

UNIVERSIDAD NACIONAL DEL ALTIPLANO-PUNO

FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



**DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL
REGISTRO CIVIL DE LA MUNICIPALIDAD DISTRITAL DE
CABANA – 2012**

TESIS

PRESENTADA POR:

Bach. VILMA ESCOBEDO CHOQUE

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO

PUNO – PERÚ

2012



UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



TESIS

DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL REGISTRO CIVIL DE LA MUNICIPALIDAD DISTRITAL DE CABANA – 2012

PRESENTADA POR:

Bach. VILMA ESCOBEDO CHOQUE



A la Coordinación de Investigación de la Facultad de Ingeniería Estadística e Informática de la Universidad Nacional del Altiplano, para optar el Título Profesional de:

INGENIERO ESTADÍSTICO E INFORMÁTICO

APROBADA POR:

PRESIDENTE : 
 M.Sc. ALEJANDRO APAZA TARQUI

PRIMER MIEMBRO : 
 ING. CHARLES I. MENDOZA MOLLOCONDO

SEGUNDO MIEMBRO : _____
 M.Sc. FRANCISCO CURRO PÉREZ

DIRECTOR : 
 M.Sc. CONFESOR M. VARGAS VELARDE

ASESOR : 
 M.Sc. LEONEL COYLA IDME

Área: Informática
 Tema: Sistemas de información **2012**
 Fecha de sustentación: 20/12/2012

DEDICATORIAS

En primer lugar quiero agradecer a Dios Todo Poderoso, por brindarme la oportunidad de vivir y por darme salud y las fuerzas necesarias a lo largo de toda mi trayectoria. Tus bendiciones serán siempre recibidas Diosito. ¡Gracias Señor por llenar mi vida de tantas bendiciones y todo esto te lo debo a ti!. Gracias siempre.

Gran motivo me inspira a dedicar y agradecer a todos quienes me ayudaron:

A mi madre, a mi padre y a mis hermanos, con quienes cuento para todos mis logros y que siempre depositan en mi todo el amor para hacer más placentero y hermoso el camino.

VILMAESCH.

AGRADECIMIENTOS

A mi alma Mater Universidad Nacional del Altiplano de Puno, en forma especial a la Escuela Profesional de Ingeniería Estadística e Informática, por abrirme sus puertas y brindarme la oportunidad de cursar mis estudios superiores y a todos los profesores que me brindaron parte de sus conocimientos durante esta etapa ayudando a formarme como profesional. Mil gracias.

VILMA.

ÍNDICE

Pág.

RESUMEN

CAPÍTULO I

PLAN DE INVESTIGACIÓN

| | |
|---|----------|
| 1.1. EL PROBLEMA | 1 |
| 1.2. JUSTIFICACIÓN DE LA INVESTIGACIÓN | 3 |
| 1.3. OBJETIVOS | 3 |
| 1.3.1. OBJETIVO GENERAL | 3 |
| 1.3.2. OBJETIVOS ESPECÍFICOS | 4 |
| 1.4. HIPÓTESIS | 4 |

CAPÍTULO II

MARCO TEÓRICO

| | |
|---|-----------|
| 2.1. ANTECEDENTES DE LA INVESTIGACIÓN | 5 |
| 2.2. BASE TEÓRICA | 7 |
| 2.2.1. SISTEMA DE INFORMACIÓN | 7 |
| 2.2.2. TIPOS DE SISTEMA DE INFORMACIÓN | 9 |
| 2.2.3. METODOLOGÍA ÁGIL | 10 |
| 2.2.4. METODOLOGÍA APLICADA XP (XTREME PROGRAMMING)... | 11 |
| 2.2.5. LENGUAJE DEL MODELADO UNIFICADO (UML) | 12 |
| 2.2.6. ARQUITECTURA CLIENTE – SERVIDOR | 15 |
| 2.2.7. HERRAMIENTAS DE DESARROLLO | 16 |
| 2.2.8. APPSERV | 18 |
| 2.2.9. LENGUAJE DE PROGRAMACIÓN | 18 |
| 2.2.10. SISTEMA DE ADMINISTRACIÓN DE BASE DE DATOS..... | 21 |
| 2.2.11. WEB APACHE | 21 |
| 2.2.12. INTERFAZ DE USUARIO | 22 |
| 2.2.13. HARDWARE | 23 |
| 2.2.14. ESTÁNDARES DE CALIDAD | 23 |
| 2.3. DEFINICIÓN DE TÉRMINOS BÁSICOS..... | 24 |
| 2.3.1. DESARROLLO DE SOFTWARE | 24 |
| 2.3.2. SERVIDOR | 24 |
| 2.3.3. CÓDIGO FUENTE | 24 |
| 2.3.4. DIGITAR | 25 |
| 2.3.5. SERVIDOR | 25 |
| 2.3.6. SERVIDOR WEB | 25 |
| 2.3.7. OPTIMIZAR..... | 25 |

| | | |
|---------|--|----|
| 2.3.8. | REPORTE | 25 |
| 2.3.9. | GESTIÓN | 25 |
| 2.3.10. | REGISTRO CIVIL | 26 |
| 2.3.11. | REGISTRADOR | 26 |
| 2.3.12. | DIGITAR | 26 |
| 2.3.13. | ACTAS DE NACIMIENTO | 26 |
| 2.3.14. | ACTAS DE MATRIMONIO | 26 |
| 2.3.15. | ACTA DE DEFUNCIÓN | 26 |
| 2.3.16. | REGISTRO NACIONAL DE IDENTIFICACIÓN Y ESTADO CIVIL (RENIEC) | 27 |

CAPÍTULO III

MATERIALES Y MÉTODOS

| | | |
|-------------|--|-----------|
| 3.1. | LOCALIZACIÓN | 28 |
| 3.2. | POBLACIÓN | 28 |
| 3.3. | RECOPIACIÓN DE LA INFORMACIÓN | 28 |
| 3.4. | METODOLOGÍA DE INVESTIGACIÓN | 29 |
| 3.4.1. | METODOLOGÍA XTREMME PROGRAMMING XP | 29 |

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

| | | |
|-------------|--|-----------|
| 4.1. | PROCESO DEL DESARROLLO DEL SOFTWARE | 30 |
| 4.1.1. | ANÁLISIS DE SISTEMA | 30 |
| 4.1.2. | REQUERIMIENTOS FUNCIONALES | 33 |
| 4.1.3. | DISEÑO DEL SISTEMA | 35 |
| 4.1.3.1. | DIAGRAMAS DEL UML | 35 |
| 4.1.3.2. | VERSIONES E ITERACIONES | 42 |
| 4.1.4. | CODIFICACIÓN | 50 |
| 4.1.5. | PRUEBAS | 52 |
| 4.1.6. | MANTENIMIENTO | 56 |

| | |
|---|-----------|
| RECOMENDACIONES Y SUGERENCIAS..... | 57 |
|---|-----------|

| | |
|---------------------------|-----------|
| BIBLIOGRAFÍA | 58 |
|---------------------------|-----------|

ANEXOS

ÍNDICE DE DIAGRAMAS

| | |
|---|----|
| Diagrama 01: Caso de uso general del sistema..... | 37 |
| Diagrama 02: Diagrama de secuencia validar usuario | 38 |
| Diagrama 03: Diagrama de secuencia administrar usuario | 39 |
| Diagrama 04: Diagrama de secuencia generar reporte mensual | 39 |
| Diagrama 05: Diagrama de colaboración validar usuario con el sistema | 40 |
| Diagrama 06: Diagrama de colaboración administrar usuario..... | 40 |
| Diagrama 07: Diagrama de colaboración generar reporte mensual | 41 |
| Diagrama 08: Diagrama de clase del sistema | 41 |
| Diagrama 09: Modelo entidad – relación..... | 42 |

RESUMEN

El Registro Civil de la Municipalidad Distrital de Cabana, está encargado de realizar inscripciones y expedir de partidas de nacimiento, matrimonios y defunciones a los usuarios pero no contaba con un sistema automatizado para generar procesos más eficaces en función de las necesidades del servicio que se presta, razón por la cual se desarrolló el presente trabajo de investigación con el objetivo: Desarrollar un sistema de información para administrar el Registro Civil de la Municipalidad de Cabana.

Para el desarrollo del sistema de información, se utilizó la metodología Xtreme Programming (XP) porque se amolda a los requerimientos, hasta la puesta del funcionamiento del sistema. Para el diseño se utilizó el UML, con su herramienta Rational Rose. Para poder evaluar la calidad del software se utilizó el estándar ISO – 9126 y también se utilizó la prueba estadística no paramétrica de rangos de wilcoxón, ya que la muestra no tenía distribución normal.

Implementado el sistema de información se ha cumplido con la automatización del Registro Civil, el cual permitió cubrir todas las necesidades, donde se obtuvo resultados satisfactorios de los usuarios donde se aplicó la prueba estadística no paramétrica de rangos de wilcoxón con el 95% de nivel significativo y también se demostró según el estándar ISO-9126 que el producto de desarrollo del sistema de información cumple los requisitos en una escala de 101.

Palabras Claves: Sistema, diseño, registrar, modelado.

ABSTRACT

The Registry of the District Municipality of Cabana, taking care to make attention to users, register and issue birth certificates, marriages and deaths, but did not have a system that allows you to generate more efficient processes based on the needs of service provided, this was the reason that the present research work for which the next target was raised was developed: To develop an information system to manage the Registry Office of the Municipality of Cabana.

For the development of the information system, the methodology Xtreme Programming (XP) was used because it conforms to the requirements, and commissioning of the system operation. To design the UML was used with the tool Rational Rose. To assess the quality of the ISO standard software was used - 9126 and nonparametric statistical test of Wilcoxon was also used as the sample was not normally distributed.

Implemented information system has complied with the automation of the Civil Registry, which allowed all needs where successful user where the nonparametric statistical test of Wilcoxon was applied to the 95% significance level was obtained and demonstrated according to the ISO -9126 standard product development information system meets the requirements on a scale of 101.

Keywords: System, design, recording, modeling.

INTRODUCCIÓN

El área de Registro Civil desempeña un papel fundamental en la conformación del sistema de Registro Nacional de Identificación, ya que a través de los actas del estado civil se podrá certificar y acreditar la veracidad de la identidad de las personas la cual se planteó como objetivo desarrollar un sistema de información

La metodología de trabajo, XP (Xtreme Programming) por ser un método de desarrollo de software que abarcó todo el ciclo de vida de las aplicaciones hasta la puesta en operación del sistema. Sus procesos de desarrollo son los roles, planificación, diseño y codificación, y la herramienta para el modelamiento del sistema de negocio. El trabajo está conformado por cinco (6) capítulos:

Capítulo I: Este capítulo contempla el planteamiento del problema, los objetivos de la investigación, la justificación y la hipótesis.

Capítulo II: Está constituido por los antecedentes que apoyan la investigación y las bases teóricas que le dan sustento al trabajo investigativo.

Capítulo III: Este capítulo está constituido por Materiales y Métodos

Capítulo IV: En este capítulo se detalla los resultados de la metodología que se implementó y la explicación del trabajo realizado durante el periodo de desarrollo.

Capítulo V: Para finalizar se plantean las conclusiones.

Capítulo VI: Recomendaciones y sugerencias.

CAPÍTULO I

PLAN DE INVESTIGACIÓN

1.1. EL PROBLEMA

El avance de las tecnologías de información, en estos últimos años se han desarrollado crecientemente en las instituciones públicas o privadas el uso del diseño y ejecución de diversos software de información para el control de diversos tipos de manejo de información, usando las herramientas de hardware y software que nos facilitan a realizar diferentes trabajos en diferentes modalidades para demostrar ser eficiente y eficaz en sus servicios.

Dentro de este contexto se enmarca, el área de Registro Civil de la Municipalidad Distrital de Cabana, está orientada a brindar atención a toda la población que está inscrita en el área de Registro Civil. La problemática que se tiene actualmente en el área de registro Civil de la Municipalidad Distrital de Cabana, es en cuanto a la atención de usuarios en inscripciones y expedir partidas de nacimientos, matrimonios y defunciones la cual es inscritas y buscadas manualmente donde demanda tiempo. También una de las dificultades es en los reportes estadísticos mensuales que se debe expedir a la RENIEC de su jurisdicción.

Las partidas de nacimiento, matrimonios y defunciones se registran y se almacenan en un libro físico que es entregado departe del RENIEC, dificultando en la mayoría de los casos, su ubicación y manipulación. Esta situación retrasa el proceso de atención al cliente, ya que el encargado o jefe de Registro Civil necesita tener el libro de registros a mano, al momento de realizar la búsqueda. Además, el libro físico de partidas de nacimiento, matrimonios y defunciones están expuestas en un archivador bibliotecario, donde es libre, siendo susceptible a extravíos o manipulación por personas ajenas. Los reportes estadísticos son necesarios para el control y evaluación del servicio que se prestó durante el mes, el cual los lleva el jefe de registro y estadística con una herramienta ofimática de procesamiento de texto (Word), debido a que es difícil realizar los conteos que se atienden por día, esto resulta un proceso lento y genera mucho trabajo y demanda de tiempo.

Por las razones expuestas, se hace pertinente realizar este proyecto de tesis, con la finalidad de brindarle al área de Registro Civil de la Municipalidad Distrital de Cabana, una aplicación que optimice la totalidad de sus procesos, incorporando módulos, métricas de calidad y confiabilidad de la información. Para llevar este proyecto a su culminación se utilizó la metodología extreme programming (XP), ya que por su fácil adecuación para proyectos pequeños resulta muy útil al momento de hacer el análisis y diseño del sistema.

En relación con estos planteamientos se pretende dar solución a la siguiente interrogante:

¿El sistema será eficiente en la administración del Registro Civil de la Municipalidad Distrital de Cabana - 2012?

1.2. JUSTIFICACIÓN DE LA INVESTIGACIÓN

El área de Registro Civil desempeña un papel fundamental en la conformación del sistema de Registro Nacional de Identificación, ya que a través de los actas del estado civil se podrá certificar y acreditar la veracidad de la identidad de las personas, actualmente no cuenta con ningún sistema de información, se ha notado una deficiencia en cuanto a las inscripciones y solicitudes de partidas de nacimiento, matrimonios, defunciones y también en los reportes estadísticos mensuales que se emiten a la RENIEC.

Con esta presente investigación se pretende demostrar que el desarrollo del sistema de información permitirá al área de registro civil ser más eficiente y eficaz en cuanto a la calidad de los servicios que ofrece. Asegurar que las informaciones estadísticas generadas estén disponibles en cualquier momento que la RENIEC de su jurisdicción lo disponga.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar un sistema de información para administrar el Registro Civil de la Municipalidad de Cabana - 2012.

1.3.2. OBJETIVOS ESPECÍFICOS

- ✓ Analizar y Diseñar los servicios de la institución y adaptarlos al Sistema de Información.
- ✓ Implementar la Base de Datos del Sistema de información.
- ✓ Implementar los módulos de almacenamiento registros, buscadadas y reportes para el gestor de base de datos.
- ✓ Desarrollar una Interfaz amigable para fácil uso del sistema.

1.4. HIPÓTESIS

Con este desarrollo de sistema de información es eficiente la administración del Registro Civil de la Municipalidad Distrital de Cabana.

CAPÍTULO II

MARCO TEÓRICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Tesis por: Edgar Primero Castillo Barriga “DESARROLLO DE SISTEMA DE INFORMACIÓN ORIENTADA A OBJETOS PARA LA GESTIÓN DE REGISTRO CIVIL DE LA MUNICIPALIDAD DISTRITAL DE CHUCUITO JULI - 2003”. Teniendo como conclusión final:

El sistema de información orientado a objetos se implantara de manera óptima, mejorando la gestión de la oficina de registro civil de la Municipalidad Distrital de Chucuto - Juli, el cual genera información oportuna y veraz, tomándose decisiones en forma eficiente y eficaz.

Para el desarrollo del sistema no existe un método peculiar sin embargo la metodología orientada a objetos es un método que fácilmente se adecua a la realidad del sistema, permitiéndonos desarrollar una aplicación con gran nivel de performance, eficiencia y reutilización.

La utilización del método de Rumbaugh aplicando la metodología OMT, sirvió como herramienta fundamental, para el desarrollo del sistema de información, pues nos permitió hacer un detallado análisis y diseño

orientado a objetos en forma organizada, para luego pasar rápidamente a la implementación del sistema.

El tratamiento del sistema, se usó el prototipo en espiral, permitiendo satisfacer las expectativas y requerimientos del usuario.

Tesis presentado por: Rubén Braulio Flores Mamani “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE REGISTRO CIVIL EN LA MUNICIPALIDAD PROVINCIAL DE PUNO PERIODO 2005”.Teniendo como Conclusión:

La aplicación del sistema de información de registro civil de la Municipalidad Provincial de Puno tiene un impacto positivo en la administración de la función registral.

El sistema no ha tenido problemas de adaptación para el manejo del sistema, por la simplicidad del diseño de la interfaz.

El proceso unificado de desarrollo es una buena herramienta para el desarrollo de aplicación cliente – servidor.

Tesis realizado por: Wilber Quispe Hanco, “DESARROLLO PROTOTIPO DE SISTEMA CLIENTE - SERVIDOR PARA EL REGISTRO CIVIL DEL DISTRITO DE ASILLO CON INTERACCIÓN A WEB - 2005”.Teniendo como Conclusión:

La implementación de sistema permite al registro civil del Distrito de Asillo, contar con una herramienta eficiente que ayude en el desarrollo de áreas de almacenamiento reportes y toma de decisiones.

La elección del desarrollo del sistema, resulto muy práctico y útil modelar el sistema utilizando el lenguaje unificado de modelado / UML.

La implementación del sistema han sido muy consistentes y adecuados las tecnologías de programación orientada a objetos: PHP, MySql, Java Scrip.

La implementación del sistema dio como resultado principal a través de la encuesta ejecutada, lo siguiente:

El 100 % de los operadores afirman que el control de descripciones de actas de nacimiento, Matrimonio y defunciones es óptimo, eficiente y confiable.

El 100 % de los usuarios consideran que el acceso de la información a través de la web para realizar solicitudes, comentarios y acceso a la información es óptimo eficiente y confiable.

2.2. BASE TEÓRICA

2.2.1. SISTEMA DE INFORMACIÓN

Un Sistema de información es un conjunto organizado de elementos que interactúan entre sí dirigidos a recoger, procesar, almacenar y distribuir información de manera que pueda servir de apoyo a las personas adecuadas dentro de una empresa o negocio, de modo que desempeñen sus actividades de manera eficaz y eficiente.

UN SISTEMA DE INFORMACIÓN REALIZA CUATRO ACTIVIDADES BÁSICAS: Entrada, almacenamiento, procesamiento y salida de información.

ENTRADA DE INFORMACIÓN: Es el proceso mediante el cual el sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas. Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáner, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

ALMACENAMIENTO DE INFORMACIÓN: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

PROCESAMIENTO DE INFORMACIÓN: Es la capacidad del sistema de Información para efectuar cálculos de acuerdo con una

secuencia de operaciones pre-establecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

SALIDA DE INFORMACIÓN: La salida es la capacidad de un sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficados y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interface automática de salida.¹

2.2.2. TIPOS DE SISTEMA DE INFORMACIÓN

SISTEMA DE PROCESAMIENTO DE TRANSACCIONES: Cuando un sistema recopila, almacena y altera la información creada a partir de transacciones llevadas a cabo dentro de una organización se denomina sistema de procesamiento de transacciones. Tiene como finalidad procesar las transacciones

¹ Sabana, M. M. (2006). Modelamiento e Implementación de Base de Datos. (1ª ed.) Lima: Grupo Editorial Megabyte S.A.C. Pág. 28.

diarias de una empresa, acumulando toda la información recibida en una base de datos para su posterior consulta.

SISTEMA DE INFORMACIÓN GERENCIAL: Un sistema de información gerencial es aquel utilizado por la empresa para solventar inconvenientes en la misma. Es decir, el objetivo del mismo es la suministración de información para la resolución de problemas a través de la interacción entre tecnologías y personas. Los datos aportados por el sistema deben disponer de cuatro cualidades elementales: calidad, oportunidad, cantidad y relevancia.

SISTEMA DE INFORMACIÓN EJECUTIVA: Esta tecnología es utilizada por los gerentes de una empresa, ya que permite acceder a la información interna y externa de la misma, disponiendo de los datos que puedan llegar a afectar su buen rendimiento.

De esta manera, el ejecutivo podrá conocer el estado de todos los indicadores, incluso aquellos que no cumplan con las expectativas y a partir de esto, tomar las medidas que considere adecuadas.²

2.2.3. METODOLOGÍA ÁGIL

Las Metodologías Ágiles constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de

² <<http://www.tiposde.org/informatica/89-tipos-de-sistemas-de-informacion/>> Consultado: [21, de Marzo; 2012].

e-Project que las metodologías convencionales (ISO-9000, CMM, etc.) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

2.2.4. METODOLOGÍA APLICADA XP (XTREME PROGRAMMING)

XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos muy cambiantes.³

CRITERIOS DE LA METODOLOGÍA

Software que funcione es más importante que documentos exhaustiva, es decir si el software no funciona no vale nada.

La colaboración con el cliente es más importante que la negociación de contratos.

El éxito de los proyectos se basa en una realimentación permanente.

La respuesta ante el cambio es más importante que el siguiente de un plan.

³ <<http://foro.chuidiang.com/metodologias-agiles/como-utilizo-la-metodologia-xp/html>> Consultado: [24, de marzo, 2012].

2.2.5. LENGUAJE DEL MODELADO UNIFICADO (UML)

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. En otras palabras, es el lenguaje en el que está descrito el modelo.

Los diagramas del UML son la representación gráfica de una colección de elementos con sus relaciones, ofreciendo así una vista del sistema a modelar. Para poder representar de forma correcta un sistema, el lenguaje presenta una amplia variedad de diagramas para así visualizar el sistema desde diversas perspectivas.

Entre esos diagramas se encuentran:

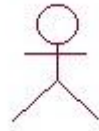
- A. Diagramas de Casos de Uso
- B. Diagramas de Secuencia
- C. Diagramas de Actividades
- D. Diagramas de clases

DIAGRAMAS DE CASOS DE USO

Los Casos de Uso son parte del análisis (qué). De forma que al ser parte del análisis nos ayudan a describir qué es lo que el sistema debe hacer. Los Casos de Uso son qué hace el sistema desde el punto de vista del usuario. Es decir, describimos el uso del sistema y cómo este interactúa con el usuario.

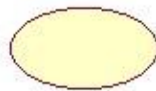
Elementos

Actor:



Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Caso de Uso:



Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

DIAGRAMAS DE SECUENCIA

Los mensajes pueden ser bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada. Los

mensajes síncronos tienen una caja vertical en un lateral del objeto invocante que muestra el flujo del control del programa.

DIAGRAMAS DE COLABORACIÓN

En estos diagramas de colaboración se muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

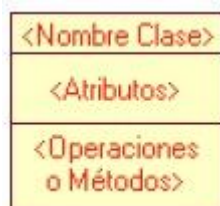
En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

DIAGRAMAS DE CLASES

En el diagrama de clases se muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Además los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases

«conocen» a qué otras clases o qué clases «son parte» de otras clases.

En UML, una clase es representada por un rectángulo que posee tres divisiones:



En donde:

Superior: Contiene el nombre de la Clase (Nacimiento, Defunciones, Matrimonios)

Intermedio: Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).

Inferior: Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).⁴

2.2.6. ARQUITECTURA CLIENTE – SERVIDOR

Arquitectura hardware y software adecuada para el proceso distribuido, en el que la comunicación se establece de uno a varios. Un proceso es un programa en ejecución. Proceso cliente

⁴ Perdita, S. R. (2002). Utilización de UML en Ingeniería del Software con Objetos y Componentes.

es el que solicita el servicio. Proceso servidor es capaz de proporcionar un servicio.

Un proceso cliente se puede comunicar con varios procesos servidores y un servidor se puede comunicar con varios clientes. Los procesos se pueden ejecutar en la misma máquina o en distintas máquinas comunicadas a través de su red, por lo general, la parte de la aplicación correspondiente al cliente se optimiza para la interacción con el usuario, ejecutándose en su propia máquina, mientras que la parte correspondiente al servidor proporciona la funcionalidad multiusuario centralizada y se ejecuta en la máquina remota.

Una aplicación Cliente/Servidor típica es un servidor de base de datos al que varios usuarios realizan consultas simultáneamente. El proceso cliente realiza una consulta, el proceso servidor le envía las

Tablas resultantes de la consulta y el proceso cliente las interpreta y muestra el resultado en pantalla.

2.2.7. HERRAMIENTAS DE DESARROLLO

Rational Rose

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas, se utiliza para antes de proceder a construirlo utilizando el UML con sus diagramas de caso

de uso, diagramas de secuencia, diagramas de colaboración y diagramas de clase.

Macromedia Dreamweaver

Es una aplicación en forma de suite (basada en la forma de estudio de (Adobe Flash) que está destinada a la construcción, diseño y edición de sitios, videos y aplicaciones Web basados en estándares. Creado inicialmente por Macromedia (actualmente producido por Adobe Systems) es el programa más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Expression Web y tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras. Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer y no validaba como HTML estándar.

Macromedia Fireworks

Es una aplicación versátil en forma de estudio que ofrece un ambiente eficiente para la creación rápida de prototipos de sitios Web e interfaces de usuario, permite crear y editar imágenes de mapas de bits y vectoriales, diseñar efectos web, recortar y optimizar elementos gráficos, ayudando a resolver los principales

problemas que enfrentan los diseñadores gráficos y los creadores de sitios webs.

2.2.8. APPSERV

Appserv es una herramienta OpenSource para Windows con Apache, MySQL, PHP y otras adicciones, en la cual estas aplicaciones se configuran en forma automática, lo que permite ejecutar un servidor web completo.

Como extra incorpora phpMyAdmin para el manejo de MySQL.

Inmediatamente después de la instalación está disponible completamente funcional del servidor web Apache , que se ejecuta en el equipo local, que puede ejecutar un número ilimitado de sitios para ser muy eficaz para desarrollar y depurar scripts PHP sin subir archivos a un servidor remoto. El programa se distribuye gratuitamente bajo licencia GNU General Public License y es gratis, fácil de usar web-servidor capaz de servir páginas dinámicas. Con esta configuración, puede implementar rápidamente un equipo hecho y derecho y el servidor web rápido, con acceso desde la red local, sin necesidad de configuración adicional.

2.2.9. LENGUAJE DE PROGRAMACIÓN

Un lenguaje de programación se refiere a cualquier lenguaje artificial que pueda ser empleado para definir una secuencia de

instrucciones para su procesamiento por una computadora u ordenador.

LENGUAJE DE MARCADO HIPERTEXTO (HTML)

HTML son las siglas de "lenguaje de marcado hipertexto" (Hypertext Markup Lenguaje) y es un lenguaje universal de internet. Podemos crear todo tipo de páginas web con HTML, desde simples páginas web conteniendo solo texto, hasta aquellas que incluyen animaciones complejas.

Sin embargo, no necesitamos utilizar aplicaciones complicadas para desarrollar estas páginas, pues todo lo que necesitamos es un simple editor de textos, como el bloc de notas (notepad), Word pad o Word, para guardar un documento como una página web necesitamos hacerlo con una extensión .HTML (o htm).

Todos los componentes que aparecen en una página web se llaman etiquetas tags o elementos, estas etiquetas indican lo que el código hará y colocan el texto de los documentos HTML. La mayoría de las etiquetas vienen en pares, aunque algunas vienen solas.

Formularios

Una etiqueta de apertura marca el principio de un código HTML y una de cierre lo hace con su final. La etiqueta de inicio contiene el nombre de la misma entre los caracteres de <<menor que>> (<) <<mayor que >> (>). La etiqueta del cierre es muy similar al de la

apertura. Con la única diferencia que debemos añadir la barra diagonal (/) delante del nombre de la etiqueta. Por lo tanto hace efecto en el código que hay entre ellas. Interpretar la mayoría de las etiquetas es sencillo ya que son fáciles de recordar debido a que suelen ser la abreviatura de sus nombres.⁵

PRE - PROCESADOR DE TEXTO (PHP)

PHP es un "pre-procesador de hipertexto" (Hipertexto Pre-Procesador) y por ende se ejecuta en un servidor web remoto para procesar páginas web antes de que sean cargadas en el navegador, además de sus potentes características, PHP, es un lenguaje simple que ha sido diseñado específicamente para el desarrollo de páginas web. PHP es un software de código abierto (open source) y puede ser descargado gratuitamente.⁶

JAVASCRIPT

Es un lenguaje de SCRIPT compacto basado en objetos, originariamente era denominado Live Script y fue desarrollado por Netscape para su navegador. Fue este el primer cliente en reincorporarlo, se ejecuta en 16 plataformas diferentes, incluyendo el entorno de Microsoft.

Java script permite la realización de aplicaciones de propósito general a través de WWW y aunque no está diseñado para el desarrollo de grandes aplicaciones es suficiente para

⁵ De la Cruz, V. J. (2004). PHP y MySQL. (1ª ed). Lima: Grupo Editorial Megabyte S.A.C. Pág. 205.

⁶ De la Cruz, V. J. (2004). PHP y MySQL. (1ª ed.) Lima: Grupo Editorial Megabyte S.A.C. Pág. 3.

implementación de aplicaciones WWW completas o interfaces hacia más complejas.

2.2.10. SISTEMA DE ADMINISTRACIÓN DE BASE DE DATOS

MySQL es un sistema para la administración de base de datos relacional (RDBMS) se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir la necesidad de cualquier tipo de organización.

MySQL incluye todos los elementos necesarios para instalar el programa, prepara diferentes niveles de acceso de usuario, administrar el sistema, proteger los datos y es utilizado en la actualidad en casi todos los sistemas operativos.

MySQL utiliza el lenguaje de consulta estructurado (SQL) se trata del lenguaje utilizado por todas las bases de datos, este lenguaje permite crear base de datos, así como agregar, manipular y recuperar datos en función de criterios específicos.⁷

2.2.11. WEB APACHE

Es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue

⁷ LUKE. W. y LAURA. T. (2005). Desarrollo Web con PHP y MySQL.

reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

2.2.12. INTERFAZ DE USUARIO

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar. Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, teclado, ratón, los beeps y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora. La mejor interacción humano-máquina a través de una adecuada interfaz (Interfaz de Usuario), que le brinde tanto comodidad, como eficiencia.⁸

⁸ <http://es.wikipedia.org/wiki/Interfaz_de_usuario> Consultado: [25, de marzo, 2012.]

2.2.13. HARDWARE

Se denomina hardware o soporte físico al conjunto de elementos materiales que componen un ordenador. Hardware también son los componentes físicos de una computadora tales como el disco duro, CD-ROM, disquetera (floppy), etc. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos. El hardware se refiere a todos los componentes físicos (que se pueden tocar) de la computadora: discos, unidades de disco, monitor, teclado, ratón (mouse), impresora, placas, chips y demás periféricos.⁹

2.2.14. ESTÁNDARES DE CALIDAD

Norma de Calidad iso-9126: Es un estándar internacional para la evaluación del Software, fue originalmente desarrollado en 1991 para proporcionar un esquema para la evaluación de calidad del software.

La normativa define seis características de la aplicación, estas seis características son divididas en: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de sub

⁹ <<http://www.misrespuestas.com/que-es-hardware.html>> Consultado: [25, de Marzo, 2012.]

características que permiten profundizar en la evaluación de la calidad de productos de software.¹⁰

2.3. DEFINICIÓN DE TÉRMINOS BÁSICOS

2.3.1. DESARROLLO DE SOFTWARE

Desarrollar un software significa construirlo simplemente mediante su descripción. Está es una muy buena razón para considerar la actividad de desarrollo de software como una ingeniería. En un nivel más general, la relación existente entre un software y su entorno es clara ya que el software es introducido en el mundo de modo de provocar ciertos efectos en el mismo.

2.3.2. SERVIDOR

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes.

2.3.3. CÓDIGO FUENTE

También denominado programa fuente, es un texto escrito en un lenguaje de programación para crear el programa. Es la forma del programa legible por el programador.

¹⁰ <<http://www.alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Joaquin%20Ruiz%20Expo.html>>, Consultado: [22, de marzo, 2012.]

2.3.4. DIGITAR

Incorporar datos a la computadora utilizando el teclado.

2.3.5. SERVIDOR

Es un equipo o Hardware diseñado para brindar servicios basados en HTTP para proveer intercambio o producción de información bajo criterios de los protocolos de aplicaciones.

2.3.6. SERVIDOR WEB

Un servidor web es un programa que implementa el protocolo HTTP (Hyper Text Transfer Protocol). Este protocolo permite que el internet basado en este protocolo se mantenga libre en la mayoría de servidores.

2.3.7. OPTIMIZAR

Buscar la mejor manera de realizar una actividad.

2.3.8. REPORTE

Informe que se emite o presenta con base a la realización de una actividad o tarea.

2.3.9. GESTIÓN

El término gestión, por lo tanto, implica al conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto.

2.3.10. REGISTRO CIVIL

Es la institución pública que tiene por finalidad la inscripción de los hechos y actos fundamentalmente de las personas como son los nacimientos, matrimonios, defunciones y otros.

2.3.11. REGISTRADOR

Jefe de Registro Civil, es la persona que se encarga de registrar a los usuarios en las actas de partida de nacimiento, matrimonio o defunciones.

2.3.12. DIGITAR

Es la persona que hace uso de los servicios proporcionados del área de Registro Civil, solicita o inscribe algún tipo de partida ya sea, nacimiento, matrimonio o defunción.

2.3.13. ACTAS DE NACIMIENTO

Documento donde se registra a la persona recién nacida.

2.3.14. ACTAS DE MATRIMONIO

Documento donde se registra a las personas unidas en matrimonio.

2.3.15. ACTA DE DEFUNCIÓN

Documento donde se registra a la persona que acaba de fallecer.

2.3.16. REGISTRO NACIONAL DE IDENTIFICACIÓN Y ESTADO CIVIL (RENIEC)

El Registro Nacional de Identificación y Estado Civil (RENIEC), es un organismo autónomo del Estado Peruano encargado de la identificación de los peruanos, otorgando el Documento Nacional de Identidad (DNI), registrando hechos vitales como nacimientos, matrimonios, defunciones, divorcios y otros que modifican el estado civil.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. LOCALIZACIÓN

El presente trabajo de investigación se realizó en el Área de de Registro Civil de la Municipalidad Distrital de Cabana, Provincia de San Román, Departamento Puno.

3.2. POBLACIÓN

La población de quienes validaron el sistema estuvo conformada por los operadores del sistema, en un total de 5200. El personal encargado del Registro Civil de la Municipalidad Distrital de Cabana.

3.3. RECOPIACIÓN DE LA INFORMACIÓN

Para la recopilación de datos se utilizó el siguiente instrumento que fue aplicado en los inscriptores autorizados en el área de Registro Civil de la Municipalidad Distrital de Cabana.

La ficha de evaluación del ISO-9126 que ve la calidad del software.

3.4. METODOLOGÍA DE INVESTIGACIÓN

3.4.1. METODOLOGÍA XTREMME PROGRAMMING XP

Existen numerosas metodologías ágiles para el desarrollo de software, aunque las tres metodologías más utilizadas son: Crystal methodologies, Scrum, XP (Extreme programming) escogiendo esta última como la metodología aplicada para el desarrollo de este proyecto.

XP (Extreme programming) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todo los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos muy cambiantes.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.2. PROCESO DEL DESARROLLO DEL SOFTWARE

4.1.1. ANÁLISIS DE SISTEMA

A) DESCRIPCIÓN DEL SISTEMA ACTUAL

En el Registro Civil el proceso de trámite de solicitar y expedir partidas de nacimiento, matrimonios y defunciones es de la siguiente manera donde el declarante tiene que acercarse al jefe de Registro Civil ahí es donde solicita registrarse o ser expedido de una partida nacimiento, matrimonio o defunción, donde el jefe de Registro Civil busca en el libro físico almacenado, dificultando, en la mayoría de los casos, su ubicación y manipulación. Esta situación retrasa el proceso de atención al cliente, ya que el encargado o jefe de Registro Civil necesita tener el libro de registros a mano, al momento de realizar la búsqueda.

Los reportes estadísticos son necesarios para el control y evaluación del servicio que se prestó durante el mes, el cual las lleva el jefe de registro y estadística con una herramienta ofimática de procesamiento de texto (Word), debido a que es

difícil realizar los conteos que se atienden por día, esto resulta un proceso lento y genera mucho trabajo y demanda de tiempo.

B) ANÁLISIS DE VIABILIDAD

La administración del Registro Civil está en la capacidad de manipular el sistema, el cual se asignaran usuarios y contraseñas para el ingreso al sistema, esto por motivos de seguridad para la información.

La administración cuenta con equipos adecuados para la implementación del sistema, será necesario instalar AppServ para la prueba correspondiente.

C) ANÁLISIS DE ROLES

Programador: Es mi persona quien va crear el código del sistema.

Ciente: Seria el jefe de Registro Civil y el desarrollador de la investigación.

Encargado de pruebas (Tester): Seria el jefe de Registro Civil quien ejecutó las pruebas regularmente, e informa los resultados y apreciaciones al desarrollador del sistema.

Encargado de seguimiento (Tracker): Seria el Jefe de Registro Civil el que proporciona realimentación al desarrollador. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo

real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

Entrenador (Coach): Mi persona fue el responsable del proceso global para realizar las guías para las prácticas.

Consultor: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

D) ANÁLISIS DE PLANIFICACIÓN

Historias de usuario

Los clientes describieron brevemente Los siguiente:

Tabla N° 02. Historias de usuario

| Numero de Historia | Historia de usuarios | Tareas |
|--------------------|---|--|
| 1 | Creación de la Base de Datos, donde se almacenara la información. | -Diseño e implementación de la Base de datos. |
| 2 | La aplicación debe ser amigable y de fácil uso para los posibles interesados en ella. | -Diseño e implementación de los módulos de consultas. |
| 3 | Cualquier persona (estudiante, profesor o administrativo) pueda realizar consultas acerca de los proyectos a partir de alguno de sus datos | -Diseño e implementación de los módulos de consulta. |
| 4 | Creación de una aplicación que guarde la información de los proyectos de grado presentados por los estudiantes. | -Diseño e implementación de los módulos de inserción de datos. |
| 5 | Creación de una aplicación que Modifique la información de los proyectos de grado presentados por los estudiantes. | -Diseño e implementación de los módulos de edición de datos. |
| 6 | Debe existir una parte privada en la aplicación para la administración de los datos, la cual solo pueda ser accedida por una persona autorizada | -Diseño e implementación de un módulo que solicite el ingreso de un nombre. -Usuario y contraseña para acceder a la parte privada. -Diseño e implementación de un modelo que permita crear un súper - usuario. |
| 7 | La aplicación debe tener las restricciones necesarias para evitar errores en la base de datos. | Se debe diseñar e implementar las validaciones respectivas en los diferentes campos que se ingresaran en la base de datos. |
| 8 | La aplicación debe estar disponible en la red | Elegir una herramienta de desarrollo para aplicaciones web. |

4.1.2. REQUERIMIENTOS FUNCIONALES:

RF1: Deben existir perfiles para el ingreso al sistema, este le dará mayor seguridad a la información para el manejo.

RF2: El sistema debe registrar partidas de nacimiento, matrimonios y defunciones.

RF3: El sistema debe realizar búsqueda de las partidas de nacimiento, matrimonios y defunciones.

RF4: El sistema debe generar reporte mensual de las partidas de nacimiento, matrimonios y defunciones.

REQUERIMIENTOS NO FUNCIONALES:

RN1: Interfaz agradable para un fácil entendimiento del software.

RN2: Disponibilidad el sistema se encontrar disponible todo los días.

RN3: Estabilidad el sistema soporta varios usuarios a la vez.

RN4: Portabilidad estará diseñado en un lenguaje multiplataforma.

RN5: Mantenimiento y Escalabilidad diseñado pensando en el crecimiento del sistema.

REQUERIMIENTOS TÉCNICOS:**Tabla N° 03. Requerimientos técnicos para el sistema**

| Software | Versión | Descripción |
|-----------------|----------------|---|
| Php | 5.x | Es un lenguaje de programación web a lado del servidor |
| MySql | 5.0 | Es un sistema de base de datos multihilo y multiusuario una versión estable de uso bajo GNU GPL. |
| Java Script | 3.0 | Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos. |
| HTML | 4.01 | Lenguaje de marcado de hipertexto para la elaboración de páginas web. |
| CSS | 2.0 | El nombre hojas de estilo en cascada, es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML |
| AppServ | 2.5.10 | Appserv es una herramienta OpenSource para Windows con Apache, MySQL, PHP y otras adiciones, en la cual estas aplicaciones se configuran en forma automática, lo que permite ejecutar un servidor web completo. |

4.1.3. DISEÑO DEL SISTEMA**4.1.3.1. DIAGRAMAS DEL UML**

Es la representación gráfica del sistema se desarrolló utilizando el lenguaje Unificado del Modelado, donde nos ha permitido especificar con los siguientes diagramas:

A) DIAGRAMA DE CASOS DE USO: En su forma general el sistema.

Casos de uso: Validar usuario, registrar o inscribir partidas de nacimiento, matrimonios, defunciones, realiza búsquedas de partidas de nacimiento, matrimonios, defunciones e imprime reporte general mensual.

Actores: Usuario (Registrador).

Propósito: Validar y administrar los usuarios que harán uso del sistema.

Resumen: Este caso de uso general: 1). Validar al usuario: Este caso de uso restringe el acceso de usuarios al sistema, y establece que cada usuario cuente con un nombre de usuario y una clave de acceso al sistema. 2). Administración del sistema: Registra y realiza búsquedas de partidas de nacimientos, matrimonios, defunciones 3). Generar reportes a los usuarios.

La figura mostrara el resumen de registros e inscripciones de partidas de nacimiento, matrimonios y fallecidos.

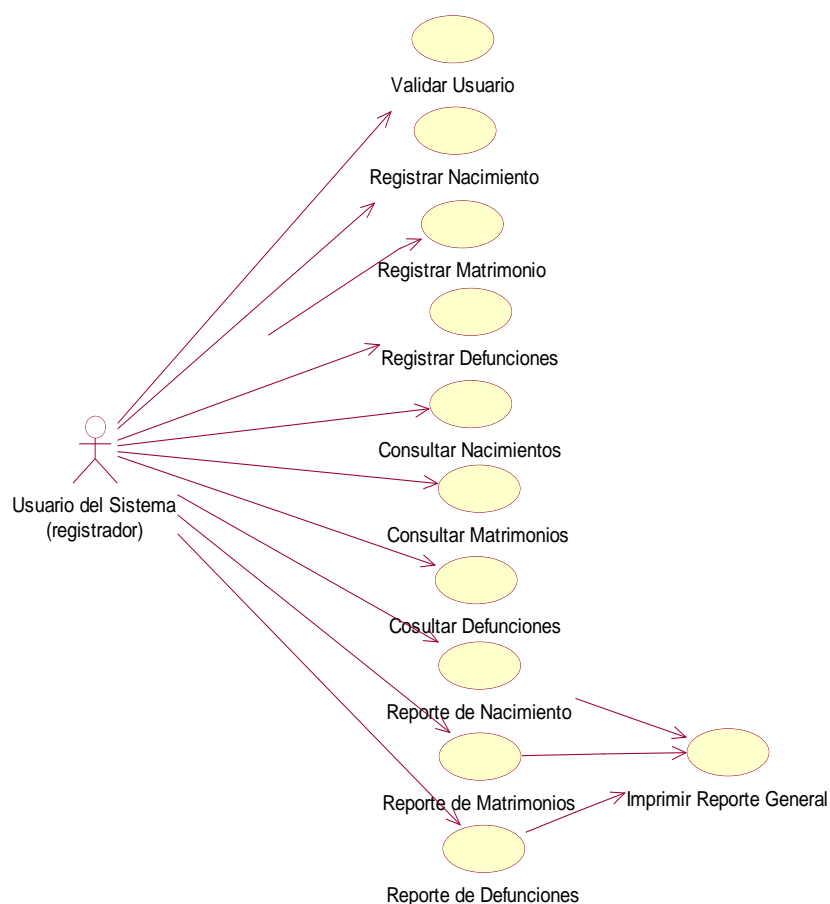


Diagrama 01: Caso de uso general del sistema. Fuente: Autor (2012).

B) DIAGRAMA DE SECUENCIA

INTERFAZ PARA VALIDAR USUARIO: El usuario ingresa el login para validar el usuario y contraseña el sistema comprueba el login y da como respuesta de acceso total al sistema.

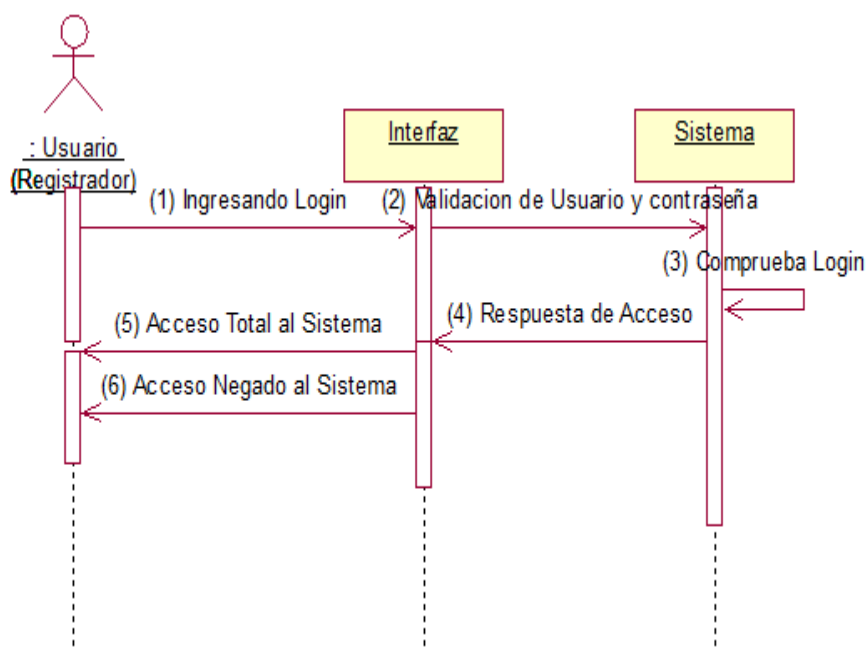


Diagrama 02: Diagrama de secuencia validar usuario. Fuente: autor 2012.

INTERFAZ DE ADMINISTRACIÓN DE USUARIO: El cliente solicita inscripción de partida de nacimiento o matrimonio o defunción, el usuario del sistema (registrador) solicita datos al cliente (declarante), y el declarante proporciona datos al usuario (registrador), el usuario ingresa datos al sistema y ordena guardar y el sistema registra y guarda datos, donde el sistema informa registro guardado correctamente.

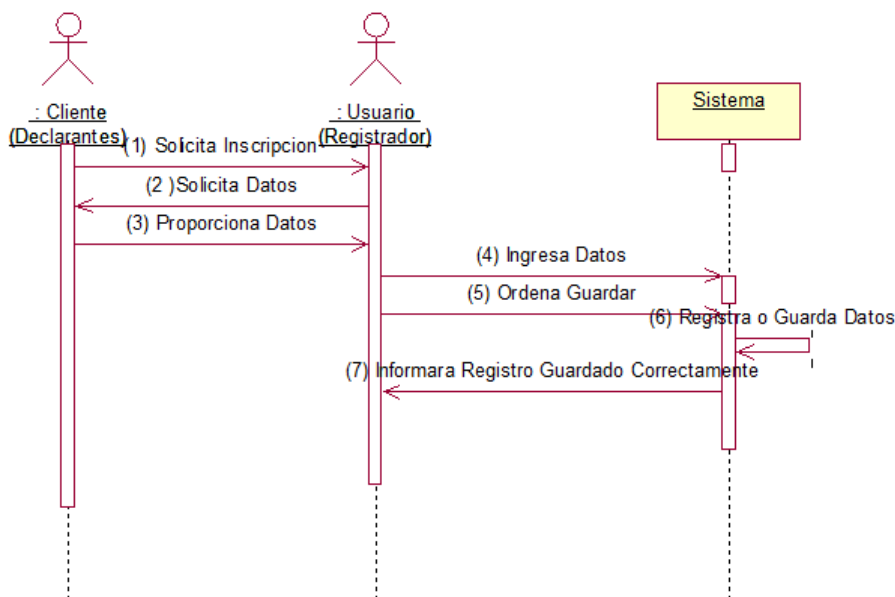


Diagrama 03: Diagrama de secuencia administrar usuario. Fuente: autor 2012.

INTERFAZ GENERAR REPORTE MENSUAL AL USUARIO:

Generar reportes a los usuarios. Estos reportes mostraran el resumen de registros o inscripciones de partidas nacimiento, matrimonios y defunciones en total al mes.

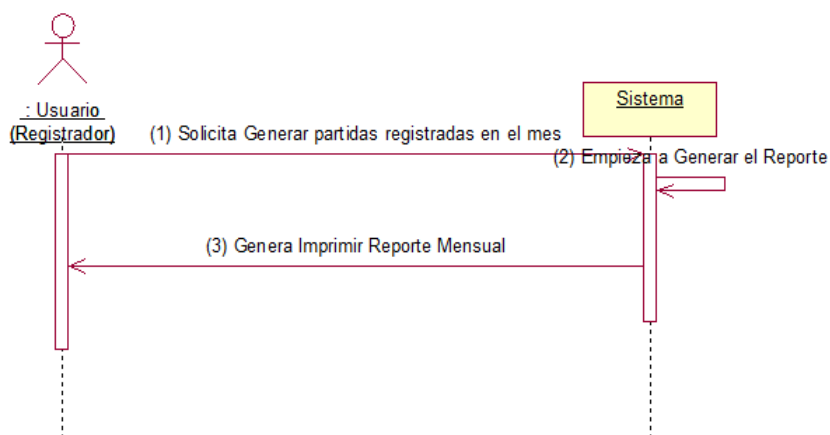


Diagrama 04: Diagrama de secuencia generar reporte mensual. Fuente: autor 2012.

C) DIAGRAMAS DE COLABORACIÓN

VALIDAR USUARIO CON EL SISTEMA

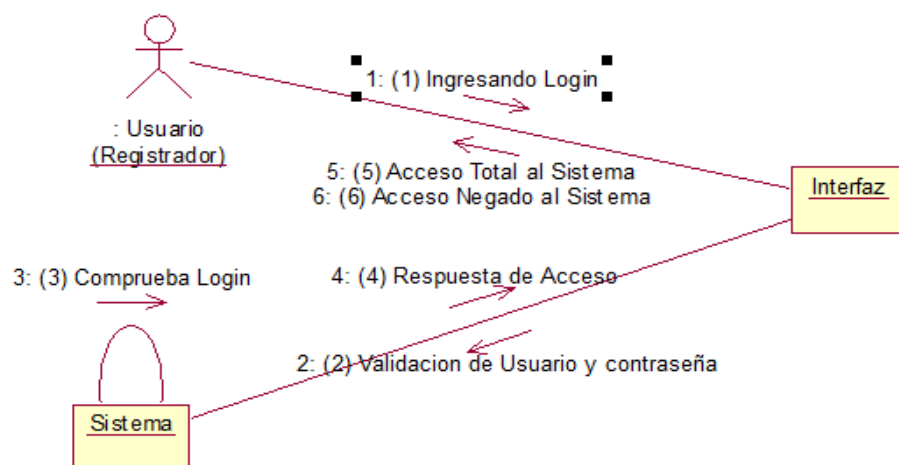


Diagrama 05: Diagrama de colaboración validar usuario con el sistema. Fuente: autor 2012.

DIAGRAMA DE COLABORACIÓN ADMINISTRAR USUARIO

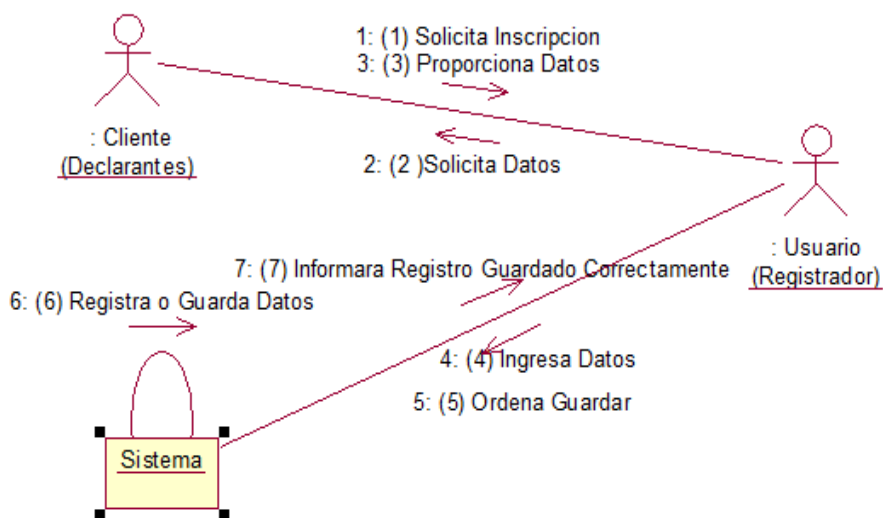


Diagrama 06: Diagrama de colaboración administrar usuario. Fuente autor 2012.

GENERAR REPORTE MENSUAL

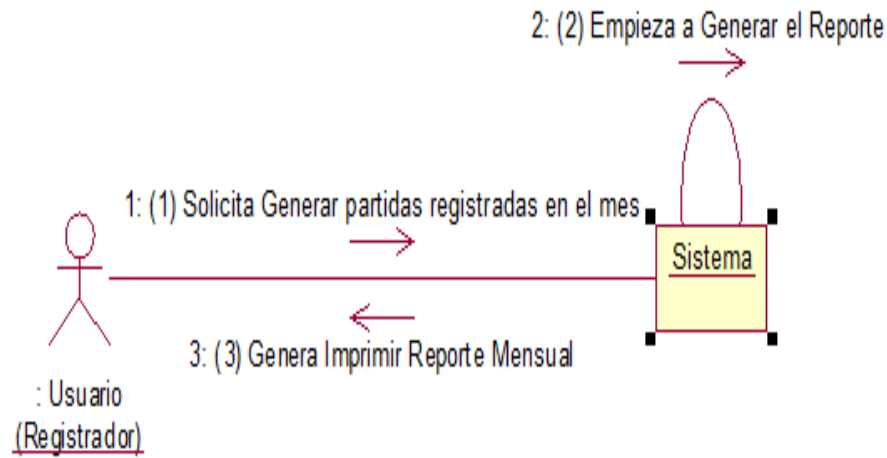


Diagrama 07: Diagrama de colaboración generar reporte mensual. Fuente: autor 2012.

D) DIAGRAMAS DE CLASE

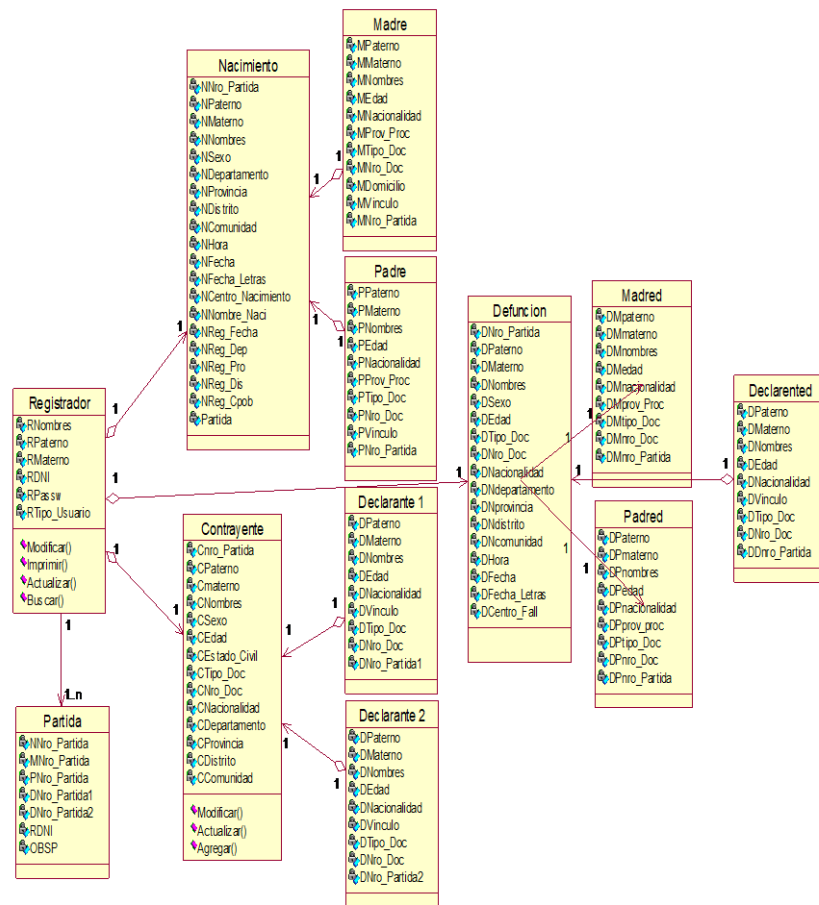


Diagrama 08: Diagrama de clase del sistema. Fuente: autor 2012.

4.1.3.2. VERSIONES E ITERACIONES

Versión 0.1

Iteración 1:

El diseño de la base de datos es la primordial para empezar con el desarrollo de cualquier aplicación, ya que un buen diseño de esta, es la base fundamental para el óptimo funcionamiento y éxito de cualquier software.

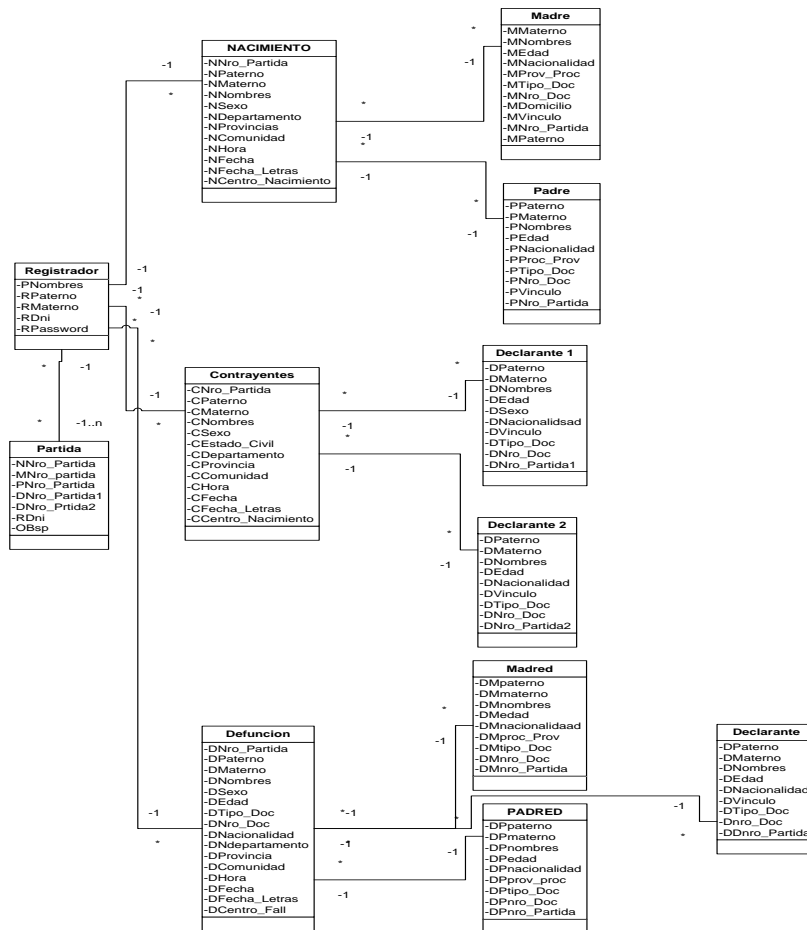


Diagrama 09: Modelo entidad – relación.

A continuación se describen las tablas:

Tabla REGISTRADOR: En esta tabla contiene datos del registrador es el encargado de ingresar y manipular el sistema.

RNombres: Este campo almacena el nombre del registrador.

RPaterno: Este campo almacena el apellido paterno del registrador.

RMaterno: Este campo almacena el apellido materno del registrador.

RDNI: Este campo almacena el número de identificación.

RPassword: Este campo almacena el nombre del registrador.

Tabla NACIMIENTO: En esta tabla se almacena todos los datos del recién nacido.

NNro_Partida: En este campo se almacena el número de partida de nacimiento.

NPaterno: En este campo se almacena el apellido paterno del recién nacido.

NMaterno: En este campo se almacena el apellido materno del recién nacido.

NNombres: En este campo se almacena el nombre del recién nacido.

NSexo: En este campo se almacena el sexo del recién nacido.

NDepartamento: En este campo se almacena el departamento a cual pertenece el recién nacido.

NProvincia: En este campo se almacena la provincia a la cual pertenece el recién nacido.

NComunidad: En este campo se almacena la comunidad a la cual pertenece el recién nacido.

NHora: En este campo se almacena la hora de nacimiento.

NFecha: En este campo se almacena la fecha de nacimiento.

NFecha_Letras: En este campo se almacena la fecha de nacimiento en letras.

NCentro_Nacimiento: En este campo se almacena el lugar de nacimiento: puede ser un hospital, domicilio u otros.

Tabla de CONTRAYENTES: En esta tabla se almacena todos los datos de los contrayentes.

CNro_Partida: En este campo se almacena el número de partida de nacimiento.

CPaterno: En este campo se almacena el apellido paterno del contrayente.

CMaterno: En este campo se almacena el apellido materno del contrayente.

CNombres: En este campo se almacena el nombre del contrayente.

CSexo: En este campo se almacena el sexo del contrayente.

CEstado_Civil: en este campo se almacena el estado civil del contrayente.

CDepartamento: En este campo se almacena el departamento a cual pertenece el contrayente.

CProvincia: En este campo se almacena la provincia a la cual pertenece el contrayente.

CComunidad: En este campo se almacena la comunidad a la cual pertenece el contrayente.

CHora: En este campo se almacena la hora de la celebración del matrimonio.

CFecha: En este campo se almacena la fecha del matrimonio.

CFecha_Letras: En este campo se almacena la fecha de celebración en letras.

CCentro_Nacimiento: En este campo se almacena el lugar de celebración del matrimonio.

Tabla DEFUNCIÓN

DNro_Partida: En este campo se almacena el número de partida.

DPaterno: En este campo se almacena el apellido paterno del difunto.

DMaterno: En este campo se almacena el apellido materno del difunto.

DNombres: En este campo se almacena el nombre del difunto.

DSexo: En este campo se almacena el sexo del difunto.

DDepartamento: En este campo se almacena el departamento.

DProvincia: En este campo se almacena la provincia.

DComunidad: En este campo se almacena la comunidad.

DHora: En este campo se almacena la hora del fallecimiento.

DFecha: En este campo se almacena la fecha del fallecimiento.

DFecha_Letras: En este campo se almacena la fecha del fallecimiento en letras.

DCentro_Nacimiento: En este campo se almacena el lugar de fallecimiento.

Versión 0.2

Iteración 1:

Como la aplicación debe de estar en red para que se pueda consultar, se seleccionó ASP.NET, ya que es uno del software de desarrollo para aplicaciones web más utilizados, es por esa razón que se seleccionó esta herramienta, para el desarrollo de esta aplicación.

Iteración 2:

Como se busca tener un diseño amigable, intuitivo y de fácil uso para la aplicación se creó una página principal (Master Page), la cual es una plantilla de la que heredan todas las demás páginas de la aplicación. De esta forma se tiene una estructura fija para todo el sitio web y mediante un ContentPlaceholder central, se heredaran totales las páginas, donde se pudo colocar los elementos visuales específicos de cada una de ellas.



Ilustración 1. Página maestra.

Iteración 3:

Como en la iteración 2 no se observó un avance significativo para sacar una nueva visión, se hizo necesario implementar el módulo de consultas, para ello se ingresaron directamente datos a la base de datos.

Los controles que proporciona la herramienta ASP.NET se pudieron crear los enlaces al motor de base de datos y a sus deferentes tablas, para así hacer las consultas respectivas.

Consulta por partidas de nacimiento: Simplemente se realiza su respectivo filtrado y se visualizara los datos deseados.

Consulta por partidas de matrimonio: Simplemente se realiza su respectivo filtrado y se visualizara los datos deseados.

Consulta por partidas de defunción: Simplemente se realiza su respectivo filtrado y se visualizara los datos deseados.

De esta manera se terminó la versión 0.2, cumpliendo con las tareas de la historia de usuario número 2, 3, 8.

Esta versión se le enseñó a los clientes, los cuales dieron su visto bueno al sentirse conformes con los avances presentados.

Versión 0.3

Iteración 1:

En esta versión se implementó el módulo para la edición de datos.

En esta versión se implementó el módulo de ingresar nuevos datos, porque por medio de los controles que proporciona la herramienta se podían editar los datos que existían en la base de datos.

De esta manera se terminó la versión 0.3, cumpliendo con las tareas de la historia de usuario número 4.

Esta versión se le enseñó a los clientes, los cuales dieron su visto bueno al asistir conformes con los avances presentados.

Versión 0.4

Iteración 1:

En esta versión se implementó el módulo para la edición de datos.

Este módulo fue muy similar al módulo de ingresar nuevos datos, porque por medio de los controles que proporciona la herramienta se podían editar los datos que existían en la base de datos.

De esta manera se terminó la versión 0.4, cumpliendo con las tareas de la historia de usuario número 5.

Esta versión se le enseñó a los clientes, los cuales dieron su visto bueno al asistir conformes con los avances presentados.

Versión 0.5 hasta la 0.9.

De la versión 0.5 hasta la versión 0.9 se trabajó en el diseño estético de la aplicación.

En cada una de las iteraciones que involucraba la entrada de datos por parte del usuario de la aplicación, se diseñaron e implementaron los diferentes tipos de validaciones respectivas para cada uno de los campos que lo requerían.

En este punto fue primordial validar los campos requeridos, los cuales no debían quedar vacíos ya que representarían un error al momento de ser guardados en la base de datos.

Cambio de tareas.

Como lo recomienda la Metodología Ágil XP, durante todo el desarrollo de la aplicación el desarrollador intercambia tareas continuamente logrando así un mejor desarrollo.

4.1.4. CODIFICACIÓN

LENGUAJE DE MARCADO HIPERTEXTO (HTML).

Todos los componentes que aparecen en una página web se llaman etiquetas tags o elementos, estas etiquetas indican lo que el código hará y colocan el texto de los documentos HTML. La mayoría de las etiquetas vienen en pares, aunque algunas vienen solas.

Formularios

Una etiqueta de apertura marca el principio de un código HTML y una de cierre lo hace con su final. La etiqueta de inicio contiene el nombre de la misma entre los caracteres de <<menor que>> (<) <<mayor que >> (>). La etiqueta del cierre es muy similar al de la apertura. Con la única diferencia que debemos añadir la barra diagonal (/) delante del nombre de la etiqueta. Por lo tanto hace efecto en el código que hay entre ellas. Interpretar la mayoría de las etiquetas es sencillo ya que son fáciles de recordar debido a que suelen ser la abreviatura de sus nombres.

PRE - PROCESADOR DE TEXTO (PHP).

PHP es un software de código abierto (open source) y puede ser descargado gratuitamente.

Características fundamentales de PHP.

- Es un lenguaje de script de código abierto para servidores.
- Es independiente del sistema operativo y siendo utilizado en Microsoft Windows Seven.
- Utiliza una amplia gama de servidores Web, útiles como Apache.
- Se conecta gran cantidad de base de datos, como MySQL.
- Una de las características que distingue PHP, es el que proporciona soporte a sitios web, de comercio electrónica que manejen bases de datos.
- Para proporcionar estos servicios, PHP utilizo el protocolo HTTP.

JAVA SCRIPT.

JavaScript permite la realización de aplicaciones de propósito general a través de WWW y aunque no está diseñado para el desarrollo de grandes aplicaciones es suficiente para la implementación de aplicaciones WWW completas o interfaces hacia más complejas.

ANÁLISIS DE VIABILIDAD

La administración del Registro Civil está en la capacidad de manipular el sistema, el cual se asignaran usuarios y contraseñas para el ingreso al sistema, esto por motivos de seguridad para la información.

La administración cuenta con equipos adecuados para la implementación del sistema, será necesario instalar AppServ para la prueba correspondiente.

4.1.5. PRUEBAS

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.

ESTÁNDAR ISO 9126

**FICHA DE EVALUACIÓN DE LA CALIDAD DEL PRODUCTO
ESTÁNDAR ISO – 9126.**

| INDICADORES | PUNTUACIÓN | | | | |
|--|------------|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 1. FUNCIONALIDAD | | | | | |
| Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios. | | | | X | |
| Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado. | | | | | X |
| Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas especificados | | | | | X |
| Seguridad: referido a la capacidad del producto software para proteger la información y los datos | | | | X | |
| Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad | | | | X | |
| 2. FIABILIDAD | | | | | |
| Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software. | | | X | | |
| Tolerancia a fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz. | | | X | | |
| Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo. | | | | X | |
| Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad. | | | X | | |
| 3. USABILIDAD | | | | | |
| Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso. | | | X | | |
| Facilidad de aprendizaje: la capacidad del producto software para permitir al usuario | | | | X | |

| | | | | | |
|---|--|--|---|---|--|
| aprender su aplicación. | | | | | |
| Atracción: la capacidad del producto software para atraer al usuario. | | | X | | |
| Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad. | | | | X | |
| Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle. | | | | X | |
| 4. EFICIENCIA | | | | | |
| Comportamiento temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas. | | | | X | |
| Utilización de recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones. | | | X | | |
| Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia. | | | X | | |
| 5. MANTENIBILIDAD | | | | | |
| Analizabilidad: Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas. | | | | X | |
| Cambiabilidad: Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación. | | | | X | |
| Estabilidad: Capacidad del producto software de evitar los efectos inesperados de las modificaciones. | | | | X | |
| Facilidad de prueba: Capacidad del producto software de permitir validar las partes modificadas. | | | | X | |
| Conformidad: Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad. | | | X | | |

| INDICADORES | PUNTUACIÓN | | | | |
|---|------------|---|----|----|------------|
| | 1 | 2 | 3 | 4 | 5 |
| 6. PORTABILIDAD | | | | | |
| Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado. | | | | X | |
| Facilidad de instalación: la capacidad del producto software para ser instalado en un ambiente determinado. | | | X | | |
| Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos. | | | X | | |
| Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente. | | | X | | |
| Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad. | | | X | | |
| SUB TOTALES | | | 39 | 52 | 10 |
| TOTAL | | | | | 101 |

Fuente: Ficha de evaluación ISO-9126

Tabla N° 03. Escala valorativa

| Indicador Cualitativo | Valor |
|-----------------------|-------|
| Deficiente | 1 |
| Malo | 2 |
| Regular | 3 |
| Bueno | 4 |
| Muy bueno | 5 |

| Clasificación | Intervalo | Decisión |
|--------------------------|---------------|----------|
| A) Inaceptable | [27 - 54 > | |
| B) Mínimamente aceptable | [54 - 81 > | |
| C) Aceptable | [81 - 95 > | |
| D) Cumple los requisitos | [95 - 122 > | 101 |
| E) Excede los requisitos | [122 - 135] | |

Conclusión: Por lo tanto cumple los requisitos por que 101 se encuentra dentro de los rangos de [95 – 122 > eso quiere decir que está cumpliendo con el Estándar ISO - 9126.

4.1.6. MANTENIMIENTO

El software deba adaptarse a cambios del entorno externo
(sistema operativo o dispositivos periféricos).

RECOMENDACIONES Y SUGERENCIAS

El sistema de Siscabana, es una aplicación que debe ejecutarse en plataforma como Windows XP o posterior.

Seguir utilizando la metodología XTREME PROGRAMMING XP, para el proceso de desarrollo de software, ya que sus criterios son:

Software que funcione es más importante que documentos exhaustiva, es decir si el software no funciona no vale nada.

La colaboración con el cliente es más importante que la negociación de contratos.

El éxito de los proyectos se basa en una realimentación permanente.

La respuesta ante el cambio es más importante que el siguiente de un plan.

BIBLIOGRAFÍA

SABANA, M. M. (2006). MODELAMIENTO E IMPLEMENTACIÓN DE BASE DE DATOS. 1ª ed. Lima: Grupo Editorial Megabyte S.A.C. Pág. 28.

DE LA CRUZ VILLAR, JHOEL. (2004). PHP y MySQL. (1ª ed.). Lima: Grupo Editorial Megabyte S.A.C. Pág. 205.

DE LA CRUZ VILLAR, JHOEL. (2004) PHP y MySQL. 1ª ed. Lima: Grupo Editorial Megabyte S.A.C. Pág. 3.

LUKE. W. y LAURA. T. (2005). Desarrollo Web con PHP y MySQL.

EDGAR PRIMER CASTILLO BARRIGA. “Desarrollo de Sistemas de Información Orientado a Objetos para la gestión del Registro Civil de la Municipalidad Provincial de chuchito-Juli”.2003.

RUBÉN BRAULIO FLORES MAMANI “Diseño e implementación de un sistema de registro civil en la municipalidad provincial de puno periodo 2005”

REFERENCIAS DE INTERNET

<<http://foro.chuidiang.com/metodologias-agiles/como-utilizo-la-metodologia-xp/html>> Consultado: [24, de marzo, 2012].

<<http://www.slideshare.net/NoeGonzalezMendoza/arquitectura-cliente-servidor>>, Consultado:[22, de marzo del 2012]

<<http://www.misrespuestas.com/que-es-hardware.html>> Consultado: [25, de Marzo, 2012.]

<<http://www.alarcos.incr.uclm.es/doc/cmsi/trabajos/Joaquin%20Ruiz%20Expo.html>>, Consultado: [22, de marzo, 2012.]

ANEXOS**ANEXO 1**

La fórmula que nos permitirá determinar el tamaño de muestra es por ser población finita (< 100.000 unidades de análisis).

$$n_0 = \frac{NZ_{\alpha/2}^2 PQ}{(N - 1)E^2 + Z_{\alpha/2}^2 PQ}$$

Dónde:

n_0 : Tamaño de la muestra.

$Z_{\alpha/2}^2$: Nivel de confianza.

P : Variabilidad Positiva.

Q : Variabilidad Negativa.

N : Población General.

E : Precisión o error.

Para esto hallamos la proporción de aceptación del desarrollo de Software de información, tomado de la encuesta piloto.

$$p = \frac{a}{n}$$

Remplazando la fórmula de los datos obtenidos de la encuesta piloto obtendremos el valor de p.

$$p = \frac{20}{30}$$

$$p = 0.67$$

Proporción de usuarios que si tan de acuerdo con el desarrollo del sistema de información para el área de Registro Civil.

Ahora hallamos el valor de que aplicando la siguiente formula.

$$q = 1 - p$$

$$q = 1 - 0.67$$

$$q = 0.33$$

Proporción de usuarios que no están de acuerdo con el desarrollo del sistema de información. Ahora remplazamos en la formula general los valores hallados:

$$n_0 = \frac{(5200)(1.96)^2 (0.67)(0.33)}{(5200 - 1)(0.1)^2 + (1.96)^2 (0.67)(0.33)}$$

$$n_0 = 100$$

Remplazando en la formula se obtuvo el tamaño de muestra siendo 100 persona.

ANEXO 2**VALIDACIÓN DEL SISTEMA DE INFORMACIÓN CON PRUEBA
ESTADÍSTICA****PRUEBA DE KOLGOMOROV-SMIRNOV**

Se desea comparar la diferencia de tiempos de atención a los usuarios del Registro Civil antes y después de implantar el sistema información.

TIPO DE ESTUDIO: longitudinal.

NIVEL INVESTIGATIVO: Relacional.

OBJETIVO ESTADÍSTICO: Numérico.

TIPO DE DISTRIBUCIÓN: Sin normalidad.

Se efectuó una prueba a 100 usuarios que realizaban su trámite de partidas nacimiento, defunciones o matrimonios para saber si su calidad de servicio de atención al cliente ha mejorado con el entrenamiento del sistema.

Se desea comparar la diferencia de tiempos de un grupo de clientes antes y después de un entrenamiento.

Tabla N° 02
Resultados de una prueba estadística no paramétrica

| | TIEMPO (EN MINUTOS) | | DIFERENCIA |
|----------------------------|---------------------|---------|------------|
| | ANTES | DESPUES | |
| N | 100 | 100 | 100 |
| MEDIA | 16.06 | 7.91 | - 8.15 |
| DESVIACION ESTANDAR | 1.399 | 1.207 | 1.888 |
| VARIANZA | 1.957 | 1.457 | 3.565 |
| Z (K-S) | 1.792 | 1.946 | 1.383 |
| P-VALOR | .003 | .001 | 0.044 |

EL RITUAL DE LA SIGNIFICACIÓN ESTADÍSTICA

1. PLANTEAR LA HIPÓTESIS

Hipótesis Nula: el tiempo de atención a los clientes después no difiere de la medida del tiempo antes en un periodo de entrenamiento del sistema.

Hipótesis Alternativa: el tiempo de atención a los clientes después difiere de la medida antes en un periodo de entrenamiento del sistema.

2. ESTABLECER EL NIVEL DE SIGNIFICANCIA.

$\alpha = 0.05$ o el 5 %.

3. SELECCIONAR ESTADÍSTICO DE PRUEBA.

Rangos de Wilcoxon.

4. VALOR DE “W” CALCULADO: -8.712

VALOR DE P CALCULADO: 0.000007

Interpretando dando respuesta a la hipótesis: Por lo tanto se acepta la hipótesis alterna quiere decir que el tiempo de atención después difiere de la medida antes de un periodo de entrenamiento del sistema

Interpretando p:

Como el valor de $P = 0.000007$ es menor que $\alpha = 0.05$, se rechaza la hipótesis nula y podemos concluir el tiempo después difiere de la evaluación antes de la prueba del sistema.

ANEXO 3

Tabla N° 01:
Ficha de evaluación de la calidad del producto.

| INDICADORES | PUNTUACIÓN | | | | |
|--|------------|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 1. FUNCIONALIDAD | | | | | |
| Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios. | | | | | |
| Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado. | | | | | |
| Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas especificados | | | | | |
| Seguridad: referido a la capacidad del producto software para proteger la información y los datos | | | | | |
| Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad | | | | | |
| 2. FIABILIDAD | | | | | |
| Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software. | | | | | |
| Tolerancia a fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz. | | | | | |
| Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo. | | | | | |
| Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad. | | | | | |
| 3. USABILIDAD | | | | | |
| Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso. | | | | | |
| Facilidad de aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación. | | | | | |
| Atracción: la capacidad del producto software | | | | | |

| | | | | | |
|---|-------------------|----------|----------|----------|----------|
| para atraer al usuario. | | | | | |
| Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad. | | | | | |
| Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle. | | | | | |
| 4. EFICIENCIA | | | | | |
| Comportamiento temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas. | | | | | |
| Utilización de recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones. | | | | | |
| Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia. | | | | | |
| 5. MANTENIBILIDAD | | | | | |
| Analizabilidad: Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas. | | | | | |
| Cambiabilidad: Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación. | | | | | |
| Estabilidad: Capacidad del producto software de evitar los efectos inesperados de las modificaciones. | | | | | |
| Facilidad de prueba: Capacidad del producto software de permitir validar las partes modificadas. | | | | | |
| Conformidad: Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad. | | | | | |
| INDICADORES | PUNTUACIÓN | | | | |
| | 1 | 2 | 3 | 4 | 5 |
| 6. PORTABILIDAD | | | | | |
| Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado. | | | | | |
| Facilidad de instalación: la capacidad del | | | | | |

| | | | | | |
|--|---|---|----|----|----|
| producto software para ser instalado en un ambiente determinado. | | | | | |
| Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos. | | | | | |
| Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente. | | | | | |
| Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad. | | | | | |
| SUB TOTALES | 1 | 2 | 21 | 44 | 35 |
| TOTAL | | | | | |

Fuente: Ficha de evaluación ISO-9126

Tabla N° 03:
Escala valorativa

| Indicador Cualitativo | Valor |
|-----------------------|-------|
| Deficiente | 1 |
| Malo | 2 |
| Regular | 3 |
| Bueno | 4 |
| Muy bueno | 5 |

Fuente: Escala valorativa ISO-912

ANEXO 4

MANUAL DE USUARIO DE SISTEMA DE REGISTRO CIVIL

INTRODUCCIÓN

El sistema de registro civil ha sido desarrollada para la municipalidad distrital de Cabana con el propósito de optimizar los las operaciones de registro civil.

El sistema está conformado por:

Un componente **Appserv** es una herramienta OpenSource para Windows con Apache, MySQL, PHP y otras adicciones, en la cual estas aplicaciones se configuran en forma automática, lo que permite ejecutar un servidor web completo.

El sistema de Registro Civil de la Municipalidad Distrital de Cabana

El presente manual ha sido preparado con la finalidad de ser una herramienta ágil, que facilite la mejor utilización de las opciones y posibilidades del sistema del registro civil.

SISTEMA CLIENTE-SERVIDOR.

1. REQUERIMIENTOS TÉCNICOS.

SISTEMA SERVIDOR.

Una computadora personal con las siguientes personales.

a) Requerimiento mínimo del hardware

- 512 Mb. de RAM.
- Microprocesador Pentium IV.
- 100 Mb. De espacio en disco duro.
- Monitor con resolución por 600*800 pixeles.

b) Requerimientos mínimos de software.

- Sistema operativo Windows XP 32 bits.
- APPSERV 5.0.

c) Otros.

- Acceso ha Intranet

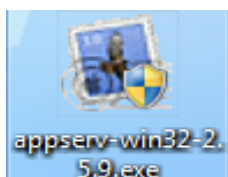
2. INSTALACIÓN DE COMPONENTES

Instalación De Appserv 2.5.9.

El software appserv 5.0 nos sirve para instalar un servidor Apache completamente configurado en tu PC, de manera que puedes crear tus páginas web añadiéndole contenido PHP sin tener que subirlas a un servidor.

Los pasos para la instalación y configuración se detallan a continuación:

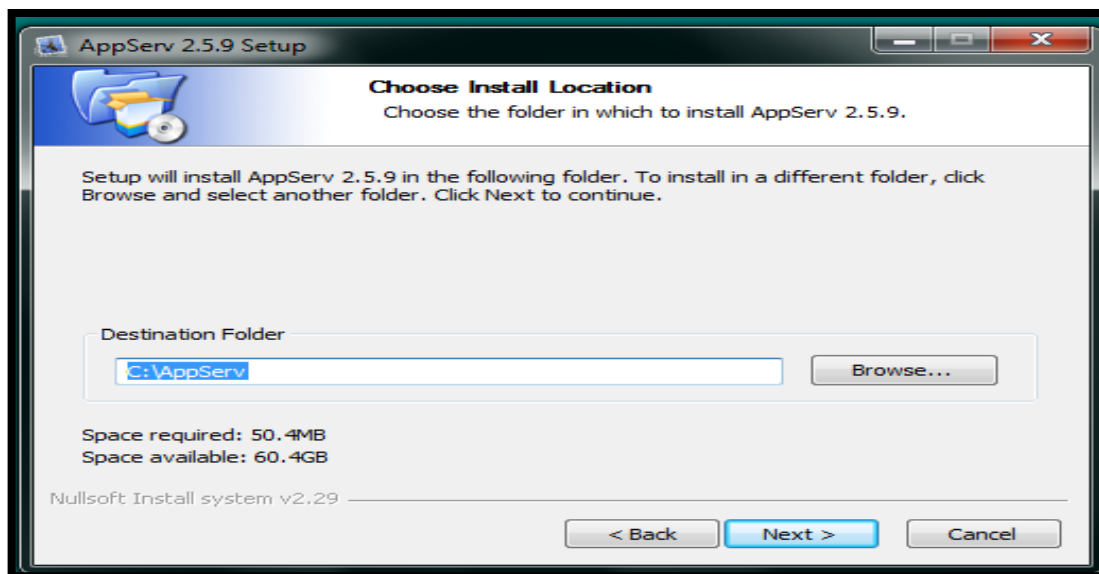
1. Ejecute el archivo. Appserv-win32-2.5.9.exe



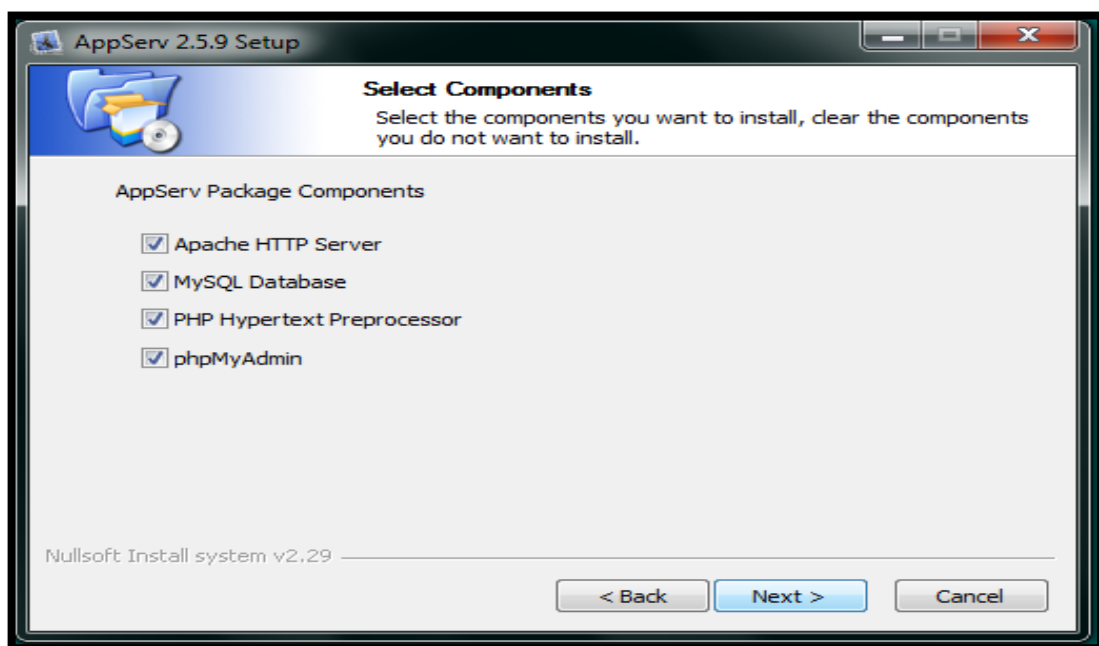
2. Una vez descargado el archivo, vamos a ejecutarlo. Y se lanzara el asistente de instalación AppServ.



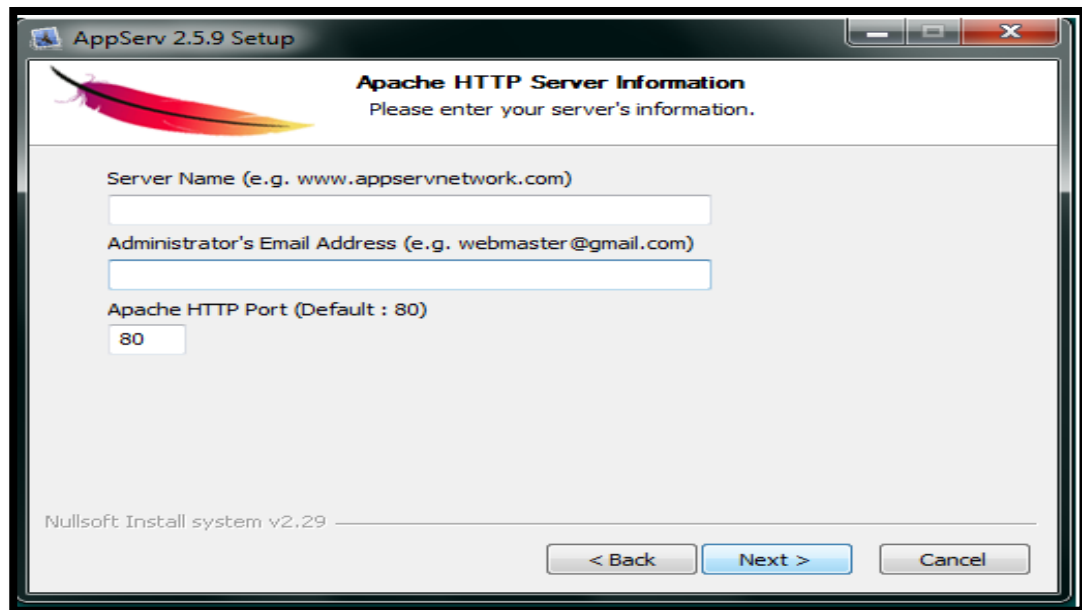
Le damos a <<Next>> para comenzar con la instalación. Aceptamos los términos, y luego especificamos la ruta donde queremos instalar el AppServ



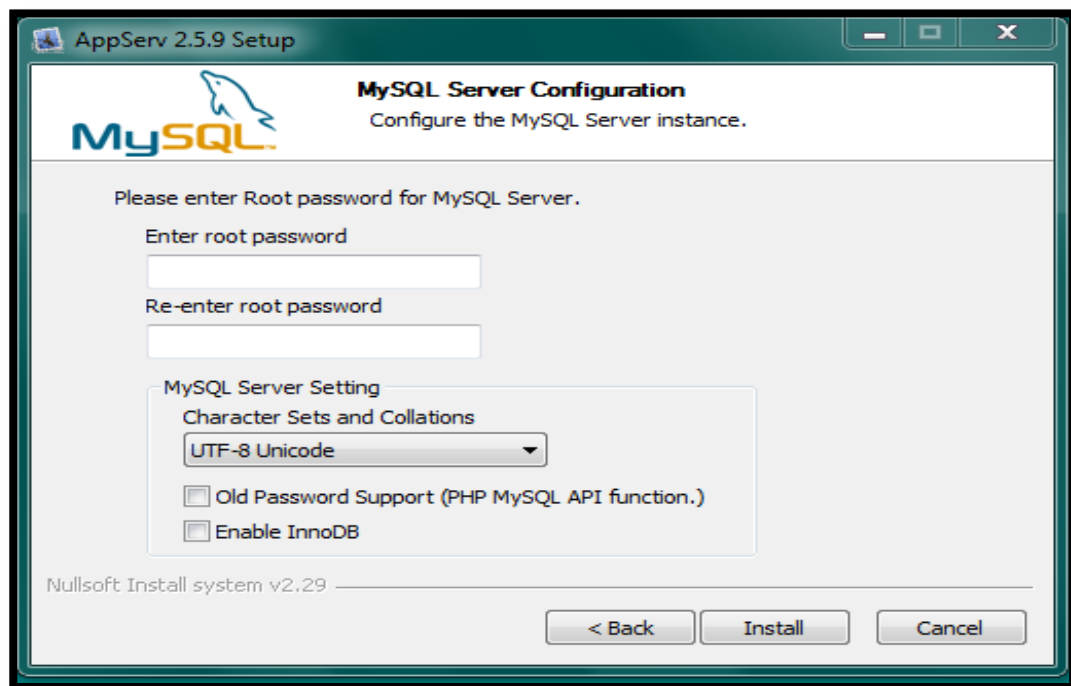
OJO: Elegir con mucho cuidado esta ruta, debido a que es donde deberás guardar tus archivos PHP. A continuación, deberás especificar qué componentes quieres instalar, lo que recomiendo es que los marques todos:



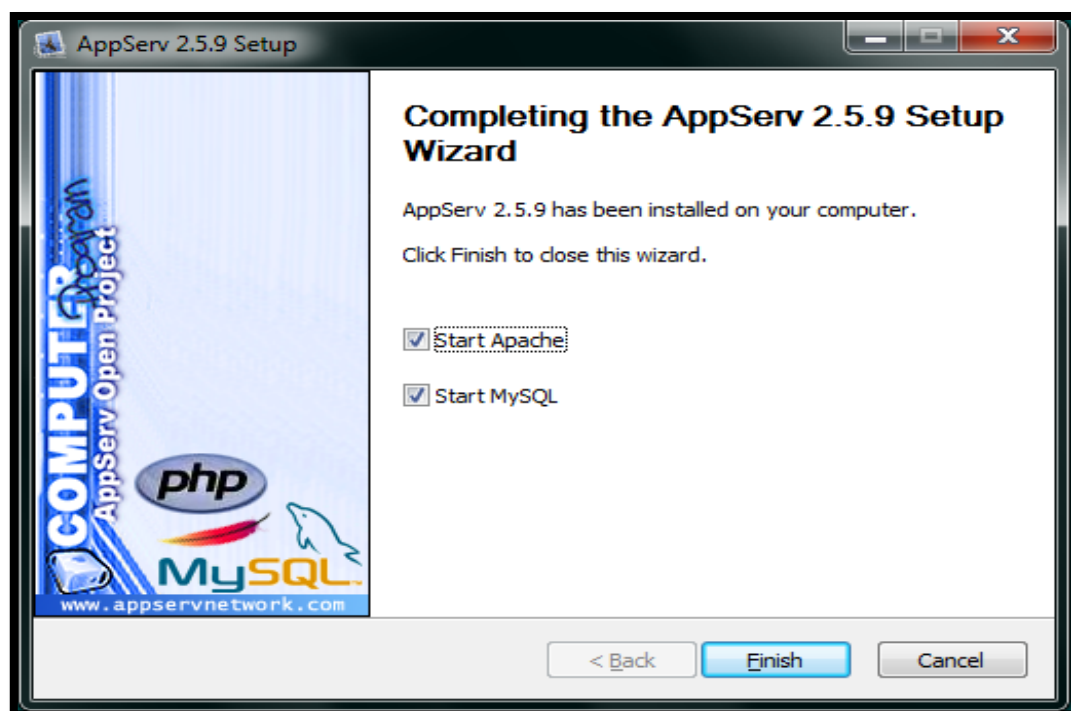
Una vez que has marcado los componentes que vas a usar, se te solicitará que escribas algunos datos del servidor, en nombre de servidor poner cualquier, al igual que en el e-mail del administrador. Lo único que no tienes que tocar es el puerto por defecto (80).



Ponemos la Contraseña para el acceso a la base de datos.



hora sólo tenemos que iniciar el servidor Apache y MySQL, marcando las casillas y dándole clic a <<Finish>>.



Si todo ha salido bien, entramos a la página siguiente <http://localhost/> nos mostrara tal como se muestra en la Figura.

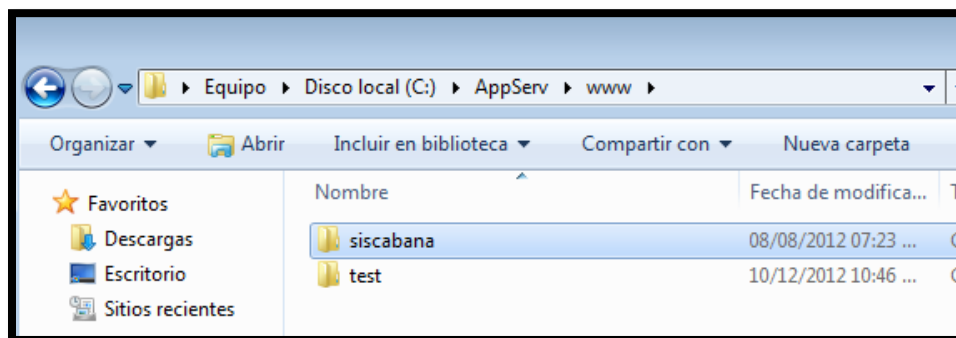


Y para entrar al phpMyAdmin, sólo basta con <http://localhost/phpmyadmin>.

3. INSTALACIÓN DEL SISTEMA DEL REGISTRO CIVIL

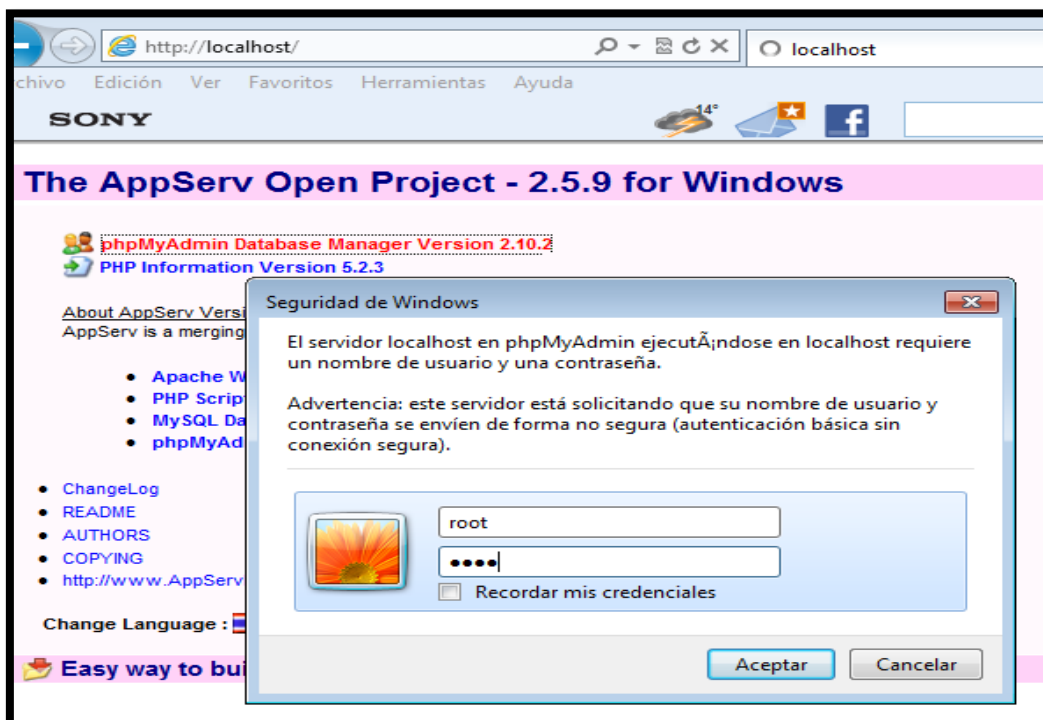
Los pasos necesarios para implementar el sistema son los siguientes:

- a) Copie la carpeta siscabana dentro de la carpeta www tal como se muestra en la imagen

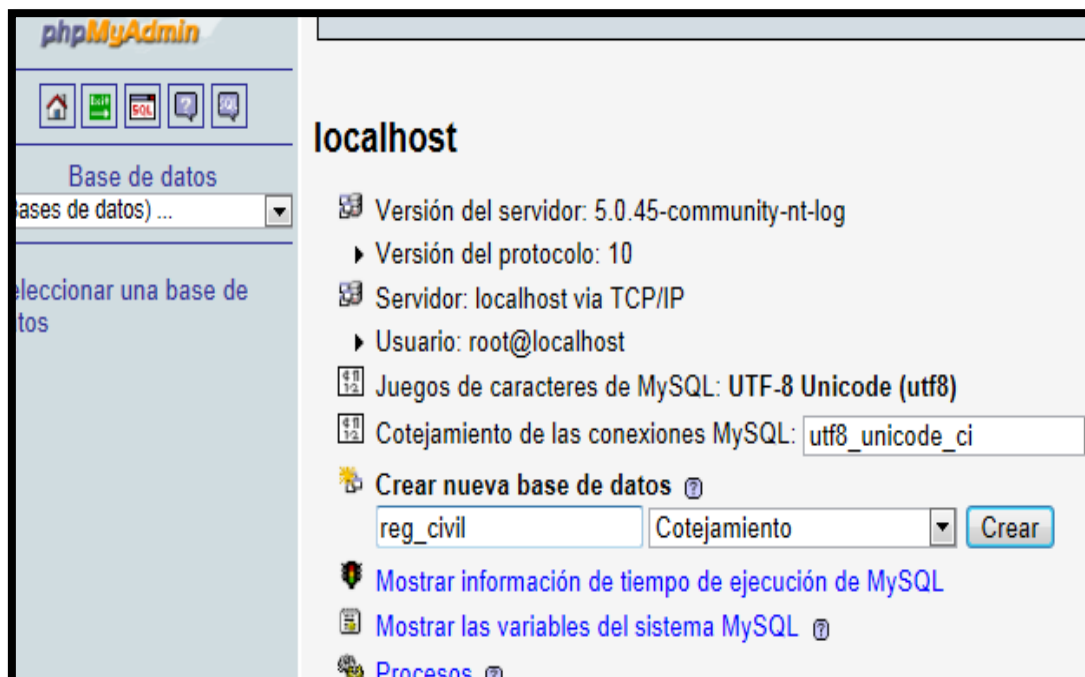


4. INSTALACIÓN DE LA BASE DE DATOS

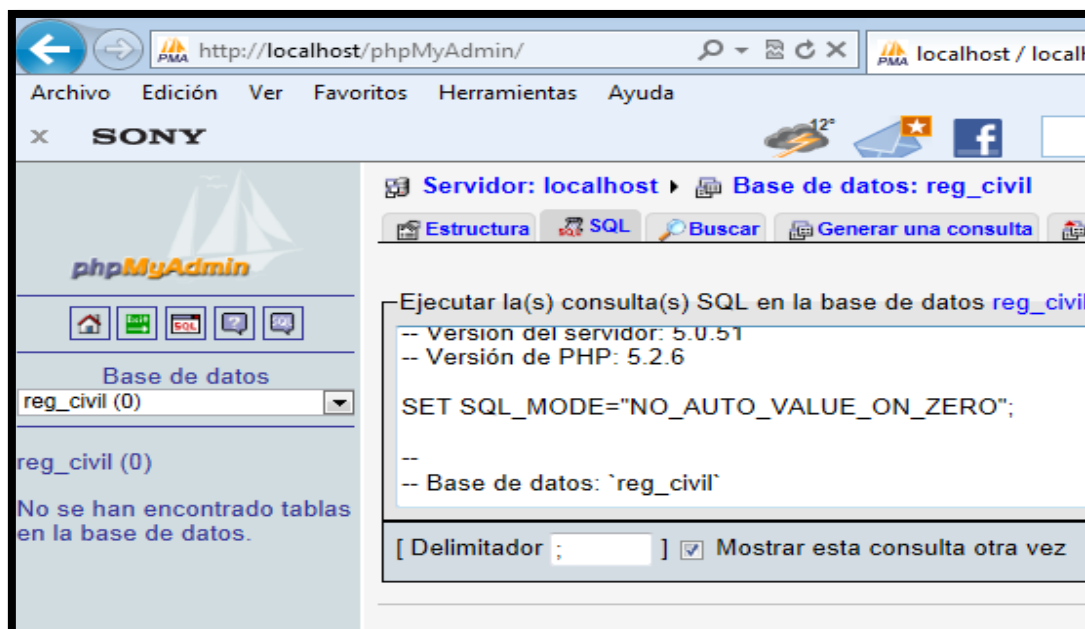
- a) Entramos a la siguiente dirección <http://localhost>
 - ✓ Seleccionamos el link phpMyAdmin Database.
 - ✓ Escribimos nuestro usuario y contraseña root / root.



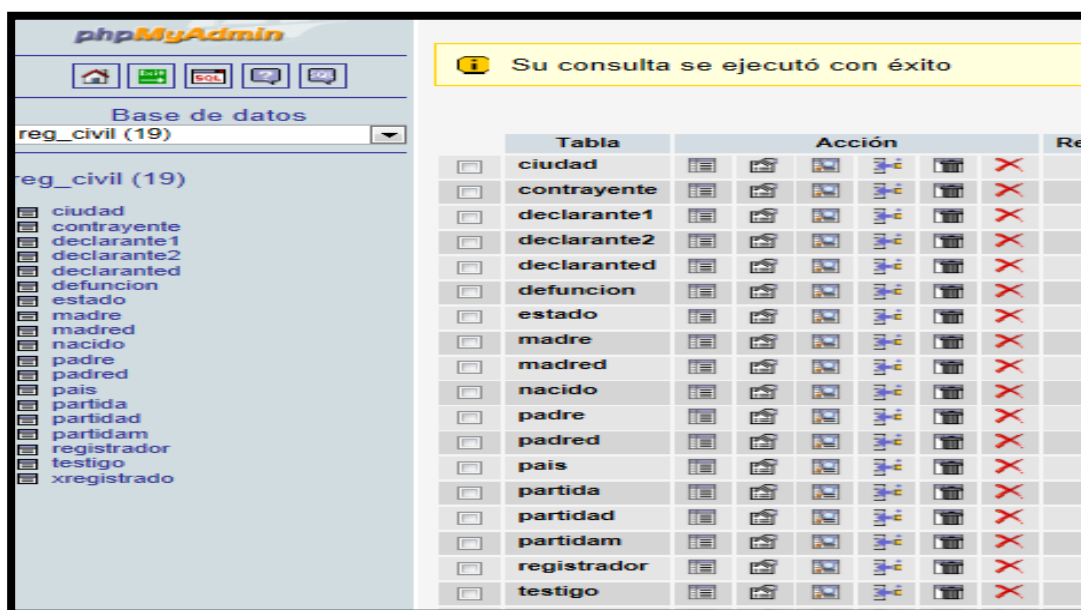
b) En el casillero Crear nueva base de datos escribimos “reg_civil” y presionamos Crear como se muestra en la Figura.



En Base de datos seleccionamos “reg_civil” luego entramos a SQL tal como y pegamos el contenido del archivo sql.txt que se encuentra en la carpeta “siscabana” y presionamos continuar tal como se muestra en la Figura.

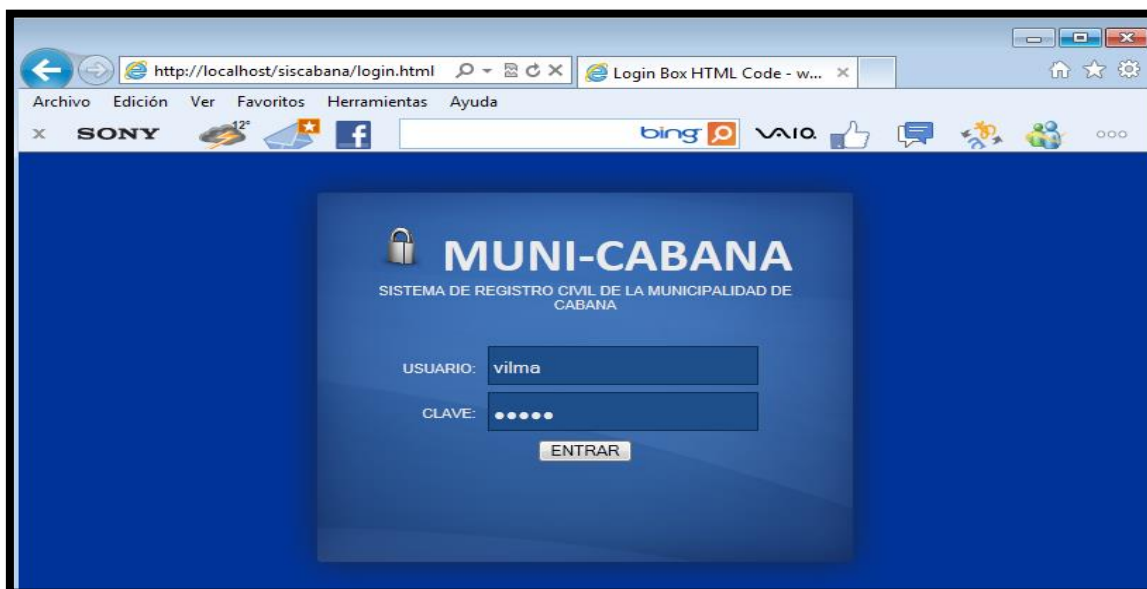


- La base de datos debería quedar tal como se muestra en la Figura.

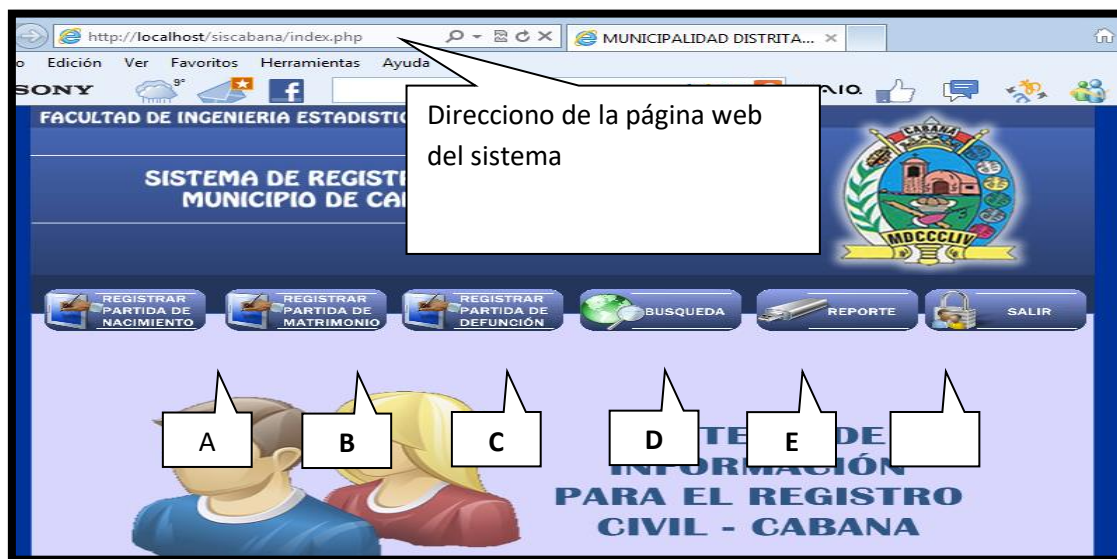


DESCRIPCIÓN DEL USO DEL SISTEMA DE REGISTRO CIVIL

LOGIN: Entramos a la siguiente dirección <http://localhost/siscabana/login.html> digitamos el usuario y clave del registrador proporcionado por el administrador de la BD para este ejemplo usaremos usuario “vilma” clave “vilma”



PARTES DEL SISTEMA



Se describen los controles que componen la interfaz:

A: Botón para el registro de nuevas partidas de nacimientos.

B: Botón para el registro de nuevas partidas de matrimonios.

C: Botón para el registro de nuevas partidas de defunciones.

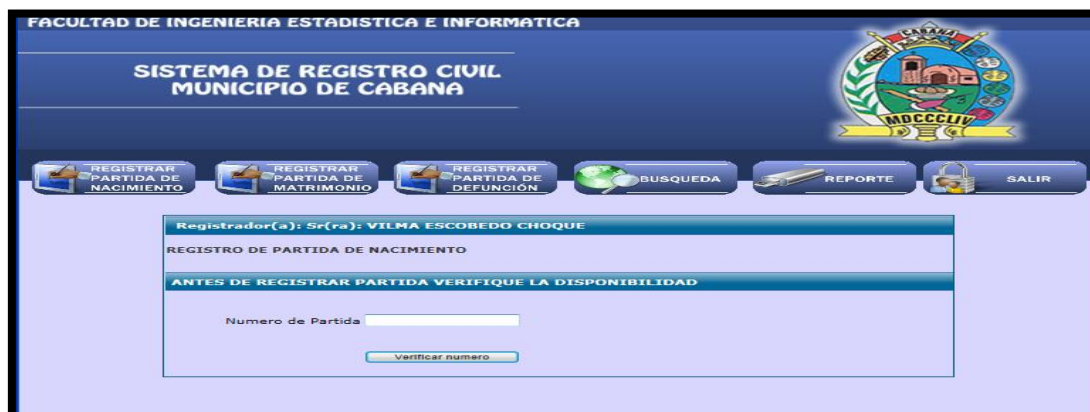
D: Búsqueda de partidas de matrimonio, nacimiento y defunciones para poder corregirlos o revisar datos.

E: Botón de reporte de partidas de matrimonio, nacimiento y defunciones para su impresión.

F: Botón para salir del sistema.

REGISTRÓ DE PARTIDA DE NACIMIENTO

Primeramente tienes que ingresar el número de partida de nacimiento para su verificación en el sistema si existe o no.



Luego llene los datos de la persona a registrar y presione el botón registrar para su almacenamiento en la BD.

Primeramente ingrese el número de la partida de matrimonio para su verificación si existe o no en el sistema.

Luego llene los datos de los contrayentes y presione el botón registrar para su almacenamiento en la BD.

REGISTRO DE PARTIDAS DE MATRIMONIOS >> REGISTRADOR: VILMA ESCOBEDO CHOQUE

Partida Numero

OFICINA REGISTRAL

Fecha(dd/mm/aaaa) Departamento Provincia Distrito C.P./Comunidad (**)

DATOS DEL CONTRAYENTE

Primer Apellido Segundo Apellido Pre-Nombres Edad Estado Civil

Nacionalidad Docum. Identidad Numero Doc. Depto Provincia

Distrito C.P./Comunidad (**)

DATOS DE LA CONTRAYENTE

Primer Apellido Segundo Apellido Pre-Nombres Edad Estado Civil

Nacionalidad Docum. Identidad Numero Doc. Depto Provincia

Distrito C.P./Comunidad (**)

TESTIGO

Primer Apellido Segundo Apellido Pre-Nombres Edad Docum. Identidad

Numero Doc. Nacionalidad

TESTIGO

Primer Apellido Segundo Apellido Pre-Nombres Edad Docum. Identidad

Numero Doc. Nacionalidad

REGISTRADOR(A)

Expediente Nro. Hora(hh:mm AM/PM) Fecha

Observaciones

Los campos señalados con (**) no son obligatorios, puede omitirlos.

REGISTRÓ DE PARTIDA DE DEFUNCIÓN

Primeramente ingrese el número de la partida de defunción para su verificación si existe o no en el sistema.

FACULTAD DE INGENIERIA ESTADISTICA E INFORMÁTICA

SISTEMA DE REGISTRO CIVIL MUNICIPIO DE CABANA

Registrador(a): Sr(ra): VILMA ESCOBEDO CHOQUE

REGISTRO DE PARTIDA DE DEFUNCIÓN

ANTES DE REGISTRAR PARTIDA VERIFIQUE LA DISPONIBILIDAD

Numero de Partida

Luego llene los datos del difunto y presione el botón registrar para su almacenamiento en la BD.

REGISTRO DE PARTIDAS DE DEFUNCIÓN >> REGISTRADOR: VILMA ESCOBEDO CHOQUE

Partida Numero

OFICINA REGISTRAL

Fecha(dd/mm/aaaa) Departamento Provincia Distrito C.P./Comunidad (*)

DATOS DEL DIFUNTO

Primer Apellido Segundo Apellido Pre-Nombres Sexo Edad

Docum. Identidad Numero Doc. Nacionalidad

Natural de:

Deplo Provincia Distrito C.P./Comunidad (*)

Lugar de Ocurrencia

Deplo Provincia Distrito C.P./Comunidad (*) Hora (hh:mm PM)

Fecha(dd/mm/aaaa) Centro de Fallec. Nombre y Dirección

Estado Civil Apell. y Nomb.(Cony)

DATOS DE LA MADRE

Primer Apellido Segundo Apellido Pre-Nombres Edad Nacionalidad

Natural de(prov.) Docum. Identidad Numero Doc.

DATOS DEL PADRE

Primer Apellido Segundo Apellido Pre-Nombres Edad Nacionalidad

Natural de(prov.) Docum. Identidad Numero Doc.

DECLARANTE

Primer Apellido Segundo Apellido Pre-Nombres Edad Nacionalidad

Vinculo Docum. Identidad Numero Doc.

Observaciones

Los campos señalados con (*) no son obligatorios, puede omitirlos.

BÚSQUEDA DE PARTIDAS

Las búsquedas pueden realizarse para las tres tipos de partida nacimientos, matrimonios y defunciones y para cada una de estas por número de partida, nombre y fecha de registro

SISTEMA DE REGISTRO CIVIL MUNICIPIO DE CABANA

REGISTRAR PARTIDA DE NACIMIENTO | REGISTRAR PARTIDA DE MATRIMONIO | REGISTRAR PARTIDA DE DEFUNCIÓN | **BUSQUEDA** | REPORTE | SALIR

Registrador(a): Sr(ra): VILMA ESCOBEDO CHOQUE

SELECCIONE UNA OPCION DE BUSQUEDA

TIPO DE PARTIDA BUSCAR POR

PARTIDA DE NACIMIENTO (Ingrese el numero de partida)

PARTIDA DE MATRIMONIO (Ingrese apellidos)

PARTIDA DE DEFUNCIÓN (Ingrese apellidos)

Fecha de nacimiento Fecha válida (dd/mm/aaaa)

ANEXO 5

CÓDIGO

Base_datos.sql

```
--
-- Estructura de tabla para la tabla `contrayente`
--
```

```
CREATE TABLE `contrayente` (
  `CNRO_PARTIDA` varchar(10) NOT NULL,
  `CPATERNO` varchar(25) NOT NULL,
  `CMATERNO` varchar(25) NOT NULL,
  `CNOMBRES` varchar(30) NOT NULL,
  `CSEXO` varchar(1) NOT NULL default "",
  `CEDAD` varchar(2) NOT NULL,
  `CESTADO_CIVIL` varchar(12) NOT NULL,
  `CTIPO_DOC` varchar(20) NOT NULL,
  `CNRO_DOC` varchar(11) NOT NULL,
  `CNACIONALIDAD` varchar(20) NOT NULL,
  `CDEPARTAMENTO` varchar(20) NOT NULL,
  `CPROVINCIA` varchar(25) NOT NULL,
  `CDISTRITO` varchar(30) NOT NULL,
  `CCOMUNIDAD` varchar(40) default NULL,
  PRIMARY KEY (`CNRO_PARTIDA`,`CNRO_DOC`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

```
--
-- Estructura de tabla para la tabla `declarante1`
--
```

```
CREATE TABLE `declarante1` (
  `DPATERNO` varchar(20) default NULL,
  `DMATERNO` varchar(20) default NULL,
  `DNOMBRES` varchar(30) NOT NULL,
  `DEDAD` varchar(2) NOT NULL,
  `DNACIONALIDAD` varchar(20) NOT NULL,
  `DVINCULO` varchar(30) NOT NULL,
  `DTIPO_DOC` varchar(20) default NULL,
  `DNRO_DOC` varchar(11) default NULL,
  `DNRO_PARTIDA1` varchar(10) NOT NULL,
  PRIMARY KEY (`DNRO_PARTIDA1`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

```
--
-- Volcar la base de datos para la tabla `declarante1`
--
```

```
INSERT INTO `declarante1` VALUES ('CALVIN', 'JALLO', 'MARGARITA', '32', 'PERUANA', 'MADRE', 'DNI',
'01245783', '456');
INSERT INTO `declarante1` VALUES ('MANTILLA', 'BELAUNDE', 'MARIA FERNANDA', '21', 'PERUANA',
'TIA', 'DNI', '01010101', '1000');
```

```

-----
--
-- Estructura de tabla para la tabla `declarante2`
--
CREATE TABLE `declarante2` (
  `DPATERNO` varchar(20) default NULL,
  `DMATERNO` varchar(20) default NULL,
  `DNOMBRES` varchar(30) NOT NULL,
  `DEDAD` varchar(2) NOT NULL,
  `DNACIONALIDAD` varchar(20) NOT NULL,
  `DVINCULO` varchar(30) NOT NULL,
  `DTIPO_DOC` varchar(20) default NULL,
  `DNRO_DOC` varchar(11) default NULL,
  `DNRO_PARTIDA2` varchar(10) NOT NULL,
  PRIMARY KEY (`DNRO_PARTIDA2`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Volcar la base de datos para la tabla `declarante2`
--

INSERT INTO `declarante2` VALUES ('CARPIO', 'TERRAZAS', 'LUIS MIGUEL', '35', 'EXTRANJERA', 'PADRE',
'CARNE EXTRANJERIA', '7894612', '456');
INSERT INTO `declarante2` VALUES ('GARCIA', 'TOLEDO', 'ENRIQUE', '24', 'PERUANA', 'TIO', 'DNI',
'02020202', '1000');

-----
--
-- Estructura de tabla para la tabla `declaranted`
--
CREATE TABLE `declaranted` (
  `DPATERNO` varchar(20) default NULL,
  `DMATERNO` varchar(20) default NULL,
  `DNOMBRES` varchar(30) NOT NULL,
  `DEDAD` varchar(2) NOT NULL,
  `DNACIONALIDAD` varchar(20) default NULL,
  `DVINCULO` varchar(30) NOT NULL,
  `DTIPO_DOC` varchar(20) default NULL,
  `DNRO_DOC` varchar(11) default NULL,
  `DDNRO_PARTIDA` varchar(10) NOT NULL,
  PRIMARY KEY (`DDNRO_PARTIDA`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Volcar la base de datos para la tabla `declaranted`
--

INSERT INTO `declaranted` VALUES ('DEL CASTILLO', 'GALVEZ', 'JORGE', '60', 'PERUANA', 'AMIGO', 'DNI',
'56235689', '1000');
-----

```

```
--
-- Estructura de tabla para la tabla `defuncion`
--
```

```
CREATE TABLE `defuncion` (
  `DNRO_PARTIDA` varchar(10) NOT NULL default '',
  `DPATERNO` varchar(25) default NULL,
  `DMATERNO` varchar(25) default NULL,
  `DNOMBRES` varchar(30) NOT NULL,
  `DSEXO` varchar(10) NOT NULL,
  `DEDAD` varchar(3) NOT NULL,
  `DTIPO_DOC` varchar(20) default NULL,
  `DNRO_DOC` varchar(11) default NULL,
  `DNACIONALIDAD` varchar(20) default NULL,
  `DNDEPARTAMENTO` varchar(20) default NULL,
  `DNPROVINCIA` varchar(25) default NULL,
  `DNDISTRITO` varchar(30) default NULL,
  `DNCOMUNIDAD` varchar(40) default NULL,
  `DODEPARTAMENTO` varchar(20) default NULL,
  `DOPROVINCIA` varchar(25) default NULL,
  `DODISTRITO` varchar(30) default NULL,
  `DOCOMUNIDAD` varchar(40) default NULL,
  `DHORA` varchar(10) NOT NULL,
  `DFECHA` date NOT NULL,
  `DFECHA_LETRAS` varchar(40) NOT NULL,
  `DCENTRO_FALL` varchar(20) NOT NULL,
  `DNOMBR_C_FALL` varchar(40) NOT NULL,
  `DECONYUGAL` varchar(13) NOT NULL,
  `DCONYUGUE` varchar(50) default NULL,
  `DREG_FECHA` date NOT NULL,
  `DREG_DEP` varchar(25) NOT NULL,
  `DREG_PRO` varchar(25) NOT NULL,
  `DREG_DIS` varchar(25) default NULL,
  `DREG_CPOB` varchar(40) default NULL,
  `DPARTIDA` varchar(20) default NULL,
  PRIMARY KEY (`DNRO_PARTIDA`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_FORMAT=FIXED;
```

```
--
-- Volcar la base de datos para la tabla `defuncion`
--
```

```
INSERT INTO `defuncion` VALUES ('1000', 'GARCIA', 'PEREZ', 'ALAN', 'MASCULINO', '52', 'DNI',
'45128956', 'PERUANA', '21', '175', '1703', 'LIMA', '4', '36', '362', 'LIMA', '20:56 PM', '0006-08-09', '',
'HOSPITAL', 'ALMENARA', 'CASADO', 'NORES MAMANI DE GARCIA', '0006-10-09', '4', '36', '363', 'LIMA',
'1000d.jpg');
```

```
-----
```

```
--
-- Estructura de tabla para la tabla `estado`
--
```

```
CREATE TABLE `estado` (
  `idestado` double(53,0) default NULL,
  `estado` varchar(255) default NULL,
  `pais` double(53,0) default NULL
```

```

) ENGINE=MyISAM DEFAULT CHARSET=utf8;

-----

--
-- Estructura de tabla para la tabla `madre`
--

CREATE TABLE `madre` (
  `MPATERNO` varchar(20) default NULL,
  `MMATERNO` varchar(20) default NULL,
  `MNOMBRES` varchar(30) NOT NULL,
  `MEDAD` varchar(2) NOT NULL,
  `MNACIONALIDAD` varchar(20) NOT NULL,
  `MPROV_PROC` varchar(30) NOT NULL,
  `MTIPO_DOC` varchar(20) default NULL,
  `MNRO_DOC` varchar(11) default NULL,
  `MDOMICILIO` varchar(40) default NULL,
  `MVINCULO` varchar(20) NOT NULL default 'MADRE',
  `MNRO_PARTIDA` varchar(10) NOT NULL,
  PRIMARY KEY (`MNRO_PARTIDA`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Volcar la base de datos para la tabla `madre`
--

INSERT INTO `madre` VALUES ('CALVIN', 'JALLO', 'MARGARITA', '32', 'PERUANA', 'JULIACA', 'DNI',
'01245783', 'JR. BOLIVAR 245', 'MADRE', '456');
INSERT INTO `madre` VALUES ('MANTILLA', 'BELAUNDE', 'MARIA FERNANDA', '21', 'PERUANA',
'CASTILLA', 'DNI', '01010101', 'CHANU CHANU II ETAPA L-5', 'MADRE', '1000');

-----

--
-- Estructura de tabla para la tabla `madred`
--

CREATE TABLE `madred` (
  `DMPATERNO` varchar(20) default NULL,
  `DMMATERNO` varchar(20) default NULL,
  `DMNOMBRES` varchar(30) NOT NULL,
  `DMEDAD` varchar(2) NOT NULL,
  `DMNACIONALIDAD` varchar(20) NOT NULL,
  `DMPROV_PROC` varchar(30) NOT NULL,
  `DMTIPO_DOC` varchar(20) default NULL,
  `DMNRO_DOC` varchar(11) default NULL,
  `DMNRO_PARTIDA` varchar(10) NOT NULL,
  PRIMARY KEY (`DMNRO_PARTIDA`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Volcar la base de datos para la tabla `madred`
--

INSERT INTO `madred` VALUES ('PEREZ', 'TORRES', 'JUANA', '56', 'PERUANA', 'CUSCO', 'DNI', '56895623',
'1000');

```

--
-- Estructura de tabla para la tabla `nacido`
--

```
CREATE TABLE `nacido` (
  `NNRO_PARTIDA` varchar(10) NOT NULL default "",
  `NPATERNO` varchar(25) default NULL,
  `NMATERNO` varchar(25) default NULL,
  `NNOMBRES` varchar(30) NOT NULL,
  `NSEXO` varchar(10) NOT NULL,
  `NDEPARTAMENTO` varchar(20) default NULL,
  `NPROVINCIA` varchar(25) default NULL,
  `NDISTRITO` varchar(30) default NULL,
  `NCOMUNIDAD` varchar(40) default NULL,
  `NHORA` varchar(10) NOT NULL,
  `NFECHA` date NOT NULL,
  `NFECHA_LETRAS` varchar(40) NOT NULL,
  `NCENTRO_NACI` varchar(20) NOT NULL,
  `NNOMBRE_NACI` varchar(40) NOT NULL,
  `NREG_FECHA` date NOT NULL,
  `NREG_DEP` varchar(25) NOT NULL,
  `NREG_PRO` varchar(25) NOT NULL,
  `NREG_DIS` varchar(25) default NULL,
  `NREG_CPOB` varchar(40) default NULL,
  `PARTIDA` varchar(20) default NULL,
  PRIMARY KEY (`NNRO_PARTIDA`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 ROW_FORMAT=FIXED;
```

--
-- Volcar la base de datos para la tabla `nacido`
--

```
INSERT INTO `nacido` VALUES ('456', 'CARPIO', 'CALVIN', 'JUAN ALBERTO', 'MASCULINO', '20', '173',
'1693', 'DESCONOCIDA', '07:60 AM', '1986-01-21', 'NO ESPECIFICA', 'HOSPITAL', 'MANUEL NUÑES
BUTRON', '1987-01-01', '20', '161', '1597', 'DESCONOCIDA', NULL);
INSERT INTO `nacido` VALUES ('1000', 'GARCIA', 'MANTILLA', 'ALAN JESUS1', 'MASCULINO', '6', '57', '581',
'LAS CASUALINAS', '12:53 PM', '0006-01-09', 'NO ESPECIFICA', 'CLINICA', 'SAN JOSE DE ASIS', '0006-01-
09', '7', '71', '713', 'LAS CASUALINAS', '1000n.jpg');
```

--
-- Estructura de tabla para la tabla `padre`
--

```
CREATE TABLE `padre` (
  `PPATERNO` varchar(20) default NULL,
  `PMATERNO` varchar(20) default NULL,
  `PNOMBRES` varchar(30) NOT NULL,
  `PEDAD` varchar(2) NOT NULL,
  `PNACIONALIDAD` varchar(20) NOT NULL,
  `PPROV_PROC` varchar(30) NOT NULL,
  `PTIPO_DOC` varchar(20) default NULL,
  `PNRO_DOC` varchar(11) default NULL,
```

```

`PVINCULO` varchar(20) NOT NULL default 'PADRE',
`PNRO_PARTIDA` varchar(10) NOT NULL,
PRIMARY KEY (`PNRO_PARTIDA`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

--
-- Volcar la base de datos para la tabla `padre`
--

INSERT INTO `padre` VALUES ('CARPIO', 'TERRAZAS', 'LUIS MIGUEL', '35', 'EXTRANJERA', 'MADRID',
'CARNE EXTRANJERIA', '7894612', 'PADRE', '456');
INSERT INTO `padre` VALUES ('GARCIA', 'TOLEDO', 'ENRIQUE', '24', 'PERUANA', 'CALLO', 'DNI',
'02020202', 'PADRE', '1000');

-----

--
-- Estructura de tabla para la tabla `padred`
--

CREATE TABLE `padred` (
  `DPPATERNO` varchar(20) default NULL,
  `DPMATERNO` varchar(20) default NULL,
  `DPNOMBRES` varchar(30) NOT NULL,
  `DPEIDAD` varchar(2) NOT NULL,
  `DPNACIONALIDAD` varchar(20) NOT NULL,
  `DPPROV_PROC` varchar(30) NOT NULL,
  `DPTIPO_DOC` varchar(20) default NULL,
  `DPNRO_DOC` varchar(11) default NULL,
  `DPNRO_PARTIDAD` varchar(10) NOT NULL,
  PRIMARY KEY (`DPNRO_PARTIDAD`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

```

Registro.php

```

1. <?php require_once('Connections/cnx.php'); ?>
2. <?php
3. function GetSQLValueString($theValue, $theType, $theDefinedValue = "", $theNotDefinedValue = "")
4. {
5.     $theValue = (!get_magic_quotes_gpc()) ? addslashes($theValue) : $theValue;
6.     switch ($theType) {
7.         case "text":
8.             $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
9.             break;
10.        case "long":
11.        case "int":
12.            $theValue = ($theValue != "") ? intval($theValue) : "NULL";
13.            break;
14.        case "double":
15.            $theValue = ($theValue != "") ? "'" . doubleval($theValue) . "'" : "NULL";
16.            break;
17.        case "date":
18.            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
19.            break;
20.        case "defined":
21.            $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
22.            break;
23.        }
24.        return $theValue;
25.    }
26.    $editFormAction = $_SERVER['PHP_SELF'];
27.    if (isset($_SERVER['QUERY_STRING'])) {

```

```

28. $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);
29. }
30. if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
31. $insertSQL = sprintf("INSERT INTO nacido (NPATERNO, NMATERNO, NNOMBRES, NSEXO, NDEPARTAMENTO,
    NPROVINCIA, NDISTRICTO, NCOMUNIDAD, NHORA, NFECHA, NFECHA_LETRAS, NCENTRO NACI, NNOMBRE NACI,
    NREG_FECHA, NREG_DEP, NREG_PRO, NREG_DIS, NREG_CPOB, NNRO_PARTIDA) VALUES (%s, %s, %s, %s, %s, %s,
    %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)",
32. GetSQLValueString($_POST['npaterno'], "text"),
33. GetSQLValueString($_POST['nmaterno'], "text"),
34. GetSQLValueString($_POST['nnombres'], "text"),
35. GetSQLValueString($_POST['nsexo'], "text"),
36. GetSQLValueString($_POST['cb01'], "text"),
37. GetSQLValueString($_POST['cb02'], "text"),
38. GetSQLValueString($_POST['cb03'], "text"),
39. GetSQLValueString($_POST['ncomunidad'], "text"),
40. GetSQLValueString($_POST['nhora'], "text"),
41. GetSQLValueString($_POST['nfecha'], "date"),
42. GetSQLValueString($_POST['nfecha_letras'], "text"),
43. GetSQLValueString($_POST['ncentro'], "text"),
44. GetSQLValueString($_POST['nnombre_dir'], "text"),
45. GetSQLValueString($_POST['nfechareg'], "date"),
46. GetSQLValueString($_POST['pais'], "text"),
47. GetSQLValueString($_POST['estado'], "text"),
48. GetSQLValueString($_POST['ciudad'], "text"),
49. GetSQLValueString($_POST['comunidad'], "text"),
50. GetSQLValueString($_POST['nropartida'], "text"));
51. mysql_select_db($database_cnx, $cnx);
52. $Result1 = mysql_query($insertSQL, $cnx) or die(mysql_error());
53. }
54. if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
55. $insertSQL = sprintf("INSERT INTO madre (MPATERNO, MMATERNO, MNOMBRES, MEDAD, MNACIONALIDAD,
    MPROV_PROC, MTIPO_DOC, MNRO_DOC, MDOMICILIO, MNRO_PARTIDA) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s,
    %s, %s)",
56. GetSQLValueString($_POST['mpaterno'], "text"),
57. GetSQLValueString($_POST['mmaterno'], "text"),
58. GetSQLValueString($_POST['mnombres'], "text"),
59. GetSQLValueString($_POST['medad'], "text"),
60. GetSQLValueString($_POST['mnacionalidad'], "text"),
61. GetSQLValueString($_POST['mnatural'], "text"),
62. GetSQLValueString($_POST['mtipo_doc'], "text"),
63. GetSQLValueString($_POST['mnro_doc'], "text"),
64. GetSQLValueString($_POST['mdireccion'], "text"),
65. GetSQLValueString($_POST['nropartida'], "text"));
66. mysql_select_db($database_cnx, $cnx);
67. $Result2 = mysql_query($insertSQL, $cnx) or die(mysql_error());
68. }
69. if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
70. $insertSQL = sprintf("INSERT INTO padre (PPATERNO, PMATERNO, PNOMBRES, PEDAD, PNACIONALIDAD,
    PPROV_PROC, PTIPO_DOC, PNRO_DOC, PNRO_PARTIDA) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)",
71. GetSQLValueString($_POST['ppaterno'], "text"),
72. GetSQLValueString($_POST['pmaterno'], "text"),
73. GetSQLValueString($_POST['pnombres'], "text"),
74. GetSQLValueString($_POST['pedad'], "text"),
75. GetSQLValueString($_POST['pnacionalidad'], "text"),
76. GetSQLValueString($_POST['pnatural'], "text"),
77. GetSQLValueString($_POST['ptipo_doc'], "text"),
78. GetSQLValueString($_POST['pnro_doc'], "text"),
79. GetSQLValueString($_POST['nropartida'], "text"));
80. mysql_select_db($database_cnx, $cnx);
81. $Result3 = mysql_query($insertSQL, $cnx) or die(mysql_error());
82. }
83. if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
84. $insertSQL = sprintf("INSERT INTO declarantel (DPATERNO, DMATERNO, DNOMBRES, DEDAD, DNACIONALIDAD,
    DVINCULO, DTIPO_DOC, DNRO_DOC, DNRO_PARTIDA) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)",
85. GetSQLValueString($_POST['d0paterno'], "text"),
86. GetSQLValueString($_POST['d0materno'], "text"),
87. GetSQLValueString($_POST['d0nombres'], "text"),
88. GetSQLValueString($_POST['d0edad'], "text"),
89. GetSQLValueString($_POST['d0nacionalidad'], "text"),
90. GetSQLValueString($_POST['d0vinculo'], "text"),
91. GetSQLValueString($_POST['d0tipo_doc'], "text"),
92. GetSQLValueString($_POST['d0nro_doc'], "text"),

```



```

93. GetSQLValueString($_POST['nropartida'], "text");
94. mysql_select_db($database_cnx, $cnx);
95. $Result4 = mysql_query($insertSQL, $cnx) or die(mysql_error());
96. }
97. if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
98. $insertSQL = sprintf("INSERT INTO declarante2 (DPATERNO, DMATERNO, DNOMBRES, DEDAD, DNACIONALIDAD,
    DVINCULO, DTIPO_DOC, DNRO_DOC, DNRO_PARTIDA2) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)",
99. GetSQLValueString($_POST['dlpaterno'], "text"),
100. GetSQLValueString($_POST['dlmaterno'], "text"),
101. GetSQLValueString($_POST['dlnombres'], "text"),
102. GetSQLValueString($_POST['dleedad'], "text"),
103. GetSQLValueString($_POST['dlnacionalidad'], "text"),
104. GetSQLValueString($_POST['dlvinculo'], "text"),
105. GetSQLValueString($_POST['dltipo_doc'], "text"),
106. GetSQLValueString($_POST['dlnro_doc'], "text"),
107. GetSQLValueString($_POST['nropartida'], "text"));
108. mysql_select_db($database_cnx, $cnx);
109. $Result5 = mysql_query($insertSQL, $cnx) or die(mysql_error());
110. }
111. if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
112. $insertSQL = sprintf("INSERT INTO partida (NNRO_PARTIDA, MNRO_PARTIDA, PNRO_PARTIDA, DNRO_PARTIDA1,
    DNRO_PARTIDA2, RDNI, OBSP) VALUES (%s, %s, %s, %s, %s, %s, %s)",
113. GetSQLValueString($_POST['nropartida'], "text"),
114. GetSQLValueString($_POST['nropartida'], "text"),
115. GetSQLValueString($_POST['nropartida'], "text"),
116. GetSQLValueString($_POST['nropartida'], "text"),
117. GetSQLValueString($_POST['nropartida'], "text"),
118. GetSQLValueString($_POST['rdni'], "text"),
119. GetSQLValueString($_POST['obsp'], "text"));
120. mysql_select_db($database_cnx, $cnx);
121. $Result6 = mysql_query($insertSQL, $cnx) or die(mysql_error());
122. }
123. if ($Result1 && $Result2 && $Result3 && $Result4 && $Result5 && $Result6)
124. header("Location: confirma.php");
125. ?>

```

Index.html

```

1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml">
3. <head>
4. <SCRIPT language=Javascript src="funciones.js"></SCRIPT>
5.
6. <link href="css/faces.css" rel="stylesheet" type="text/css">
7. <link href="css/frames.css" rel="stylesheet" type="text/css">
8. <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
9. <title>Documento sin título</title>
10. <style type="text/css">
11. .Estilol { color: #FF0000;
12. font-weight: bold;
13. }
14. </style>
15. </head>
16. <body>
17. <table class="bordeazulosuro" width="662" border="0" align="center" cellpadding="1">
18. <tr>
19. <td colspan="4" class="footer_cab">&nbsp;<span class="letras_cabana"><strong>Registrador (a):
    Sr (ra): <? echo $_SESSION['pne']. " ".$_SESSION['pae']. "
    ".$_SESSION['sae']; ?></strong></span></td>
20. </tr>
21. <tr>
22. <td align="center" colspan="4" ><p class="letras_cabana"><strong><br />
23. BUSCADOR DE PARTIDA SEGUN CODIGO</strong></p>
24. <form id="form1" name="form1" method="post" action="pdfs/printpdf.php">
25. <p>
26. <input type="text" name="partida" />
27. </p>
28. <p>
29. <input type="submit" name="hh" id="hh" value="Buscar" />
30. </p>
31. </form>
32. <p class="letras_cabana"></p></td>

```

```

33. </tr>
34. <tr>
35. <td colspan="4" class="footer_"><span class="letras_cabana">CUADRO ESTADISTICO</span></td>
36. </tr>
37. <tr>
38. <td width="271" class="menu_default2_on">&nbsp;</td>
39. <td width="88" class="menu_default2_on">&nbsp;</td>
40. <td width="65">&nbsp;</td>
41. <td width="223">&nbsp;</td>
42. </tr>
43. <tr>
44. <td class="menu_default2_on">&nbsp;</td>
45. <td class="menu_default2_on">&nbsp;</td>
46. <td>&nbsp;</td>
47. <td>&nbsp;</td>
48. </tr>
49. <tr>
50. <td><div align="right"></div></td>
51. <td></td>
52. <td>&nbsp;</td>
53. <td>&nbsp;</td>
54. </tr>
55. <tr>
56. <td align="center" colspan="4"><p class="letras_subparrafo">HECHOS VITALES POR MESES 2012</p>
57. <p class="letras_subparrafo">&nbsp;</p></td>
58. </tr>
59. <tr>
60. <td>&nbsp;</td>
61. <td class="userpanel">Enero</td>
62. <td class="userpanel"><a href="reporte/enero_2012.doc" target="blank" class="enlace_imagen"> </a></td>
63. <td>&nbsp;</td>
64. </tr>
65. <tr>
66. <td>&nbsp;</td>
67. <td class="letras_subparrafo">Febrero</td>
68. <td></td>
69. <td>&nbsp;</td>
70. </tr>
71. <tr>
72. <td></td>
73. <td class="userpanel">Marzo</td>
74. <td class="userpanel"></td>
75. <td>&nbsp;</td>
76. </tr>
77. <tr>
78. <td></td>
79. <td class="letras_subparrafo">Abril</td>
80. <td></td>
81. <td>&nbsp;</td>
82. </tr>
83. <tr>
84. <td></td>
85. <td class="userpanel">Mayo</td>
86. <td class="userpanel"></td>
87. <td>&nbsp;</td>
88. </tr>
89. <tr>
90. <td></td>
91. <td class="letras_subparrafo">Junio</td>
92. <td></td>
93. <td>&nbsp;</td>
94. </tr>
95. <tr>
96. <td></td>
97. <td class="userpanel">Julio</td>
98. <td class="userpanel"></td>
99. <td>&nbsp;</td>
100. </tr>
101. <tr>

```

```

102. <td></td>
103. <td class="letras_subparrafo">Agosto</td>
104. <td></td>
105. <td>&nbsp;</td>
106. </tr>
107. <tr>
108. <td></td>
109. <td class="userpanel">Septiembre</td>
110. <td class="userpanel"></td>
111. <td>&nbsp;</td>
112. </tr>
113. <tr>
114. <td></td>
115. <td class="letras_subparrafo">Octubre</td>
116. <td></td>
117. <td>&nbsp;</td>
118. </tr>
119. <tr>
120. <td></td>
121. <td class="userpanel">Noviembre</td>
122. <td class="userpanel"></td>
123. <td>&nbsp;</td>
124. </tr>
125. <tr>
126. <td></td>
127. <td class="letras_subparrafo">Diciembre</td>
128. <td></td>
129. <td>&nbsp;</td>
130. </tr>
131. <tr>
132. <td colspan="4"><p>&nbsp;</p></td>
133. </tr>
134. </table>
135. <p>&nbsp;</p>
136. </body>
137. </html>

```