

**UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS**



**“DESARROLLO DE UN SISTEMA PARA LA AUTOMATIZACIÓN DE LA
GESTION DE LAS PARTIDAS DE NACIMIENTO DE UN REGISTRO CIVIL
VENEZOLANO BAJO LICENCIA DE SOFTWARE LIBRE”**

REALIZADO POR:

Br. Hector Cova

C.I.: 16.562.748

Br. Rafael Solorzano

C.I.: 16.478.353

Trabajo de grado presentado como requisito parcial para optar al Título de

INGENIERO EN COMPUTACIÓN

Puerto la Cruz, Agosto de 2010.

**UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS**



**“DESARROLLO DE UN SISTEMA PARA LA AUTOMATIZACIÓN DE LA
GESTION DE LAS PARTIDAS DE NACIMIENTO DE UN REGISTRO CIVIL
VENEZOLANO BAJO LICENCIA DE SOFTWARE LIBRE”**

ASESOR:

**Lic. José Bastardo Msc.
Asesor Académico**

Puerto la Cruz, Agosto de 2010.

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UN SISTEMA PARA LA AUTOMATIZACIÓN DE LA
GESTION DE LAS PARTIDAS DE NACIMIENTO DE UN REGISTRO CIVIL
VENEZOLANO BAJO LICENCIA DE SOFTWARE LIBRE”**

JURADO CALIFICADOR:

Lic. José Bastardo Msc.
Asesor Académico

Ing. Claudio Cortinez
Jurado Académico

Ing. Manuel Carrasquero
Jurado Académico

Puerto la Cruz, Agosto de 2010.

RESOLUCIÓN

ARTÍCULO N° 41 Del Reglamento Del Trabajo de Grado

“Los trabajos de grado son de exclusiva propiedad de la Universidad y sólo podrán ser utilizados para otros fines con el conocimiento del Consejo de Núcleo respectivo, quién lo participará al Consejo Universitario”.

AGRADECIMIENTOS

Agradezco únicamente a mi madre como persona.

Y agradezco a Ludwig van Beethoven y al Dota como entes de support.

Rafael Ricardo Solórzano Robles

AGRADECIMIENTOS

Primeramente a Dios. A mi madre y mi padre, ya me gradué (¿por fin no?).

Hector Manuel Cova Rosario

RESUMEN

Los registros civiles venezolanos son los entes gubernamentales establecidos para mantener las inscripciones de los hechos y actos vitales de las personas que modifican su estado civil, su relación familiar y su identificación personal. Por lo que dichos registros requieren de numerosos procesos de gestión con un cierto grado de complejidad. De esta manera inmediatamente surge la necesidad del desarrollo de un sistema para automatizar los procesos y procedimientos de los registros civiles venezolanos. En este trabajo se propone la automatización de las actividades asociadas a las partidas de nacimiento en dichos registros. Como lugar para desarrollar y probar el sistema, se escogió la alcaldía de Guanta, debido a que sufre la necesidad de una automatización de su registro civil. El sistema propuesto brindaría soluciones al registro y gestión de ciudadanos, gestión de partidas de nacimiento y al control de los usuarios, todo esto dividido entre una aplicación de escritorio para los funcionarios, una página web para todas las personas y una base de datos centrada en el ciudadano que almacena toda la información pertinente a la elaboración de los documentos del registro civil. El diseño y desarrollo del proyecto se realizara según las fases de Proceso Unificado de Desarrollo de software y utilizando el Lenguaje Unificado de Modelado UML 2.0. El sistema se construirá e implementara utilizando los lenguajes C# y PHP, y el gestor de bases de datos PostgreSQL.

ÍNDICE GENERAL

RESOLUCIÓN	iv
AGRADECIMIENTOS	v
RESUMEN	vii
ÍNDICE GENERAL	viii
ÍNDICE DE TABLAS	xiii
ÍNDICE DE FIGURAS	xv
CAPÍTULO I: INTRODUCCIÓN	19
1.1 Planteamiento del problema	19
1.1.1 Descripción de la problemática	19
1.1.2 Propósito	20
1.1.3 Alcance	21
1.1.4 Originalidad	21
1.1.5 Importancia	22
1.2 Objetivos	22
1.2.1 Objetivo General	22
1.2.2 Objetivos Específicos	23
CAPÍTULO II: MARCO TEÓRICO	24
2.1 Antecedentes	24
2.2 Bases Teóricas	26
2.2.1 Programación orientada a objetos	26
2.2.2 Características de los lenguajes de programación orientada a objetos	27
2.2.2.1 Encapsulación	27
2.2.2.2 Herencia	28
2.2.2.3 Polimorfismo	28
2.2.2.4 Abstracción	29

2.2.3 Clases y Objetos.....	30
2.2.4 Proceso Unificado Racional (RUP).....	30
2.2.4.1 Centrado en los Modelos	31
2.2.4.2 Guiado por los Casos de Uso	31
2.2.4.3 Centrado en la Arquitectura	31
2.2.4.4 Iterativo e Incremental.....	31
2.2.5 Ciclo de vida RUP.....	31
2.2.5.1 Principales características de RUP	33
2.2.6 UML	33
2.2.7 WebML	34
2.2.8 Base de Datos	35
2.2.8.1 Elementos de almacenamiento de una Base de Datos.....	35
2.2.8.2 Tipos de Base de Datos	36
2.2.9 C#	37
2.2.10 HTML	37
2.2.11 PHP	37
2.2.12 CSS	38
2.2.13 Debían GNU/Linux.....	38
2.2.14 PostgreSQL	38
CAPÍTULO III: FASE DE INICIO	39
3.1 Introducción.....	39
3.2 Planificación de la fase de inicio.....	40
3.3 Requisitos.....	40
3.3.1 Lista de requisitos o características	40
3.3.2 Identificación de los riesgos.....	42
3.4 Modelo de dominio	44
3.5 Modelo de casos de usos.....	46
3.5.1 Identificación de casos de usos	46
3.6 Diagrama de actividad.....	54

3.6.1 Navegar en las secciones del site	55
3.6.2 Registrarse como usuario para acceso a secciones protegidas ...	55
3.6.3 Interactuar con componentes implementados	56
3.6.4 Gestionar contenidos y parámetros	57
3.6.5 Configurar componentes de páginas	58
3.6.6 Solicitud de Documentos	58
3.6.7 Envío de Mensajes	59
3.6.8 Servicios del Registro	59
3.6.9 Gestión del usuario	59
3.6.10 Configuración.....	61
3.6.11 Ayuda.....	62
3.6.12 Cerrar sesión	62
3.7 Diagramas preliminares de clases y paquetes	62
3.7.1 Paquete SRC.....	62
3.7.2 El paquete itextsharp	63
3.7.3 El paquete Mono.Security.....	63
3.7.4 El Paquete npgsql.....	63
3.7.5 El paquete System.....	64
3.7.6 Clases del paquete Src.....	67
3.7.7 Clases de conexión a la base de datos	75
3.8 Evaluación de la Fase de inicio	79
CAPÍTULO IV: FASE DE ELABORACIÓN.....	81
4.1 Introducción.....	81
4.2 Planificación de la fase de elaboración	81
4.3 Requisitos.....	82
4.3.1 Lista de requisitos ó características	82
4.3.2 Modelo de casos de usos actualizado	83
4.3.3 Mitigación de los riesgos	84
4.4 Análisis	85

4.4.1 Diagrama de actividades	85
4.4.1.1 Navegar a una sección del website	85
4.4.1.2 Validar nivel de usuario	86
4.5 Diagrama de secuencia.....	87
4.6 Diagrama de capas	93
4.7 Diagrama de Base de Datos	95
4.7.1 Identificación de las tablas.....	95
4.7.2 Representación del diagrama relacional de base de datos	100
4.7.3 Representación de los libros dentro del registro civil	101
4.8 Evaluación de la fase de elaboración.....	102
CAPÍTULO V: FASE DE CONSTRUCCIÓN	104
5.1 Introducción.....	104
5.1.1 Planificación de la fase de construcción	105
5.1.2 Escogencia del lenguaje de programación	105
5.1.3 Escogencia del sistema gestor de base de datos.....	107
5.2 Implementación	108
5.2.1 Diagrama de Componentes Totales	108
5.3 Pruebas	110
5.3.1 Pruebas por unidad	110
5.3.2 Pruebas de Integración.....	116
5.3.2.1 Integración de solicitud de partida de nacimiento	118
5.3.2.2 Integración de certificado de nacimiento	119
5.3.2.3 Integración de personas.....	120
5.3.2.4 Procedimiento de prueba del sistema	120
CAPÍTULO VI: FASE DE TRANSICIÓN	141
6.1 Introducción.....	141
6.2 Mejoras Realizadas al Sistema	141
6.3 Manual de Usuario	142
CONCLUSIONES	144

RECOMENDACIONES..... 146
METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO..... 147

ÍNDICE DE TABLAS

Tabla 3.1 Lista de riesgos y contingencia	43
Tabla 3.2 Casos de uso Navegar en las secciones del site administradas por el sistema.....	48
Tabla 3.3 Casos de uso Interactuar con componentes implementados.....	49
Tabla 3.4 Casos de uso Registrarse como usuario para acceso a secciones protegidas	49
Tabla 3.5 Casos de uso Solicitud de documentos	49
Tabla 3.6 Casos de uso Envío de mensajes.....	50
Tabla 3.7 Casos de uso Gestionar contenidos y parámetros	50
Tabla 3.8 Casos de uso Configurar componentes en páginas	50
Tabla 3.9 Casos de uso Servicios del Registro.....	51
Tabla 3.10 Casos de uso Gestión del usuario	51
Tabla 3.11 Casos de uso Configuración	51
Tabla 3.12 Casos de uso Ayuda	52
Tabla 3.13 Casos de uso Cerrar sesión.....	52
Tabla 3.14 Casos de uso Declaraciones	52
Tabla 3.15 Casos de uso Copias certificadas	52
Tabla 3.16 Casos de uso Nacimientos	53
Tabla 3.17 Casos de uso Ordinario (Ambos Padres).....	53
Tabla 3.18 Casos de uso Ordinario (Madre).....	53
Tabla 3.19 Casos de uso Reconocimiento Posterior (Paternidad).....	54
Tabla 3.20 Casos de uso Inserción de Acta de Nacimiento.....	54
Tabla 3.21 Descripción de la clase DB	75
Tabla 3.22 Descripción de la clase ConectarPostgres.....	76
Tabla 3.23 Descripción de la clase ConectarPostgresAux	77
Tabla 4.1 Casos de uso Validar acceso como usuario registrado	84

Tabla 4.2 Casos de uso Validar nivel de usuario	84
Tabla 4.3 Tabla de base de datos que conforman el sistema.....	95
Tabla 4.4 Representación de los libros dentro de un registro civil	102
Tabla 5.1: Clases de equivalencia del componente de solicitud de partidas.	110
Tabla 5.2: Clases de equivalencia del componente certificados de nacimiento.	111
Tabla 5.3: Clases de equivalencia del componente de personas.	112
Tabla 5.4: Casos de prueba para el componente de solicitud de partidas..	114
Tabla 5.5: Casos de prueba para el componente certificados de nacimiento.	114
Tabla 5.6: Casos de prueba para el componente de personas.	115
Tabla 5.7: Leyenda de colores para fases del diagrama de componentes totales	117
Tabla 5.8: Integración para el componente de solicitud de partidas de nacimiento.	119
Tabla 5.9: Integración para el componente de solicitud de certificado de nacimiento.	119
Tabla 5.10: Integración para el componente de personas.	120

ÍNDICE DE FIGURAS

Figura 3.1 Diagrama de flujo de trabajo de las fases del proceso unificado, con identificación de la fase conceptualización (inicio).....	39
Figura 3.2 Modelo de dominio del sistema	46
Figura 3.3 Diagrama de casos de uso	47
Figura 3.4 Diagrama de casos de uso	48
Figura 3.5 Diagrama de actividad Navegar a una sección del website.....	55
Figura 3.6 Diagrama de actividad registrarse como usuario.....	56
Figura 3.7 Diagrama de actividad Interactuar con componentes implementados.....	57
Figura 3.8 Diagrama de actividad Gestionar contenidos y parámetros.....	57
Figura 3.9 Diagrama de actividad configurar componentes en páginas	58
Figura 3.10 Diagrama de actividad Solicitud de documentos	60
Figura 3.11 Diagrama de actividad gestión del usuario	60
Figura 3.12 Diagrama de actividad servicios del registro.....	61
Figura 3.13 Diagrama de actividad gestión del usuario	61
Figura 3.14 Diagrama de paquetes para Src.....	65
Figura 3.15 Diagrama de paquetes Npgsql y ItextSharp	66
Figura 3.16 Diagrama de paquetes System.....	66
Figura 3.17 Diagrama de clases para ClaseValidacion, Persona, CertificadoNac, RegistroCivil, Usuario (modo compacto)	68
Figura 3.18 Diagrama de clases para Persona, CertificadoNac, RegistroCivil, Usuario (modo expandido).....	69
Figura 3.19 Diagrama de clases para ClaseValidacion (modo expandido)...	70
Figura 3.20 Diagrama de clases para EngineForm (modo compacto).....	71
Figura 3.21 Diagrama de clases para EngineForm (modo expandido).....	72
Figura 3.22 Diagrama de clases para EngineUc (modo compacto).....	73

Figura 3.23 Diagrama de clases para EngineUc (modo expandido).....	74
Figura 3.24 Diagrama de clases para StaticVar, iTextSharpEvents, BitmapFilter, ClaseGeneradora, EngineShowToolTips, Consola.....	74
Figura 3.25 Diagrama de clases para DB	75
Figura 3.26 Diagrama de clases para ConectarPostgres y ConectarPostgresAux (modo expandido)	78
Figura 4.1 Diagrama de flujo de trabajo de las fases del proceso unificado, con identificación de la fase de elaboración.	81
Figura 4.2 Diagrama de casos de uso	83
Figura 4.3 Diagrama de actividad Navegar en las secciones del website	86
Figura 4.4 Diagrama de actividad Navegar en las secciones del website	87
Figura 4.5 Diagrama de actividad Navegar en las secciones del website	89
Figura 4.6 Diagrama de secuencia actualizar ciudadano	90
Figura 4.7 Diagrama de secuencia Acta Nacimiento Ordinaria	91
Figura 4.8 Diagrama de secuencia Acta Nacimiento Ordinaria	93
Figura 4.9 Diagrama de capas del sistema.....	94
Figura 4.10 diagrama del modelo relacional.....	100
Figura 5.1 Diagrama de flujo de trabajo de las fases del proceso unificado, identificación de la fase de construcción.....	104
Figura 5.2: Interfaz del entorno de programación # Develop.	106
Figura 5.3: Interfaz del entorno de programación Kate.....	107
Figura 5.4: Interfaz de pgAdmin III.....	108
Figura 5.5: Diagrama de componentes totales para el sistema.	109
Figura 5.6: Diagrama de componentes por fase de integración.	118
Figura 5.7: Ejecución del SRC en plataforma Windows XP	121
Figura 5.8: Componente principal recibiendo la petición de actualización del ciudadano	122
Figura 5.9: Componente FormPersona recibiendo la actualización pertinente	123

Figura 5.10: Componente PerfilUsuario recibiendo la actualización pertinente	124
Figura 5.11: Componente PerfilUsuario recibiendo la actualización pertinente	125
Figura 5.12: Componentes ActaNac2Padres, EngineUCRequisitos	126
Figura 5.13: Componentes ActaNac2Padres, QuienPresenta	127
Figura 5.14: Componentes ActaNac2Padres, FormPersona, solicitando datos del padre	128
Figura 5.15: Componentes ActaNac2Padres, FormPersona, solicitando datos de la madre	129
Figura 5.16: Componentes ActaNac2Padres, FormPersona, solicitando datos del primer testigo	130
Figura 5.17: Componentes ActaNac2Padres, FormPersona, solicitando datos del segundo testigo.....	131
Figura 5.18: Componentes ActaNac2Padres, Defensoria, solicitando datos del nacimiento.....	132
Figura 5.19: Componentes ActaNac2Padres, Certificado, solicitando datos del certificado de nacimiento.....	133
Figura 5.20: Componentes ActaNac2Padres, EngineUCRegistroCivil, mostrando los datos elaborados	134
Figura 5.21: Programa evince mostrando el acta de nacimiento elaborada por el sistema en formato pdf.....	135
Figura 5.22: Programa evince mostrando la copia certificada del acta de nacimiento en formato pdf.	136
Figura 5.23: Página principal del portal web.	137
Figura 5.24: Pagina de registro.....	137
Figura 5.25: Pagina principal con usuario identificado.....	138
Figura 5.26: Pagina para envió de comunicado.....	139
Figura 5.27: Pagina para solicitud de partida de nacimiento.	140

Figura 5.28: Pagina descarga de documentos solicitados.....	140
Figura 6.1 Diagrama de flujo de trabajo de las fases del proceso unificado, identificación de la fase de transición.	141

CAPÍTULO I: INTRODUCCIÓN

1.1 Planteamiento del problema

1.1.1 Descripción de la problemática

Desde el año 1867 al 1940 las actas del estado civil de las personas se redactaban de manera manuscrita, narrando los acontecimientos. A partir de 1941 se incorporó un formato que indicaba la manera de llenarlos, el llenado seguía siendo de manera manuscrita. En 1981 es cuando se adopta el formato de imprenta y a partir de 1982 el llenado es de manera mecanográfica, además se incorpora la CRIP (Clave de Registro de Identidad Personal), que fue concebida como un elemento de información que permite identificar y vincular en forma única y personal a cada individuo, a través de una estructura de 18 dígitos, en el Sistema Nacional de Registro de Población.

Las inscripciones de los distintos actos registrales como: actas de nacimientos, matrimonios, y defunciones, se realizaban con máquinas de escribir mecánicas, ocasionando el desperdicio de formas valoradas y propiciando la lentitud en la atención al ciudadano. También la expedición de copias certificadas se realizaba en máquina de escribir, por lo que el proceso era lento, retrasándose en la generación de cada copia certificada. Además al cometer errores los datos que no se podían borrar, por lo que se hacía una reposición, todo ello ocasionaba un descontrol en la integración de los expedientes.

1.1.2 Propósito

Como solución a la problemática actual de la gestión del registro civil, nace la necesidad de contar con un sistema que permita administrar los procesos de la gestión de las partidas de nacimiento que pueda brindar los siguientes beneficios a la institución:

- Drástica reducción de los costos operativos.
- Disponibilidad inmediata de la información.
- Intercambio instantáneo de la información.
- Rapidez en la toma de decisiones.
- Actualización constante de la base de datos.
- Mayor eficiencia.
- Información oportuna.

Por esta causa se desarrollará una forma sistemática de las actas de las partidas de nacimiento para consulta e intercambio de información, reduciendo el tiempo y el desperdicio de formas valoradas en la expedición de copias certificadas e inscripción de los distintos actos registrales, mediante el llenado a través de la computadora.

La automatización del Registro Civil es un paso importante, en la medida en que permite organizar la información ya existente creando una base de datos única de cada persona, la cual sería actualizada de manera permanente con los cambios que se operen en su vida civil, permitiendo realizar las comparaciones y cotejos correspondientes.

Gracias a este Sistema se capturará la información existente de todos los actos registrales referentes al nacimiento.

Se considera a la captura como el proceso a través del cual se pasa, traslada o lleva en medios magnéticos u ópticos la información de los actos y hechos del estado civil de las personas que sólo se tenían en papel y que eran elaboradas y archivadas manualmente asentados en libros y formatos, acción necesaria para generar y conformar la Base de Datos Nacional.

Dentro de la Automatización del Registro Civil se busca alcanzar un desarrollo tecnológico de la institución registral, el cual incluye la incorporación de equipo de cómputo para realizar las diversas actividades y procedimientos, esto orientado a incrementar la eficiencia en los servicios prestados.

1.1.3 Alcance

El alcance de este trabajo es bien visible, es el desarrollar e implementar un sistema que automatice la gestión del registro civil de Guanta, en lo que a partidas de nacimiento corresponde y proporcionar un portal web que brinde beneficios a los usuarios, dejando una plataforma libre para la continuación, mejora y mantenimiento futuro.

1.1.4 Originalidad

Algunos gobiernos estatales de Venezuela, en ocasiones anteriores, han presentado la intención de automatizar los registros civiles del país, sin embargo esto nunca se ha visto reflejado en una implementación estable y de uso masivo. Recientemente la vicepresidenta del Consejo Nacional

Electoral (CNE), Sandra Oblitas, aseguró que la automatización de los datos del Registro Civil tomara al menos 2 años en ponerse en marcha a través de las 337 alcaldías. Dicha ley está contemplada en la Ley Orgánica que regula esta materia y entró en vigencia el pasado 15 de marzo de 2009. Mientras que en el área WEB, las alcaldías si han reflejado un uso estable de portales web, aunque limitados a brindar información, sin implementar una interacción directa con el registro civil.

1.1.5 Importancia

La importancia de este proyecto es grande, y su necesidad aun mayor, ya que los procesos que llevan a cabo los registros civiles son primordiales para la sociedad venezolana, y desde hace mucho tiempo se hace evidente la necesidad de una automatización de estos procesos.

El desarrollo e implementación de este sistema sembrara la semilla para el resto del país, funcionando como, incentivo, base y punto de partida para la automatización de muchas áreas que lo requieren en el país, como lo son los registros civiles de toda la nación.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un sistema para la automatización de la gestión de partidas de nacimiento de un registro civil venezolano bajo la licencia de software libre.

1.2.2 Objetivos Específicos

- Recolectar toda la información acerca del proceso de declaración para el trámite de las partidas de nacimiento según el manual de normas y procedimientos del registro civil proporcionado por el consejo nacional electoral (CNE).
- Diseñar la arquitectura de software, definiendo todos los diagramas asociados.
- Realizar el diseño de interfaz de la aplicación que servirá como medio de comunicación entre el programa y el usuario, así como también los módulos adicionales añadidos a la arquitectura.
- Realizar el diseño de interfaz web que servirá como medio de comunicación entre el sistema del registro civil y el público en general, así como también los módulos adicionales añadidos a la arquitectura.
- Realizar la codificación de todos los componentes diseñados.
- Realizar pruebas, tanto de unidad como de integración realizando las depuraciones respectivas.

CAPÍTULO II: MARCO TEÓRICO

2.1 Antecedentes

El desarrollo de este proyecto se sustentará en la revisión bibliográfica de los trabajos de grado reseñados a continuación, los cuales servirán de apoyo en cuanto a las metodologías y técnicas utilizadas así como en los resultados arrojados por dichos estudios:

- González J., Solórzano L. (2009) presentaron un Trabajo de Grado titulado "Desarrollo de una aplicación web para el control de los expedientes en el departamento de seguridad de una entidad financiera". El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Este sistema permite la automatización de los procesos involucrados en el control de expedientes llevados a cabo por el departamento de seguridad de una entidad bancaria.
- Flores R., Tinoco M. (2008) "Desarrollo de un sistema para la automatización de la gestión de actividades de la delegación de relaciones inter-institucionales de la Universidad de Oriente Núcleo Anzoátegui". El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Este sistema consiste en el desarrollo de un software para la gestión de soporte técnico y administración de los bienes de la institución. el desarrollo del proyecto estuvo basado en el proceso unificado de desarrollo.
- Rodríguez D. (2009) "Desarrollo de un sistema tipo cliente/servidor para la gestión de la instalación de los circuitos digitales y analógicos

del área de datos de una empresa de telecomunicaciones ubicada en el estado Anzoátegui". El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Este sistema contempla la gestión de las actividades llevadas a cabo dentro de un departamento en la empresa CANTV, tales como control de inventario, almacenamiento y actualización de datos técnicos sobre las características de los circuitos instalados y por instalar.

- Ponce Molina J. (2009) "Desarrollo de un sistema basado en tecnología web haciendo uso de herramientas de software libre para la gestión de soporte técnico y manejo de inventario en la fundación para el desarrollo de la ciencia y tecnología del estado Anzoátegui (Fundacite)". El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Se propuso el desarrollo de un software para la gestión de soporte técnico y administración para el control interno, registrar los movimientos y rendir cuentas de los bienes muebles e inmuebles al ministerio al que están inscritos.
- Rojas Vargas H. (2009) "Desarrollo de una aplicación web para la gestión de solicitudes de pasantía y tesis de pasantías de recursos humanos de PDVSA S.A. distrito norte". El cual fue presentado en la Universidad de Oriente Núcleo Anzoátegui. Esta aplicación se utilizó para controlar las actividades dentro de la empresa, permitiendo la selección del personal aspirante a realizar importantes labores de pasantías y tesis con el fin de obtener beneficios de seguridad de la información

2.2 Bases Teóricas

2.2.1 Programación orientada a objetos

La programación orientada a objetos tiene la ventaja de ser un paradigma natural donde se pueden programar sistemas. Los seres humanos perciben el mundo como si estuviera formado por objetos: mesas, sillas, computadores, vehículos, partidos de fútbol, etc. También es un instinto humano intentar organizar esos objetos disponiéndolos de una forma concreta, optando por destacar determinadas características de algunos objetos que los destacan de otros. Los niveles de dicha categoría y los métodos de clasificación de objetos en el mundo son infinitos. La forma en que las personas clasifican las cosas depende en gran medida de lo que deseen hacer con ellas y las características que más le llamen la atención. A la vez que se agrupan los objetos atendiendo a esquemas de clasificación, también se tiende a resaltar determinados atributos de objetos mostrando su preferencia sobre otros. Esta idea de crear jerarquías de objetos relacionados se utiliza en la programación orientada a objetos.

En 1960, los investigadores ya observaron que muchas de las entidades del modelo de programas de computadora se podían nombrar y que se podían describir sus propiedades y comportamientos. La programación orientada a objetos se puede describir rápidamente como identificación de objetos importantes, su organización en jerarquías, la adición de atributos a los objetos que describen características relevantes en el contexto del problema y de adición de las funciones (métodos) a los objetos para realizar las tareas necesarias en el objeto. Los detalles son algo más complejo, pero lo fundamental es que se trata de un proceso simple y natural.

No obstante, que sea un proceso simple y natural no significa que sea sencillo, ya que un conjunto de objetos se podría clasificar de muchas formas distintas. La clave es la posibilidad de identificar los atributos importantes de objetos y formar abstracciones y jerarquías idóneas. Incluso en el contexto de un dominio problemático, a veces resulta bastante difícil determinar los niveles correctos de abstracción y jerarquías de clasificación correcta. Simplemente, la decisión de la clase o grupo al que un objeto pertenece puede ser una tarea bastante difícil.

2.2.2 Características de los lenguajes de programación orientada a objetos

Los lenguajes de programación orientada a objetos (como C++, C# y Java) se caracterizan por cuatro conceptos claves: encapsulación, herencia, polimorfismo y abstracción, que son compatibles con este aspecto natural de identificación y clasificación de objetos.

2.2.2.1 Encapsulación

La encapsulación facilita la comprensión de los grandes programas; la ocultación de datos les hace más eficaces. Los objetos pueden interactuar con otros objetos sólo a través de los atributos y métodos del objeto que se muestran públicamente. Cuantos más atributos y métodos se muestren públicamente, más difícil será modificar la clase sin que ello afecte al código que utiliza la clase. Una variable oculta se podría cambiar de un long a un double, sin que ello afecte al código que utilicen los objetos creados (instanciados) de esa clase. El programador sólo se debería preocupar por los métodos en las clase que han tenido acceso a esa clase, en lugar de por

todos los sitios en el programa que un objeto ha instanciados desde los que se puede llamar a la clase.

2.2.2.2 Herencia

La herencia proporciona dos ventajas evidentes a los programas. La primera, y más importante, es que permite crear jerarquías que expresen las relaciones entre los diferentes tipos.

La segunda ventaja es que las clases pueden heredar características más generales de las clases superiores dentro de la jerarquía. En lugar de desarrollar nuevas clases a partir de cero, las clases nuevas podrán heredar las funciones de las clases existentes y, a continuación, modificar o ampliar esta función. La clase principal desde la que la nueva clase hereda las propiedades se conoce como la clase básica y la nueva clase se denomina la clase derivada.

2.2.2.3 Polimorfismo

Es una palabra que significa múltiples formas y es una de las características más importantes de la programación orientada a objetos. En realidad, polimorfismo es la propiedad por la que a la misma palabra se le asignan múltiples definiciones. En esencia, polimorfismo es la capacidad para enviar el mismo mensaje a objetos totalmente diferentes, cada uno de los cuales responde a ese mensaje de un modo específico.

2.2.2.4 Abstracción

Una de las consideraciones más importantes para tratar el problema de la complejidad del software es el concepto de abstracción. La idea básica de la abstracción es reducir el nivel de primitivas o representaciones básicas necesarias para producir un sistema de software. De manera sencilla esto se logra mediante el uso de lenguajes de programación que contengan estructuras de datos de alto nivel. En otras palabras, la pregunta opuesta sería: ¿por qué no programar en código binario, o sea 0s y 1s? La respuesta es que ninguna persona sería capaz de comprender una aplicación al verse el código y por otro lado requeriría de programas extremadamente extensos para representar la aplicación completa dada la simplicidad de la primitiva básica. Los sistemas de software construidos con lenguajes de programación de más alto nivel reducen el número total de líneas de código por lo tanto reducen su complejidad. Con la programación orientada a objetos se definen dos niveles de abstracción. El nivel más alto, el de los objetos, es utilizado para describir la aplicación mientras que el nivel más bajo, el de los datos y las funciones, es utilizado para describir sus detalles.

Este nivel inferior corresponde al único nivel de la programación tradicional. Esto refleja que la complejidad se maneja de mejor manera con la tecnología orientada a objetos. En general cuanto más podamos simplificar las tareas de desarrollo mejor será el manejo de la complejidad. Por otro lado el objeto como estructura básica sirve para separar el “qué” de una aplicación del “cómo”, o sea sus detalles, al contrario de la programación tradicional donde el “qué” y el “cómo” se resuelven a la vez.

2.2.3 Clases y Objetos

Las clases son estructuras o plantillas que sirven para definir un objeto. Una clase de un objeto contiene una colección de métodos y definiciones de datos. Si se diseña una clase que representa a un cliente no se ha creado un objeto. Una clase describe la constitución de un objeto y sirve como plantilla para construir objetos, especificando la interfaz pública de un objeto. Una clase tiene un nombre y especifica los miembros que pertenecen a la clase, que pueden ser campos (datos) y métodos (procedimientos). Un objeto es una colección de datos y las subrutinas o métodos que operan sobre ellas, los objetos representan cosas físicas o abstractas, pero que tienen un estado y un comportamiento. Así ciertas propiedades definen a un objeto, y ciertas propiedades definen lo que hacen. Un objeto es una instancia (ejemplar, caso) de una clase y, por consiguiente, puede, naturalmente, haber muchos objetos de una clase.

2.2.4 Proceso Unificado Racional (RUP)

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y, junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

El proceso unificado de modelado se caracteriza por:

2.2.4.1 Centrado en los Modelos

Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.

2.2.4.2 Guiado por los Casos de Uso

Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.

2.2.4.3 Centrado en la Arquitectura

Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.

2.2.4.4 Iterativo e Incremental

Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

2.2.5 Ciclo de vida RUP

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una línea base de la arquitectura.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requerimientos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

2.2.5.1 Principales características de RUP

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

2.2.6 UML

Por sus siglas en inglés (Unified Modeling Language), es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

2.2.7 WebML

WebML (Web Modeling Language) es un lenguaje conceptual para el diseño en alto nivel de aplicaciones que manejan grandes volúmenes de información. Fue definido en 1998 y ha sido implementado y adoptado por muchas universidades a nivel mundial desde su creación.

WebML provee especificaciones gráficas, aun formales, empotradas en un proceso de diseño completo, el cual puede ser asistido por herramientas de diseño visual. Los objetivos principales del diseño WebML son:

- Expresar la estructura de una aplicación Web con un alto nivel de descripción, el cual puede ser usado para consultar, evolucionar y mantener.
- Proveer múltiples vistas del mismo contenido.

- Separar el contenido de la información de su composición en páginas, navegación, y presentación, la cual puede ser definida y evolucionada independientemente.
- Almacenar la meta información colectada durante el proceso de diseño dentro de un repositorio, el cual puede ser usado durante el tiempo de vida de la aplicación para generar dinámicamente páginas Web.
- Modelar usuarios y comunidades explícitamente en el repositorio, para permitir la especificación de políticas de personalización y aplicaciones uno a uno.
- Habilitar la especificación de operaciones de manipulación de datos para actualizar el contenido de la página o interactuar con servicios externos arbitrarios.

2.2.8 Base de Datos

De forma sencilla puede indicar que una base de datos no es más que un conjunto de información relacionada que se encuentra agrupada o estructurada.

2.2.8.1 Elementos de almacenamiento de una Base de Datos

- a) **Campo:** Es la unidad básica de una base de datos.
- b) **Registro:** Es el conjunto de información referida a una misma persona u objeto.

2.2.8.2 Tipos de Base de Datos

Las bases de datos se pueden dividir en cuatro tipos básicos:

- Bases de datos de fichero plano (o ficheros por bloques).
- Bases de datos relacionales.
- Bases de datos orientadas a objetos.
- Bases de datos híbridas.

a) Base de datos de fichero: Las bases de datos de fichero plano consisten en ficheros de texto divididos en filas y columnas. Estas bases de datos son las más primitivas y quizás ni tan siquiera merezcan considerarse como tales. Pueden ser útiles para aplicaciones muy simples, pero no para aplicaciones medianas o complejas, debido a sus grandes limitaciones.

b) Base de datos relacional: Las bases de datos relacionales son las más populares actualmente. Su nombre proviene de su gran ventaja sobre las bases de datos de fichero plano: la posibilidad de relacionar varias tablas de datos entre sí, compartiendo información y evitando la duplicidad y los problemas que ello conlleva (espacio de almacenamiento y redundancia).

Existen numerosas bases de datos relacionales para distintas plataformas (Access, Paradox, Oracle, Sybase,) y son ampliamente utilizadas. Sin embargo, tienen un punto débil: la mayoría de ellas no admite la incorporación de objetos multimedia tales como sonidos, imágenes o animaciones.

2.2.9 C#

Es un lenguaje de programación (pronunciado si sharp en inglés) simple, moderno, orientado a objetos y fuertemente tipificado. C# tiene sus raíces en la familia de lenguajes C y es familiar inmediatamente con programadores de C, C++ y Java. C# esta estandarizado por ECMA en el estándar ECMA-334 y por ISO/IEC en el estándar ISO/IEC 23270, El compilador de Microsoft de C# para .NET Framework es una implementación conformada por ambos estándares.

2.2.10 HTML

Siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

2.2.11 PHP

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools), es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting).

2.2.12 CSS

Las hojas de estilo en cascada (en inglés Cascading Style Sheets), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

2.2.13 Debían GNU/Linux

Debían es un sistema operativo (S.O.) libre, para su computadora. El sistema operativo es el conjunto de programas básicos y utilidades que hacen que funcione su computadora. Debían utiliza el núcleo Linux (el corazón del sistema operativo), pero la mayor parte de las herramientas básicas vienen del Proyecto GNU; de ahí el nombre GNU/Linux.

2.2.14 PostgreSQL

PostgreSQL es un poderoso sistema de bases de datos de código abierto y orientado a objetos. Tiene más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una fuerte reputación de robustez, integridad de datos y fiabilidad. Corre en la mayoría de los sistemas operativos.

CAPÍTULO III: FASE DE INICIO

3.1 Introducción

La fase inicial ó de conceptualización del proceso unificado permite definir el ámbito del sistema para justificar la factibilidad del proyecto. En este capítulo se plantea un modelo conceptual de la arquitectura de acuerdo a las especificaciones funcionales que se obtengan de la identificación de los requisitos exigidos por el cliente. Igualmente, se efectúa un análisis y definición de riesgos mediante el uso de las herramientas de UML (modelo de dominio, casos de uso, actividades, clases de diseño y paquete).

En la fase de conceptualización figura 3.1 se desarrollan iteraciones preliminares del flujo de trabajo normal. La intención es recopilar la mayor cantidad de requisitos.



Figura 3.1 Diagrama de flujo de trabajo de las fases del proceso unificado, con identificación de la fase conceptualización (inicio).

3.2 Planificación de la fase de inicio

El desarrollo de la fase de inicio se plantea de manera progresiva, parte con un listado de los requisitos y una breve descripción de estos, sigue con la definición del modelo de dominio en donde se describían las principales clases que interactúan en el sistema, el desarrollo de los casos de usos para diagramar una estructura inicial según los requisitos listados. Una vez elaborados estos diagramas, se refinan y estructuran las funcionalidades a través de los diagramas de actividades, se construirán los diagramas preliminares de clases y paquetes para encapsular los elementos que se van definiendo durante la realización del análisis del sistema.

3.3 Requisitos

La definición de los requisitos del sistema representa uno de los pasos más importantes en el cumplimiento de los objetivos del proyecto y en la satisfacción de las necesidades del cliente. Bien cita Fred Brooks:

“La parte más dura en la construcción de un sistema de software es decidir cómo construirlo... Ninguna otra parte del trabajo mutila el resultado del sistema si está hecho mal. Ninguna otra parte es más difícil para rectificarlo después.”. A través de los requisitos se comenzará a definir las funcionalidades que el sistema deberá tener.

3.3.1 Lista de requisitos o características

La lista presentada a continuación es producto de la recopilación de información necesaria para el entendimiento del problema y los procesos que

se quieren automatizar, a manera de generar un sistema adecuado. Según lo anterior el sistema debe comprender y permitir los siguientes requerimientos generales:

- Establecer los valores predeterminados y parámetros con que iniciará la ejecución formal de un website.
- Funcionar como una aplicación, desarrollada en el lenguaje de programación C# y utilizando el manejador de base de datos PostgreSQL 8.4
- Funcionar como una aplicación web, desarrollada con la tecnología PHP utilizando igualmente el manejador de base de datos PostgreSQL.
- Incluir componentes codificados que encapsulen las siguientes funcionalidades comunes del website.
- Desarrollar las clases, controles y componentes de forma totalmente compatible con la plataforma Mono 2.6.1.
- Generar automáticamente toda la estructura tabular requerida en la base de datos de acuerdo a los parámetros configurados.
- Agregar o eliminar funcionalidades a un website puesto en producción, garantizando la mantenencia de los datos existentes.

- Implementar opciones de seguridad que minimicen los riesgos de ataques o mal uso del sistema en general.
- Incluir la opción para permitir a los internautas registrarse a través del sitio web para el caso de solicitar algún servicio.
- Incluir la opción para permitir a los usuarios del registro (usuario común) registrarse en la aplicación para el caso de solicitar algún documento.
- Implementar el componente de contenido general de forma que permita la visualización de los documentos generados por el escribiente ó secretaria mediante un asistente.
- Desarrollar un buzón de mensajería que servirá como base para el envío y recepción de documentos generados por el personal del registro civil o solicitudes a través del portal web por el internauta.
- Desarrollar servicios web que permitan la distribución e interoperabilidad de la información de carácter público que se incluya en el sitio web.

3.3.2 Identificación de los riesgos

La identificación de los riesgos es un intento sistemático para especificar las amenazas al plan del proyecto. Identificando los riesgos conocidos y predecibles, se da un paso adelante para evitarlos cuando sea posible y controlarlos cuando sea necesario.

Se pueden clasificar dos tipos diferenciados de riesgos: genéricos y específicos, los riesgos genéricos son una amenaza potencial para todos los proyectos de software, los específicos se pueden identificar como los que tienen una clara visión de la tecnología, el personal y el entorno específico del proyecto en cuestión. Tanto los riesgos genéricos como los específicos se deberían identificar sistemáticamente, un método para identificar riesgos es crear una lista de comprobación de elementos de riesgos, la lista de comprobación se enfoca en un subconjunto de riesgos conocidos y predecibles, que pueden ser organizados de diferentes maneras, un formato de listas de comprobación de elementos de riesgos simplemente son las características relevantes para cada sub-categoría genérica. A continuación se definen cuales de estos elementos genéricos podrían afectar la ejecución del proyecto organizados de forma categorizada.

Tabla 3.1 Lista de riesgos y contingencia

Riesgo	Contingencia
Errores en la manipulación de la data	Verificar las secuencias de código encargadas de la gestión de la data.
Incapacidad de uso del sistema por parte de los usuarios finales	Comprobar el correcto diseño de las interfaces gráficas y del manual de usuario.
Escogencia de herramientas inadecuadas para el desarrollo del sistema	Indagar sobre el alcance de estas herramientas y sus posibles alternativas.
Tamaño del producto	El software a desarrollar puede considerarse complejo en la medida que, para cumplir con los requerimientos del registro civil, se necesita detallar muchas características y comportamientos, para que estos no afecten en su implementación la originalidad y calidad de los diseño de la empresa.
Impacto en el negocio	Verificar lo asociado con las limitaciones impuestas por la gestión o por el mercado.

3.4 Modelo de dominio

Para capturar los tipos más importantes de objetos en el contexto del sistema se presenta el modelo de dominio. Los objetos o clases del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabajará el sistema.

Su objetivo es contribuir a la comprensión del contexto del sistema, y por extensión de los requisitos que se desprenden de este contexto. Es decir, contribuye a una comprensión del problema que el sistema resolverá en relación a su contexto. El modo interno por el cual el sistema resolverá este problema se tratará en los siguientes flujos de trabajo, figura 3.2. Las clases que se presentan en el modelo de dominio son las siguientes:

Registro civil: representa la dirección, el ente encargado de registrar, generar y entregar documentos.

Website: conjunto de páginas, imágenes y otros formatos que pueden ser accedidos y visualizados por internautas o clientes vía internet. Se considera dinámico cuando ofrece opciones para que el cliente solicite de forma frecuente la información ofrecida por él.

Registrador(a) Civil: Los registradores civiles tienen la facultad de conferir fe pública a todas las actuaciones, declaraciones y certificaciones, que con tal carácter autoricen, confiriéndole eficacia y pleno valor probatorio.

Escribiente/Secretaria(o): Son funcionarios que tienen la facultad de atender alguna solicitud y entregar las certificaciones de los documentos correspondientes a cada solicitud.

Documentos: Representa el testimonio material del hecho o acto realizado en el Registro civil, registrado en una unidad de información.

Usuario ó Ciudadano Común: personas que requieren información, oportuna y veraz en relación con los trámites que se realicen ante el Registro Civil.

Internauta: el usuario común de internet que podrá visualizar el website.

Página: cada pantalla de un website que organiza y visualiza los gráficos, texto y formularios con los que interactuarán los internautas.

Desarrollador Web: personal capacitado para realizar el diseño gráfico de todas las páginas y componentes del website.

Administrador/programador: personal encargado de mantener la integridad de la información, procurando garantizar la seguridad física y jurídica, así como la confiabilidad e inalterabilidad de sus datos.

Componente: referido como un control, representa cada bloque de elementos presentes en una página del sitio Web como en el software de aplicación con los que podrán interactuar los internautas ó usuarios del sistema en base a alguna función específica.

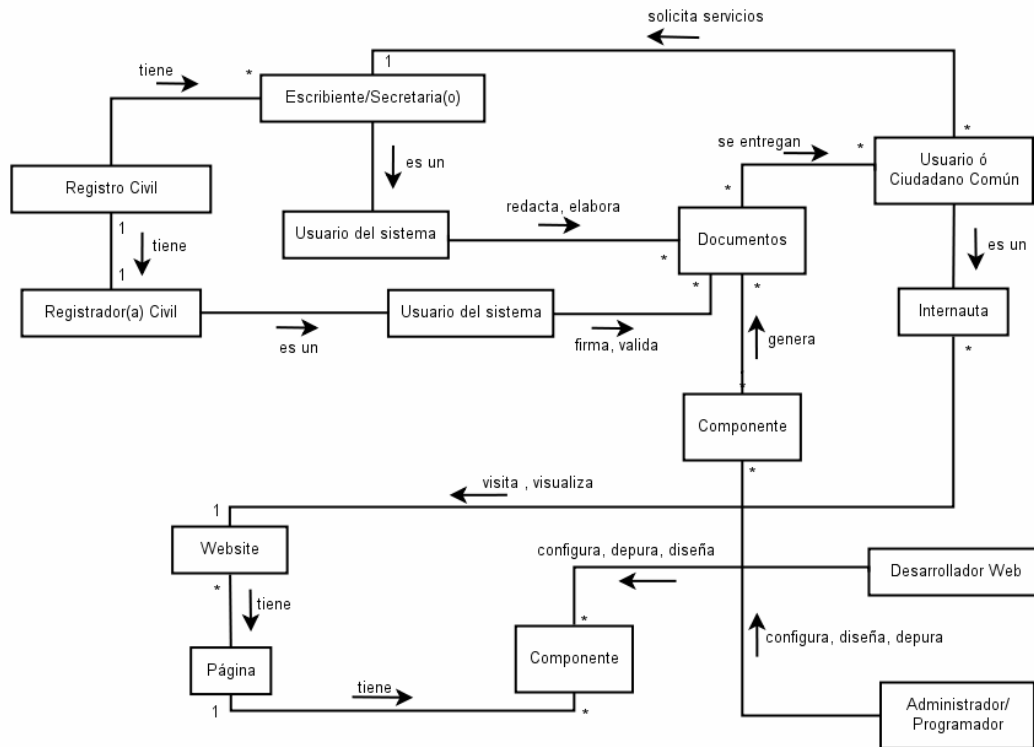


Figura 3.2 Modelo de dominio del sistema

3.5 Modelo de casos de usos

En esta sección se identifica los entes que interactúan y están involucrados con los procesos y los casos de uso necesarios para delimitar el sistema, el alcance del proyecto y detallar los más importantes.

3.5.1 Identificación de casos de usos

De acuerdo al contexto y requisitos del sistema, se puede modelar una solución haciendo uso de dos subsistemas: uno accesible por todos los internautas, y otra, una aplicación de software con el que solo interactúan los usuarios del registro civil, ambos englobarán la totalidad de un mismo

proyecto, recordando que las herramientas que brindará este proyecto servirá para implementar la variedad y así poder brindar toda la información necesaria para un registro civil.

En la figura 3.3 y 3.4 muestran los diagramas de casos de uso del sistema en general, dentro de las tablas 3.2 al 3.20 describen cada uno de ellos.

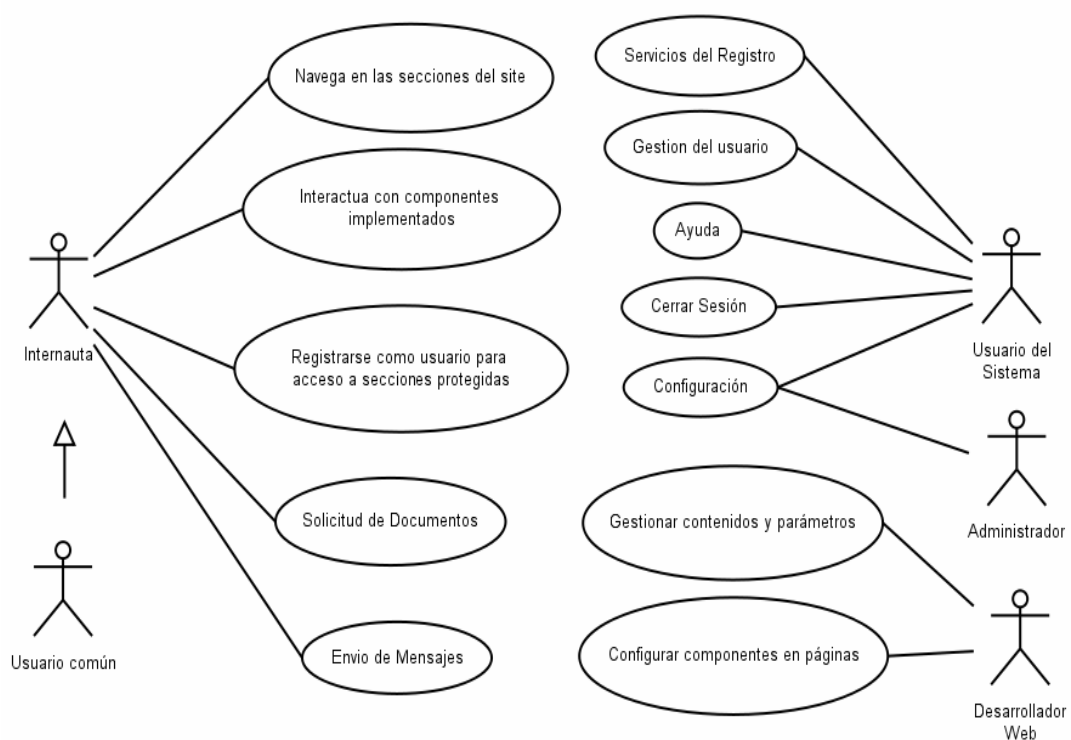


Figura 3.3 Diagrama de casos de uso

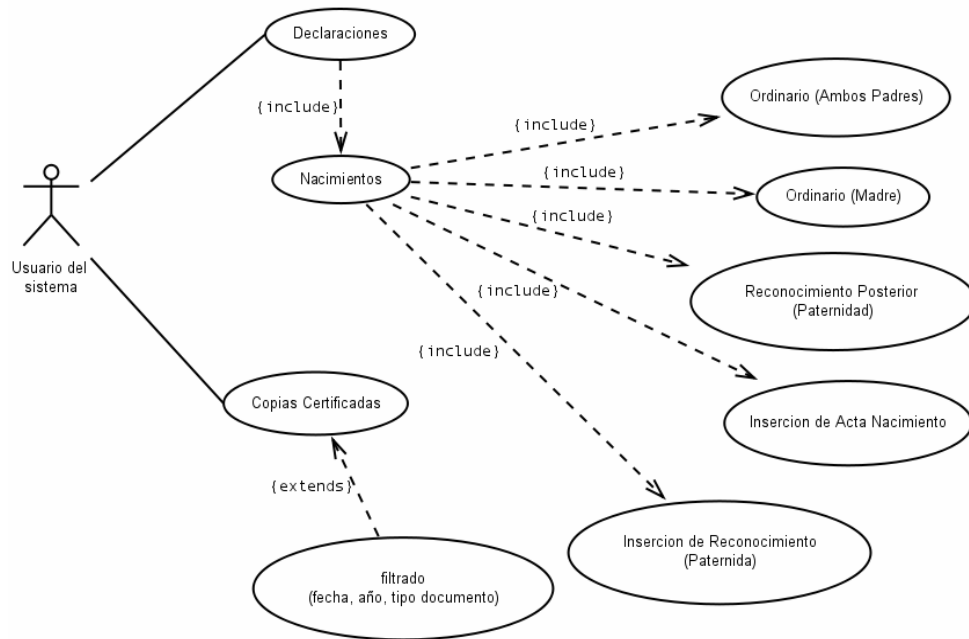


Figura 3.4 Diagrama de casos de uso

Tabla 3.2 Casos de uso Navegar en las secciones del site administradas por el sistema

ACTOR PRINCIPAL:	Internauta
ALCANCE:	Área pública
NIVEL:	Sub-función
DESCRIPCIÓN:	Cualquier internauta, al ingresar la dirección del sitio web publicado podrá acceder al diseño web realizado, mediante menús y vínculos como herramientas para navegar entre todas las páginas y secciones públicas del website. De igual forma, se muestran las interfaces de todos los componentes implementados, de acuerdo a sus configuraciones.

Tabla 3.3 Casos de uso Interactuar con componentes implementados

ACTOR PRINCIPAL:	Internauta
ALCANCE:	Área pública
NIVEL:	Sub-función
DESCRIPCIÓN:	De acuerdo a las características de cada componente implementado, se le permitirá al usuario ciertas interacciones y funciones específicas.

Tabla 3.4 Casos de uso Registrarse como usuario para acceso a secciones protegidas

ACTOR PRINCIPAL:	Internauta
ALCANCE:	Área pública
NIVEL:	Sub-función
DESCRIPCIÓN:	<p>Ciertos websites podrían ofrecer el registro libre de usuarios para dar acceso a funcionalidades que requieran de una identificación y/o seguimiento como usuario.</p> <p>De igual manera, aunque otros website no ofrezcan el registro de forma libre para los usuarios, pueden proteger secciones de forma que solo sean accesibles por los internautas a los que el cliente haya facilitado credenciales válidas.</p>

Tabla 3.5 Casos de uso Solicitud de documentos

ACTOR PRINCIPAL:	Internauta
ALCANCE:	Área privada
NIVEL:	Sub-función
DESCRIPCIÓN:	El internauta ingresa los datos claves para buscar el documento a solicitar

Tabla 3.6 Casos de uso Envío de mensajes

ACTOR PRINCIPAL:	Internauta
ALCANCE:	Área privada
NIVEL:	Sub-función
DESCRIPCIÓN:	El internauta selecciona un funcionario que se encuentre conectado al sistema para escribir y enviar un mensaje.

Tabla 3.7 Casos de uso Gestionar contenidos y parámetros

ACTOR PRINCIPAL:	Programador Web
ALCANCE:	Área privada
NIVEL:	Sub-función
DESCRIPCIÓN:	El programador administrar los distintos componentes de contenido (por ejemplo, noticias, galerías y documentos) así como establecer parámetros de los componentes estructurales.

Tabla 3.8 Casos de uso Configurar componentes en páginas

ACTOR PRINCIPAL:	Programador Web
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El programador podrá configurar la forma en que se ensamblen los componentes en las páginas de un sitio, estableciendo los parámetros para que la implementación este bien adaptada al diseño e ideas originales para dicho website.

Tabla 3.9 Casos de uso Servicios del Registro

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema podrá acceder a las distintas actividades que realiza un funcionario en un registro civil dentro de las actividades programadas en el software tenemos declaraciones y copias certificadas.

Tabla 3.10 Casos de uso Gestión del usuario

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema podrá acceder a las distintas actividades relacionadas con su perfil como usuario del sistema como por ejemplo cambio de contraseña, cambio de imagen, esquema de color del software, redacción de actas, buzón de mensajería.

Tabla 3.11 Casos de uso Configuración

ACTOR PRINCIPAL:	Usuario del sistema ó Administrador
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El administrador como el usuario del sistema podrá acceder a las distintas actividades relacionadas con su perfil y nivel de usuario del sistema como por ejemplo restaurar valores por defecto, impresora por defecto, entre otras actividades.

Tabla 3.12 Casos de uso Ayuda

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema podrá acceder al manual de instrucciones de uso del software.

Tabla 3.13 Casos de uso Cerrar sesión

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema podrá salir del software de aplicación.

Tabla 3.14 Casos de uso Declaraciones

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema accede al componente nacimientos para elaborar los distintos tipos de documentos existentes en este segmento.

Tabla 3.15 Casos de uso Copias certificadas

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema accede al componente que genera una interfaz para la búsqueda de documentos existentes en la base de datos.

Tabla 3.16 Casos de uso Nacimientos

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema accede al componente nacimientos allí podrá escoger el tipo de documento a elaborar.

Tabla 3.17 Casos de uso Ordinario (Ambos Padres)

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema genera un asistente para la elaboración de una partida de nacimiento en donde se ve reflejado la presencia de ambos padres y los testigos en el acta de nacimiento.

Tabla 3.18 Casos de uso Ordinario (Madre)

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema genera un asistente para la elaboración de una partida de nacimiento en donde se ve reflejado la presencia sólo de la madre y los testigos en el acta de nacimiento.

Tabla 3.19 Casos de uso Reconocimiento Posterior (Paternidad)

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema genera un asistente para la elaboración de una partida de nacimiento en donde se ve reflejado el reconocimiento que hace el padre a un niño en un acta de nacimiento ya existente en la base de datos del software.

Tabla 3.20 Casos de uso Inserción de Acta de Nacimiento

ACTOR PRINCIPAL:	Usuario del sistema
ALCANCE:	Área privada
NIVEL:	Resumen
DESCRIPCIÓN:	El usuario del sistema genera un asistente para la elaboración de una partida de nacimiento en donde se ve reflejado la inserción que se realiza en los libros de nacimientos del registro civil.

3.6 Diagrama de actividad

Comprende las actividades del website público y software de aplicación. Las señales son emitidas por los internautas que accedan a las páginas, así como también a los usuarios del sistema que acceden a los componentes del software de aplicación.

3.6.1 Navegar en las secciones del site

Esta actividad se ejecuta cada vez que un internauta intenta acceder a una página del website. En la figura 3.5 se visualiza todos los contenidos y componentes de la página, quedando abierta la posibilidad de que el usuario accione al seleccionar un vínculo o interactúe con algún control o componente.

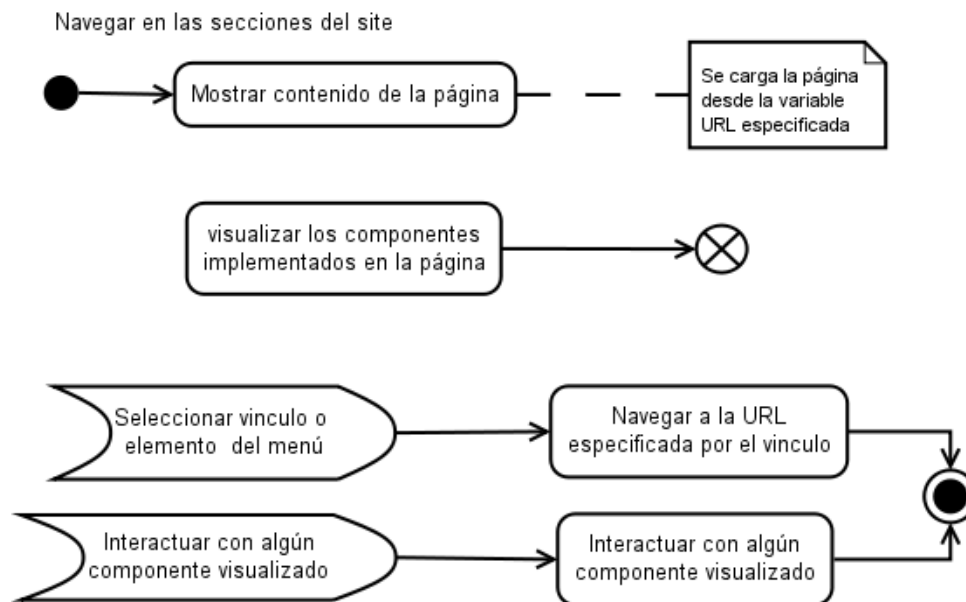


Figura 3.5 Diagrama de actividad Navegar a una sección del website

3.6.2 Registrarse como usuario para acceso a secciones protegidas

Esta actividad figura 3.6 se ejecuta cuando un internauta decide registrarse para acceder a secciones protegidas del website, introduciendo sus datos y siendo verificada la unicidad de su identificación.

3.6.3 Interactuar con componentes implementados

El usuario puede accionar cualquier elemento de dichos controles figura 3.7, no siendo restrictiva el orden en que se quiera interactuar con ellos. La forma en que cada componente procese y reacciones las señales del usuario serán detalladas en el próximo capítulo.

Registrarse como usuario para acceso a secciones protegidas

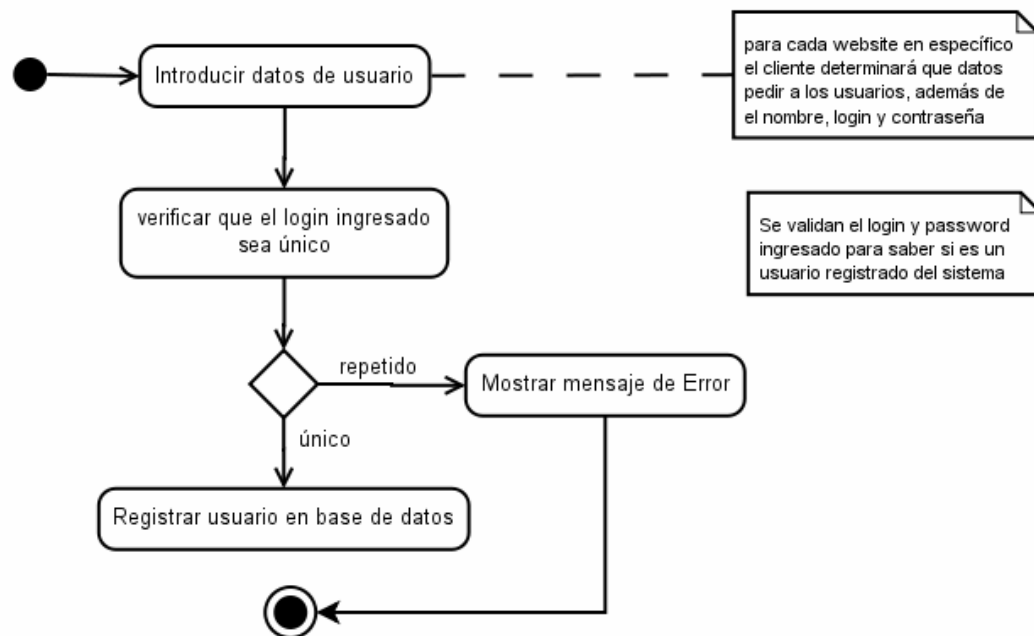


Figura 3.6 Diagrama de actividad registrarse como usuario

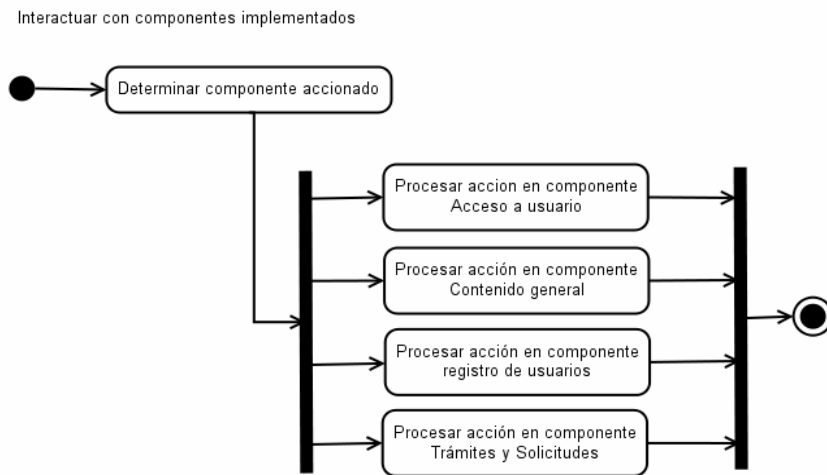


Figura 3.7 Diagrama de actividad Interactuar con componentes implementados

3.6.4 Gestionar contenidos y parámetros

Al ser seleccionado algún elemento del menú de accesos permitidos para el usuario, se visualiza la aplicación de configuración que corresponda figura 3.8. Cada una de estas aplicaciones será detallada en el próximo capítulo.

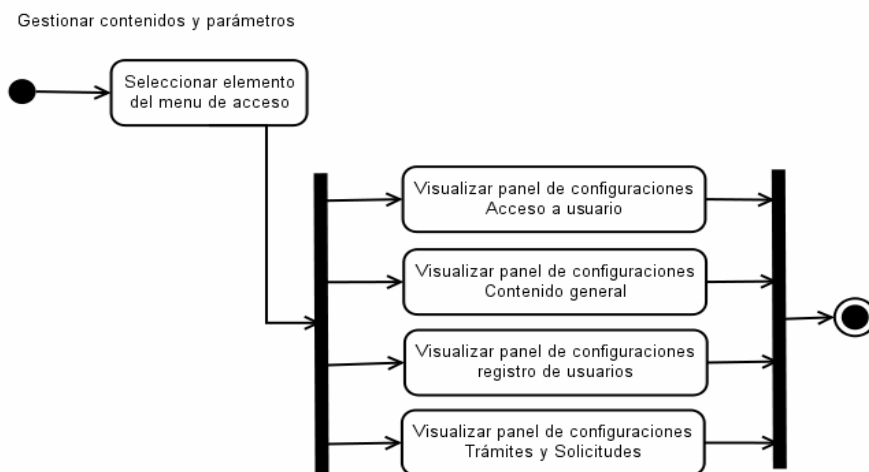


Figura 3.8 Diagrama de actividad Gestionar contenidos y parámetros

3.6.5 Configurar componentes de páginas

Esta actividad figura 3.9 correspondiente a la etapa de desarrollo del website, siendo el programador web el único actor que podrá realizarla. El programador deberá abrir el código de cada página que quiera configurar, agregar el marcado de los componentes que quiera implementar y luego podrá ajustar el resto de los parámetros y propiedades que garantizarán la correcta composición entre diseño y funciones.

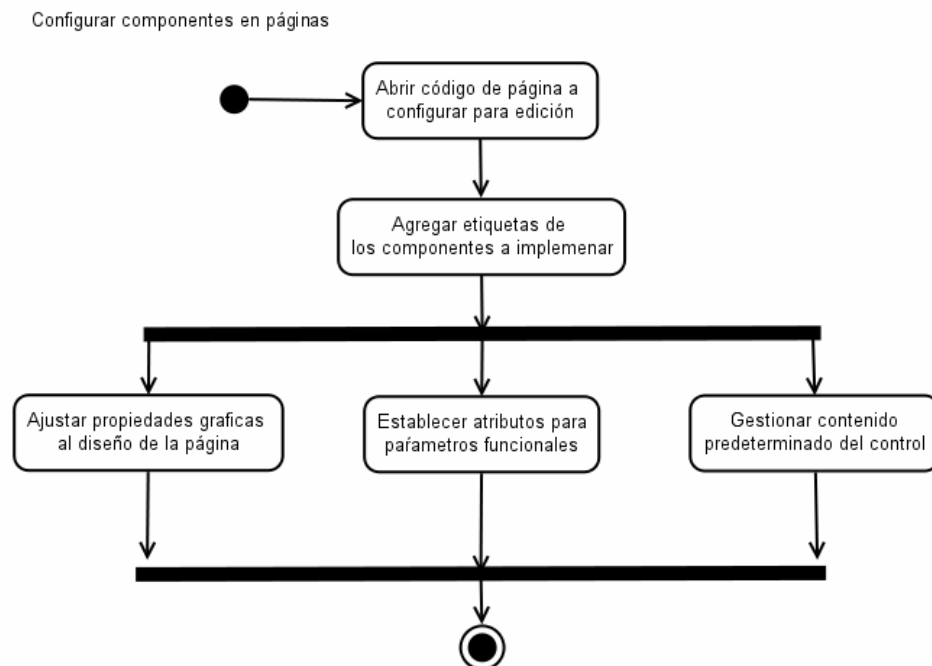


Figura 3.9 Diagrama de actividad configurar componentes en páginas

3.6.6 Solicitud de Documentos

Esta actividad figura 3.10 contempla las actividades que realiza un internauta (loggeado) al requerir un documento dentro del website.

3.6.7 Envío de Mensajes

La actividad figura 3.11 corresponde a las acciones que realiza un internauta (loggeado) para enviar un mensaje a un funcionario que se encuentre activo en el software de aplicación.

3.6.8 Servicios del Registro

Esta actividad figura 3.12 corresponde a las actividades que realiza un usuario del sistema dentro del software de aplicación, el programador podrá realizar e incluir modificaciones, deberá abrir el código del componente que quiera configurar, agregar el marcado del o los componentes que quiera implementar y luego podrá ajustar el resto de los parámetros y propiedades que garantizarán la correcta composición entre diseño y funciones.

3.6.9 Gestión del usuario

La figura 3.13 representa las actividades que realiza un usuario del sistema para la gestión dentro del software de aplicación.

Solicitar Documentos

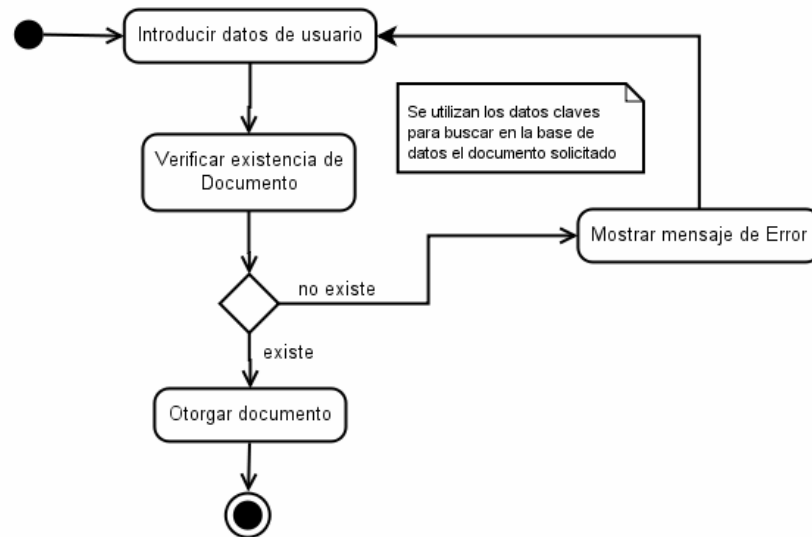


Figura 3.10 Diagrama de actividad Solicitud de documentos

Envío de Mensaje

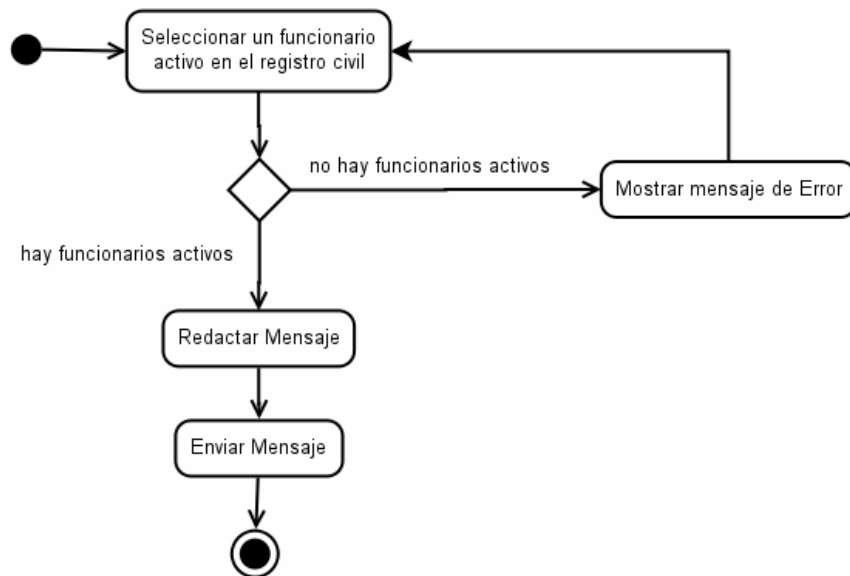


Figura 3.11 Diagrama de actividad gestión del usuario

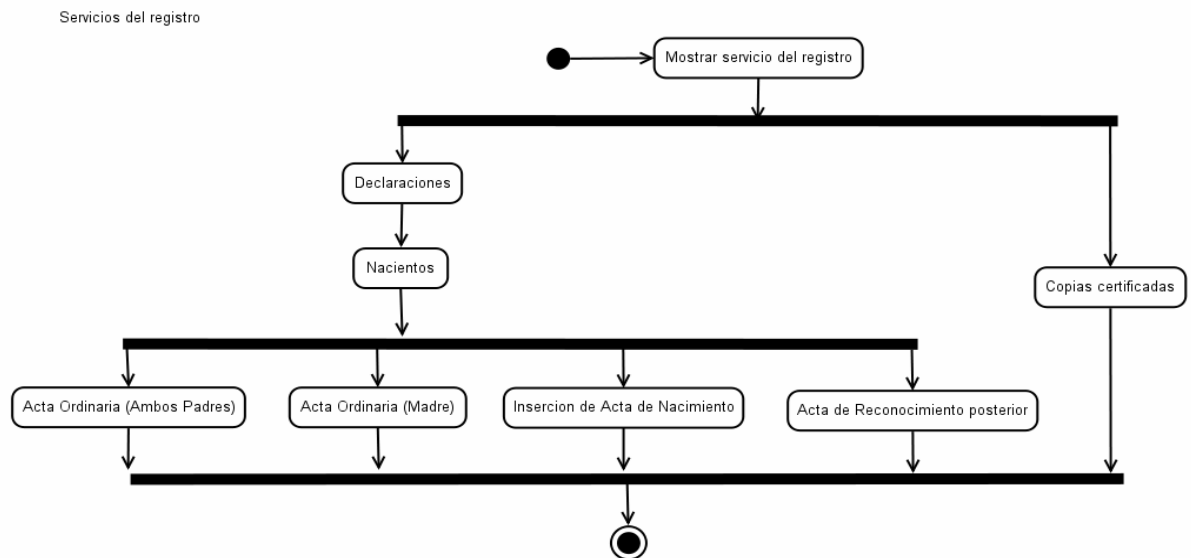


Figura 3.12 Diagrama de actividad servicios del registro

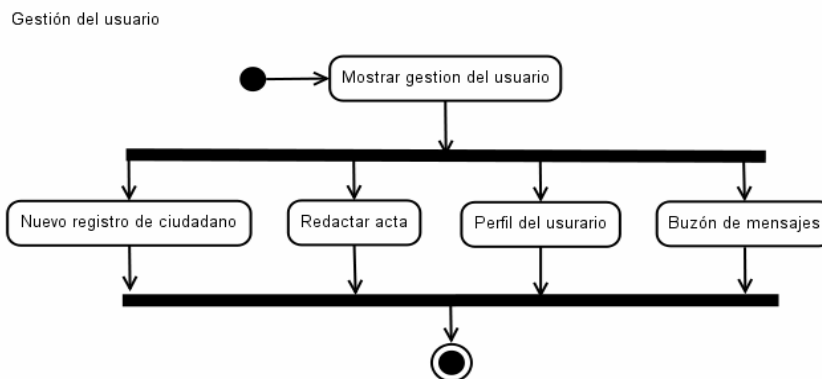


Figura 3.13 Diagrama de actividad gestión del usuario

3.6.10 Configuración

Representa el conjunto de datos que determina el valor de algunas variables del software de aplicación, estas opciones generalmente son definidas definida por el usuario (personalizada), de lo contrario el programa cargará la configuración por defecto.

3.6.11 Ayuda

Representa el conjunto de datos que determina la información del software como el manual de usuario, versión del programa, entre otros.

3.6.12 Cerrar sesión

Representa la salida del software a la pantalla principal de identificación de usuario.

3.7 Diagramas preliminares de clases y paquetes

El modelo realizado en esta fase de inicio con los diagramas de caso de uso y actividades, ofrece una noción global de la forma en que se pueden establecer y organizar las clases para representar los aspectos del sistema en general.

Los diagramas de clases de más alto nivel es lógicamente un dibujo de los paquetes que componen el sistema. A su vez cada paquete tendrá un diagrama que muestra las clases del mismo.

3.7.1 Paquete SRC

El paquete SRC figura 3.14 representa la totalidad del sistema (software de aplicación) a desarrollar y está formado por 3 paquetes.

- En el paquete Consola se encuentra la lógica necesaria para iniciar el programa en modo consola permitiendo así cargar el conjunto de

imágenes a utilizar y establecer los parámetros de conexión con el servidor.

- El paquete Interfaz define las clases que organizarán al software para el entorno gráfico dividido en dos sub paquetes como el UserControl y Formularios.
- El paquete Compuesto contiene las clases necesarias para accesos a la base de datos y clases necesarias para el control de la data del software de aplicación.

3.7.2 El paquete itextsharp

Es una biblioteca *Open Source* para crear y manipular archivos PDF, RTF, y HTML en C#. Fue escrita por Bruno Lowagie, Paulo Soares, y otros; está distribuida bajo la *Mozilla Public License con la LGPL*.

3.7.3 El paquete Mono.Security

Ofrece las piezas que faltan para la plataforma .NET para la seguridad, se utiliza a menudo para proporcionar funcionalidades muy necesarias (como algunos algoritmos criptográficos, firma de código, los certificados X.509).

3.7.4 El Paquete npgsql

E un neto proveedor de datos. Permite a cualquier programa desarrollado para la plataforma .NET el acceso al servidor de base de datos

para PostgreSQL. Se lleva a cabo en el 100% de código en C # y funciona con Postgresql 7.4 y las anteriores.

La figura 3.15 muestra los paquetes itextsharp y Npgsql.

3.7.5 El paquete System

El resto de los elementos que el sistema va a usar son los que ofrece el Proyecto Mono. La plataforma Mono proporciona un completo conjunto de clases que proporcionan una sólida base para construir aplicaciones, véase en la figura 3.16.

- System.Data y System.Data.Dbtype: Agrupa todos los elementos que proveerán los tipos de datos que serán compatibles con Npgsql.
- System.Xml: proporciona compatibilidad basada en estándares para procesar XML.
- System.Drawing 2.0 y System.Windows.Form 2.0: Son todas las entidades que procesan la interfaz gráfica de programación de las aplicaciones (*API*), estas proporcionan un acceso a los elementos nativos de la interfaz envolviendo la actual API de *Windows*, Mono es un proyecto liderado por Novell y su System.Windows.Forms en su versión 2.0 contiene el 100% de las clases y métodos que también se pueden ejecutar de forma nativa en Mac OS X y Linux.

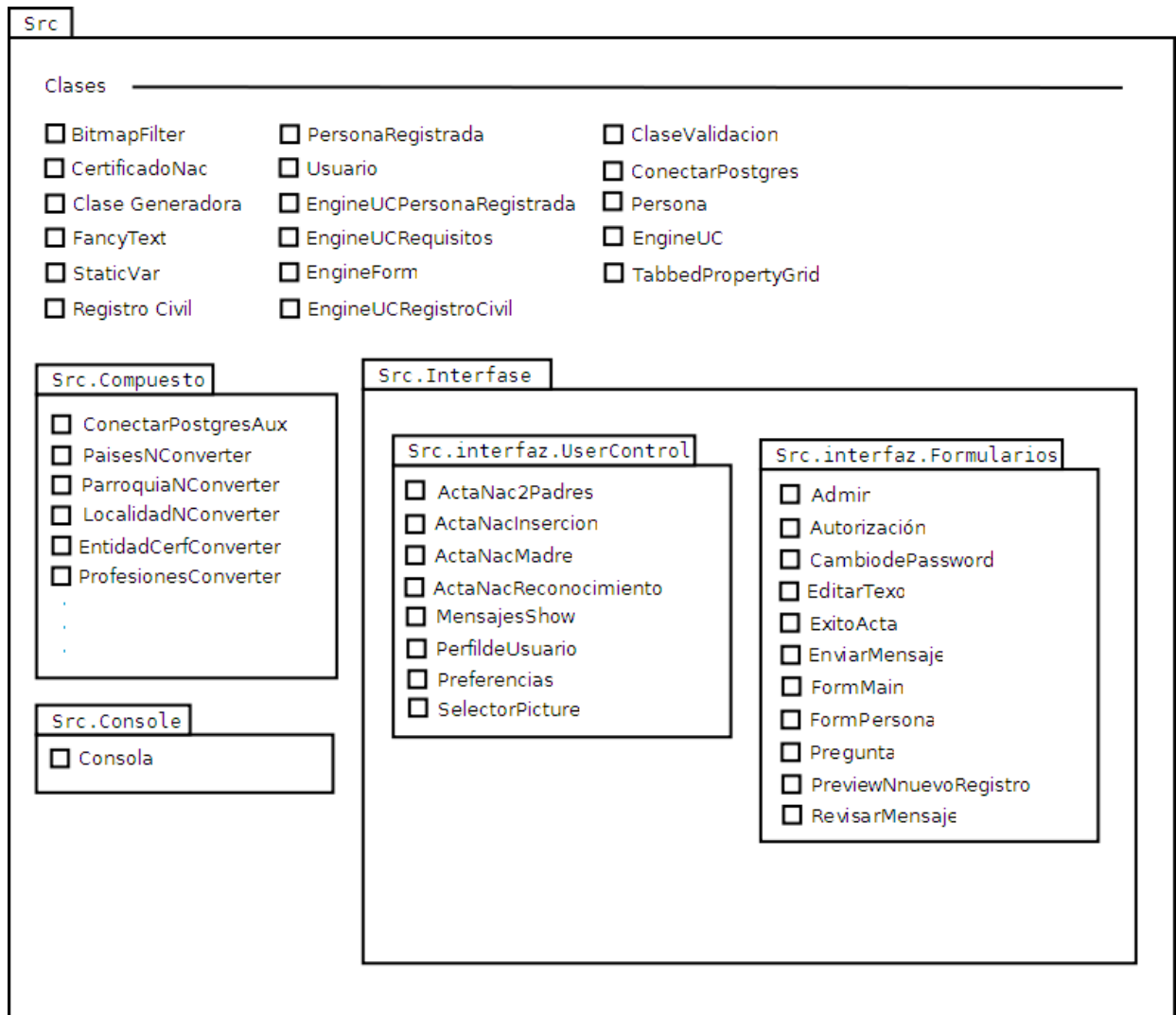


Figura 3.14 Diagrama de paquetes para Src

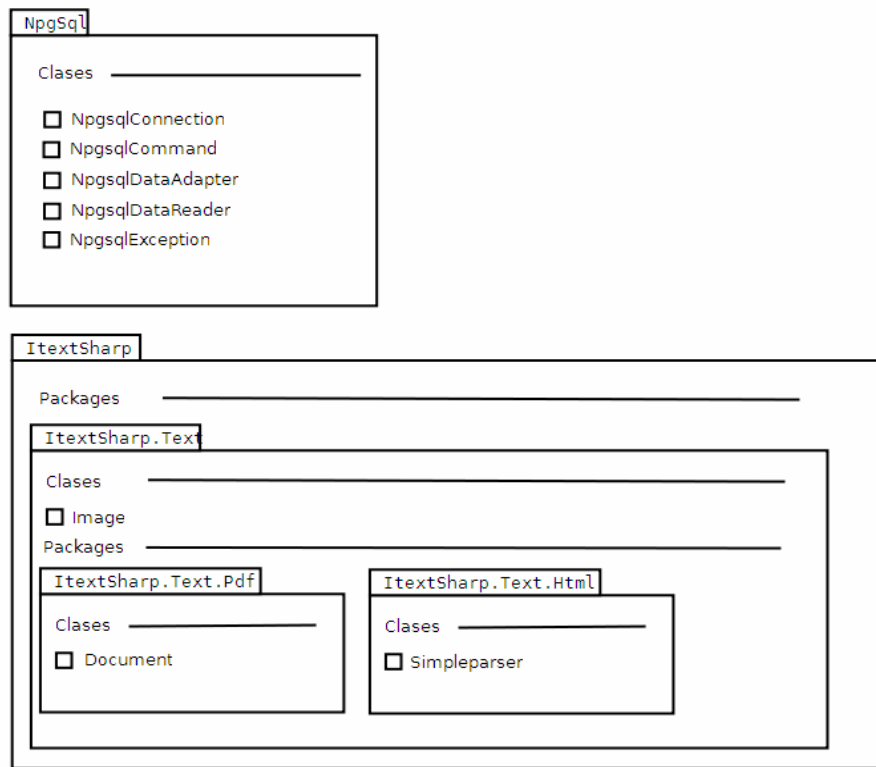


Figura 3.15 Diagrama de paquetes Npgsql y ItextSharp

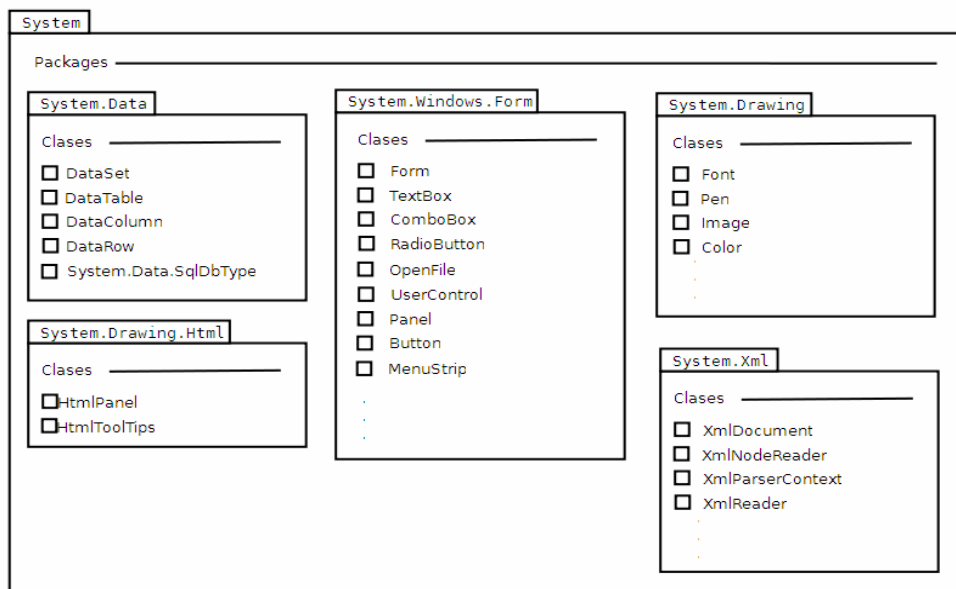


Figura 3.16 Diagrama de paquetes System

3.7.6 Clases del paquete Src

Contendrá principalmente la ClaseGeneradora se encarga de generar los documentos según sea el parámetro enviado, la clase BitmapFilter está encargada de manipular las imágenes del software y procesarlas añadiendo efectos si son requeridos, las clases CertificadoNac, Persona, Usuario, RegistroCivil expresadas en las figuras 3.17 y 3.18, hereda las funciones de validación de la ClaseValidación figura 3.19, para tratar de reducir al mínimo el error humano por tipear datos al sistema.

La clase EngineForm expresada en las figuras 3.20 y 3.21, está encargada de mostrar el formulario en pantalla. Y la clase EngineUC expresada en las figuras 3.22 y 3.23, se encarga de las subventanas en el software, estas clases son necesarias para la ejecución del software en el entorno gráfico.

Las siguientes clases no pertenecen al paquete EngineUC por que varían en su contenido EngineUCPersonaRegistrada, EngineUCRequisitos, EngineUCRegistroCivil, por lo tanto se consideró separarlas del paquete.

La clase ConectarPostgres se considera una de las más importantes de todo el proyecto ya que contiene toda la información para el acceso a la base de datos y funciones especiales necesarias para que el software funcione correctamente.

La figura 3.24 muestra en modo de detalla el diagrama de clases para StaticVar, ItextSharpEvents, BitmapFilter, ClaseGeneradora, EngineToolTips y la clase consola

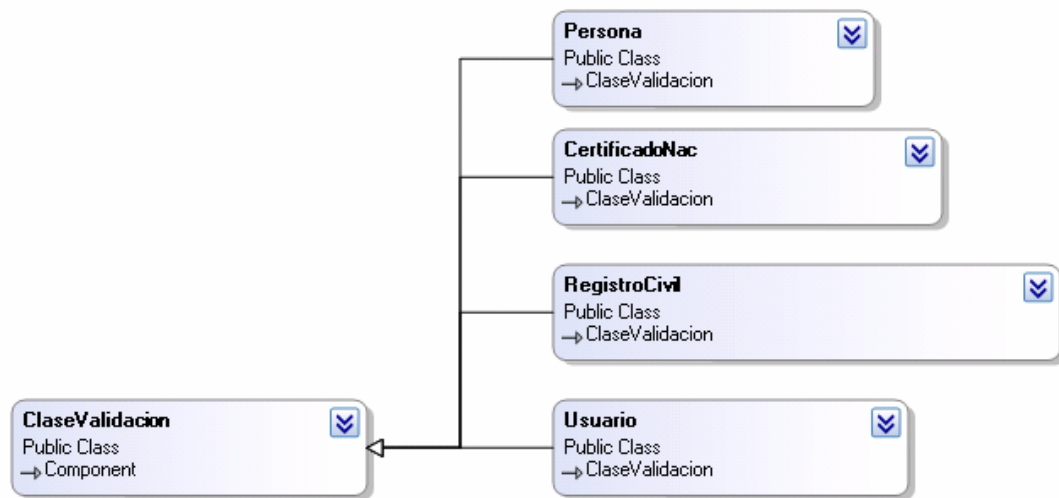


Figura 3.17 Diagrama de clases para ClaseValidacion, Persona, CertificadoNac, RegistroCivil, Usuario (modo compacto)



Figura 3.18 Diagrama de clases para Persona, CertificadoNac, RegistroCivl, Usuario (modo expandido)

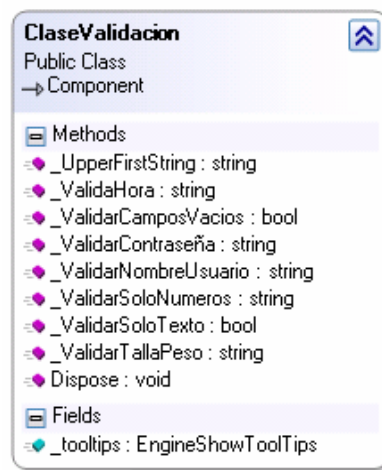


Figura 3.19 Diagrama de clases para ClaseValidacion (modo expandido)

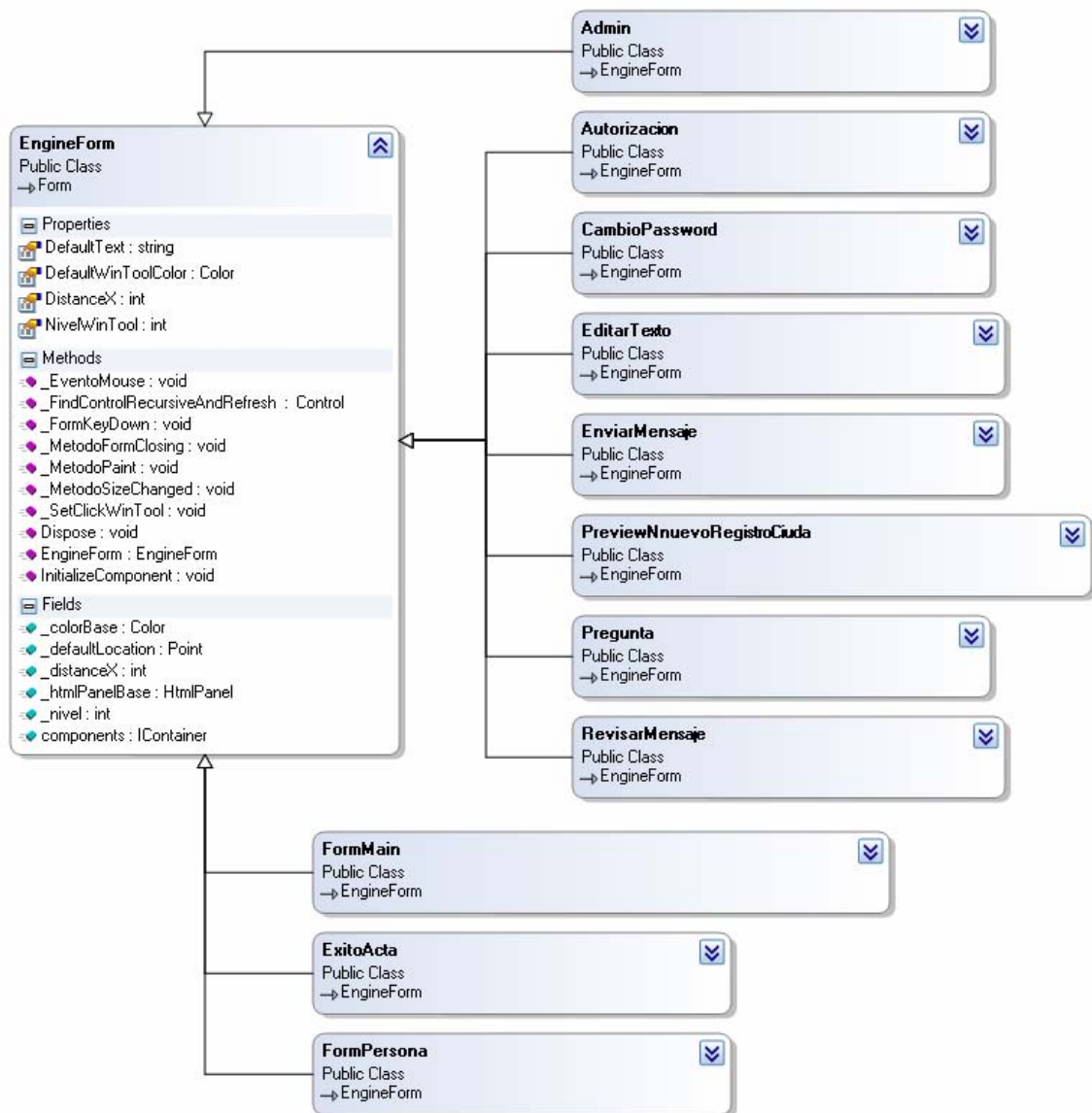


Figura 3.20 Diagrama de clases para EngineForm (modo compacto)

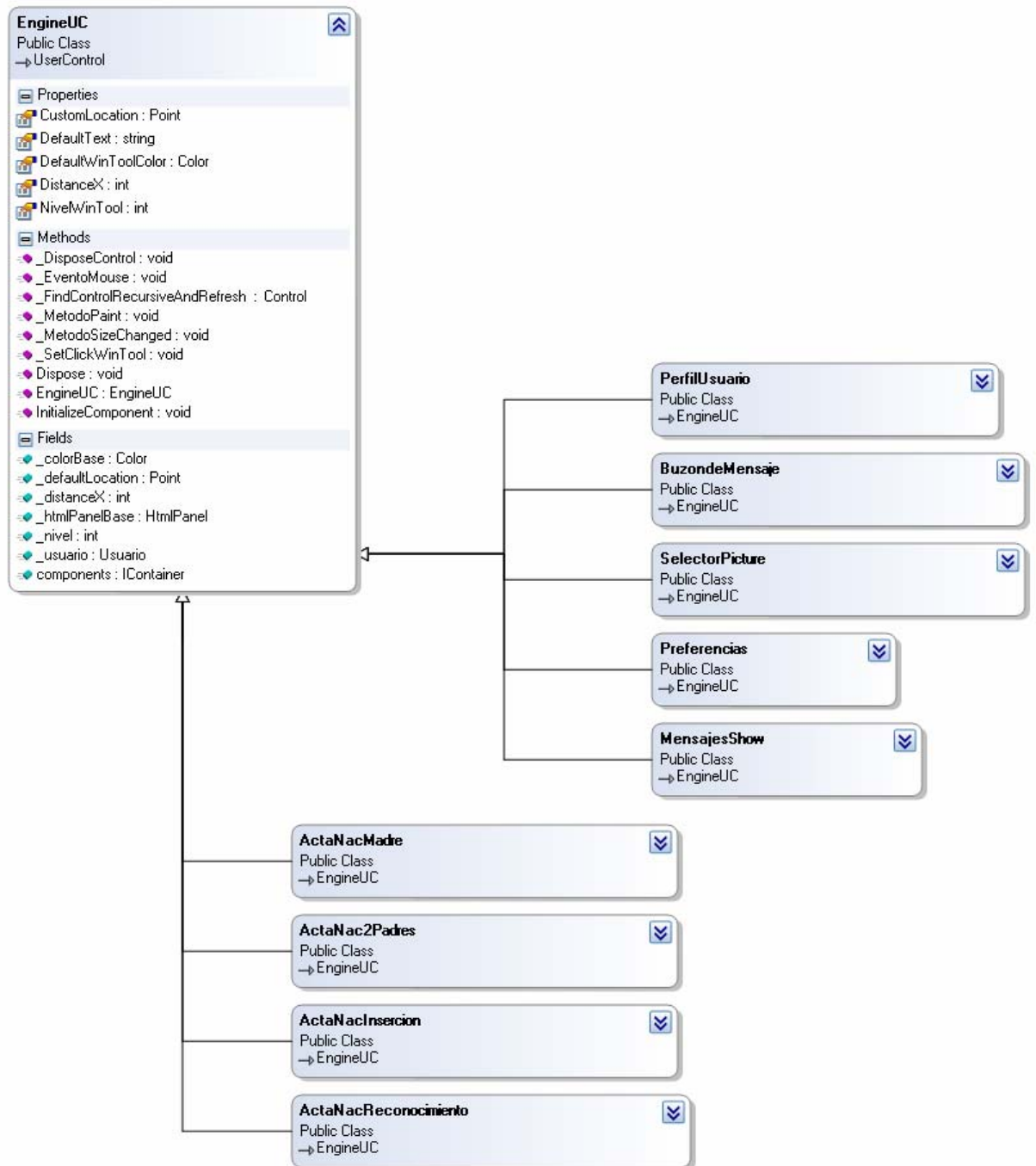


Figura 3.22 Diagrama de clases para EngineUc (modo compacto)

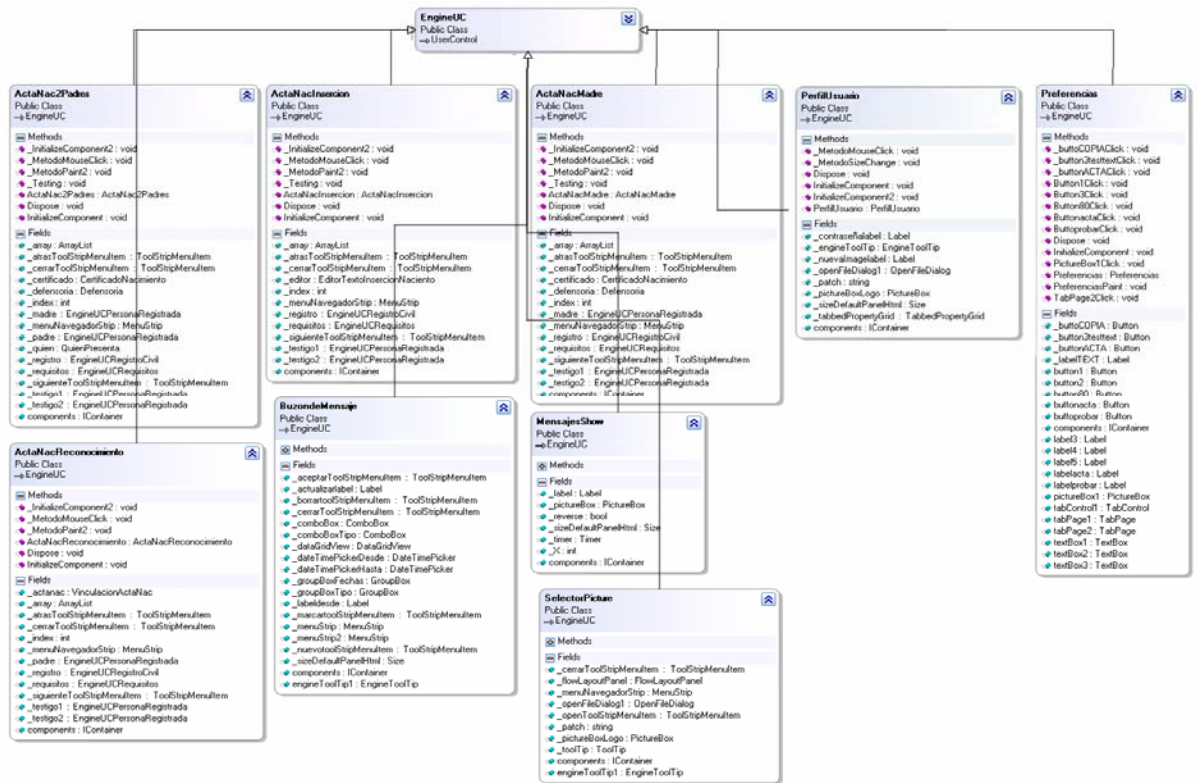


Figura 3.23 Diagrama de clases para EngineUc (modo expandido)

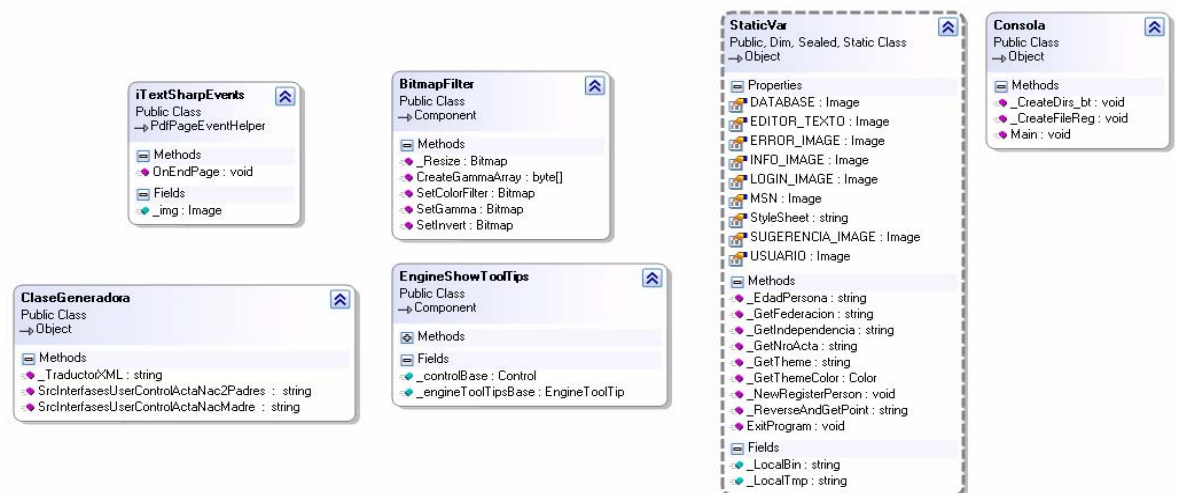


Figura 3.24 Diagrama de clases para StaticVar, iTextSharpEvents, BitmapFilter, ClaseGeneradora, EngineShowToolTips, Consola

3.7.7 Clases de conexión a la base de datos

En esta sección se describen las clases que conectarán el portal web tabla 3.21 y la aplicación de escritorio tabla 3.22 y 3.23, respectivamente, con la base de datos única, ubicada en el servidor.

Tabla 3.21 Descripción de la clase DB

\$db_handle	Este dato según la definición de la documentación oficial de PHP es "PostgreSQL connection resource on success" la cual almacena el valor devuelto por la función que se conecta a la base de datos, dicho valor toma el valor de False en caso de que falle dicha conexión
Método DB	Es el constructor de la clase, ejecutado en el momento de la instanciación de un objeto. Intenta realizar una conexión exitosa con la base de datos.
Método consultar	Es un método que ejecuta cualquier sentencia SQL que se requiera en la base de datos con la que se conectó el objeto en el momento de la instanciación. Y retorna como valor el resultado de dicha ejecución.

La figura 3.25 describe el diagrama de clases para DB y la figura 3.26 describe las clases ConectarPostgres y ConectarPostgresAux.

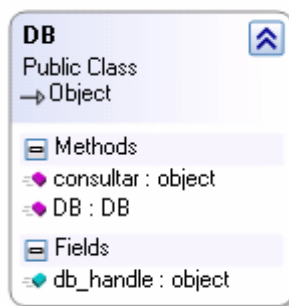


Figura 3.25 Diagrama de clases para DB

Tabla 3.22 Descripción de la clase ConectarPostgres (1/2)

DATABASE, _DATABASE	Especifica el nombre de la base de datos a conectarse.
ERROR , _ERROR	Almacena en un tipo de dato string los posibles errores durante la ejecución de la clase con la base de datos.
ID , _ID	Especifica el usuario de la base de datos.
PASSWORD, _PASSWORD	Almacena la contraseña para el acceso de la base de datos.
PORT, _PORT	Puerto de conexión default: 5432.
SENTENCE, _SENTENCE	Almacena la sentencia que será ejecutada.
SERVER, _SERVER	Dirección IP del servidor remoto o local default: localhost.
STATE	Almacena el estado de la conexión. True ó False.
Método _Conectar	Conecta a la base de datos y devuelve True si todo es un éxito de lo contrario devuelve False.
Método _Desconectar	Desconecta la clase de la base de datos.
Método _ExecuteAny	Ejecuta cualquier sentencia retornando un tipo object como resultado.
Método _ExecuteQuickQuery_ds	Ejecuta la sentencia y retorna un tipo DataSet.
Método _ExecuteQuickQuery_Ndr	Ejecuta la sentencia y retorna un tipo DataReader de la librería Npgsql
Método _FuncionImage	Estable imagen de almacenamiento ó actualización soporta solo tipos jpg, png.

Tabla 3.22 Descripción de la clase ConectarPostgres (2/2)

Método <code>_Get *</code>	Son métodos relacionados con la consulta de días, año, hora varían según el tipo de dato retornado
Método <code>BYTE_TO_IMAGE</code>	Establece un conjunto de byte a formato de imagen. Solo soporta png, jpg.
Método <code>EXECUTE_getIMAGE</code>	Inserta la imagen a la base de datos
Método <code>IMAGE_TO_BYTE</code>	Transforma una imagen a formato binario.

Tabla 3.23 Descripción de la clase ConectarPostgresAux

Método <code>_AperturadeLibro</code>	Este método crea un nuevo libro de nacimientos basado en el año actual.
Método <code>_GetNextTomoActaLibro Year</code>	Este método retorna la siguiente acta basado en el año especificado en los parámetros respectivos.
Método <code>_gimmeData</code> y <code>_GimmeData2</code>	Son métodos que proporcionan valores según los parámetros de la sentencia previamente ya elaborada.

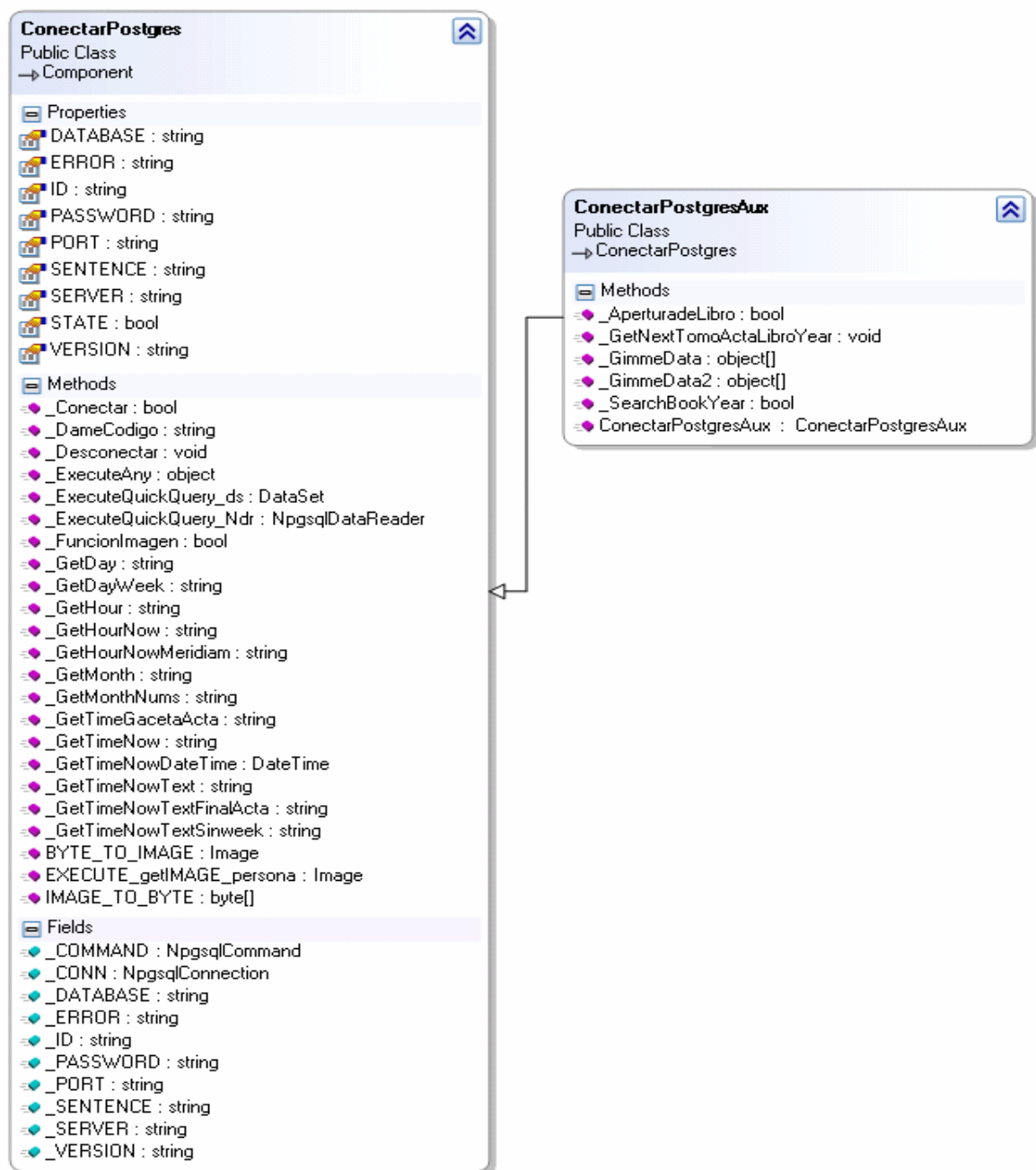


Figura 3.26 Diagrama de clases para ConectarPostgres y ConectarPostgresAux (modo expandido)

3.8 Evaluación de la Fase de inicio

Con esta primera fase del proceso unificado se obtuvo una visión general de los requerimientos del proyecto, características claves y riesgos principales, lográndose esto mediante:

- La determinación de los requisitos generales del sistema, donde se trató de establecer el alcance o límite de forma precisa.
- La creación del modelo de dominio para definir el ámbito del sistema, identificar todas las posibles entidades y relaciones entre estas. Esto permitió determinar la división del sistema en dos áreas, una pública (accesible por todos los usuarios de internet) y otra privada (para la administración interna) dirigido a los usuarios del registro civil.
- La definición de los casos de uso principales para cada subsistema, que permitieron plasmar lo solicitado en los requisitos del sistema, siendo el nivel de definición de estos muy generales debido a que en las próximas fases se extenderá cada uno a un nivel de detalle superior.
- La identificación de los riesgos del sistema.
- El desarrollo de la iteración de análisis, en la cual se desarrollaron los diagramas de actividades para representar los flujos de ejecución de las operaciones generales determinadas en los casos de uso.

- Por último, se desarrollaron los diagramas de clases y paquetes en una iteración de diseño, mostrando las posibles clases del sistema y sus interrelaciones. Al mismo tiempo, se agruparon cada una de estas en los posibles paquetes de clases con los que se pueda construir el sistema.

CAPÍTULO IV: FASE DE ELABORACIÓN

4.1 Introducción

El propósito de la etapa de elaboración figura 4.1, es crear la línea base de la arquitectura para así disponer de unos cimientos sólidos sobre los que se basará el diseño e implementación durante la fase de construcción. La arquitectura evoluciona considerando los requisitos más significativos (aquellos que tienen un fuerte impacto en la arquitectura del sistema) y evaluación de riesgos.



Figura 4.1 Diagrama de flujo de trabajo de las fases del proceso unificado, con identificación de la fase de elaboración.

4.2 Planificación de la fase de elaboración

En esta fase serán modificados y depurados los modelos creados en la fase de inicio, generando así otro conjunto de modelos que irán perfilando

una solución más cercana al mundo real, se construye la línea base de la arquitectura incluyendo técnicas de diseño globales. Se establecen los procesos, capas de software, paquetes, subsistemas, identificando sus responsabilidades y se efectúa la depuración de las interfaces internas y externas, refinándolas e incluyendo parámetros y valores de retorno.

Por último se realizarán en la iteración de implementación el diagrama de despliegue de la aplicación web, así como también se desarrollará un diagrama parcial de componentes.

4.3 Requisitos

En esta nueva iteración de la fase de elaboración no se identificaron nuevos actores, en los casos de usos se han captado nuevos requisitos para terminar de validar todas las funcionalidades del sistema.

4.3.1 Lista de requisitos ó características

A la lista de requisitos identificados en la fase de inicio, se anexan las siguientes características:

- Incluir controles para inicio de sesión en el componente de acceso de usuarios web, con el fin de limitar el acceso a las secciones que así lo requieran por parte de los internautas y garantizar así la seguridad del sitio web
- Incluir un control en la sección de actividades del registro para controlar el acceso por parte del personal del registro civil.

4.3.2 Modelo de casos de usos actualizado

En esta sección se presentan los casos de uso que conforman al sistema en general, incluyendo aquellos desarrollados en la fase de inicio, sus modificaciones y el resto de los casos de uso específicos que completan los requisitos.

Los actores y casos de uso generales que conforma el sistema no varían con respecto a los identificados y descritos en la fase de inicio. Sin embargo, se incluyó un caso de uso en el diagrama de la figura 3.3, Este nuevo caso de uso está relacionado con la adición del requisito descrito en esta fase de elaboración, e implica la validación de las credenciales del internauta al tratar de acceder a una página/componente que haya sido marcada como protegida.

En la figura 4.2 se muestra el diagrama de casos de uso actualizado del sistema y en la tabla 4.1 y 4.2 se describe el caso de uso añadido en esta iteración.

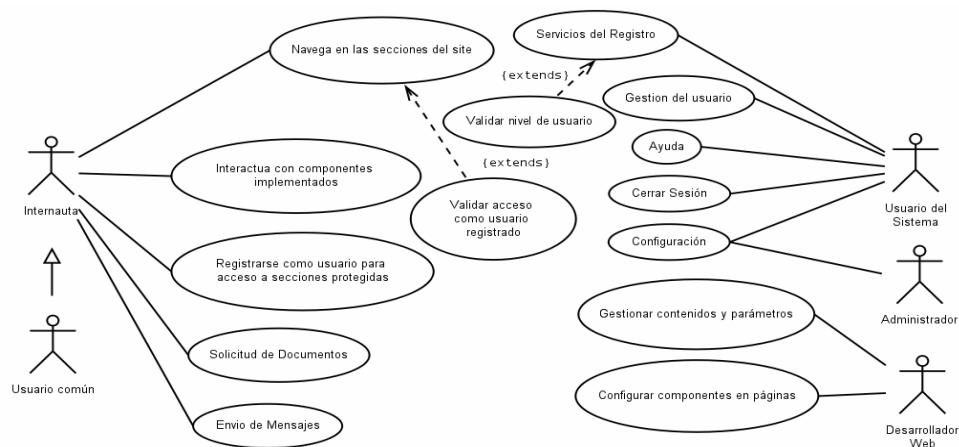


Figura 4.2 Diagrama de casos de uso

Tabla 4.1 Casos de uso Validar acceso como usuario registrado

ACTOR PRINCIPAL:	Internauta
ALCANCE:	Área pública
NIVEL:	Sub-función
DESCRIPCIÓN:	Al acceder a una sección diseñada como protegida, deberá validar sus credenciales de acceso. Al iniciar la navegación de una página del website se verificará si esta página fue marcada como protegida y en caso de que así fuese, solicitará la validación de credenciales.

Tabla 4.2 Casos de uso Validar nivel de usuario

ACTOR PRINCIPAL:	Internauta
ALCANCE:	Área pública
NIVEL:	Sub-función
DESCRIPCIÓN:	Si el usuario del sistema intenta acceder a una sección protegida, deberá validar sus credenciales de acceso ó simplemente no podrá acceder a la sección solicitada.

4.3.3 Mitigación de los riesgos

Tras la identificación de los riesgos potenciales en la fase de inicio, se presenta a continuación una lista de las soluciones que se consideraron más adecuadas y oportunas, evaluando todos los factores necesarios, incluyendo los puntos que influyen o pudieran interferir directamente con el desarrollo del proyecto.

- Tras el riesgo en el crecimiento de la complejidad del sistema, se ha determinado una gran cantidad de similitudes en los componentes

requeridos por el sistema, lo que permitirá generalizar muchas de las funcionalidades en los componentes previstos en la iteración de la fase de inicio.

- Respecto al uso de la tecnología PHP (PHP Hypertext Pre-processor) ya se ha visto soporte de esta por parte de los principales proveedores de hospedaje a nivel mundial. Por esta razón, deja de ser un riesgo el uso de esta plataforma para la implementación del website.
- Respecto al uso de la tecnología C# (ci-sharp en ingles), su utilización se adapta perfectamente bajo el Proyecto Mono por ser un proyecto independiente de la plataforma, además actualmente Mono funciona en GNU/Linux, Mac OS X, Windows.

4.4 Análisis

Se presenta a continuación un análisis de los requisitos anteriormente agregados o modificados.

4.4.1 Diagrama de actividades

Los aspectos agregados en la iteración de requisitos de esta fase sólo alteraron la representación de una actividad y adicionó una nueva.

4.4.1.1 Navegar a una sección del website

Esta actividad se ejecuta cada vez que un internauta intenta acceder a una página. Primero se verifica si dicha página es de acceso público o

privado, ya que se requiere que el usuario este registrado en el portal para poder acceder a las secciones privadas del mismo. En caso de que el usuario no esté registrado, puede proceder a registrarse en la sección del portal destinada a dicho proceso. Una vez que el usuario esta registrado y correctamente identificado en el portal, estará en capacidad de hacer uso de los servicios avanzados de la WEB, como lo son, el envío de comunicados a los funcionarios públicos y la solicitud de documentos civiles en línea. Todo este conjunto de procesos se ven reflejados en el diagrama de actividades de la figura 4.3.

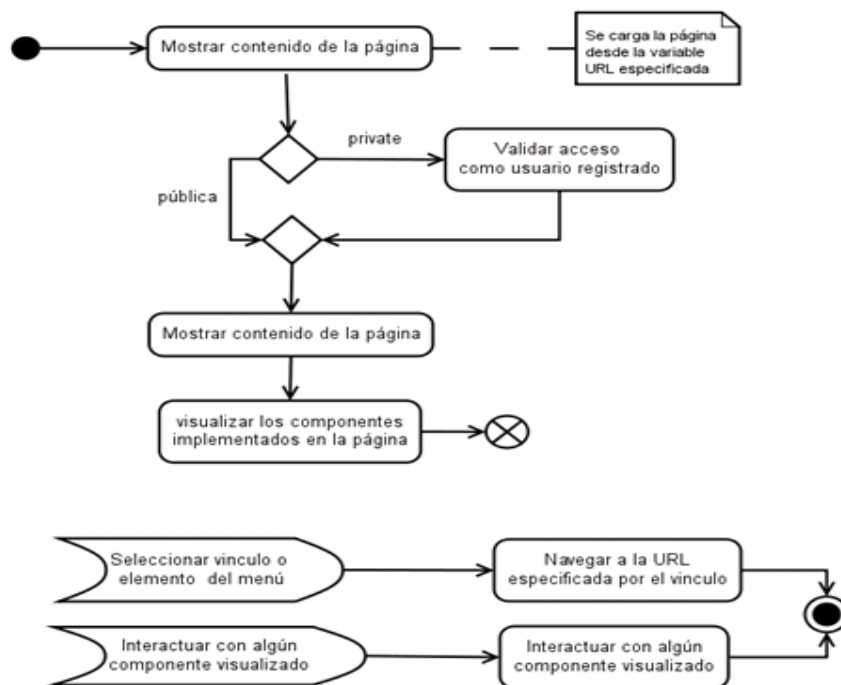


Figura 4.3 Diagrama de actividad Navegar en las secciones del website

4.4.1.2 Validar nivel de usuario

La figura 4.4 muestra la validación se realiza cuando el usuario de la aplicación intenta acceder a un control con restricciones, el nivel de acceso

será evaluado, si es permitida, continúa y despliega el control y en caso contrario, mostrará un mensaje donde refleja en nivel requerido.

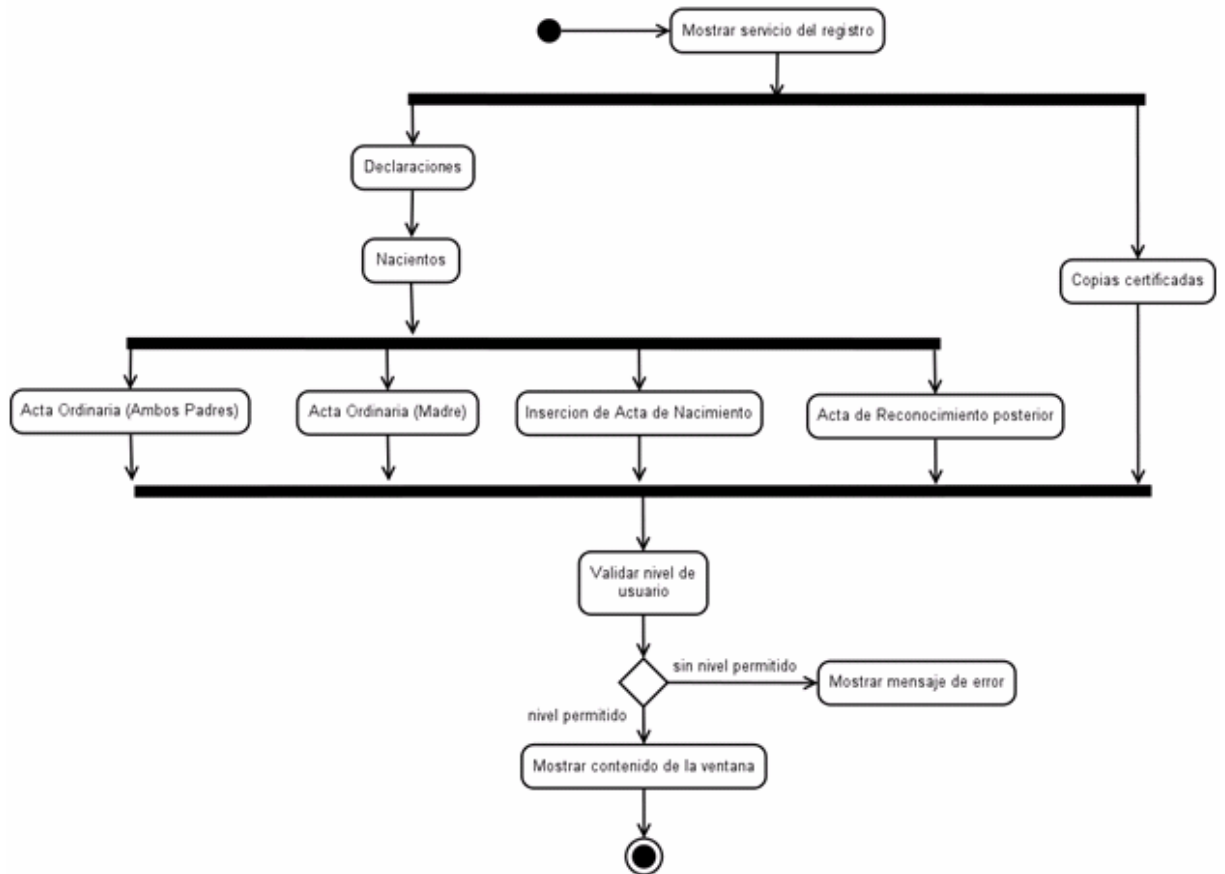


Figura 4.4 Diagrama de actividad Navegar en las secciones del website

4.5 Diagrama de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase; contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos. Estos diagramas modelan la secuencia lógica a través del tiempo. A continuación se muestran los diagramas de secuencia

de tres de los casos de usos más importantes del sistema en general. En la figura 4.5 y 4.6 se muestra el diagrama de secuencia para el caso de uso Gestión del usuario ► Nuevo registro de ciudadano en sus dos modalidades actualización y nuevo registro, en el se aprecia como el usuario del sistema solicita la inscripción de un nuevo ciudadano al registro, una vez activada la interfaz FormMain.cs se activa la interfaz gráfica PreviewNuevoRegistroCiuda.cs para la captura de los datos requeridos. Seguidamente, se realiza la verificación de la existencia del registro; el proceso de inscripción en el registro debería realizarse una vez a lo largo de su vida mediante el FormPersona.cs, su data no se elimina de la base de datos, sólo se actualiza en caso de ser necesario y con una autorización previa. Una vez se recibe la data del registro (exista o no) se puede procede a solicitar la activación de la cámara Web a través de la clase SelectorPicture.cs para capturar la foto del ciudadano. Una vez capturada la foto, se envía la data de la interfaz gráfica al gestor de inscripción que realiza el procesamiento de la data y el registro físico de la misma a través de una instancia de la clase ConectarPostgres.cs.

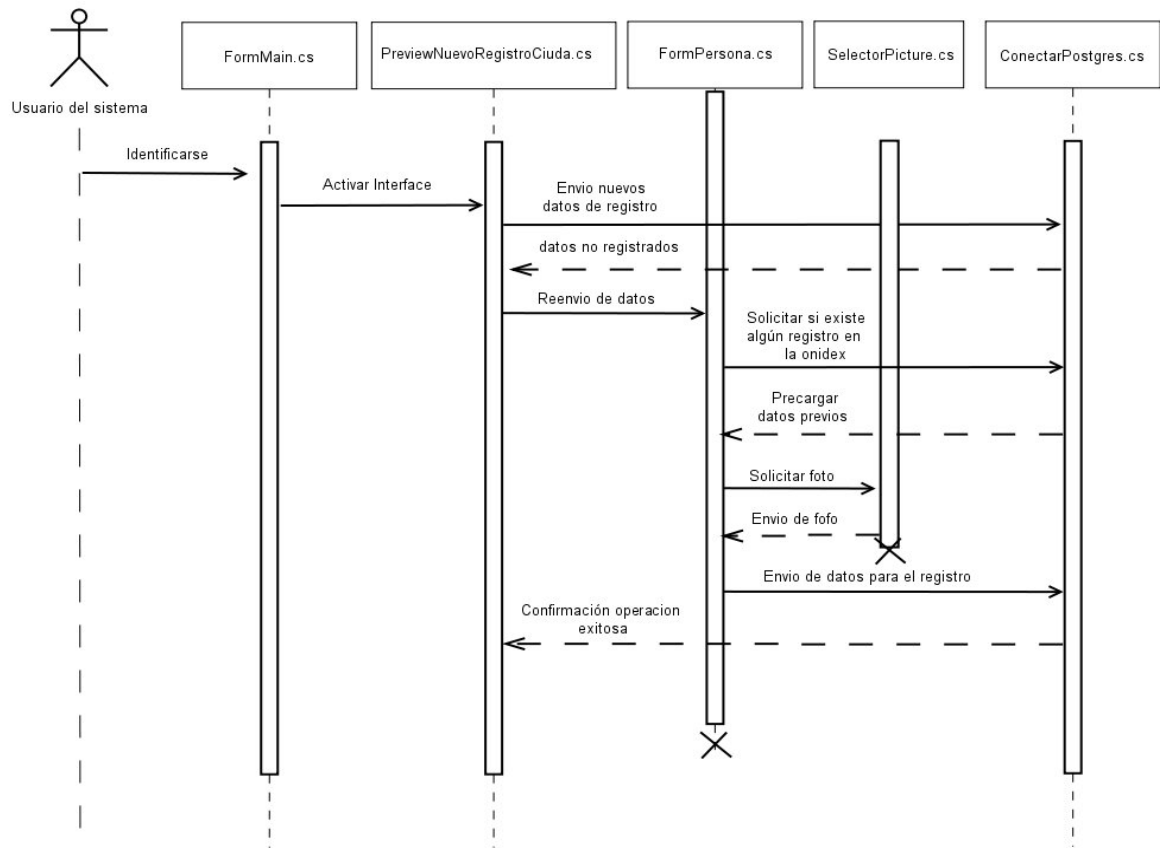


Figura 4.5 Diagrama de actividad Navegar en las secciones del website

Para la actualización de un ciudadano una vez activada la interfaz FormMain.cs se activa la interfaz gráfica PreviewNuevoRegistroCiuda.cs para la captura de los datos para actualizar. Seguidamente, se realiza la verificación de la existencia del registro; el proceso de actualización en el registro debería realizarse bajo una autorización ya que existe una restricción de nivel de usuario el FormPersona.cs se activa con los datos solicitados, su data no se actualizará hasta que la clase Autorización.cs sea válida, se puede solicitar la activación de la cámara Web a través de la clase SelectorPicture.cs para capturar una nueva foto del ciudadano. Una vez capturada la foto o no, se envía la data de la interfaz gráfica al gestor de

actualización que realiza el procesamiento de la data y el registro físico de la misma a través de una instancia de la clase ConectarPostgres.cs.

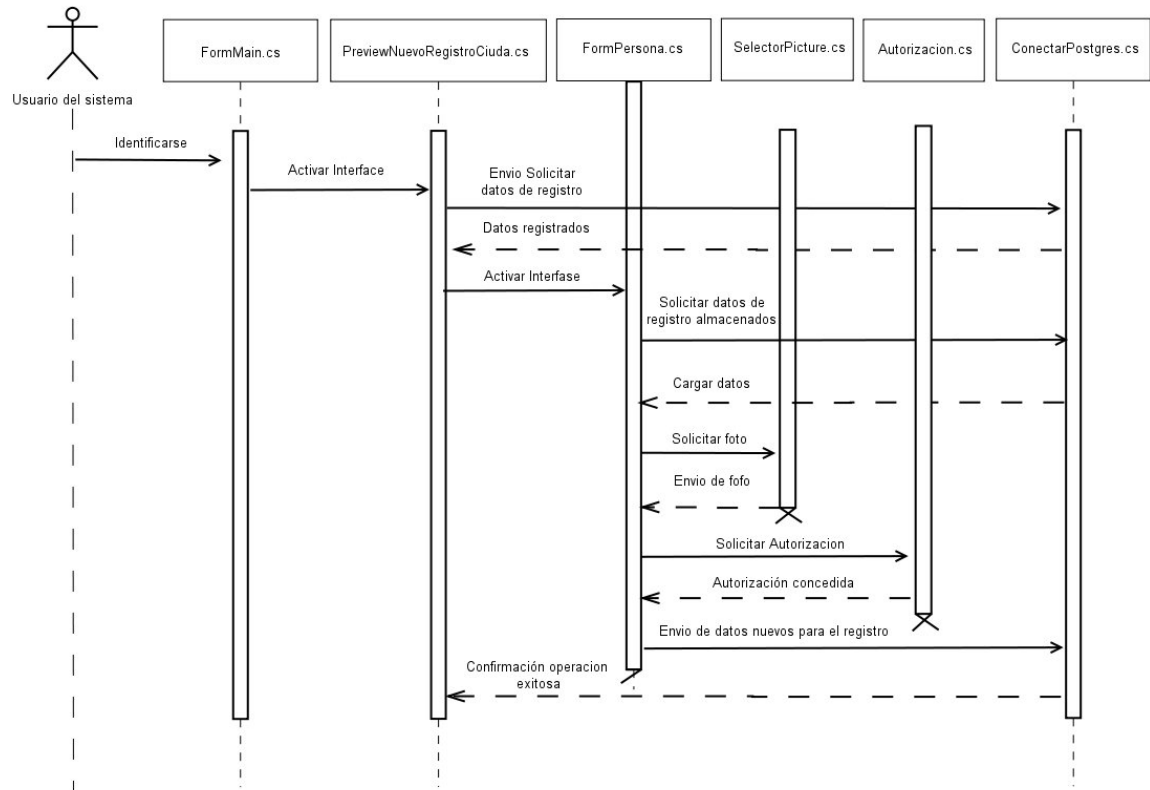


Figura 4.6 Diagrama de secuencia actualizar ciudadano

En la figura 4.7 se muestra el diagrama de secuencia para el caso de uso Declaraciones ► Nacimientos ► Ordinario (Ambos Padres). Se inicia cuando el usuario del sistema solicita iniciar la elaboración del acta, una vez dentro del FormMain.cs solicita que se active la interfaz gráfica ActaNac2Padres.cs se verifica si posee el nivel de usuario necesario para elaborar el acta, luego se activa la interfaz EngineUCRequisitos.cs con los requisitos necesarios para elaborar el acta, envía la data recogida, se activa la interfaz EngineUCPersonaRegistrada.cs que solicita los datos de los

padres si algunos de ellos no se encuentra registrados ó se desea actualizar datos se procede a los pasos de la figura 4.5 y 4.6 respectivamente. El gestor de información de la clase ActaNac2Padres.cs una vez que padres y testigos son identificados procede a llenar los datos del certificado de nacimiento expedido por el centro hospitalario ó clínica mediante la clase CertificadoNacimiento.cs, de ahí se pasa la data para solicitar si el acta se realiza mediante el tiempo estipulado por la ley controlado por la clase Defensoria.cs, luego se activa otra vez la clase EngineUCPersonaRegistrada.cs que solicita los datos de los testigos si algunos de ellos no se encuentra registrados ó se desea actualizar datos se procede a los pasos de la figura 4.5 y 4.6 respectivamente. El objeto gestor de nacimientos activa la interfaz EngineUCRegistroCivil.cs ya que este contiene todo lo referente al acta de nacimiento elaborada, número de acta, tomo, año, registrador(a) civil y escribiente donde se puede visualizar el acta de nacimiento tanto como la que va al libro y como la copia certificada que es entregada al usuario final, se valida la información y procede a enviar la data a la clase ConectarPostgres.cs para registrar el proceso.

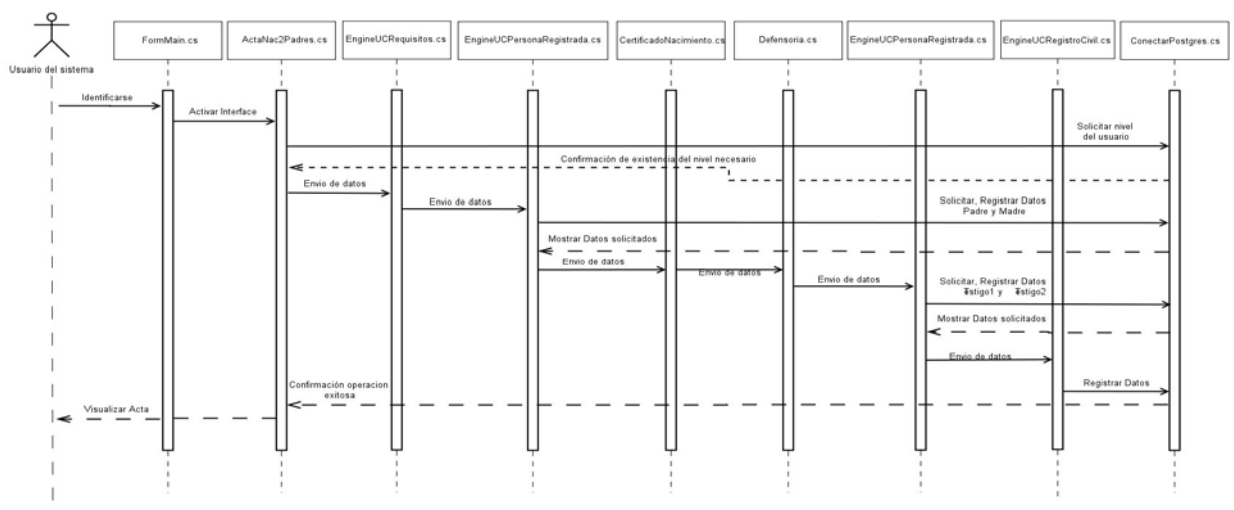


Figura 4.7 Diagrama de secuencia Acta Nacimiento Ordinaria

En la figura 4.8 se muestra el diagrama de secuencia para el caso de uso Solicitar Documentos. Se inicia cuando el internauta solicita algún documento del registro civil, una vez dentro del index.php puede identificarse como usuario registrado ó registrarse mediante registro.php, en caso de algún registro y el internauta ha olvidado su contraseña puede dirigirse a recuperar.php donde podrá recuperar todos sus datos, luego de identificarse como usuario del portal web puede realizar las siguientes operaciones: solicitar algún documento existente dentro de la base de datos del registro civil, en la cual se activa la interfaz solicitam.php donde podrá seleccionar el documento necesitado, además puede tener interacción directa con algún funcionario que esté conectado al sistema SRC mediante la página solicitudes_izq.php, todo esto mediante la clase ClaseBaseDeDatos la cual permite el acceso a la base de datos interna del registro civil desde la pagina.

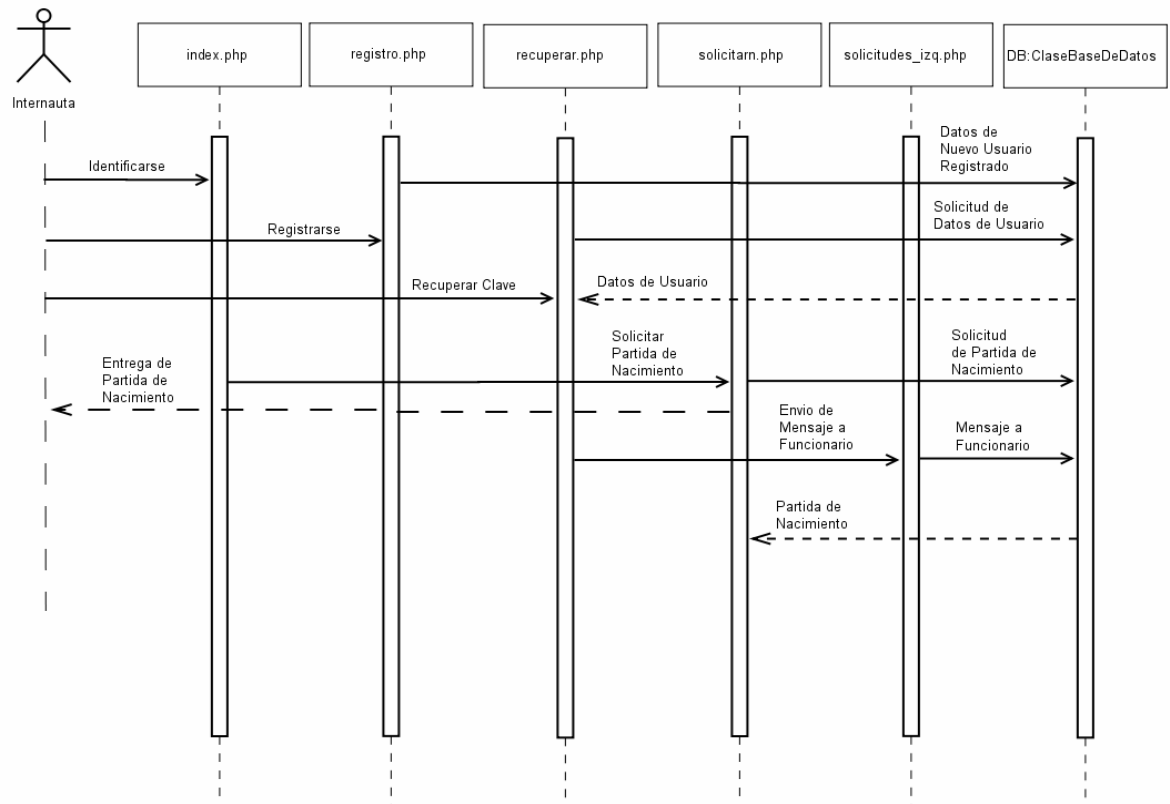


Figura 4.8 Diagrama de secuencia Acta Nacimiento Ordinaria

4.6 Diagrama de capas

La arquitectura de los sistemas de cómputo suele estructurarse en especies de estratos. Cada uno de estos estratos, mejor llamados capas, incluirá paquetes funcionales relacionados con la naturaleza de la capa.

Este diagrama busca representar cómo se relacionan los distintos paquetes que conforman al sistema, estructurados en cuatro capas que serán organizadas desde la de más alto nivel hasta la de más bajo: específica, general, intermedia y de software. El hardware y el software se complementan para definir el sistema informático. Las aplicaciones de

software facilitan a los usuarios comunicar sus necesidades al computador sin tener que aprender a programar. La Figura 4.9 muestra un diagrama de capas donde se describe el sistema informático en términos de varios niveles desde el más externo (donde se encuentra el usuario) hasta el más interno (donde se encuentra el hardware). La información fluye entre las diferentes capas a nivel real y a nivel lógico.

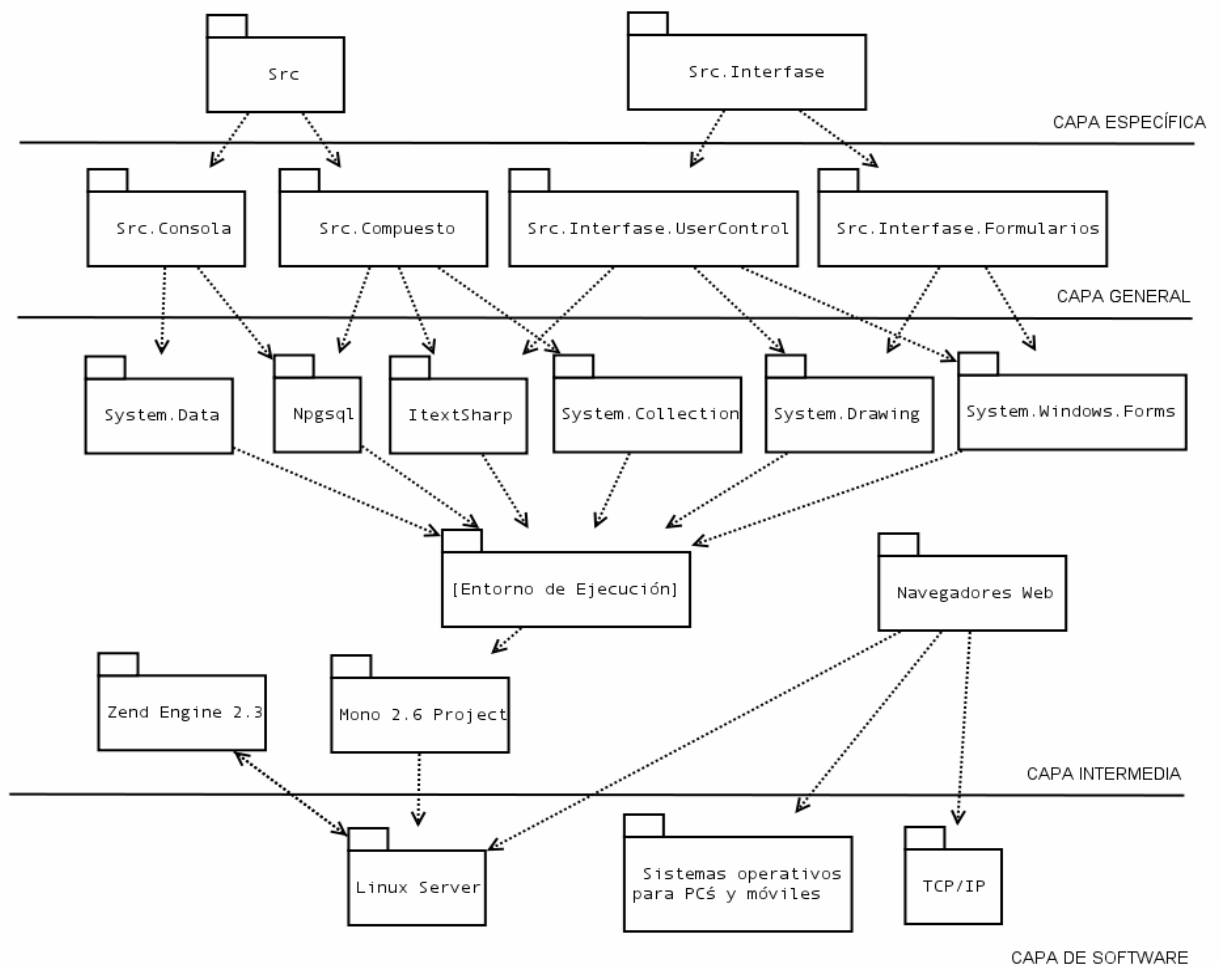


Figura 4.9 Diagrama de capas del sistema



4.7 Diagrama de Base de Datos

La base de datos relacional es el modelo de base de datos más difundido en la actualidad, en gran medida debido a que ofrecen sistemas simples y eficaces para representar y manipular los datos. Antes de realizar el diseño relacional de las tablas que conforman el sistema, a continuación se identificarán y explicarán cada una de ellas.

4.7.1 Identificación de las tablas


En esta etapa del diseño de la base de datos se han identificado un total de 27 tablas con las que contará el sistema. A continuación se muestran cada una de estas.



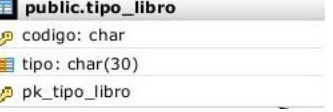
Tabla 4.3 Tabla de base de datos que conforman el sistema (parte 1/4)

Tabla	Entidad	Descripción
usuarios_sistema		Aquí se almacenan cada uno de los usuarios del sistema, tanto administradores como el personal del registro civil representado en el campo usuario; el campo cedula hace referencia a la tabla persona donde están todos sus datos personales.
persona		Se almacena toda la información de la persona los nombres de la persona se almacenan en un campo mientras que los apellidos son separados para ser utilizados dentro del proceso de elaboración de las actas.

buzón		<p>Se lleva un registro de mensajes donde se hace una interacción entre los usuarios del sistema y el público en general, también dentro del proceso de mensajería se encuentra el envío de documentos en formato XML almacenado en el campo ref_xml, el campo tipo hace referencia a la tabla tipo_documento donde se encuentra una gran cantidad de documentos que procesa el registro.</p>
entidades		<p>Representa todas las entidades de una gran cantidad de países, el campo pertenece_pais hace referencia a el país de acuerdo con la entidad especificada</p>
estados_civiles		<p>Representa todos los estados civiles establecidos por el código civil de Venezuela.</p>
etnias_indigenas		<p>Representa todas las etnias indígenas presentes en el territorio venezolano.</p>
localidades		<p>Se establece una localidad por parroquia existente en el territorio venezolano, el campo pertenece_parroquia hace referencia a la parroquia que pertenece esta localidad.</p>
lugares		<p>Representa el lugar de nacimiento tanto como el de residencia actual de la persona representada por el campo código por ejemplo: N16562748, R16562748, los demás campos hacen referencia a sus tablas respectivas.</p>

municipios		Se establece un municipio por entidad existente en el territorio venezolano, el campo pertenece_entidad hace referencia a la entidad que pertenece este municipio.
nacionalidades		Representa las nacionalidades existentes en este planeta.
persona_nivel_educativo		Hace referencia al nivel educativo que tiene una persona si sabe leer y escribir, nivel educativo básico, primario, etc., grado educativo.
ocupaciones_habituales		Representa la ocupación habitual de la persona. Obrero, albañil, etc.
onidex		Hace referencia a los ciudadanos registrados en la base de datos de la onidex venezolano del año 2004, este trabajo se realizó transfiriendo una bases de datos en Microsoft® Access a una nueva en Postgres 8.4. Esta tabla contiene 19.964.300 registros
países		Representa los países dentro de los cinco continentes de este planeta.
parroquias		Se establece una parroquia por municipio existente en el territorio venezolano, el campo pertenece_municipio hace referencia al municipio que pertenece esta parroquia.

persona_pasaporte		Establece si algún ciudadano no posee algún documento de identificación nacional.
persona_contacto		Establece el contacto del ciudadano ó persona registrada al sistema como teléfono y correo electrónico.
persona_foto		Establece alguna fotografía del ciudadano registrado. Se hace transformando la imagen a formato binario.
persona_registro		Hace referencia al registro inicial se almacena su fecha del primer registro y la última actualización que realiza el ciudadano dentro del sistema.
pertenece_etnia		Establece si algún ciudadano pertenece a alguna etnia indígena y si domina el lenguaje.
profesiones		Establece las profesiones que algún ciudadano pudiera tener.
registro_civil		Esta tabla hace referencia a los elementos que son variables dentro de un registro civil ya que los cargos de registradores civiles son de libre remoción y a veces llevan cambios repentinos con esta tabla soluciona el problema de reelaborar el software sino actualizar la tabla.

usuarios_web		<p>Establece los datos que se requieren para que los internautas accedan a los sitios protegidos y así tener una validación del internauta y el ciudadano registrado basado en la base de datos de la onidex.</p>
tipo_documento		<p>Establece los distintos tipos de documentos que genera un registro civil como por ejemplo: Acta de Nacimiento Ordinaria, Inserción de Acta de nacimiento.</p>
tomo_libro		<p>Establece los tomos de los libros que se pudieran generar dentro del sistema como por ejemplo: Primero, Segundo, Tercero y así sucesivamente.</p>
tipo_libro		<p>Establece los libros del sistema ejemplo: Nacimiento, Matrimonio, Defunción, etc.</p>

4.7.2 Representación del diagrama relacional de base de datos

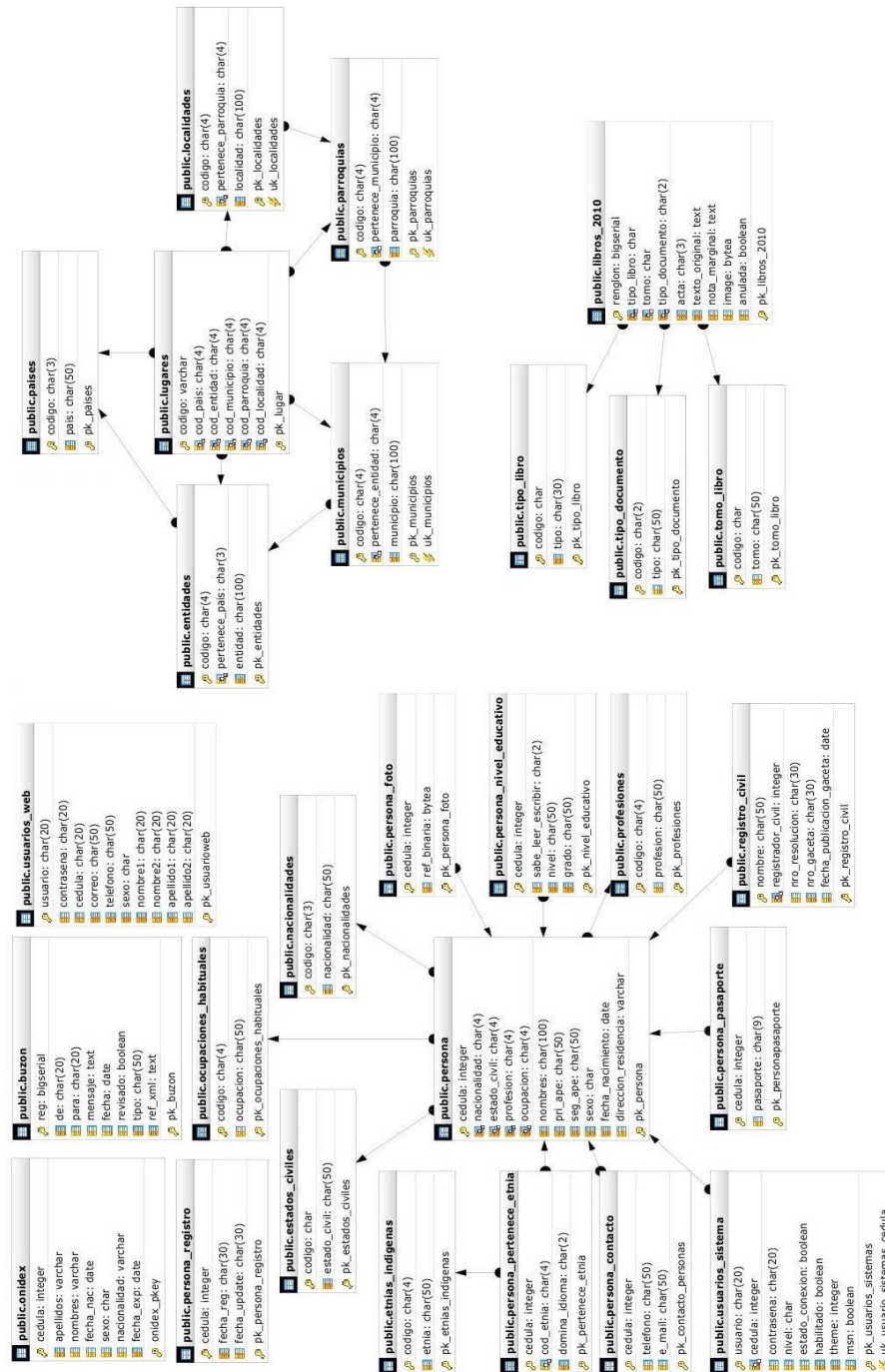


Figura 4.10 diagrama del modelo relacional.

4.7.3 Representación de los libros dentro del registro civil

Dentro de la base de datos del sistema existe una dinámica bien simple, durante la apertura de los libros dentro de un registro civil (parte física), el registrador civil toma un libro en blanco de 250 páginas tamaño oficio y procede a la apertura del primer tomo físico, allí se encuentran todas las actas dependiendo del libro aperturado, eso se realiza todos los años uno tras otros y si en algunos de los libros sus actas sobrepasan de el acta número 250, se hace la apertura del segundo tomo y así sucesivamente.

La dinámica del sistema es crear una tabla por cada año transcurrido. Es como visualizar un estante donde se encuentran una gran cantidad de libros ordenados por tipos de libros y tomos, pero todos dentro del mismo año, el campo renglon es creado con un tipo de dato *bigint* de 8 bytes en secuencia inicializada en -9223372036854775808 este tipo de dato establece un rango considerable de -9223372036854775808 hasta 9223372036854775807 lo suficiente como para insertar una gran cantidad de actas en distintos tipos de libros controlados por los campos *tipo_libro* y *tomo*, las actas son controladas por el campo *acta* desde 001 hasta 250, el campo *texto_original* establece el verdadero texto del documento en formato HTML, cabe destacar lo importante de las notas marginales que se realizan dentro del acta sin alterar el documento original también se elabora en formato HTML, por ser necesaria la utilización del sistema digitalizado se agrega un campo *image* donde se añade la posibilidad de hacer digitalización del documento ó documentos anteriores al sistema. El campo *anulada* se establece de tipo booleana para referenciar la anulación de una acta más no la eliminación física de la misma.

La apertura de los libros sigue la siguiente secuencia libros_#año# por ejemplo estamos en el años 2011 se crea una tabla llamada “libros_2011”.

Tabla 4.4 Representación de los libros dentro de un registro civil

Tabla	Entidad	Descripción
Libros_2010	 <pre> public.libros_2010 ├── renglon: bigserial ├── tipo_libro: char ├── tomo: char ├── tipo_documento: char(2) ├── acta: char(3) ├── texto_original: text ├── nota_marginal: text ├── image: bytea ├── anulada: boolean └── pk_libros_2010 </pre>	<p>Aquí se almacenan cada uno de los usuarios del sistema, tanto administradores como el personal del registro civil representado en el campo usuario; el campo cedula hace referencia a la tabla persona donde están todos sus datos personales.</p>

4.8 Evaluación de la fase de elaboración

En esta fase del proceso unificado se alcanzó un alto nivel de detalle en el análisis de requerimientos y diseño del sistema, así como un adelanto en la implementación, lográndose esto mediante:

- La determinación de un requisito adicional del sistema, estableciendo de forma definitiva el alcance del sistema.
- La adición de un caso de uso para representar el nuevo requisito, actualizado el diagrama de casos de uso correspondientes.

- El desarrollo de la iteración de análisis, en la que se realizó el diagrama de casos de usos correspondiente.
- Realización de la iteración de diseño, en la que se desarrolló el diagrama de secuencia para uno los casos de usos más importantes del sistema en general, la representación de la arquitectura del software en el diagrama de capas. Se realizó el diagrama de base de datos, detallando cada tabla junto con sus relaciones.

CAPÍTULO V: FASE DE CONSTRUCCIÓN

5.1 Introducción

La finalidad principal de la fase de construcción es alcanzar la capacidad operacional del producto. Durante esta fase todos los componentes, características y requisitos identificados y detallados en las fases anteriores deben ser adecuados a las posibilidades de la herramienta de desarrollo que se va a usar, para luego ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del sistema.

En esta fase se hace hincapié en las iteraciones de implementación y prueba del flujo de trabajo normal como se puede apreciar en la figura 5.1, debido a que su objetivo es lograr el desarrollo del software con calidad de producción, mediante la implementación de toda la funcionalidad y ejecución de las pruebas.

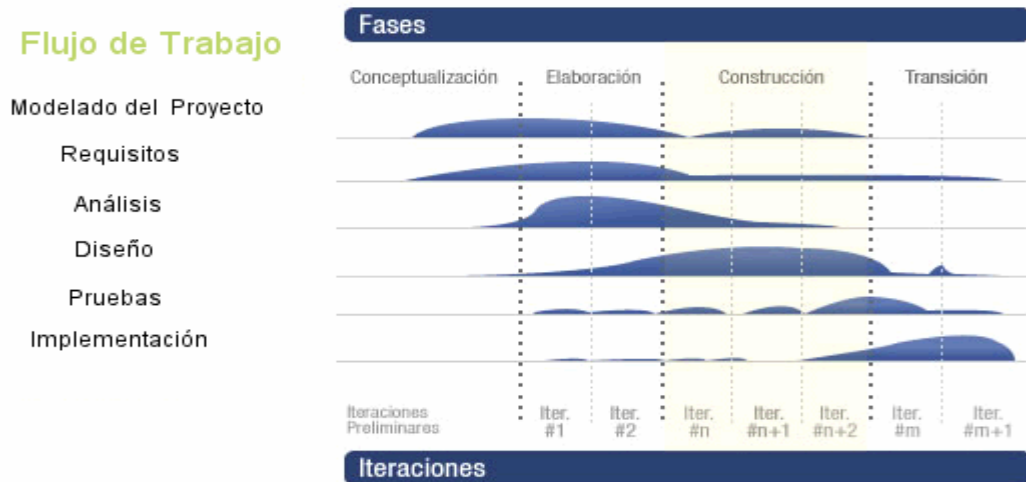


Figura 5.1 Diagrama de flujo de trabajo de las fases del proceso unificado, identificación de la fase de construcción.

5.1.1 Planificación de la fase de construcción

Para obtener la versión funcional o beta del sistema en general, se codificarán durante esta fase todas las secciones según el diseño en las fases anteriores.

En la iteración de implementación se diagramarán los componentes del software, realizándolos a un alto nivel de detalle y mostrando la codificación e interfaz gráfica de los principales componentes. Finalmente, se describirán las pruebas por unidad y las pruebas de integración durante la iteración de pruebas.

5.1.2 Escogencia del lenguaje de programación

Para el desarrollo del SRC (Sistema del Registro Civil) se debe contar con un lenguaje orientado a objetos, debido a que este paradigma permite una mayor modularidad y escalabilidad del software, así como un mayor nivel de abstracción entre los diferentes módulos que conforman el sistema. Por lo tanto se escogieron los lenguajes de programación C# para la aplicación, debido a que es simple, orientado a objetos, distribuido, interpretado, robusto, seguro, portable, de altas prestaciones, multitarea y dinámico. Y se utilizarán el lenguaje PHP para el sitio web, debido a que es un lenguaje híbrido, el cual puede ser utilizado tanto estructurado como orientado a objetos, es robusto, seguro, independiente del sistema y de la arquitectura, de altas prestaciones y además es abstracto al usuario ya que es implementado del lado del servidor.

Como entorno de desarrollo integrado (IDE) para la aplicación se escogió Sharp Develop. Esta es una herramienta libre y gratuita que le

proporciona al usuario una interfaz para el diseño de interfaces graficas, capacidades de compilación y generación automática de archivos ejecutables (.EXE), entre otras.

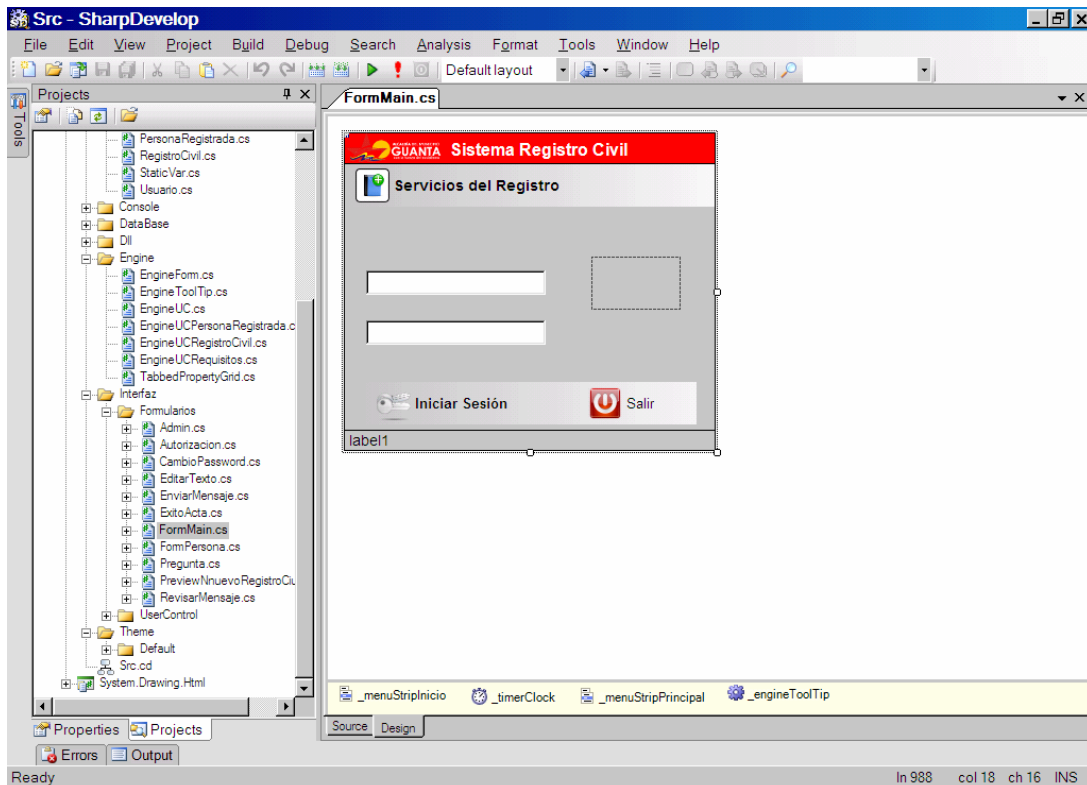


Figura 5.2: Interfaz del entorno de programación # Develop.
Fuente: Elaboración propia.

Para el sitio web se escogió como entorno de desarrollo Kate (KDE Advanced Text Editor). Es un editor multilenguaje de programación perteneciente al entorno grafico KDE, es un editor con altos niveles de personalización y muy simplista, el cual hace la programación en el muy eficiente y ajustada exactamente a las necesidades del desarrollador.

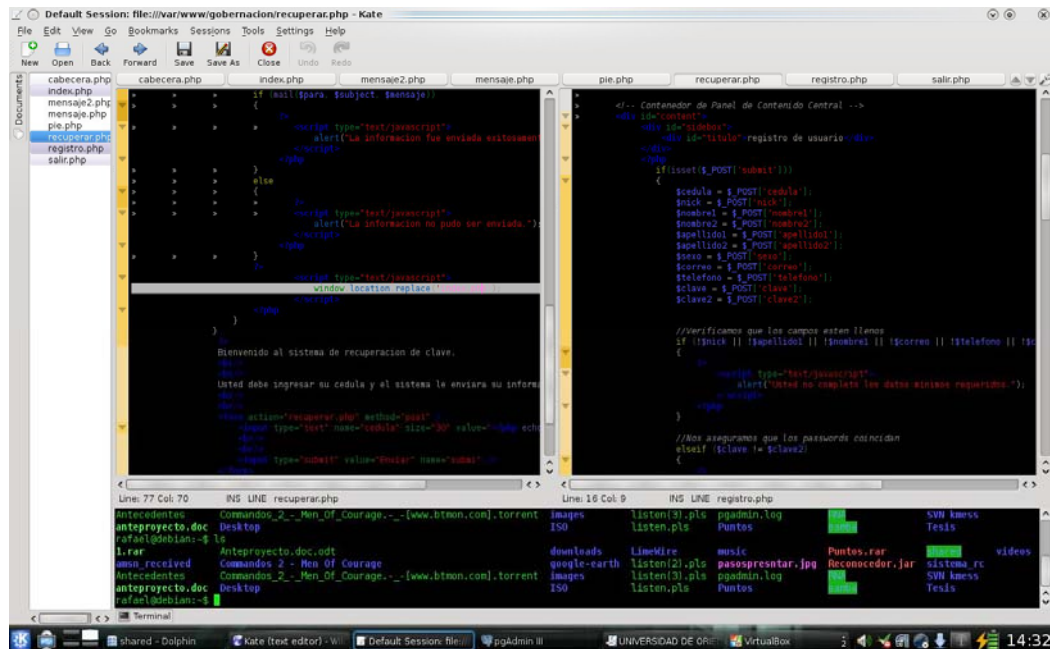


Figura 5.3: Interfaz del entorno de programación Kate.
Fuente: Elaboración propia.

5.1.3 Escogencia del sistema gestor de base de datos

Para el desarrollo de la base de datos para el sistema SRC se utilizó PostgreSQL, particularmente su herramienta pgAdmin III, que es un sistema gestor de base de datos objeto-relacional. Entre sus características se encuentran que es software libre, multihilo, multiusuario y altamente extensible.

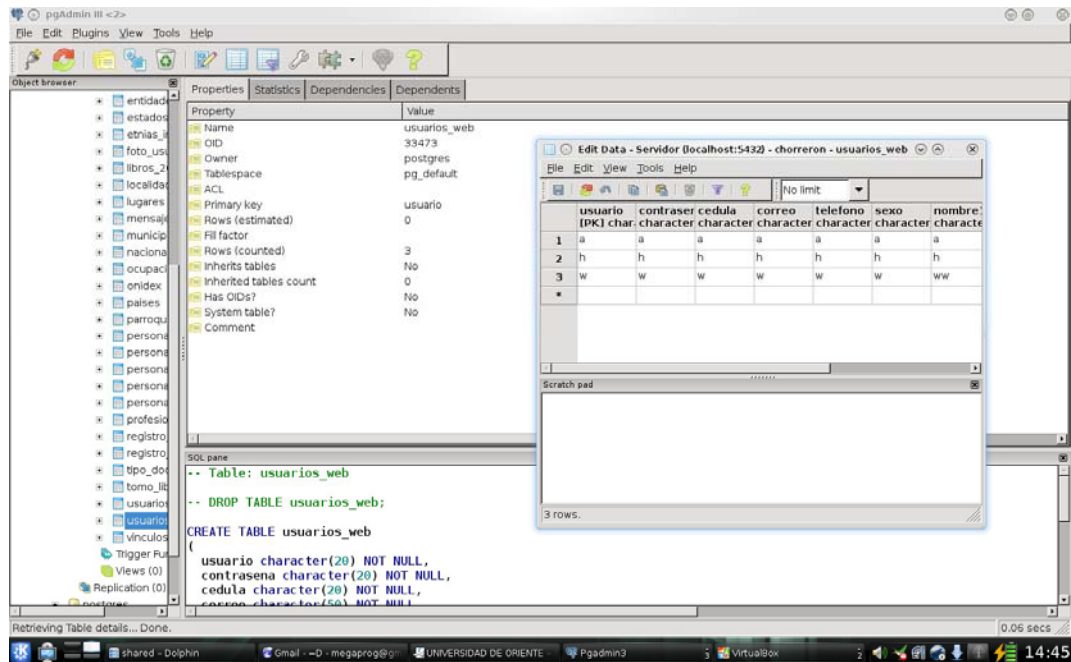


Figura 5.4: Interfaz de pgAdmin III.
Fuente: Elaboración propia.

5.2 Implementación

La implementación trata al sistema en términos de desarrollo de componentes y codificación del software, su relación con la base de datos, integración de módulos y la explicación acerca de las funcionalidades del sistema y su correcto uso, revelando así toda la estructura en forma de código abierto e identificando de las actividades que este puede realizar.

5.2.1 Diagrama de Componentes Totales

En la figura 5.4 se muestra el diagrama de componentes totales, indicando la totalidad de los componentes y las clases requeridas para su funcionamiento.

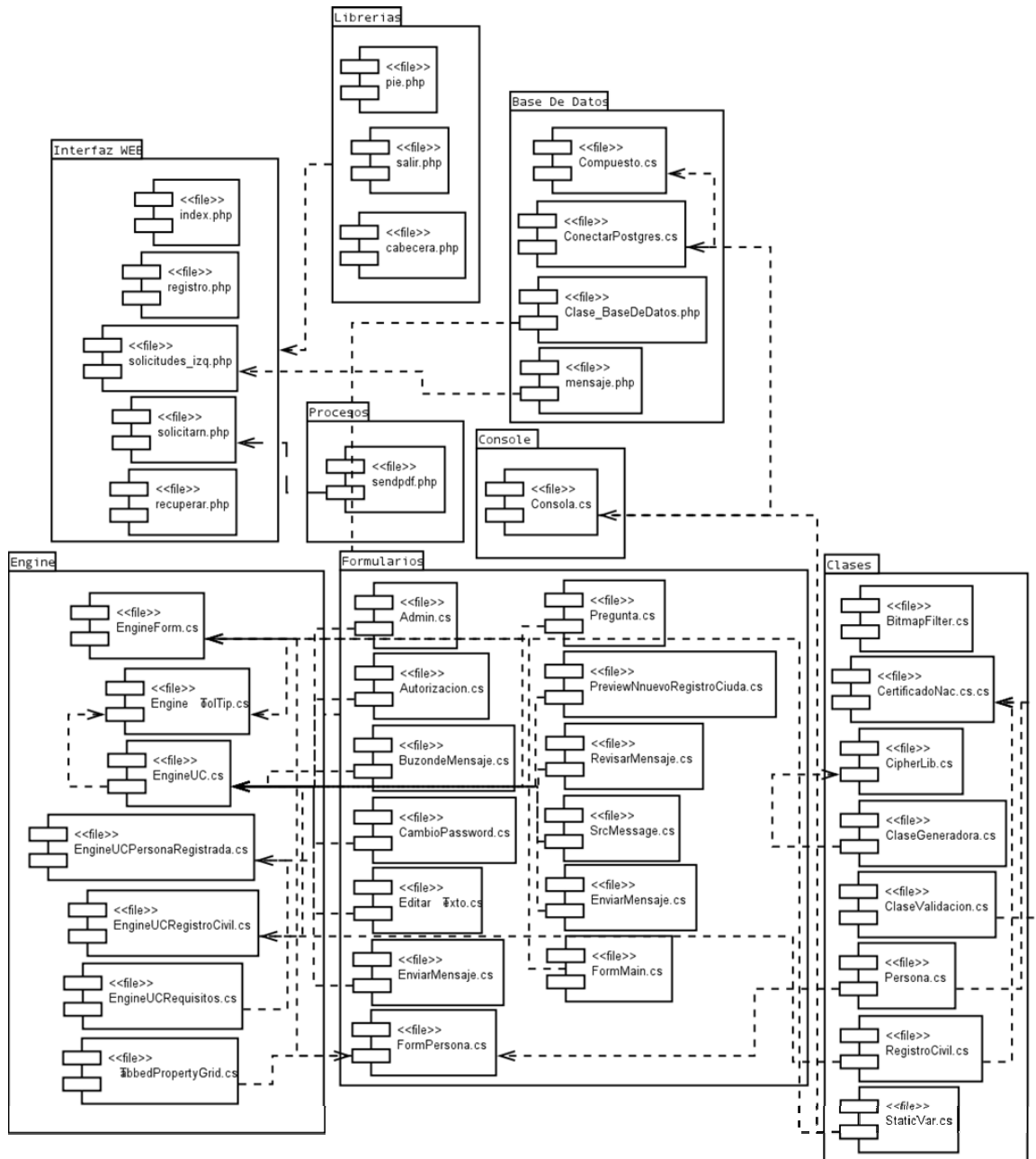


Figura 5.5: Diagrama de componentes totales para el sistema.
Fuente: Elaboración propia.

5.3 Pruebas

Las pruebas son la herramienta que permite validar y comprobar el software, es decir, son los procesos que determinan si el software satisface los requisitos y trabaja de la manera establecida. Estas pruebas tienen como finalidad verificar la interacción entre los objetos, al igual que la integración adecuada de componentes, confirmar que se satisfagan los requerimientos, identificar las fallas y corregirlas antes de la instalación.

5.3.1 Pruebas por unidad

Las pruebas por unidad se aplicaron mediante la prueba de la caja negra sobre los diversos componentes del sistema. Para la realización de este tipo de pruebas, se identifican un conjunto de valores que pueden ser introducidos por un actor, y se expresan como clases de equivalencia para poder abarcar la totalidad de las ocurrencias de un evento de inserción de datos.

A continuación en la tabla 5.1, se representan las clases de equivalencia del componente de solicitudes, el mismo se encarga de las descargas de partidas de nacimiento según parámetros de búsqueda.

Tabla 5.1: Clases de equivalencia del componente de solicitud de partidas. (1/2)

Número	Campo	Clase de Equivalencia	Válido	Inválido
1	Año Registro	Cadena de caracteres		X
2	Año Registro	Cadena alfanumérica		X
3	Año Registro	Caracteres especiales		X
4	Año Registro	Cadena numérica = 4	X	
5	Año Registro	Cadena numérica ≠ 4		X
6	Año Registro	Cadena Vacía		X

Tabla 5.1: Clases de equivalencia del componente de solicitud de partidas. (2/2)

7	C.I. Principal	Cadena de caracteres		X
8	C.I. Principal	Cadena alfanumérica		X
9	C.I. Principal	Caracteres especiales		X
10	C.I. Principal	Cadena numérica	X	
11	C.I. Principal	Cadena Vacía		X
12	C.I. Secundaria	Cadena de caracteres		X
13	C.I. Secundaria	Cadena alfanumérica		X
14	C.I. Secundaria	Caracteres especiales		X
15	C.I. Secundaria	Cadena numérica	X	
16	C.I. Secundaria	Cadena Vacía	X	

En la tabla 5.2, se representan las clases de equivalencia del componente de certificados de nacimientos, mediante el cual se generan dichos documentos.

Tabla 5.2: Clases de equivalencia del componente certificados de nacimiento. (1/2)

Número	Campo	Clase de Equivalencia	Válido	Inválido
1	Nombre Niño	Cadena vacía		X
2	Nombre Niño	Cadena de caracteres	X	
3	Nombre Niño	Cadena alfanumérica		X
4	Nombre Niño	Caracteres especiales		X
5	Nombre Niño	Cadena numérica		X
6	Hora Nacimiento	Cadena vacía		X
7	Hora Nacimiento	Cadena de caracteres		X
8	Hora Nacimiento	Cadena alfanumérica	X	
9	Hora Nacimiento	Caracteres especiales		X
10	Hora Nacimiento	Cadena numérica		X
11	C.I. Responsable	Cadena Vacía		X
12	C.I. Responsable	Cadena de caracteres		X
13	C.I. Responsable	Cadena alfanumérica		X
14	C.I. Responsable	Caracteres especiales		X
15	C.I. Responsable	Cadena numérica	X	
16	Nro. MPPS	Cadena Vacía		X

Tabla 5.2: Clases de equivalencia del componente certificados de nacimiento.(2/2)

17	Nro. MPPS	Cadena de caracteres		X
18	Nro. MPPS	Cadena alfanumérica		X
19	Nro. MPPS	Caracteres especiales		X
20	Nro. MPPS	Cadena numérica	X	
21	Nro. Certificado	Cadena Vacía		X
22	Nro. Certificado	Cadena de caracteres		X
23	Nro. Certificado	Cadena alfanumérica		X
24	Nro. Certificado	Caracteres especiales		X
25	Nro. Certificado	Cadena numérica	X	

En la tabla 5.3, se representan las clases de equivalencia del componente de personas, el mismo se encarga de agregar registrar ciudadanos en el sistema del registro civil.

Tabla 5.3: Clases de equivalencia del componente de personas.(1/2)

Número	Campo	Clase de Equivalencia	Válido	Inválido
1	Nombres	Cadena vacía		X
2	Nombres	Cadena de caracteres	X	
3	Nombres	Cadena alfanumérica		X
4	Nombres	Caracteres especiales		X
5	Nombres	Cadena numérica		X
6	1er. Apellido	Cadena vacía		X
7	1er. Apellido	Cadena de caracteres	X	
8	1er. Apellido	Cadena alfanumérica		X
9	1er. Apellido	Caracteres especiales		X
10	1er. Apellido	Cadena numérica		X
11	2do. Apellido	Cadena vacía		X
12	2do. Apellido	Cadena de caracteres	X	
13	2do. Apellido	Cadena alfanumérica		X
14	2do. Apellido	Caracteres especiales		X
15	2do. Apellido	Cadena numérica		X
16	Dir. Urb./Res/...	Cadena vacía		X
17	Dir. Urb./Res/...	Cadena de caracteres	X	

Tabla 5.3: Clases de equivalencia del componente de personas.(2/2)

18	Dir. Urb./Res/...	Cadena alfanumérica		X
19	Dir. Urb./Res/...	Caracteres especiales		X
20	Dir. Urb./Res/...	Cadena numérica		X
21	Dir. Ave./Calle/...	Cadena vacía		X
22	Dir. Ave./Calle/...	Cadena de caracteres	X	
23	Dir. Ave./Calle/...	Cadena alfanumérica		X
24	Dir. Ave./Calle/...	Caracteres especiales		X
25	Dir. Ave./Calle/...	Cadena numérica		X
26	Nro. Casa/Apto...	Cadena vacía		X
27	Nro. Casa/Apto...	Cadena de caracteres	X	
28	Nro. Casa/Apto...	Cadena alfanumérica		X
29	Nro. Casa/Apto...	Caracteres especiales		X
30	Nro. Casa/Apto...	Cadena numérica		X
31	Teléfono	Cadena Vacía		X
32	Teléfono	Cadena de caracteres		X
33	Teléfono	Cadena alfanumérica		X
34	Teléfono	Caracteres especiales		X
35	Teléfono	Cadena numérica	X	
36	Email	Cadena vacía		X
37	Email	Cadena de caracteres	X	
38	Email	Cadena alfanumérica		X
39	Email	Caracteres especiales		X
40	Email	Cadena numérica		X

En las tabla 5.4, 5.5 y 5.6 se presentan algunos casos de prueba de caja negra, donde se incluirán datos que serán cotejados por las clases de equivalencia referenciadas anteriormente. Si se cumplen todas las clases involucra con dicho dato, entonces la salida será validada.

Tabla 5.4: Casos de prueba para el componente de solicitud de partidas.

Campo	Caso de Prueba	Salida	Clases Cubiertas
Año Registro	""	Inválido	6
Año Registro	!#qew123	Inválido	1, 2, 3, 5
Año Registro	que	Inválido	1
Año Registro	123	Inválido	5
Año Registro	12345	Inválido	5
Año Registro	1234	Válido	4
C.I. Principal	""	Inválido	11
C.I. Principal	!#qew123	Inválido	7, 8, 9
C.I. Principal	que	Inválido	7
C.I. Principal	123456	Válido	10
C.I. Secundaria	""	Válido	16
C.I. Secundaria	!#qew123	Inválido	12,13, 14
C.I. Secundaria	erewq	Inválido	12
C.I. Secundaria	123456	Válido	15

Tabla 5.5: Casos de prueba para el componente certificados de nacimiento.(1/2)

Campo	Caso de Prueba	Salida	Clases Cubiertas
Nombre Niño	""	Inválido	1
Nombre Niño	!#qew123	Inválido	2,3,4
Nombre Niño	qwer	Válido	2
Nombre Niño	123	Inválido	5
Hora Nacimiento	""	Inválido	6
Hora Nacimiento	!#qew123	Inválido	7,8,9
Hora Nacimiento	qwer	Inválido	2
Hora Nacimiento	123	Inválido	10
Hora Nacimiento	12:30	Válido	8
C.I. Responsable	""	Inválido	11
C.I. Responsable	!#qew123	Inválido	12, 13, 14
C.I. Responsable	qwer	Inválido	12

Tabla 5.5: Casos de prueba para el componente certificados de nacimiento. (2/2)

C.I. Responsable	123456	Válido	15
Nro. MPPS	""	Inválido	16
Nro. MPPS	!#qew123	Inválido	17, 18, 19
Nro. MPPS	qwer	Inválido	17
Nro. MPPS	123456	Válido	20
Nro. Certificado	""	Inválido	21
Nro. Certificado	!#qew123	Inválido	22, 23, 24
Nro. Certificado	qwer	Inválido	22
Nro. Certificado	123456	Válido	25

Tabla 5.6: Casos de prueba para el componente de personas. (1/2)

Campo	Caso de Prueba	Salida	Clases Cubiertas
Nombres	""	Inválido	1
Nombres	!#qew123	Inválido	2,3,4
Nombres	qwer	Válido	2
Nombres	123	Inválido	5
1er. Apellido	""	Inválido	6
1er. Apellido	!#qew123	Inválido	7,8,9
1er. Apellido	qwer	Válido	7
1er. Apellido	123	Inválido	10
2do. Apellido	""	Inválido	11
2do. Apellido	!#qew123	Inválido	12,13, 14
2do. Apellido	qwer	Válido	12
2do. Apellido	123	Inválido	15
Dir. Urb./Res/...	""	Inválido	16
Dir. Urb./Res/...	!#qew123	Inválido	17,18,19

Tabla 5.6: Casos de prueba para el componente de personas. (2/2)

Dir. Urb./Res/...	qwer	Válido	17
Dir. Urb./Res/...	123	Inválido	20
Dir. Ave./Calle/...	""	Inválido	21
Dir. Ave./Calle/...	!#qew123	Inválido	22,23,24
Dir. Ave./Calle/...	qwer	Válido	22
Dir. Ave./Calle/...	123	Inválido	25
Nro. Casa/Apto...	""	Inválido	26
Nro. Casa/Apto...	!#qew123	Inválido	27, 28, 29
Nro. Casa/Apto...	qwer	Válido	27
Nro. Casa/Apto...	123	Inválido	30
Teléfono	""	Inválido	31
Teléfono	!#qew123	Inválido	32, 33, 34
Teléfono	qwer	Inválido	32
Teléfono	123	Válido	35
Email	""	Inválido	36
Email	!#qew123	Inválido	37, 38, 39
Email	qwer	Válido	37
Email	123	Inválido	40

5.3.2 Pruebas de Integración

El objetivo general de las pruebas de integración es detectar las fallas de interacción entre las distintas clases que conforman al sistema. Debido a que cada clase examinada por separado se inserta de manera gradual dentro de la estructura, las pruebas de integración son realmente un mecanismo para verificar el correcto ensamblaje del sistema como un todo. Al efectuar la integración de los módulos, se concentra el esfuerzo en la

búsqueda de fallas que puedan generar excepciones arrojadas por los métodos; el empleo de operaciones equivocadas, e invocación incorrecta de los métodos.

Luego de verificar la calidad de los componentes, se procede a comprobar la eficiencia de un conjunto de componentes integrados por fase, estas se pueden observar distinguidas en la figura 5.8, en donde se muestra el diagrama de componentes por fase de integración del sistema.

Tabla 5.7: Leyenda de colores para fases del diagrama de componentes totales

Fase								
Color	1	2	3	4	5	6	7	8

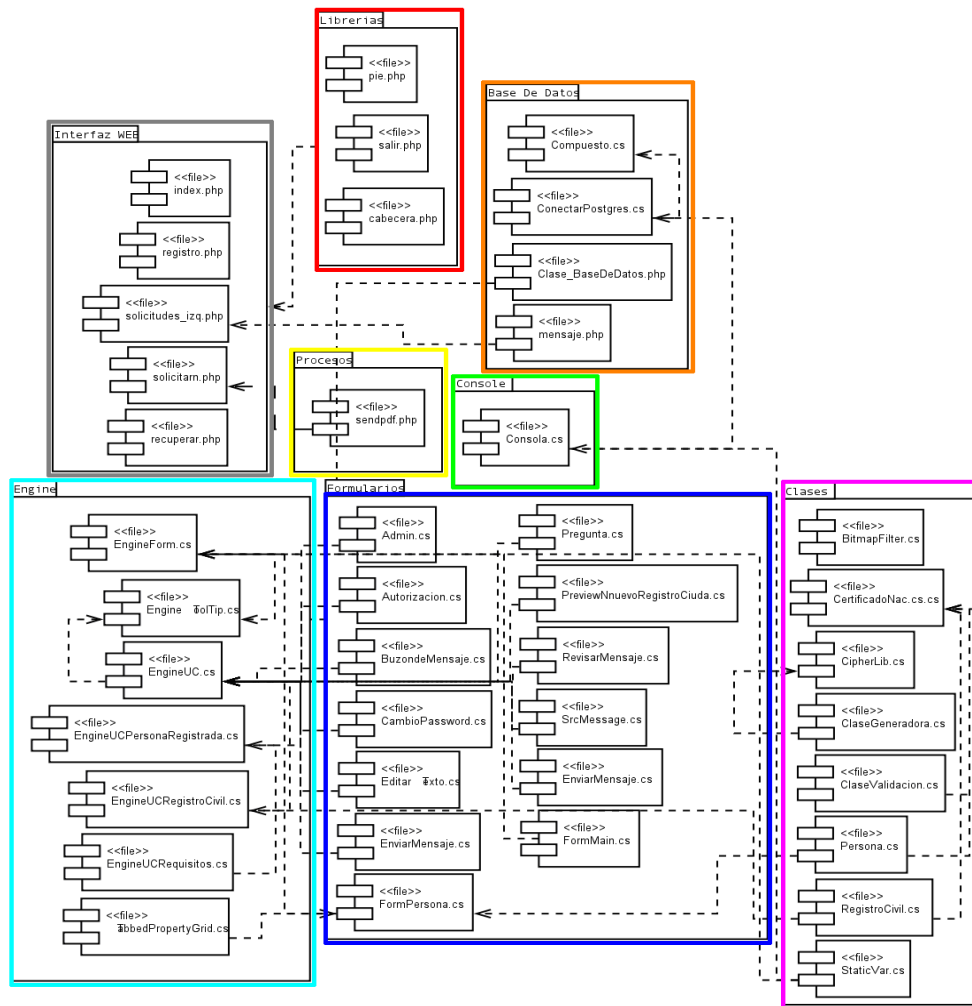


Figura 5.6: Diagrama de componentes por fase de integración.
Fuente: Elaboración propia

5.3.2.1 Integración de solicitud de partida de nacimiento

Este caso de prueba verifica la funcionalidad de la implementación del componente de solicitud de partida de nacimiento.

Tabla 5.8: Integración para el componente de solicitud de partidas de nacimiento.

Campo	Datos
Año Registro	1983
C.I. Principal	16478353
C.I. Secundaria	16123456

a) Resultados

No se hallaron errores en esta fase.

5.3.2.2 Integración de certificado de nacimiento

Este caso de prueba verifica la funcionalidad de la implementación del componente de solicitud de certificado de nacimiento.

Tabla 5.9: Integración para el componente de solicitud de certificado de nacimiento.

Campo	Datos
Nombre Niño	Luís Carlos
Hora Nacimiento	12:30
C.I. Responsable	12345678
Nro. MPPS	123
Nro. Certificado	123

a) Resultados

Se encontró necesaria la creación de una rutina para validar el campo Hora Nacimiento para evitar errores en su ingreso al sistema por parte de los usuarios.

No se hallaron más errores en esta fase.

5.3.2.3 Integración de personas

Este caso de prueba verifica la funcionalidad de la implementación del componente de personas.

Tabla 5.10: Integración para el componente de personas.

Campo	Datos
Nombres	Luís Carlos
1er. Apellido	12345678
2do. Apellido	12345678
Nro. MPPS	123
Nro. Certificado	123
Dir. Urb./Res/...	Residencia UDO
Dir. Ave./Calle/...	Calle 1
Nro. Casa/Apto...	Casa 1A
Teléfono	123456789
Email	test@udo.com

a) Resultados

Luego de esta corrección no se hallaron más errores en esta fase.

5.3.2.4 Procedimiento de prueba del sistema

Para la prueba de todo el sistema, se utilizara un caso que emule la realidad, y un entorno que represente la configuración física en la cual trabajará el sistema.

Se utilizaron 2 computadoras, una funciona como el servidor del sistema que gestiona tanto la base de datos como el servidor web, y otra que

funcionará como la máquina del sistema, desde donde accede el funcionario. Dichas máquinas se encuentran conectadas en red mediante el uso de un router.

a) Ejecutando la aplicación e identificándose

Ejecutar la aplicación mediante su acceso directo (Windows) o mediante su comando de ejecución (Linux) y agregar el nombre de usuario y contraseña válida para identificarse.



Figura 5.7: Ejecución del SRC en plataforma Windows XP

b) Agregando o actualizando un ciudadano en el sistema

- Acceder desde la interfaz principal al menú Gestión del usuario ► Nuevo registro de ciudadano para agregar una persona al registro civil.
- Llenar el campo “Cédula/Passport” con los datos del ciudadano.



Figura 5.8: Componente principal recibiendo la petición de actualización del ciudadano

- Presionar el botón “Actualizar Datos” el cual permite modificar los datos del ciudadano.
- Se activa la parte de la interfaz “Actualizando Ciudadano(a)” relativa al registro del ciudadano.
- Actualizar los campos necesarios.

- Presionar el botón “Actualizar Datos” ó “Cancelar de ser necesario”.

Sistema Registro Civil

Servicios del Registro | Gestión del Usuario | Configuración | Ayuda | Cerrar Sesión | Ventanas

GUANTÁ Actualizando Ciudadana(o)

Aviso Importante

- Con la cédula de identidad **Original y Legible** en mano certifique que los datos ingresados son los correctos.
- Los datos que no aparecen automáticamente en el menú desplegable podrán ser añadidos al sistema.
- Los datos que usted añada al sistema sea cuidadoso en agregar los respectivos acentos para evitar las deudas futuras.
- Para la carga de la imagen debe de tener una cámara Web instalada en el equipo la configuración puede variar según sea en sistema operativo que este utilizando **Windows** o **Linux**.

1. Datos Personales	
1.1 Nombres	Isaías
1.2 Primer Apellido	Medina
1.3 Segundo Apellido	Angrita
1.4 Sexo	M
1.5.0 Cédula de Identidad	1
1.5.1 Pasaporte N°	Ninguno
1.6 Nacionalidad	Venezolana
1.7 Estado Civil	Soltero
1.8 Profesión	Ninguno
1.9 Ocupación	Militar
2. Datos de su Nacimiento	
2.1 País	Venezuela
2.2 Entidad	Anzoátegui
2.3 Municipio	Guanta
2.4 Parroquia	Chorrerón
2.5 Localidad	Chorrerón
2.6 Fecha de Nacimiento	7/6/1983
3. Datos de Nivel Educativo	
3.1 Lee y Escribe	Sí

Cancelar | www.google.co.ve | www.cna.gov.ve | Actualizar Ciudadana(o) ✓

Copyright © 2010 - Alcaldía de Guanta
 Todos los Derechos Reservados.
[Términos de Uso](#) | Diseñado por Meq@Prog.C.A

Hoy 04/06/2010, Hora: 04:15 p.m.

Figura 5.9: Componente FormPersona recibiendo la actualización pertinente

c) Cambiando el color de la aplicación

- Acceder desde la interfaz principal al menú Gestión del usuario ► Perfil del usuario.
- Dirigirse a la casilla color de esquema y selecciones el color de su agrado.
- Presionar el botón “Actualizar Datos”.
- Cerrar la ventana en la esquina superior derecha presionando la “X”.

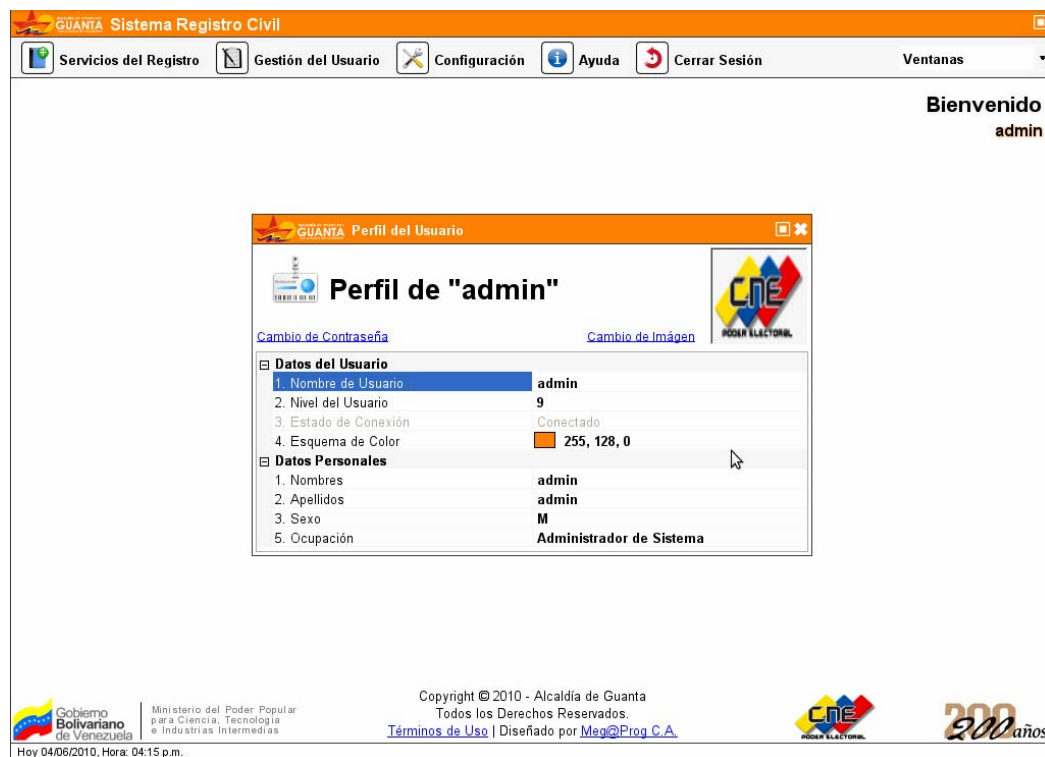


Figura 5.10: Componente PerfilUsuario recibiendo la actualización pertinente

d) Realizando una partida de nacimiento ordinaria (ambos padres)

- Acceder desde la interfaz principal al menú Servicios del Registro
 - Declaraciones ► Nacimientos ► Ordinario (Ambos Padres).

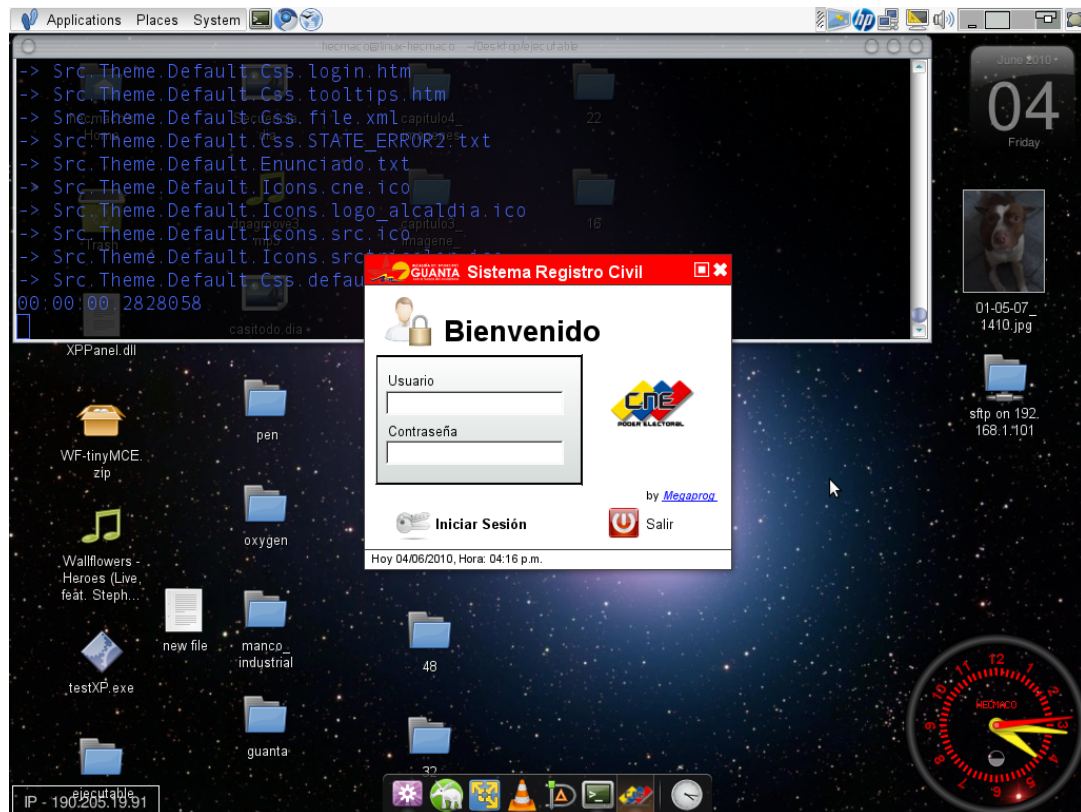


Figura 5.11: Componente PerfilUsuario recibiendo la actualización pertinente

- Establecer los requisitos y continuar si son los correctos, presionar el botón “Siguiente”.

Applications Places System

GUANTÁ Sistema Registro Civil

Servicios del Registro Gestión del Usuario Configuración Ayuda Cerrar Sesión Ventanas

GUANTÁ Declaración para un Acta de Nacimiento Ordinaria (Ambos Padres)

Información de Requisitos

Para la elaboración del siguiente documento se requiere lo siguiente:

1. Certificado de Nacimiento Original y con todos sus datos llenos, marcas de los pies del niño/a , médico tratante, entre otros.
2. Cédula de Identidad de ambos padres (**Original y Copia**).
3. Dós (2) Testigos ambos con su Cédula de Identidad (**Original y Copia**).
4. Los testigos deben estar residienciados en esta Parroquia.

Nota: si los padres del niño/a son casados favor de presentar Acta de Matrimonio (**Original y Copia**).

[Imprimir](#)

Cumple con todos los requisitos Solicitados

Cancelar Atras Siguiente

Copyright © 2010 - Alcaldía de Guanta
 Todos los Derechos Reservados.
[Términos de Uso](#) | Diseñado por [Meg@Prog C.A.](#)

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para Ciencia, Tecnología e Industrias Intermedias

CNE PODER ELECTORAL

200 años

Hoy 04/06/2010, Hora: 04:17 p.m.

Figura 5.12: Componentes ActaNac2Padres, EngineUCRequisitos

- Establecer el orden de redacción del acta, presionar el botón “Siguiente”.

Applications Places System

GUANTÁ Sistema Registro Civil

Servicios del Registro Gestión del Usuario Configuración Ayuda Cerrar Sesión Ventanas

GUANTÁ Declaración para un Acta de Nacimiento Ordinaria (Ambos Padres)

Pregunta

¿ Para la redacción del documento quién presentará al niño/a ?

Nota: Esto afectará la redacción del acta de Nacimiento.

Presenta el Padre Presenta la Madre

¿Solo uno de los declarantes está presente?

Cancelar Atras Siguiente

Copyright © 2010 - Alcaldía de Guanta
Todos los Derechos Reservados.
[Términos de Uso](#) | Diseñado por [Meg@Prog.C.A.](#)

Gobierno Bolivariano de Venezuela Ministerio del Poder Popular para Ciencia, Tecnología e Industrias Intermedias

CNE Poder Electoral

200 años

Hoy 04/06/2010, Hora: 04:17 p.m.

Figura 5.13: Componentes ActaNac2Padres, QuienPresenta

- Llenar el campo “Cédula/Passport” para los datos del padre, presionar el botón “Siguiente”.

GUANTÁ Sistema Registro Civil

Servicios del Registro | Gestión del Usuario | Configuración | Ayuda | Cerrar Sesión | Ventanas

GUANTÁ Declaración para un Acta de Nacimiento Ordinaria (Ambos Padres)

Datos del Padre

[registrar nuevo ciudadano](#)

1. Datos Personales	
1.1 Nombres	Isaías
1.2 Primer Apellido	Medina
1.3 Segundo Apellido	Anzarita
1.4 Sexo	M
1.5.0 Cédula de Identidad	1
1.5.1 Pasaporte	Ninguno
1.6 Nacionalidad	Venezolana
1.7 Estado Civil	Soltero
1.8 Profesión	Ninguno
1.9 Ocupación	Militar
2. Datos de su Nacimiento	
2.1 País	Venezuela
2.2 Entidad	Anzoátegui
2.3 Municipio	Guanta
2.4 Parroquia	Chorrerón
2.5 Localidad	Chorrerón
2.6 Fecha de Nacimiento	7/6/1983
3. Datos de Nivel Educativo	
3.1 Lee y Escribe	Sí

Copyright © 2010 - Alcaldía de Guanta
 Todos los Derechos Reservados.
[Términos de Uso](#) | Diseñado por [Meg@Prog C.A.](#)

Hoy 04/06/2010, Hora: 04:17 p.m.

Figura 5.14: Componentes ActaNac2Padres, FormPersona, solicitando datos del padre

- Llenar el campo “Cédula/Passport” para los datos de la madre, presionar el botón “Siguiente”.

Applications Places System

GUANTÁ Sistema Registro Civil

Servicios del Registro Gestión del Usuario Configuración Ayuda Cerrar Sesión Ventanas

GUANTÁ Declaración para un Acta de Nacimiento Ordinaria (Ambos Padres)

Datos del Primer Testigo

[registrar nuevo ciudadano](#)

1. Datos Personales	
1.1 Nombres	Hector Manuel
1.2 Primer Apellido	Cova
1.3 Segundo Apellido	Rosario
1.4 Sexo	M
1.5.0 Cédula de Identidad	16562748
1.5.1 Pasaporte	Ninguno
1.6 Nacionalidad	Venezolana
1.7 Estado Civil	Soltero
1.8 Profesión	Ninguno
1.9 Ocupación	Programador
2. Datos de su Nacimiento	
2.1 País	Venezuela
2.2 Entidad	Anzoátegui
2.3 Municipio	Juan Antonio Sotillo
2.4 Parroquia	Puerto La Cruz
2.5 Localidad	Puerto La Cruz
2.6 Fecha de Nacimiento	5/7/1983
3. Datos de Nivel Educativo	
3.1 Lee y Escribe	Sí

Copyright © 2010 - Alcaldía de Guanta
 Todos los Derechos Reservados.
[Términos de Uso](#) | Diseñado por [Meg@Prog C.A.](#)

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para Ciencia, Tecnología e Industrias Intermedias | CNE | 200 años

Hoy 04/06/2010, Hora: 04:18 p.m.

Figura 5.16: Componentes ActaNac2Padres, FormPersona, solicitando datos del primer testigo

- Llenar el campo “Cédula/Passport” para los datos del segundo testigo, presionar el botón “Siguiente”.

GUANTÁ Sistema Registro Civil

Servicios del Registro | Gestión del Usuario | Configuración | Ayuda | Cerrar Sesión | Ventanas

GUANTÁ Actualizando Ciudadana(o)

Aviso importante

- Con la cédula de identidad **Original y Legible** en mano certifique que los datos ingresados son los correctos.
- Los datos que no aparecen automáticamente en el menú desplegable podrán ser añadidos al sistema.
- Los datos que usted añada al sistema sea cuidadoso en agregar los respectivos acentos para evitar las deputaciones futuras.
- Para la carga de la imagen debe de tener una cámara Web instalada en el equipo la configuración puede variar según sea en sistema operativo que este utilizando **Windows** ® ó **Linux**.

1. Datos Personales

1.1 Nombres	Rafael
1.2 Primer Apellido	Solorzano
1.3 Segundo Apellido	Robles
1.4 Sexo	M
1.5.0 Cédula de Identidad	16478353
1.5.1 Pasaporte N°	Ninguno
1.6 Nacionalidad	Venezolana
1.7 Estado Civil	Soltero
1.8 Profesión	Ninguno
1.9 Ocupación	Programador

2. Datos de su Nacimiento

2.1 País	Venezuela
2.2 Entidad	Anzoátegui
2.3 Municipio	Juan Antonio Sotillo
2.4 Parroquia	Pozuelos
2.5 Localidad	Pozuelos
2.6 Fecha de Nacimiento	8/27/1983

3. Datos de Nivel Educativo

3.1 Lee y Escribe	Sí
-------------------	----

Cancelar | www.google.co.ve | www.cns.gov.ve | Actualizar Ciudadana(o) ✓

Copyright © 2010 - Alcaldía de Guanta
 Todos los Derechos Reservados.
[Términos de Uso](#) | Diseñado por [Meg@Prog C.A.](#)

Hoy 04/06/2010, Hora: 04:26 p.m.

Figura 5.17: Componentes ActaNac2Padres, FormPersona, solicitando datos del segundo testigo

- Contestar la pregunta si el niño(a) es presentado después de los 90 días de nacido establecido por la ley, presionar la sección “No”, presionar el botón “Siguiente”.

The screenshot shows a web browser window with the title 'GUANTÁ Sistema Registro Civil'. The browser's address bar and tabs are visible at the top. The application's navigation menu includes 'Servicios del Registro', 'Gestión del Usuario', 'Configuración', 'Ayuda', and 'Cerrar Sesión'. The main content area is titled 'GUANTÁ Declaración para un Acta de Nacimiento Ordinaria (Ambos Padres)'. The form section is titled 'Datos de la Defensoria' and contains the following elements:

- A question: '¿El niño se Presenta después de los 90 días del nacimiento?' with radio buttons for 'Si' and 'No'.
- A dropdown menu labeled 'Lugar del Centro:' with the text 'Seleccione una Sede...'.
- A text input field labeled 'Identificación del Oficio:' with the placeholder text 'Escriba el Número de Oficio.'.

At the bottom of the form, there are navigation buttons: 'Cancelar' (with a red X icon), 'Atras' (with a left arrow), and 'Siguiente' (with a right arrow). The footer of the page includes the logos for 'Gobierno Bolivariano de Venezuela' and 'Ministerio del Poder Popular para Ciencia, Tecnología e Industrias Intermedias', along with copyright information: 'Copyright © 2010 - Alcaldía de Guanta Todos los Derechos Reservados.' and 'Diseñado por Meg@Prog C.A.'. There are also logos for 'CNE' and '200 años'.

Figura 5.18: Componentes ActaNac2Padres, Defensoria, solicitando datos del nacimiento

- Contestar la pregunta si posee un certificado de nacimiento original expedido por centro hospitalario, presionar la sección “Si”, rellenar todos los campos de acuerdo con el certificado, presionar el botón “Siguiente”.

Applications Places System

GUANTÁ Sistema Registro Civil

Servicios del Registro | Gestión del Usuario | Configuración | Ayuda | Cerrar Sesión | Ventanas

GUANTÁ Declaración para un Acta de Nacimiento Ordinaria (Ambos Padres)

Datos del Certificado de Nacimiento

¿ Poseen un Certificado de nacimiento? Si No

Nota: es necesario rellenar todos los campos.

* Datos de la Defensoría	
Lugar de la Defensoría	No requerido
N° de Oficio enmanado	No requerido
1. Lugar de Ocurrencia	
1.1 País	Venezuela
1.2 Entidad	Anzoátegui
1.3 Municipio	Guanta
1.4 Parroquia	Chorrerón
1.5 Localidad	Chorrerón
1.6 Centro Hospitalario	El Hospital Doctor Cesar Rodríguez Rodríguez
2. Datos del niño/niña	
2.1 Nombre(s) del niño(a)	JOSE ALEJANDRO
2.2 Fecha de Nacimiento	6/11/2010
2.3 Hora de Nacimiento	12:59
2.4 Hora Meridiam	A.M.
2.5 Sexo	M
3. Responsable Certificación Médico(a) / Partero(a)	
3.1 Cedula / Pasaporte	123456789
3.2 N° MPPS	123456789
4. Datos del Nacimiento	

Cancelar Atras Siguiente

Copyright © 2010 - Alcaldía de Guanta
 Todos los Derechos Reservados.
[Términos de Uso](#) | Diseñado por [Meg@Prog C.A.](#)

Gobierno Bolivariano de Venezuela | Ministerio del Poder Popular para Ciencia, Tecnología e Industrias Intermedias | CNE Poder Electoral | 200 años

Hoy 04/06/2010, Hora: 04:28 p.m.

Figura 5.19: Componentes ActaNac2Padres, Certificado, solicitando datos del certificado de nacimiento

- Revisar los datos proporcionados por el sistema sobre el acta, tomo, folio, año. Registrador(a) civil, escribiente, así podrá revisar el acta de nacimiento como la copia certificada antes de su aprobación y registro en formato pdf.

Applications Places System

GUANTA Sistema Registro Civil

Servicios del Registro Gestión del Usuario Configuración Ayuda Cerrar Sesión Ventanas

GUANTA Declaración para un Acta de Nacimiento Ordinaria (Ambos Padres)

Información del Registro Civil

Municipio Guanta
Parroquia Chorrerón

Registradora Civil: Abogado. Maria Elena Gonzalez Martinez
Resolucion N°: 123456
Gaceta N°: 404, Ext. de fecha trece (13) de Enero del año dos mil nueve (2009)
Escribiente: admin admin
Días de Independencia: 199°
Días de la Federación: 151°
Acta N°: 005
Folio: 005
Libro/Tomo: Segundo
Año: 2010

Copia Certificada
Acta

acepto que todos los datos proporcionados acá son correctos para la elaboración del Acta de Nacimiento.

Cancelar Atras Siguiente

Copyright © 2010 - Alcaldía de Guanta
 Todos los Derechos Reservados.
[Términos de Uso](#) | Diseñado por [Meg@Prog C.A.](#)

Gobierno Bolivariano de Venezuela Ministerio del Poder Popular para Ciencia, Tecnología e Industrias Intermedias CNE Poder Electoral 200 años

Hoy 04/06/2010, Hora: 04:28 p.m.

Figura 5.20: Componentes ActaNac2Padres, EngineUCRegistroCivil, mostrando los datos elaborados

- Seleccionar la casilla para aceptar que todos los datos proporcionados son los correctos, presionar el botón “Finalizar”.

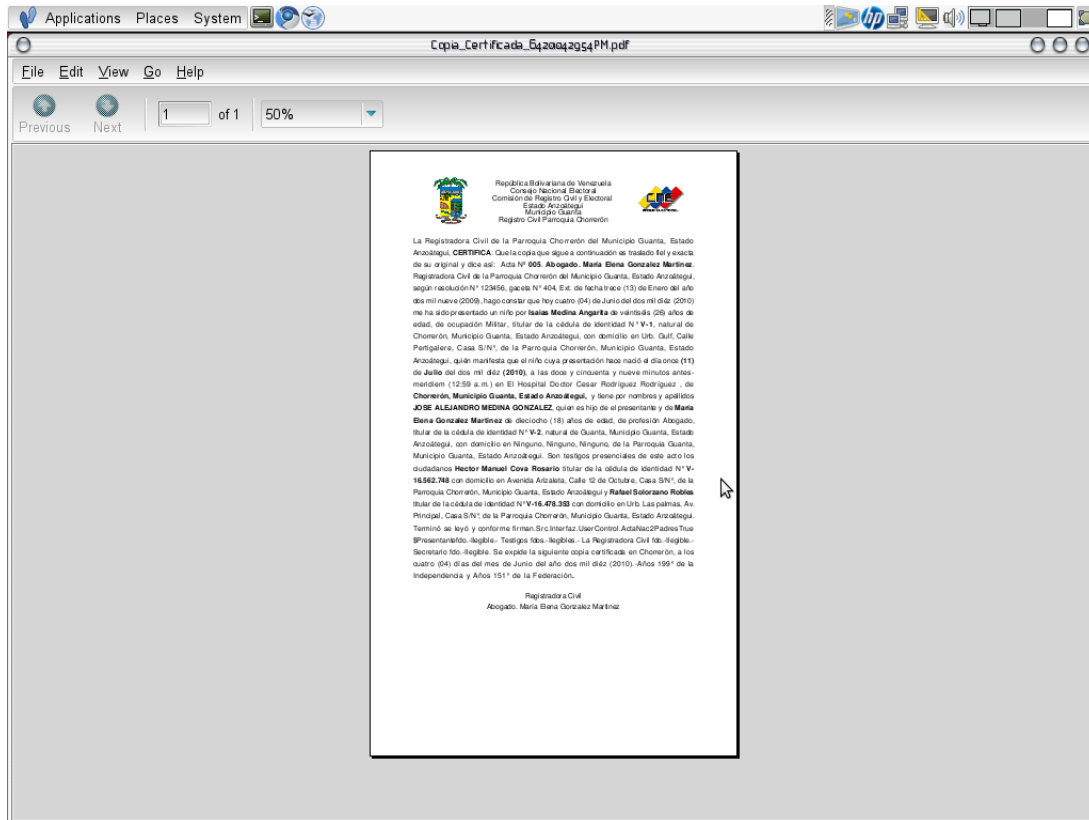


Figura 5.22: Programa evince mostrando la copia certificada del acta de nacimiento en formato pdf.

e) Ingresando en la página web.

Ingresar en el URL de la página web mediante cualquier navegador.

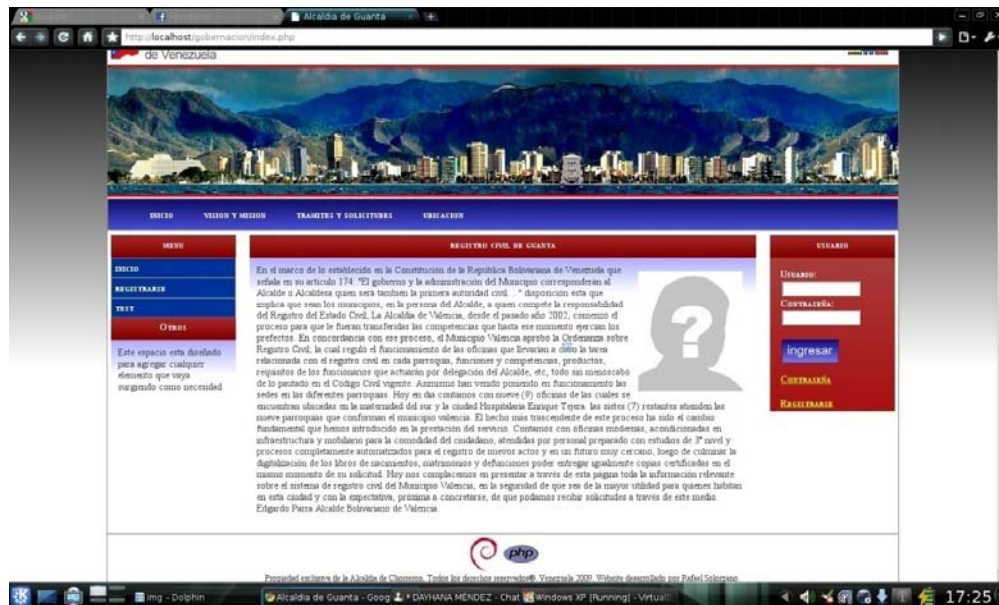


Figura 5.23: Página principal del portal web.

- Ingresar en la sección de registro de usuario.
- Llenar los campos con la información pertinente.

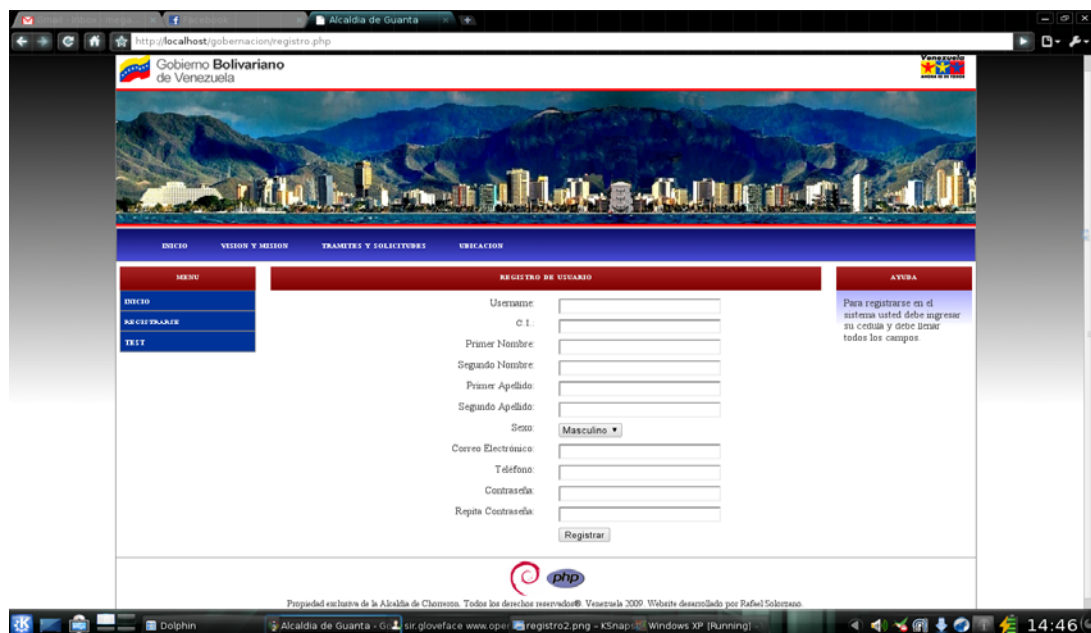


Figura 5.24: Pagina de registro.

- Finalizado el registro, volver a la página principal.
- Identificarse en el portal web.

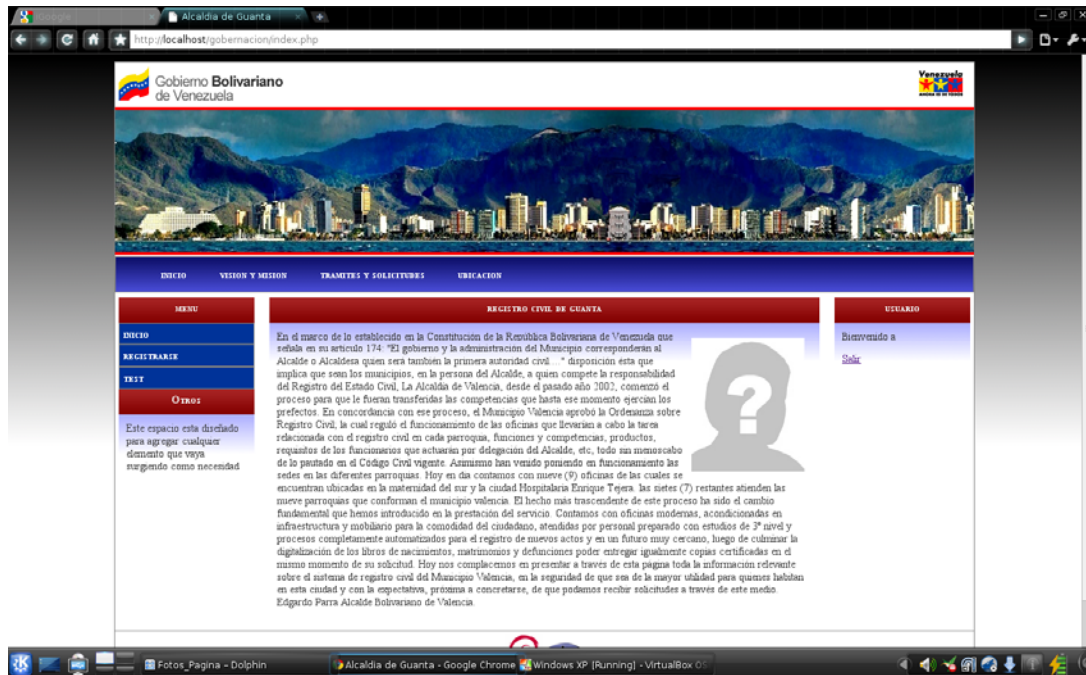


Figura 5.25: Pagina principal con usuario identificado.

f) Enviando una comunicación a un funcionario.

- Ingresar en la página de Trámites y Solicitudes.
- Seleccionar Mensajes en el sub-menu.
- Seleccionar uno de los funcionarios disponibles en el momento.
- Redactar el comunicado.
- Enviar el comunicado.

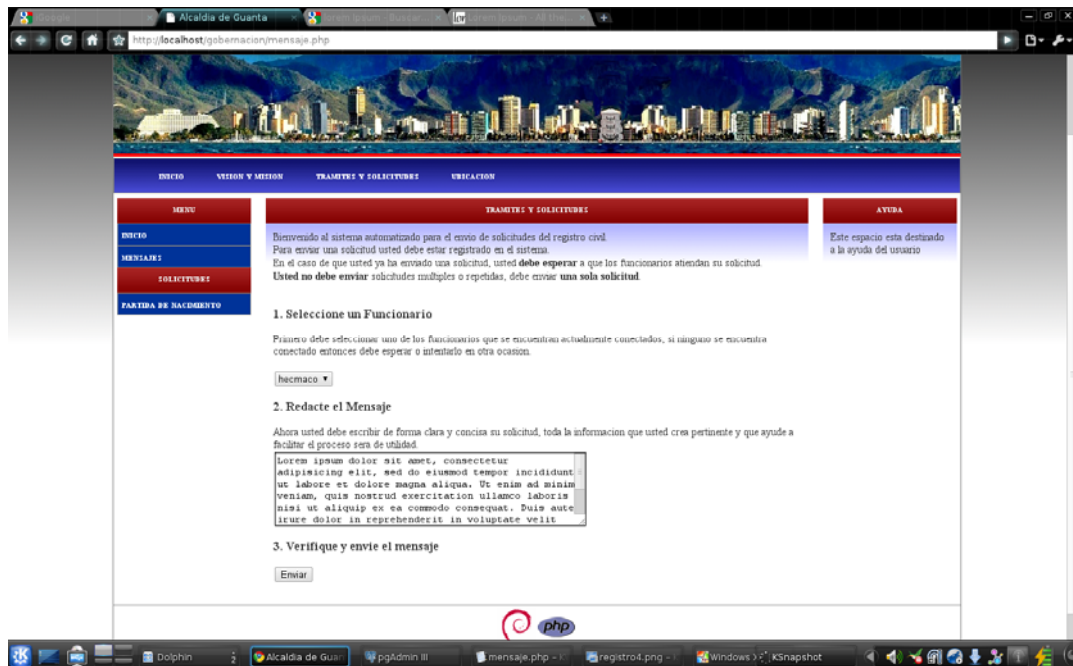


Figura 5.26: Pagina para envió de comunicado.

g) Solicitando partida de nacimiento.

- Ingresar en la página de Trámites y Solicitudes.
- Seleccionar Partida de Nacimiento en el sub-menu.
- Llenar el campo de año, con el año en el que se registro el solicitador.
- Llenar al menos uno de los campos de cedula, con los datos de los representantes legales del solicitador.

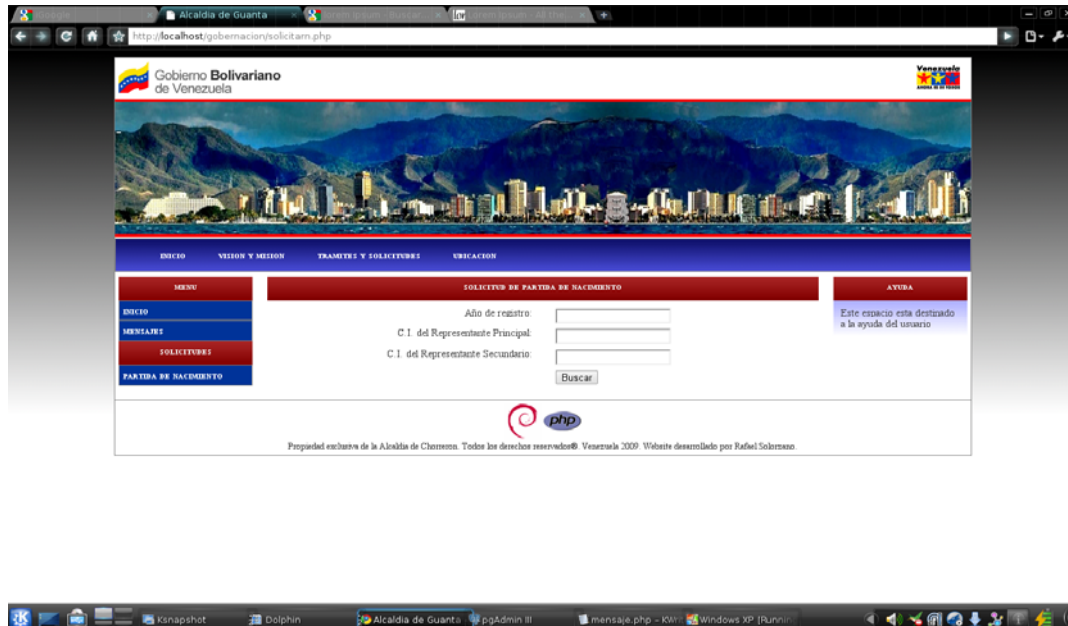


Figura 5.27: Pagina para solicitud de partida de nacimiento.

- En caso de existir coincidencias con los datos proporcionados, proceder a descargar los documentos requeridos.

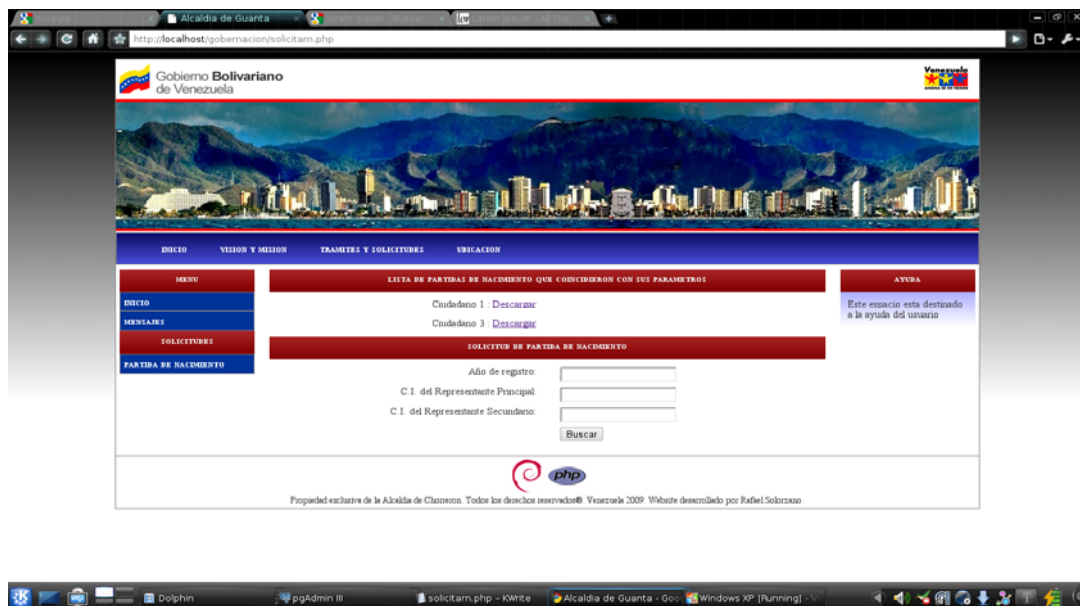


Figura 5.28: Pagina descarga de documentos solicitados.

CAPÍTULO VI: FASE DE TRANSICIÓN

6.1 Introducción

La fase de transición figura 6.1 tiene como objetivo colocar el producto en manos de los usuarios finales, durante la cual se pueden desarrollar nuevas versiones mejoradas del producto, completar la documentación mediante la elaboración del manual de usuario, adiestrar al usuario en el manejo del producto, y en general se realizan tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.



Figura 6.1 Diagrama de flujo de trabajo de las fases del proceso unificado, identificación de la fase de transición.

6.2 Mejoras Realizadas al Sistema

Durante esta fase se realizaron mejoras en módulos específicos del sistema, los cuales se mencionan a continuación:

- Se realizó la validación de la longitud máxima del campo de nombre del archivo de respaldo de la base de datos, para evitar errores con el sistema operativo al realizar la creación y almacenamiento del archivo.
- Se creó una rutina que evita que se sobrescriban archivos del sistema operativo al momento de guardar los archivos de reportes PDF, HTML y Backup de base de datos.
- Depuración de la carpeta ~/tmp para eliminar documentos generados por el sistema y evitar así el consumo del espacio del disco.

6.3 Manual de Usuario

Como último producto se desarrolló el documento que servirá como base principal para los usuarios del sistema, para su correcto uso. El manual de usuario se muestra en la sección de anexos, marcado como Anexo. El manual de usuario está dirigido a los usuarios finales del sistema, que serán los funcionarios del registro civil.

El manual es elaborado de la siguiente manera: Se comenzó describiendo los diversos componentes que conforman la interfaz gráfica de inicio del sistema: Pantalla de inicio, Menú de actividades del registro, Gestión del usuario, Ayuda, de manera tal que los usuarios se familiaricen con los distintos componentes que la conforman. Seguidamente se procedió a describir detalladamente cada uno de los componentes de la interfaz que permiten la realización de todas las actividades requeridas por el sistema. Finalmente se incluyen los requisitos especiales de instalación del sistema,

que son necesarios para el correcto funcionamiento del software y que, de igual manera, garantizar el correcto enlace con la base de datos.

CONCLUSIONES

Con este trabajo de investigación se ha creado una herramienta que facilitara la gestión de los ciudadanos del registro civil y sus empleados. Todo el sistema lleva un control de los ciudadanos del municipio, sus estados civiles, documentos legales, parentescos y trámites legales, además de brindar beneficios mediante el portal web, como solicitud de documentos y comunicación bilateral con el registro civil. Lo cual garantiza un crecimiento tecnológico y un impulso en el desarrollo de la automatización del país.

Este proyecto fue realizado con el objetivo fundamental de optimizar el funcionamiento del registro civil, para facilitar así su uso por parte de los empleados y ciudadanos que hacen vida en el municipio o alcaldía.

El RUP, (Rational Unified Process), como metodología elegida, aunado al uso de UML como instrumento principal para la documentación y guía, permitieron la organización y perfeccionamiento del sistema, efectuando todas las etapas de este proceso y ayudó a la prevención de fallas que se pudieron presentar, evitando así un replanteamiento del proyecto, canalizando los diferentes flujos de trabajo necesarios por cada una de las fases de desarrollo.

Durante la fase de inicio se determinaron los requisitos específicos necesarios para la construcción del sistema y posteriormente fueron actualizados en la fase de elaboración, derivando esto en el punto de partida para la definición de una completa funcionalidad.

La implementación de la base de datos, diseñada utilizando el modelo relacional, permite el almacenamiento y recuperación de los datos de manera sencilla y eficiente.

El uso de Java, PHP y PostgreSQL, como lenguajes de programación y Sistema Manejador de Base de Datos respectivamente, permitieron el desarrollo del sistema de manera clara y efectiva, debido a que estas tecnologías son fácilmente utilizables en cualquier entorno computacional, puesto a que sus requerimientos para el funcionamiento son mínimos y al ser tecnologías de Software Libre se pueden obtener en la Web.

Se realizó de manera exitosa la integración, pruebas y documentación del funcionamiento del sistema, se evaluó y depuró el sistema en varias iteraciones, obteniendo un software altamente funcional, al que puede realizarse mantenimiento y actualizaciones.

RECOMENDACIONES

- Integrar al sistema los módulos de matrimonios en todas sus modalidades.
- Integrar al sistema los módulos de defunciones en todas sus modalidades.
- Integrar al sistema los módulos de estadísticas y reportes.
- Mantener el sistema actualizado con las actas y verificar el llenado manual según sea el procedimiento que emana del manual de normas y procedimientos del registro civil.
- Utilizar una cámara Web, para garantizar así la correcta inscripción y actualización del ciudadano dentro del registro civil.
- Integrar al sistema un módulo de firma digital para la entrega de los documentos en línea.
- Apegarse a las rutinas de mantenimiento establecidas en los manuales de usuario.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

TÍTULO	“DESARROLLO DE UN SISTEMA PARA LA AUTOMATIZACIÓN DE LA GESTIÓN DE LAS PARTIDAS DE NACIMIENTO DE UN REGISTRO CIVIL VENEZOLANO BAJO LICENCIA DE SOFTWARE LIBRE”
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Cova, Héctor	CVLAC: 16.562.748 E MAIL: hecmaco@hotmail.com
Solórzano, Rafael	CVLAC: 16.478.353 E MAIL: megaprog@gmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

Software Libre

Registro Civil

Proceso Unificado de Desarrollo de Software

Lenguaje Unificado de Modelado UML 2.0

Lenguaje C#

Lenguaje PHP

PostgreSQL

Automatización de la Gestión de Partidas de Nacimiento

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

ÁREA	SUBÁREA
Ingeniería y Ciencias Aplicadas	Ingeniería en Computación

RESUMEN (ABSTRACT):

Los registros civiles venezolanos son los entes gubernamentales establecidos para mantener las inscripciones de los hechos y actos vitales de las personas que modifican su estado civil, su relación familiar y su identificación personal. Por lo que dichos registros requieren de numerosos procesos de gestión con un cierto grado de complejidad. De esta manera inmediatamente surge la necesidad del desarrollo de un sistema para automatizar los procesos y procedimientos de los registros civiles venezolanos. En este trabajo se propone la automatización de las actividades asociadas a las partidas de nacimiento en dichos registros. Como lugar para desarrollar y probar el sistema, se escogió la alcaldía de Guanta, debido a que sufre la necesidad de una automatización de su registro civil. El sistema propuesto brindaría soluciones al registro y gestión de ciudadanos, gestión de partidas de nacimiento y al control de los usuarios, todo esto dividido entre una aplicación de escritorio para los funcionarios, una página web para todas las personas y una base de datos centrada en el ciudadano que almacena toda la información pertinente a la elaboración de los documentos del registro civil. El diseño y desarrollo del proyecto se realizara según las fases de Proceso Unificado de Desarrollo de software y utilizando el Lenguaje Unificado de Modelado UML 2.0. El sistema se construirá e implementara utilizando los lenguajes C# y PHP, y el gestor de bases de datos PostgreSQL.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Bastardo, José	ROL	CA	AS	TU X	JU
	CVLAC:	6.890.832			
	E_MAIL	josebastardo@gmail.com			
	E_MAIL				
Cortinez, Claudio	ROL	CA	AS	TU	JU X
	CVLAC:	12.155.334			
	E_MAIL	cl_cortinez@cantv.net			
	E_MAIL				
Carrasquero, Manuel	ROL	CA	AS	TU	JU X
	CVLAC:	7.374.987			
	E_MAIL	Manuelscm@gmail.com			
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2010	08	02
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
TESIS.AutomatizacionPartidasNacimiento.doc	Application/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H I J K
L M N O P Q R S T U V W X Y Z . a b c d e f g h i j k l m n o p q r s t u v w x y z . 0 1
2 3 4 5 6 7 8 9 .

ALCANCE

ESPACIAL: _____ (OPCIONAL)

TEMPORAL: _____ (OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas

INSTITUCIÓN:

Universidad de Oriente/ Núcleo de Anzoátegui

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO

DERECHOS

Artículo 41: “Los Trabajos de Grado son exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario, para su autorización”.

Cova Héctor
AUTOR

Solórzano Rafael
AUTOR

Bastardo José
TUTOR

Cortinez Claudio
JURADO

Carrasquero Manuel
JURADO

Luis F. Rojas
POR LA SUBCOMISIÓN DE TESIS