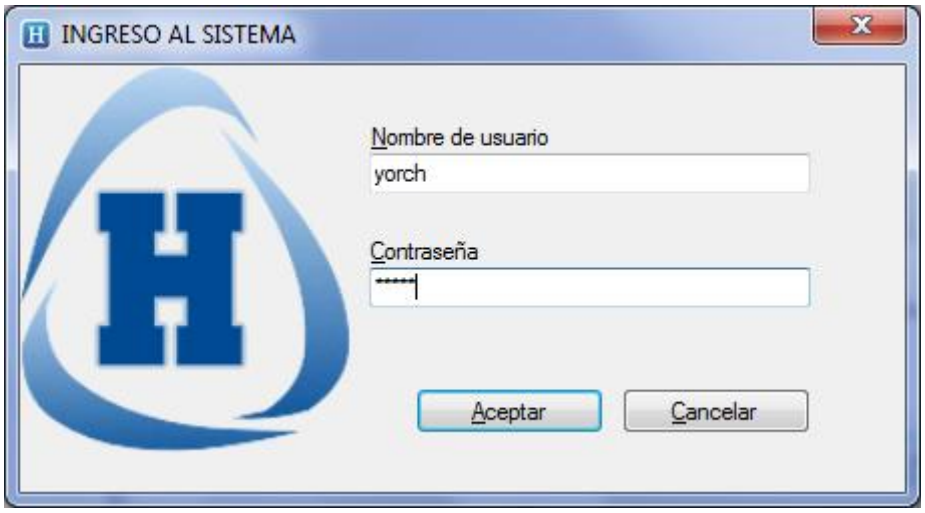
Si introducimos valores que no estén registrados en la base datos.

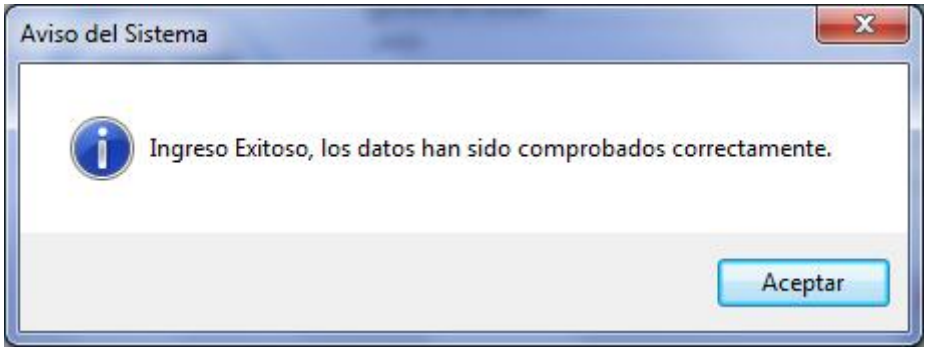


El sistema enviará un mensaje de alerta diciendo que ese usuario y clave no están registrados.



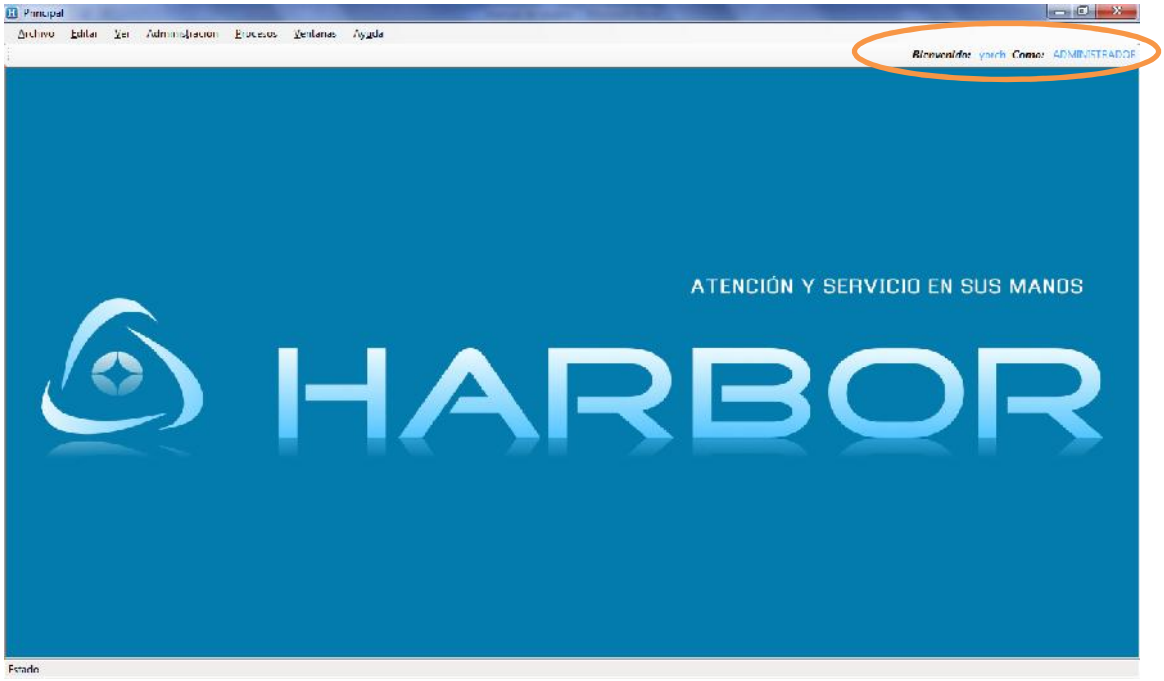
En caso contrario si introducimos el usuario y clave adecuados el sistema nos enviará un mensaje de alerta diciendo que los datos han sido verificados y tiene acceso al formulario principal dependiendo de la categoría o rol del usuario.



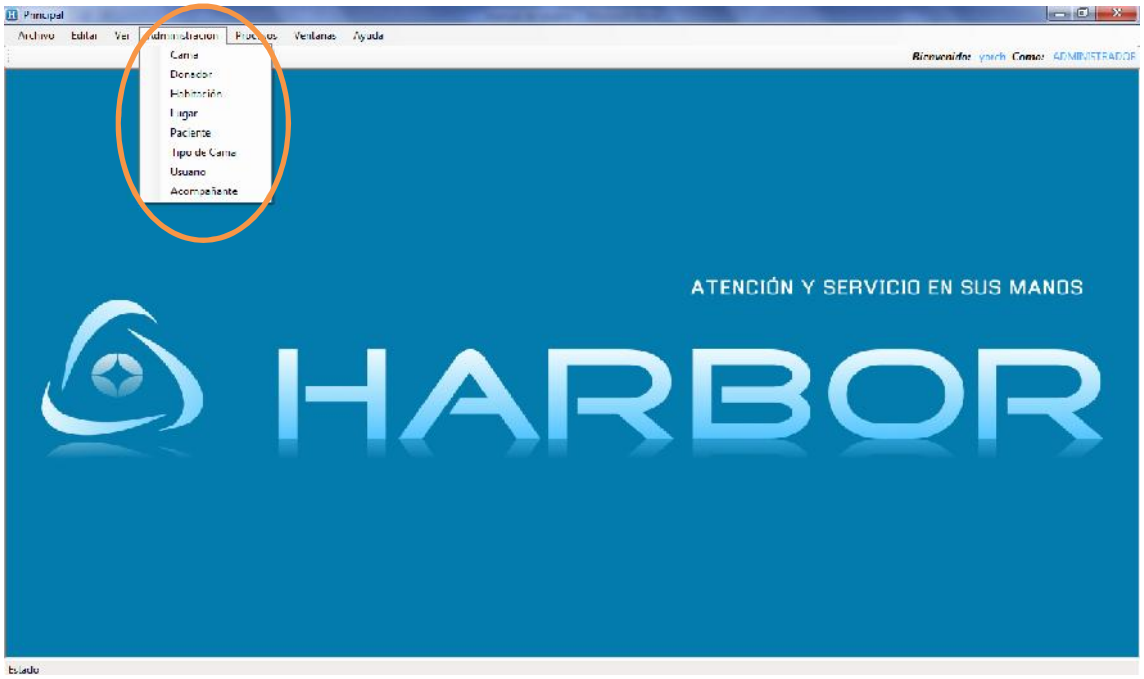


Ingreso como ADMINISTRADOR

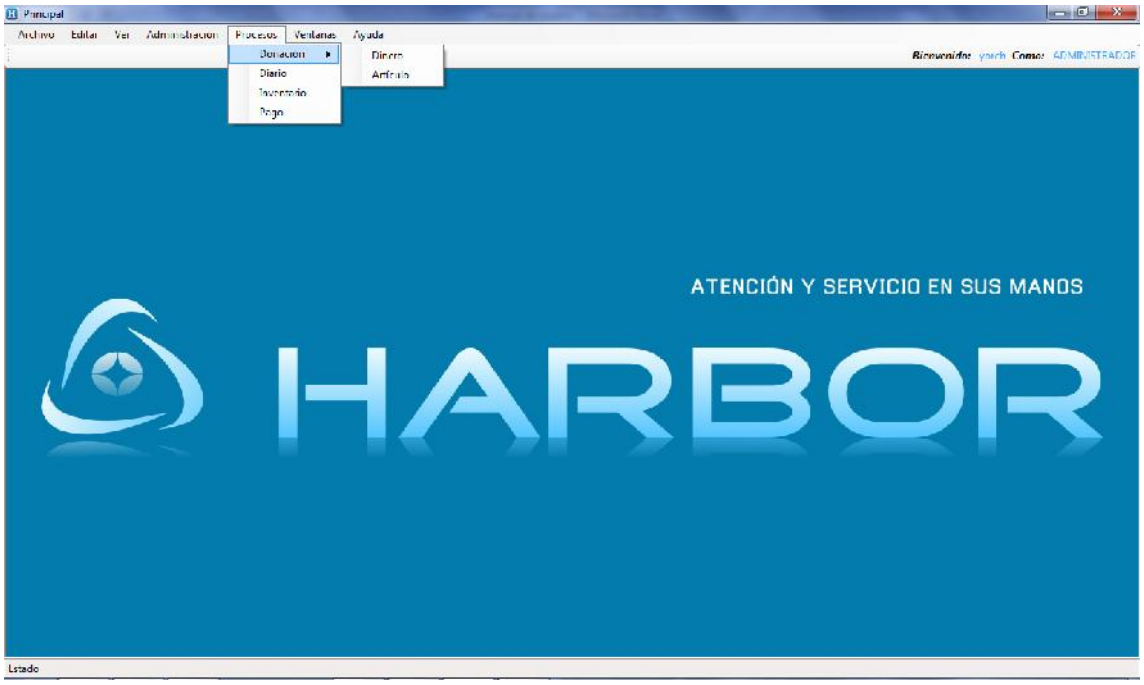
En este caso como el usuario "yorch" tiene un rol de ADMINISTRADOR, el formulario Principal tendrá todas las opciones disponibles del sistema. Además se observará en la parte superior izquierda el usuario que está registrado y el rol que ejerce.



Aquí podemos observar a detalle todas las opciones que posee administración y donde el ADMINISTRADOR puede efectuar todas las labores que realiza el sistema.

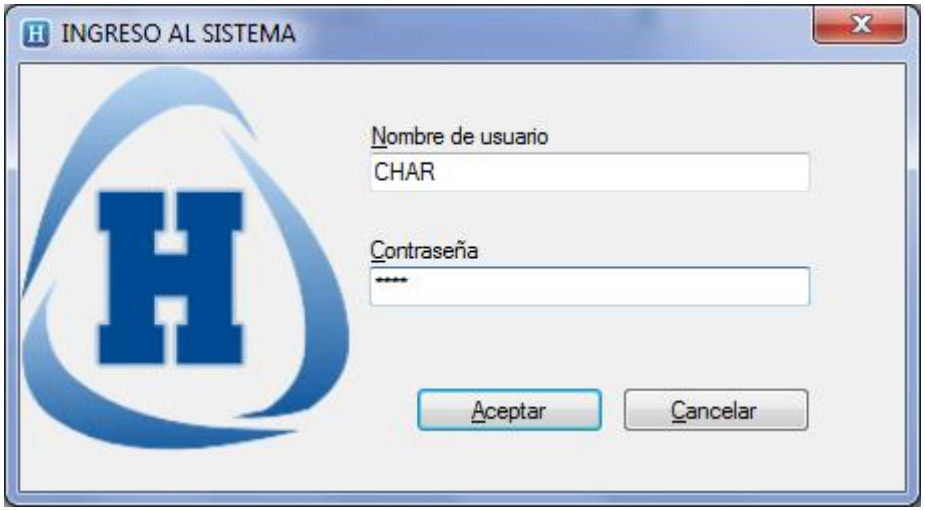


Aquí el ADMINISTRADOR puede observar y utilizar todas las opciones de los procesos involucrados en el sistema ya que es la parte fundamental de la implementación del sistema

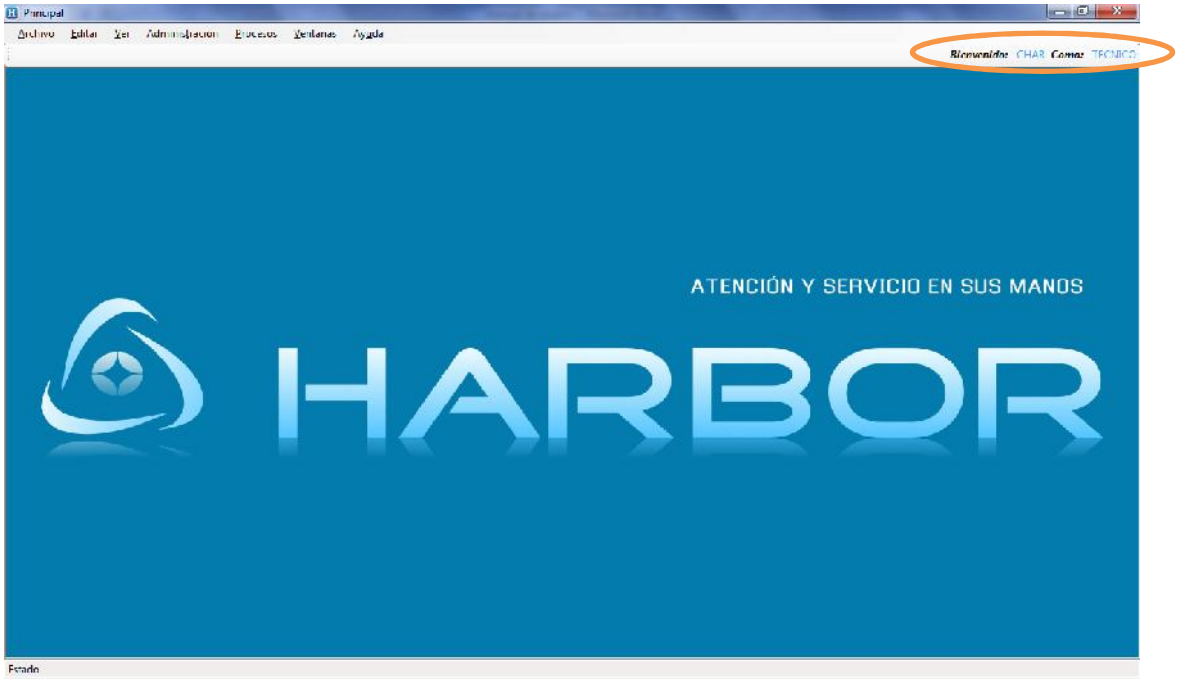


Ingreso como TÉCNICO

El ingreso del usuario que tenga el rol de técnico podrá tener todas las opciones al igual que el ADMINISTRADOR, sin embrago no tendrá la posibilidad de crear nuevos usuarios para que ingresen al sistema.



Aquí podemos observar que está el nombre del usuario ingresado y el rol q esta cumpliendo

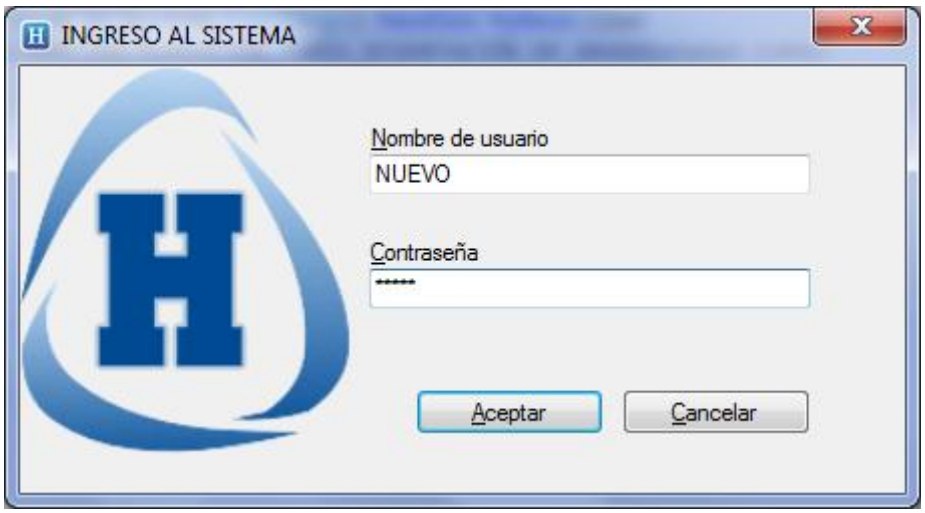


En este caso podremos ver que no está visible la opción de crear usuarios. Esto es porque la administración de usuarios se lo deja exclusivamente para los administradores del sistema siendo el caso de seguridad ante quien utilice el programa.

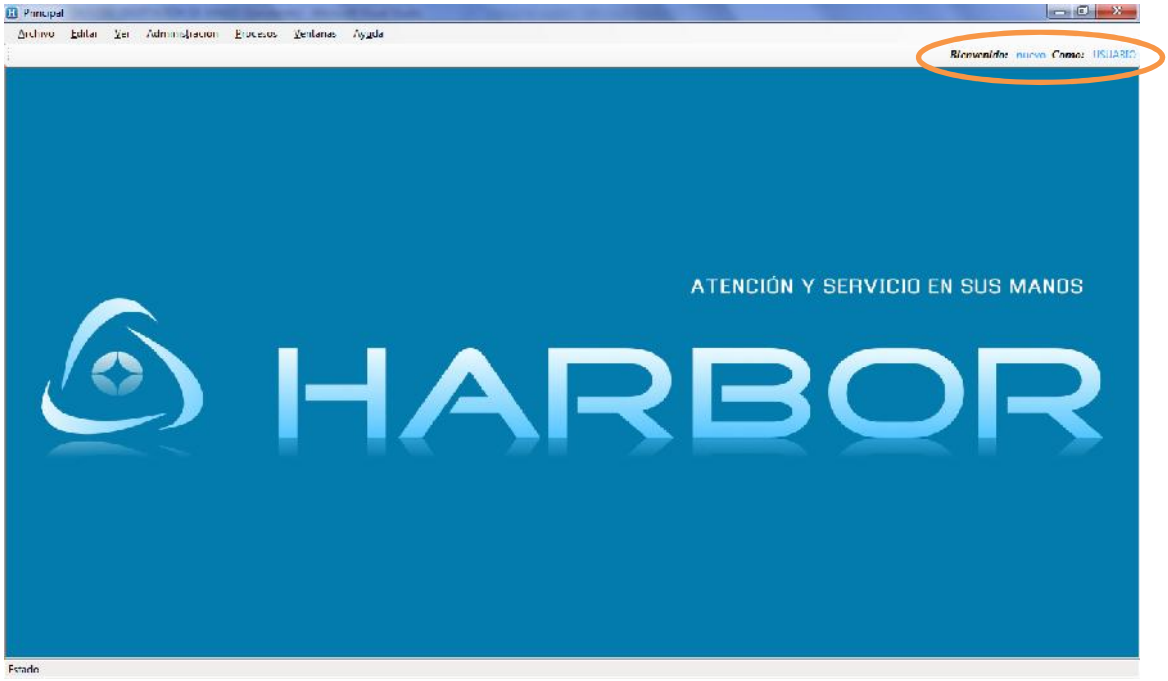


Ingreso como USUARIO

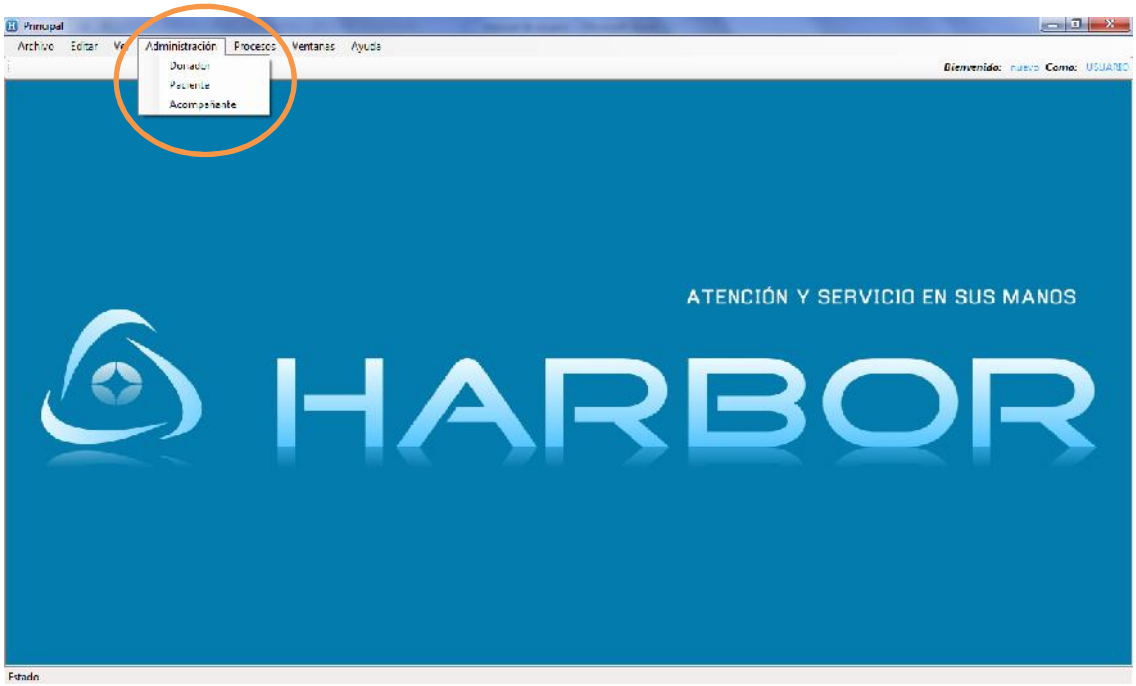
En el rol de usuario vamos a tener menos opciones de creación de algunas opciones ya que el rol USUARIO, utiliza más los procesos que las mismas administraciones.



Aquí podemos observar el nombre del usuario registrador y el rol ocupado en este caso USUARIO.



Como era de suponer el menú de Administración no tiene todas sus opciones.

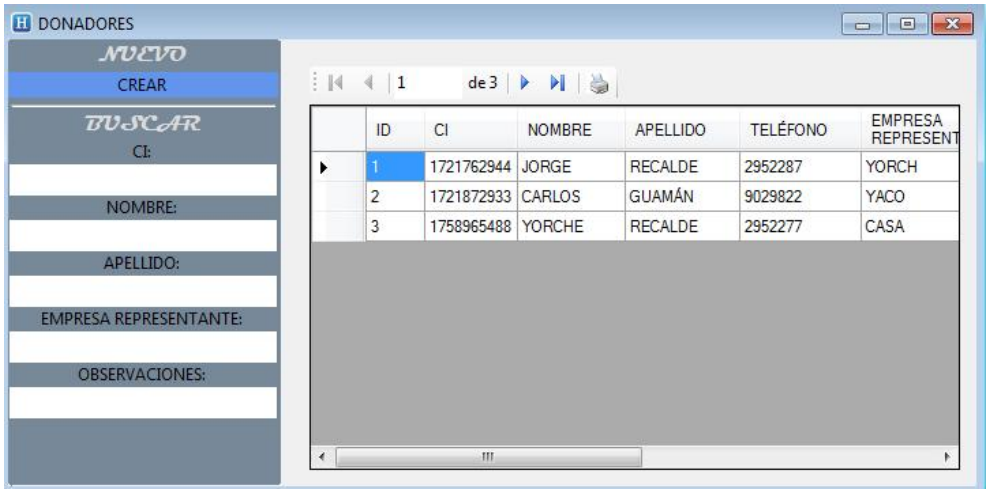


ADMINISTRACIONES

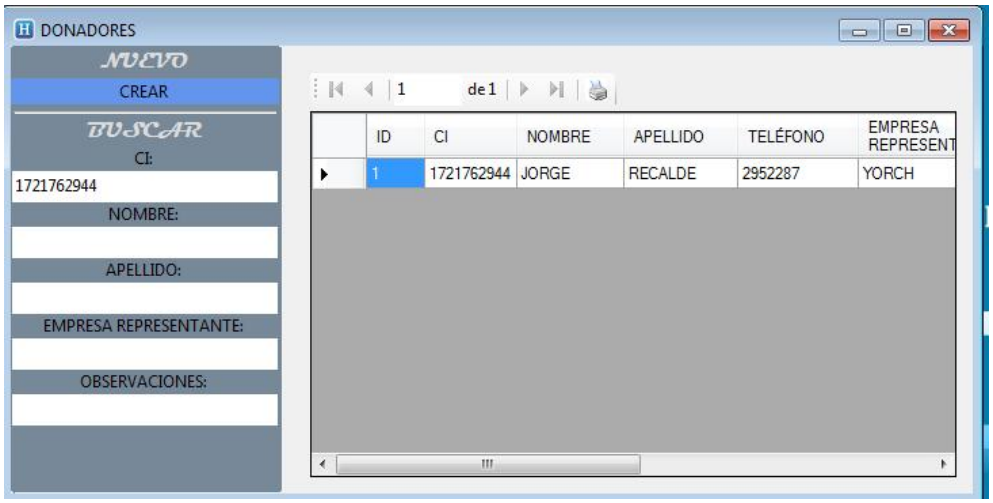
Los donadores serán quienes dan a la Fundación algún tipo de ayuda ya sea económica o de artículos de primera necesidad para la misma.

Búsqueda de donador

En el formulario donador podemos observar todos los donadores registrados en la bases de datos del sistema, pero para una mejor ayuda podemos realizar búsquedas de cualquier tipo ya sea CI, nombre, apellido, empresa o alguna observación del donador

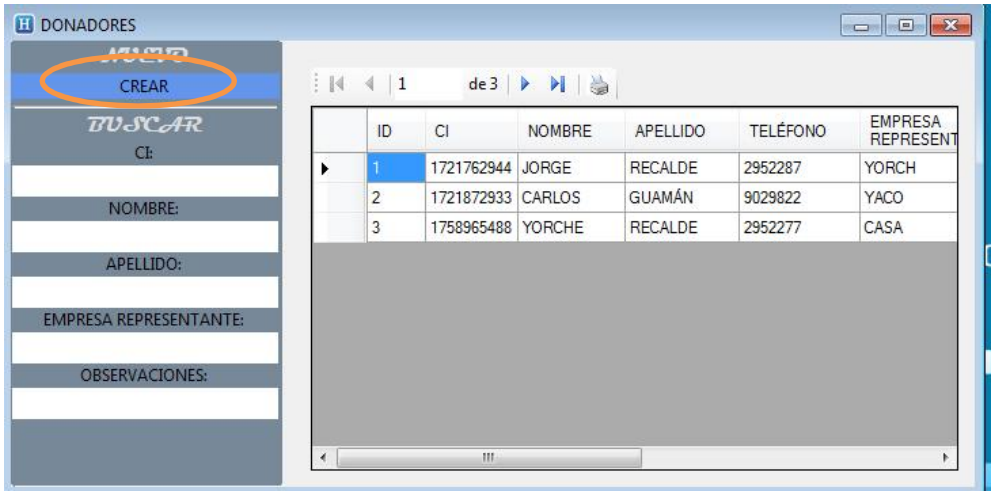


Si se realiza por estos parámetros el sistema irá escogiendo la búsqueda más cercana a la buscada y se presentará solo los datos del donador buscado. No habrá necesidad de utilizar de botones ya que mientras digita su cédula o escribe los parámetros estos se irán buscando solos.

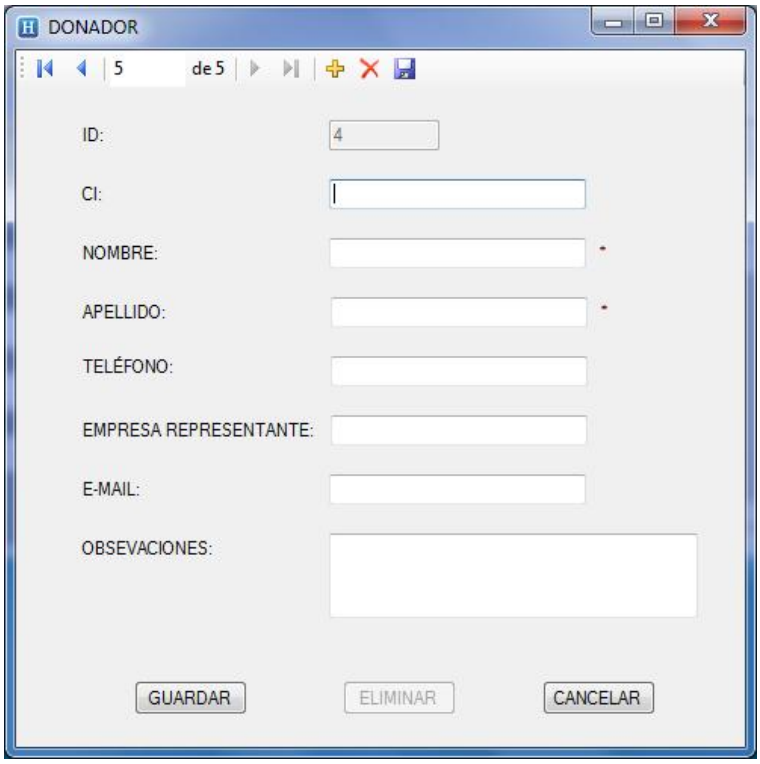


Creación de Donador

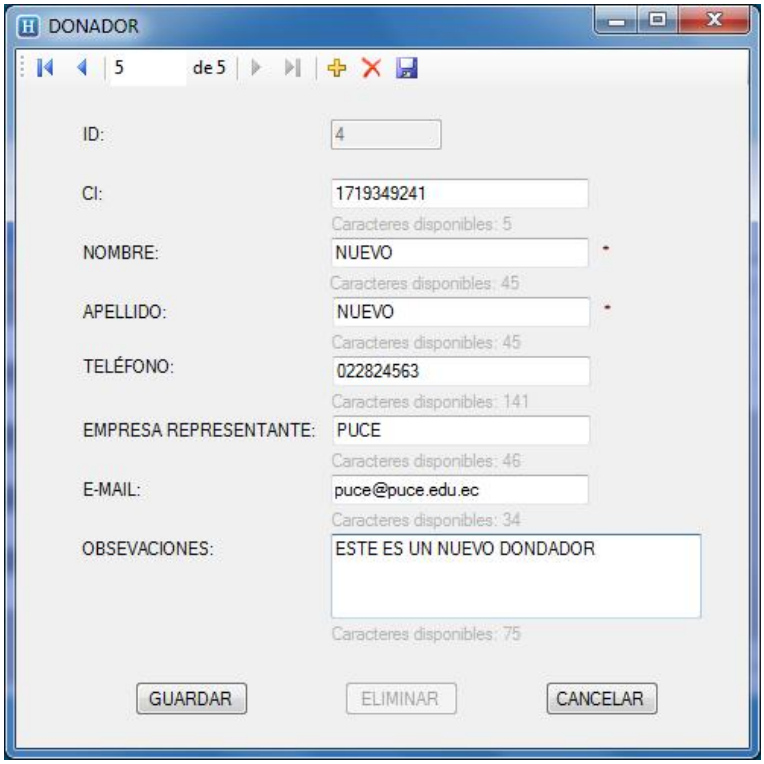
Para esto debemos presionar en el botón crear para trasladarnos a otro formulario.



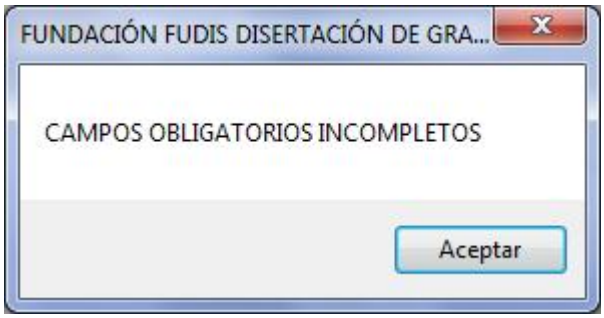
Al presionar el botón crear abriremos otro formulario donde podemos llenar a detalle los diferentes campos del nuevo donador

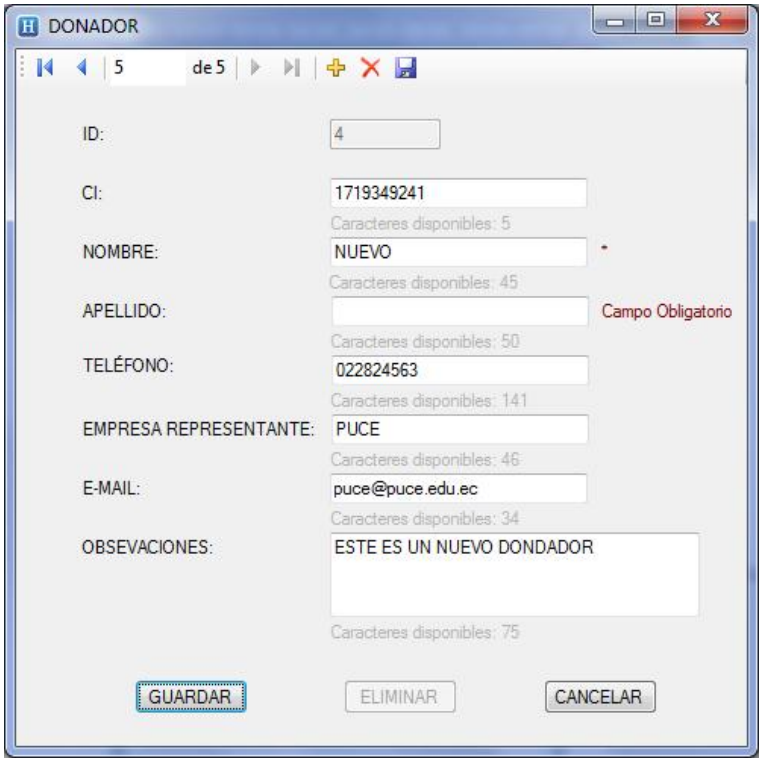


En este nuevo formulario todos los campos estarán verificando de acuerdo al tipo de ingreso ya sea número, solo caracteres alfanuméricos o todos en un solo cuadro de texto.

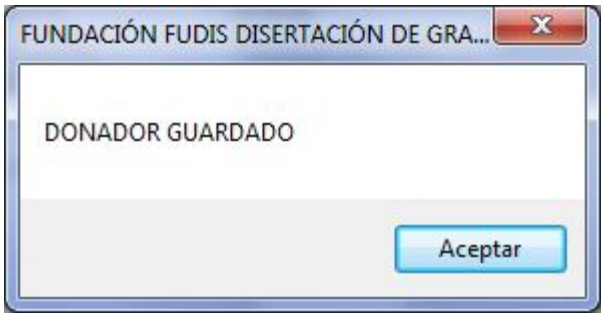


Si existiese algún campo necesario para el registro el sistema muestra un mensaje de alerta de los campos obligatorios a ser llenados.

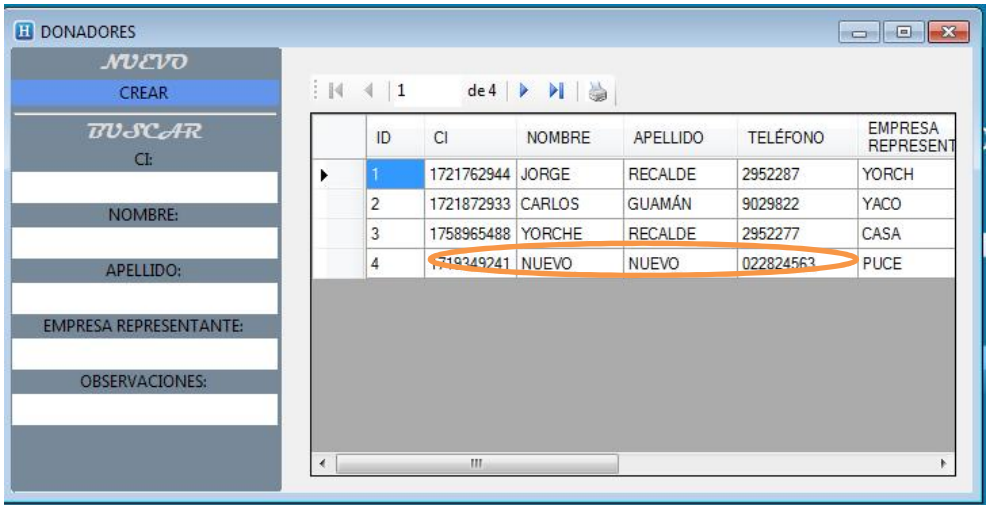




Una vez corregido los errores procedemos a presionar el botón guardar donde se almacenarán el nuevo registro donador.

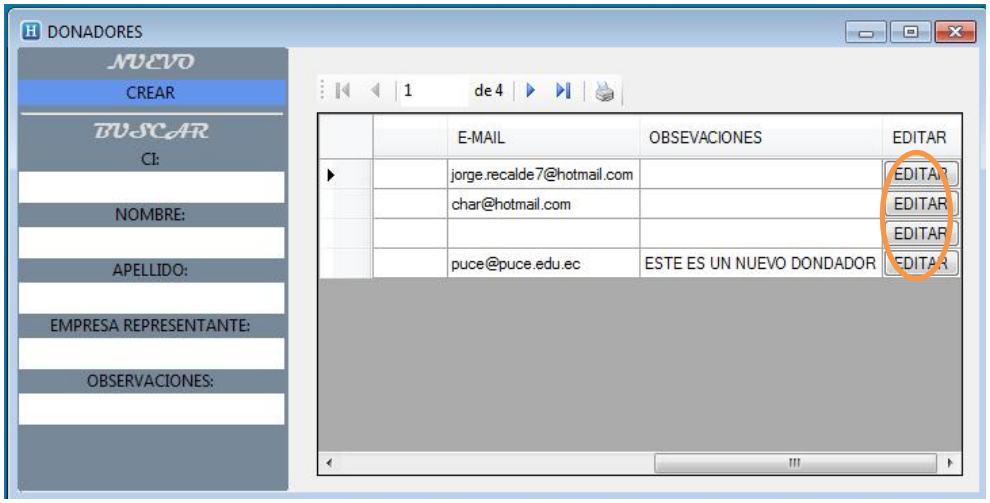


Después de esto regresamos al formulario de donador donde podemos observar en nuestra tabla en nuevo donador ingresado.

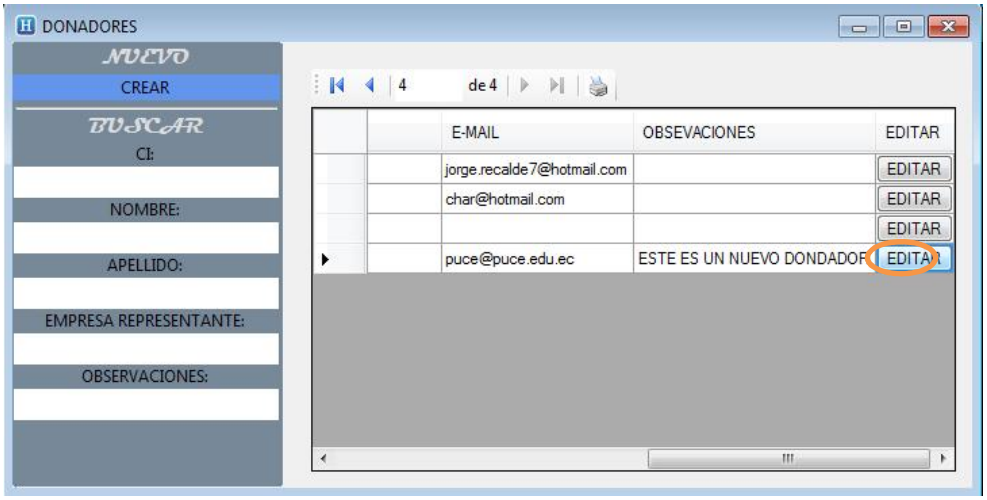


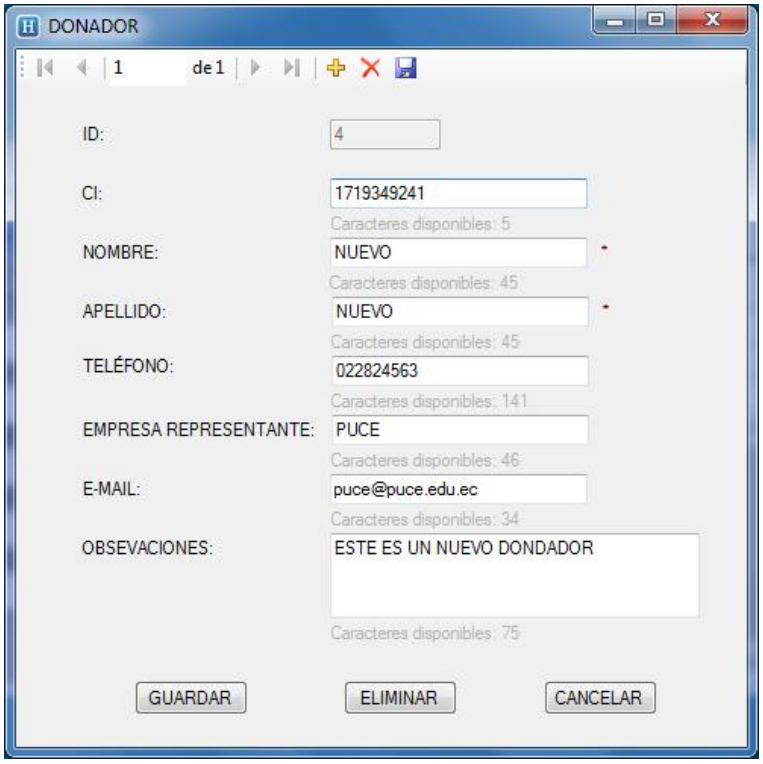
Edición de Donador

Para la edición de un registro procedemos buscar en nuestra grilla o tabla la opción editar que se encuentra generalmente al inicio o al final de la grilla.

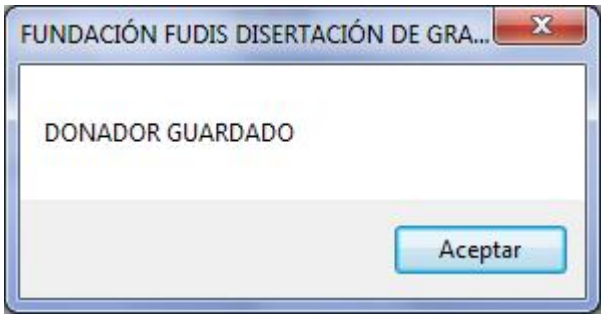


Presionamos el botón y abriremos otro formulario donde podemos editar la información del registro que deseemos.

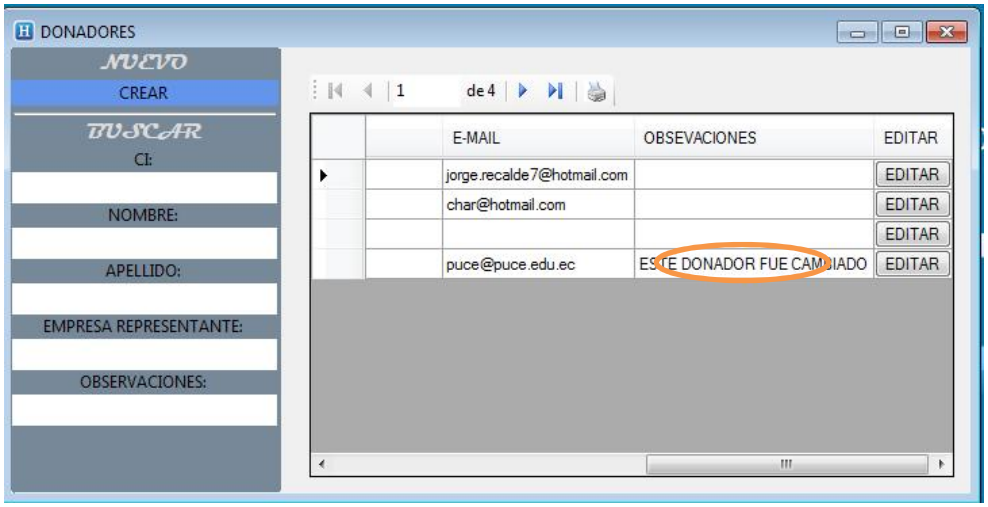




Para el caso modificaremos las observaciones cambiando “ESTE ES UN NUEVO DONADOR” por “ESTE DONADOR FUE CAMBIADO” y mostrará mensaje de confirmación.

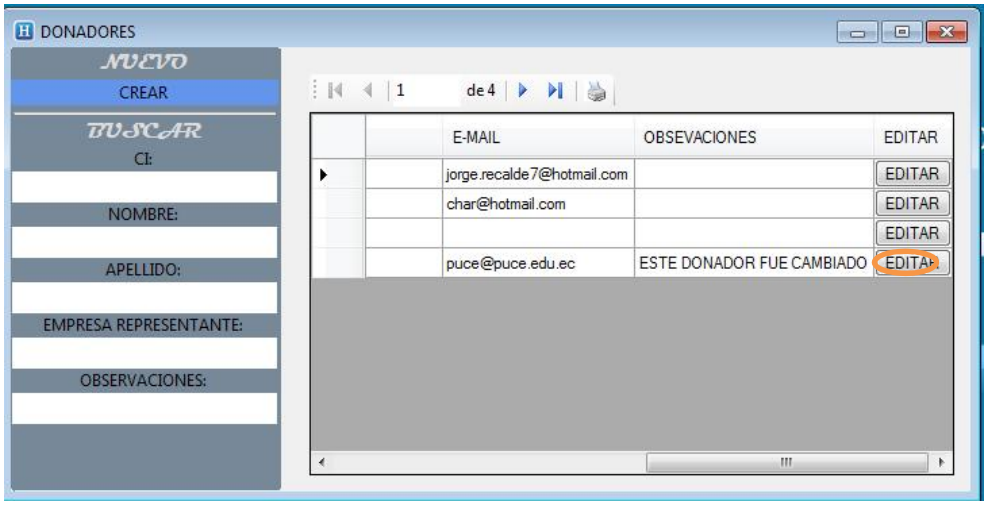


Como podemos observar aquí ya están los cambios realizados en el registro

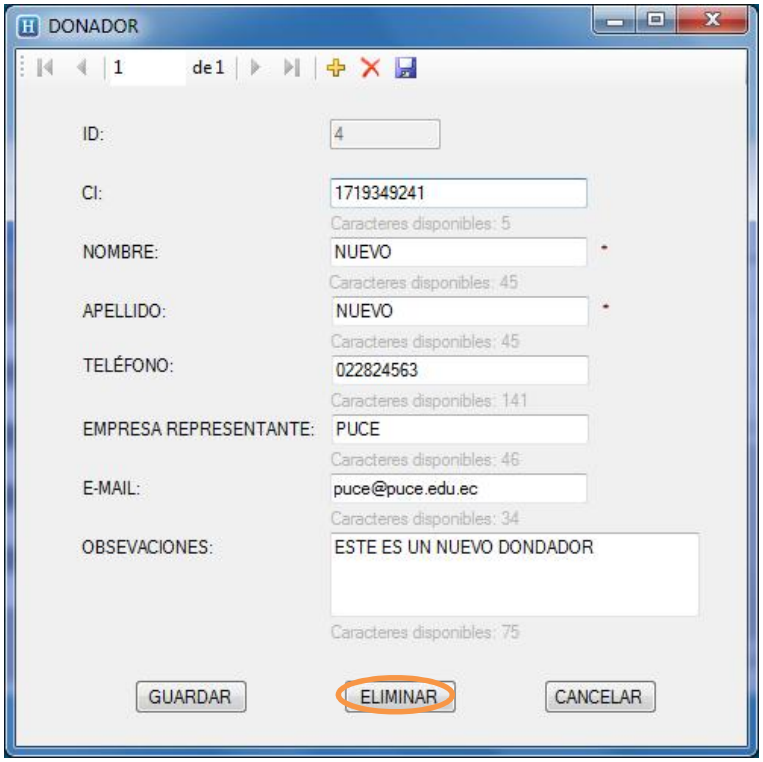


Eliminación de Donador

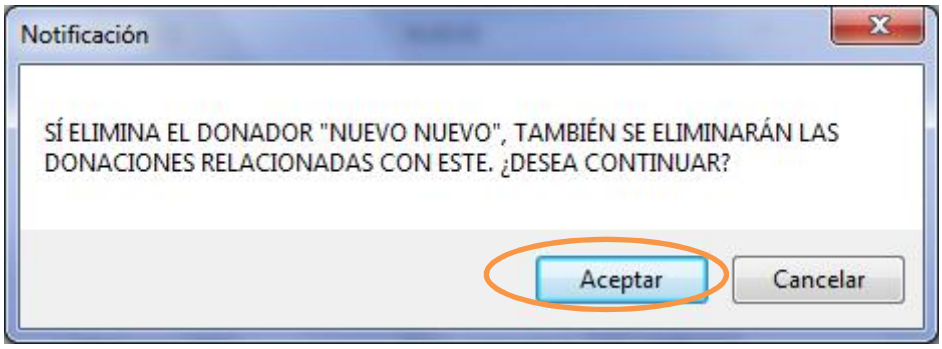
Para eliminar los registros existente debemos ir al igual que editar a buscar el botón editar para poder ir al formulario de eliminación.



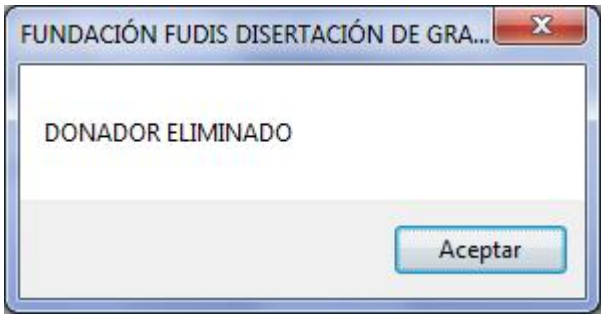
Aquí presionamos el botón eliminar para que se efectúe la acción.



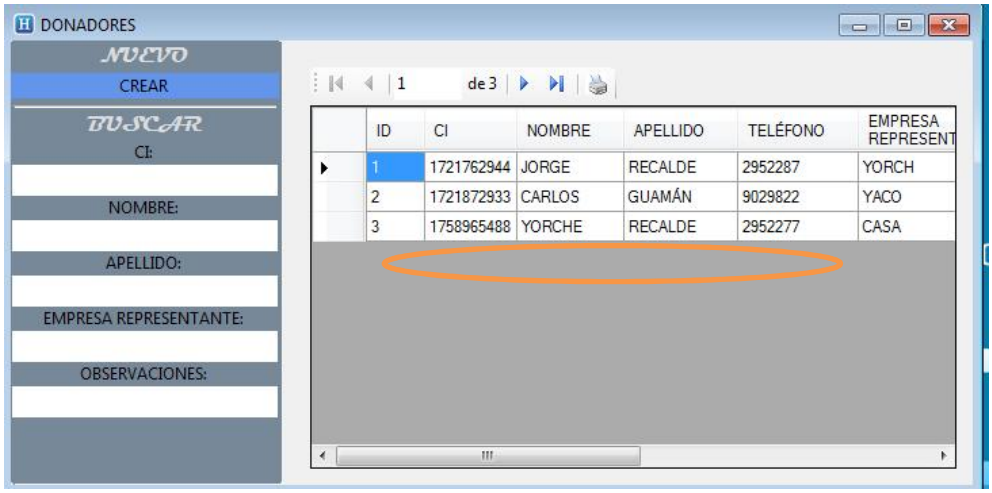
Cuando fue presionado el botón eliminar se presentará un mensaje de confirmación de la acción a realizarse.



Si presionamos Aceptar nos mostrará un aviso del sistema.



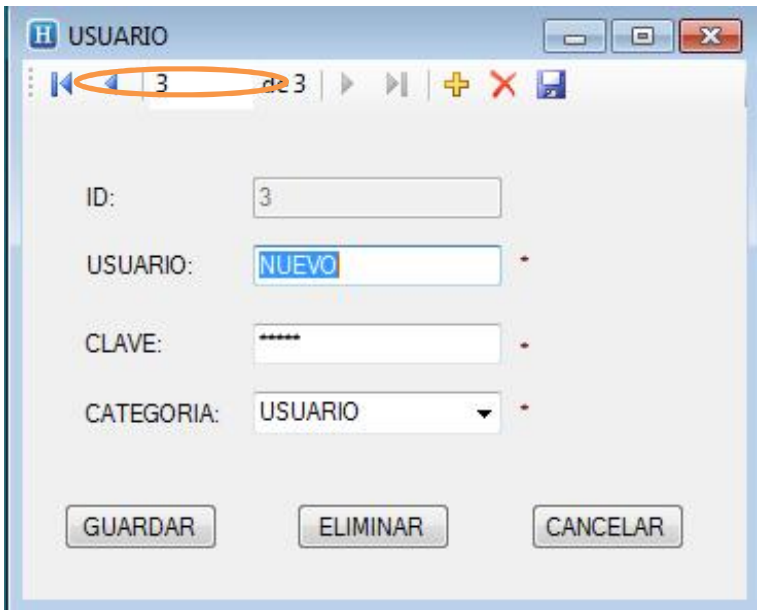
En el formulario de donador podremos ver la eliminación realizada.

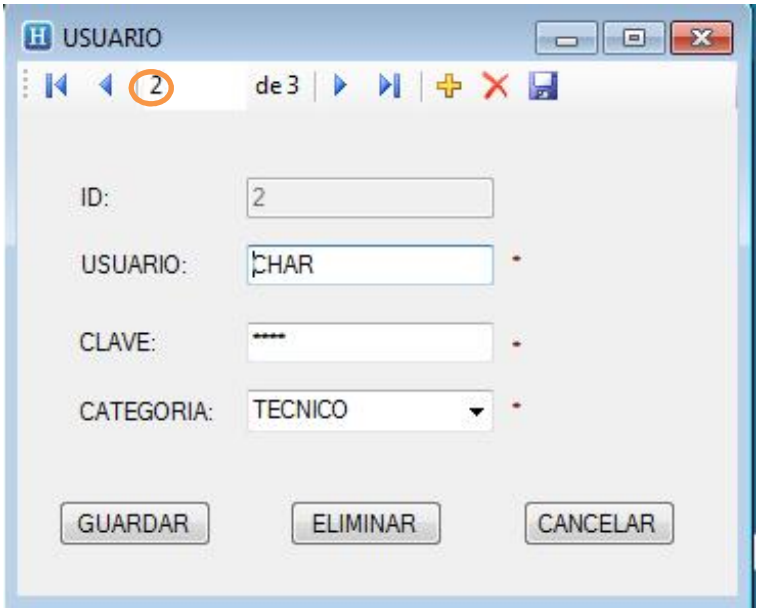


Todas las administraciones son similares a esta administración sin embargo la opción de Usuario se diferencia de las otras ya que en este caso solo se pueden administrar algunos registros.

Búsqueda de Usuario

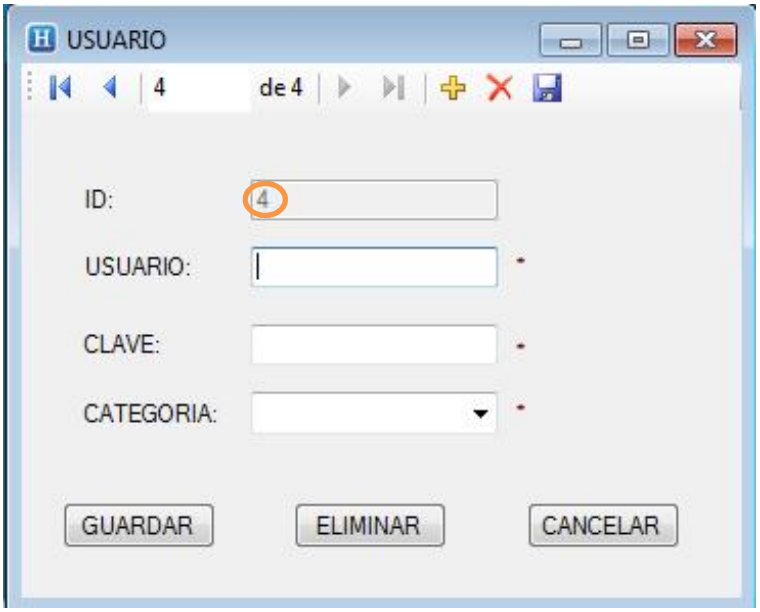
Para la búsqueda de usuarios se lo realizará mediante el uso de la barra ToolStrip donde podemos desplazarnos por cada uno de los registros almacenados en la base de datos



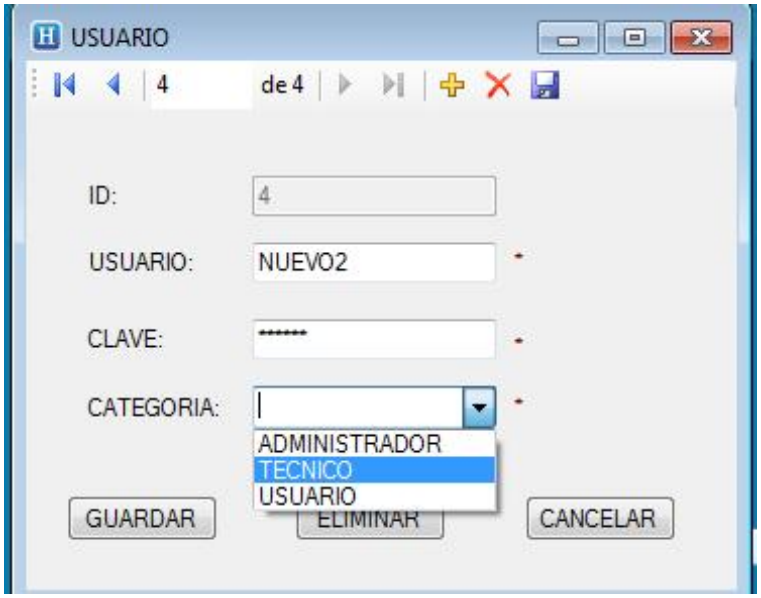


Creación de Usuario

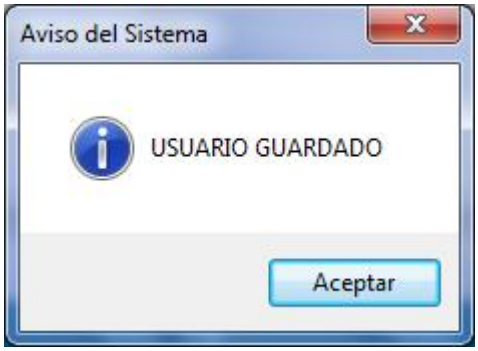
Para esto debemos presionar en el botón donde está la imagen  en la cual se nos creará un nuevo ID para el nuevo registro y este será auto numérico.



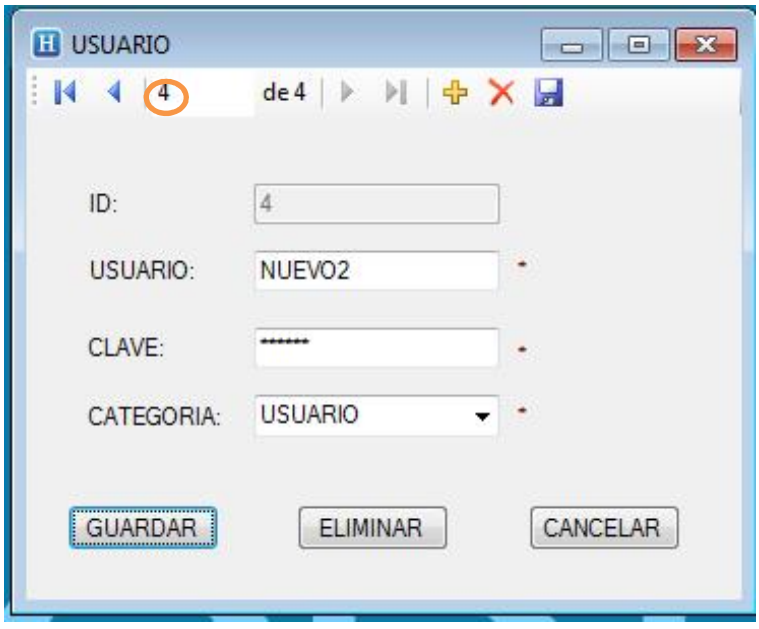
Aquí procedemos a ingresar los datos del nuevo registro y debemos poner el rol que va a cumplir en el manejo del sistema



Presionamos guardar y nos mostrará un aviso de aceptación del sistema de que ingreso fue exitoso.

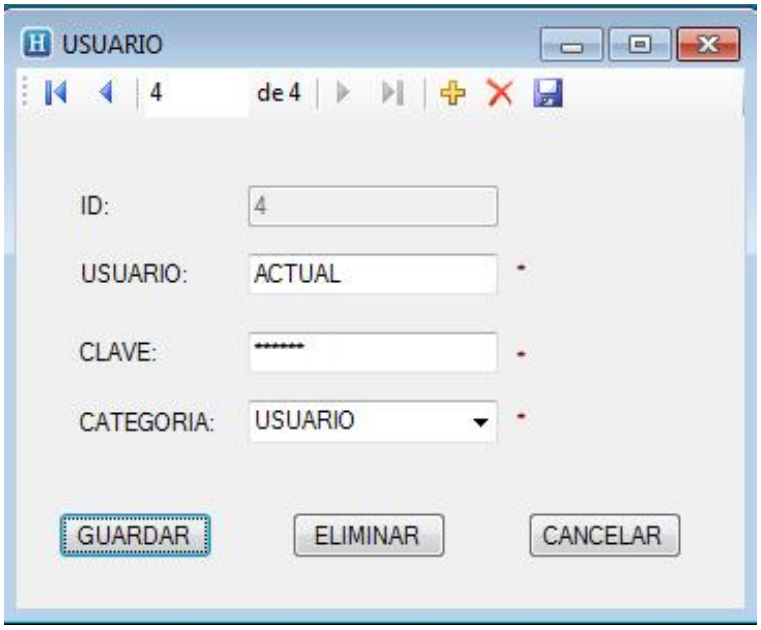


Aquí ya podemos ver el nuevo registro como parte de la base de datos.

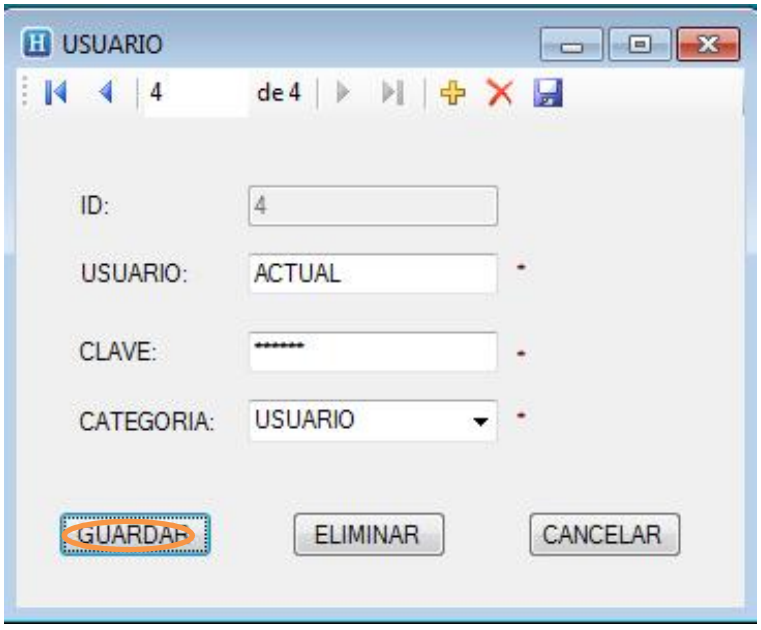


Edición de Usuario

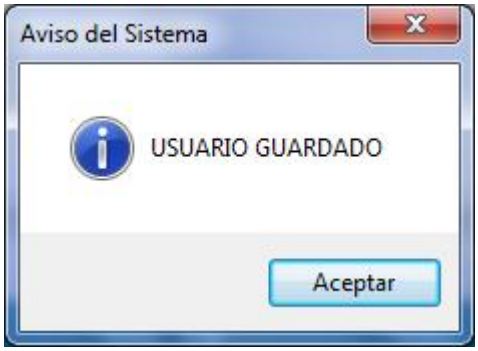
En el mismo formulario podemos modificar en este caso modificaremos el usuario “NUEVO2” por “ACTUAL”.

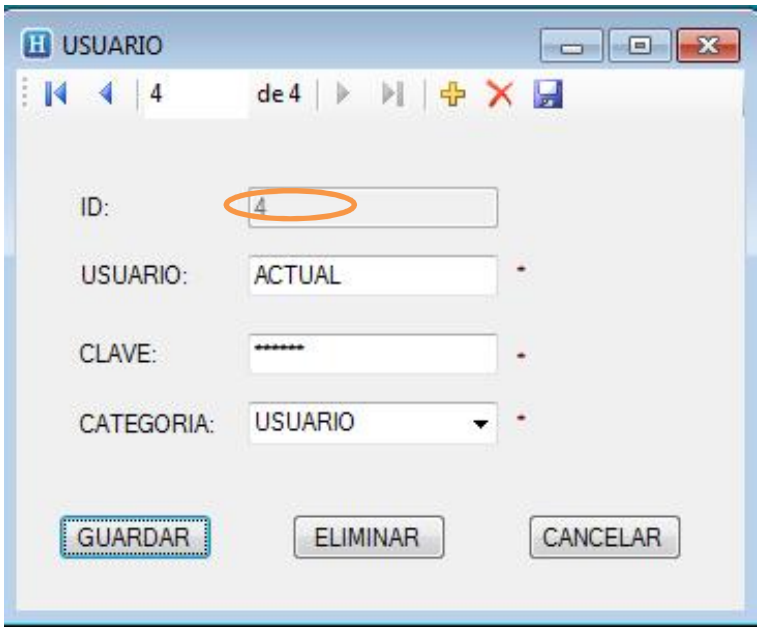


Presionamos el botón guardar para que ejecute los cambios.



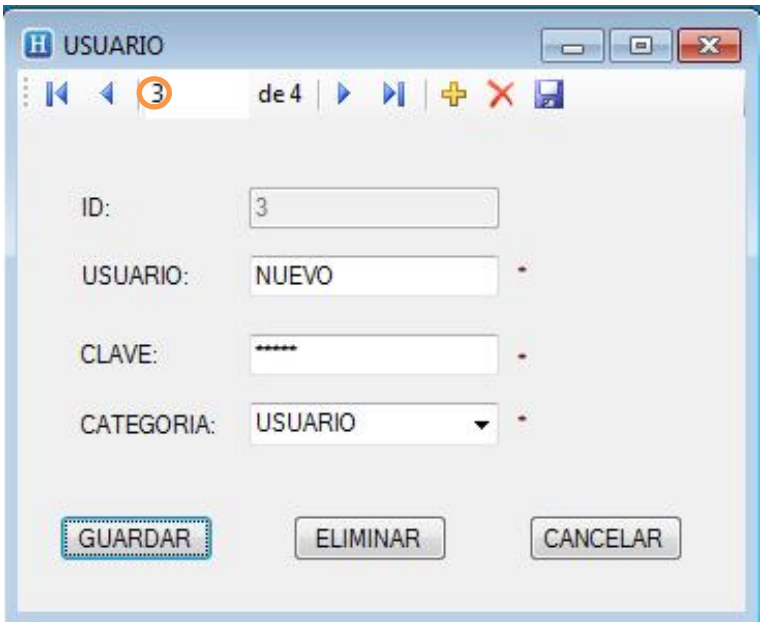
Esto nos mostrará un mensaje de guardado los cambios en el registro.

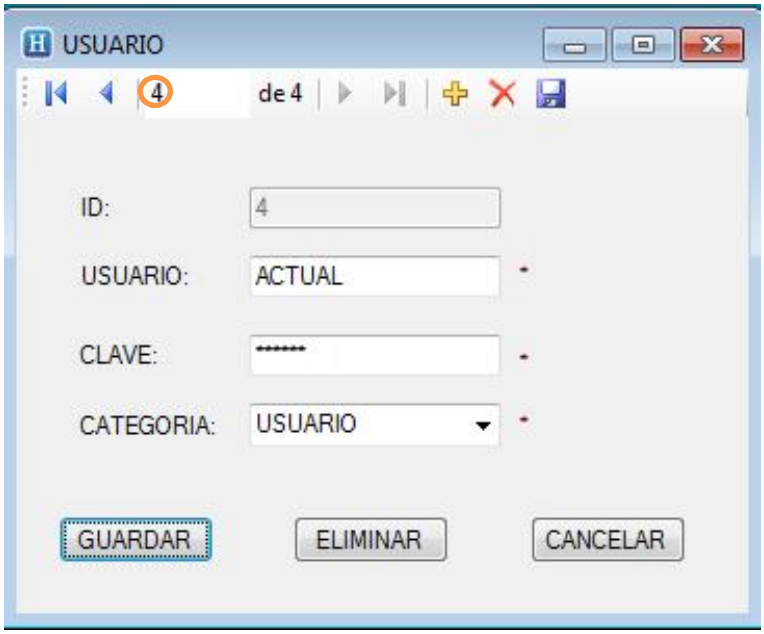




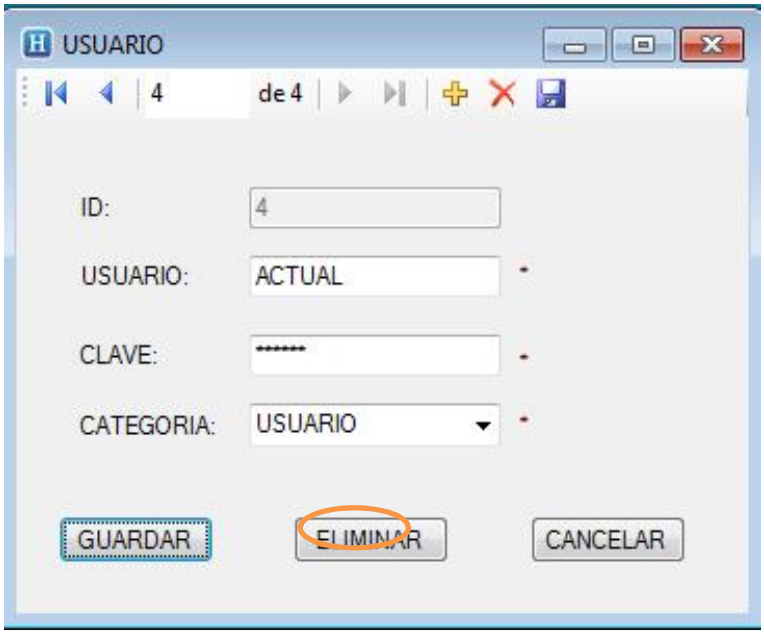
Eliminación de Usuario

En el mismo formulario buscamos el registro que vamos a eliminar y presionamos el botón eliminar.

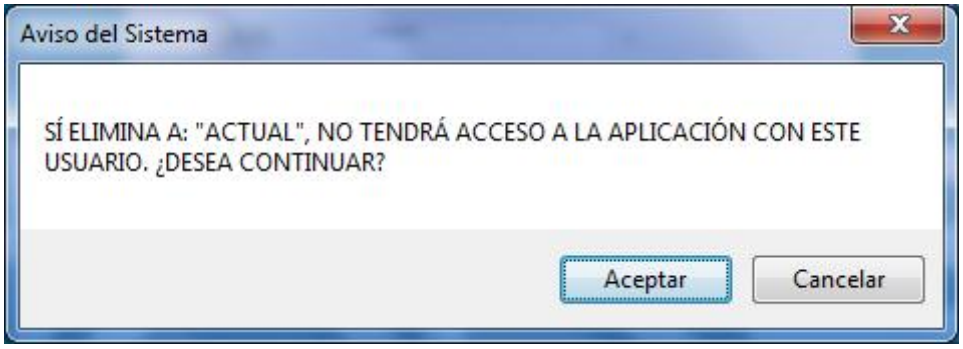




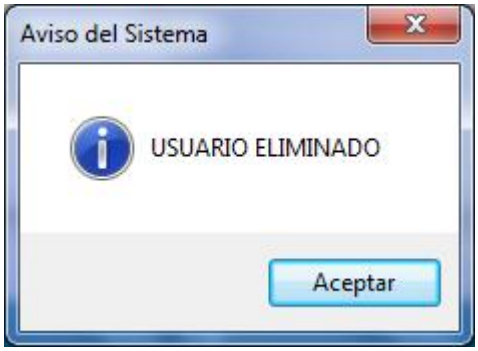
En este caso vamos a eliminar el último registro que insertamos.



Una vez que presionamos el botón eliminar nos presenta un aviso de sistema si estamos seguros de eliminar este registro.



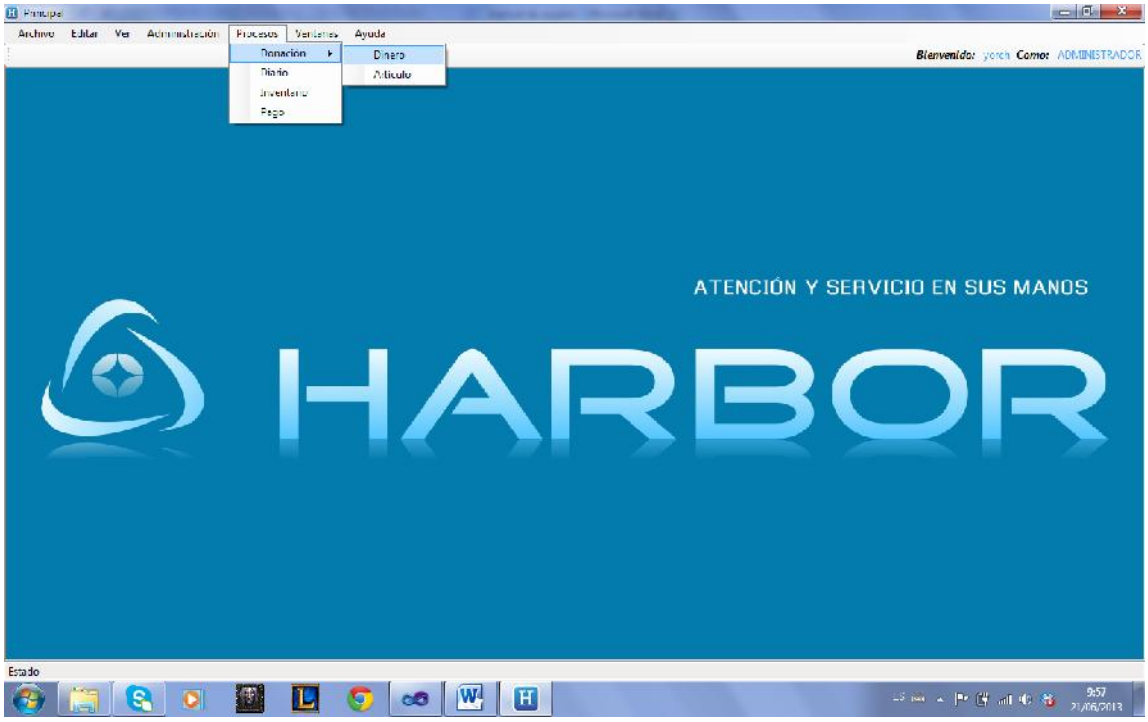
Presionamos Aceptar y realizará la eliminación y nos mostrará el mensaje de validación.



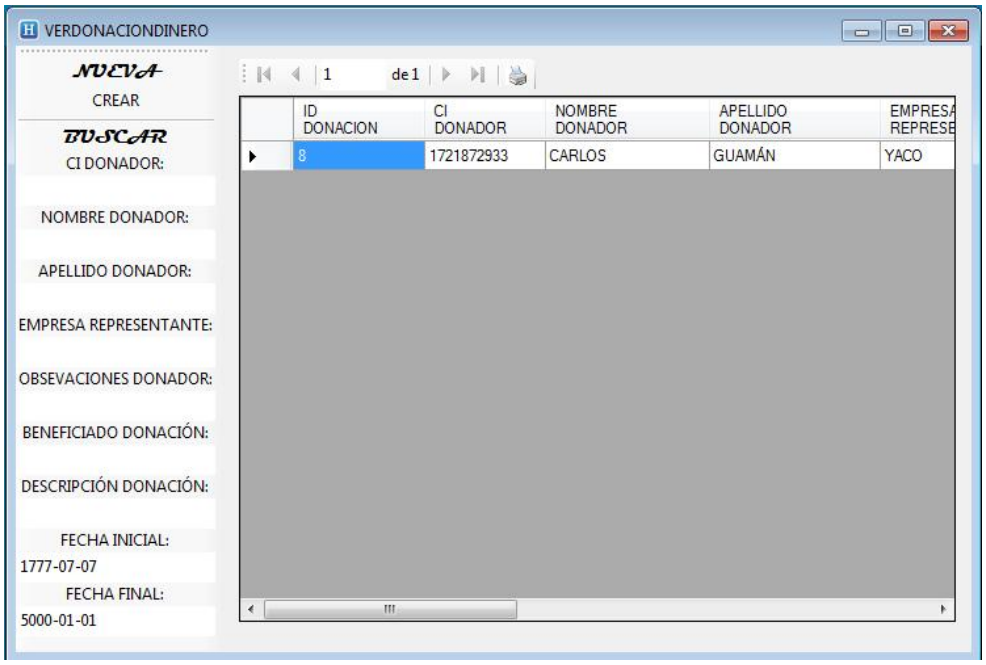
PROCESOS

Búsqueda de Donaciones de dinero

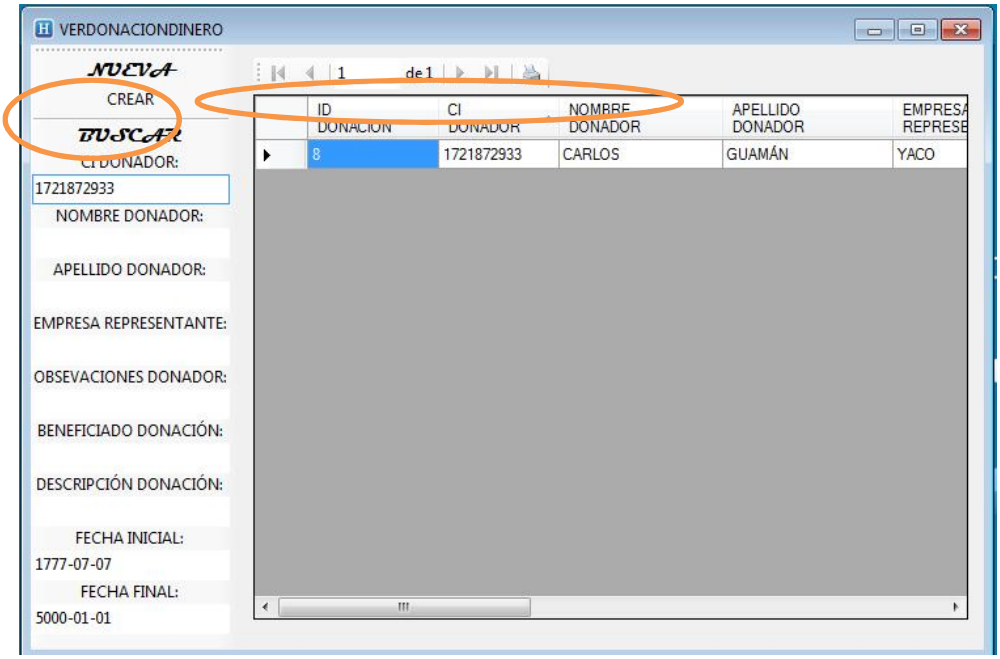
Para utilizar el proceso de Donación debemos ir al menú de opciones del formulario principal y presionar en la opción Procesos, donde se desplegará un submenú, ahí escogemos la opción Donación y elegimos la opción Dinero.



Aquí nos mostrará el formulario de Donaciones

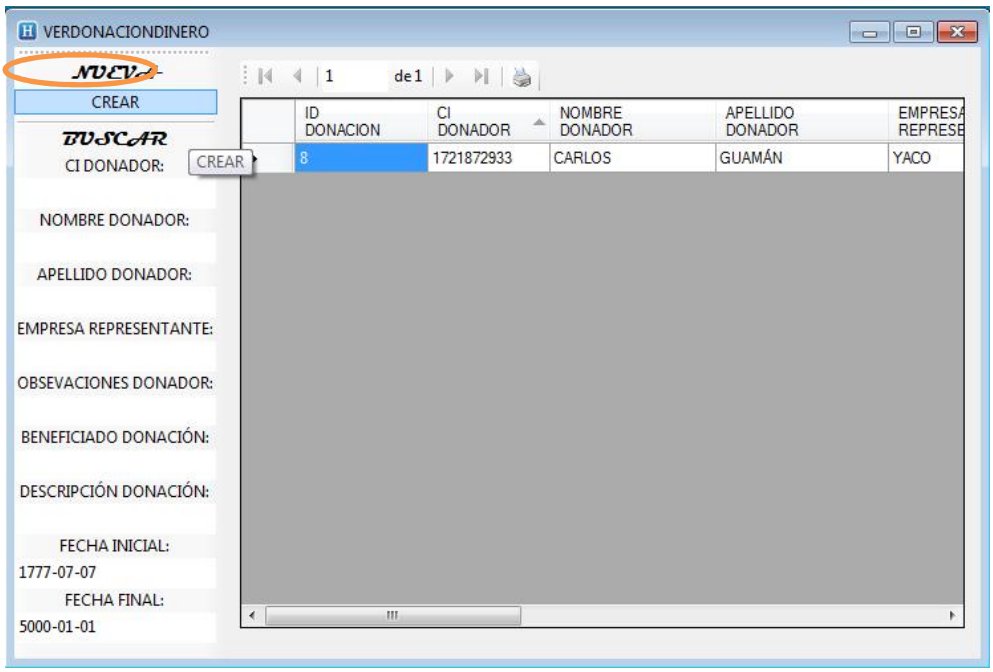


En este formulario podemos observar las donaciones realizadas hasta la fecha y podemos buscar alguna donación de acuerdo a los parámetros que deseemos.

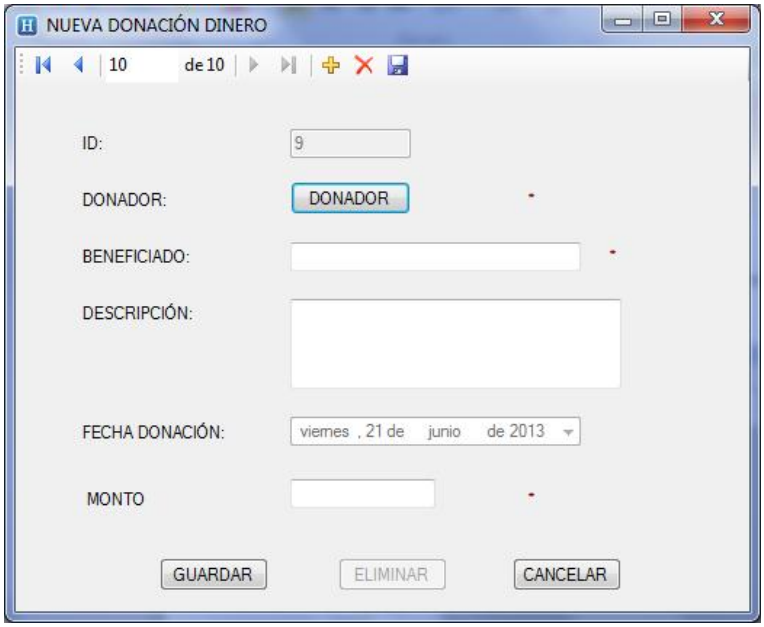


Crear Donación de Dinero a FUDIS

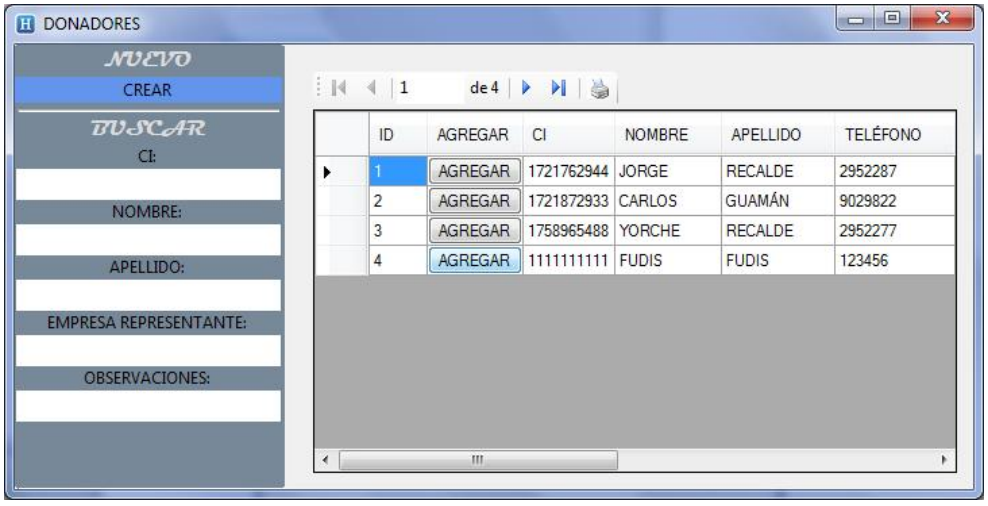
Para crear o registrar una nueva donación de dinero debemos presionar en el botón crear donde nos abrirá un nuevo formulario.



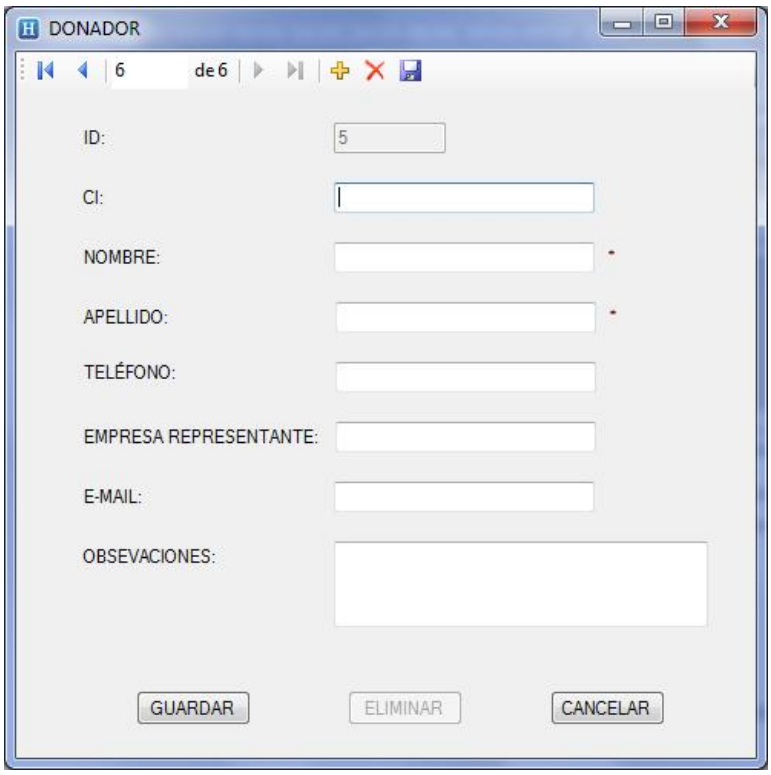
En el formulario de nuevo registro podemos ingresar la donación a quién nosotros queramos ya sea, a la misma fundación o a otras fundaciones que se desee donar.



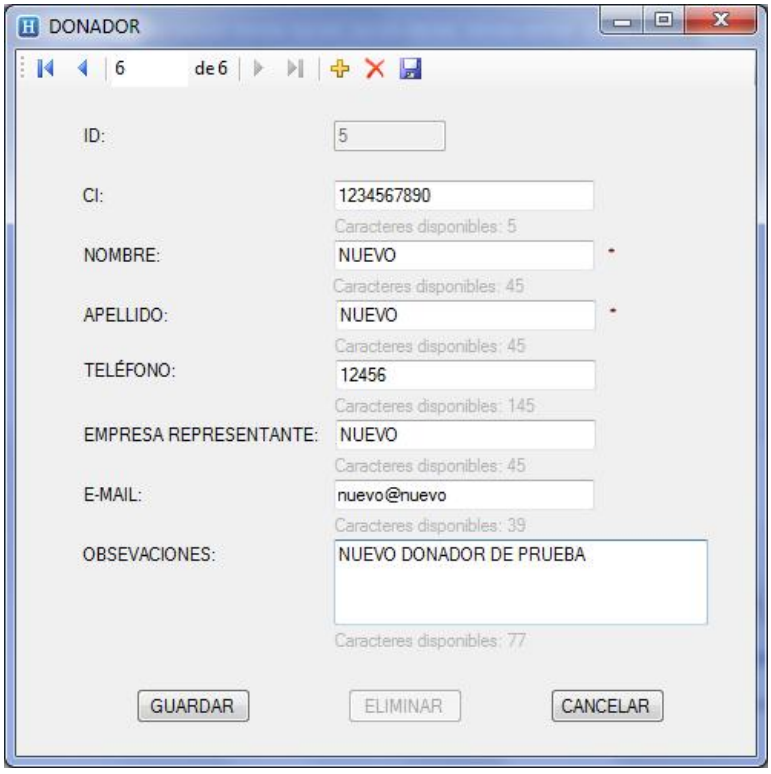
Para saber el donador que realiza la donación debemos presionar el botón DONADOR y nos abrirá otro formulario donde escogeremos el donador.



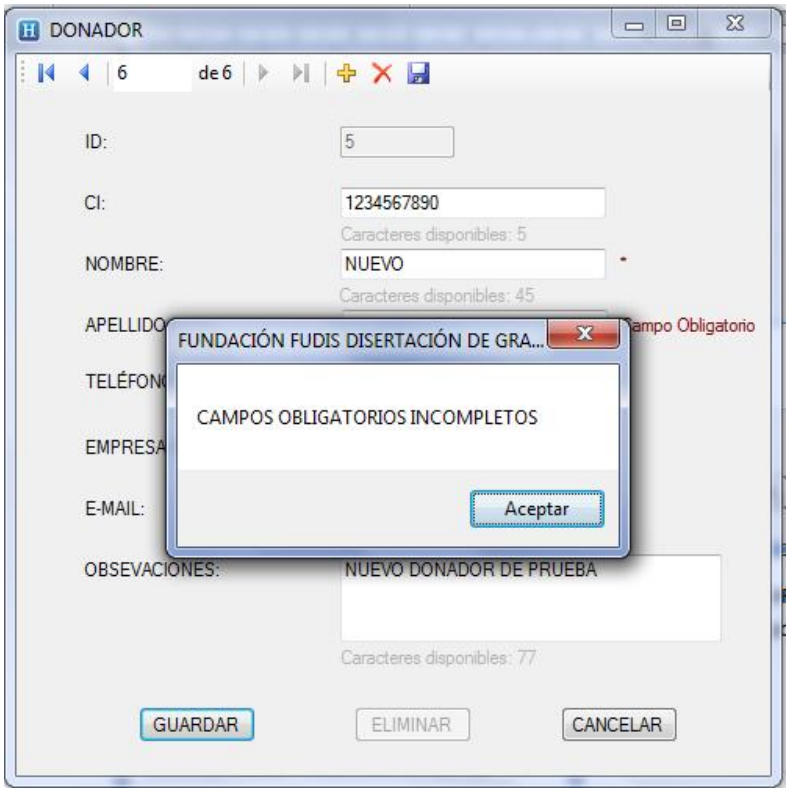
Aquí podemos buscar los donadores existentes y escoger alguno a nuestra conveniencia. Si no existe el donador que queremos podemos agregarlo solo debemos presionar en el botón CREAR, aquí nos abrirá un nuevo formulario llamado donador.



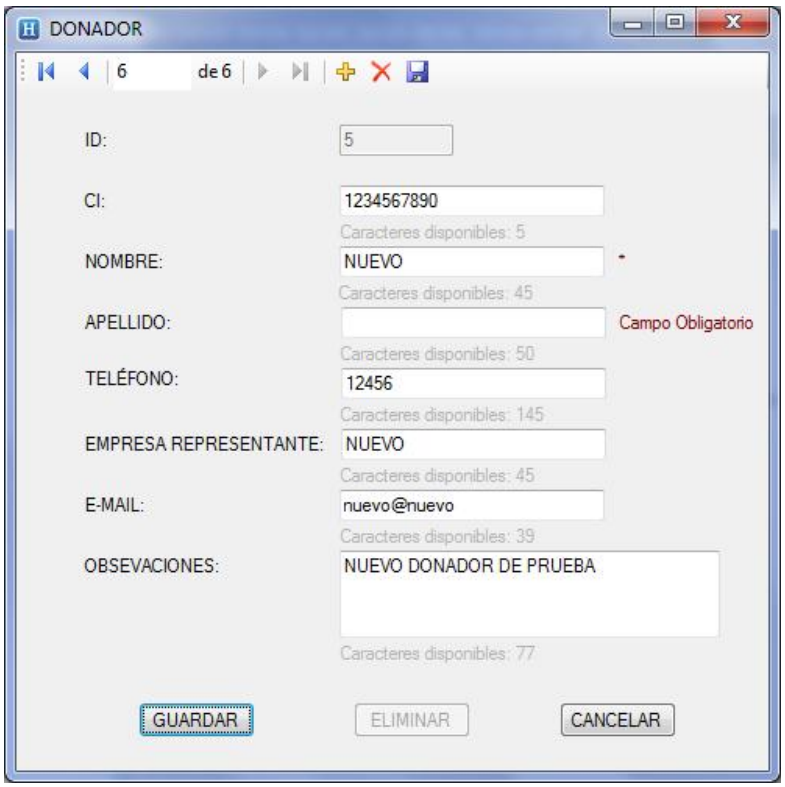
Aquí debemos introducir los datos necesarios u obligatorios para poder crear el nuevo donador.

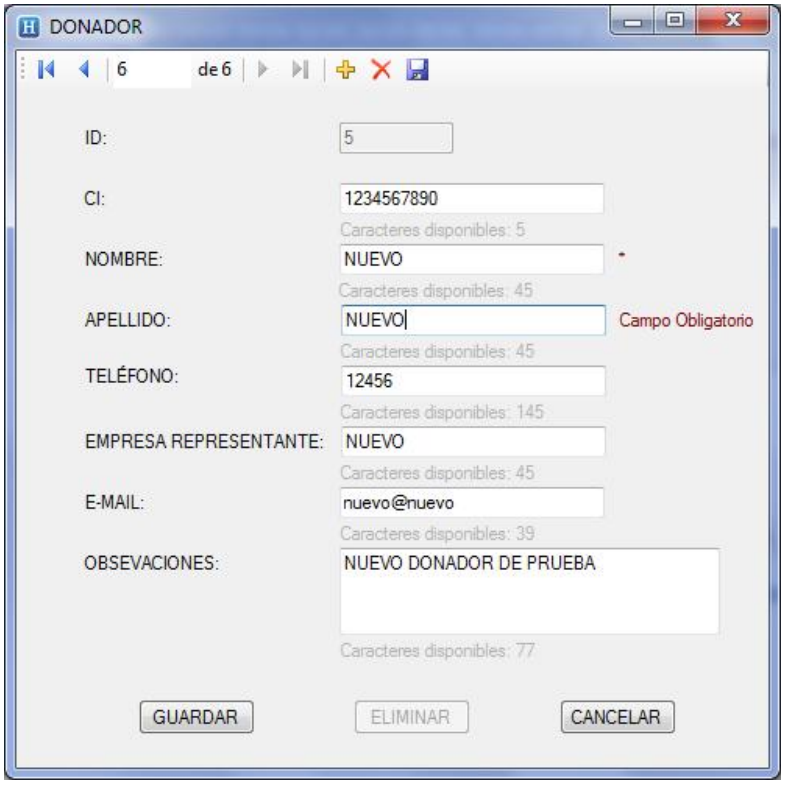


Para registrar este nuevo donador debemos presionar en el botón GUARDAR, si en alguno de los campos obligatorios no son llenados mostrará los mensajes correspondientes.

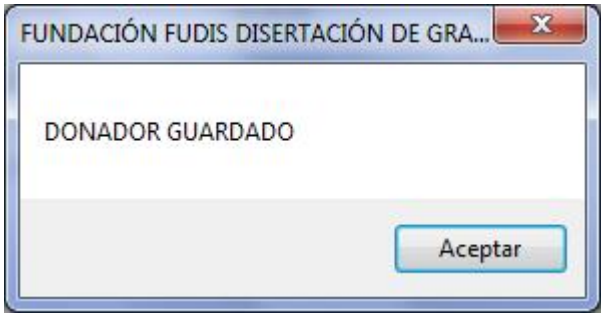


Es este caso no llenamos el campo obligatorio apellido y nos muestra un mensaje para ayudarnos.

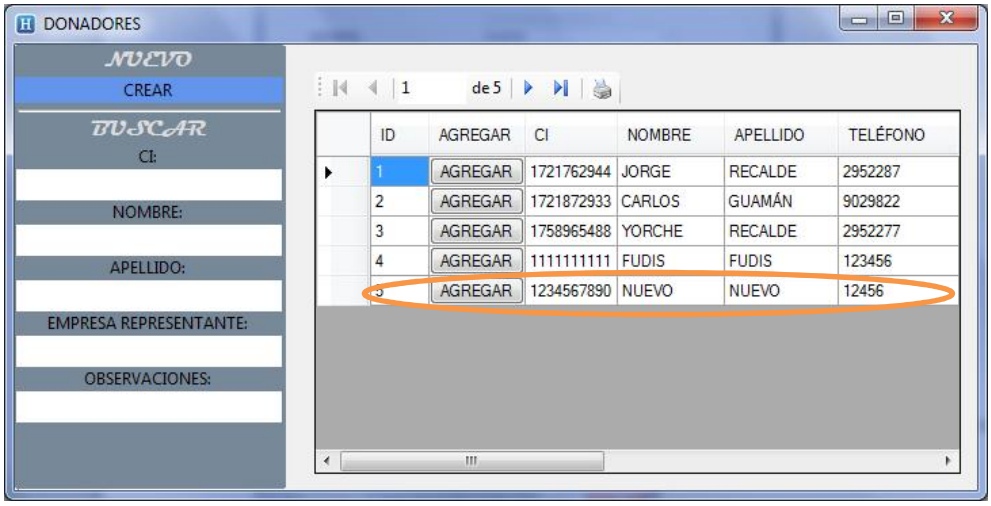




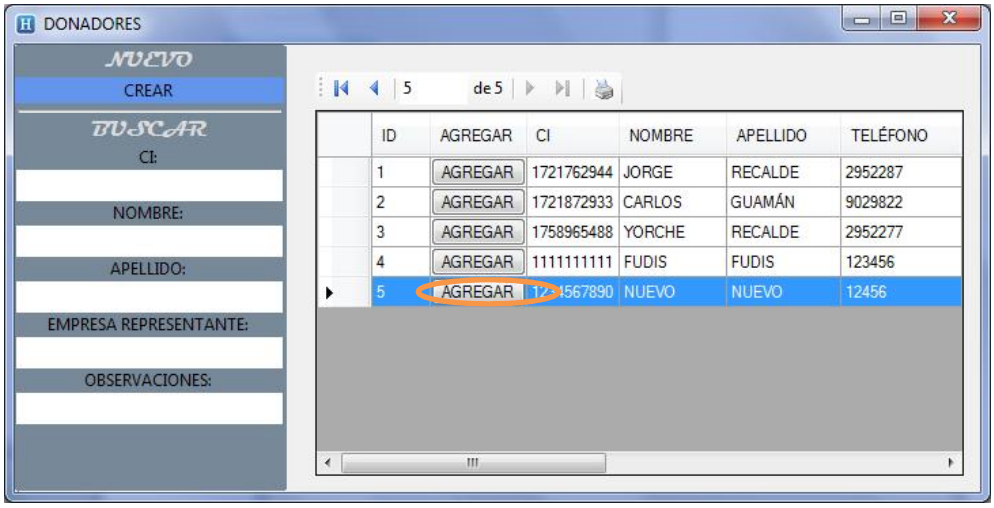
Una vez llenado todos los campos procedemos a presionar GUARDAR, donde nos mostrará un mensaje de aviso.

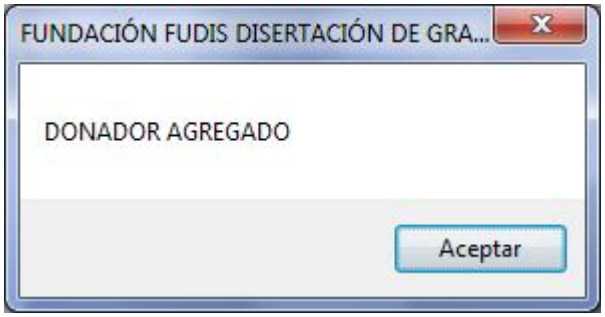


Ya podemos ver el nuevo donador registrado en la tabla de donadores

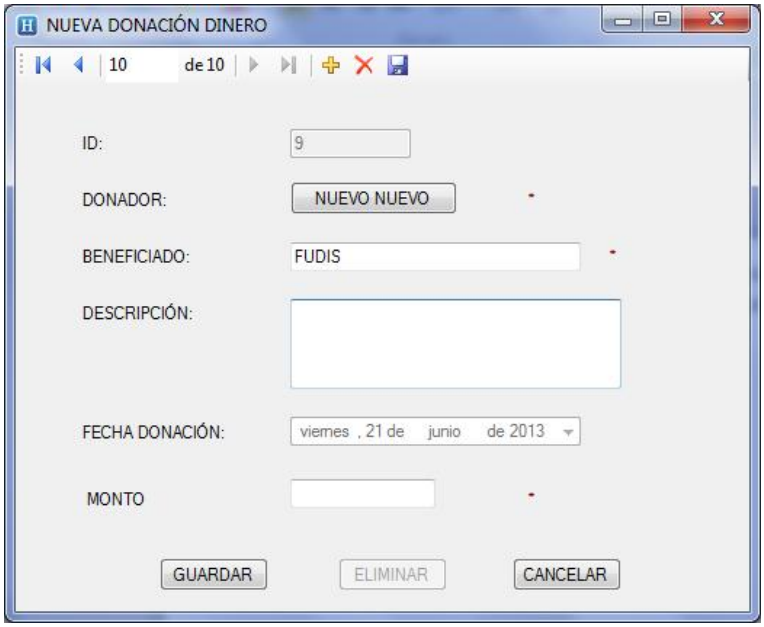


Continuamos en el ejemplo de donación de DINERO, este caso procedemos a escoger a "NUEVO", y presionamos el botón AGREGAR

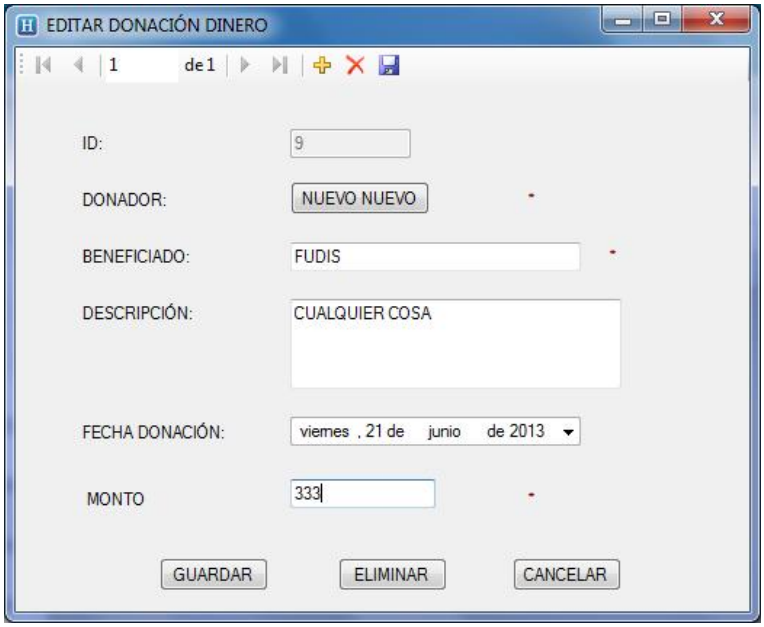




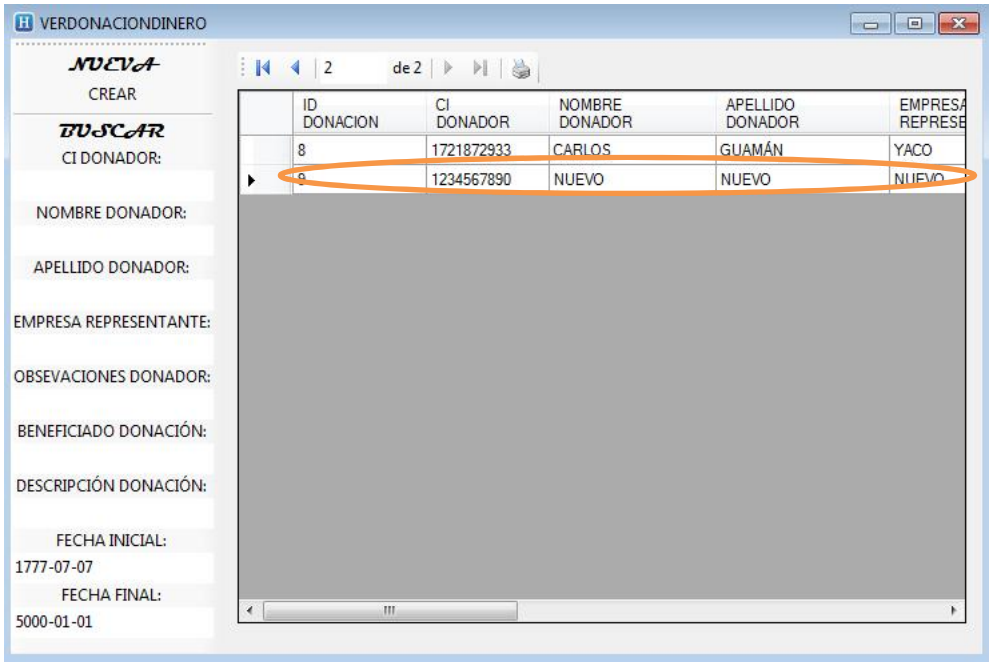
Una vez elegido el donador regresamos al formulario nueva donación de dinero donde podemos ver el nombre y apellido del donador, además podemos ver que en BENEFICIADO aparece el nombre de FUDIS.



Si queremos podemos introducir alguna descripción sin que sea necesaria, y se asignará la fecha actual en la que se realiza la donación y ponemos la cantidad que va a ser donada en este caso “333”.

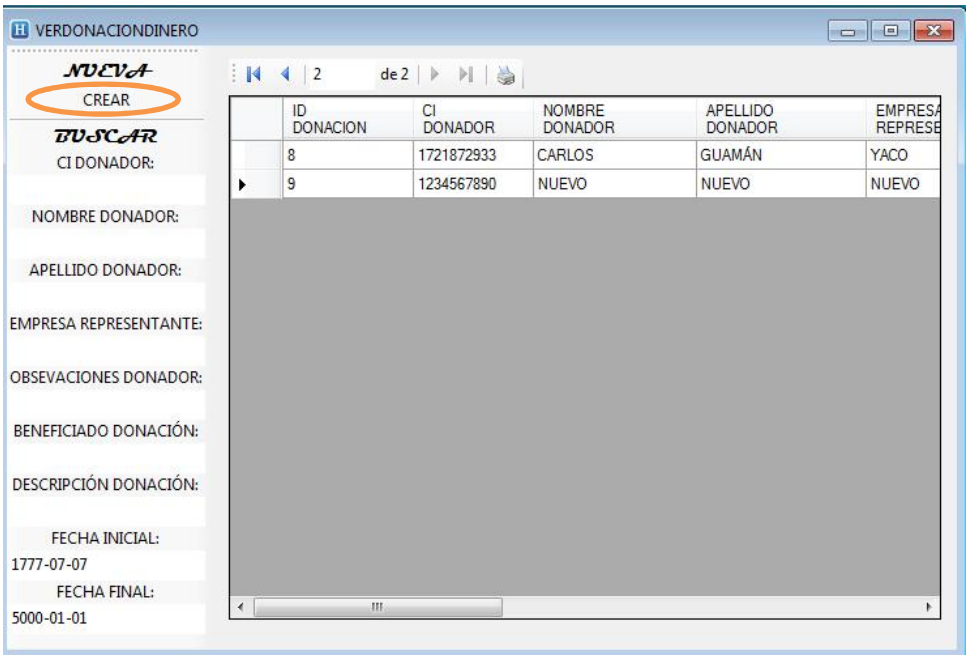


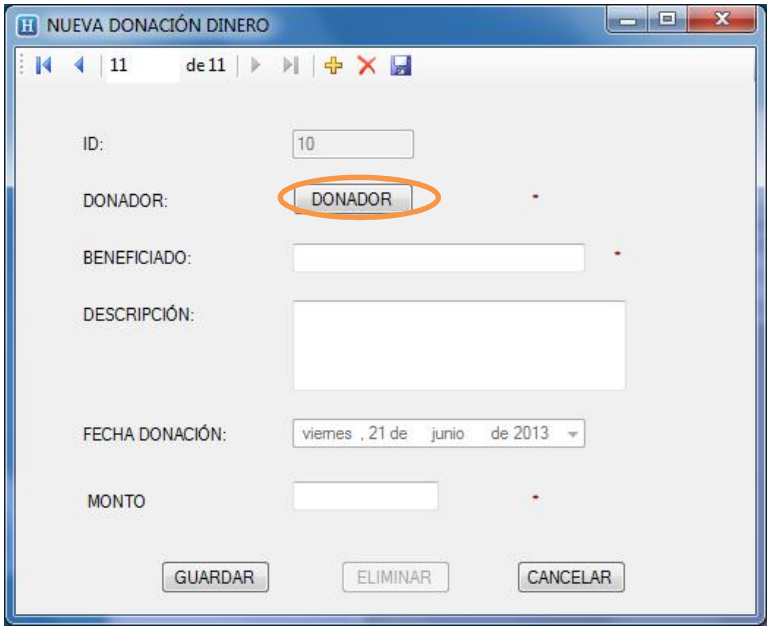
Una vez realizada este registro de donación regresamos al formulario de donaciones donde podemos ver la donación actualizada.



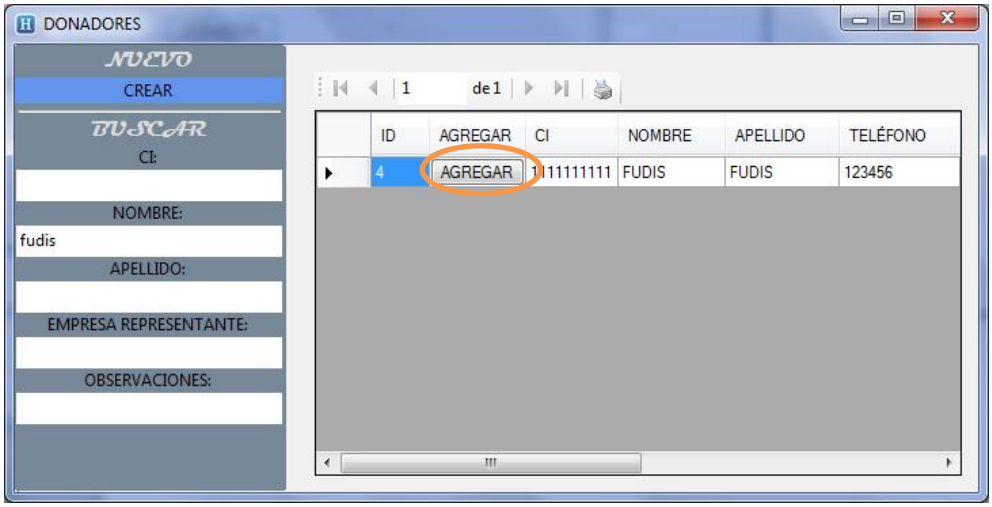
Crear Donación de dinero a otras instituciones

Para crear o registrar una nueva donación de dinero debemos presionar en el botón crear donde nos abrirá un nuevo formulario.

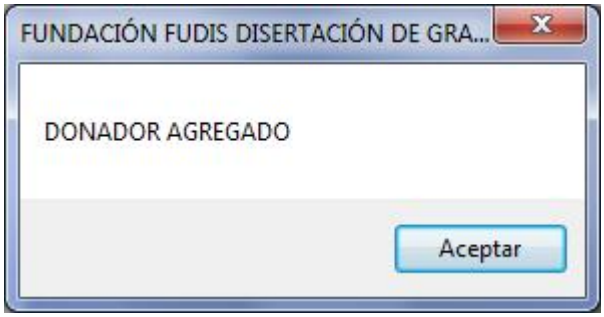




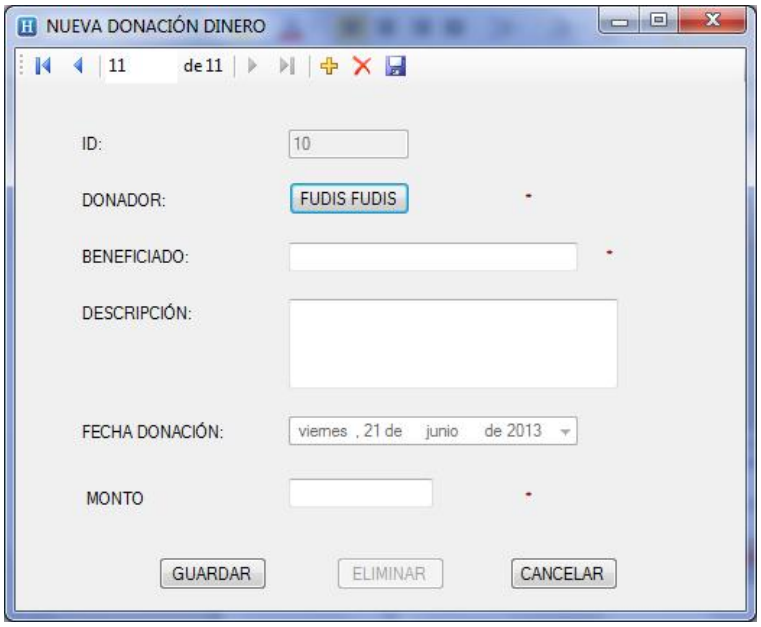
En el formulario de nuevo registro podemos ingresar la donación a quién nosotros queramos en este caso será de FUDIS a otras fundaciones que se desea donar. Para esto buscamos a FUDIS y procedemos a agregar el donador.



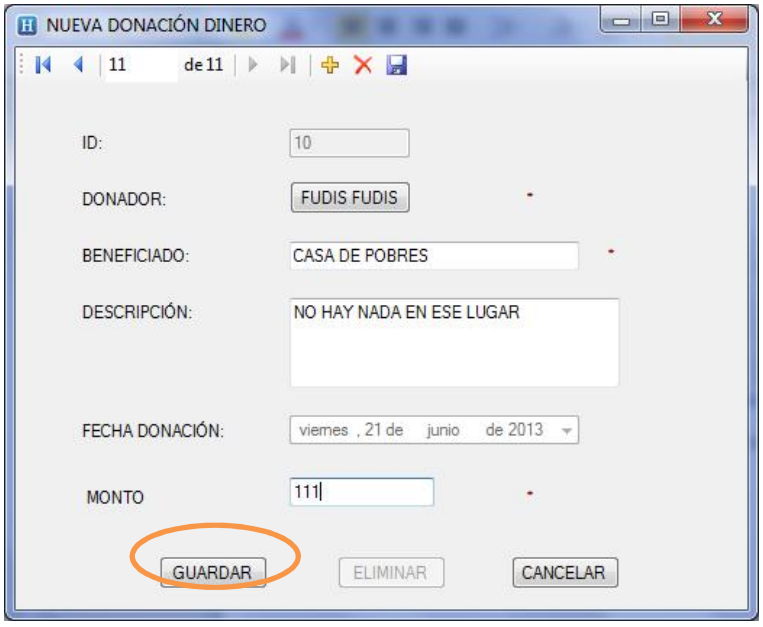
Una vez escogido el donadador FUDIS nos muestra mensajes de aviso.



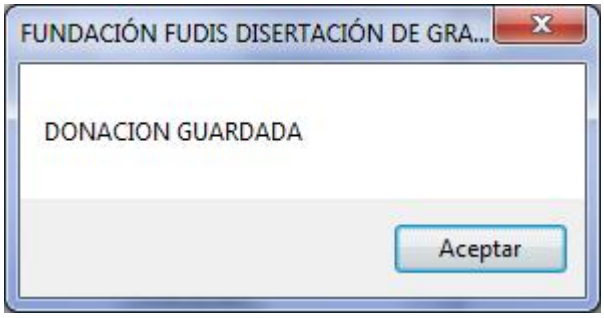
En este formulario podemos ver que el nombre de FUDIS ya aparece en el formulario de donación.



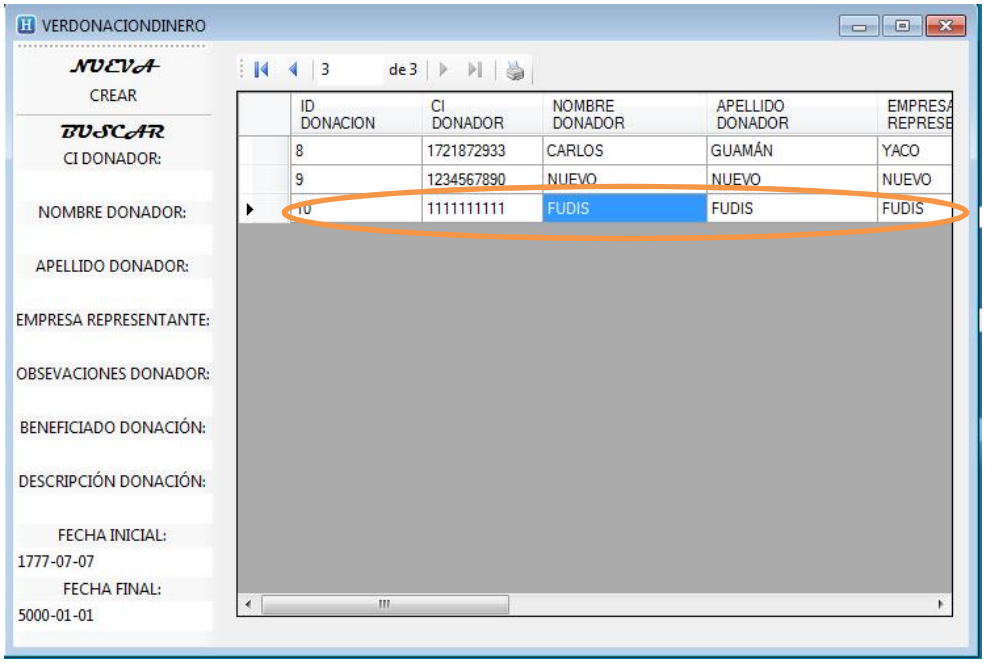
Procedemos a llenar los campos necesarios y presionamos en botón GUARDAR



Esto registrará la nueva donación de dinero de FUDIS hacia otras instituciones. Realizado esto mostrará mensajes de aviso.

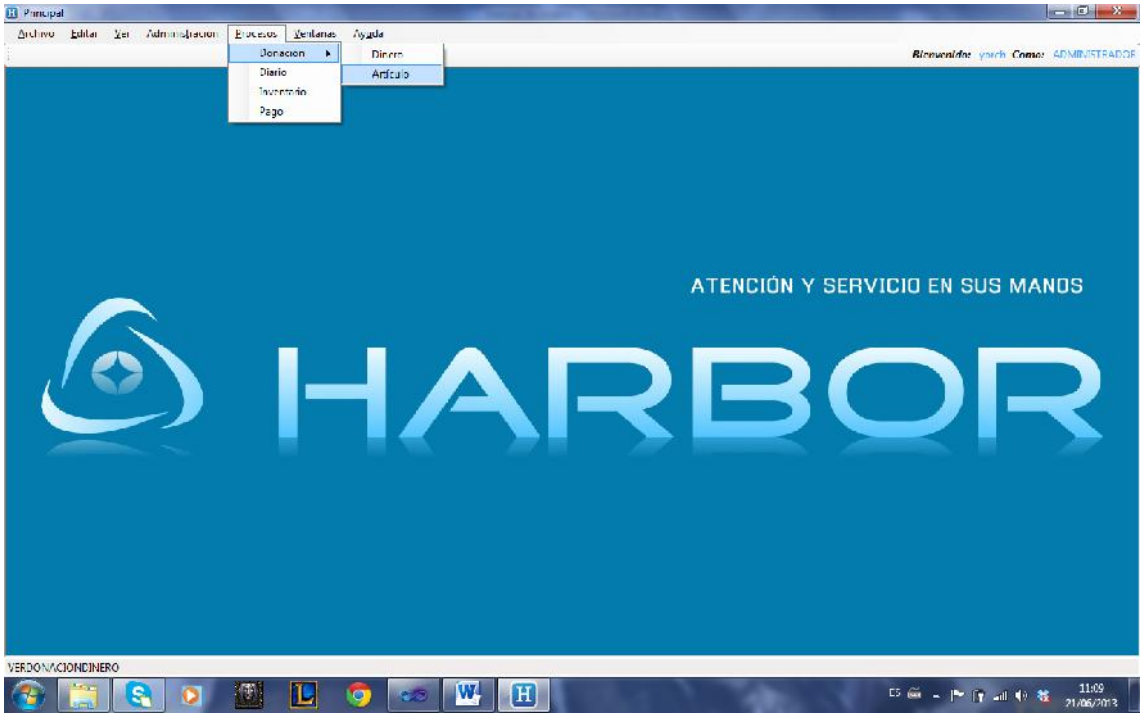


Con lo antes mencionado ya podemos ver la donación de dinero realizada en el formulario general de donaciones.

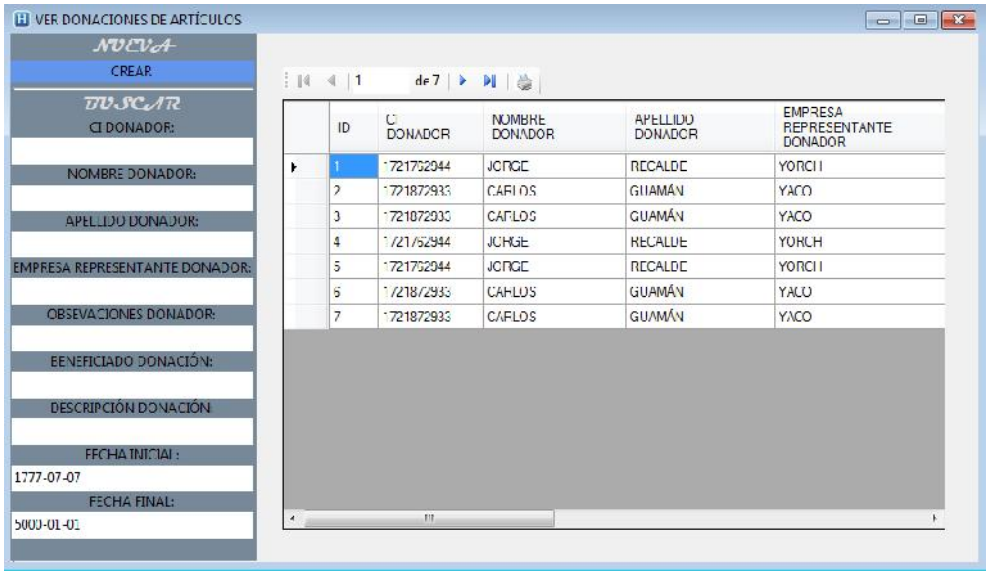


Crear Donación de Artículos a FUDIS

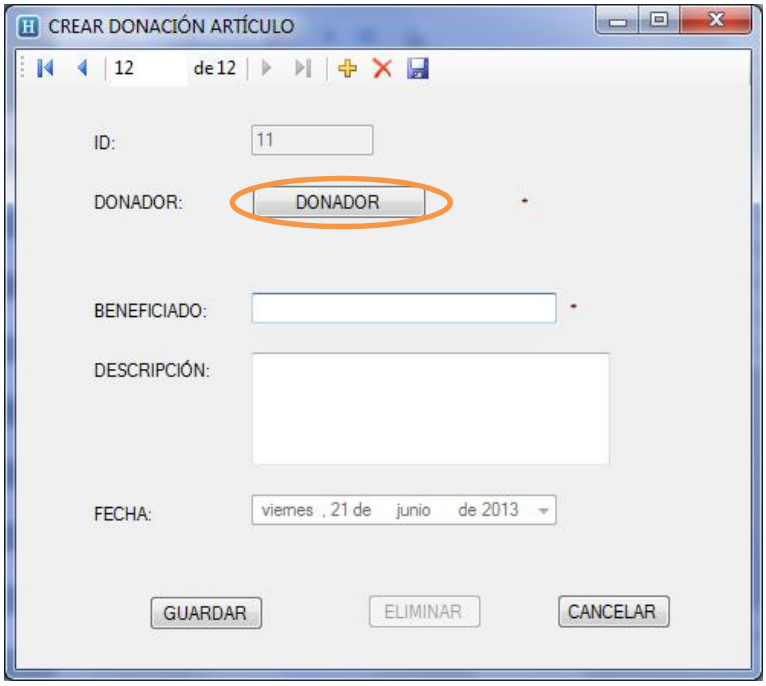
Debemos ir a Procesos, Donación, Artículo y abriremos el formulario general de donaciones de articulos.



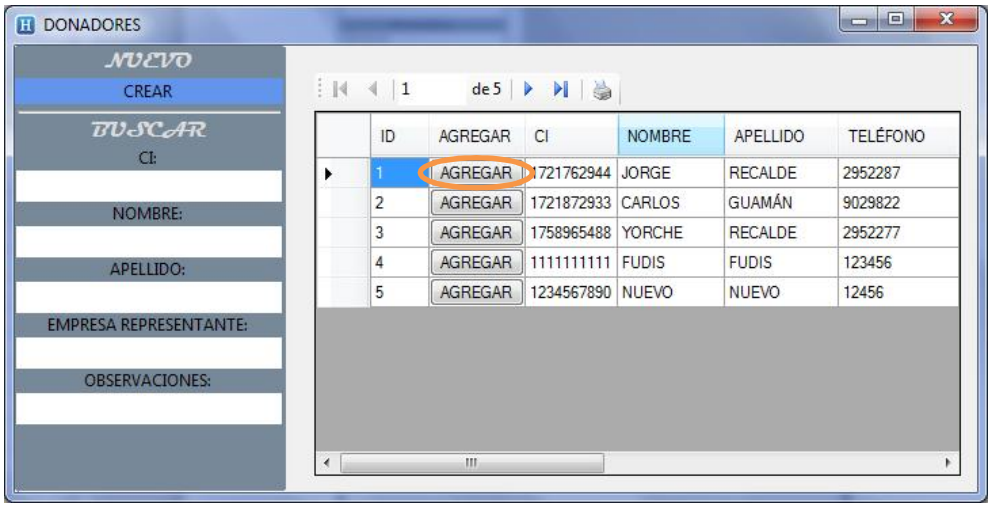
Para crear procedemos a presionar el botón CREAR el cual nos abrirá otro formulario.

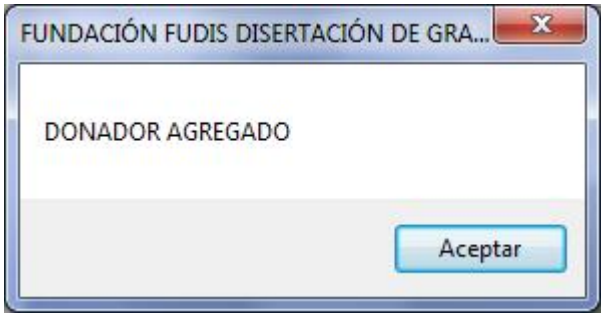


Aquí debemos escoger el donador. Para esto presionamos en el botón DONADOR el cual nos abrirá un formulario de búsqueda y selección de donadores.

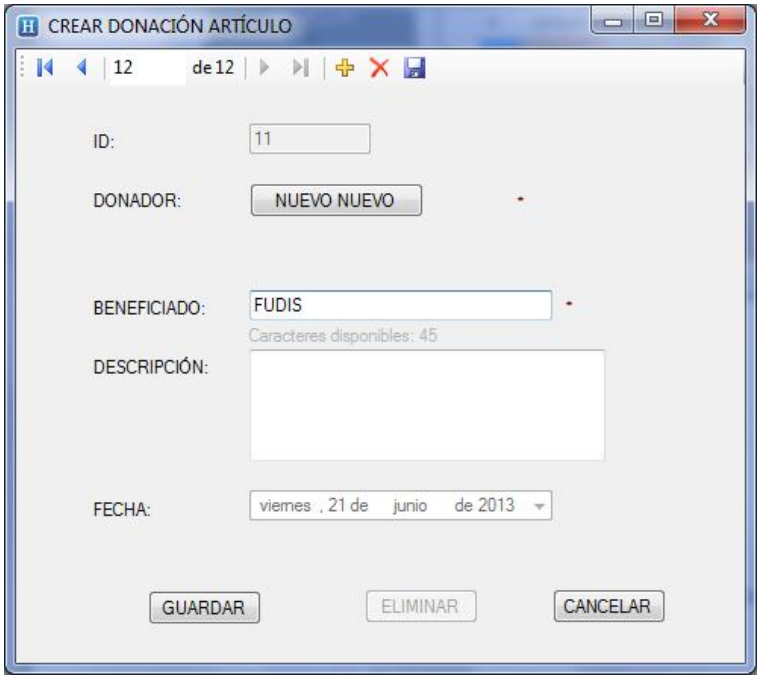


En este formulario podemos buscar y escoger a nuestro donador.



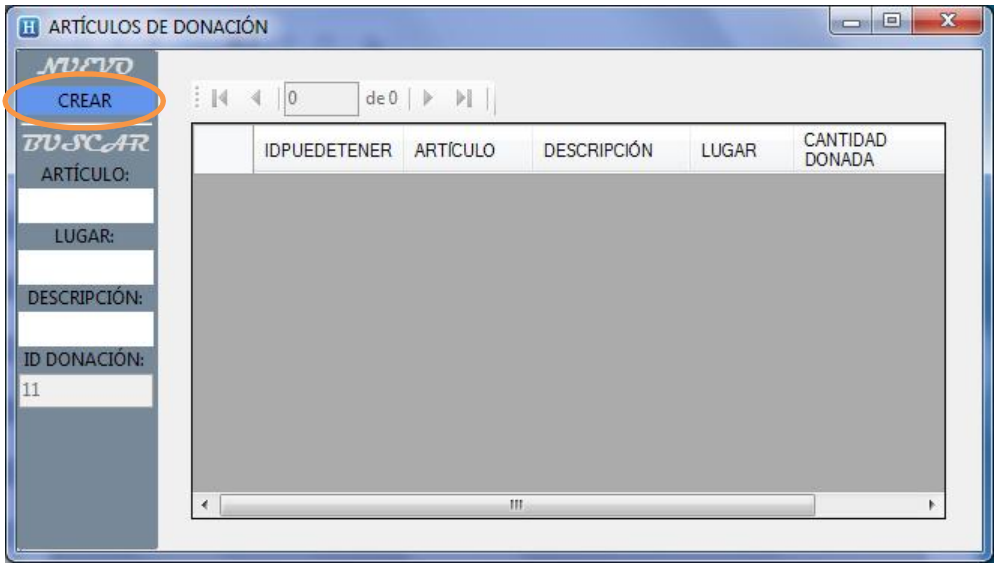


Volvemos al formulario de donación de artículo y observamos que ya tiene asignado el benefactor como "NUEVO" y el beneficiario como "FUDIS".

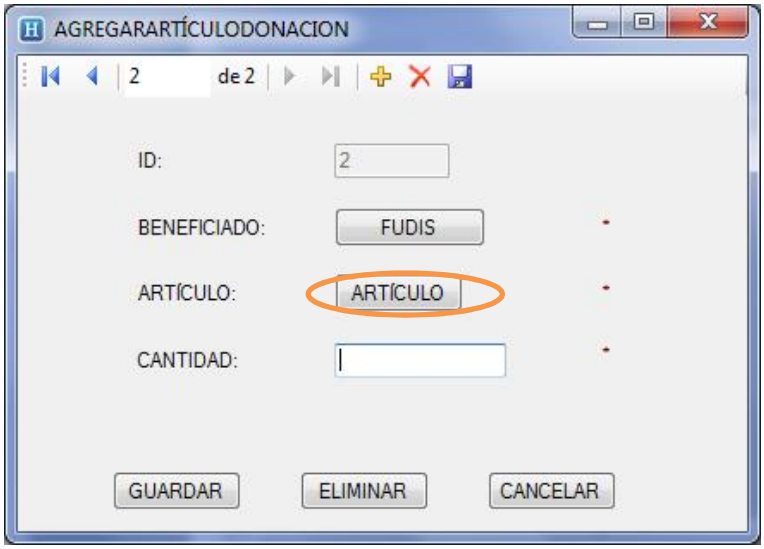


Llenamos los campos necesarios y obligatorios y presionamos GUARDAR y nos abrirá otro formulario donde podemos asignar el detalle del artículo y también podemos ver las donaciones que se han realizado con este donador.

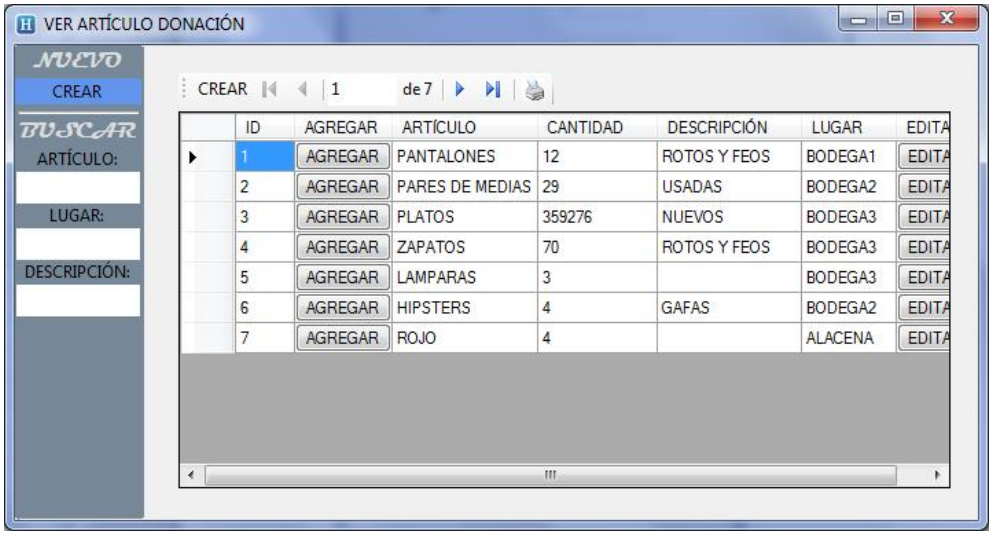
Para el caso no hay ninguna donación anterior así que procedemos a crear un detalle de artículos a donar, para esto presionamos el botón CREAR.



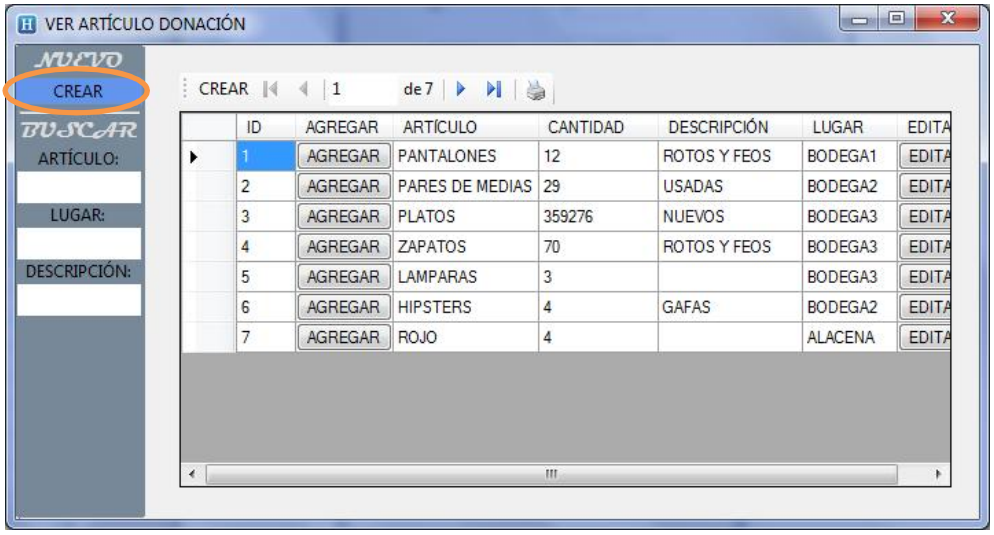
Este formulario nos permite ver quién es el beneficiado y seleccionar l artículo que vamos a donar, presionando el botón ARTÍCULO.



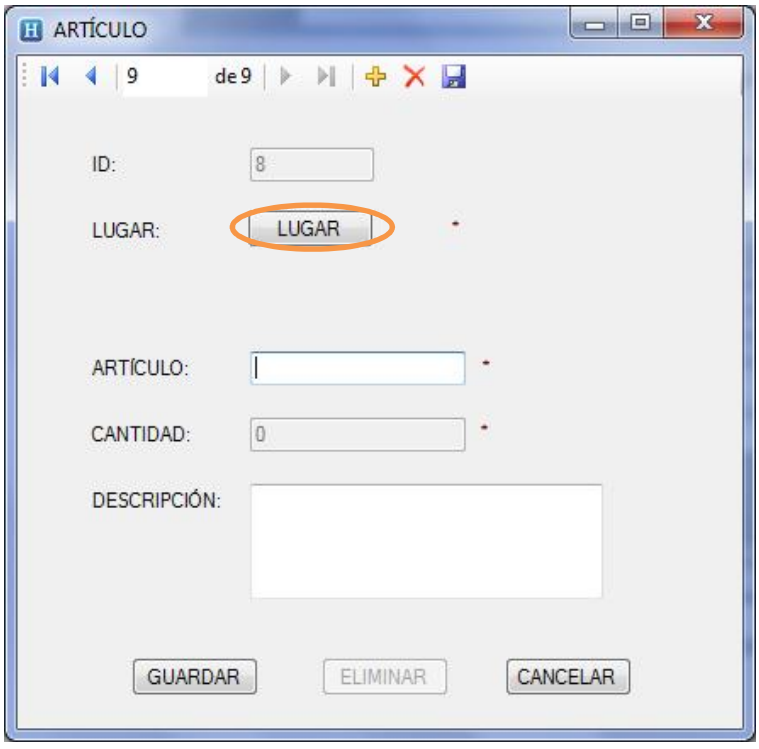
En este nuevo formulario podemos encontrar todos artículos que posee la fundación y escoger alguno de ellos para realizar la donación.



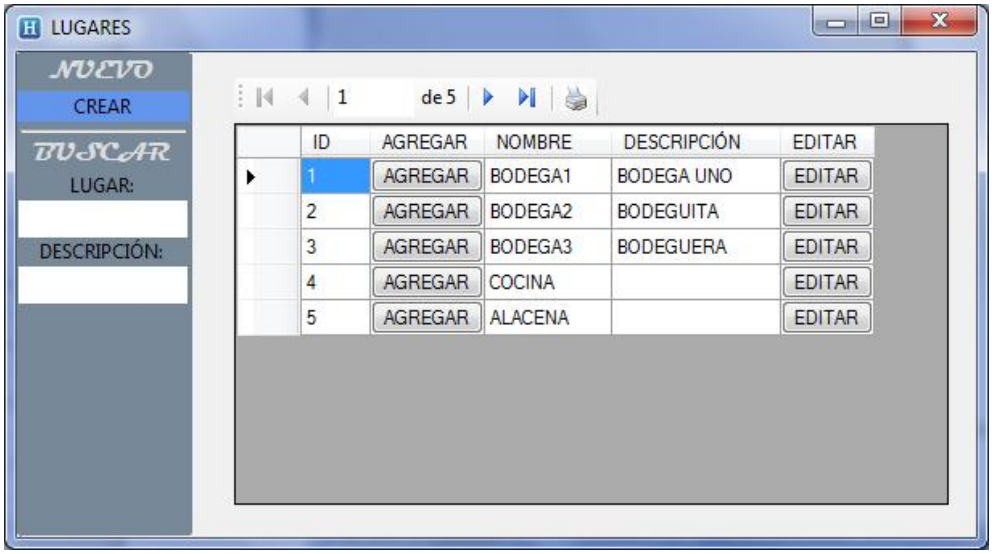
Si en el caso de que no hubiere el artículo que quiere donar procedemos a crear un nuevo artículo presionando el botón CREAR.



En este nuevo formulario podemos escoger el lugar en donde se va a almacenar este artículo. Para esto presionamos el botón LUGAR

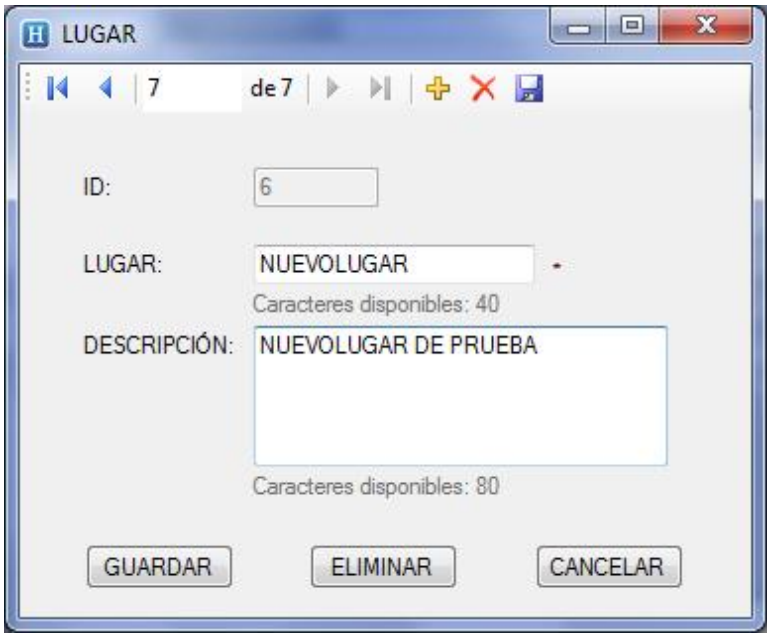


En este formulario podemos encontrar los lugares de almacenamiento de los artículos.

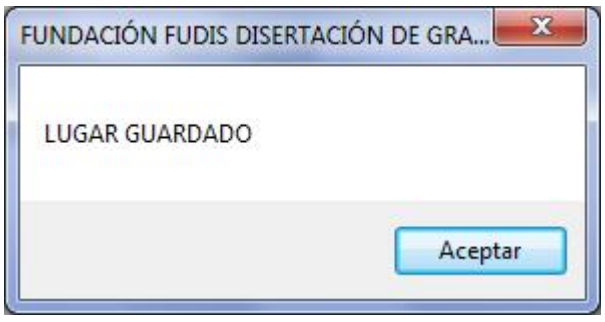


Si fuere el caso de que no hay el lugar de almacenamiento en la base de datos procedemos a crear un nuevo lugar de almacenamiento. Para esto presionamos el botón CREAR.

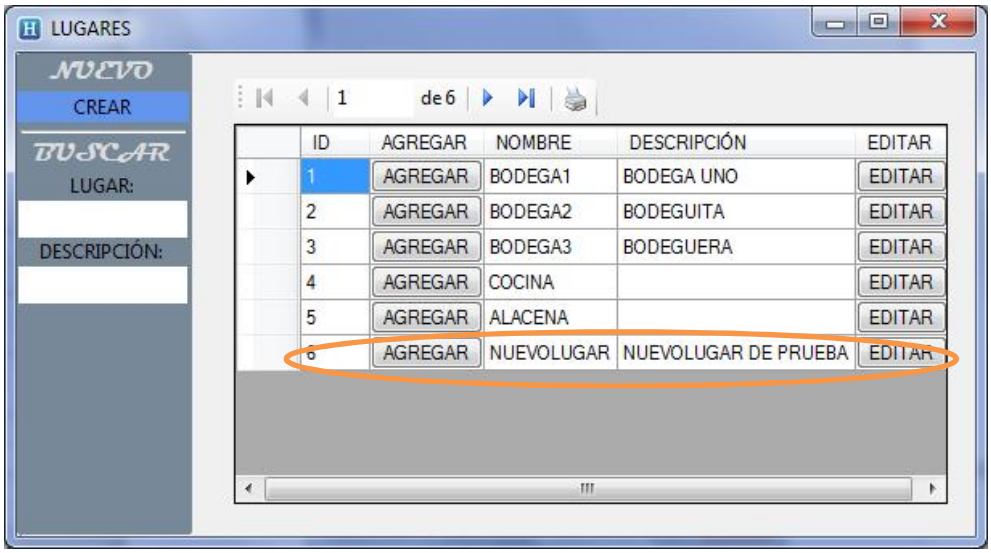
En este formulario procedemos a llenar los datos necesarios y presionamos GUARDAR



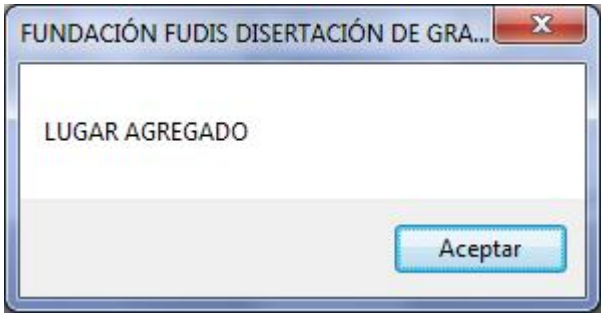
Nos muestra un mensaje de aviso que el nuevo lugar se guardó correctamente



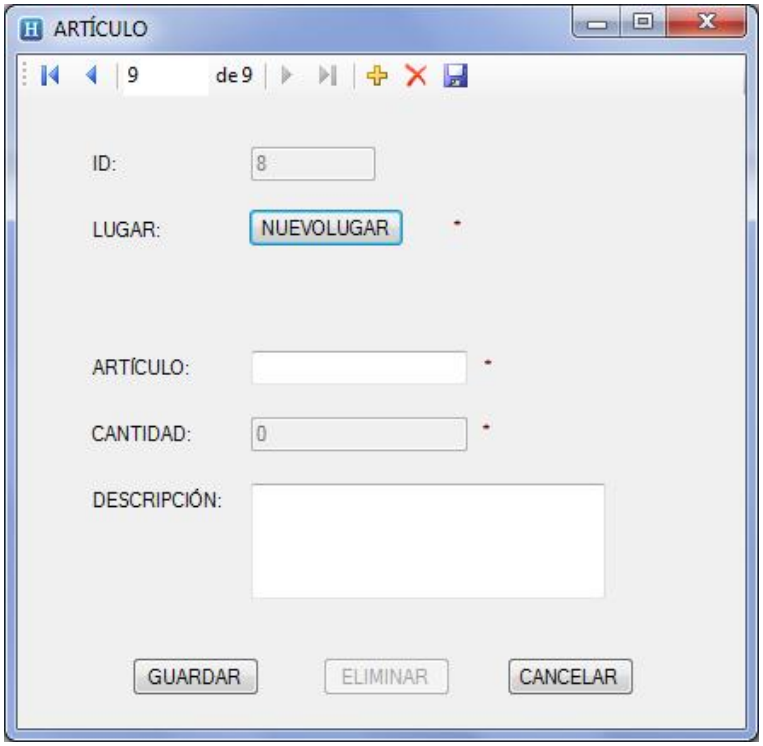
Regresamos al formulario de búsqueda de lugares de almacenamiento y podemos ver que el lugar ya fue guardado



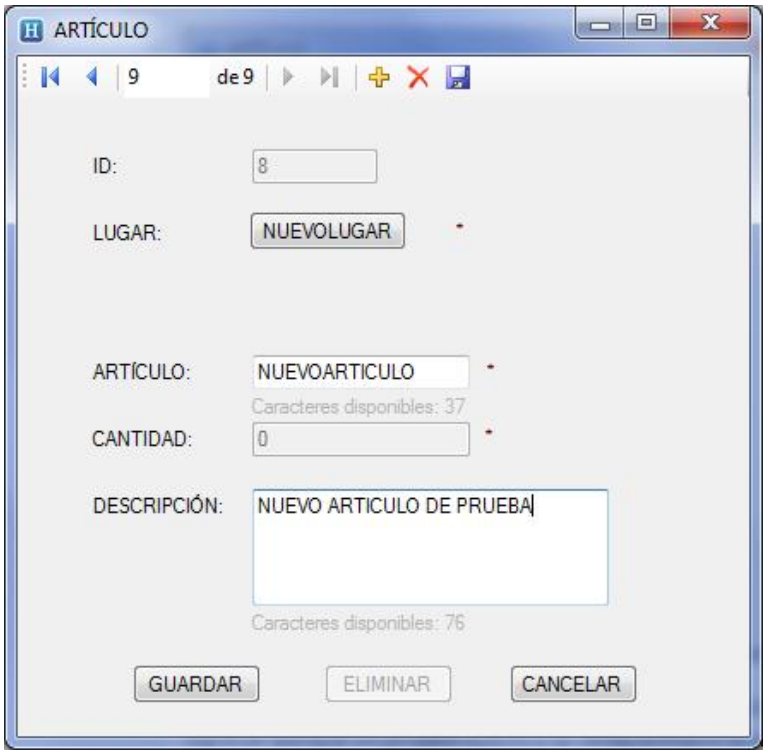
Presionamos AGREGAR y volvemos al formulario de artículos.



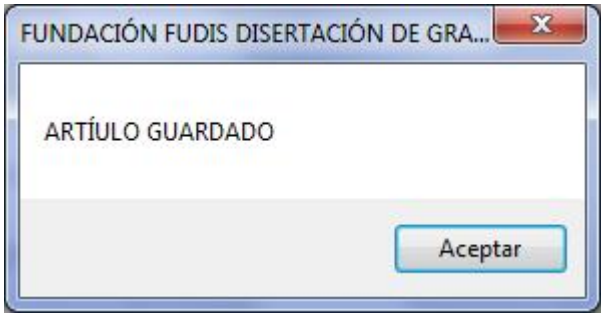
Observamos que ya tenemos los datos del lugar de almacenamiento dentro del formulario de artículo.



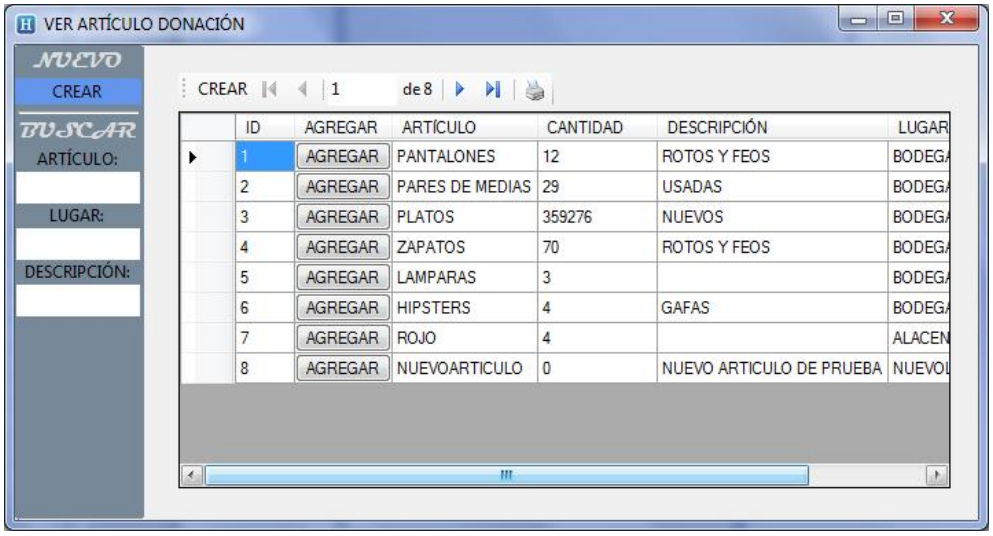
Aquí ponemos los datos correspondientes al nuevo artículo. En la cantidad no se podrá ingresar porque es un dato nuevo y se debe primero tener registrado el nombre del artículo nuevo.



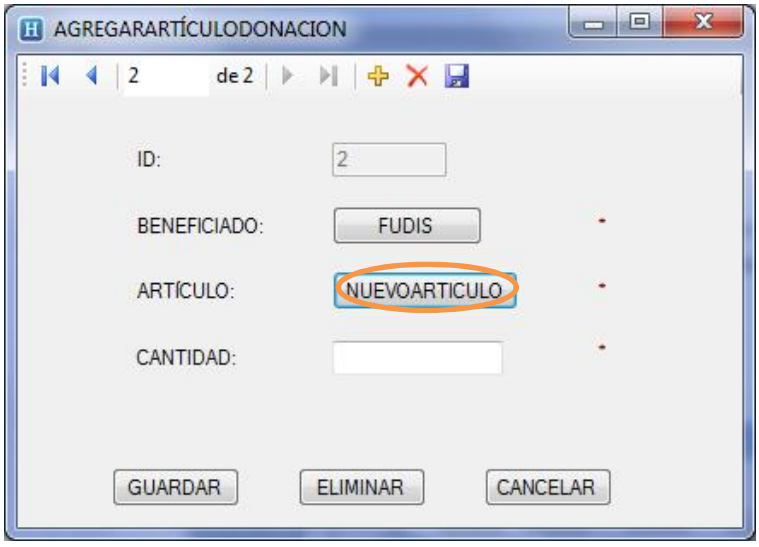
Presionamos guardar, nos mostrará que el nuevo artículo ya esta registrado.



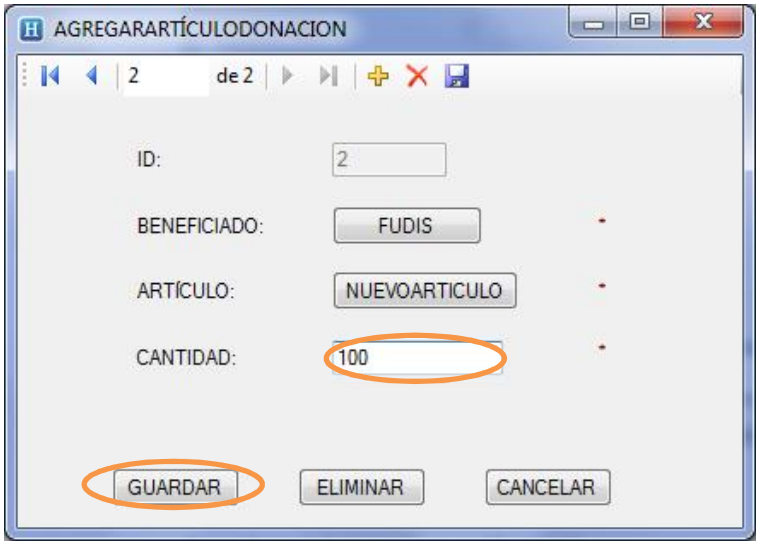
Hemos vuelto al formulario donde tenemos los artículos almacenados y vemos que ya esta el registro del nombre de nuestro nuevo artículo.

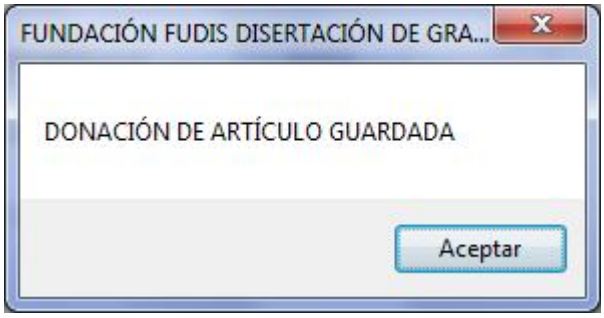


Presionamos GUARDAR y volvemos al formulario de donación de articulo y vemos que el nombre del articulo a registrar ya está con el nombre del nuevo artículo registrado.

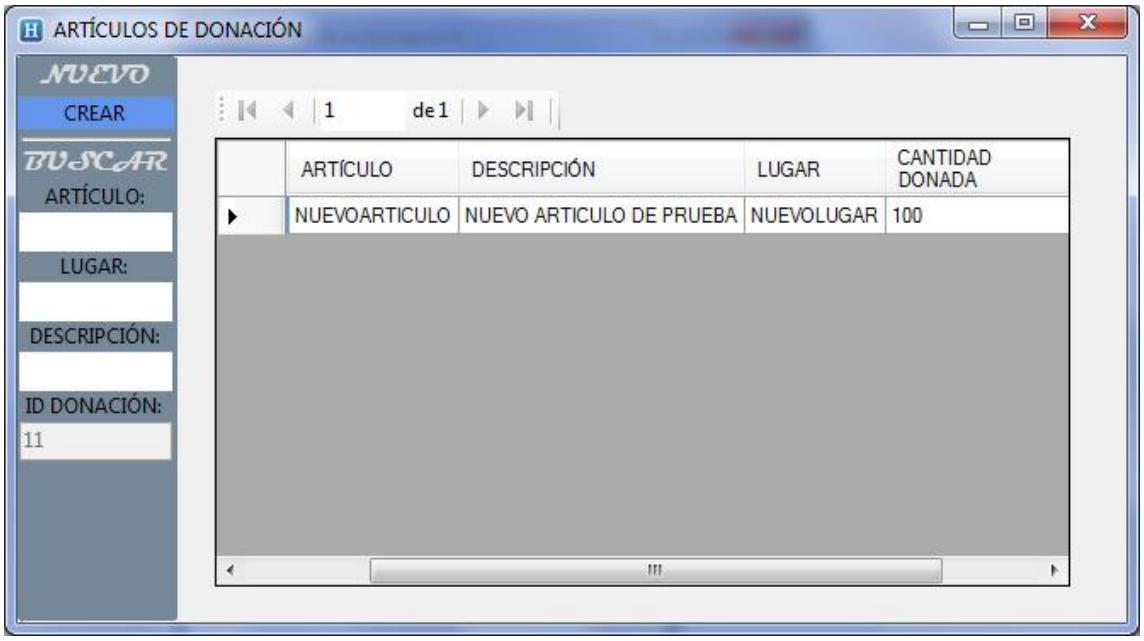


Aquí procedemos a ingresar la cantidad que se va a donar en este caso “100”. Presionamos guardar y nos registrará la donación y mostrará mensajes de aviso.



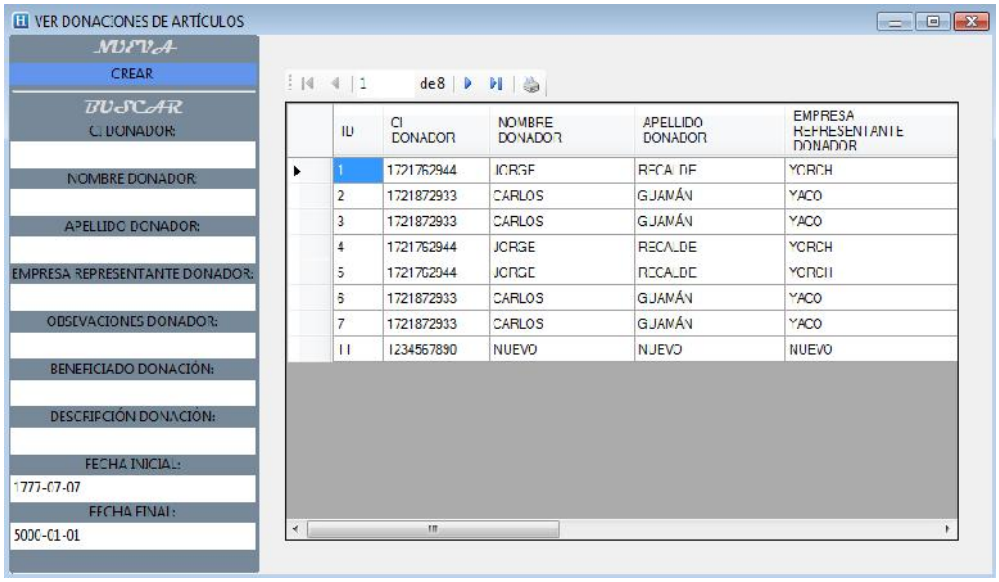


Regresamos al formulario de registro de donaciones de acuerdo al DONADOR y observamos que ya tiene el registro de donación de NUEVOARTICULO, en el lugar NUEVOLUGAR y que su cantidad es 100.

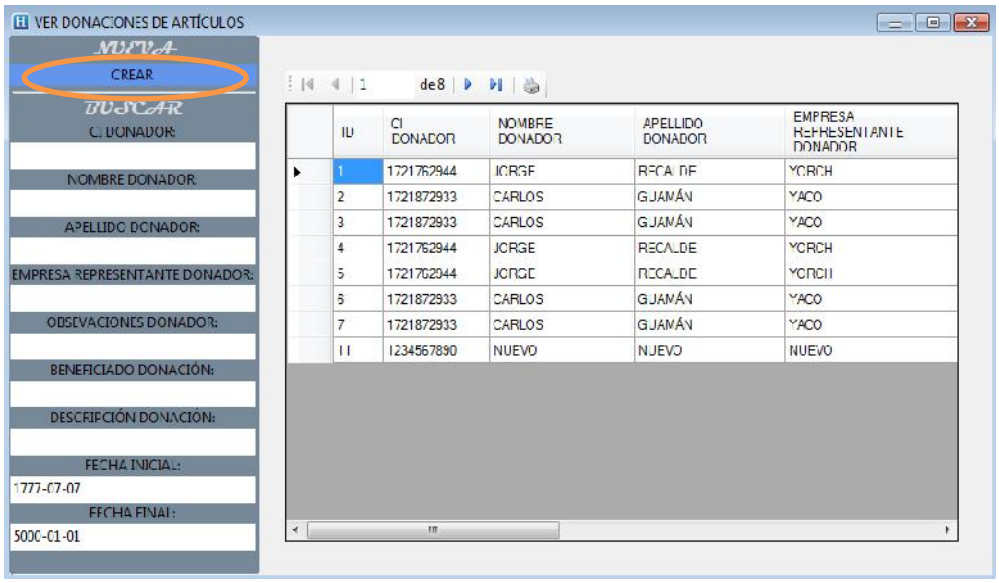


Crear Donación de Artículos a otras Instituciones

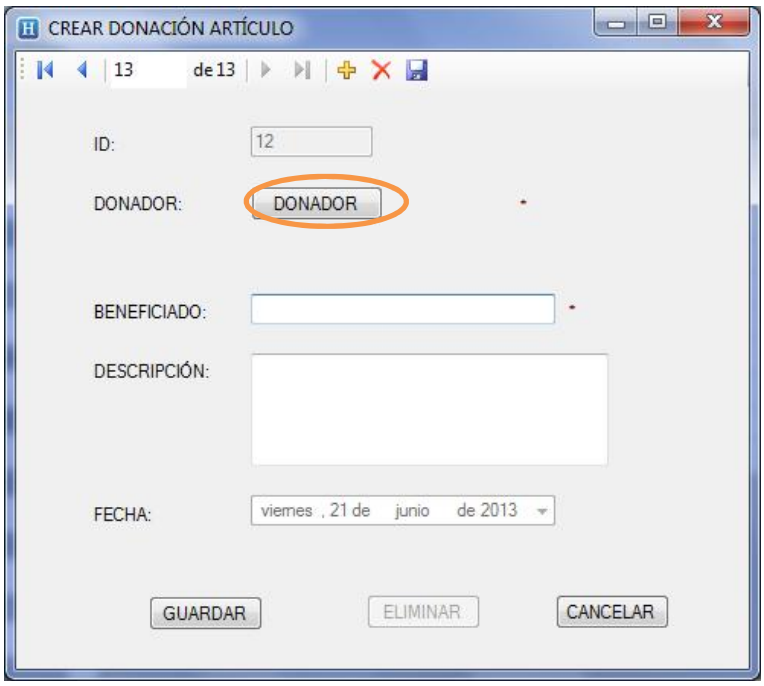
De la misma forma que hicimos donaciones de artículos para FUDIS. PROCESOS, DONACIONES, ARTICULOS.



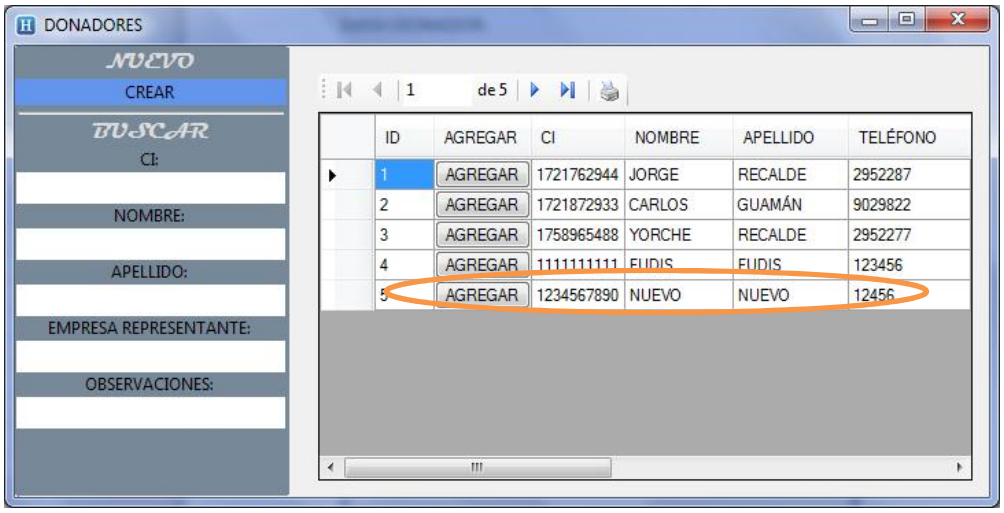
Procedemos a crear la nueva donación de parte de FUDIS hacia otras instiruciones para esto presionamos el botón CREAR.



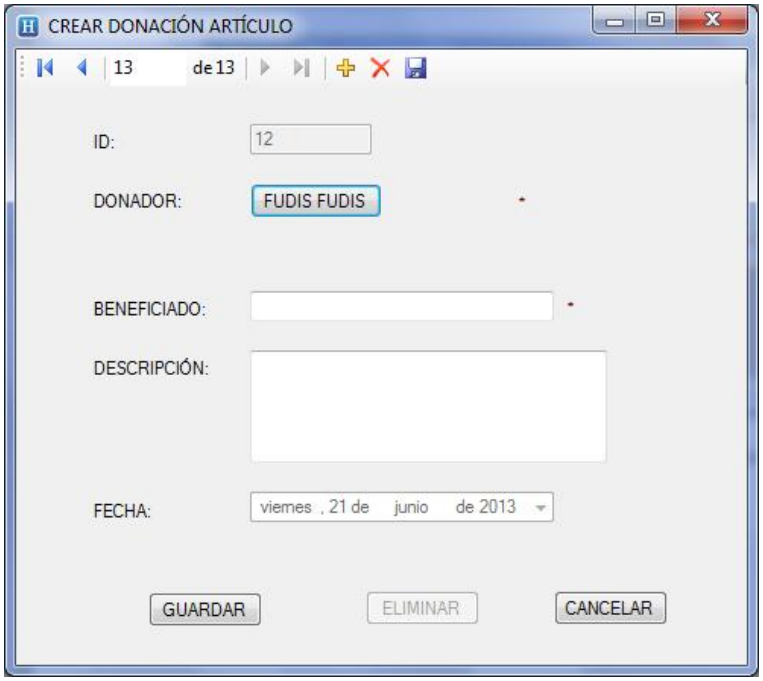
En este formulario escogemos el donador que este caso será FUDIS , para esto presionamos el botón DONADOR.



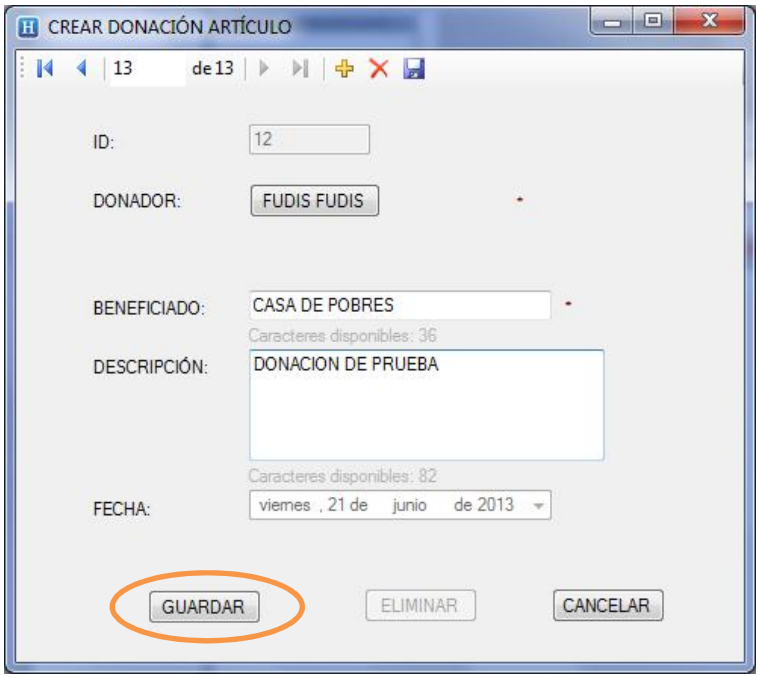
Buscamos y escogemos al donador FUDIS de la tabla.



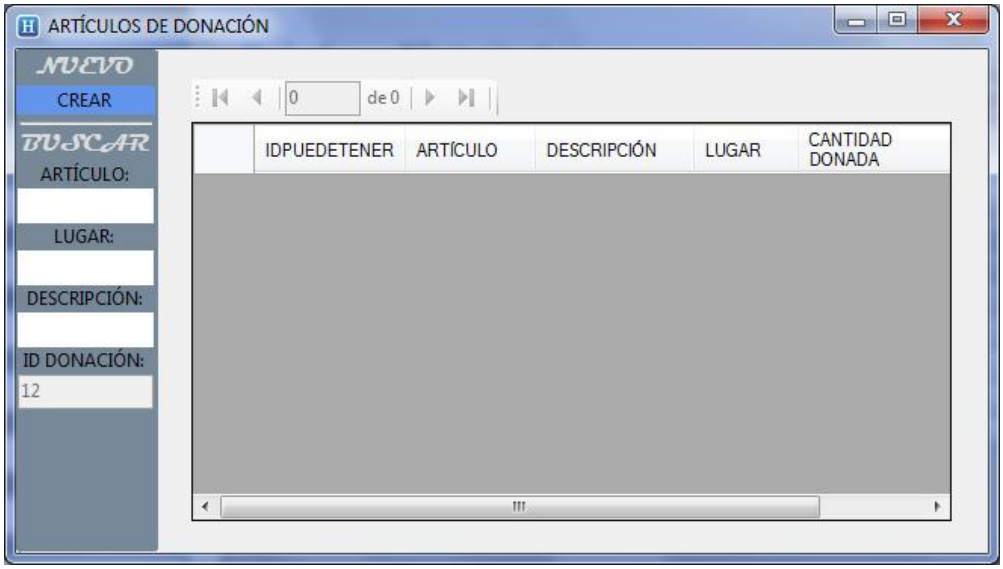
Regresamos al formulario de donación y vemos que FUDIS ya está agreado.



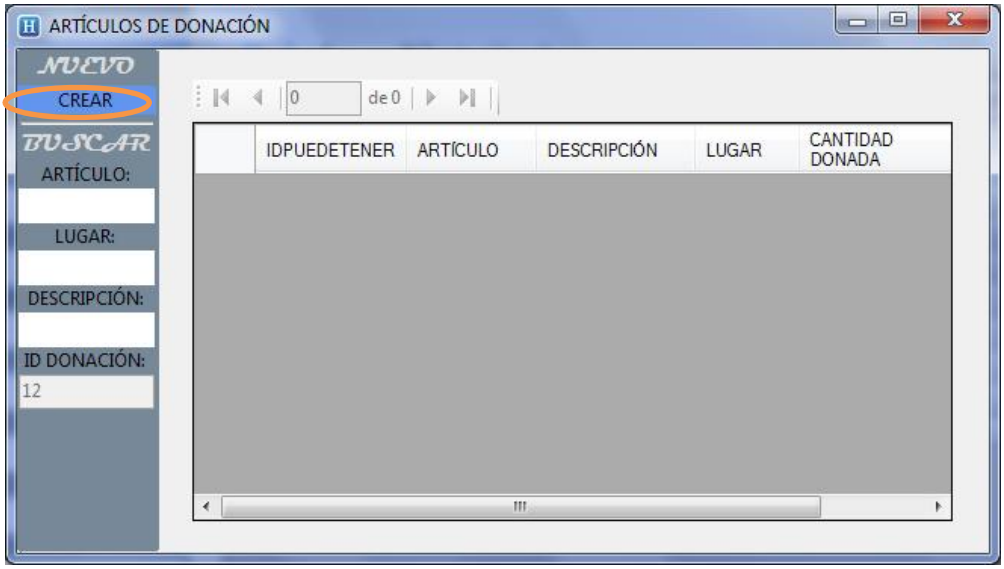
Llenamos los campos y presionamos GUARDAR



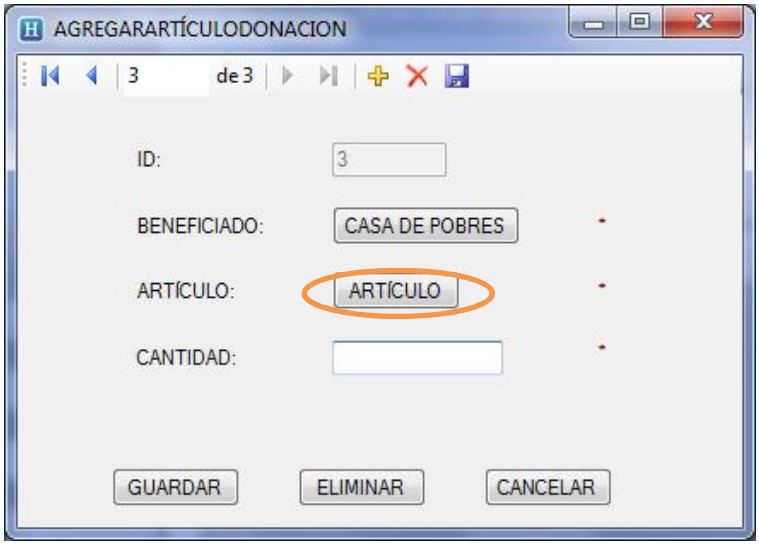
Como es la primera vez que FUDIS realiza una donación no tiene registros almacenados.



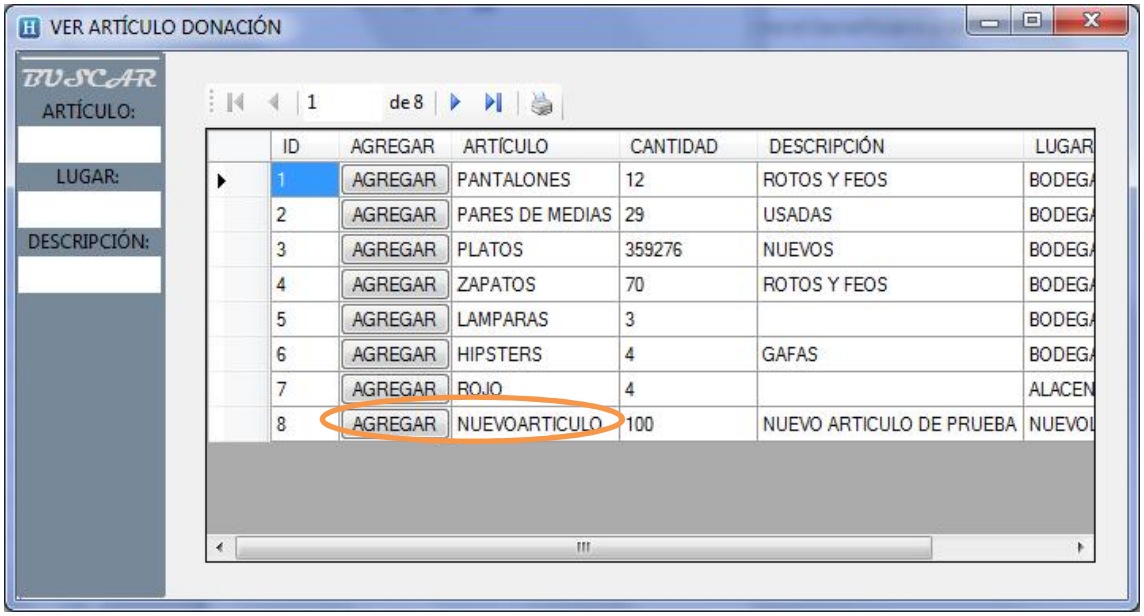
Procedemos a crear nuevos registros de artículos presionando el botón CREAR.



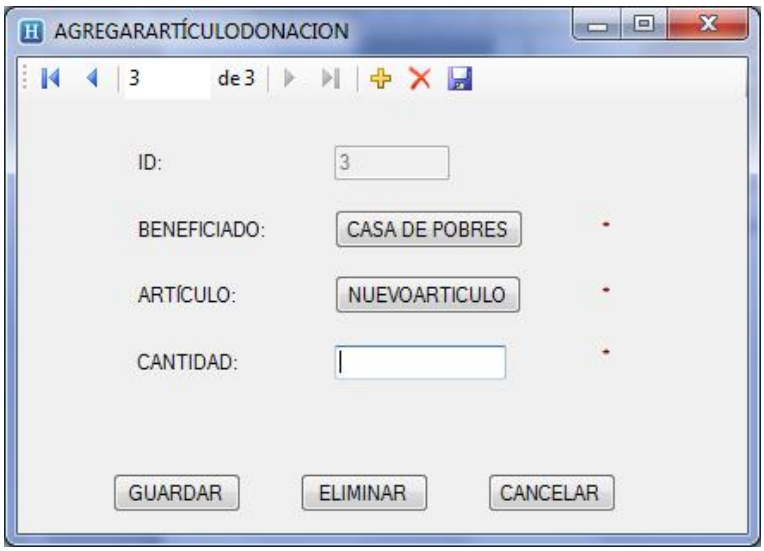
Podemos ver en este formulario que tiene el beneficiario y debemos seleccionar los artículos que vamos a donar. Para esto presionamos el botón ARTÍCULO.



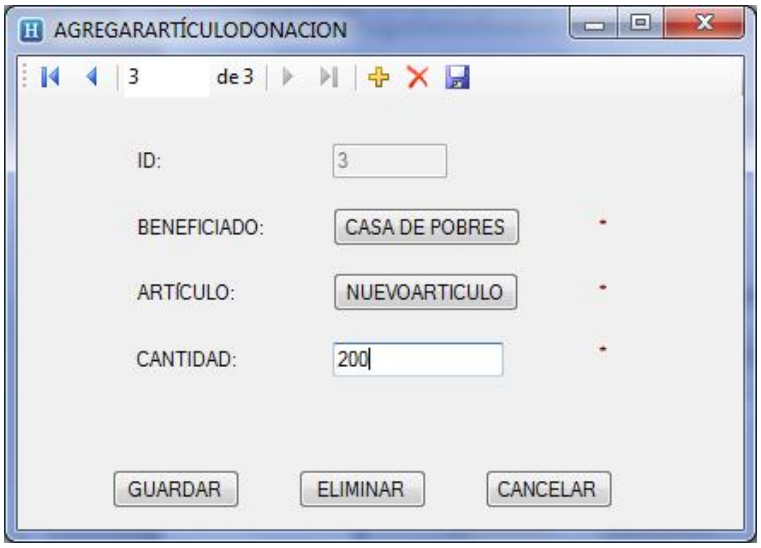
En este formulario nos muestra los artículos que tenemos en bodega. Pero el botón crear nuevo artículo no existe porque no se puede donar algo que no se tiene. Para el ejemplo seleccionamos el artículo "NUEVOARTICULO" y presionamos el botón AGREGAR.

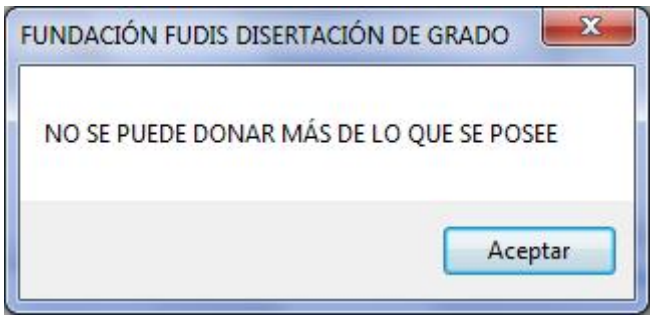


Regresamos al formulario de donación de articulo y vemos que el beneficiario se mantiene y que el articulo ha sido seleccionado.

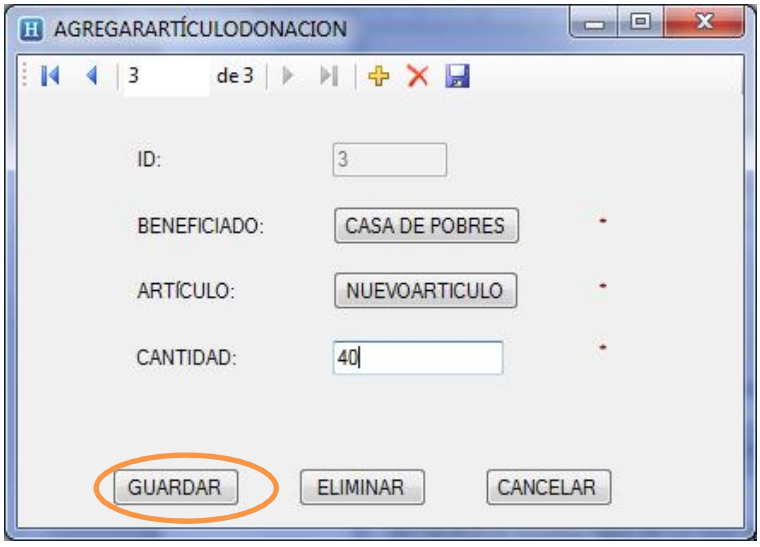


Procedemos a ingresar la cantidad que deseamos donar. Para el caso pondremos “200” artículos que vamos a donar y obviamente que no hay esa cantidad, nos presentará un mensaje que no se puede donar más de lo que se tiene, para el efecto presionamos el botón GUARDAR.

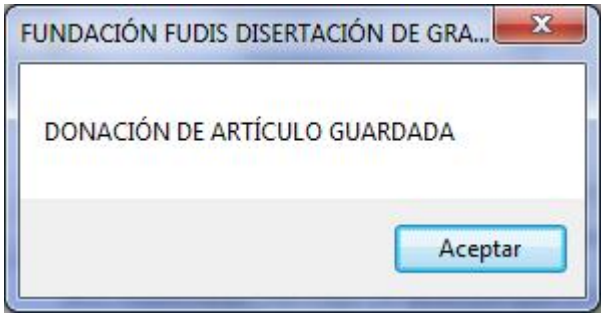




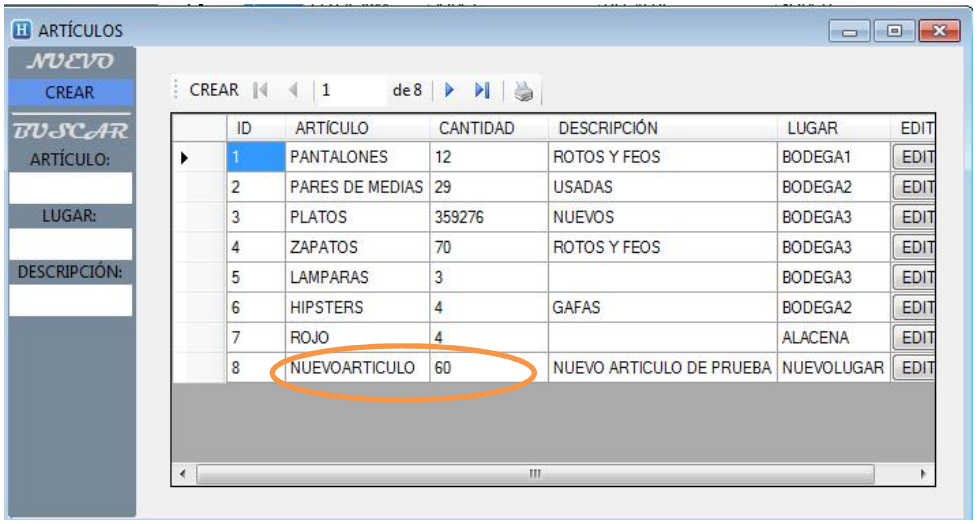
Con este antecedente cambiamos la cantidad y ponemos “40” y presionamos GUARDAR.



Nos muestra un mensaje de aviso porque el registro se realizó.

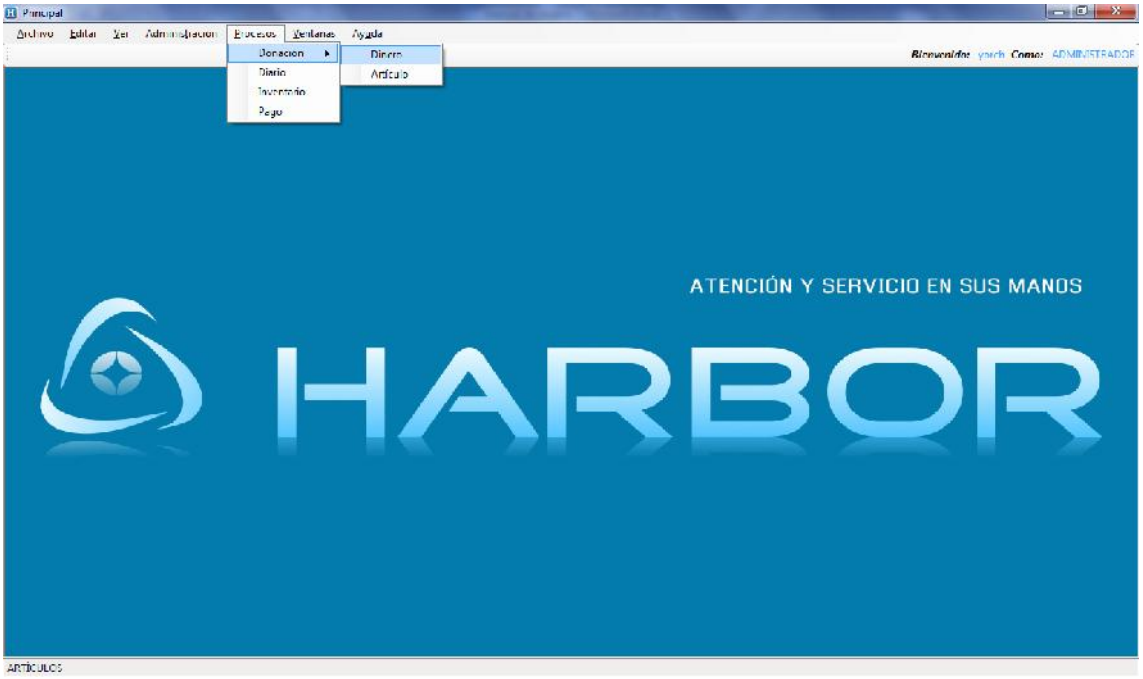


Vemos que el valor de "ARTICULONUEVO" ha disminuido.

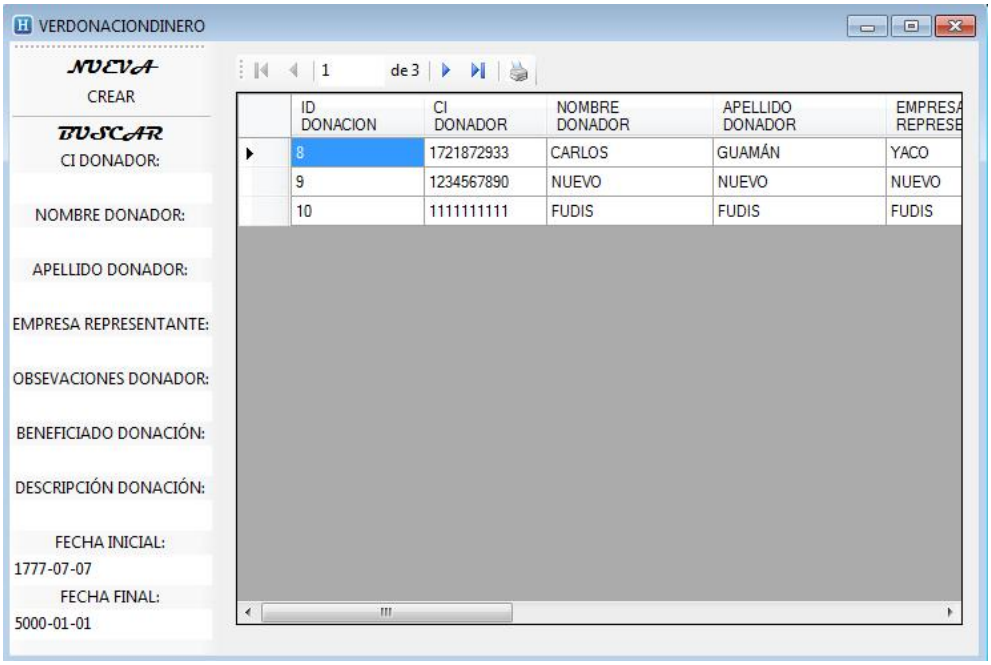


Editar donaciones

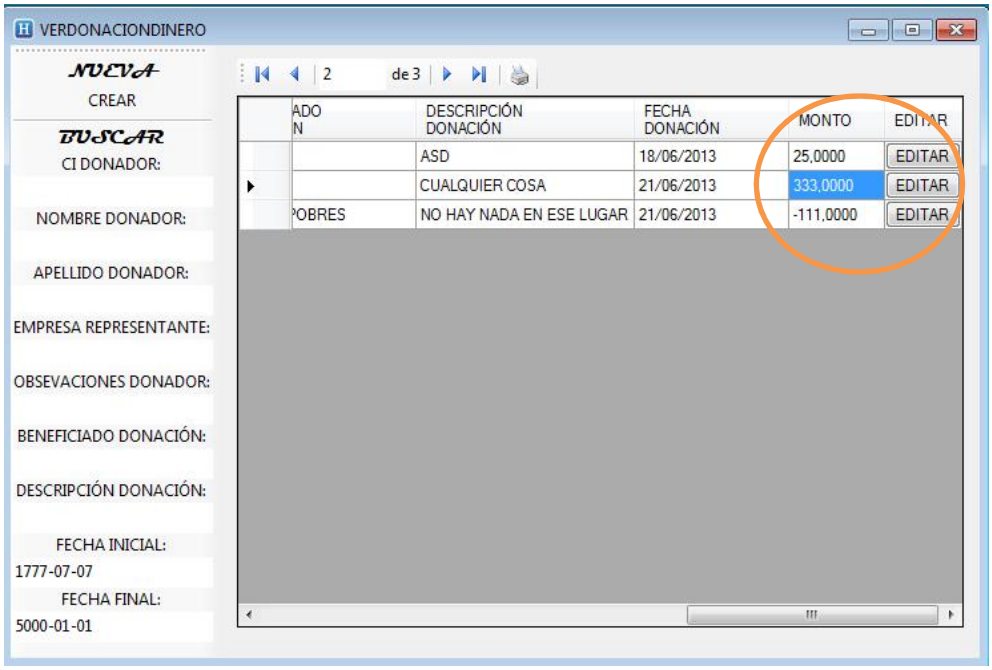
Para la edición de donaciones de dinero o artículos se lo realiza de igual manera. Para el ejemplo realizaremos una edición de donación de dinero.



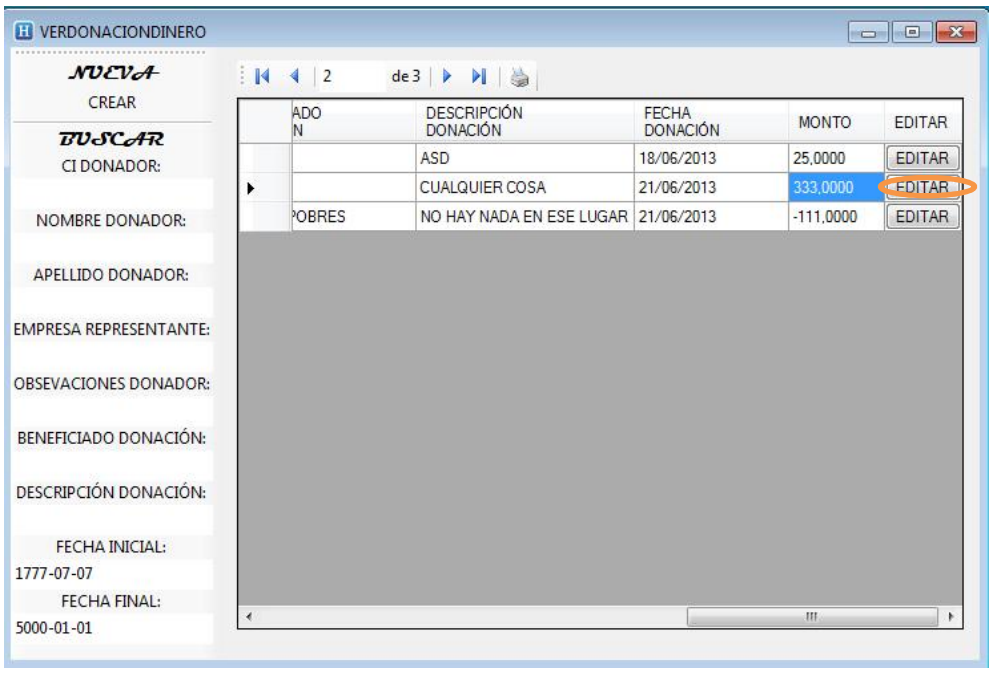
Presionamos la opción dinero y vemos los registros de donaciones de dinero realizadas



Con el desplazador horizontal de la tabla de donaciones de dinero recorremos todo hacia la derecha al final de la tabla donde podemos encontrar el botón EDITAR. En la tabla podemos encontrar valores negativos como positivos indicando que si es POSITIVO es ingreso de dinero y si es NEGATIVO es salida de dinero.



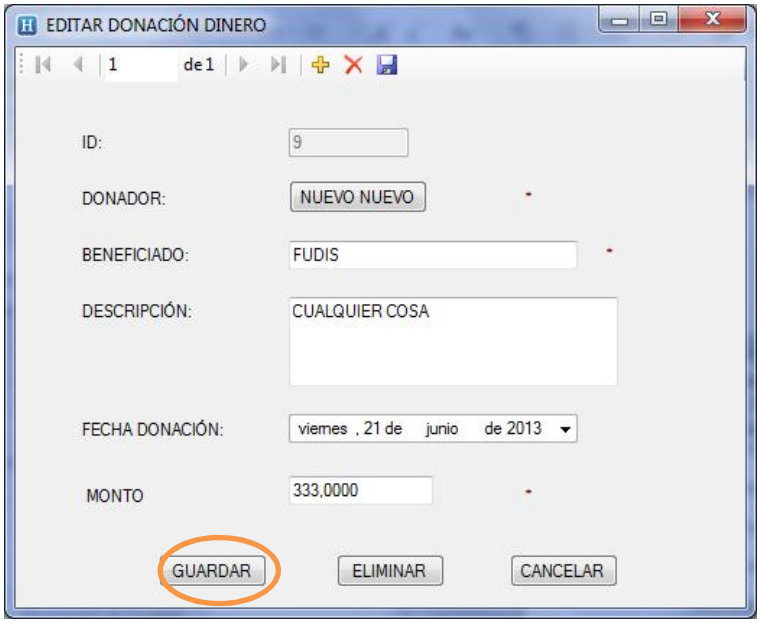
Presionamos el botón EDITAR en el registro que deseamos modificar



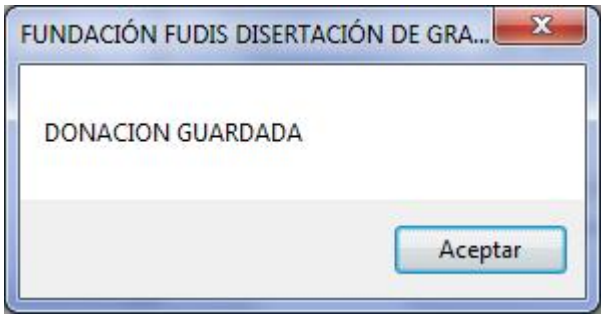
En este formulario encontramos la información del registro a modificarse.



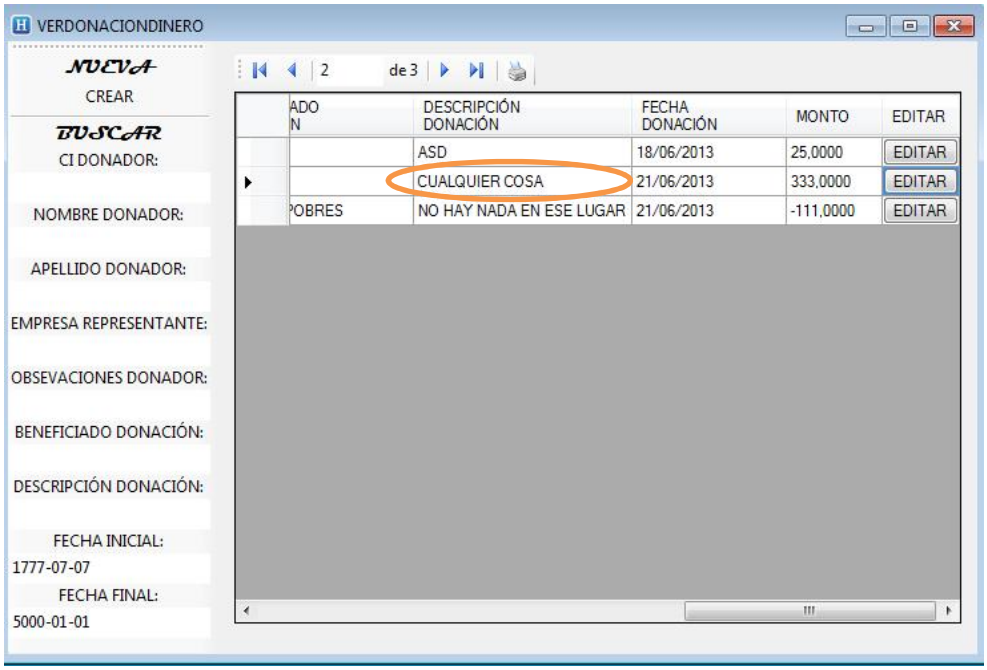
En este caso vamos a cambiar la descripción "CUALQUIERCOSA" por "CAMBIO REALIZADO" Y Presionamos GUARDAR



Nos mostrará un mensaje de aviso de registro realizado.

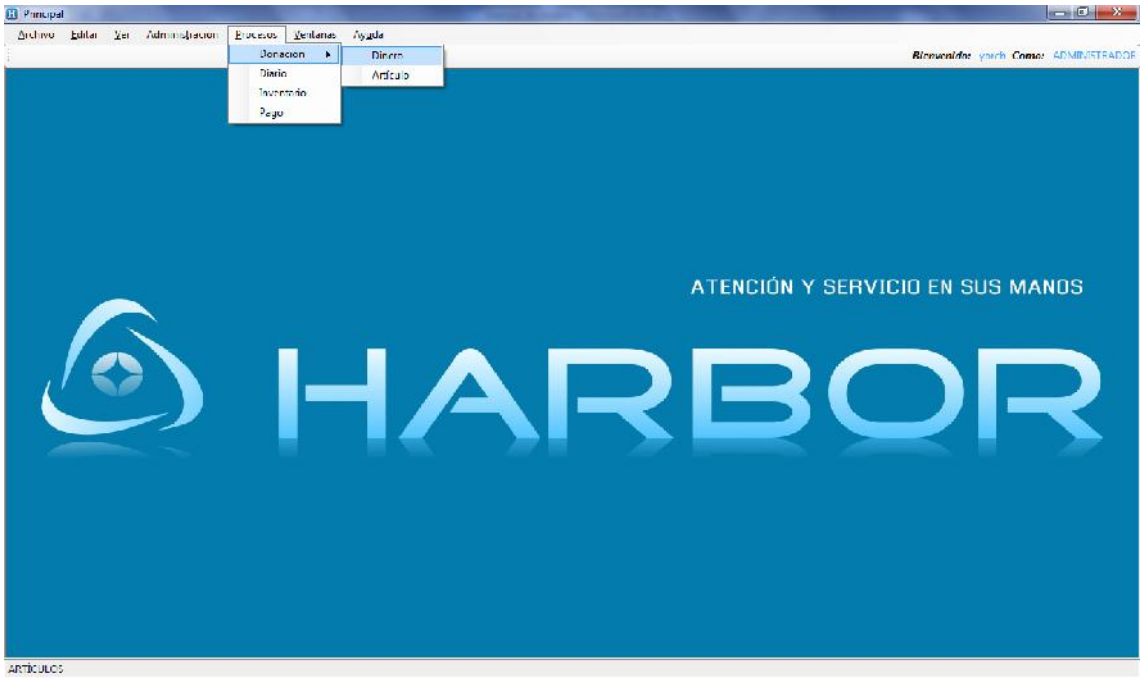


Y regresamos al formulario de donaciones de dinero y observamos el cambio realizado.

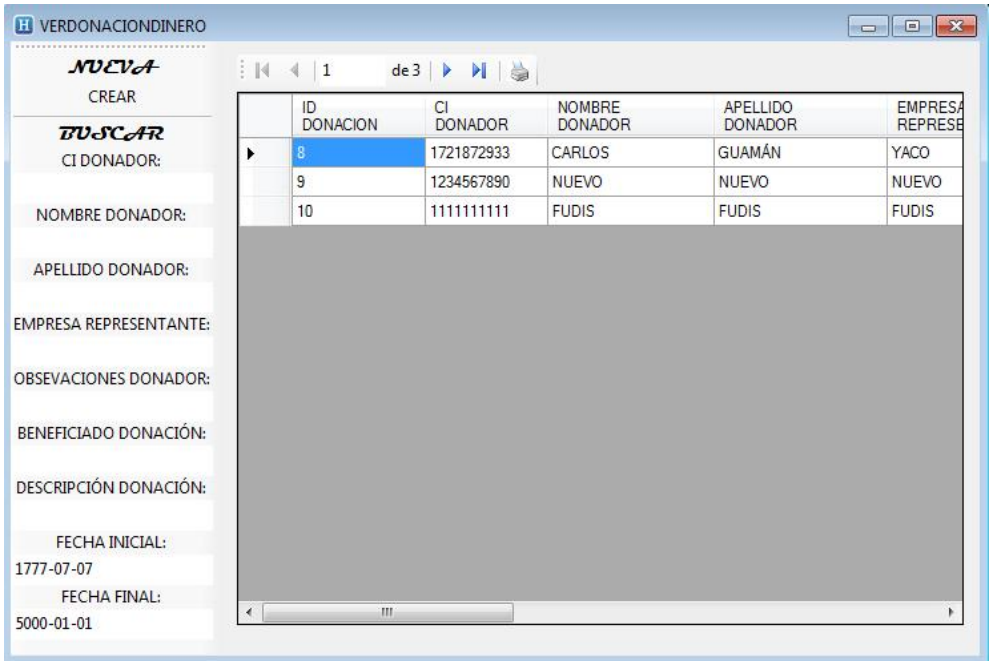


Eliminar donaciones

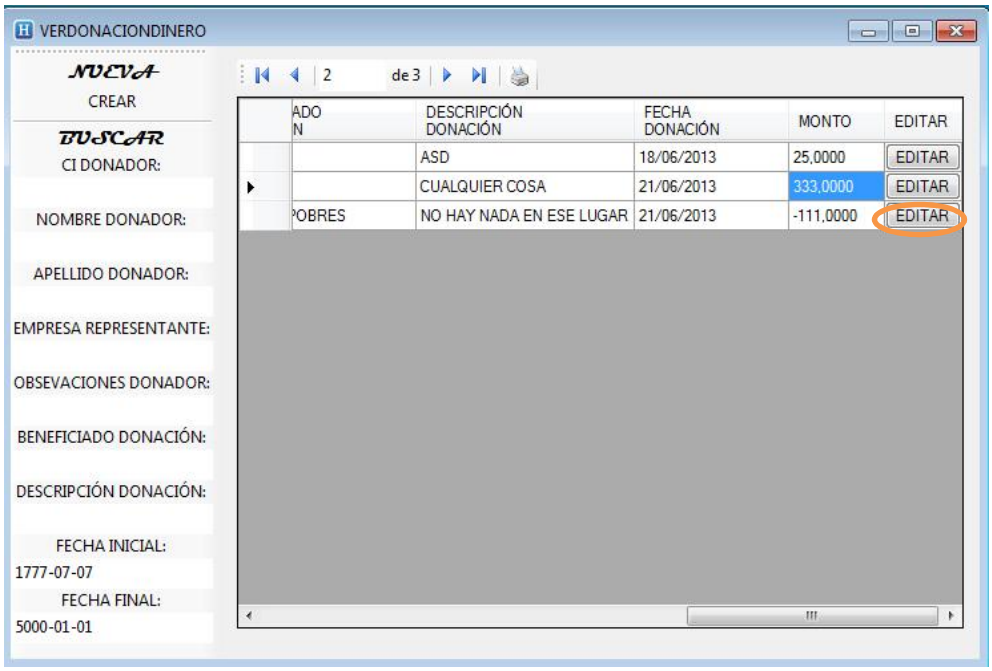
Para la edición de donaciones de dinero o artículos se lo realiza de igual manera. Para el ejemplo realizaremos una edición de donación de dinero.



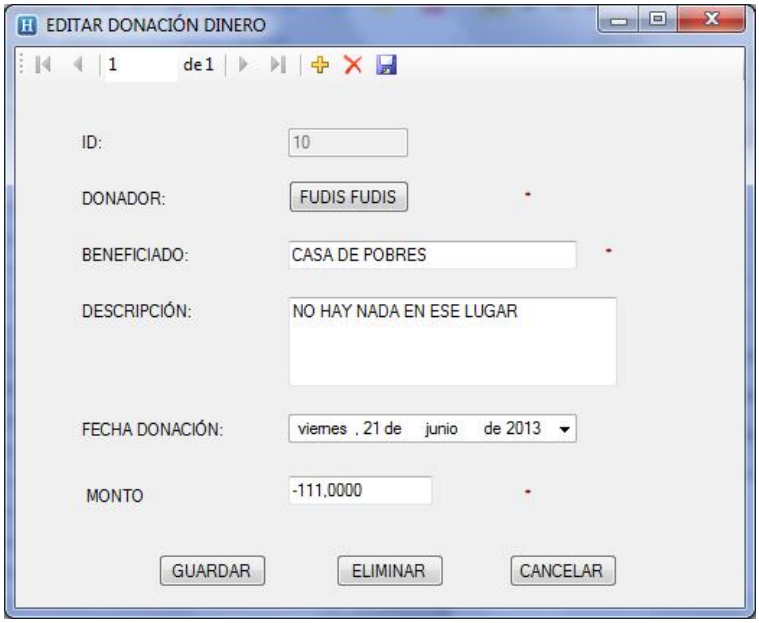
Presionamos la opción dinero y vemos los registros de donaciones de dinero realizadas



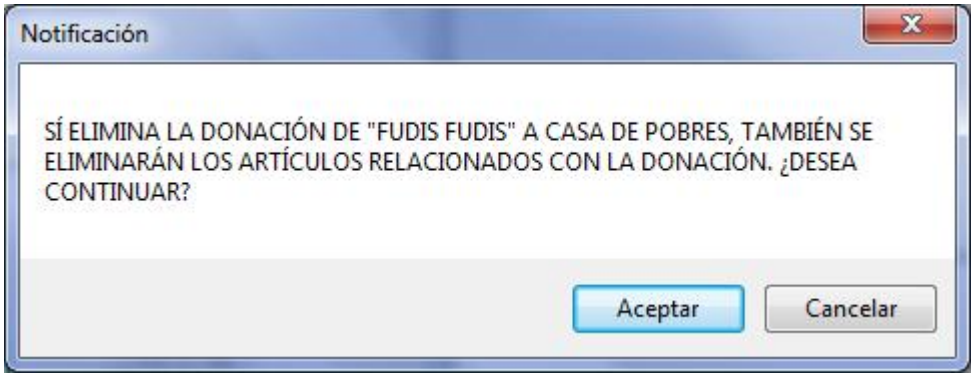
Con el desplazador horizontal de la tabla de donaciones de dinero recorremos todo hacia la derecha al final de la tabla donde podemos encontrar el botón EDITAR.



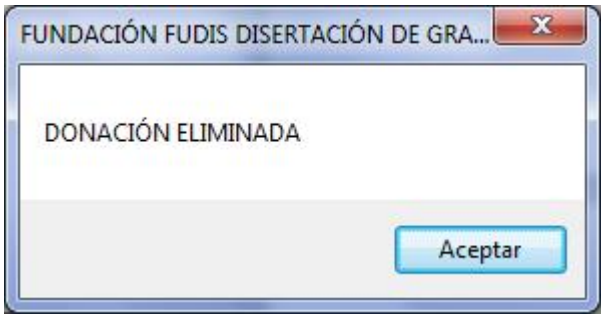
En este formulario que vemos es una donación que realizo FUDIS a otra institución



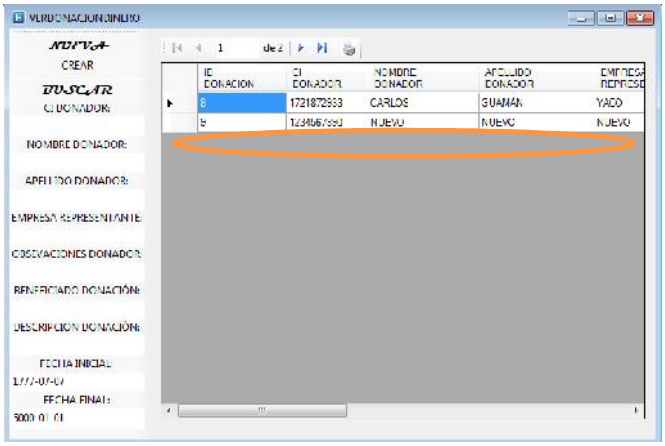
Presionamos eliminar y nos muestra un mensaje de aviso para ver si estamos seguros que deseamos eliminar este registro



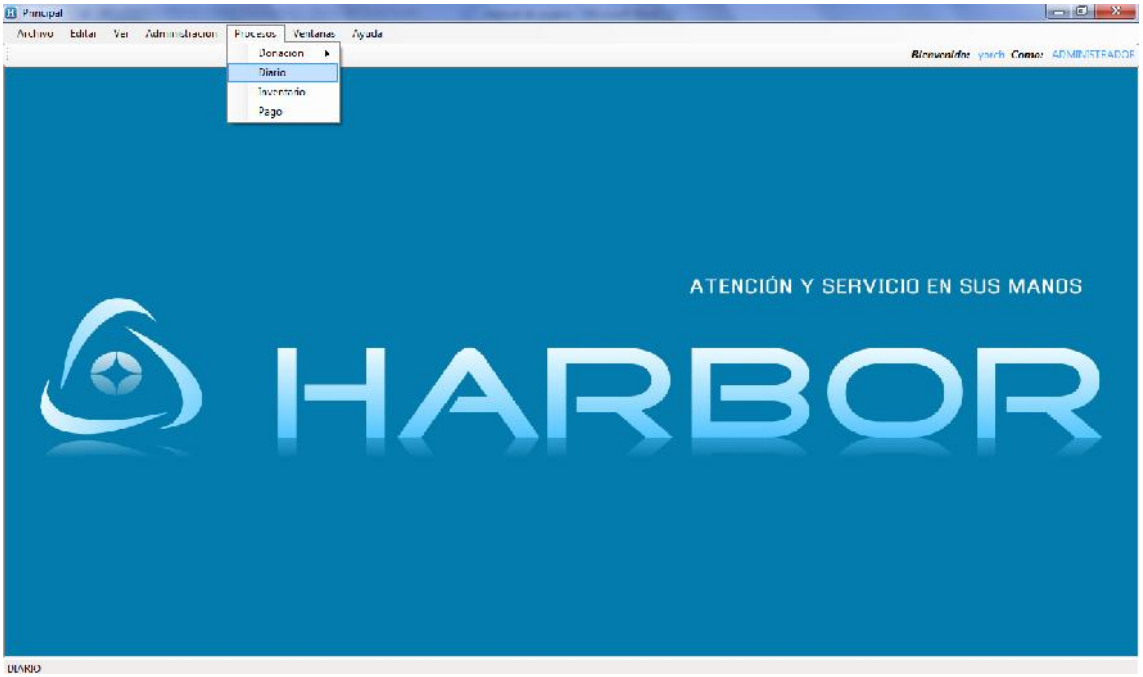
Una vez realizado esta operación nos muestra otro mensaje de aviso de eliminación



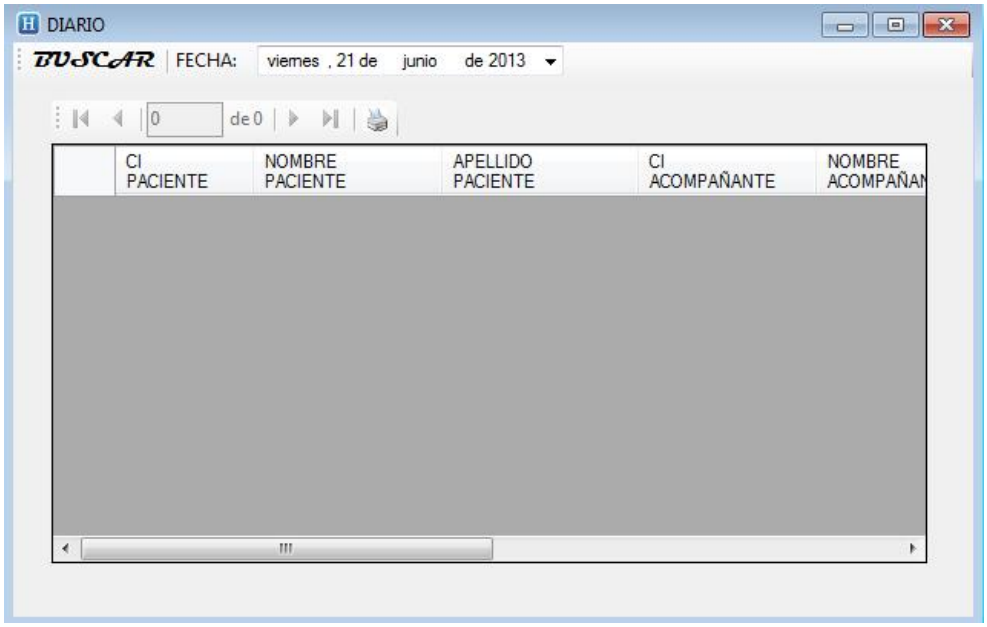
Regresamos al formulario de búsqueda de donaciones y vems que el registro de donaciones de dinero que realizo FUDIS a otra institución fue eliminado



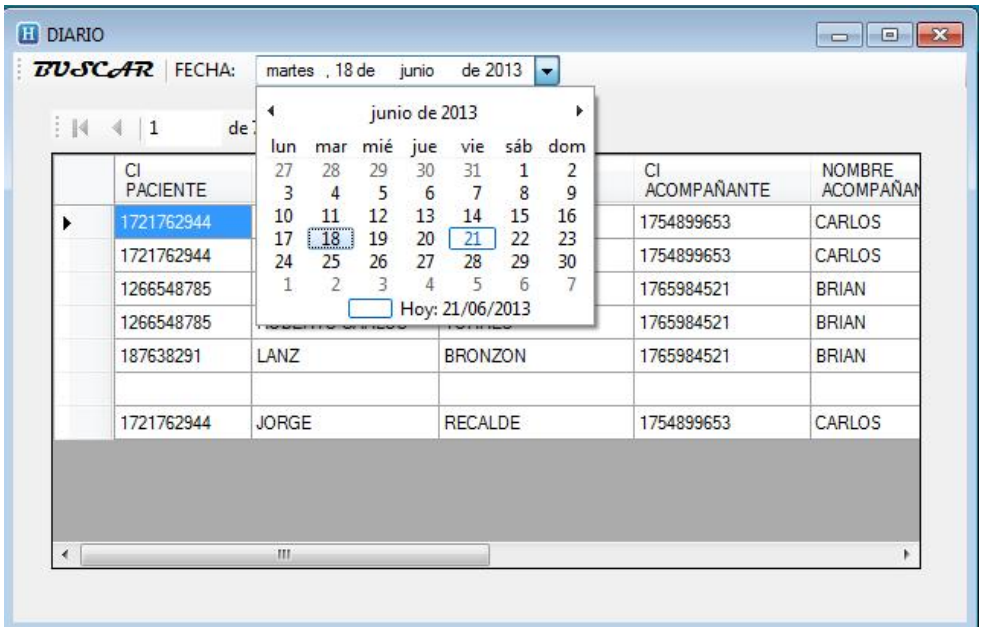
Diario



Para este formulario vamos a seleccionar la fecha en la que queremos que se muestren las transacciones realizadas el día que deseemos.



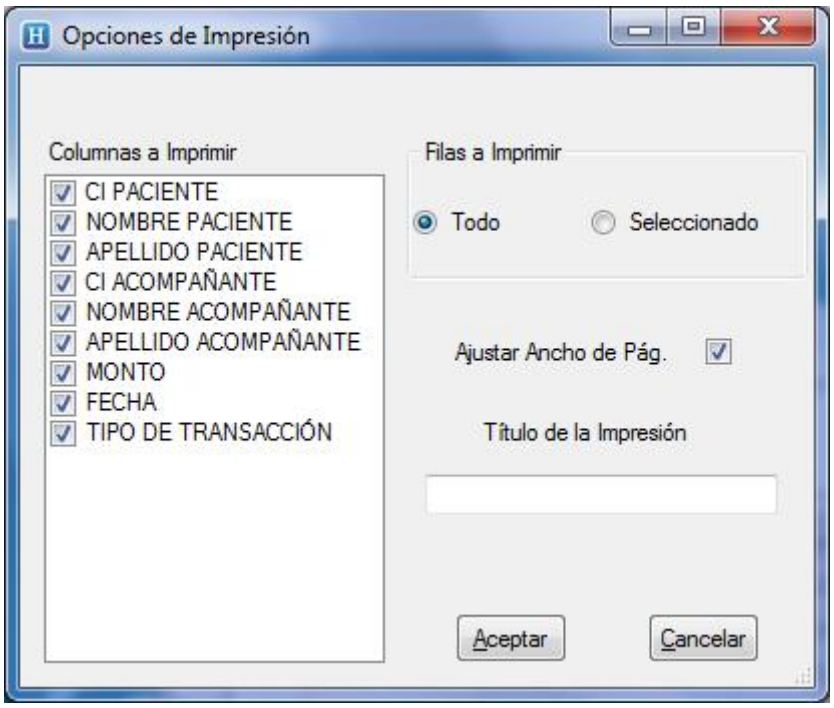
En este caso seleccionamos la fecha 18-06-2013



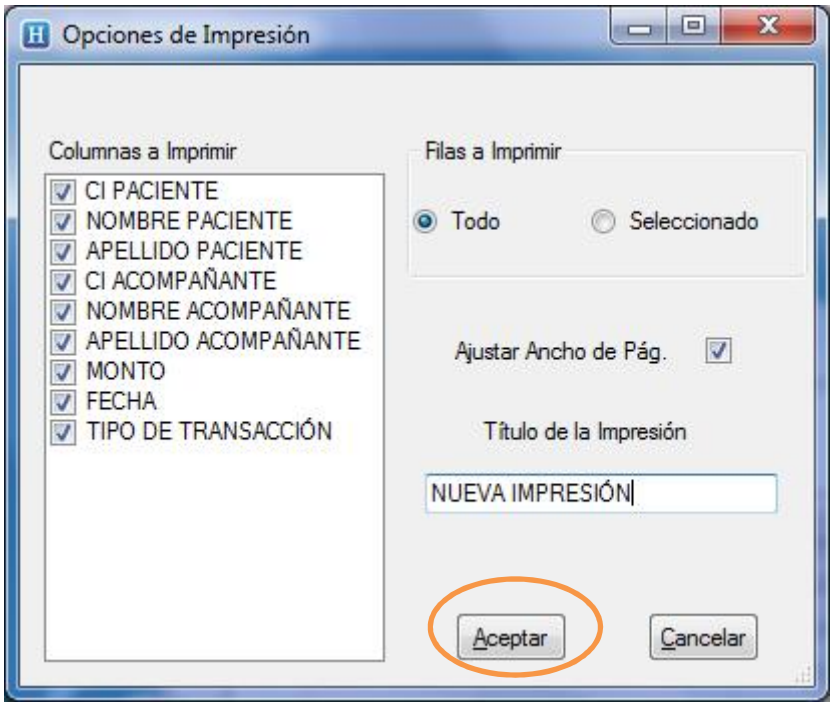
Aquí podemos observar el detalle de todas las transacciones realizadas ese día con el detalle de cada una.

Impresión Diario

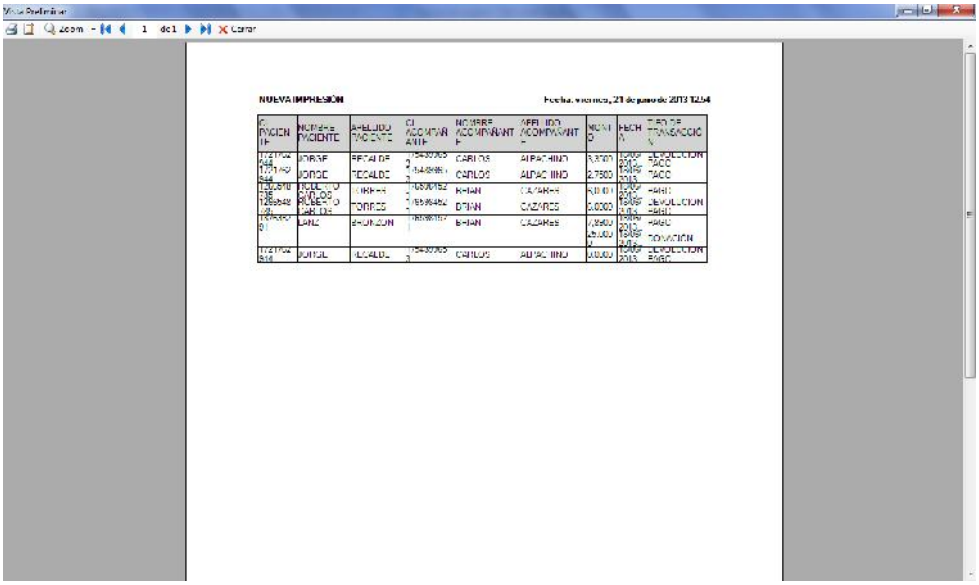
Presionamos el botón  para abrir el siguiente formulario donde encontraremos una lista de parámetros para ser impresos.



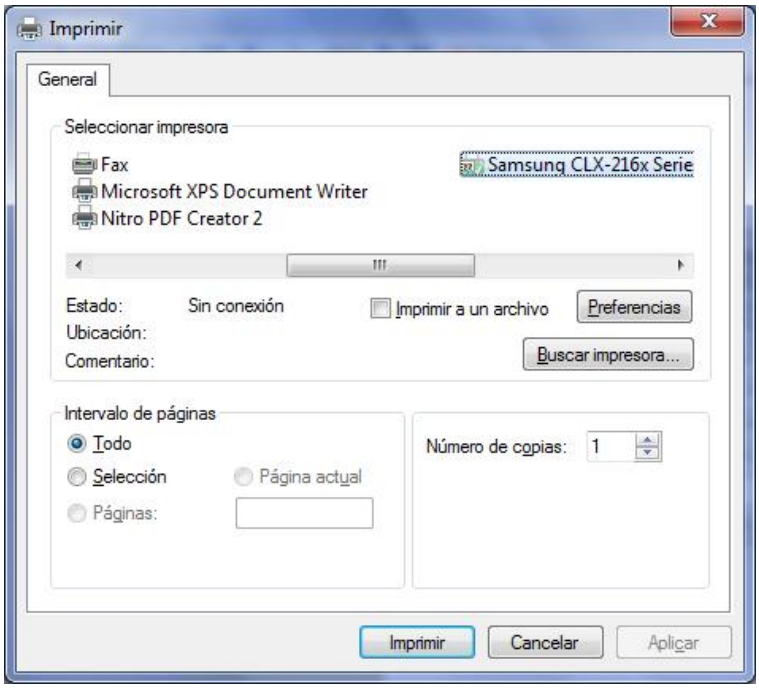
Seleccionamos estos parámetros y ponemos un título con el cual queremos que se imprima el documento y presionamos ACEPTAR.



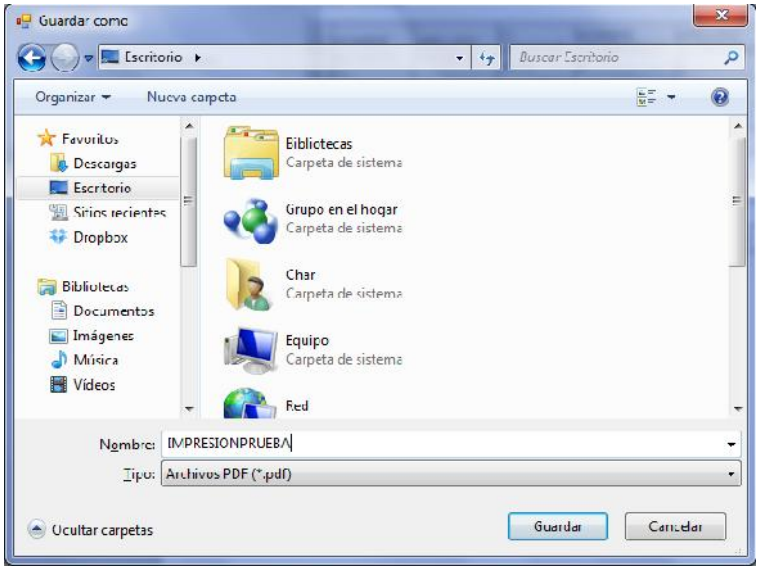
Con esto obtenemos una vista previa con los datos proporcionados por la tabla donaciones.



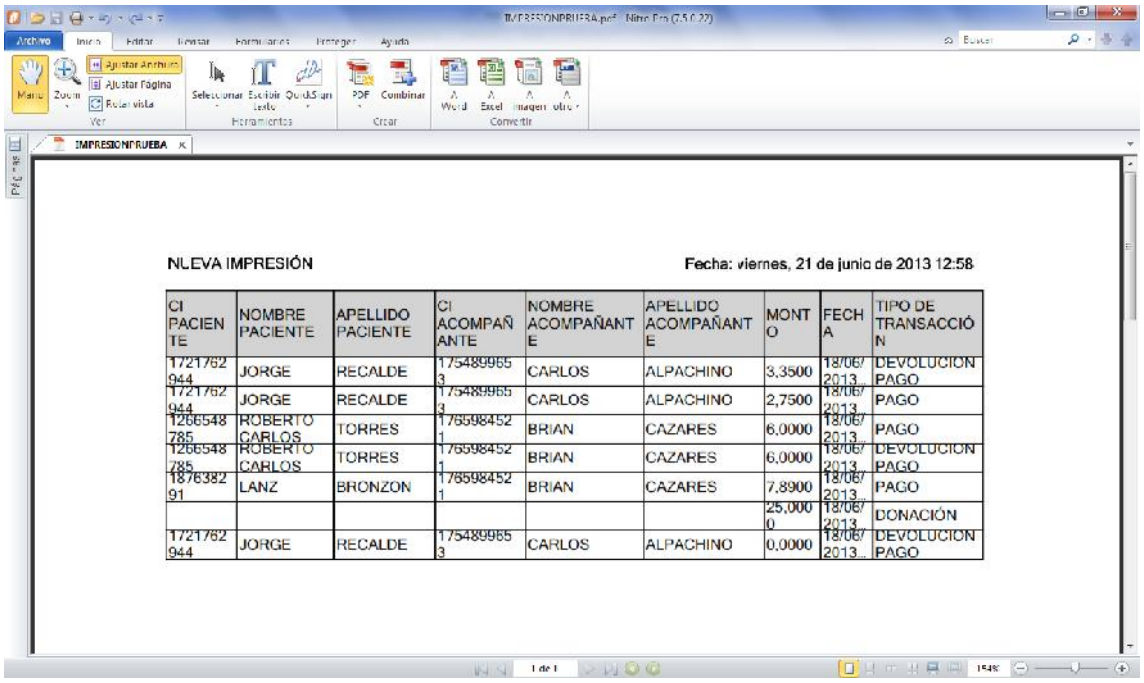
Para imprimir la hoja, presionamos el botón IMPRIMIR Y obtenemos otro formulario donde escogemos en que impresora va a ser impreso o qué tipo de documento quiere ser almacenado.



Presionamos imprimir y se ejecutara la impresión del documento. En este ejemplo lo vamos a imprimir en un documento PDF, también es necesario ubicar el lugar donde queremos guardar el documento.



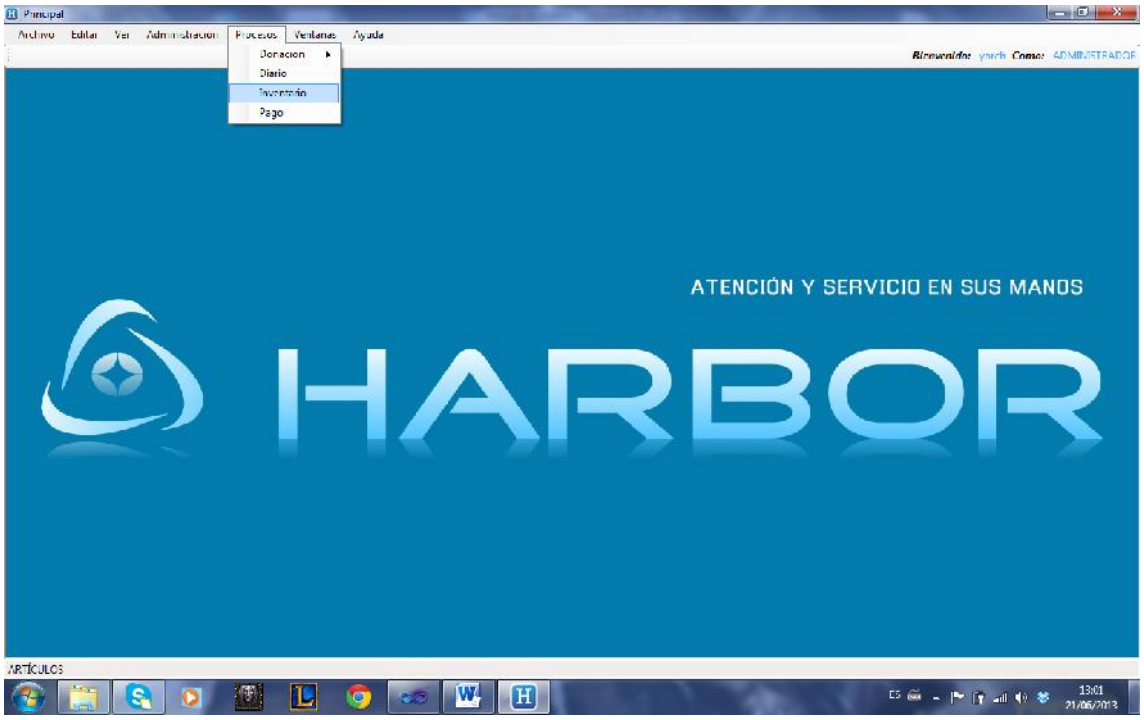
El documento resultado podemos verlo de la siguiente manera.



NUEVA IMPRESIÓN — Fecha: viernes, 21 de junio de 2013 12:58

CI PACIENTE	NOMBRE PACIENTE	APELLIDO PACIENTE	CI ACOMPAÑANTE	NOMBRE ACOMPAÑANTE	APELLIDO ACOMPAÑANTE	MONTO	FECHA	TIPO DE TRANSACCIÓN
1721762944	JORGE	RECALDE	1754899653	CARLOS	ALPACHINO	3,3500	18/06/2013...	DEVOLUCION PAGO
1721762944	JORGE	RECALDE	1754899653	CARLOS	ALPACHINO	2,7500	18/06/2013...	PAGO
1266548785	ROBERTO CARLOS	TORRES	1765984521	BRIAN	CAZARES	6,0000	18/06/2013...	PAGO
1266548785	ROBERTO CARLOS	TORRES	1765984521	BRIAN	CAZARES	6,0000	18/06/2013...	DEVOLUCION PAGO
187638291	LANZ	BRONZON	1765984521	BRIAN	CAZARES	7,8900	18/06/2013...	PAGO
						25,0000	18/06/2013...	DONACIÓN
1721762944	JORGE	RECALDE	1754899653	CARLOS	ALPACHINO	0,0000	18/06/2013...	DEVOLUCION PAGO

Inventario

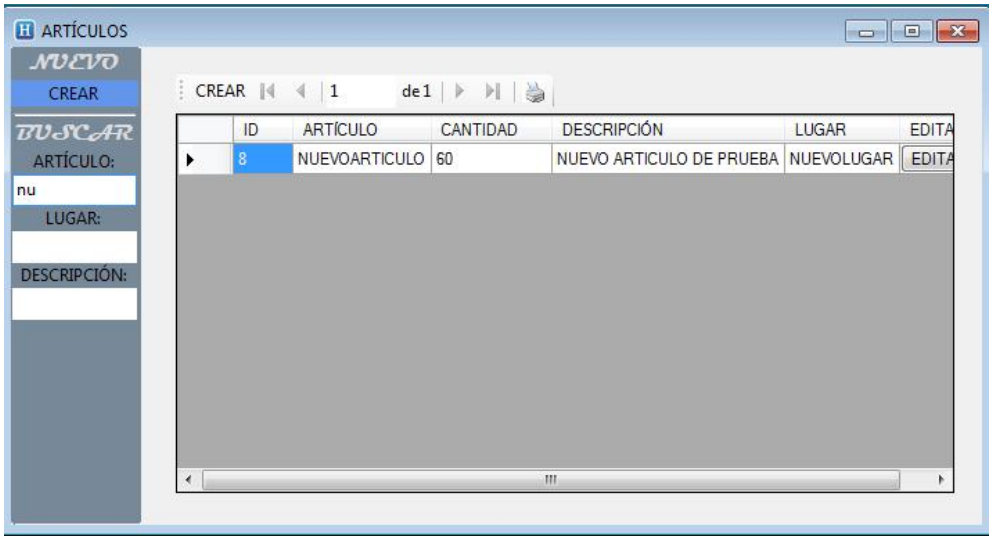


En el formulario de inventario podeos ver todas los artículos que posee FUDIS y en el lugar donde se almacena.



ID	ARTÍCULO	CANTIDAD	DESCRIPCIÓN	LUGAR	EDIT
1	PANTALONES	12	ROTOS Y FEOS	BODEGA1	EDIT
2	PARES DE MEDIAS	29	USADAS	BODEGA2	EDIT
3	PLATOS	359276	NUEVOS	BODEGA3	EDIT
4	ZAPATOS	70	ROTOS Y FEOS	BODEGA3	EDIT
5	LAMPARAS	3		BODEGA3	EDIT
6	HIPSTERS	4	GAFAS	BODEGA2	EDIT
7	ROJO	4		ALACENA	EDIT
8	NUEVOARTICULO	60	NUEVO ARTICULO DE PRUEBA	NUEVOLUGAR	EDIT

En este formulario podemos buscar el artículo que queremos ver a detalle por ejemplo buscaremos el artículo “NUEVOARTICULO” y nos mostrará el detalle de ese artículo


ARTÍCULOS
NUEVO
CREAR
BUSCAR
ARTÍCULO:
nu
LUGAR:
DESCRIPCIÓN:
CREAR
1
de 1
ID
ARTÍCULO
CANTIDAD
DESCRIPCIÓN
LUGAR
EDITA
8
NUEVOARTICULO
60
NUEVO ARTICULO DE PRUEBA
NUEVOLUGAR
EDITA