

1. INTRODUCCIÓN

El Departamento de Informática en la presente gestión tiene a bien en apoyar el trabajo de Registro Civil Departamental de Tarija.

El presente trabajo, titulado “**Sistema de Información, Recepción y Seguimiento de Trámites del Registro Civil**”, es el resultado de la inquietud y la necesidad que tiene el ciudadano en la solicitud de un trámite, con este proyecto se pretende mejorar el procedimiento de los trámites que la ciudadanía diariamente solicita en ventanillas del Registro Civil.

2. ANTECEDENTES

El Registro Civil es una institución de orden público encargado de registrar los actos jurídicos y hechos naturales referidos al estado civil de las personas. Su funcionamiento y administración están encomendados a la Corte Nacional y Cortes Departamentales Electorales.

El Servicio Nacional de Registro Civil administra cuatro clases de registros: nacimiento, matrimonio, defunción y reconocimientos. Además está constituido por organismos directivos y operativos.

Organismos directivos. Los organismos directivos son: la Corte Nacional Electoral y las Cortes Departamentales Electorales conforme a Ley.

Organismos operativos. Los organismos operativos son: la Dirección Nacional de Registro Civil, Direcciones Departamentales de Registro Civil y Oficialías de Registro Civil.

La Corte Departamental Electoral de Tarija en la actualidad no cuenta con una página propia de la institución dicha página más bien es un enlace con la página de la Corte Nacional Electoral de Bolivia, la cual no contiene información de

Registro Civil, la poca información que existe está relacionada con pasadas elecciones departamentales.

Además cabe mencionar que en el sitio Web existente el usuario no puede realizar ningún tipo de interacción.

Por otra parte en el país los departamentos de Potosí y Cochabamba cuentan con sitios Web informativos en los que se puede observar muy poca información relacionada al registro civil. Sus direcciones son las siguientes:

- Corte Departamental Electoral de Potosí:

<http://www.potosi.cne.org.bo/>

Contiene información relacionada con la ley 2616 y los pasos que se siguen en registro civil para la inscripción de un ciudadano.

- Corte Departamental Electoral de Cochabamba:

<http://www.cochabamba.cne.org.bo/>

Contiene información de listas de notarias y sus direcciones pero a nivel Departamental.

También existen otros países que disponen de información concerniente al Registro Civil como ser:

- Chile

<http://www.servei.cl/servei/index.aspx?channel=5>

- México

<http://sonora.gob.mx/rcivil/>

3. PLANTEAMIENTO DEL PROBLEMA

El problema principal identificado es la existencia de errores que se le presentan al ciudadano en su partida computarizada, en cualquiera de las 3 categorías, como ser: Nacimiento, Matrimonio y Defunción, motivo por el cual tiene que realizar largas filas y esperar varias horas para ser atendido.

4. OBJETIVOS

4.1. OBJETIVO GENERAL

El objetivo principal es el de desarrollar un Sistema que le permita al ciudadano realizar el registro y seguimiento de su trámite, mediante conexión a Internet, otorgándole al mismo la posibilidad de verificar el estado de su partida computarizada y realizar el respectivo llenado de un formulario el cual será atendido por el Registro Civil para iniciar su trámite administrativo.

4.2 OBJETIVOS ESPECÍFICOS

- Diseñar una base de datos para almacenar toda la información relacionada al Sistema, que mantenga integridad y seguridad.
- Diseñar un módulo de información que le permita al usuario mantenerse informado de lo concerniente al registro civil.
- Desarrollar el Sistema aplicando la metodología RUP para el ciclo de vida del sistema.
- Desarrollar un módulo de solicitud, registro recepción, seguimiento de trámites.

5. JUSTIFICACIÓN

5.1. JUSTIFICACIÓN SOCIAL

Toda institución y más aun la Corte Departamental de Tarija tiene la gran necesidad de atender los requerimientos de la ciudadanía en general con la mayor eficiencia posible en los diferentes trámites que se llevan a cabo en el Registro Civil. Desde el punto de vista social, el ciudadano podrá obtener un certificado de manera más rápida, evitándose de realizar las largas filas en ventanillas.

Además con la implementación de dicho Sistema se tendrá un beneficio para todos los funcionarios que desempeñan su trabajo en esta área.

5.2. JUSTIFICACIÓN ECONÓMICA

Desde el punto de vista económico, dicho sistema no significará un gasto para la institución porque la misma cuenta con un servidor, donde la información estará disponible las 24 horas del día y el ciudadano podrá tener acceso a la información en cualquier momento.

5.3. JUSTIFICACIÓN TECNOLÓGICA

Desde el punto de vista tecnológico la institución dispone del hardware requerido, es decir programas, base de datos, un servidor propio, servicios de Internet y con un ambiente adecuado para la implementación del sistema.

6. ALCANCES Y LIMITACIONES

6.1. ALCANCES

- Se mostrará información Departamental del número de Oficialías de Registro Civil y sus respectivas ubicaciones.
- Se publicará noticias concernientes a las actividades del Registro Civil.
- Por ser una aplicación en ambiente Internet, esta aplicación puede tener un alcance nacional.
- El Sistema tendrá relación directa con el Sistema Regina.
- Para poder verificar el estado de su trámite el ciudadano sólo lo podrá realizar previa autenticación.

6.2. LIMITACIONES

- El Sistema sólo abarcará el área de Registro Civil.
- Los tipos de trámites estarán sujetos a la normatividad administrativa vigente del Registro Civil.
- Los datos de las partidas del Registro Civil estarán sujetos a la información contenida en el REGINA, así como su actualización.
- El Sistema no abarca el registro ni la impresión de certificados.
- El Sistema no realiza el trámite de inscripciones.
- El sistema sólo se centra en trámites administrativos y no están contemplados los trámites judiciales.
- El sistema no abarca la categoría de Reconocimientos.

1. CAPÍTULO I MARCO TEÓRICO

1.1. ACERCA DEL REGISTRO CIVIL

Como todos saben, el Registro Civil tiene a su cargo el registro de los actos y hechos jurídicos referentes al estado civil de las personas.

Es a partir del Registro que pueden asumirse y ejercerse a plenitud los derechos reconocidos por la Constitución y las leyes. Fundamentalmente, acceder al derecho de identidad. Si no hay evidencia jurídica de la existencia de una persona, sencillamente no existe ante la ley.

El Registro Civil es de orden público y desde la promulgación de la Ley No.1367, de 9 de noviembre de 1992, depende del organismo electoral. Fue creado durante la administración presidencial de Severo Fernández Alonso, mediante Ley de 26 de noviembre de 1898, pero recién empezó a funcionar efectivamente el año 1940.

La Ley 2616 de 18 de diciembre de 2003, además de establecer la gratuidad del registro y primer certificado de nacimiento, determinó que la corrección de errores en las partidas debía resolverse a través de trámite administrativo, siempre que no se modifique la fecha de nacimiento del inscrito, su lugar de nacimiento ni su filiación, avance cualitativo en la normativa relativa a la materia, ya que deja atrás los largos y costosos trámites judiciales que anteriormente debían realizarse. Para hacer efectiva su aplicación, la Corte Nacional Electoral aprobó el 18 de marzo de 2004, a través de la Resolución 15/2004, el “Reglamento” para dichos trámites. Este instrumento fue oportunamente evaluado y mediante Resolución No. 284/2005 de 20 de diciembre de 2005 fue sustituido por el nuevo “Reglamento de Rectificaciones, Complementaciones, Ratificaciones y Cancelación de partidas”, documento que, sin duda, permitirá resolver con mayor economía y eficiencia los problemas de errores de registro en las partidas de nacimiento, matrimonio y defunción.

La ley 2616, promulgada como ley de la República en el Palacio de Gobierno de la ciudad de La Paz, a los dieciocho días del mes de diciembre de dos mil tres años, ha

incorporado importantes cambios que modifican las tradicionales y burocráticas formas de adquirir una identidad.

La Corte Nacional Electoral para apoyar y dar solución al alto porcentaje de población de niños, niñas y adolescentes que no son inscritos como ciudadanos formalmente registrados y reconocidos como tales y en cumplimiento al artículo 97 de la Ley N° 2026 Código Niño, Niña y Adolescente, lanza una campaña nacional de gratuidad para registrar y dotar de certificados de nacimiento a niños, niñas hasta los 12 años, que se inicio el 1° de julio hasta el 31 de diciembre del año 2003, sin embargo en este tiempo de campaña, por falta de promulgación de la ley por el Poder Ejecutivo, tan solo se beneficiaron 10.873 niños y niñas de 0 a 7 años de edad en todo el Departamento de Tarija.

Ahora con la Ley N° 2616 el Estado corrige esta situación y decreta el registro y certificación gratuita de todos los niños y niñas hasta los 12 años, adolescentes y mayores sin límite de edad de acuerdo al reglamento de trámites administrativos y judiciales elaborado por la Corte Nacional Electoral.

Que a fin de dar cumplimiento progresivo a lo establecido respecto del Derecho a la Identidad y asumiendo la decisión de hacer efectivo el registro y la certificación gratuita contó con el decidido apoyo económico del Plan Internacional habiendo suscrito Convenios en el marco de cooperación interinstitucional, conforme a su política institucional de apoyo financiero a proyectos o campañas de certificación en favor de la población indocumentada.

1.2. HERRAMIENTAS Y TÉCNICAS

1.2.1. INGENIERÍA DE SOFTWARE

La Ingeniería de Software surge de la ingeniería de sistemas y de hardware. Abarca un conjunto de tres elementos claves: Métodos, herramientas y procedimientos que facilitan al gestor controlar el proceso del desarrollo del software y suministrar a los

que practiquen la ingeniería, las bases para construir software de alta calidad de una forma productiva.

Los **métodos** de la ingeniería de software indican como construir técnicamente el software, los métodos incluyen: Planificación y estimación del proyecto, análisis de requerimientos del sistema y de software, diseño de la estructura de datos, arquitectura de programas y procedimientos algorítmicos, codificación, prueba y mantenimiento.

Las **herramientas** de la ingeniería del software suministran un soporte automático o semiautomático para los métodos.

Los **procedimientos** de la ingeniería del software son el pegamento que junta los métodos y las herramientas, y facilita un desarrollo racional y oportuno del software de computadora. Los procedimientos definen la secuencia en que se aplican los métodos, las entregas (documentos, informes, formas, etc.) que se requieren, los controles que ayudan a asegurar la calidad y coordinar los cambios, y las directrices que ayudan a asegurar la calidad y coordinar los cambios, y las directrices que ayudan a los gestores del software a evaluar el progreso.

Los pasos de la Ingeniería de Software se denominan paradigmas de la Ingeniería del Software.

1.3. CICLO DE VIDA DEL SISTEMA

Se utilizará la metodología RUP para el ciclo de vida del Sistema, porque esta metodología nos ayuda a establecer refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo evolutivo.

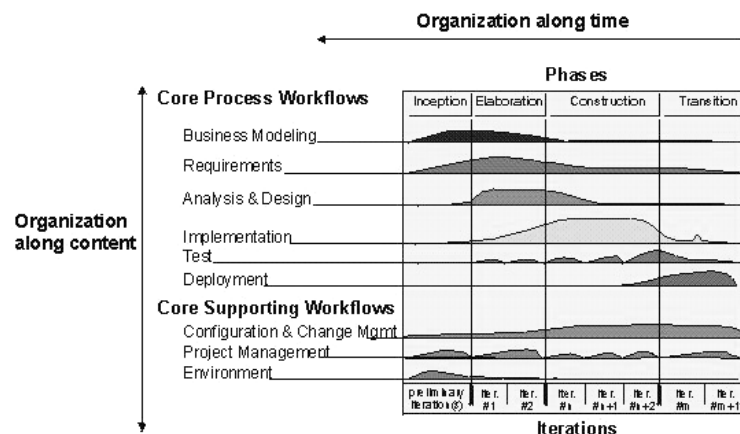
1.3.1 RUP (Rational Unified Process)

Es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un software.

Para este proyecto se ha establecido un cronograma de actividades de acuerdo a la metodología RUP.

1.3.2. CICLOS Y FASES DE LA METODOLOGÍA RUP

RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases:



- **Inicio**

Se debe llevar a cabo la planificación del proyecto y delimitar el alcance del mismo. La idea inicial para desarrollar un Sistema debe estar bien fundamentada. Durante esta etapa es frecuente crear un prototipo ejecutable que sirva para probar los conceptos.

Los objetivos de la fase de Iniciación son:

- Establece la oportunidad y alcance del proyecto
- Identificar todas las entidades externas con la que se trata (actores) y se define la interacción a un alto nivel de abstracción.

- **Elaboración**

En esta etapa se debe analizar el dominio del problema, definir la visión del producto y su arquitectura. Se debe especificar criterios de pruebas precisos que permitan efectuar la evaluación del modelo.

Los objetivos de esta fase son:

- Analizar el dominio del problema
- Establecer una arquitectura sólida
- Desarrollar un plan del proyecto
- Eliminar los elementos de mayor riesgo para el exitoso desarrollo del proyecto.

- **Construcción**

Se lleva a cabo el Diseño arquitectónico a una Base Arquitectónica ejecutable para ponerla a disposición de los usuarios. Ésto implica el refinado del Diseño para completar la Implementación y las pruebas del software.

Los objetivos de esta fase de construcción son:

- Todo es aprobado en profundidad.
- El énfasis está en la producción eficiente y no en la creación intelectual.
- Se puede realizar la construcción en paralelo, pero para ésto se elige una planificación detallada y arquitectura muy estable.

- **Transición**

Una vez que el sistema es puesto en manos de los usuarios finales, el sistema debe seguir mejorando o ajustándose continuamente de acuerdo a los requerimientos adicionales o problemas no destacados.

Al final de la fase de transición debe decidirse si se han satisfecho los objetivos del ciclo de vida del proyecto, de este modo se puede determinar si se empezará otro ciclo de desarrollo.

Los objetivos de esta fase son:

- Traspasar el sistema ya desarrollado a la comunidad de los usuarios.
- Una vez implantado surgirán nuevos elementos que implicarán nuevos desarrollos como ser:
- Pruebas para validar el producto de acuerdo a las expectativas del cliente.
- Conversión de datos.
- Entrenamiento de usuarios.
- Ejecución en paralelo con el sistema antiguo.

Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

1.3.3. VENTAJAS DE RUP

- Aumenta la productividad de los desarrolladores mediante acceso a la Base de conocimiento, plantillas y herramientas.
- Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos.
- RUP es una guía de cómo usar UML de la forma más efectiva.
- Existen herramientas de apoyo a todo el proceso:

Modelamiento visual, programación, pruebas, etc.

RUP pretende implementar las mejores prácticas actuales en ingeniería de software:

- Desarrollo iterativo del software.
- Administración de requerimientos.
- Uso de arquitecturas basadas en componentes.
- Modelamiento visual del software.
- Verificación de la calidad del software.
- Control de cambios.

RUP describe cómo:

- Obtener los requerimientos.
- Organizarlos.
- Documentar requerimientos de funcionalidad y restricciones.
- Rastrear y documentar decisiones.
- Captar y comunicar requerimientos del negocio.

Los casos de uso y los escenarios indicados por el proceso han probado ser una buena forma de captar requerimientos y guiar el diseño, la implementación y las pruebas.

En el desarrollo del proyecto esta metodología permitirá documentar cada una de las fases tomando en cuenta los diagramas de UML.

1.4. UML (Unified Modeling Language)

Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos, para crear una notación unificada en la que basar la construcción de sus herramientas CASE.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora muchas ventajas para el modelado.

UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

1.4.1. MODELOS DE UML

Un modelo representa a un sistema software desde una perspectiva específica. Al igual que la planta y el alzado de una figura en dibujo técnico nos muestran la misma figura vista desde distintos ángulos, cada modelo nos permite fijarnos en un aspecto distinto del sistema.

Los modelos de UML son los siguientes:

- Diagrama de Casos de Uso.
- Diagrama de Secuencia.
- Diagrama de Colaboración.
- Diagrama de Estados.
- Diagramas de Actividad.
- Diagramas de Clases.
- Diagramas de Objetos.

- Diagramas de Componentes.
- **Diagramas de Casos de Uso** para modelar los procesos. Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del Sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

- **Elementos**

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

- **Actores**

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo.

- **Casos de Uso**

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

- **Diagramas de Secuencia** para modelar el paso de mensaje entre objetos. Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

- **Diagramas de Colaboración** para modelar interacciones entre objetos. Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.
- **Diagramas de Estado** para modelar el comportamiento de los objetos en el sistema. Un Diagrama de Estados muestra la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.
- **Diagramas de Actividad** para modelar el comportamiento de los casos de uso, objetos u operaciones. Un diagrama de actividad es una especialización del diagrama de Estados, organiza respecto de las acciones y usada para especificar: un método, un caso de uso o un proceso de negocio.
- **Diagramas de Clases** para modelar la estructura estática de las clases en el sistema. El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.
- **Diagramas de Objetos** para modelar la estructura estática de los objetos en el sistema. Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de las clases del diagrama de clases. Muestra a los objetos y sus relaciones, pero en un momento concreto del sistema. Estos diagramas contienen objetos y enlaces. En los diagramas de objetos también

se pueden incorporar clases, para mostrar la clase de la que es un objeto representado.

- **Diagramas de Componentes** para modelar componentes. Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestra las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

En el presente proyecto, UML permite establecer una notación estándar para definir, organizar y visualizar los elementos que configuran la arquitectura de un sistema, materializándolos en modelos.

El Lenguaje de Modelado Unificado (UML), se aplicará en los flujos de trabajo de Análisis y Diseño.

1.5. DETERMINACIÓN DE REQUERIMIENTOS

La ingeniería de requerimientos se define como un “conjunto de actividades que utiliza técnicas y herramientas para analizar un problema y se concluye con la especificación de una solución (a veces más de una)”.

La Ingeniería de Requerimientos se usa para definir las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requerimientos del producto.

El uso del término “ingeniería” implica que se deben utilizar técnicas sistemáticas y repetibles para asegurar que los requerimientos del sistema estén completos y sean consistentes y relevantes.

1.5.1. REQUERIMIENTOS

Para que el desarrollo del software tenga éxito, se debe comprender bien los requerimientos del sistema. Normalmente, un tema de la Ingeniería de Software tiene diferentes significados. Existen muchas definiciones que existen para requerimiento:

(1) Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales.

- **Los requerimientos funcionales** definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.
- **Los requerimientos no funcionales** tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

El Análisis de Requerimientos es la que establece el nexo entre la ingeniería de sistemas y el diseño del software.

Existen métodos para la identificación del problema principal, para el desarrollo del sistema en el análisis de requerimientos utilizamos el Método ISAC ya que este método establece las mejores condiciones para realizar un estudio más profundo y detallado para determinar los problemas con que enfrenta el usuario y determinar la solución.

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se

desea producir, en base a los requerimientos del usuario. Además que permite identificar el problema relacionado con el desarrollo de la tesis.

1.5.2. MÉTODO ISAC

La abreviación ISAC proviene de “Information System Work and Analisis of Changes ”, herramienta para el desarrollo del Sistema de Información y Análisis de Cambio, este método está orientado al cliente para la determinación de requerimientos, partiendo de un análisis de los problemas y proponiendo soluciones reales, se caracteriza por el alto grado de participación de usuarios y desarrolladores. En este trabajo se empleó este método para identificar los problemas y los requerimientos de la Dirección de Informática, Dirección de Registro Civil y la ciudadanía en general.

El método ISAC es el método de la ingeniería de requerimientos para la especificación de requerimientos y tiene las siguientes características.

- Método orientado al cliente.
- A partir de un análisis de los problemas encuentra soluciones reales para problemas reales.
- El análisis del problema y el estudio de actividades están caracterizados por un alto grado de participación de usuarios y desarrolladores.

El Método ISAC presenta cuatro fases:

- Análisis de Cambio.
- Estudio de Actividad.
- Análisis de información.
- Implementación.

EL método ISAC presenta un modelo de actividades especificadas en niveles. Luego se determinan las propiedades y requerimientos del sistema de información.

Para recolectar información se pueden emplear entrevistas, cuestionarios, observación personal, estudio de documentación y/o antecedentes.

Pasos de la fase de Análisis de Cambio:

- Lista de Problemas.
- Lista de Grupos de Interés.
- Cuantificar los problemas.
- Analizar los problemas.
- Analizar metas.
- Definir necesidades de cambio.
- Especificar situaciones deseadas.
- Evaluar Acciones.
- Elegir el paquete de soluciones.

Para ver la elaboración del Método ISAC (Ver ANEXOS).

En el desarrollo del presente proyecto el método permitirá identificar el problema principal y se aplicará en el flujo de trabajo de Análisis.

1.6. TECNOLOGÍAS WEB

Las tecnologías Web son el conjunto de herramientas que nos ayudan a gestionar mejor el contenido de una página Web, además de mejorar el posicionamiento en Google.

1.6.1. PÁGINA WEB

Una página de Internet o página Web es un documento electrónico que contiene información específica de un tema en particular y que es almacenado en algún sistema de cómputo que se encuentre conectado a la red mundial de información denominada Internet, de tal forma que este documento pueda ser consultado por cualquier persona que se conecte a esta red mundial de comunicaciones y que cuente con los permisos apropiados para hacerlo. Una página Web es la unidad básica del World Wide Web.

Una página Web tiene la característica peculiar de que el texto se combina con imágenes para hacer que el documento sea dinámico y permita que se puedan ejecutar diferentes acciones, una tras otra, a través de la selección de texto remarcado o de las imágenes, acción que nos puede conducir a otra sección dentro del documento, abrir otra página Web, iniciar un mensaje de correo electrónico o transportarnos a otro Sitio Web totalmente distinto a través de sus hipervínculos.

Estos documentos pueden ser elaborados por los gobiernos, instituciones educativas, instituciones públicas o privadas, empresas o cualquier otro tipo de asociación, y por las propias personas en lo individual.

1.6.2. SITIO WEB

Es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada home page, con un nombre de dominio y dirección en Internet específicos.

Empleados por las instituciones públicas y privadas, organizaciones e individuos para comunicarse con el mundo entero. En el caso particular de las empresas, este mensaje tiene que ver con la oferta de sus bienes y servicios a través de Internet, y en general para brindar eficiencia en sus funciones de mercadotecnia.

Su Sitio Web no necesariamente debe localizarse en el sistema de cómputo de su negocio. Los documentos que integran el Sitio Web pueden ubicarse en un equipo en otra localidad, inclusive en otro país. El único requisito es que el equipo en el que residan los documentos esté conectado a la red mundial de Internet. Este equipo de cómputo o Servidor Web, como se le denomina técnicamente, puede contener más de un sitio Web y atender concurrentemente a los visitantes de cada uno de los diferentes sitios.

Al igual que los edificios, oficinas y casas, los Sitios Web requieren de una dirección particular para que los usuarios puedan acceder a la información contenida en ellos. Estas direcciones, o URLs (por sus siglas en inglés Uniform Resource Locator), aparecen cotidianamente en todos los medios de comunicación como son prensa escrita, radio, televisión, revistas, publicaciones técnicas y en el propio Internet a través de los motores de búsqueda (por su denominación en inglés search engines). Los nombres de estos sitios Web obedecen a un sistema mundial de nomenclatura y están regidos por el ICANN (Internet Corporation for Assigned Names and Numbers).

Los Sitios Web pueden ser de diversos géneros, destacando los sitios de negocios, servicio, comercio electrónico en línea, imagen corporativa, entretenimiento y sitios informativos.

1.6.3. SITIOS WEB DINÁMICOS

Un sitio Web dinámico es uno que puede tener cambios frecuentes en la información. Cuando el servidor Web recibe una petición para una determinada página, la página se genera automáticamente por el software como respuesta directa a la petición de la página; por lo tanto abriendo muchas posibilidades incluyendo por ejemplo: a sitio puede mostrar el estado actual de un diálogo entre usuarios, monitorizar una situación cambiante, o proporcionar información de alguna manera personalizada a los requisitos del usuario individual.

Hay un amplio abanico de sistemas de software, como Active Server Pages (ASP), Java Server Pages (JSP) y el lenguaje de programación PHP que están disponibles para generar sistemas de sitios Web dinámicos. Los sitios dinámicos a menudo incluyen contenido que se recupera de una o más bases de datos o usando tecnologías basadas en XML como por ejemplo el RSS.

El contenido estático puede también ser generado de manera dinámica periódicamente o si ocurren ciertas condiciones para la regeneración para evitar la pérdida de rendimiento de iniciar el motor dinámico para cada usuario o para cada conexión.

Hay plugins disponibles para navegadores, que se usan para mostrar contenido activo como Flash, Shockwave o applets escritos en Java. El HTML dinámico también proporciona para los usuarios interactividad y el elemento de actualización en tiempo real entre páginas Web (las páginas no tienen que cargarse o recargarse para efectuar cualquier cambio), principalmente usando el DOM y JavaScript, el soporte de los cuales está integrado en la mayoría de navegadores Web modernos.

1.7. PROTOCOLO TCP/IP

El protocolo TCP/IP es una familia de protocolos desarrollados para permitir la comunicación entre computadoras de diferentes redes, independientemente del fabricante.

Las siglas TCP/IP se refieren a dos protocolos dentro de red los cuales son:

- Transmisión Control Protocol: Protocolo de Control de Transmisión
- Internet Protocol: Protocolo de Internet

Existen dos clases de protocolos dentro de los que es TCP/IP:

- Protocolos a nivel de red
- Protocolos a nivel a nivel de aplicación

1.8. URL: Uniform Resource Locator

Uniform Resource Locator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

Las URL fueron una innovación fundamental en la historia del Internet. Fueron usadas por primera vez por Tim Berners-Lee en 1991, para permitir a los autores de documentos establecer hiperenlaces en la World Wide Web (WWW o Web). Desde 1994, en los estándares del Internet, el concepto de URL ha sido incorporado dentro

del más general de **URI** (**U**niform **R**esource **I**dentifier - Identificador Uniforme de Recurso), pero el término URL aún se utiliza ampliamente.

Aunque nunca fueron mencionadas como tal en ningún estándar, mucha gente cree que las iniciales URL significan Universal Resource Locator (Localizador Universal de Recurso). Esta interpretación puede ser debida al hecho de que, aunque la U en URL siempre ha significado Uniforme, la U de URI significó en un principio Universal, antes de la publicación del RFC 2396.

El URL es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en el Internet. Existe un URL único para cada página de cada uno de los documentos de la World Wide Web, para todos los elementos de Gopher y todos los grupos de debate USENET, y así sucesivamente.

El URL de un recurso de información es su dirección en Internet, la cual permite que el navegador la encuentre y la muestre de forma adecuada. Por ello el URL combina el nombre del ordenador que proporciona la información, el directorio donde se encuentra, el nombre del fichero y el protocolo a usar para recuperar los datos.

Y reemplaza la dirección numérica o IP de los servidores haciendo de esta manera más fácil la navegación.

1.9. HTTP: HyperText Transfer Protocol

El **protocolo de transferencia de hipertexto** (**HTTP**, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas Web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página Web, y la respuesta de esa Web, remitiendo la información que se verá en pantalla. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con mensajes y otros similares.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Por esto se popularizaron las cookies, que son pequeños ficheros guardados en el propio ordenador que puede leer un sitio Web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente. Gracias a esta identificación, el sitio Web puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio.

HTTP dispone de una variante cifrada mediante SSL llamada HTTPS.

1.10. PHP (Hypertext Processor)

PHP (acrónimo de “PHP: Hypertext Processor”) es un lenguaje “open source” interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

En vez de escribir un programa con muchos comandos para crear una salida en HTML, escribimos el código HTML con cierto código PHP embebido (introducido) en el mismo, que producirá cierta salida (en nuestro ejemplo, producir un texto). El código PHP reincluye entre etiquetas especiales de comienzo y final que nos permitirá entrar y salir del modo PHP.

Lo que distingue a PHP de la tecnología JAVA scrip, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tendríamos un script en nuestro servidor, el cliente solamente recibirá el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor Web puede ser incluso configurado para que procese todos los ficheros HTML con PHP.

Aunque el desarrollo PHP está concentrado en la programación de script en la parte del servidor, se puede utilizar para muchas otras cosas.

Porque utilizar PHP

- PHP no soporta punteros, como C, de forma que no existen los problemas de depuración provocados por éstos.
- Se pueden hacer grandes cosas con pocas líneas de código.
- El código PHP es mucho más legible que el de PERL.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un e-commerce, xml, creación de PDF, etc)
- Al poderse encapsular dentro de código html se puede recoger el trabajo del diseñador gráfico e incrustar el código PHP posteriormente.
- Esta siendo utilizado con éxito en varios millones de sitios web.
- Hay multitud de aplicaciones PHP para resolver problemas concretos (weblogs, tiendas virtuales, periódicos, etc) listas para usar.
- Es multiplataforma, funciona en todas las plataformas que soporten apache.
- Es software libre. Se puede obtener en la Web y su código está disponible bajo la licencia GPL.

Seguridad en PHP

PHP es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza. PHP esta diseñado específicamente para ser un lenguaje mas seguro para escribir programas CGI que PERL o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación, PHP le puede dar la combinación precisa de libertad y seguridad que usted necesita.

Ya que hay muchas maneras de utilizar PHP, existen varias opciones de configuración diseñadas para controlar su comportamiento.

Un amplio rango de opciones le garantizan que pueda usar PHP para muchos propósitos distintos, pero también quiere decir que hay combinaciones de éstas opciones y configuraciones de servidor que pueda resultar en un entorno inseguro.

1.11. BASE DE DATOS

Las Bases de Datos son parte fundamental para la realización de sistemas, permiten gestionar estos datos es lo que se denomina sistema gestor de base de datos.

1.11.1. DEFINICIÓN DE BASE DE DATOS

Una **base de datos** es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior. En la actualidad, y gracias al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

Una Base de Datos puede definirse como: “Un conjunto auto descriptivo de registros integrados”.

La utilización de Base de Datos como plataforma para el desarrollo de sistemas de Gestión en las organizaciones se ha incrementado notablemente en los últimos años, esto se debe a las ventajas que ofrece: Globalización de la información, eliminación

de información inconsistente, permite compartir información, independencia de datos y permitir mantener la integridad de la información.

1.11.2. SISTEMA MANEJADOR DE BASE DE DATOS (DBMS)

El Sistema Manejador de Base de Datos (DBMS) es un conjunto de programas que se encargan de manejar la creación y los accesos a las Bases de Datos. El DBMS para cumplir con sus tareas se apoya en el Lenguaje de Definición de Datos (DDL), el Lenguaje de Manipulación de Datos (DML) y también en el Lenguaje de Consulta.

1.11.3. LENGUAJE DE DEFINICIÓN DE DATOS (DDL)

Significa Lenguaje de Definición de Datos (DDL) el cual es parte de un modelo de datos usado para definir la estructura de datos de la Base de Datos, las vistas y subesquemas.

Es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

El lenguaje de programación SQL, el más difundido entre los gestores de bases de datos, admite las siguientes sentencias de definición: CREATE, DROP y ALTER, cada una de las cuales se puede aplicar a las tablas, vistas, procedimientos almacenados y triggers de la Base de Datos.

Otras que se incluyen dentro del DDL, pero que su existencia depende de la implementación del estándar SQL que lleve a cabo el gestor de BD son GRANT y REVOKE, los cuales nos sirven para otorgar permisos o quitarlos, ya sea a usuarios específicos o a un rol creado dentro de la BD.

1.11.4. LENGUAJE DE MANIPULACIÓN DE DATOS (DML)

Lenguaje de Manipulación de Datos (DML) es la parte de un Modelo de Datos empleada para escribir el procesamiento de una Base de Datos.

La manipulación de datos se refiere a las operaciones de insertar, recuperar, eliminar o modificar datos; dichas operaciones son realizadas a través del lenguaje de manipulación de datos (DML, Data Manipulation Language), que es quién permite el acceso de los usuarios a los datos.

Existen básicamente 2 tipos de lenguajes de manipulación de datos:

Procedimentales:

Los LMD requieren que el usuario especifique que datos se necesitan y cómo obtenerlos.

No procedimentales:

Los LMD requieren que el usuario especifique que datos se necesitan y sin especificar cómo obtenerlos.

1.12. MySQL

Se trabajará con MySQL como servidor de la base de datos de acuerdo a las características del sistema, es el que más se adecua a los requerimientos.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones¹. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Ésto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

Lenguajes de programación

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de éstos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL.

Aplicaciones

MySQL es muy utilizado en aplicaciones web como MediaWiki o Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es

una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Plataformas

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. También existe MySQL para OpenVMS en <http://www.pi-net.dyndns.org/anonymous/kits/>.

EMS MySQL Manager es un potente y efectiva herramienta para la administración y mantenimiento de un servidor MySQL.

Posee una interfaz gráfica de usuario que posibilita creación/edición de base de datos MySQL de una manera simple y sencilla.

Soporta Script SQL, construcciones visuales de consulta SQL, extrae o imprime metadatos, exporta/importa datos, mantenimiento y gestión de privilegios de usuarios. Todos ellos y muchos más servicios que harán que el trabajo con MySQL sea más práctico, agradable y sencillo.

Características principales:

- Soporte completo para las versiones de MySQL desde 3.23 hasta 5.2.
- Soporte de datos UTF8.
- Navegación y administración rápida de la base de datos.
- Administración simple de todos los objetos MySQL.
- Herramientas de manipulación de datos avanzada.
- Conexión a MySQL Server a través de HTTP.
- Conexión a MySQL Server a través de SSH.

- Potente Administración de seguridad.
- Excelentes Herramientas visuales y de texto para elaboración de consultas.
- Impresionantes opciones de exportación e importación de datos.
- Diseñador Visual de Base de Datos completamente rediseñado.
- Asistentes fáciles de usar para efectuar servicios MySQL.

1.13. GESTOR BASE DE DATOS

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una base de datos

Especificar tipos, estructuras y restricciones de datos.

- Construir la base de datos

Guardar los datos en algún medio controlado por el mismo SGBD

- Manipular la base de datos

Realizar consultas, actualizarla, generar informes.

Así se trata de un software de propósito general.

Algunas de las características deseables en un Sistema Gestor de base de datos SGBD son:

- Control de la redundancia

La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.

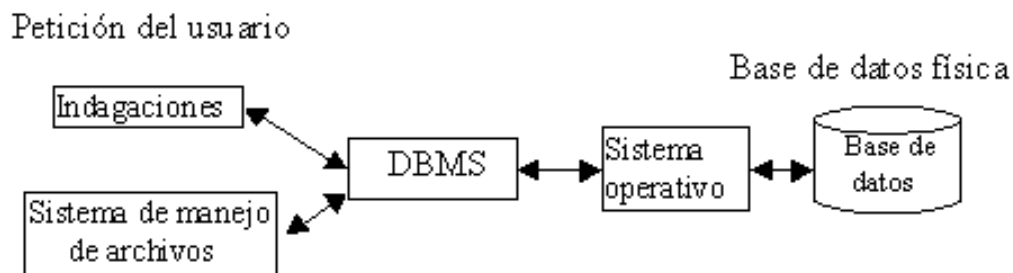
- Restricción de los accesos no autorizados

Cada usuario ha de tener unos permisos de acceso y autorización.

- Cumplimiento de las restricciones de integridad

El SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.

El DBMS es conocido también como Gestor de Base de datos.



La figura muestra el DBMS como interfase entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

1.14. APLICACIONES DE LA BASE DE DATOS CLIENTE / SERVIDOR

Es un modelo para el desarrollo de Sistemas de Información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que indica el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

2. CAPÍTULO II ANÁLISIS

2.1. DIAGRAMAS DE CASOS DE USO

2.1.1. Diagrama de Contexto General

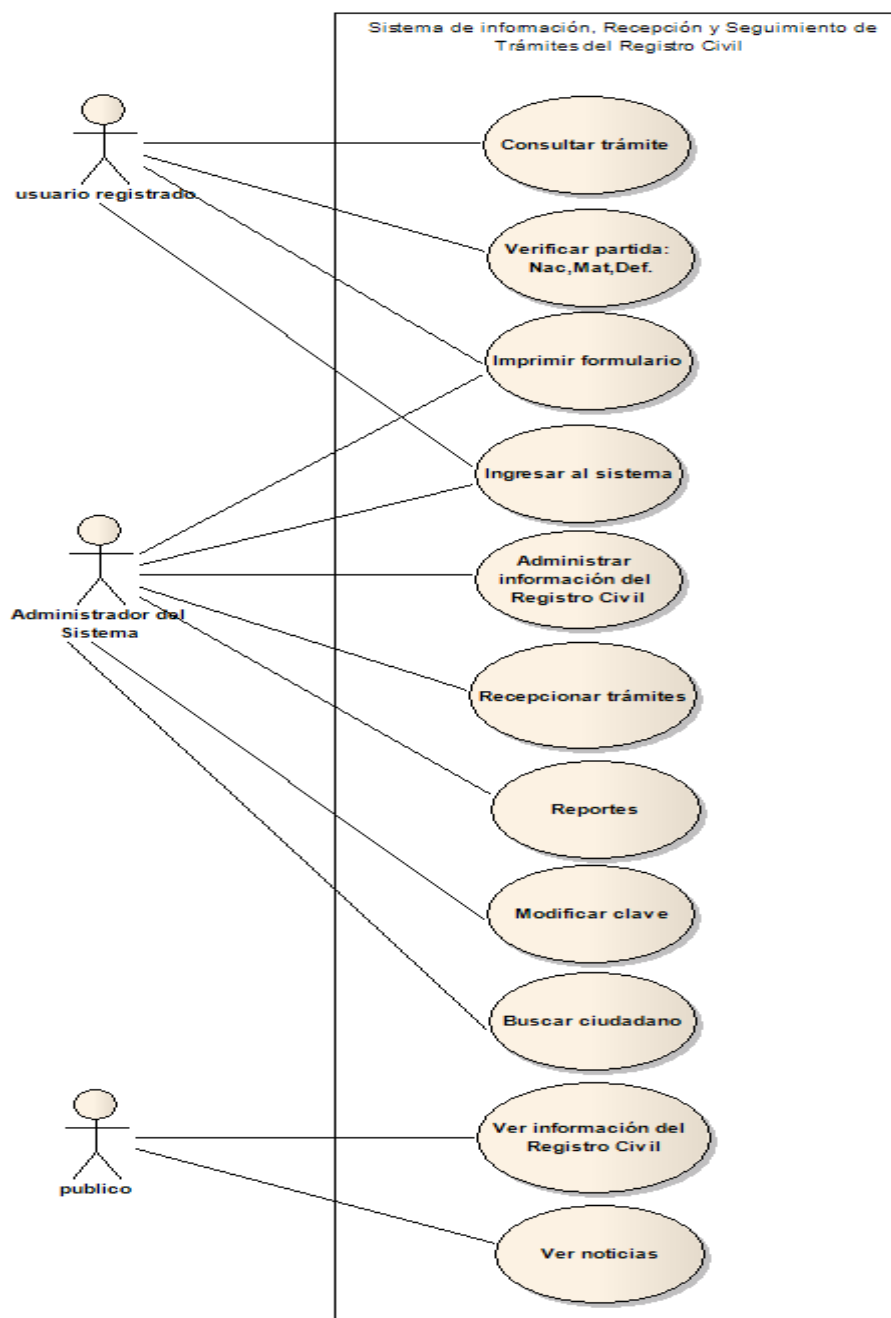


Figura1

2.1.4. Diagrama de Caso de Uso Solicitar Información

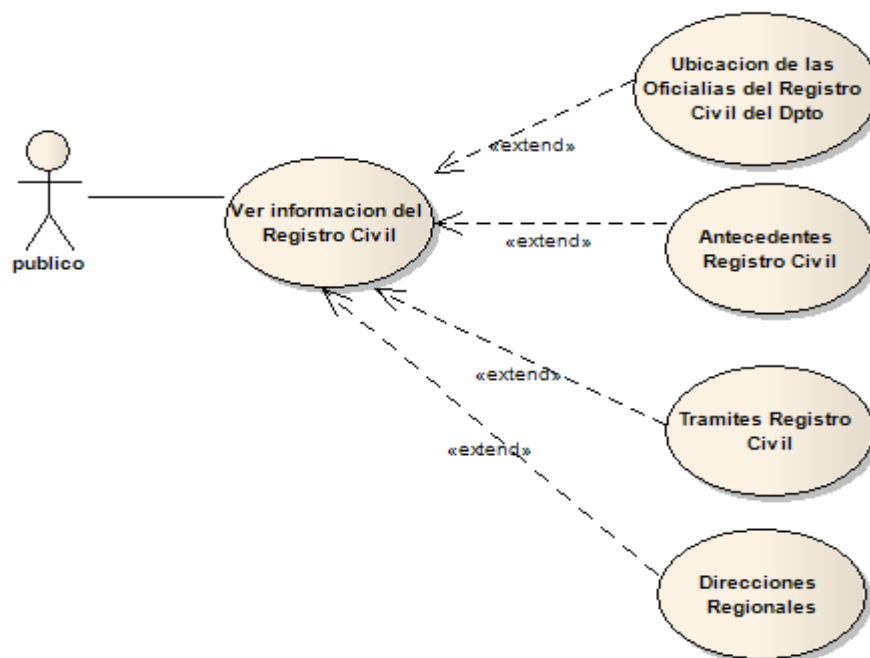


Figura4

2.1.5. Diagrama de Caso de Uso Consultar Partida

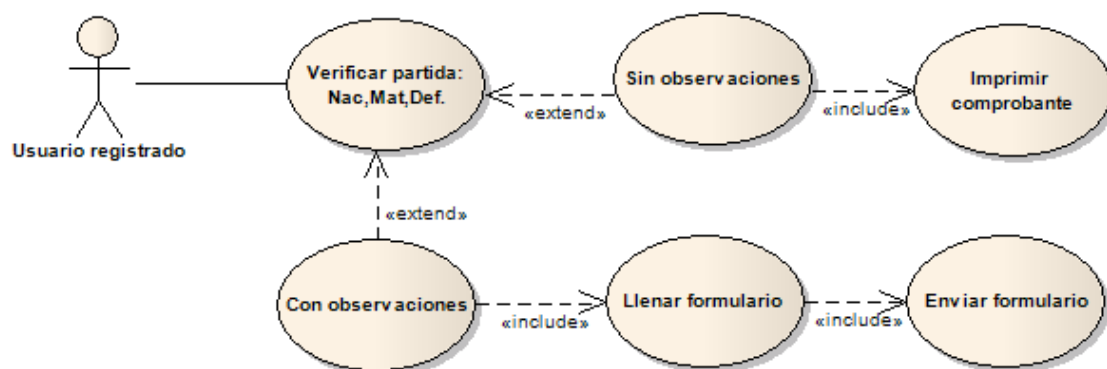


Figura5

2.1.6. Diagrama de Caso de Uso Realizar Seguimiento Trámite

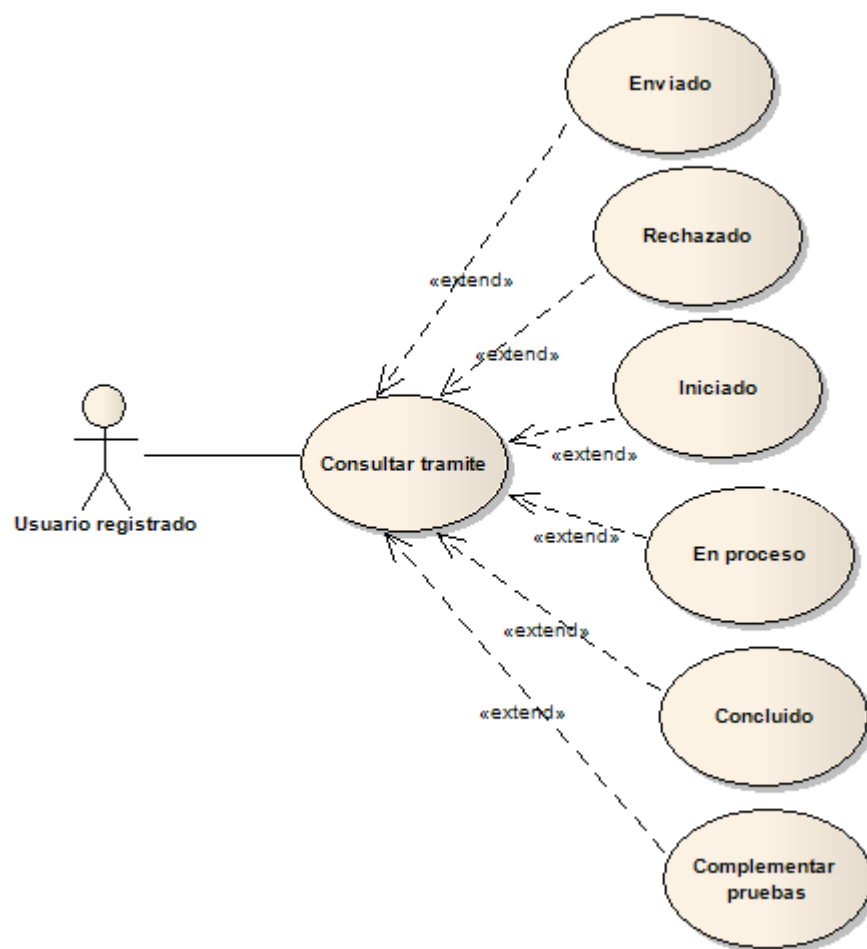


Figura6

2.1.7. Diagrama de Caso de Uso Cambiar Clave

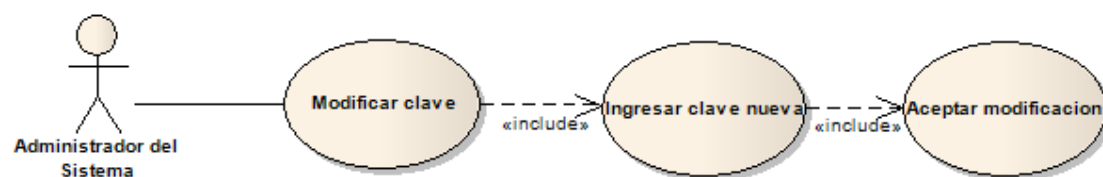


Figura7

2.1.8. Diagrama de Caso de Uso Reportes

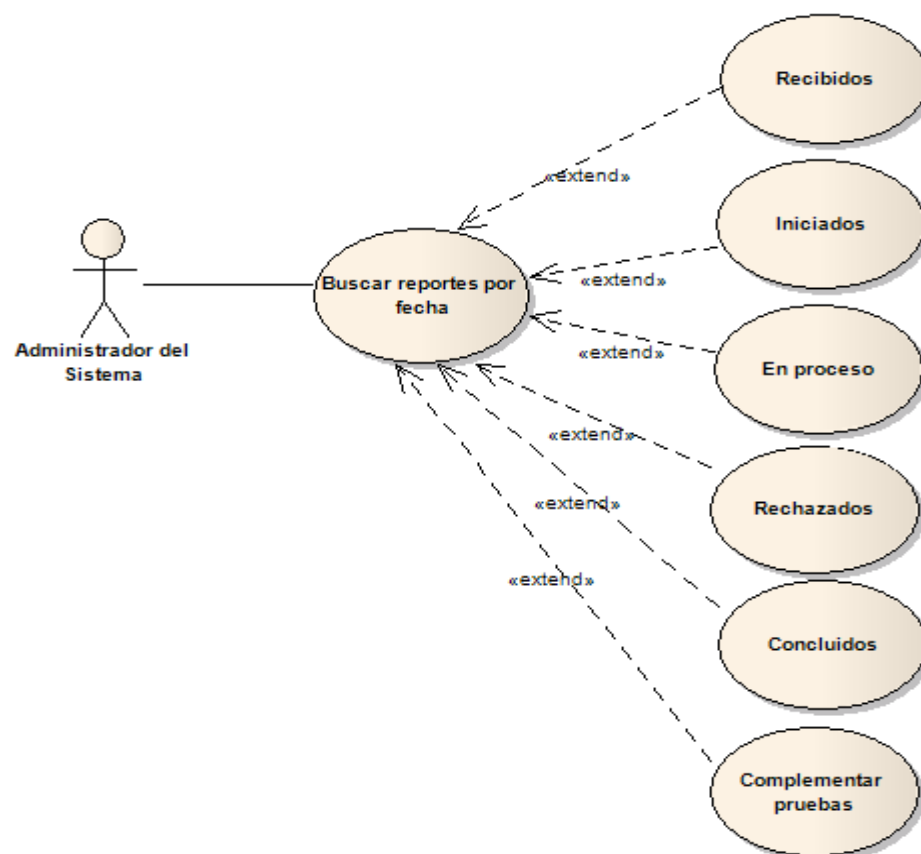


Figura8

2.1.9. Diagrama de Caso de Uso Despachar trámites

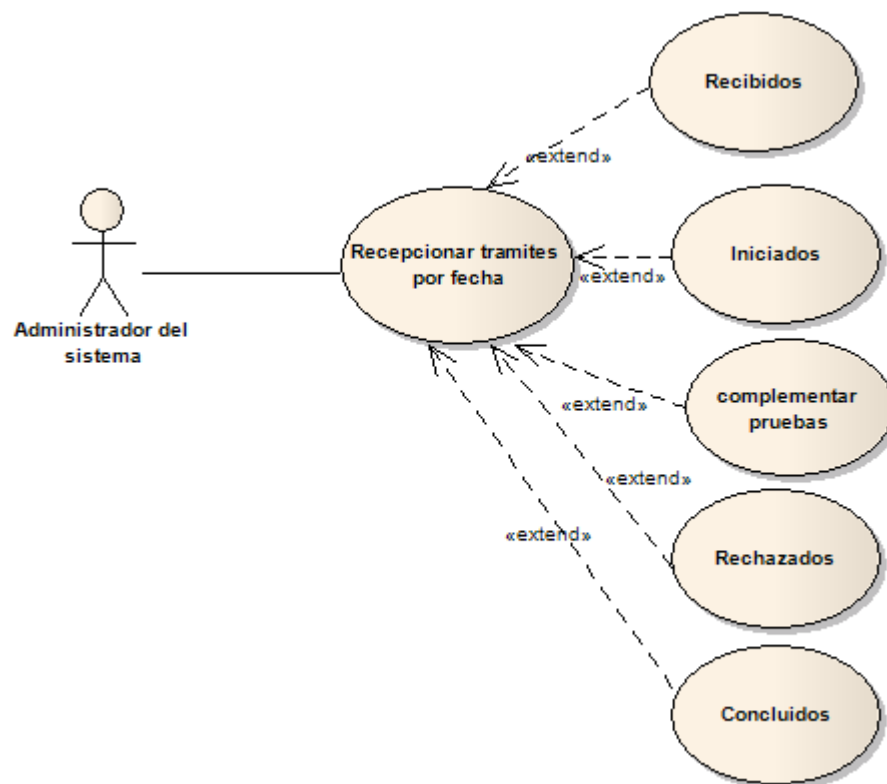


Figura9

2.2. DESCRIPCIÓN DE LOS CASOS DE USO

2.2.1 Caso de uso Verificar partida: Nacimiento, Matrimonio, Defunción.

Nombre del caso de uso	Verificar partida: Nacimiento, Matrimonio, Defunción.	
Descripción	El caso de Uso ofrece un nombre de usuario y un password al usuario.	
Actores:	usuario registrado	
Pre-Condicion	Ingresar al sistema.	
Flujo Normal	Acciones del Actor	Respuesta del sistema
	1. El usuario debe encontrarse navegando en el sistema.	2. El sistema activa la pagina principal.
Flujo Altern	• Del punto 1: El usuario tiene que saber la dirección correcta del sistema, caso contrario no podrá ingresar al mismo.	
Pos-Condicion	El usuario puede presionar cualquier botón del menú principal y seguir navegando por el sitio.	

2.2.2. Caso de uso Administrar Información Registro Civil

Nombre del caso de uso	Administrar Información Registro Civil	
Descripción	El caso de Uso se encarga de dar altas, bajas o modificaciones en el sistema.	
Actores:	Administrador del Sistema	
Pre-Condicion	El usuario se encuentre navegando en el Sistema	
Flujo Normal	Acciones del actor	Respuesta del sistema
	1. El caso de uso inicia cuando el usuario ingresa a la página del administrador. 3. El usuario registra, modifica, o elimina información y pulsa el botón aceptar.	2. El sistema le muestra el menú de la página. 4. El sistema almacena la información.
Flujo Altern	•Del punto 3: El usuario puede cancelar el proceso en cualquier momento saliendo de la pagina. •Del punto3: El usuario puede borrar la información si lo desea e introducir una nueva información antes de pulsar el botón aceptar.	
Pos-condiciones	El usuario puede continuar modificando, registrando o eliminando información o volver a la página de inicio.	

2.2.3. Caso de uso Imprimir formulario

Nombre del caso de uso	Imprimir formulario	
Descripción	El caso de Uso se encarga de imprimir el formulario que es de vital importancia para el usuario.	
Actores:	Usuario registrado, Administrador del Sistema.	
Pre- Condiciones	Ingresar al sistema.	
Flujo Normal	Acciones del Actor	Respuesta del sistema
	1. El usuario selecciona la opción imprimir.	2. El sistema activa el botón imprimir.
Flujo Alternativo	Del punto 2: Si el usuario no cuenta con una impresora o la misma no esta en buen estado no podrá realizar la opción de imprimir.	
Pos- Condiciones	El usuario puede escoger la opción imprimir cuantas veces lo desee.	

2.2.4. Caso de uso Recepcionar trámites

Nombre del caso de uso	Recepcionar trámites	
Descripción	El caso de Uso se encarga de atender los trámites enviados por el ciudadano.	
Actores:	Administrador del Sistema	
Pre- Condiciones	Ingresar al sistema.	
Flujo Normal	Acciones del Actor	Respuesta del sistema
	1. El usuario debe acceder a la opción Trámites.	2. El sistema activa la opción Buscar tramites y le muestra los diferentes estados de tramites.
Flujo Alternativo	Del punto 2: El usuario puede seleccionar cualquier otra opción del menú tramites.	
Pos- Condiciones	El usuario puede continuar seleccionando cualquier opción del menú trámites.	

2.2.5. Caso de uso Ver información del Registro Civil

Nombre del caso de uso	Ver información del Registro Civil	
Descripción	El caso de Uso se encarga de mostrar información al usuario.	
Actores:	publico	
Pre- Condiciones	Ingresar al sistema.	
Flujo Normal	Acciones del Actor	Respuesta del sistema
	1. El usuario debe ingresar al Sistema.	2. El sistema activa la pagina principal y le muestra un menú al ciudadano.
Flujo Alternativo	Del punto 2: El usuario puede seleccionar cualquier pestaña del menú.	
Pos- Condiciones	Si el usuario no introduce el nombre de la página de forma correcta no podrá ingresar al Sistema.	

2.2.6. Caso de uso Modificar clave

Nombre del caso de uso	Modificar clave	
Descripción	El caso de Uso se encarga de modificar la clave del usuario.	
Actores:	Administrador del Sistema	
Pre- Condiciones	Ingresar a la página del Administrador.	
Flujo Normal	Acciones del Actor	Respuesta del sistema
	1. El usuario debe acceder a un menú y debe ingresar a la opción seguridad. 3. El usuario introduce nuevos datos.	2. El sistema activa la pagina principal y le muestra un formulario al usuario donde el mismo puede cambiar su nombre de usuario y su password. 4. El sistema almacena el nuevo nombre de usuario y password.
Flujo Alternativo	• Del punto 2: El usuario puede cambiar de nombre de usuario y password tantas veces lo desee.	
Pos- Condiciones	Si el usuario no introduce el nombre de la página de forma correcta no podrá ingresar al Sistema.	

2.2.7. Caso de uso reportes

Nombre del caso de uso	reportes	
Descripción	El caso de Uso se encarga de mostrar reportes al usuario de los diferentes estados de trámites que solicita el ciudadano.	
Actores:	Administrador del Sistema	
Pre- Condiciones	Ingresar al sistema.	
Flujo Normal	Acciones del Actor	Respuesta del sistema
	1. El usuario debe ingresar al Sistema. 3. El usuario debe seleccionar cualquier estado.	2. El sistema activa la pagina principal y le muestra un menú al usuario. 4. El sistema le muestra un formulario al usuario donde el debe seleccionar del calendario desde y hasta cuando quiere ver reportes.
Flujo Alterno	• Del punto 3: El usuario puede seleccionar cualquier otro estado del menú.	
Pos- Condiciones	Si el usuario no introduce el nombre de la página de forma correcta no podrá ingresar al Sistema.	

2.2.8. Caso de uso Buscar ciudadano

Nombre del caso de uso	Buscar ciudadano	
Descripción	El caso de Uso se encarga Buscar un ciudadano en el Sistema en cualquiera de las 3 categorías (nacimiento, matrimonio, defunción).	
Actores:	Administrador del Sistema	
Pre- Condiciones	Ingresar al sistema.	
Flujo Normal	Acciones del Actor	Respuesta del sistema
	1. El usuario debe seleccionar del menú la opción Buscar ciudadano. 3. El usuario debe ingresar su criterio de búsqueda.	2. El sistema activa la pagina principal y le muestra un formulario al usuario. 4. El sistema le muestra al usuario a la persona buscada y otras personas con datos similares si es que los hubiera.
Flujo Alternativo	• Del punto 3: El usuario puede seleccionar cualquiera de las 3 categorías (nacimiento, matrimonio o defunción).	
Pos- Condiciones	Si el usuario no introduce el nombre de la página de forma correcta no podrá ingresar al Sistema.	

2.2.9. Caso de uso Consultar trámite

Nombre del caso de uso	Consultar trámite	
Descripción	El caso de Uso se encarga de consultar el estado del trámite del ciudadano.	
Actores:	Usuario registrado	
Pre- Condiciones	Ingresar al sistema.	
Flujo Normal	Acciones del Actor	Respuesta del sistema
	1. El usuario debe loguearse para acceder al Sistema.	2. El sistema activa la página principal y le muestra al ciudadano el estado de su trámite.
Flujo Alternativo	Del punto 1: Si el usuario introduce de manera incorrecta su nombre de usuario y password no podrá acceder al Sistema.	
Pos- Condiciones	Si el usuario no introduce el nombre de la página de forma correcta no podrá ingresar al Sistema.	

3. CAPÍTULO III DISEÑO

3.1 DIAGRAMAS DE SECUENCIA

3.1.1. Diagrama de Secuencia Administrar Información del Registro Civil

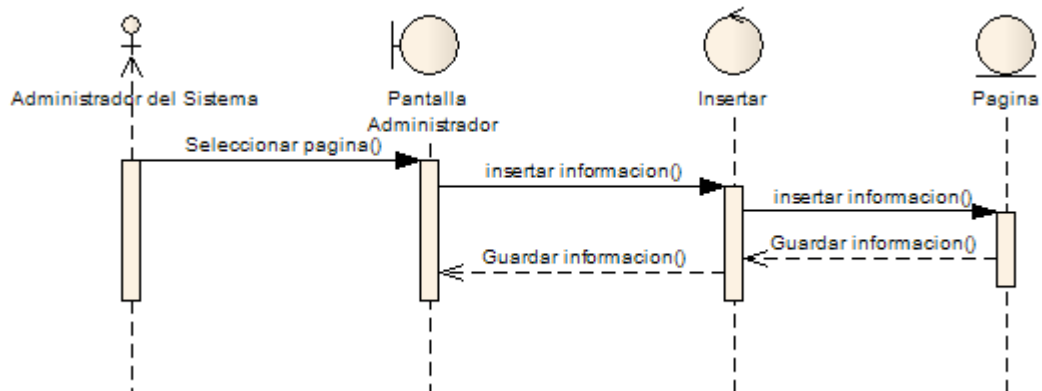


Figura10

3.1.2. Diagrama de Secuencia Ingresar al Sistema

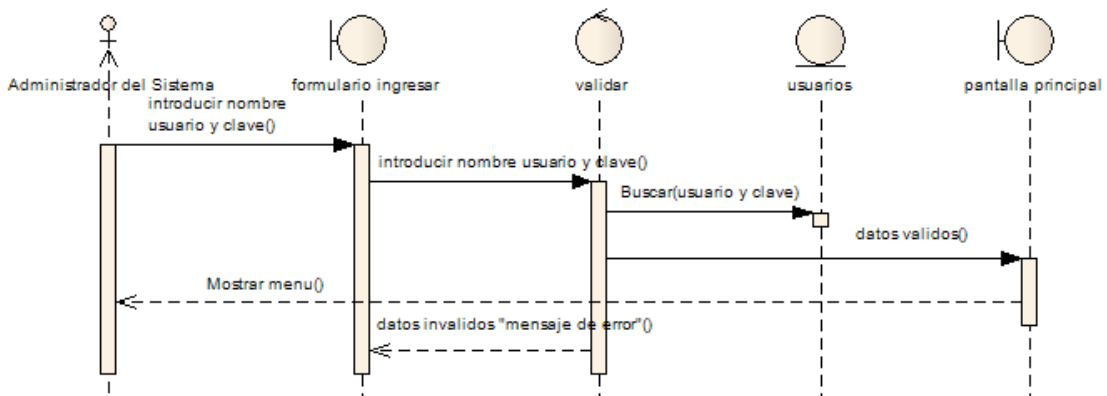


Figura11

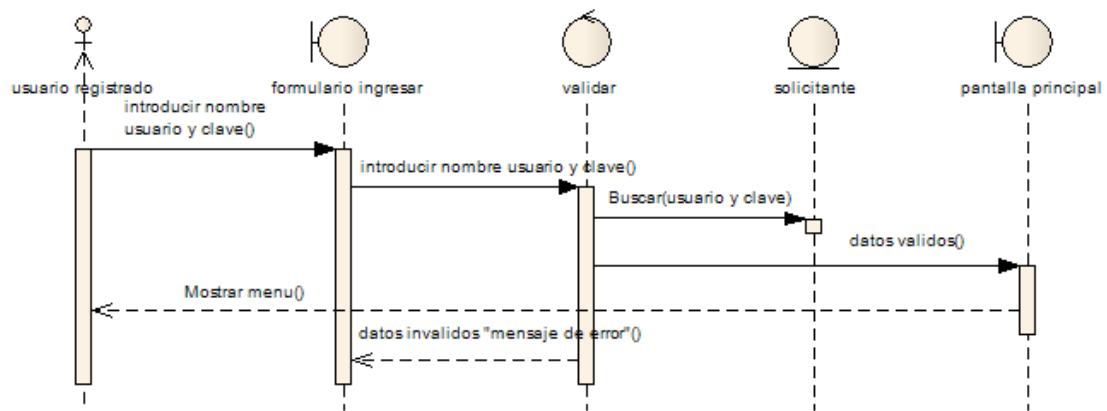


Figura12

3.1.3. Diagrama de Secuencia Modificar clave

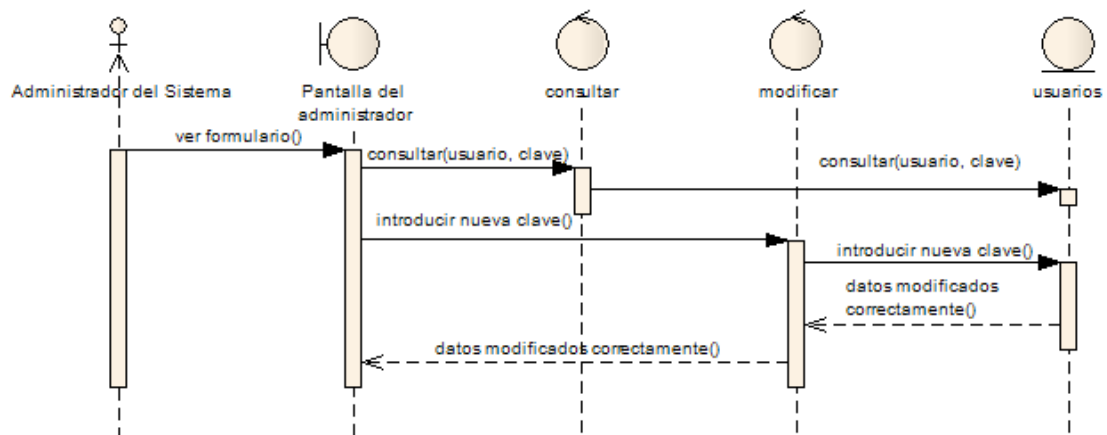


Figura13

3.1.4. Diagrama de Secuencia Ver Información

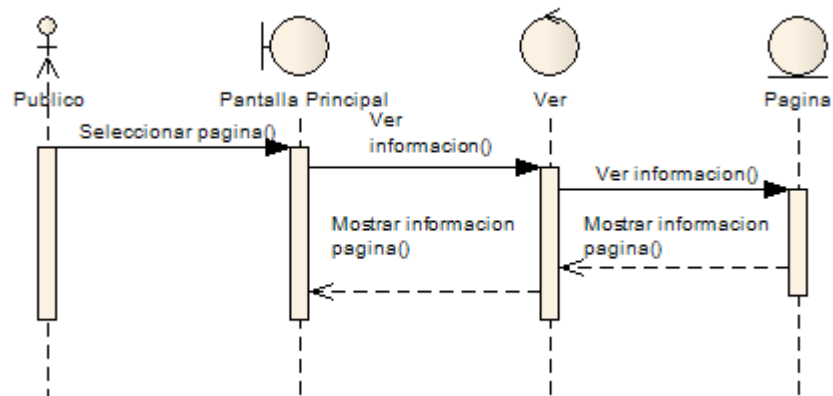


Figura14

3.1.5. Diagrama de Secuencia Buscar Ciudadano

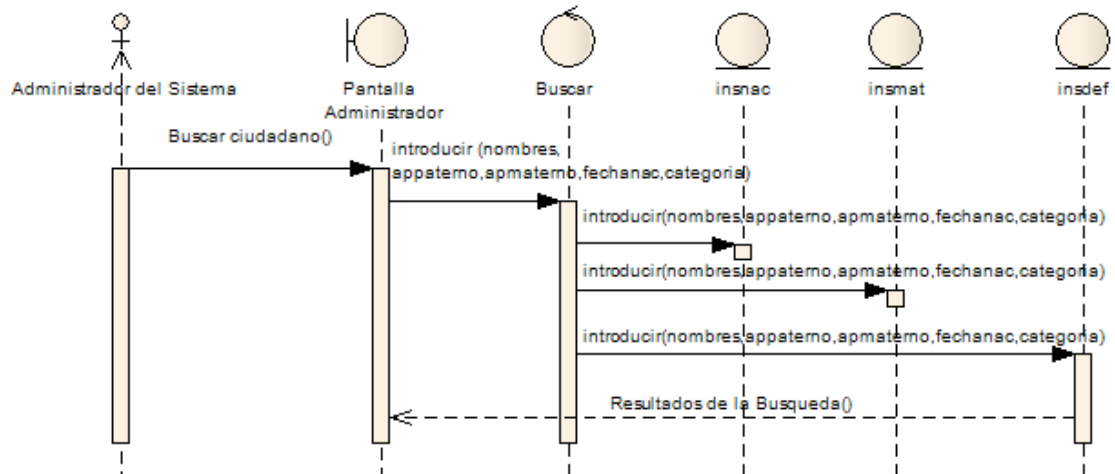


Figura15

3.1.6. Diagrama de Secuencia imprimir

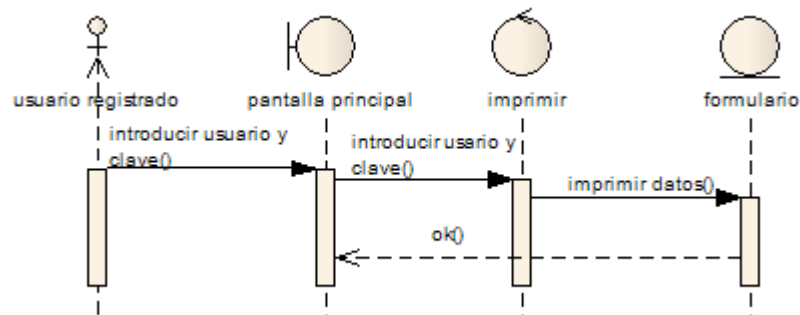


Figura16

3.1.7. Diagrama de Secuencia Reportes Trámites

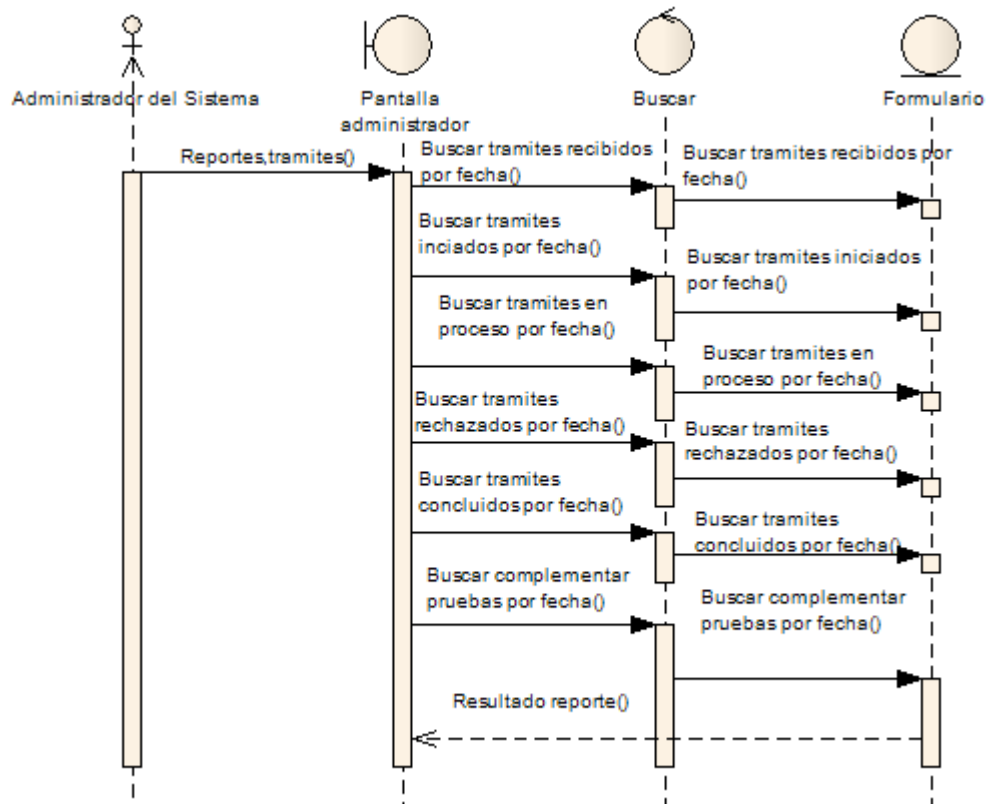


Figura17

3.1.8. Diagrama de Secuencia Llenar formulario

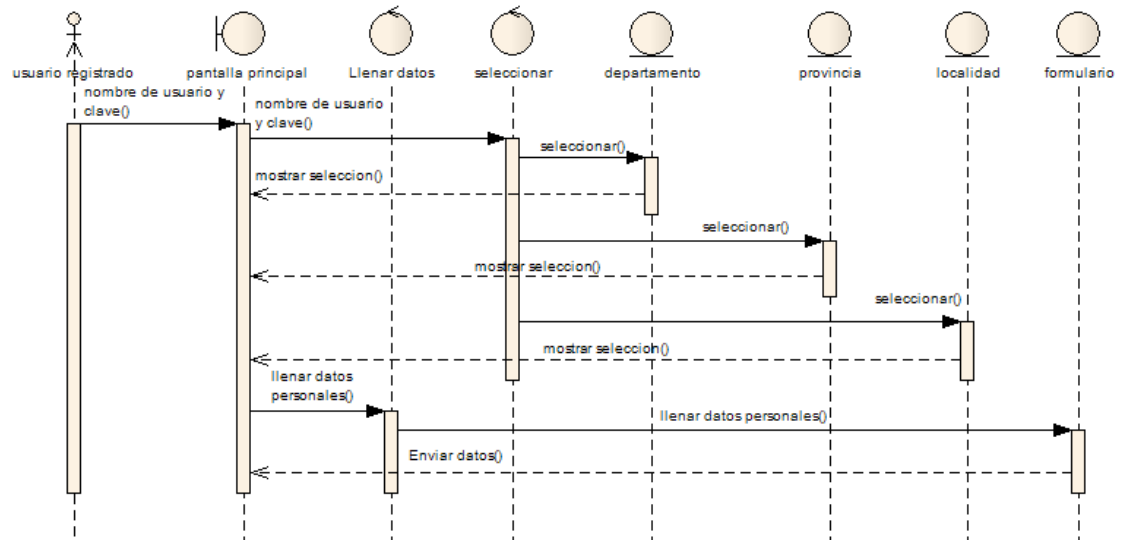


Figura18

3.1.9. Diagrama de Secuencia enviar pruebas

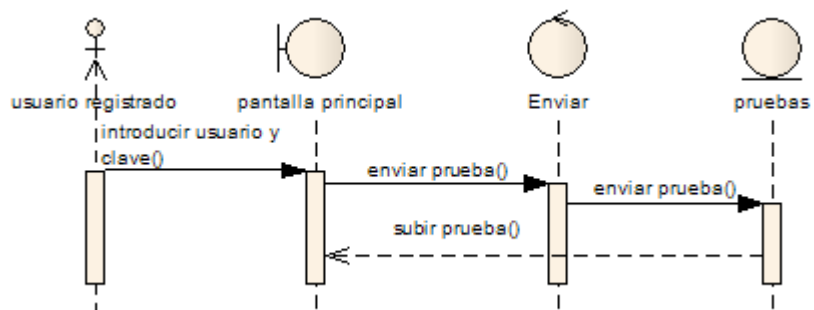


Figura19

3.2. DIAGRAMAS DE ACTIVIDADES

Los diagramas de actividades se utilizan para modelar los aspectos dinámicos de un sistema, esto implica los pasos secuenciales de un proceso.

3.2.1. Diagrama de actividades: Modificar clave

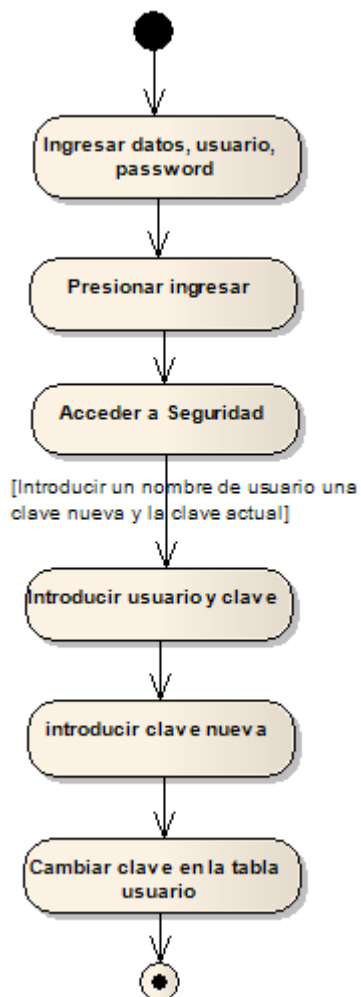


Figura20

3.2.2. Diagrama de actividades: Acceder al Sistema

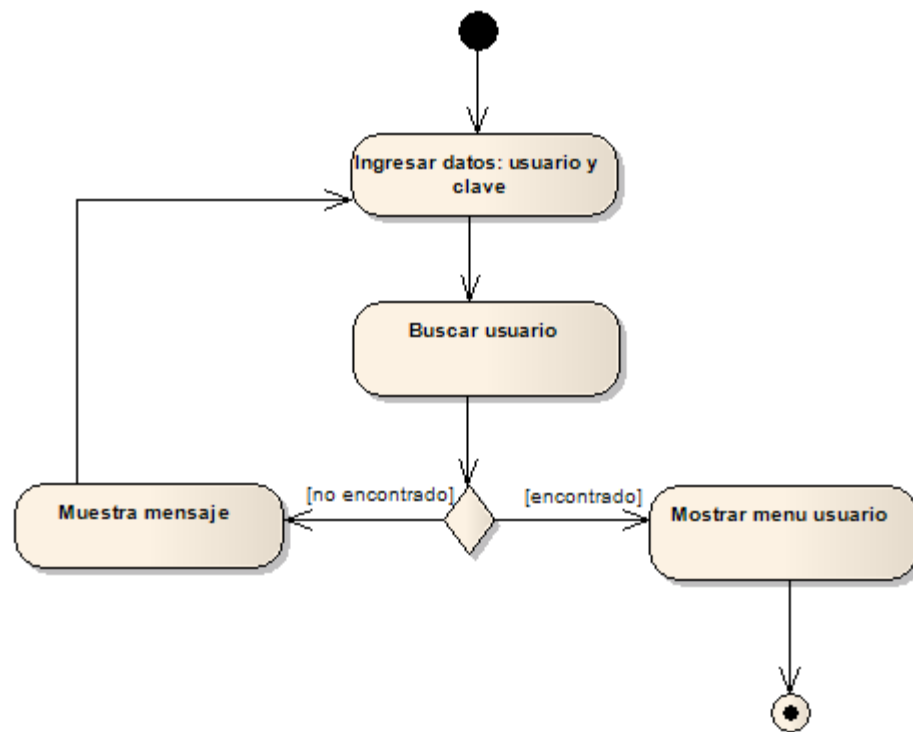


Figura21

3.2.3. Diagrama de actividades: Asignar Usuario y Clave

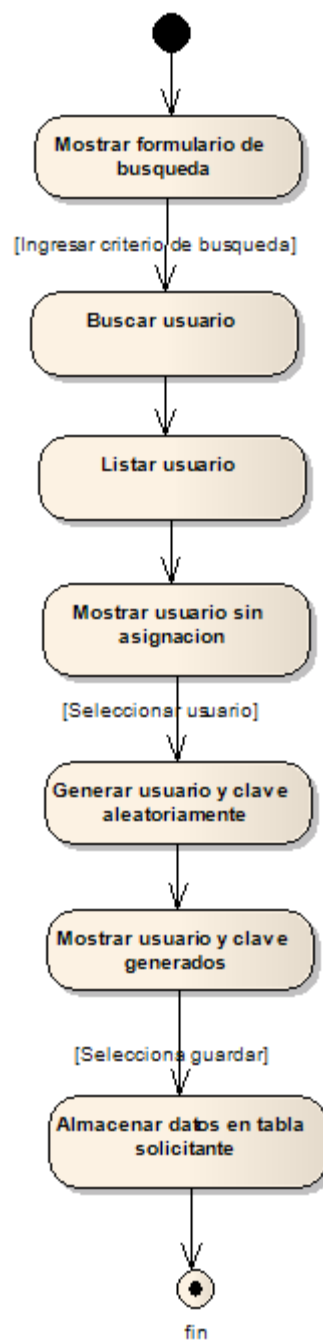


Figura22

3.2.4. Diagrama de actividades: Verificar Partida

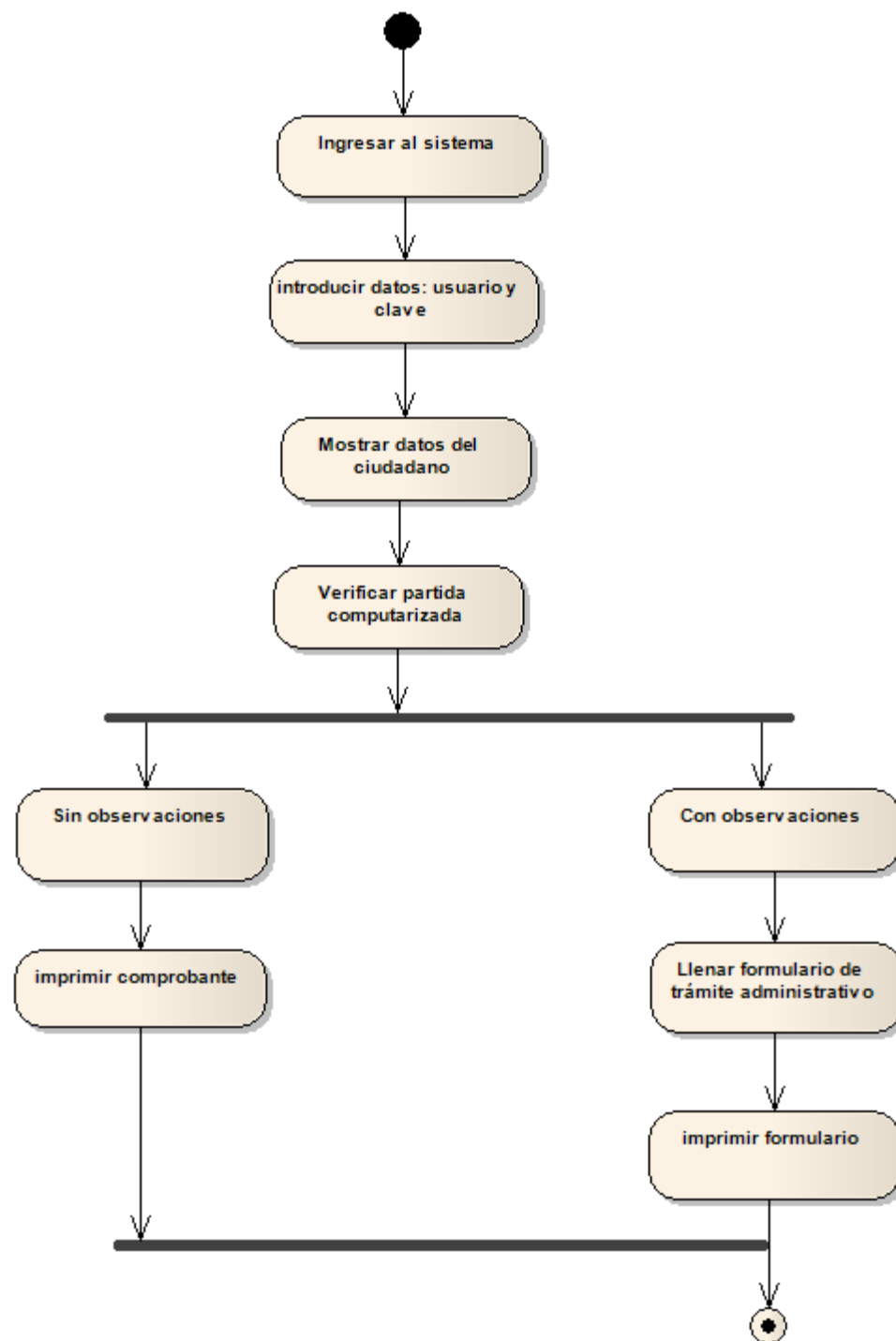


Figura23

3.2.5. Diagrama de actividades: Consultar trámite

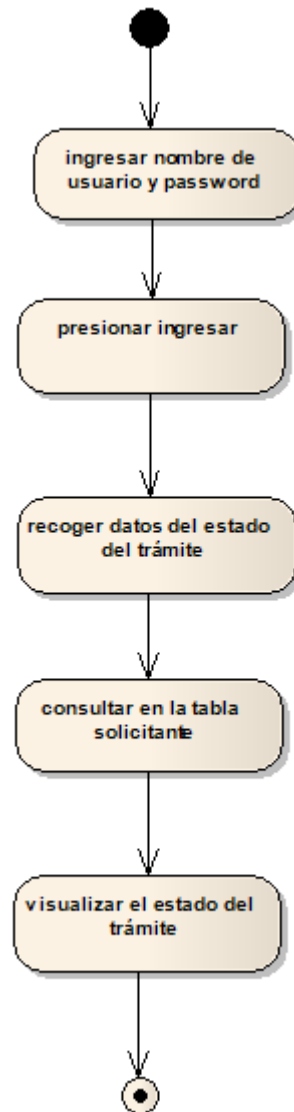


Figura24

3.3. DIAGRAMA DE CLASES.

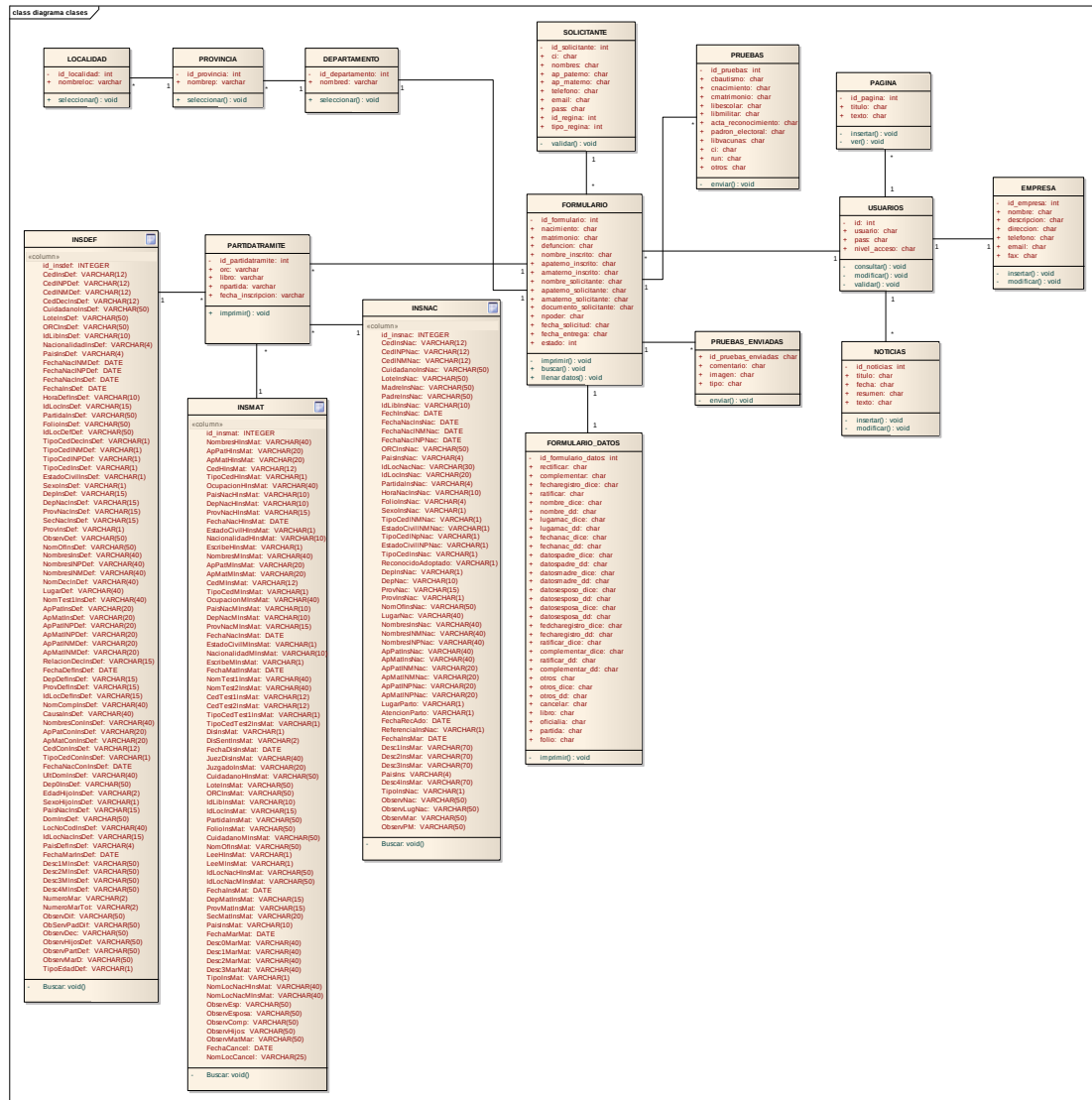


Figura25

3.3.1. ESPECIFICACIÓN DE REQUERIMIENTOS.

Administrador

Método	Notas
insertar() void <i>Public</i>	Esta operación permite insertar un nuevo registro de administrador.
modificar() void <i>Public</i>	Esta operación permite modificar los datos de un registro de administrador seleccionado.

ADMINISTRADOR INSERTAR

SI Opción='Insertar';

ENTONCES

Mostrar pantalla Insertar_Administrador

Ingresar los datos del administrador (ocupación)

SI (Campo descripción = ' ')

ENTONCES

Mensaje ('Por favor descripción')

ELSE

Guardar datos del nuevo administrador en la tabla administrador (ocupación, estado);

FIN SI

FIN SI

ADMINISTRADOR MODIFICAR

SI Opción='Modificar'

ENTONCES

SI pg_query>=1

ENTONCES

Buscar el registro del administrador seleccionado en la tabla administrador

Mostrar pantalla modificar_datos con los datos del administrador seleccionado a ser modificado.

Modificar los datos del administrador (ocupación, estado). Guardar datos modificados del administrador en la tabla administrador

FIN SI

FIN SI

Solicitante.

Método	Notas
insertar() <u>void</u> <i>Private</i>	Inserta un nuevo registro del solicitante en la tabla solicitante.
modificar() <u>void</u> <i>Public</i>	Modifica los datos de un registro del solicitante, en la tabla solicitante.
consultar() <u>void</u> <i>Public</i>	Consulta los datos de un registro del solicitante según parámetros.

SOLICITANTE INSERTAR

SI Opción='Insertar';

ENTONCES

Mostrar pantalla Insertar Solicitante

Ingresar los datos del Solicitante (Título)

SI (Campo Título=' ')

ENTONCES

Mensaje ('Por favor título)

ELSE

Guardar datos del nuevo solicitante en la tabla solicitante (Título)

FIN SI

FIN SI

ESTUDIANTE MODIFICAR

Seleccionar el registro modificar

SI Opción='Modificar'

ENTONCES

SI pg_query>=1

ENTONCES

Buscar el registro de docente seleccionado en la tabla solicitante

Mostrar pantalla modificar datos con los datos de solicitante seleccionado para ser modificado.

Modificar los datos de solicitante (Título). Guardar datos modificados de solicitante en la tabla solicitante

FIN SI

FIN SI

CONSULTAR SOLICITANTE

SI Opción='Consultar';

ENTONCES

Realiza la consulta del solicitante seleccionado

FIN SI

Usuario

Método	Notas
insertar()() void <i>Public</i>	Inserta un nuevo registro de usuario, en la tabla usuario.
modificar()() void <i>Public</i>	Modifica los datos de un registro de usuario, en la tabla usuario.
validar () void <i>Private</i>	Valida los datos de usuario como id_usuario, clave usuario, para el logueo.

USUARIO INSERTAR

Seleccionar el tipo de usuario a Insertar

SI Opción='Insertar';

ENTONCES

Mostrar pantalla Insertar_usuario

Ingresar los datos del nuevo Usuario (nombre_usuario, ap_paterno_usuario, ap_materno_usuario, clave, estado)

Seleccionar teléfonos a ingresar de la tabla teléfonos

SI (Campo nombre_usuario=' ')

ENTONCES

Mensaje ('Por favor ingrese el campo nombre_usuario')

ELSE

SI (Campo clave_usuario=' ')

ENTONCES

Mensaje ('Por favor ingrese el campo clave_usuario')

SI (Campo clave_usuario=' ')

ENTONCES

Mensaje ('Por favor ingrese el campo clave_usuario')

ELSE

Guardar datos del nuevo usuario en la tabla usuario (nombre_usuario, ap_paterno_usuario, ap_materno_usuario, dirección) y guardar los teléfonos en la tabla teléfono;

FIN SI

FIN SI

FIN SI

FIN SI

USUARIO MODIFICAR

Seleccionar el registro del usuario a modificar

SI Opción='Modificar'

ENTONCES

SI pg_query>=1

ENTONCES

Buscar el registro del usuario seleccionado en la tabla usuario

Mostrar pantalla modificar_datos con los datos del usuario seleccionado a ser modificado.

Modificar los datos del usuario (nombre_usuario, ap_paterno_usuario, ap_materno_usuario, clave_usuario, dirección). Guardar datos modificados del usuario en la tabla usuario.

FIN SI

FIN SI

USUARIO VALIDAR

SI Opción='Validar';

ENTONCES

Ingresar los datos del usuario (id_usuario, clave_usuario) para que sean verificados en la base de datos.

FIN SI

3.4.ARQUITECTURA DEL SISTEMA

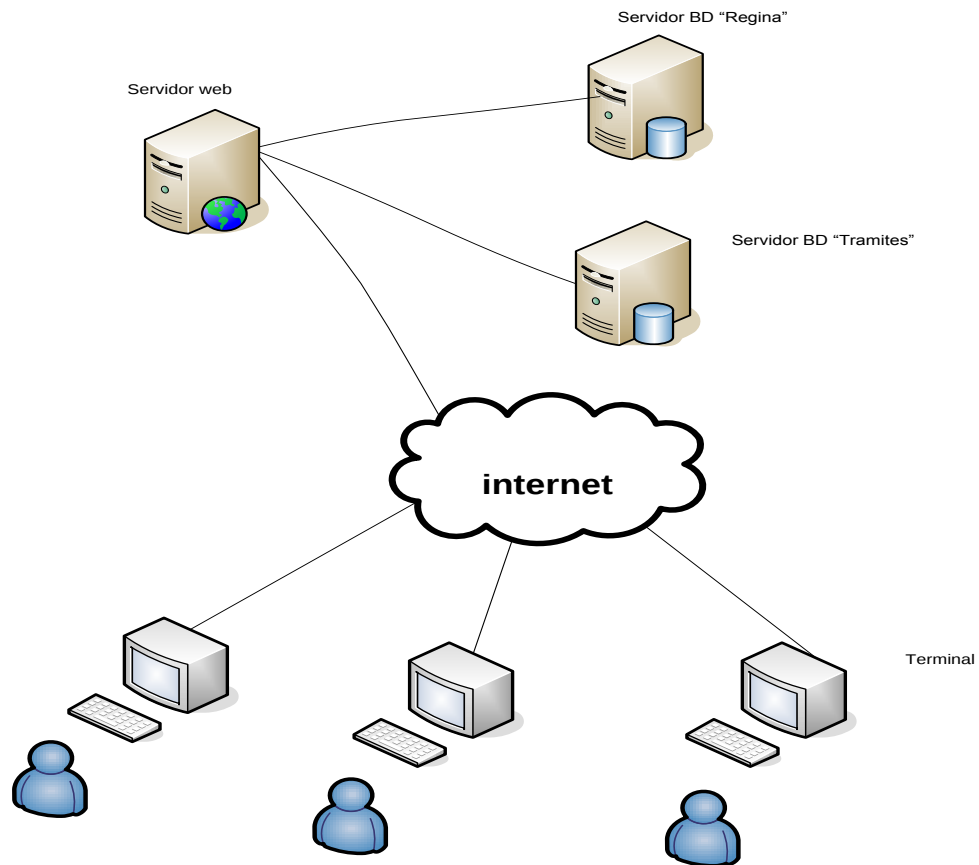


Figura26

DESCRIPCION "SERVIDOR PC-JOSE"

Procesador

Tipo: Pentium IV

Memoria: 960 MB libres

Disco Duro: 3 GHZ

Plataforma: Windows XP Profesional

3.5.DISEÑO DE LA BASE DE DATOS

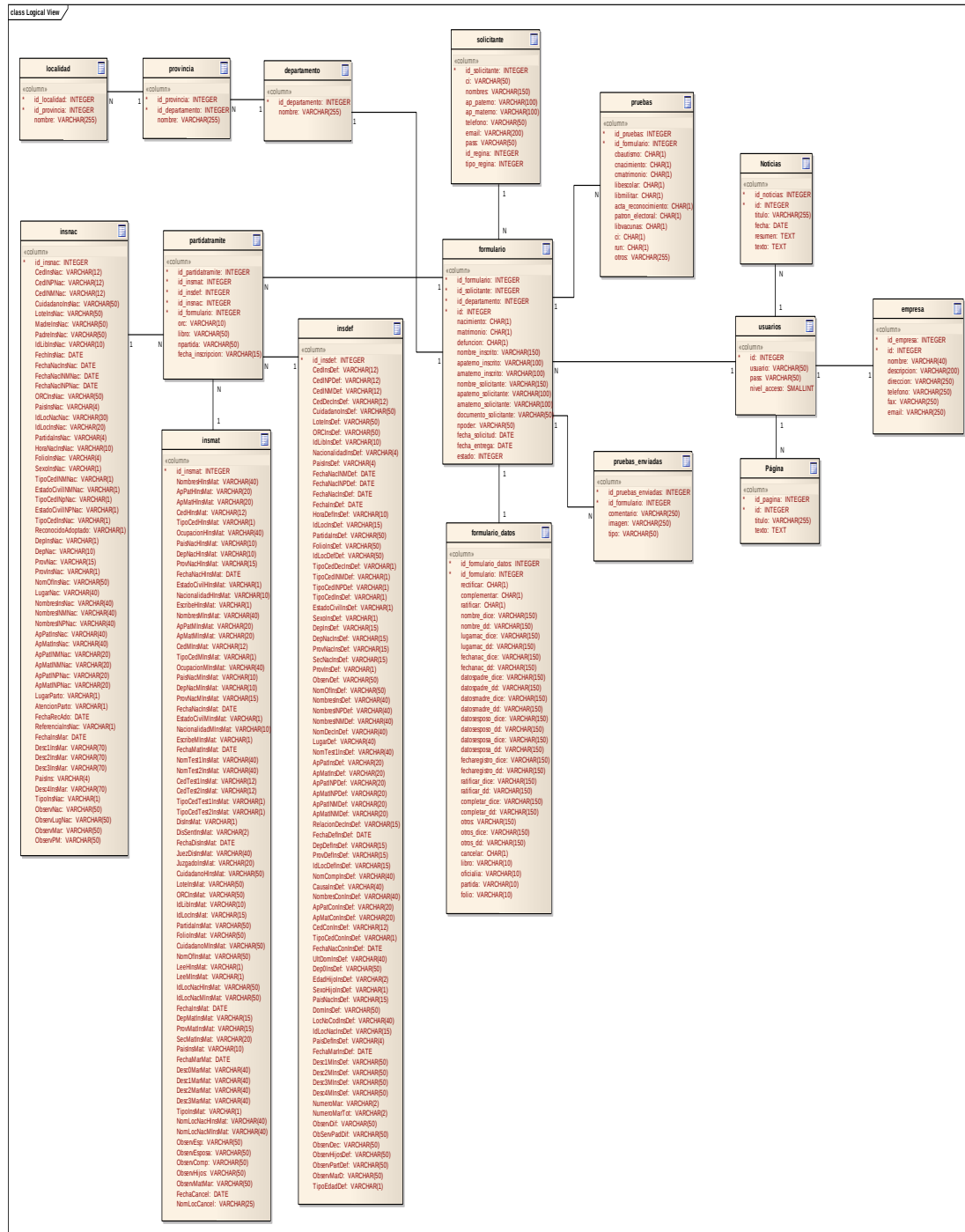


Figura27

3.6. DESCRIPCIÓN DE LAS TABLAS DEL SISTEMA Y DICCIONARIO DE DATOS

Tabla	Esquema	Columnas	Filas	Paginas	Con OID
<u>Formulario</u>	public	4	2	0	√
<u>departamento</u>	public	4	2	0	√
<u>Localidad</u>	public	4	15	0	√
<u>Noticias</u>	public	4	3	0	√
<u>partidatrámite</u>	public	4	6	2	√
<u>Provincia</u>	public	4	3	0	√
<u>Pruebas</u>	public	4	11	0	√
<u>Solicitante</u>	public	4	5	0	√
<u>Usuarios</u>	public	4	3	0	√
<u>Insnac</u>	public	4	3	0	√
<u>Pruebas enviadas</u>	public	4	4	2	√
<u>Formulario datos</u>	public	4	10	0	√
<u>Insdef</u>	public	4	50	0	√
<u>Insmat</u>	public	4	80	2	√
<u>Empresa</u>	public	4	10	0	√

Tabla: formulario

Campo	Tipo	Nulo	Predeterminado
<u>id_formulario</u>	int(11)	Sí	<i>NULL</i>
id_solicitante	int(11)	Sí	
nacimiento	char(1)	Sí	
matrimonio	char(1)	Sí	
defuncion	char(1)	Sí	
nombre_inscrito	varchar(150)	Sí	
apaterno_inscrito	varchar(100)	Sí	
amaterno_inscrito	varchar(100)	Sí	
nombre_solicitante	varchar(150)	Sí	
apaterno_solicitante	varchar(100)	Sí	
amaterno_solicitante	varchar(100)	Sí	
documento_solicitante	varchar(50)	Sí	
npoder	varchar(50)	Sí	
id_departamento	int(11)	Sí	
fecha_solicitud	date	Sí	
fecha_entrega	date	Sí	<i>NULL</i>
estado	int(11)	Sí	

Definición

```
CREATE TABLE `formulario` (
  `id` int(11) NOT NULL auto_increment,
  `id_solicitante` int(11) NOT NULL,
  `id_tipotramite` int(11) NOT NULL,
  `nombre_inscrito` varchar(150) NOT NULL,
  `apaterno_inscrito` varchar(100) NOT NULL,
  `amaterno_inscrito` varchar(100) NOT NULL,
  `nombre_solicitante` varchar(150) NOT NULL,
  `apaterno_solicitante` varchar(100) NOT NULL,
  `amaterno_solicitante` varchar(100) NOT NULL,
  `documento_solicitante` varchar(50) NOT NULL,
  `parentesco` varchar(150) NOT NULL,
  `npoder` varchar(50) NOT NULL,
  `id_departamento` int(11) NOT NULL,
  `fecha_solicitud` date NOT NULL,
  `fecha_entrega` date NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Tabla: departamento

Campo	Tipo	Nulo	Predeterminado
<u>id_departamento</u>	int(11)	Sí	<i>NULL</i>
nombre	varchar(255)	Sí	

Definición

```
CREATE TABLE `departamento` (
  `id` int(11) NOT NULL auto_increment,
  `nombre` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id` (`id`),
  KEY `id_2` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Tabla: localidad

Campo	Tipo	Nulo	Predeterminado
<u>id_localidad</u>	int(11)	Sí	<i>NULL</i>
id_provincia	int(11)	Sí	
nombre	varchar(255)	Sí	

Definición

```
CREATE TABLE `localidad` (
  `id` int(11) NOT NULL auto_increment,
  `id_provincia` int(11) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Tabla: noticias

Campo	Tipo	Nulo	Predeterminado
<u>id_noticias</u>	int(11)	Sí	<i>NULL</i>
id_usuario	int(11)	Sí	
titulo	varchar(255)	Sí	
fecha	date	Sí	
resumen	tinytext	Sí	
texto	longtext	Sí	

Definición

```

CREATE TABLE `noticias` (
  `id` int(11) NOT NULL auto_increment,
  `id_usuario` int(11) NOT NULL,
  `titulo` varchar(255) NOT NULL,
  `fecha` date NOT NULL,
  `resumen` tinytext NOT NULL,
  `texto` longtext NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: partidatramite

Campo	Tipo	Nulo	Predeterminado
<u>id_partidatramite</u>	int(11)	Sí	<i>NULL</i>
id_formulario	int(11)	Sí	
orc	varchar(10)	Sí	
libro	varchar(50)	Sí	
npartida	varchar(50)	Sí	
fecha_inscripcion	varchar(15)	Sí	<i>NULL</i>
id_localidad	int(11)	Sí	
id_provincia	int(11)	Sí	
id_departamento	int(11)	Sí	

Definición

```

CREATE TABLE `partidatramite` (
  `id` int(11) NOT NULL auto_increment,
  `id_formulario` int(11) NOT NULL,
  `id_tarea` int(11) NOT NULL,
  `libro` varchar(50) NOT NULL,
  `npartida` varchar(50) NOT NULL,
  `fecha_inscripcion` date NOT NULL,
  `id_localidad` int(11) NOT NULL,
  `id_provincia` int(11) NOT NULL,
  `id_departamento` int(11) NOT NULL,
  `folio` varchar(50) NOT NULL,
  `categoria` varchar(150) NOT NULL,
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: provincia

Campo	Tipo	Nulo	Predeterminado
<u>id_provincia</u>	int(11)	Sí	<i>NULL</i>
id_departamento	int(11)	Sí	
nombre	varchar(255)	Sí	

Definición

```

CREATE TABLE `provincia` (
  `id` int(11) NOT NULL auto_increment,
  `id_departamento` int(11) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: pruebas

Campo	Tipo	Nulo	Predeterminado
<u>id_pruebas</u>	int(11)	Sí	<i>NULL</i>
id_formulario	int(11)	Sí	<i>NULL</i>
cbautismo	char(1)	Sí	
cnacimiento	char(1)	Sí	
cmatrimonio	char(1)	Sí	
libescolar	char(1)	Sí	
libmilitar	char(1)	Sí	
acta_reconocimiento	char(1)	Sí	
patron_electoral	char(1)	Sí	
libvacunas	char(1)	Sí	
ci	char(1)	Sí	
run	char(1)	Sí	
otros	varchar(255)	Sí	

Definición

```

CREATE TABLE `pruebas` (
  `id` int(11) NOT NULL auto_increment,
  `id_partidatramite` int(11) NOT NULL,
  `cbautismo` char(1) NOT NULL,
  `cnacimiento` char(1) NOT NULL,
  `cmatrimonio` char(1) NOT NULL,
  `libescolar` char(1) NOT NULL,
  `libmilitar` char(1) NOT NULL,
  `acta_reconocimiento` char(1) NOT NULL,
  `patron_electoral` char(1) NOT NULL,

```

```

`libvacunas` char(1) NOT NULL,
`ci` char(1) NOT NULL,
`run` char(1) NOT NULL,
`otros` varchar(255) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: solicitante

Campo	Tipo	Nulo	Predeterminado
<u>id_solicitante</u>	int(11)	Sí	<i>NULL</i>
ci	varchar(50)	Sí	
nombres	varchar(150)	Sí	
ap_paterno	varchar(100)	Sí	
ap_materno	varchar(100)	Sí	
telefono	varchar(50)	Sí	
email	varchar(200)	Sí	
pass	varchar(50)	Sí	
id_regina	int(11)	Sí	
tipo_regina	int(11)	Sí	

Definición

```

CREATE TABLE `solicitante` (
  `id` int(11) NOT NULL auto_increment,
  `ci` varchar(50) NOT NULL,
  `pass` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
)

```

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

Tabla: usuarios

Campo	Tipo	Nulo	Predeterminado
<u>id</u>	int(11)	Sí	<i>NULL</i>
usuario	varchar(50)	Sí	
pass	varchar(50)	Sí	
nivel_acceso	smallint(4)	Sí	

Definición

```
CREATE TABLE `usuarios` (
  `id` int(11) NOT NULL auto_increment,
  `usuario` varchar(50) NOT NULL,
  `pass` varchar(50) NOT NULL,
  `nivel_acceso` smallint(4) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Tabla: Insnac

Campo	Tipo	Nulo	Predeterminado
<u>id_insnac</u>	int(11)	Sí	<i>NULL</i>
CedInsNac	varchar(12)	Sí	
CedINPNac	varchar(12)	Sí	
CedINMNac	varchar(12)	Sí	
CuidadanoInsNac	int(11)	Sí	
LoteInsNac	int(11)	Sí	
MadreInsNac	int(11)	Sí	
PadreInsNac	int(11)	Sí	
IdLibInsNac	varchar(10)	Sí	<i>NULL</i>
FechInsNac	date	Sí	<i>NULL</i>
FechaNacInsNac	date	Sí	
FechaNacINMNac	date	Sí	
FechaNacINPNac	date	Sí	
ORCInsNac	int(11)	Sí	
PaisInsNac	varchar(4)	Sí	
IdLocNacNac	varchar(30)	Sí	<i>NULL</i>
IdLocInsNac	varchar(20)	Sí	<i>NULL</i>
PartidaInsNac	varchar(4)	Sí	<i>NULL</i>
HoraNacInsNac	varchar(10)	Sí	<i>NULL</i>
FolioInsNac	varchar(4)	Sí	<i>NULL</i>
SexoInsNac	varchar(1)	Sí	
TipoCedINMNac	varchar(1)	Sí	
EstadoCivilINMNac	varchar(1)	Sí	
TipoCedINpNac	varchar(1)	Sí	
EstadoCivilINPNac	varchar(1)	Sí	

TipoCedInsNac	varchar(1)	Sí	
ReconocidoAdoptado	varchar(1)	Sí	
DepInsNac	varchar(1)	Sí	
DepNac	varchar(10)	Sí	<i>NULL</i>
ProvNac	varchar(15)	Sí	<i>NULL</i>
ProvInsNac	varchar(1)	Sí	
NomOfInsNac	varchar(50)	Sí	
LugarNac	varchar(40)	Sí	
NombresInsNac	varchar(40)	Sí	
NombresINMNac	varchar(40)	Sí	
NombresINPNac	varchar(40)	Sí	
ApPatInsNac	varchar(40)	Sí	
ApMatInsNac	varchar(40)	Sí	
ApPatINMNac	varchar(20)	Sí	
ApMatINMNac	varchar(20)	Sí	
ApPatINPNac	varchar(20)	Sí	
ApMatINPNac	varchar(20)	Sí	
LugarParto	varchar(1)	Sí	
AtencionParto	varchar(1)	Sí	
FechaRecAdo	date	Sí	
ReferenciaInsNac	varchar(1)	Sí	
FechaInsMar	date	Sí	
Desc1InsMar	varchar(70)	Sí	
Desc2InsMar	varchar(70)	Sí	
Desc3InsMar	varchar(70)	Sí	
PaisIns	varchar(4)	Sí	
Desc4InsMar	varchar(70)	Sí	

TipoInsNac	varchar(1)	Sí	
ObservNac	varchar(50)	Sí	<i>NULL</i>
ObservLugNac	varchar(50)	Sí	<i>NULL</i>
ObservMar	varchar(50)	Sí	<i>NULL</i>
ObservPM	varchar(50)	Sí	<i>NULL</i>

Definición

```

CREATE TABLE `insnac` (
  `id_insnac` int(11) NOT NULL auto_increment,
  `CedInsNac` varchar(12) NOT NULL,
  `CedINPNac` varchar(12) NOT NULL,
  `CedINMNac` varchar(12) NOT NULL,
  `CudadanoInsNac` int(11) NOT NULL,
  `LoteInsNac` int(11) NOT NULL,
  `MadreInsNac` int(11) NOT NULL,
  `PadreInsNac` int(11) NOT NULL,
  `IdLibInsNac` varchar(10) default NULL,
  `FechInsNac` date default NULL,
  `FechaNacInsNac` date NOT NULL,
  `FechaNacINMNac` date NOT NULL,
  `FechaNacINPNac` date NOT NULL,
  `ORCInsNac` int(11) NOT NULL,
  `PaisInsNac` varchar(4) NOT NULL,
  `IdLocNacNac` varchar(30) default NULL,
  `IdLocInsNac` varchar(20) default NULL,
  `PartidaInsNac` varchar(4) default NULL,
  `HoraNacInsNac` varchar(10) default NULL,
  `FolioInsNac` varchar(4) default NULL,

```

`SexoInsNac` varchar(1) NOT NULL,
`TipoCedINMNAc` varchar(1) NOT NULL,
`EstadoCivilINMNAc` varchar(1) NOT NULL,
`TipoCedINpNac` varchar(1) NOT NULL,
`EstadoCivilINPNac` varchar(1) NOT NULL,
`TipoCedInsNac` varchar(1) NOT NULL,
`ReconocidoAdoptado` varchar(1) NOT NULL,
`DepInsNac` varchar(1) NOT NULL,
`DepNac` varchar(10) default NULL,
`ProvNac` varchar(15) default NULL,
`ProvInsNac` varchar(1) NOT NULL,
`NomOfInsNac` varchar(50) NOT NULL,
`LugarNac` varchar(40) NOT NULL,
`NombresInsNac` varchar(40) NOT NULL,
`NombresINMNAc` varchar(40) NOT NULL,
`NombresINPNac` varchar(40) NOT NULL,
`ApPatInsNac` varchar(40) NOT NULL,
`ApMatInsNac` varchar(40) NOT NULL,
`ApPatINMNAc` varchar(20) NOT NULL,
`ApMatINMNAc` varchar(20) NOT NULL,
`ApPatINPNac` varchar(20) NOT NULL,
`ApMatINPNac` varchar(20) NOT NULL,
`LugarParto` varchar(1) NOT NULL,
`AtencionParto` varchar(1) NOT NULL,
`FechaRecAdo` date NOT NULL,
`ReferenciaInsNac` varchar(1) NOT NULL,
`FechaInsMar` date NOT NULL,
`Desc1InsMar` varchar(70) NOT NULL,
`Desc2InsMar` varchar(70) NOT NULL,
`Desc3InsMar` varchar(70) NOT NULL,
`PaisIns` varchar(4) NOT NULL,

```

`Desc4InsMar` varchar(70) NOT NULL,
`TipoInsNac` varchar(1) NOT NULL,
`ObservNac` varchar(50) default NULL,
`ObservLugNac` varchar(50) default NULL,
`ObservMar` varchar(50) default NULL,
`ObservPM` varchar(50) default NULL,
PRIMARY KEY (`id_insnac`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: pruebas_enviadas

Campo	Tipo	Nulo	Predeterminado
<u>id_pruebas_enviadas</u>	int(11)	Sí	<i>NULL</i>
id_formulario	int(11)	Sí	
comentario	varchar(250)	Sí	
imagen	varchar(250)	Sí	
tipo	varchar(50)	Sí	<i>NULL</i>

Definición

```

CREATE TABLE `pruebas_enviadas` (
  `id` int(11) NOT NULL auto_increment,
  `comentario` int(11) NOT NULL,
  `imagen` (255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: formulario_datos

Campo	Tipo	Nulo	Predeterminado
<u>id_formulario_datos</u>	int(11)	Sí	<i>NULL</i>
id_formulario	int(11)	Sí	
rectificar	char(1)	Sí	
complementar	char(1)	Sí	
ratificar	char(1)	Sí	
nombre_dice	varchar(150)	Sí	
nombre_dd	varchar(150)	Sí	
lugarnac_dice	varchar(150)	Sí	
lugarnac_dd	varchar(150)	Sí	
fechanac_dice	varchar(150)	Sí	
fechanac_dd	varchar(150)	Sí	
datospadre_dice	varchar(150)	Sí	
datospadre_dd	varchar(150)	Sí	
datosmadre_dice	varchar(150)	Sí	
datosmadre_dd	varchar(150)	Sí	
datosesposos_dice	varchar(150)	Sí	
datosesposos_dd	varchar(150)	Sí	
datosesposas_dice	varchar(150)	Sí	
datosesposas_dd	varchar(150)	Sí	
feharegistro_dice	varchar(150)	Sí	
feharegistro_dd	varchar(150)	Sí	
ratificar_dice	varchar(150)	Sí	
ratificar_dd	varchar(150)	Sí	
completar_dice	varchar(150)	Sí	
completar_dd	varchar(150)	Sí	

otros	varchar(150)	Sí	
otros_dice	varchar(150)	Sí	
otros_dd	varchar(150)	Sí	
cancelar	char(1)	Sí	
libro	varchar(10)	Sí	
oficialia	varchar(10)	Sí	
partida	varchar(10)	Sí	
folio	varchar(10)	Sí	

Definición

```
CREATE TABLE `formulario_datos` (
  `id` int(11) NOT NULL auto_increment,
  `id_formulario` (11) NOT NULL,
  `rectificar` (50) NOT NULL,
  `complementar` varchar(150) NOT NULL,
  `ratificar` (100) NOT NULL,
  `nombre_dice` varchar(100) NOT NULL,
  `nombre_dd` (255) NOT NULL,
  `lugarnac_dice` varchar(100) NOT NULL,
  `lugarnac_dd` varchar(100) NOT NULL,
  `fechanac_dice` varchar(255) NOT NULL,
  `fechanac_dd` varchar(255) NOT NULL,
  `datospadre_dice` varchar(255) NOT NULL,
  `datospadre_dd` varchar(255) NOT NULL,
  `datosmadre_dice` varchar(255) NOT NULL,
  `datosmadre_dd` varchar(255) NOT NULL,
  `datosesposado_dice` varchar(255) NOT NULL,
  `datosesposado_dd` varchar(255) NOT NULL,
```

```

`datosesposa_dice` varchar(255) NOT NULL,
`datosesposa_dd` varchar(255) NOT NULL,
`fecharegistro_dice` varchar(255) NOT NULL,
`fecharegistro_dd` varchar(255) NOT NULL,
`ratificar_dice` varchar(255) NOT NULL,
`ratificar_dd` varchar(255) NOT NULL,
`complementar_dice` varchar(255) NOT NULL,
`complementar_dd` varchar(255) NOT NULL,
`otros` varchar(255) NOT NULL,
`otros_dice` varchar(255) NOT NULL,
`otros_dd` varchar(255) NOT NULL,
`cancelar` varchar(255) NOT NULL,
`libro` varchar(255) NOT NULL,
`oficialia` varchar(255) NOT NULL,
`partida` varchar(255) NOT NULL,
`folio` varchar(255) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: Insdef

Campo	Tipo	Nulo	Predeterminado
<u>id_insdef</u>	int(11)	Sí	<i>NULL</i>
CedInsDef	varchar(12)	Sí	
CedINPDef	varchar(12)	Sí	
CedINMDef	varchar(12)	Sí	
CedDecInsDef	varchar(12)	Sí	
CuidadanoInsDef	int(11)	Sí	
LoteInsDef	int(11)	Sí	

ORCInsDef	int(11)	Sí	
IdLibInsDef	varchar(10)	Sí	<i>NULL</i>
NacionalidadInsDef	varchar(4)	Sí	
PaisInsDef	varchar(4)	Sí	
FechaNacINMDef	date	Sí	
FechaNacINPDef	date	Sí	
FechaNacInsDef	date	Sí	
FechaInsDef	date	Sí	
HoraDefInsDef	varchar(10)	Sí	
IdLocInsDef	varchar(15)	Sí	<i>NULL</i>
PartidaInsDef	int(11)	Sí	
FolioInsDef	int(11)	Sí	
IdLocDefDef	int(11)	Sí	
TipoCedDecInsDef	varchar(1)	Sí	
TipoCedINMDef	varchar(1)	Sí	
TipoCedINPDef	varchar(1)	Sí	
TipoCedInsDef	varchar(1)	Sí	
EstadoCivilInsDef	varchar(1)	Sí	
SexoInsDef	varchar(1)	Sí	
DepInsDef	varchar(15)	Sí	<i>NULL</i>
DepNacInsDef	varchar(15)	Sí	<i>NULL</i>
ProvNacInsDef	varchar(15)	Sí	<i>NULL</i>
SecNacInsDef	varchar(15)	Sí	<i>NULL</i>
ProvInsDef	varchar(1)	Sí	
ObservDef	varchar(50)	Sí	
NomOfInsDef	varchar(50)	Sí	
NombresInsDef	varchar(40)	Sí	

NombresINPDef	varchar(40)	Sí	
NombresINMDef	varchar(40)	Sí	
NomDecInDef	varchar(40)	Sí	
LugarDef	varchar(40)	Sí	
NomTest1InsDef	varchar(40)	Sí	
ApPatInsDef	varchar(20)	Sí	
ApMatInsDef	varchar(20)	Sí	
ApPatINPDef	varchar(20)	Sí	
ApMatINPDef	varchar(20)	Sí	
ApPatINMDef	varchar(20)	Sí	
ApMatINMDef	varchar(20)	Sí	
RelacionDecInsDef	varchar(15)	Sí	
FechaDefInsDef	date	Sí	
DepDefInsDef	varchar(15)	Sí	NULL
ProvDefInsDef	varchar(15)	Sí	NULL
IdLocDefInsDef	varchar(15)	Sí	NULL
NomCompInsDef	varchar(40)	Sí	
CausaInsDef	varchar(40)	Sí	
NombresConInsDef	varchar(40)	Sí	
ApPatConInsDef	varchar(20)	Sí	
ApMatConInsDef	varchar(20)	Sí	
CedConInsDef	varchar(12)	Sí	
TipoCedConInsDef	varchar(1)	Sí	
FechaNacConInsDef	date	Sí	
UltDomInsDef	varchar(40)	Sí	
Dep0InsDef	varchar(50)	Sí	
EdadHijoInsDef	varchar(2)	Sí	

SexoHijoInsDef	varchar(1)	Sí	
PaisNacInsDef	varchar(15)	Sí	<i>NULL</i>
DomInsDef	varchar(50)	Sí	
LocNoCodInsDef	varchar(40)	Sí	
IdLocNacInsDef	varchar(15)	Sí	<i>NULL</i>
PaisDefInsDef	varchar(4)	Sí	
FechaMarInsDef	date	Sí	
Desc1MInsDef	varchar(50)	Sí	
Desc2MInsDef	varchar(50)	Sí	
Desc3MInsDef	varchar(50)	Sí	
Desc4MInsDef	varchar(50)	Sí	
NumeroMar	varchar(2)	Sí	
NumeroMarTot	varchar(2)	Sí	
ObservDif	varchar(50)	Sí	<i>NULL</i>
ObServPadDif	varchar(50)	Sí	<i>NULL</i>
ObservDec	varchar(50)	Sí	<i>NULL</i>
ObservHijosDef	varchar(50)	Sí	<i>NULL</i>
ObservPartDef	varchar(50)	Sí	<i>NULL</i>
ObservMarD	varchar(50)	Sí	<i>NULL</i>
TipoEdadDef	varchar(1)	Sí	

Definición

```
CREATE TABLE `insdef` (
  `id_insdef` int(11) NOT NULL auto_increment,
  `CedInsDef` varchar(12) NOT NULL,
  `CedINPDef` varchar(12) NOT NULL,
  `CedINMDef` varchar(12) NOT NULL,
```

`CedDecInsDef` varchar(12) NOT NULL,
`CudadanoInsDef` int(11) NOT NULL,
`LoteInsDef` int(11) NOT NULL,
`ORCInsDef` int(11) NOT NULL,
`IdLibInsDef` varchar(10) default NULL,
`NacionalidadInsDef` varchar(4) NOT NULL,
`PaisInsDef` varchar(4) NOT NULL,
`FechaNacINMDef` date NOT NULL,
`FechaNacINPDef` date NOT NULL,
`FechaNacInsDef` date NOT NULL,
`FechaInsDef` date NOT NULL,
`HoraDefInsDef` varchar(10) NOT NULL,
`IdLocInsDef` varchar(15) default NULL,
`PartidaInsDef` int(11) NOT NULL,
`FolioInsDef` int(11) NOT NULL,
`IdLocDefDef` int(11) NOT NULL,
`TipoCedDecInsDef` varchar(1) NOT NULL,
`TipoCedINMDef` varchar(1) NOT NULL,
`TipoCedINPDef` varchar(1) NOT NULL,
`TipoCedInsDef` varchar(1) NOT NULL,
`EstadoCivilInsDef` varchar(1) NOT NULL,
`SexoInsDef` varchar(1) NOT NULL,
`DepInsDef` varchar(15) default NULL,
`DepNacInsDef` varchar(15) default NULL,
`ProvNacInsDef` varchar(15) default NULL,
`SecNacInsDef` varchar(15) default NULL,
`ProvInsDef` varchar(1) NOT NULL,
`ObservDef` varchar(50) NOT NULL,
`NomOfInsDef` varchar(50) NOT NULL,
`NombresInsDef` varchar(40) NOT NULL,
`NombresINPDef` varchar(40) NOT NULL,

`NombresINMDef` varchar(40) NOT NULL,
`NomDecInDef` varchar(40) NOT NULL,
`LugarDef` varchar(40) NOT NULL,
`NomTest1InsDef` varchar(40) NOT NULL,
`ApPatInsDef` varchar(20) NOT NULL,
`ApMatInsDef` varchar(20) NOT NULL,
`ApPatINPDef` varchar(20) NOT NULL,
`ApMatINPDef` varchar(20) NOT NULL,
`ApPatINMDef` varchar(20) NOT NULL,
`ApMatINMDef` varchar(20) NOT NULL,
`RelacionDecInsDef` varchar(15) NOT NULL,
`FechaDefInsDef` date NOT NULL,
`DepDefInsDef` varchar(15) default NULL,
`ProvDefInsDef` varchar(15) default NULL,
`IdLocDefInsDef` varchar(15) default NULL,
`NomCompInsDef` varchar(40) NOT NULL,
`CausaInsDef` varchar(40) NOT NULL,
`NombresConInsDef` varchar(40) NOT NULL,
`ApPatConInsDef` varchar(20) NOT NULL,
`ApMatConInsDef` varchar(20) NOT NULL,
`CedConInsDef` varchar(12) NOT NULL,
`TipoCedConInsDef` varchar(1) NOT NULL,
`FechaNacConInsDef` date NOT NULL,
`UltDomInsDef` varchar(40) NOT NULL,
`Dep0InsDef` varchar(50) NOT NULL,
`EdadHijoInsDef` varchar(2) NOT NULL,
`SexoHijoInsDef` varchar(1) NOT NULL,
`PaisNacInsDef` varchar(15) default NULL,
`DomInsDef` varchar(50) NOT NULL,
`LocNoCodInsDef` varchar(40) NOT NULL,
`IdLocNacInsDef` varchar(15) default NULL,

```

`PaisDefInsDef` varchar(4) NOT NULL,
`FechaMarInsDef` date NOT NULL,
`Desc1MInsDef` varchar(50) NOT NULL,
`Desc2MInsDef` varchar(50) NOT NULL,
`Desc3MInsDef` varchar(50) NOT NULL,
`Desc4MInsDef` varchar(50) NOT NULL,
`NumeroMar` varchar(2) NOT NULL,
`NumeroMarTot` varchar(2) NOT NULL,
`ObservDif` varchar(50) default NULL,
`ObServPadDif` varchar(50) default NULL,
`ObservDec` varchar(50) default NULL,
`ObservHijosDef` varchar(50) default NULL,
`ObservPartDef` varchar(50) default NULL,
`ObservMarD` varchar(50) default NULL,
`TipoEdadDef` varchar(1) NOT NULL,
PRIMARY KEY (`id_insdef`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: Insmat

Campo	Tipo	Nulo	Predeterminado
<u>id_insmat</u>	int(11)	Sí	<i>NULL</i>
NombresHInsMat	varchar(40)	Sí	
ApPatHInsMat	varchar(20)	Sí	
ApMatHInsMat	varchar(20)	Sí	
CedHInsMat	varchar(12)	Sí	
TipoCedHInsMat	varchar(1)	Sí	
OcupacionHInsMat	varchar(40)	Sí	
PaisNacHInsMat	varchar(10)	Sí	<i>NULL</i>

DepNacHInsMat	varchar(10)	Sí	<i>NULL</i>
ProvNacHInsMat	varchar(15)	Sí	<i>NULL</i>
FechaNacHInsMat	date	Sí	
EstadoCivilHInsMat	varchar(1)	Sí	
NacionalidadHInsMat	varchar(10)	Sí	<i>NULL</i>
EscribeHInsMat	varchar(1)	Sí	
NombresMInsMat	varchar(40)	Sí	
ApPatMInsMat	varchar(20)	Sí	
ApMatMInsMat	varchar(20)	Sí	
CedMInsMat	varchar(12)	Sí	
TipoCedMInsMat	varchar(1)	Sí	
OcupacionMInsMat	varchar(40)	Sí	
PaisNacMInsMat	varchar(10)	Sí	<i>NULL</i>
DepNacMInsMat	varchar(10)	Sí	<i>NULL</i>
ProvNacMInsMat	varchar(15)	Sí	<i>NULL</i>
FechaNacInsMat	date	Sí	
EstadoCivilMInsMat	varchar(1)	Sí	
NacionalidadMInsMat	varchar(10)	Sí	<i>NULL</i>
EscribeMInsMat	varchar(1)	Sí	
FechaMatInsMat	date	Sí	
NomTest1InsMat	varchar(40)	Sí	
NomTest2InsMat	varchar(40)	Sí	
CedTest1InsMat	varchar(12)	Sí	
CedTest2InsMat	varchar(12)	Sí	
TipoCedTest1InsMat	varchar(1)	Sí	
TipoCedTest2InsMat	varchar(1)	Sí	<i>NULL</i>
DisInsMat	varchar(1)	Sí	

DisSentInsMat	varchar(2)	Sí	
FechaDisInsMat	date	Sí	
JuezDisInsMat	varchar(40)	Sí	
JuzgadoInsMat	varchar(20)	Sí	
CuidadanoHInsMat	int(11)	Sí	
LoteInsMat	int(11)	Sí	
ORCInsMat	int(11)	Sí	
IdLibInsMat	varchar(10)	Sí	<i>NULL</i>
IdLocInsMat	varchar(15)	Sí	<i>NULL</i>
PartidaInsMat	int(11)	Sí	
FolioInsMat	int(11)	Sí	
CuidadanoMInsMat	int(11)	Sí	
NomOfInsMat	varchar(50)	Sí	
LeeHInsMat	varchar(1)	Sí	
LeeMInsMat	varchar(1)	Sí	
IdLocNacHInsMat	int(11)	Sí	
IdLocNacMInsMat	int(11)	Sí	
FechaInsMat	date	Sí	
DepMatInsMat	varchar(15)	Sí	<i>NULL</i>
ProvMatInsMat	varchar(15)	Sí	<i>NULL</i>
SecMatInsMat	varchar(20)	Sí	<i>NULL</i>
PaisInsMat	varchar(10)	Sí	<i>NULL</i>
FechaMarMat	date	Sí	
Desc0MarMat	varchar(40)	Sí	
Desc1MarMat	varchar(40)	Sí	
Desc2MarMat	varchar(40)	Sí	
Desc3MarMat	varchar(40)	Sí	

TipoInsMat	varchar(1)	Sí	
NomLocNacHInsMat	varchar(40)	Sí	
NomLocNacMInsMat	varchar(40)	Sí	
ObservEsp	varchar(50)	Sí	<i>NULL</i>
ObservEsposa	varchar(50)	Sí	<i>NULL</i>
ObservComp	varchar(50)	Sí	<i>NULL</i>
ObservHijos	varchar(50)	Sí	<i>NULL</i>
ObservMatMar	varchar(50)	Sí	<i>NULL</i>
FechaCancel	date	Sí	
NomLocCancel	varchar(25)	Sí	
<u>id_insmat</u>	int(11)	Sí	<i>NULL</i>
NombresHInsMat	varchar(40)	Sí	
ApPatHInsMat	varchar(20)	Sí	
ApMatHInsMat	varchar(20)	Sí	
CedHInsMat	varchar(12)	Sí	
TipoCedHInsMat	varchar(1)	Sí	
OcupacionHInsMat	varchar(40)	Sí	
PaisNacHInsMat	varchar(10)	Sí	<i>NULL</i>
DepNacHInsMat	varchar(10)	Sí	<i>NULL</i>
ProvNacHInsMat	varchar(15)	Sí	<i>NULL</i>
FechaNacHInsMat	date	Sí	
EstadoCivilHInsMat	varchar(1)	Sí	
NacionalidadHInsMat	varchar(10)	Sí	<i>NULL</i>
EscribeHInsMat	varchar(1)	Sí	
NombresMInsMat	varchar(40)	Sí	
ApPatMInsMat	varchar(20)	Sí	
ApMatMInsMat	varchar(20)	Sí	

CedMInsMat	varchar(12)	Sí	
TipoCedMInsMat	varchar(1)	Sí	
OcupacionMInsMat	varchar(40)	Sí	
PaisNacMInsMat	varchar(10)	Sí	<i>NULL</i>
DepNacMInsMat	varchar(10)	Sí	<i>NULL</i>
ProvNacMInsMat	varchar(15)	Sí	<i>NULL</i>
FechaNacInsMat	date	Sí	
EstadoCivilMInsMat	varchar(1)	Sí	
NacionalidadMInsMat	varchar(10)	Sí	<i>NULL</i>
EscribeMInsMat	varchar(1)	Sí	
FechaMatInsMat	date	Sí	
NomTest1InsMat	varchar(40)	Sí	
NomTest2InsMat	varchar(40)	Sí	
CedTest1InsMat	varchar(12)	Sí	
CedTest2InsMat	varchar(12)	Sí	
TipoCedTest1InsMat	varchar(1)	Sí	
TipoCedTest2InsMat	varchar(1)	Sí	<i>NULL</i>
DisInsMat	varchar(1)	Sí	
DisSentInsMat	varchar(2)	Sí	
FechaDisInsMat	date	Sí	
JuezDisInsMat	varchar(40)	Sí	
JuzgadoInsMat	varchar(20)	Sí	
CudadanoHInsMat	int(11)	Sí	
LoteInsMat	int(11)	Sí	
ORCInsMat	int(11)	Sí	
IdLibInsMat	varchar(10)	Sí	<i>NULL</i>
IdLocInsMat	varchar(15)	Sí	<i>NULL</i>

PartidaInsMat	int(11)	Sí	
FolioInsMat	int(11)	Sí	
CudadanoMInsMat	int(11)	Sí	
NomOfInsMat	varchar(50)	Sí	
LeeHInsMat	varchar(1)	Sí	
LeeMInsMat	varchar(1)	Sí	
IdLocNacHInsMat	int(11)	Sí	
IdLocNacMInsMat	int(11)	Sí	
FechaInsMat	date	Sí	
DepMatInsMat	varchar(15)	Sí	<i>NULL</i>
ProvMatInsMat	varchar(15)	Sí	<i>NULL</i>
SecMatInsMat	varchar(20)	Sí	<i>NULL</i>
PaisInsMat	varchar(10)	Sí	<i>NULL</i>
FechaMarMat	date	Sí	
Desc0MarMat	varchar(40)	Sí	
Desc1MarMat	varchar(40)	Sí	
Desc2MarMat	varchar(40)	Sí	
Desc3MarMat	varchar(40)	Sí	
TipoInsMat	varchar(1)	Sí	
NomLocNacHInsMat	varchar(40)	Sí	
NomLocNacMInsMat	varchar(40)	Sí	
ObservEsp	varchar(50)	Sí	<i>NULL</i>
ObservEsposa	varchar(50)	Sí	<i>NULL</i>
ObservComp	varchar(50)	Sí	<i>NULL</i>
ObservHijos	varchar(50)	Sí	<i>NULL</i>
ObservMatMar	varchar(50)	Sí	<i>NULL</i>
FechaCancel	date	Sí	

Definición

```
CREATE TABLE `insmat` (
  `id_insmat` int(11) NOT NULL auto_increment,
  `NombresHInsMat` varchar(40) NOT NULL,
  `ApPatHInsMat` varchar(20) NOT NULL,
  `ApMatHInsMat` varchar(20) NOT NULL,
  `CedHInsMat` varchar(12) NOT NULL,
  `TipoCedHInsMat` varchar(1) NOT NULL,
  `OcupacionHInsMat` varchar(40) NOT NULL,
  `PaisNacHInsMat` varchar(10) default NULL,
  `DepNacHInsMat` varchar(10) default NULL,
  `ProvNacHInsMat` varchar(15) default NULL,
  `FechaNacHInsMat` date NOT NULL,
  `EstadoCivilHInsMat` varchar(1) NOT NULL,
  `NacionalidadHInsMat` varchar(10) default NULL,
  `EscribeHInsMat` varchar(1) NOT NULL,
  `NombresMInsMat` varchar(40) NOT NULL,
  `ApPatMInsMat` varchar(20) NOT NULL,
  `ApMatMInsMat` varchar(20) NOT NULL,
  `CedMInsMat` varchar(12) NOT NULL,
  `TipoCedMInsMat` varchar(1) NOT NULL,
  `OcupacionMInsMat` varchar(40) NOT NULL,
  `PaisNacMInsMat` varchar(10) default NULL,
  `DepNacMInsMat` varchar(10) default NULL,
  `ProvNacMInsMat` varchar(15) default NULL,
  `FechaNacInsMat` date NOT NULL,
  `EstadoCivilMInsMat` varchar(1) NOT NULL,
  `NacionalidadMInsMat` varchar(10) default NULL,
  `EscribeMInsMat` varchar(1) NOT NULL,
  `FechaMatInsMat` date NOT NULL,
```

`NomTest1InsMat` varchar(40) NOT NULL,
`NomTest2InsMat` varchar(40) NOT NULL,
`CedTest1InsMat` varchar(12) NOT NULL,
`CedTest2InsMat` varchar(12) NOT NULL,
`TipoCedTest1InsMat` varchar(1) NOT NULL,
`TipoCedTest2InsMat` varchar(1) default NULL,
`DisInsMat` varchar(1) NOT NULL,
`DisSentInsMat` varchar(2) NOT NULL,
`FechaDisInsMat` date NOT NULL,
`JuezDisInsMat` varchar(40) NOT NULL,
`JuzgadoInsMat` varchar(20) NOT NULL,
`CudadanoHInsMat` int(11) NOT NULL,
`LoteInsMat` int(11) NOT NULL,
`ORCInsMat` int(11) NOT NULL,
`IdLibInsMat` varchar(10) default NULL,
`IdLocInsMat` varchar(15) default NULL,
`PartidaInsMat` int(11) NOT NULL,
`FolioInsMat` int(11) NOT NULL,
`CudadanoMInsMat` int(11) NOT NULL,
`NomOfInsMat` varchar(50) NOT NULL,
`LeeHInsMat` varchar(1) NOT NULL,
`LeeMInsMat` varchar(1) NOT NULL,
`IdLocNacHInsMat` int(11) NOT NULL,
`IdLocNacMInsMat` int(11) NOT NULL,
`FechaInsMat` date NOT NULL,
`DepMatInsMat` varchar(15) default NULL,
`ProvMatInsMat` varchar(15) default NULL,
`SecMatInsMat` varchar(20) default NULL,
`PaisInsMat` varchar(10) default NULL,
`FechaMarMat` date NOT NULL,
`Desc0MarMat` varchar(40) NOT NULL,

```

`Desc1MarMat` varchar(40) NOT NULL,
`Desc2MarMat` varchar(40) NOT NULL,
`Desc3MarMat` varchar(40) NOT NULL,
`TipoInsMat` varchar(1) NOT NULL,
`NomLocNacHInsMat` varchar(40) NOT NULL,
`NomLocNacMInsMat` varchar(40) NOT NULL,
`ObservEsp` varchar(50) default NULL,
`ObservEsposa` varchar(50) default NULL,
`ObservComp` varchar(50) default NULL,
`ObservHijos` varchar(50) default NULL,
`ObservMatMar` varchar(50) default NULL,
`FechaCancel` date NOT NULL,
`NomLocCancel` varchar(25) NOT NULL,
PRIMARY KEY (`id_insmat`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Tabla: empresa

Campo	Tipo	Nulo	Predeterminado
<u>id_empresa</u>	int(11)	Sí	<i>NULL</i>
nombre	varchar(40)	Sí	<i>NULL</i>
descripcion	varchar(200)	Sí	<i>NULL</i>
direccion	varchar(250)	Sí	<i>NULL</i>
telefono	varchar(250)	Sí	<i>NULL</i>
fax	varchar(250)	Sí	<i>NULL</i>
email	varchar(250)	Sí	<i>NULL</i>

Definición

```
CREATE TABLE `empresa` (  
  `id` int(11) NOT NULL auto_increment,  
  `nombre` int(11) NOT NULL,  
  `descripcion` varchar(50) NOT NULL,  
  `direccion` varchar(150) NOT NULL,  
  `telefono` varchar(100) NOT NULL,  
  `fax` varchar(100) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

3.7.DISEÑO Y DESCRIPCIÓN DE PANTALLAS.

P1. PANTALLA PRINCIPAL.

Pantalla principal, sirve para el ingreso de identificador de usuario y clave del usuario.



- **Nombre de usuario:** Permite introducir el usuario.
- **Clave de Acceso:** Permite introducir clave de usuario.

Botón

- **Ingresar:** La primera vez que el ciudadano verifica sus datos en el Sistema y además si tiene pendiente un trámite administrativo debe llenar un formulario. Las siguientes veces el usuario puede consultar el estado de su trámite.

P2. PANTALLA DE LLENADO DE FORMULARIO

Permite llenar el formulario de Trámites Administrativos. Pero para poder acceder a dicha pagina tiene que llenar los siguientes datos:

Corte Departamental Electoral
Registro Civil del Departamento de Tarija

INICIO ANTECEDENTES OFICIALES DIRECCIONES REGIONALES TRAMITES

FORMULARIO DE SOLICITUD DE TRAMITE ADMINISTRATIVO
LEY No. 2616

I.- DATOS DE LA PARTIDA
 CATEGORIA: Nacimiento ☒ Matrimonio ☐ Defuncion ☐
 Nombre(s) y Apellido(s) del inscrito:
 Partida ORC Libro Partida Inscrip. Depart. Provincia Localidad
 Par 1
 Par 2

II.- DATOS QUE SE DEBE
 Rectificar(RE) ☐ Complementar(CO) ☐ Ratificar(RA) ☐
 Datos que Corresponda Dice: Debe decir:
 Nombre(s) del Inscrito
 Lugar de Nacimiento
 Fecha de Nacimiento
 Datos del Padre
 Datos de la Madre
 Datos del Esposo
 Datos de la Esposa
 Fecha de Registro

Ingreso
 Nombre de Usuario:
 Clave de Acceso:
 Ingresar

- **Oficialia Número:** Se debe introducir el número de la oficialia donde fue inscrito. Este campo es obligatorio.
- **Libro Número:** Se debe ingresar el número del libro. Este campo es obligatorio.
- **Partida Número:** Se debe ingresar el número de su partida. Este campo es obligatorio.
- **Folio Número:** Se debe introducir el número de su folio. Este campo es obligatorio.

Botón

- **Continuar:** Almacena los datos del ciudadano en la Base de datos del sistema.

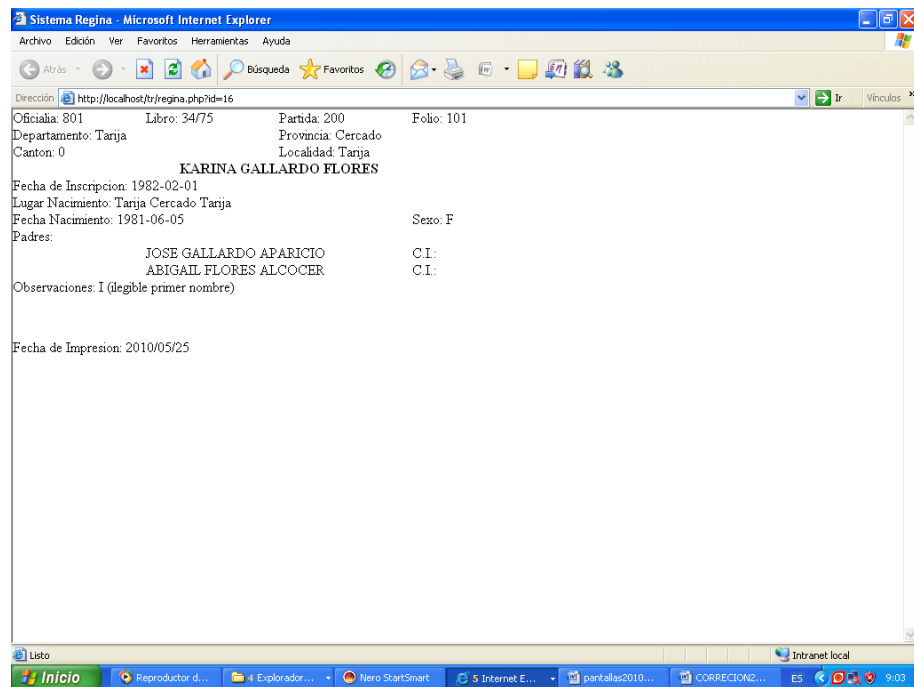
P3. PANTALLA RESULTADOS DE LA BÚSQUEDA

Visualiza algunos campos de las tablas Defunción, Matrimonio y Nacimiento; esto de acuerdo a la categoría que elija el ciudadano.



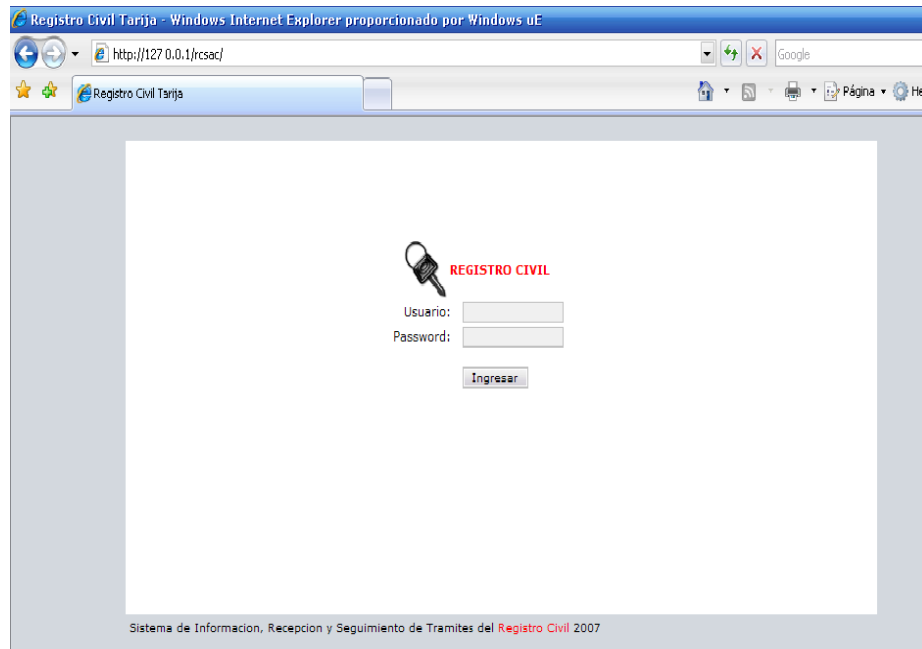
P4. PANTALLA PARTIDA COMPUTARIZADA

Es una colilla del Sistema Regina, la cual tiene que imprimir el ciudadano como una constancia de inicio de su trámite administrativo.



P5. PANTALLA DEL REGISTRO CIVIL

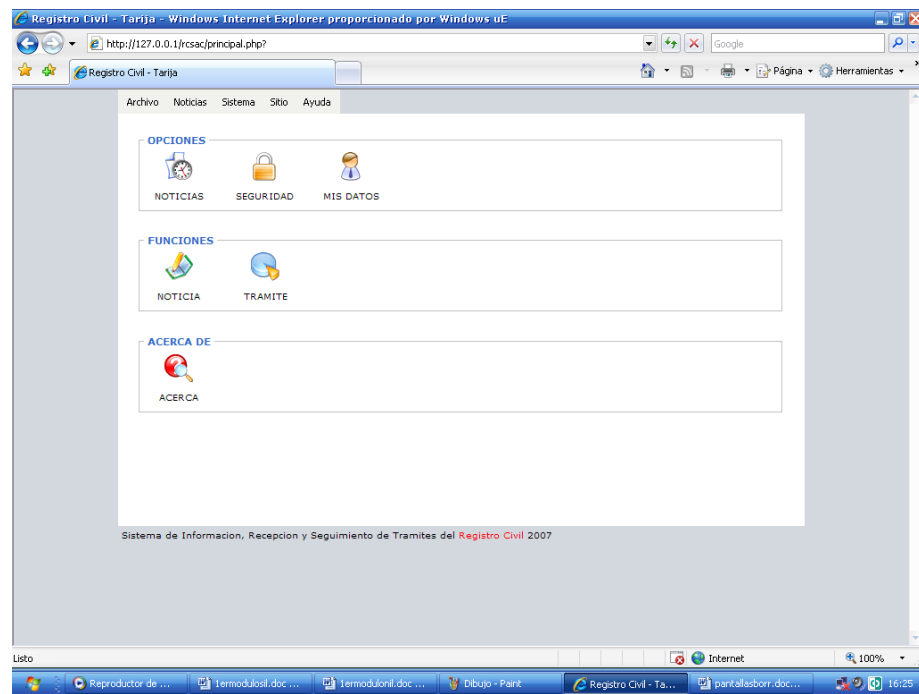
Es una pantalla que no esta disponible para el usuario (ciudadano), es únicamente para el personal del Registro Civil.



- **Usuario:** Permite introducir el usuario.
- **Password:** Permite introducir clave de usuario.

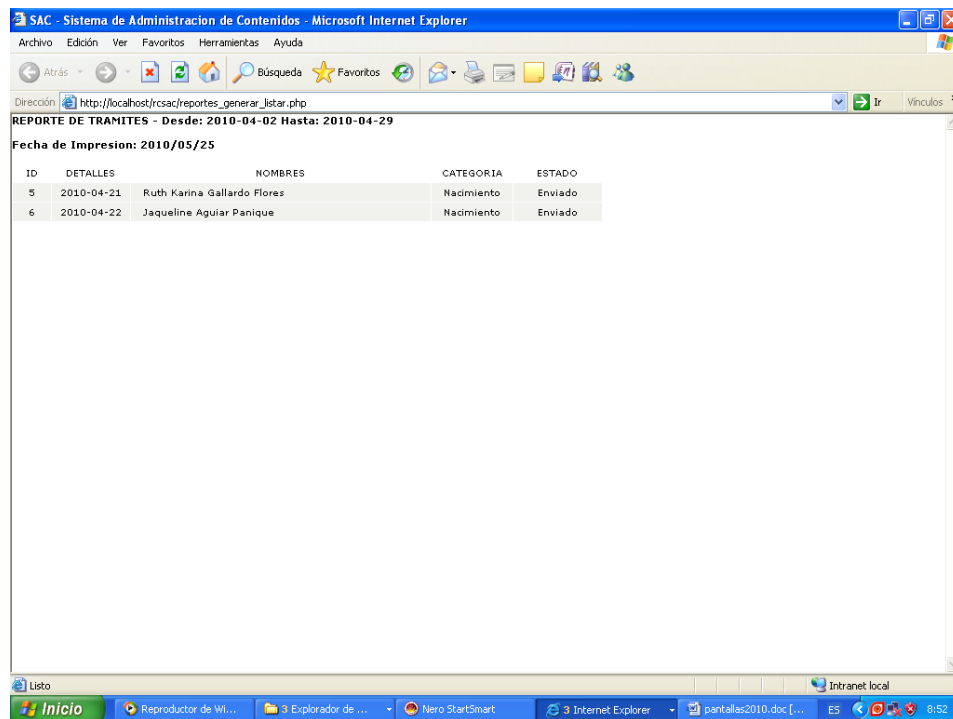
Botón

Ingresar: Permite al administrador acceder a la siguiente pantalla:



P6. PANTALLA REPORTE TRÁMITES

Es una pantalla en la podemos ver reportes por fecha en cualquiera de las 3 categorías es decir; matrimonio, defunción o nacimiento.



SAC - Sistema de Administracion de Contenidos - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

← Atrás → Búsqueda Favoritos

Dirección http://localhost/rcsac/reportes_generar_listar.php Ir Vinculos

REPORTE DE TRAMITES - Desde: 2010-04-02 Hasta: 2010-04-29

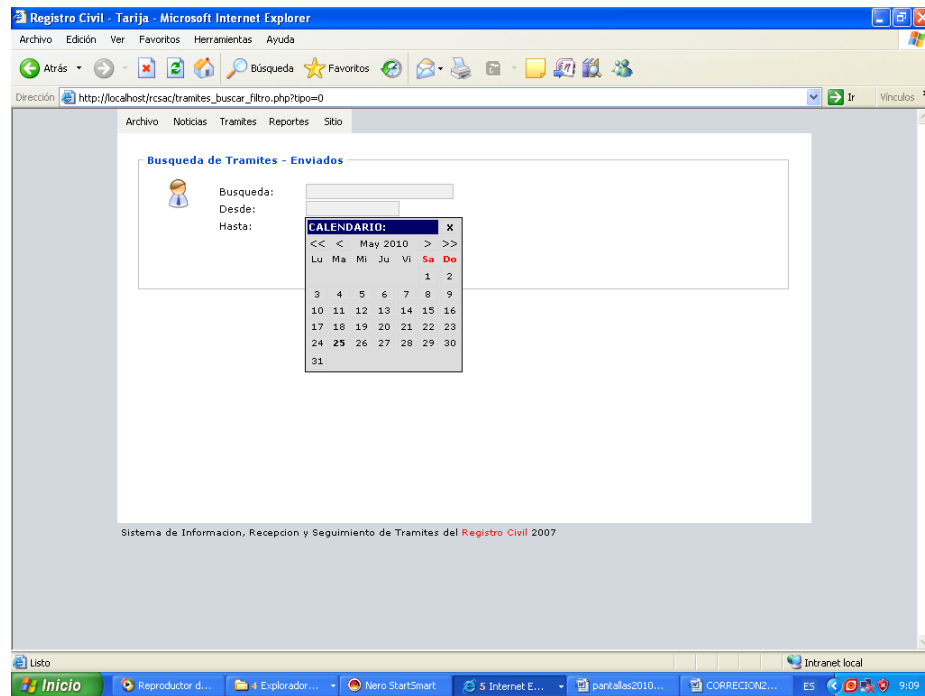
Fecha de Impresion: 2010/05/25

ID	DETALLES	NOMBRES	CATEGORIA	ESTADO
5	2010-04-21	Ruth Karina Gallardo Flores	Nacimiento	Enviado
6	2010-04-22	Jaqueline Aguiar Panique	Nacimiento	Enviado

Inicio Reproductor de Wi... Explorador de ... Nero StartSmart Internet Explorer pantallas2010.doc [... ES 8:52

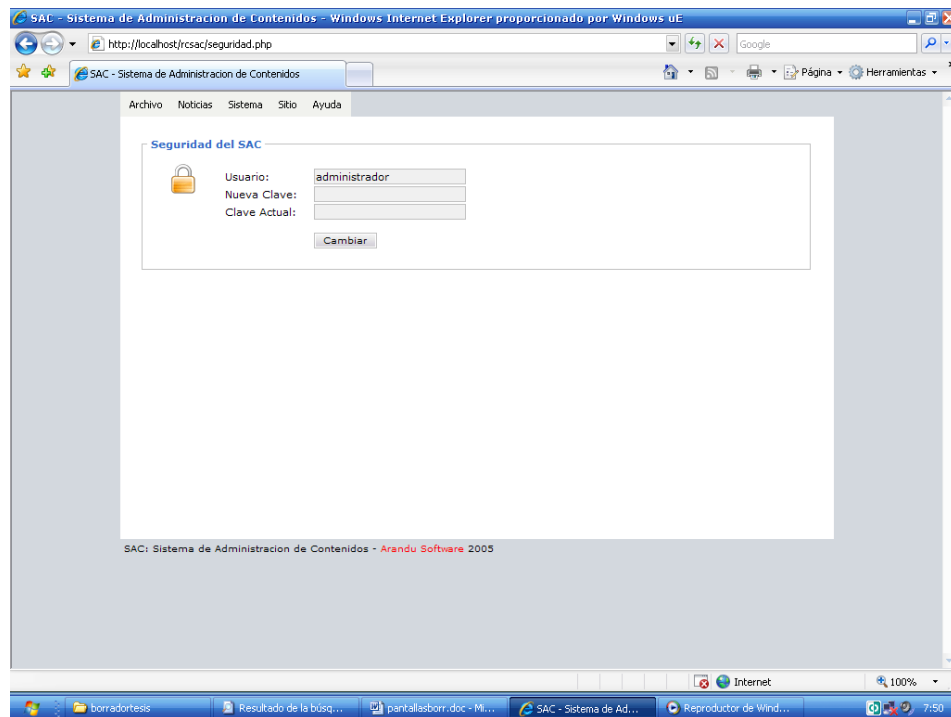
P7. PANTALLA BUSCAR TRÁMITE

Es una pantalla que le permite al usuario buscar trámites por fecha, estos trámites pueden ser: recibidos, rechazados, complementar pruebas, recibidos, iniciados y concluidos.



P8. PANTALLA MODIFICAR CLAVE

En esta pantalla el usuario puede modificar su nombre de usuario y su clave y es valida solo para el Administrador del Sistema.



P9. PANTALLAS DE INFORMACIÓN DEL REGISTRO CIVIL



Corte Departamental Electoral - OFICIAIAS - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://localhost/tr/ver.php?id=3

Corte Departamental Electoral
Registro Civil del Departamento de Tarija

INICIO ANTECEDENTES OFICIAIAS DIRECCIONES REGIONALES TRAMITES

OFICIALES DE REGISTRO CIVIL EMITEN CERTIFICADOS.

La Corte Departamental Electoral de Tarija, dentro del Proyecto de desconcentración del Servicio de Registro Civil, informa a la Ciudadanía que se encuentran en funcionamiento las Oficinas de Registro Civil CON SISTEMA COMPUTARIZADO, dónde el público podrá requerir su CERTIFICADO DUPLICADO. El costo es de Bs. 36.

A objeto de brindarle una mejor y mas rápida atención, a partir del día 15 de Noviembre de 2006 la extensión de CERTIFICADOS DE NACIMIENTO, MATRIMONIO Y DEFUNCIÓN se realizará únicamente en las Oficinas de Registro Civil ubicadas en las siguientes zonas de nuestra ciudad:

- 1.- OFICIALÍA 61102**
BARRIO: CENTRAL.
DIRECCIÓN: Ingavi esq. Daniel Campos n° 231 Semisotano - Edificio Concordia.(Croquis)
- 2.- OFICIALÍA 61101**
BARRIO: CENTRAL.
DIRECCIÓN: Daniel Campos casi la Madrid, frente a la iglesia San Francisco. (Croquis)
- 3.- OFICIALÍA 61132**
BARRIO: SENAC - TABLADITA.
DIRECCIÓN: Avda. Héroes de la Independencia, pasando 6 de Agosto, 3ra. casa a la izquierda s/h. (Croquis)
- 4.- OFICIALÍA 61122**
BARRIO: MERCADO CAMPESINO
DIRECCIÓN: Comercio, oficina de federación de campesinos, 2da. Planta.(Croquis)

Ingreso
Nombre de Usuario:

Clave de Acceso:

Ingresar

Inicio Reproductor d... Explorador... Nero StartSmart 5 Internet E... pantallas2010... CORRECCION2... ES 9:12

Corte Departamental Electoral - DIRECCIONES - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://localhost/tr/ver.php?id=4

Corte Departamental Electoral
Registro Civil del Departamento de Tarija

INICIO ANTECEDENTES OFICIAIAS DIRECCIONES REGIONALES TRAMITES

DIRECCIONES REGIONALES DE REGISTRO CIVIL

La Corte Departamental Electoral de Tarija, dentro del Proyecto de desconcentración del Servicio de Registro Civil, informa a la Ciudadanía que a partir del día Lunes 18 de Diciembre se encuentran en funcionamiento las Direcciones Regionales de Registro Civil en las ciudades de Bermejo y Yacuiba.

Estas direcciones cuentan con el SISTEMA COMPUTARIZADO, dónde el público podrá realizar sus trámites además de requerir su CERTIFICADO DUPLICADO.

DIRECCIÓN REGIONAL DE REGISTRO CIVIL BERMEJO.-
Dirección: Cochabamba entre Chuquisaca y José Yaché.
Teléfono: 6962463.
Horario de Atención: Lunes a Viernes de: 07:30 - 12:30 y 14:30 - 17:30.

DIRECCIÓN REGIONAL DE REGISTRO CIVIL YACUIBA.-
Dirección: Comercio entre San Pedro y Campero.
Teléfono: 6832464.
Horario de Atención: Lunes a Viernes de: 07:30 - 12:30 y 15:30 - 18:30.

Ingreso
Nombre de Usuario:

Clave de Acceso:

Ingresar

Inicio Reproductor d... Explorador... Nero StartSmart 5 Internet E... pantallas2010... CORRECCION2... ES 9:13

P10. PANTALLA ASIGNAR CLAVE Y PASSWORD

Esta pantalla le permite al Administrador del Sistema crear un nombre de usuario y un password, para el usuario (ciudadano), para que el mismo pueda acceder al sistema.

Registro Civil - Tarija - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección http://localhost/rscac/clave_crear.php?id=38&tipo=1

Archivo Noticias Trámites Reportes Sitio

Crear Clave

 C.I.:
Nombres:
Ap. Paterno:
Ap. Materno:
Telefono:
Usuario: 6644775
Clave: 654

Guardar Usuario

Sistema de Información, Recepción y Seguimiento de Trámites del Registro Civil 2007

Inicio Reproductor de... 4 Explorador... Nero StartSmart Internet Ex... pantallas2010... CORRECCION20... ES 9:47

4. CAPÍTULO IV PRUEBAS

4.1. PRUEBAS DE CAJA NEGRA MÉTODO DE PARTICIÓN EQUIVALENTE

Adicionar nueva Noticia

Paso 1:

Nombre Noticia: Tipo alfabético de longitud mayor o igual a 1 y menor o igual a 40.

Condición de entrada	C.E.V	C.E.I
Tipo de nueva Noticia	1.- Letra	2.- Cualquier otra cosa
Tamaño nueva Noticia	3.- Nombre Noticia $\geq$ 1 4.- Nombre Noticia $\leq$ 50	5.- Nombre Noticia=0 6.- Nombre Noticia $>$ 50

Paso 2:

Clases de equivalencias válidas

Nº de Caso de Pruebas	Nombre Menú
CP1	PAGINAS
C.E.V que cumplen: 1,3,4	

Clases de Equivalencias Inválidas

Nº de Caso de Pruebas	Nombre Menú
CP2(2)	\$\$\$\$\$\$
CP3(5)	
CP4(6)	ADMINISTRAR NOTICIA

Adicionar nueva Clave a Usuario**Paso 1:**

Clave Nueva: Tipo alfanumérico de longitud mayor o igual a 5 y menor o igual a 10.

Confirmar Clave: Tipo alfanumérico debe coincidir con Clave Nueva.

Condición de entrada	C.E.V	C.E.I
Tipo de Clave Nueva	1.- Número y/o letra	2.- Cualquier otra cosa
Tamaño Clave Nueva	3.- Clave Nueva $\geq$ 3 4.- Clave Nueva $\leq$ 10	5.- Clave Nueva $<$ 3 6.- Clave Nueva $>$ 10
Tamaño Confirmar Clave	7.- Confirmar Clave $\geq$ 5	8.- Confirmar Clave=0
Tipo de Confirmar Clave	9.- Confirmar Clave=Clave Nueva	10.- Cualquier otra cosa

Paso 2:

Clases de equivalencias válidas

Nº de Caso de Pruebas	Clave nueva	Confirmar Clave
CP1	12345	12345
C.E.V que cumplen: 1,3,4,7,9		

Clases de Equivalencias Inválidas

Nº de Caso de Pruebas	Clave nueva	Confirmar Clave
CP2(2)	\$\$\$\$\$	12345
CP3(5)		12345
CP4(6)	12345....	12345
CP5(8)	12345	
CP6(10)	12345	\$\$\$\$\$

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- El análisis de requerimientos realizado al principio del trabajo, dio como resultado la identificación del problema central dando lugar a desarrollar el sistema en sí.
- El desarrollo del sistema utilizando metodología RUP y lenguaje de modelado UML permitió realizar un sistema más acorde a las necesidades del usuario, además gracias a las diversas iteraciones con que cuenta el método RUP se pudo realizar pruebas y corregir errores.
- La base de datos diseñada sirve de plataforma al sistema permitiendo un buen manejo de la información, además mantiene la integridad de los datos.
- Se implementaron diferentes herramientas tanto de hardware como software que dieron soporte al desarrollo del sistema.
- Las pruebas de caja negra garantizan la integridad de la base de datos y la calidad de información que contiene.

5.2. RECOMENDACIONES

- En la manipulación del sistema se recomienda tener conocimientos básicos en cuanto al manejo de computadoras e Internet, para poder conocer a fondo los beneficios que puede brindar el sistema al usuario.

- Es muy importante tomar en cuenta, que ya desarrollado el software es necesario invertir diferentes tipos de recursos, para un óptimo desarrollo y funcionamiento del sistema, tales como ser:

Recursos humanos, es necesario un administrador del sistema para el buen funcionamiento del mismo.

Recursos Tecnológicos, Es preciso para el funcionamiento del sistema contar con un servidor.

- Una de las recomendaciones más relevantes es la explotación e implantación del sistema.