



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMUNICACIÓN

TÍTULO DE LA TESIS

**SISTEMA DE CONTROL DE ACCESO DE INVITADOS
DE UN FRACCIONAMIENTO**

Tesis que para obtener el título de
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

JUAN OSCAR RODRÍGUEZ BERMÚDEZ

ASESOR: M.C. PEDRO BELLO LÓPEZ

Octubre 2014



DEDICATORIAS

A mi madre que donde quiera que esté fue un ejemplo a seguir, la vida no le concedió la oportunidad de pisar una escuela pero gracias a su visión, fuerza y carácter que siempre mostro a te la adversidad, nos mostró a todos los hermanos que siempre hay una forma de salir adelante. Gracias por la oportunidad que me diste para estudiar, pero más aún, gracias por tus palabras de aliento que escuche de tus labios y que me permitieron seguir por este camino.

A ti **Trinidad Bermúdez Olvera** que supiste hacerme un hombre de bien, sin vicio y responsable con mi familia, hoy y aunque un poco tarde te rindo un tributo no nada más por este trabajo si no por cada paso que he dado en la vida y que tú siempre estarás acompañándome.

A mi hijo **Oscar Efrén Rodríguez Rosas** que espero la vida me permita ver sus logros y anhelos cumplidos, y que hoy yo sea un ejemplo en su vida, sé que tú tienes todo lo mejor de mí y de tú madre, y serás un hombre exitoso.

Tú eres la razón para seguir adelante, el motor que me permite seguir luchando para que un día tú te sientas orgulloso de mi persona.

Sé que en este momento no puedo compartir mi emoción contigo pero no pierdo la esperanza de que algún día pueda estar a tu lado apoyando tus metas.

Te dedico este esfuerzo por que eres lo único que me mantienes vivo y con la fuerza para seguir adelante. Gracias Oscar Efrén Rodríguez por ser mi hijo.

AGRADECIMIENTOS

A mi hermana Julieta Rodríguez Bermúdez que siempre ha estado y está en lo buenos y malos momentos de mi vida, te agradezco todo lo que has hecho y haces por mí, gracias por ser mi hermana.

Agradezco el apoyo recibido incondicionalmente de mi hermana Alicia Rodríguez porque cuando requerí de su tiempo y de un espacio siempre me lo ofreció sin condición.

A Carlos Rodríguez Bermúdez que siempre dio sin dudar lo que fué y lo que tenía, su ausencia ha sido dolorosa, pero nos dejó un ejemplo a seguir.

A Norma Rodríguez Bermúdez que es una luchadora incansable, que siempre está ahí cuando alguien la necesita.

A todos los sobrinos que siempre me alentaron a seguir adelante y que he sido para ellos de una u otra forma un ejemplo.

A ti Patricia Bolaños que has sabido ser mi esposa y amiga, que piensas que soy una buena persona y que valoras lo poco o mucho que nos da la vida. A tus hijos José, Dalí y Yazmín que me permiten ser parte de esta familia.

Gracias a todos los maestros que participaron con todos los compañeros de este diplomado para lograr un paso importante en nuestras vidas.

A mi asesor M.C. Pedro Bello López que siempre mostro disposición para transmitir su conocimiento, M.C. Alfonso Garcés Báez por ser un motivador innato durante este diplomado, un agradecimiento especial la M.C. Meliza Contreras Gozález que gracias a su apoyo incondicional pude lograr mi servicio social.

MISIÓN

Nuestra misión es satisfacer las necesidades de los fraccionamientos para controlar el acceso y seguridad dentro del mismo y lograr su tranquilidad, todo esto desde la comodidad del hogar.

VISIÓN

Ser una empresa líder en el mercado nacional y adaptarse a cada una de las necesidades de los fraccionamientos.

ÍNDICE
PÁG.

| | |
|---|-----------|
| INTRODUCCIÓN..... | 1 |
| ANTECEDENTES | 2 |
| CAPÍTULO I FUNDAMENTOS TEÓRICOS | 3 |
| 1.1. INGENIERÍA DEL SOFTWARE | 3 |
| 1.2. UML (LENGUAJE UNIFICADO DE MODELADO)..... | 4 |
| 1.2.1. ¿CÓMO DEFINIR UN CASO DE USO?..... | 4 |
| 1.2.2. DIAGRAMAS DE FLUJO A NIVEL CONTEXTO | 5 |
| 1.2.3. CASOS DE USO..... | 5 |
| 1.3. VISUAL PARADIGM | 7 |
| 1.4. BASES DE DATOS | 8 |
| CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA..... | 11 |
| 2.1. REQUERIMIENTOS | 11 |
| 2.2. OBJETIVOS GENERALES Y ESPECÍFICOS DEL PROYECTO..... | 12 |
| 2.3. DISEÑO DEL DIAGRAMA DEL CONTEXTO | 13 |
| 2.4. DISEÑO DEL CASO DE USO..... | 14 |
| 2.5. DESARROLLO DE LOS ESCENARIOS DE CASOS DE USO | 15 |

| | | |
|--|--|----|
| 2.6. | DESARROLLO DEL SISTEMA DE CLASE | 19 |
| 2.7. | DESARROLLO DE LOS DIAGRAMAS DE SECUENCIA..... | 20 |
| 2.8. | DESARROLLO DE LA BASE DE DATOS..... | 22 |
| 2.8.1 | NORMALIZACIÓN..... | 23 |
| 2.8.2 | PRIMERA FORMA NORMAL(1FN)..... | 23 |
| 2.8.3 | SEGUNDA FORMA NORMAL(2FN)..... | 24 |
| 2.8.4 | TERCERA FORMA NORMAL(3FN) | 24 |
| 2.8.5 | MODELO ENTIDAD-RELACIÓN..... | 25 |
| 2.8.6 | DIAGRAMA DE ENTIDAD- RELACIÓN..... | 27 |
| 2.8.7 | IMPLEMENTACIÓN DE LA BASE DE DATOS (CONTROL) | 28 |
| CAPÍTULO III IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA..... | | 31 |
| 3.1. | LENGUAJE HTML..... | 31 |
| 3.2. | WAMPSEVER | 31 |
| 3.2.1 | APACHE | 32 |
| 3.2.2 | MYSQL Y PHPMYADMIN | 32 |
| 3.2.3 | PHP..... | 32 |
| 3.3. | HOJAS DE ESTILO | 32 |
| 3.4. | PHOTOSHOP E ILLUSTRATOR | 33 |
| 3.5. | IMPLEMENTACIÓN DEL SISTEMA..... | 33 |
| 3.5.1. | PAGINA PRINCIPAL | 33 |
| 3.5.2. | ADMINISTRADOR | 35 |
| 3.5.3. | CONSULTA DE USUARIOS..... | 36 |
| 3.5.4. | GENERAR CUENTA | 37 |
| 3.5.5. | ELIMINAR CUENTA..... | 38 |
| 3.5.6. | MODIFICAR CUENTA..... | 39 |
| 3.5.7. | MOSTRAR INVITADOS | 41 |
| 3.5.8. | IMPRESIÓN DE LISTAS DE INVITADOS (ADMINISTRADOR) | 42 |
| 3.5.9. | RESIDENTES | 44 |
| 3.5.10. | ALTA DE EVENTO | 45 |
| 3.5.11. | MODIFICAR EVENTO..... | 45 |
| 3.5.12. | ELIMINAR EVENTO..... | 47 |

| | |
|---|----|
| 3.5.13. ALTA DE INVITADOS | 49 |
| 3.5.14. BAJA DE INVITADOS..... | 50 |
| 3.5.15. LISTA DE INVITADOS POR EVENTO..... | 52 |
| 3.5.16. IMPRIMIR LISTAS DE INVITADOS | 54 |
| 3.5.17. VIGILANCIA..... | 55 |
| 3.5.18. CONSULTA EVENTO (VIGILANCIA)..... | 56 |
| 3.5.19. IMPRIMIR LISTA DE INVITADOS (VIGILANCIA)..... | 57 |
| CONCLUSIONES | 59 |
| TRABAJO A FUTURO..... | 60 |
| BIBLIOGRAFÍA | 61 |
| APÉNDICE A..... | 62 |
| CREACIÓN DE LA BASE DATOS Y LAS TABLAS..... | 62 |
| APÉNDICE B..... | 66 |

INTRODUCCIÓN

El sistema de control de acceso a invitados de un fraccionamiento consiste en una aplicación web que genera una lista de invitados que asistirán a un evento social. El sistema propuesto permite a los residentes del fraccionamiento dar de alta sus eventos considerando una lista de invitados, en el caso de invitados en automóvil el medio de identificación será la placa del mismo y para el caso de invitados a pie con presentar una identificación de uno de los invitados por familia será suficiente. El sistema tiene que almacenar la información respecto a los residentes quienes tendrán acceso al sistema para registrar sus eventos incluyendo fecha y horas en que los invitados podrán entrar al fraccionamiento y el sistema generará un reporte de los eventos por día para los guardias de seguridad del fraccionamiento.

El sistema se propone ser web para que los residentes tengan el control absoluto de la hora de entrada y salida del sistema y poder hacer adecuaciones.

ANTECEDENTES

En la mayoría de los sistemas donde se investigó [8][9], la forma de manejar el control de entrada para los invitados de un evento a un fraccionamiento es por medio de video-vigilancia, circuito cerrado y por lo tanto el costo puede llegar a elevarse de forma considerable y el control puede no ser seguro ya que los visitantes entran sin control alguno del residente.

En este sistema la ventaja sobre otros es que el residente debe dar de alta su propio evento social y el podrá definir la lista de invitados que asistirán, además podrá establecer su fecha y horario, y modificarlo antes del evento en caso de algún imponderable, esto da una ventaja sobre otros sistemas ya que baja el costo de operación, además proporciona información impresa para agilizar la labor de los vigilantes al contar con la lista de invitados para verificar su asistencia.

CAPÍTULO I FUNDAMENTOS TEÓRICOS

Iniciaremos con el desarrollo del sistema utilizando la ingeniería de Software, diagramas de casos, diagramas de uso y dando forma con diagramas de contexto.

Los pasos necesarios para desarrollar el sistema propuesto son:

- Determinar la metodología de la ingeniería de software a utilizar [3].
- Elaborar la especificación de requerimientos.
- Realizar el análisis del sistema.
- Diseñar la base de datos relacional.
- Diseñar la aplicación.
- Implementar los diseños realizados.
- Realizar pruebas del sistema [4].
- Escribir el documento de tesis

1.1. INGENIERÍA DEL SOFTWARE

Ingeniería de Software es la forma que permite aprender métodos y técnicas para desarrollar, actualizar y mantener software de calidad. Es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas.

Se cree que las técnicas orientadas a objetos funcionan bien en situaciones en las que los sistemas de información complejos pasan a través de un continuo proceso de mantenimiento, adaptación y rediseño. Las metodologías orientadas a objetos utilizan el estándar de la industria para modelar sistemas orientados a objetos, conocido como lenguaje de modelado unificado (UML), para descomponer un sistema en un modelo de caso de uso [4].

1.2. UML (Lenguaje Unificado de Modelado)

La programación orientada a objetos difiere de la programación tradicional por procedimientos en cuanto a que examina a los objetos que forman parte de un sistema. Cada objeto representa una cosa o evento real dentro del sistema. Los objetos pueden ser clientes, artículos, pedidos, etcétera. Los objetos se representan y agrupan mediante clases, las cuales son una forma ideales para poder reutilizar y la facilidad de mantenimiento.

Una clase define el conjunto de atributos y comportamientos compartidos que se encuentran en cada objeto de la clase.

Para el diseño y análisis de software se utiliza la herramienta que ayudan en su creación, tales como algunos programas.

1.2.1. ¿CÓMO DEFINIR UN CASO DE USO?

En esta parte el analista debe identificar a los actores y los eventos principales, por lo regular se dibuja un diagrama con figuras tales como líneas que representan a los actores y la relación entre ellos.

La forma más fácil de crear la metodología orientada a objetos es enfocarse en la interacción pequeña y rápida de desarrollo al cual se le conoce como el método del espiral como se muestra en la figura (figura1.1).

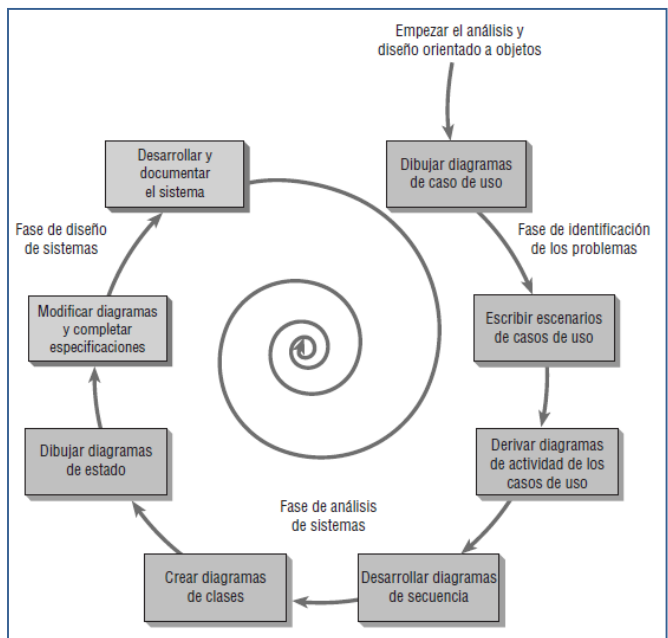


Figura 1.1 Pasos para el proceso de desarrollo de UML [4]

1.2.2. DIAGRAMAS DE FLUJO A NIVEL CONTEXTO

Los diagramas de flujo normalmente muestran el flujo de datos que fluyen en los sistemas y salen de él, además la forma de procesar la información, aquí se presenta los componentes básicos que servirán para diseñar todo el sistema computacional y utilizarlos para analizar la precisión e integridad del sistema (Figura 1.2).

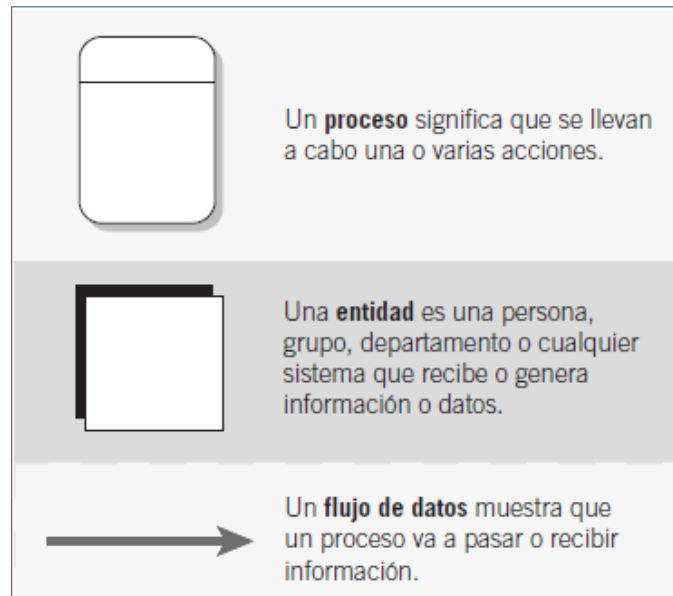


Figura 1.2 Símbolos Básicos un diagrama de flujo de datos [4]

1.2.3. CASOS DE USO

Normalmente un diagrama de caso de uso contienen los símbolos del actor y del caso de uso, combinado con las líneas conectoras. El actor juega el rol específico de los usuarios del sistema por lo que podría ser un empleado, un cliente, etc.

Los actores se dividen en grupos, el grupo de actores principales que normalmente suministran datos o reciben información del sistema, los secundarios o de soporte que son los que dan mantenimiento al sistema para mantenerlo en funcionamiento.

La tabla muestra los elementos para los modelos de los casos de uso (tabla 1.1).




| CONSTRUIR | DESCRIPCIÓN | SINTAXIS |
|---------------------|---|---|
| LIMITES DEL SISTEMA | Representa el límite entre el sistema físico y los actores que interactúan con el sistema físico |  |
| ACTORES | Un conjunto coherente de roles que los usuarios de los casos de uso juegan al interactuar con estos casos de uso |  |
| CASOS DE USO | Una secuencia de acciones, incluyendo variantes, que un sistema (u otra entidad) puede llevar a cabo, interactuando con los actores del sistema |  |

Tabla 1.1 Símbolos Básicos de casos de uso

Los cuatro tipos de símbolos básicos de relación de comportamiento se muestran en la tabla 1.2.


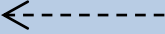
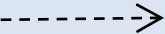

| Relación | Símbolo | Significado |
|------------|---|---|
| Comunica |  | Para conectar un actor con un caso de uso se utiliza una línea si punta |
| Incluye | << Incluye >>  | Un caso de uso contiene un comportamiento común para más de un caso de uso. La flecha apunta al caso de uso común. |
| Extiende | << Extiende >>  | Un caso de uso distintos maneja las excepciones del caso de uso básico. La flecha apunta del caso de uso extendido al básico. |
| Generaliza |  | Una "cosa" de UML es más general que otra "cosa". L flecha apunta a la "cosa" común. |

Tabla 1.2 Símbolos Básicos de relación de casos de uso.

1.3. VISUAL PARADIGM

Visual Paradigm es una herramienta de creación de diseños y diagramas UML; es ideal para los programadores del lenguaje Java, C y C++. A través de una interfaz sencilla, podemos crear diagramas de clases, casos de uso, secuencias, composición y componentes, entre otros.

Sin duda, Visual Paradigm tiene todos los elementos necesarios para facilitarnos la creación de diagramas de la documentación del programa que estemos creando [10].

Con este programa entre otras cosas se puede crear los modelos de casos de uso (Figura 1.4).

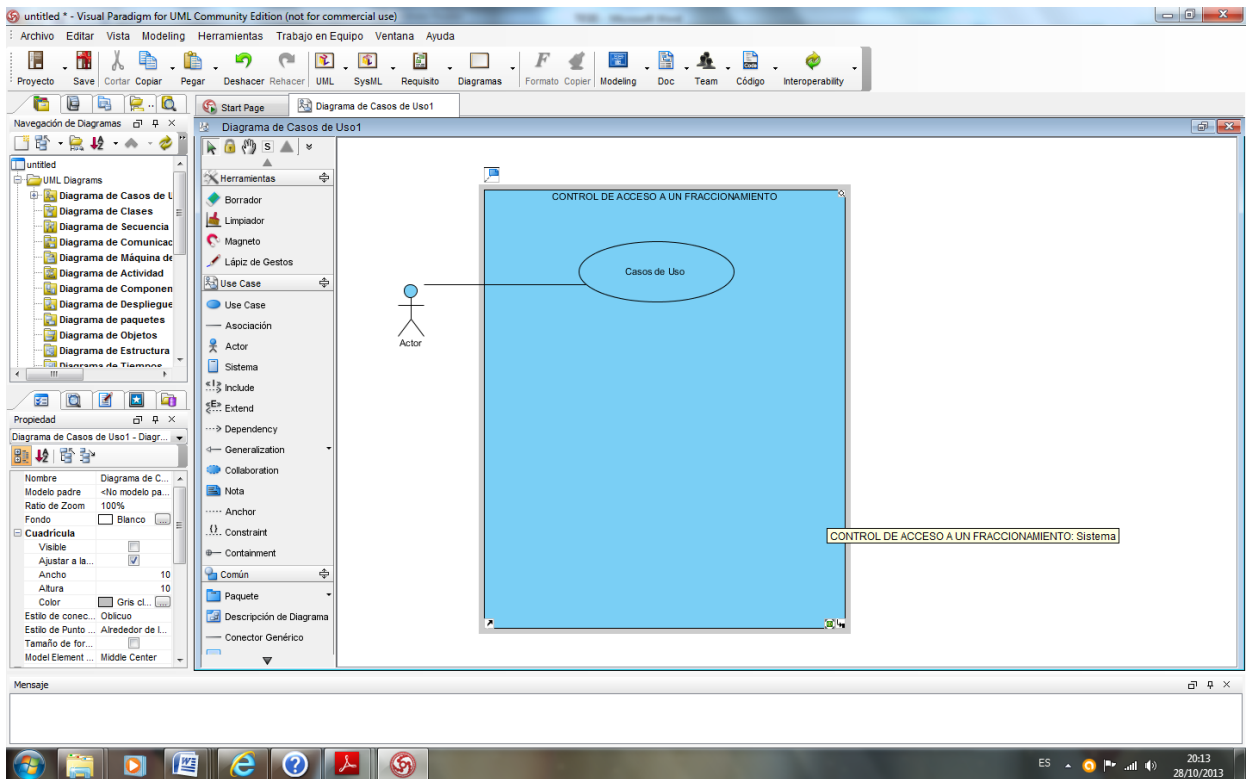


Figura 1.4 Visual Paradigm [10]

1.4. BASES DE DATOS

Las bases de datos son una parte esencial de la vida moderna de la sociedad. Hoy día la mayoría de las persona enfrentamos diversas actividades que implican cierta interacción con una base de datos. Esta interacción tiene que ver con depósitos bancarios, retiro de fondos, hacer reservaciones de hoteles, vuelos, etc. Estas actividades sin duda implican que alguien o algún programa de computador requieran el uso de una base de datos.

Es justo decir que las bases de datos son las culpables del creciente uso de la computadora ya que juegan un papel fundamental en el ámbito empresarial, comercio electrónico, ingeniería, medicina, justicia, educación, etc.

Un sistema de administración de datos (DBMS, database management system) es una colección de programas que permite a los usuarios crear y mantener una base de datos. El DBMS es un sistema de software de propósito general que facilita los procesos de definición, construcción, manipulación y compartición de datos entre varios usuarios y aplicaciones [14].

Una aplicación accede a la base de datos enviando consultas o solicitudes. Otras funciones importantes ofrecidas por el DBMS son la protección de la base de datos y su mantenimiento durante un largo periodo de tiempo. La protección incluye protección del sistema contra el funcionamiento defectuoso del hardware o el software y la protección de la seguridad contra el acceso no autorizado o malintencionado.

Una gran base de datos típica puede tener un ciclo de vida de muchos años, por lo que el DBMS debe ser capaz de mantener el sistema, permitiendo que el sistema evolucione y cambien los requisitos con el tiempo.

La mayoría de los DBMS son sistemas de software muy complejos. Como un ejemplo de nuestras definiciones iniciales, denominaremos sistema de bases de datos a la combinación de base de datos y software DBMS. La Figura 1.4 ilustra algunos de los conceptos que hemos explicado hasta ahora [14].

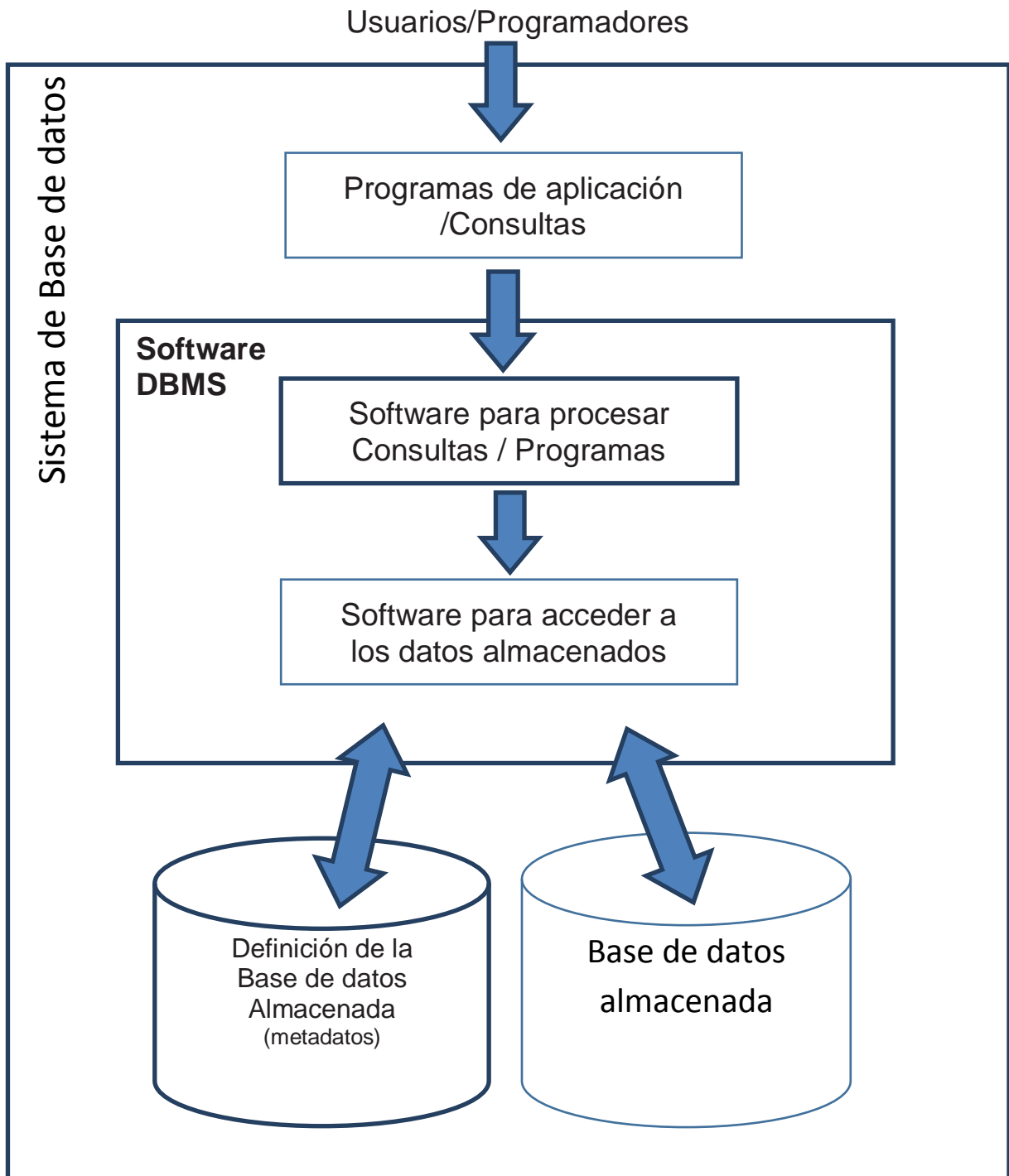


Figura 1.4 Entorno de un sistema de bases de datos simplificado [14]

CAPÍTULO II ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo implementaremos cada uno de los pasos que darán origen al sistema, empezando con los requerimientos, los objetivos, diagramas de contexto, los casos de uso, los escenarios de caso de uso, los diagramas de secuencia, etc.

2.1. REQUERIMIENTOS

Se requiere una aplicación Web, donde cualquier residente conectado a internet pueda dar de alta su evento o consultarlos.

El sistema requiere de tres tipos de usuarios: El administrador que se encarga de controlar el sistema, dar de alta a los residentes, generar su password. El residente que por acción lógica se encuentra registrado él podrá dar de alta su evento. El vigilante que tendrá sólo acceso a los eventos y podrá imprimir listas de invitadas al evento.

Administrador:

Login

Registrar residentes y a elementos de vigilancia (generar login y password)

Modificar residentes

Dar de baja a residentes y vigilantes del sistema

Ver (mostrar) eventos por residente

Generar Lista de Asistentes a Evento

Generar Lista de Placas de automóvil, Marca, Color (para que sea más fácil ubicarse)

Ver eventos por semana

Residente:

- Login
- Actualizar Datos de Registro
- Crear Evento
- Consultar eventos pasados
- Generar lista de asistentes a pie
- Generar Lista de asistentes en auto
- Eliminar eventos y asistentes
- Actualizar Listas de asistentes

Vigilante:

- Login
- Ver eventos del día/semana/mes
- Consultar asistentes en pantalla
- Generar/Imprimir lista de asistentes a pie
- Generar/Imprimir lista de asistentes en auto

2.2. OBJETIVOS GENERALES Y ESPECÍFICOS DEL PROYECTO.

Se desarrollará un sistema vía web [1] que permita registrar a los invitados que asistan a un evento social dentro de un fraccionamiento. El residente anfitrión podrá dar de alta su evento y establecer la lista de sus invitados para que la seguridad del fraccionamiento pueda agilizar la entrada de los mismos.

Además el sistema contará con una base de datos [2] que almacene la información de los residentes y solamente ellos pueden dar de alta el evento. El beneficio principal será para los invitados con auto ya que por medio de su placa podrá pasar de forma inmediata.

El sistema contará con una base de datos relacional con una interfaz amigable para la administración de la información y generar consultas y reportes impresos.

2.3. DISEÑO DEL DIAGRAMA DEL CONTEXTO

El siguiente modelo muestra el flujo de datos que el sistema de forma general realizara, Todos los actores inician con su login (control de acceso a un fraccionamiento). El administrador realiza las altas y recibe información del sistema, el residente de alta su evento y puede consultar y modificar sus eventos, el vigilante consulta los eventos y puede imprimir lista de invitados figura 2.1.

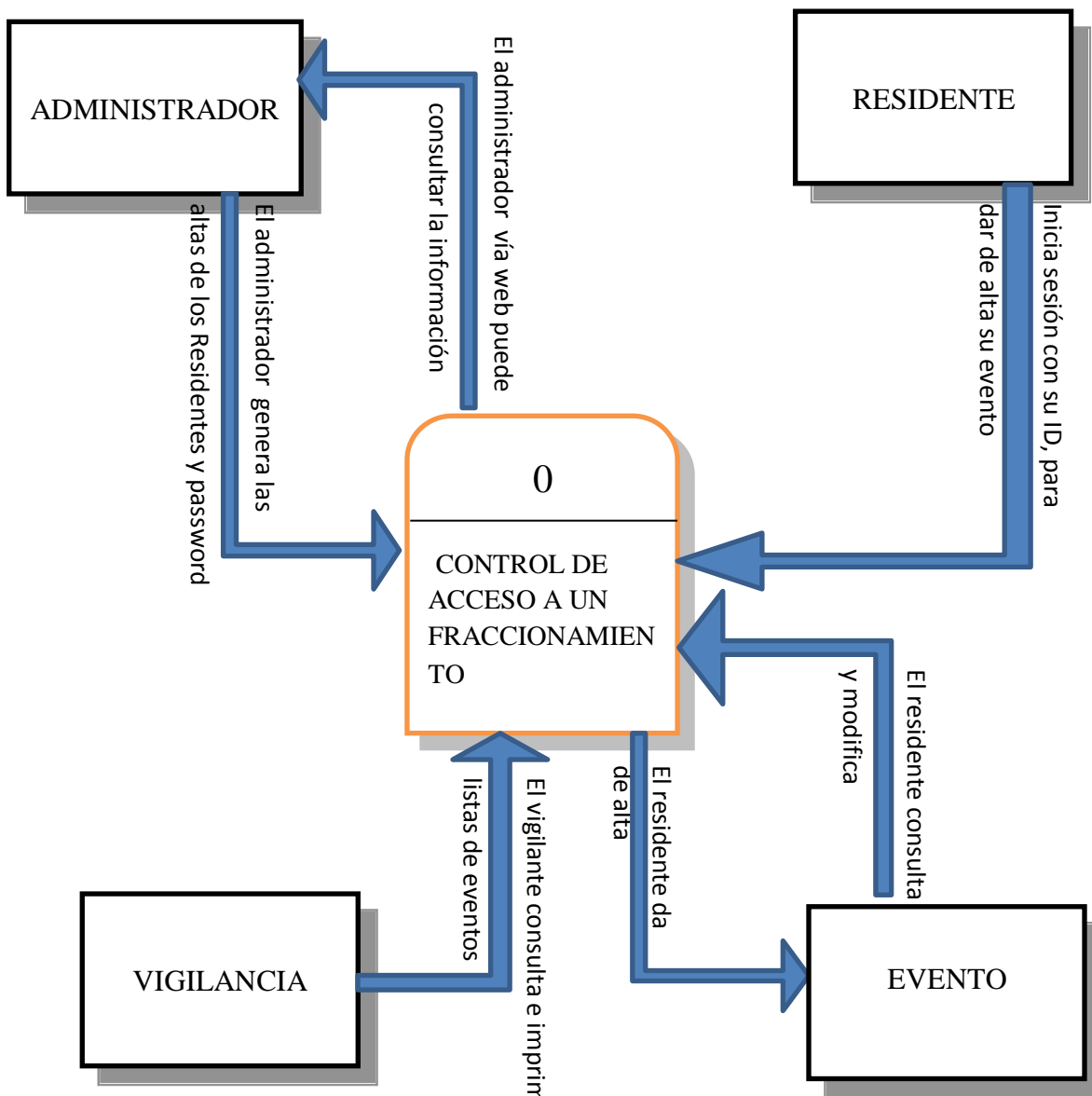


Figura 2.1 Diseño del Diagrama de contexto

2.4. DISEÑO DEL CASO DE USO

A continuación se muestra el Diagrama de Casos de Usos, el cual se diseñó con el programa Visual Paradigm [10].

- Los actores principales son el administrador y los residentes, que al iniciar un evento desencadenan un caso de uso.
- Los eventos principales o de alto nivel son: “Alta de los residente” y “Alta de los eventos”.
- Por último cada caso de uso principal describe a su vez otros casos de uso como se muestra en la figura 2.2.

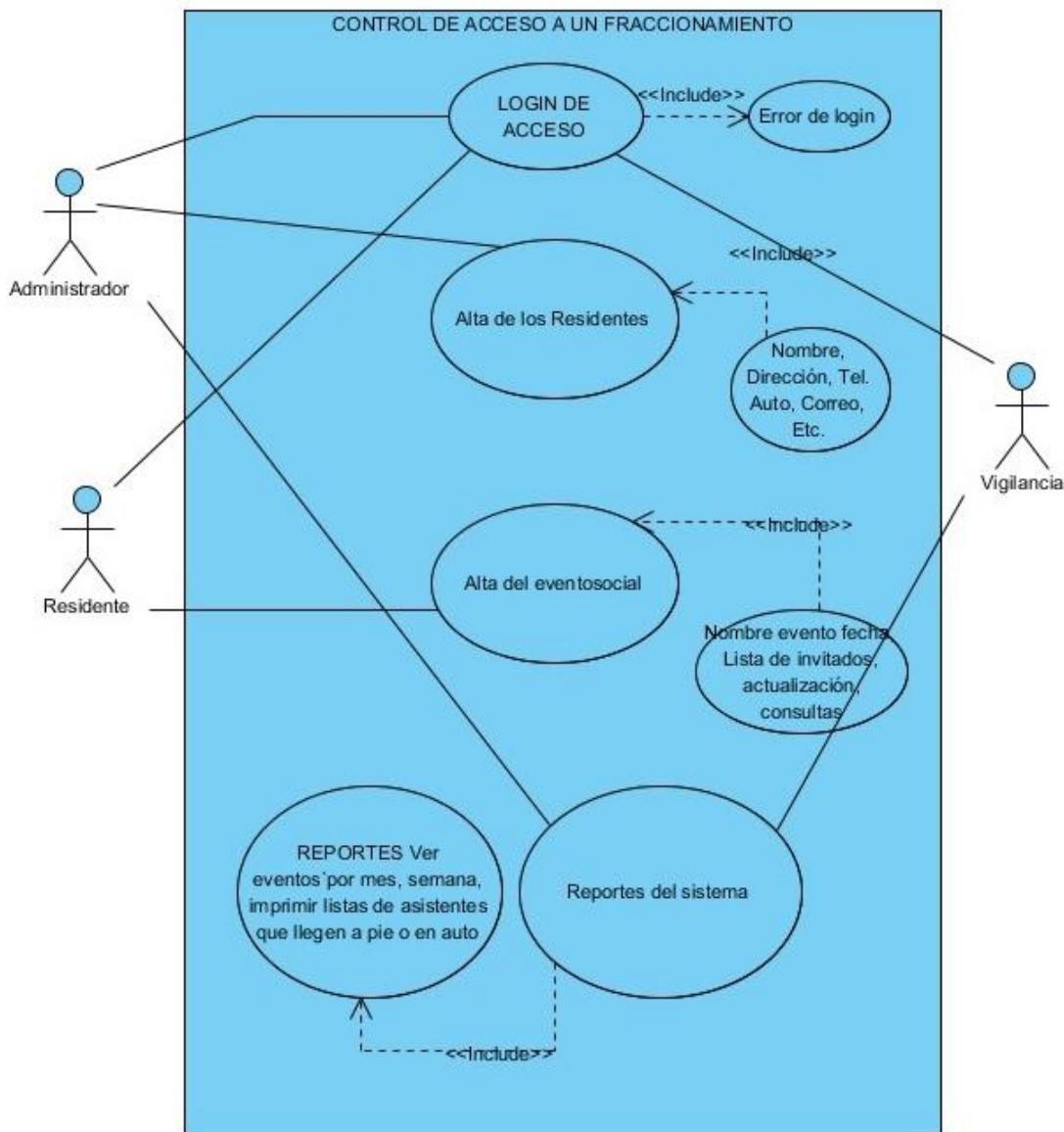


Figura 2.2 Diseño del caso de uso

2.5. DESARROLLO DE LOS ESCENARIOS DE CASOS DE USO

Para cada caso de uso se tiene una descripción o escenario para saber lo que sucede cuando un “actor” interactúa con el sistema.

En seguida se muestran cada uno de los escenarios para cada caso de uso iniciando por el principal y después las rutas alternativas.

| | | |
|---|--|---|
| NOMBRE DEL CASO DE USO: Login de Acceso | | ID ÚNICO:LA01 |
| ÁREA: | | |
| ACTOR(ES): | Administrador, Residentes y Vigilancia | |
| INTERESADOS: | Administrador, Residente, Vigilancia | |
| NIVEL: | Azul | |
| DESCRIPCIÓN: | El Actor introduce su password en línea desde un sitio web seguro | |
| EVENTOS DESENCADENADOS: | El residente utiliza el sitio web para acceder al sistema e introduce su ID de usuario y su contraseña, y hace clic en iniciar sesión. | |
| TIPO DE ESCENARIO: | | |
| Pasos realizados (ruta principal) | | Información para los pasos |
| 1. El residente inicia sesión mediante el servidor web seguro. | | ID de usuario y contraseña. |
| 2. Se lee el registro del Residente y se verifica su contraseña. | | Registro del residente ID y contraseña. |
| 3. Se muestra la información del participante y la información a la que puede acceder. | | Registro del Participante. |
| PRECONDICIONES: | El Actor ya tiene una cuenta de usuario. | |
| POS CONDICIONES: | El Actor tiene acceso al área que corresponde. | |
| SUPOSICIONES: | El Actor tiene un navegador Web, además de un ID de usuario y contraseña válidos. | |
| GARANTÍA DE ÉXITO: | El actor Inicio sesión con el sistema | |
| GARANTÍA MÍNIMA: | | |
| REQUERIMIENTOS CUMPLIDOS: | | |
| CUESTIONES PENDIENTES: | | |
| PRIORIDAD: | Media | |
| RIESGO: | Medio | |

| NOMBRE DEL CASO DE USO: Alta de Residentes | ID ÚNICO:AR02 |
|---|---|
| ÁREA: | |
| ACTOR(ES): Administrador | |
| INTERESADOS: Residente | |
| NIVEL: Azul | |
| DESCRIPCIÓN: El administrador podrá dar de alta a todos residentes en línea desde un sitio web seguro | |
| EVENTOS DESENCADENADOS: El administrador utiliza el sitio web para dar de alta a los residentes. | |
| TIPO DE DESENCADENADO: | |
| Pasos realizados (ruta principal) | Información para los pasos |
| 1. El residente inicia sesión mediante el servidor web seguro. | ID de usuario y contraseña. |
| 2. Se lee el registro del Administrador y se verifica su contraseña. | Registro del residente ID y contraseña. |
| 3. Se muestra la información del correspondiente a la Alta del residente en la página Web. | Formulario para registrar sus datos. |
| 4. El Administrador introduce la información del residente en el formulario Web de registro y hace clic en el botón Enviar. | Nombre, dirección, teléfono, auto, Email. |
| 5. Se valida la información de registro en el servidor Web. | Formulario web del evento |
| | |
| PRECONDICIONES: El residente ya tiene una cuenta de usuario. | |
| POS CONDICIONES: El Administrador dio de alta al residente | |
| SUPOSICIONES: El Administrados tiene un navegador Web, además de un ID de usuario y contraseña válidos. | |
| GARANTÍA DE ÉXITO: SE da de alta y se genera un ID del residente | |
| GARANTÍA MÍNIMA: | |
| REQUERIMIENTOS CUMPLIDOS: | |
| CUESTIONES PENDIENTES: | |
| PRIORIDAD: Media | |
| RIESGO: Medio | |

| | | |
|--|-----------------------|--|
| NOMBRE DEL CASO DE USO: Registrar el Evento Social | | ID ÚNICO: RE03. |
| ÁREA: | | |
| ACTOR(ES): | Residentes | |
| INTERESADOS: | Residente, Vigilancia | |
| NIVEL: | Azul | |
| DESCRIPCIÓN: El residente podrá dar de alta su evento en línea desde un sitio web seguro | | |
| EVENTOS DESENCADENADOS: El residente utiliza el sitio web para dar de alta su evento e introduce su ID de usuario y su contraseña, y hace clic en iniciar sesión. | | |
| TIPO DE ESCENARIO: | | |
| Pasos realizados (ruta principal) | | Información para los pasos |
| 1. El residente inicia sesión mediante el servidor web seguro. | | ID de usuario y contraseña. |
| 2. Se lee el registro del Residente y se verifica su contraseña. | | Registro del residente ID y contraseña. |
| 3. Se muestra la información del participante y la sesión en la página Web de su evento. | | Registro del Residente, registro del evento. |
| 4. El Residente introduce su información en el formulario Web de registro y hace clic en el botón Enviar. | | Formulario para su evento. |
| 5. Se valida la información de registro en el servidor Web. | | Formulario web del evento |
| 6. El Residente introduce la lista de invitados. | | Página Web de confirmación, registro del proceso de registro del evento |
| 7. Se actualiza el registro para el Residente en el Archivo maestro de Residente. | | Página Web de confirmación, registro del evento |
| 8. Se le permite imprimir su información del evento exitoso del Residente. | | Se permite consultar e imprimir sus listas |
| PRECONDICIONES: | | El residente ya tiene una cuenta de usuario. |
| POS CONDICIONES: | | El residente dio de alta su evento social |
| SUPOSICIONES: | | El participante tiene un navegador Web, además de un ID de usuario y contraseña válidos. |
| GARANTÍA DE ÉXITO: | | |
| GARANTÍA MÍNIMA: | | |
| REQUERIMIENTOS CUMPLIDOS: | | |
| CUESTIONES PENDIENTES: | | |
| PRIORIDAD: | Media | |
| RIESGO: | Medio | |

| | | |
|--|-----------------------|--|
| NOMBRE DEL CASO DE USO: Consulta e impresión de lista de el Evento Social | | ID ÚNICO: CILES04. |
| ÁREA: | | |
| ACTOR(ES): | Residentes | |
| INTERESADOS: | Residente, Vigilancia | |
| NIVEL: | Azul | |
| DESCRIPCIÓN: El residente podrá dar de alta su evento en línea desde un sitio web seguro | | |
| EVENTOS DESENCADENADOS: El residente utiliza el sitio web para dar de alta su evento e introduce su ID de usuario y su contraseña, y hace clic en iniciar sesión. | | |
| TIPO DE ESCENARIO: | | |
| Pasos realizados (ruta principal) | | Información para los pasos |
| 1. El residente inicia sesión mediante el servidor web seguro. | | ID de usuario y contraseña. |
| 2. Se lee el registro del Residente y se verifica su contraseña. | | Registro del residente ID y contraseña. |
| 3. Se muestra la información de los eventos del residente y la sesión en la página Web de su evento. | | Se muestra información del evento del Residente previa alta del evento. |
| 4. El Residente podrá imprimir sus listas de invitados en la Web. | | Mostrar su evento y listas de invitados. |
| 5. Se valida la información de registro en el servidor Web. | | Formulario web del evento |
| PRECONDICIONES: | | El residente ya tiene una cuenta de usuario. |
| POS CONDICIONES: | | El residente dio de alta su evento social |
| SUPOSICIONES: | | El participante tiene un navegador Web, además de un ID de usuario y contraseña válidos. |
| GARANTÍA DE ÉXITO: | | Obtener sus listas de invitados |
| GARANTÍA MÍNIMA: | | |
| REQUERIMIENTOS CUMPLIDOS: | | |
| CUESTIONES PENDIENTES: | | |
| PRIORIDAD: | Media | |
| RIESGO: | Medio | |

2.6. DESARROLLO DEL SISTEMA DE CLASE

En la figura 2.3 se tiene el diagrama que muestran las características y relaciones del sistema.

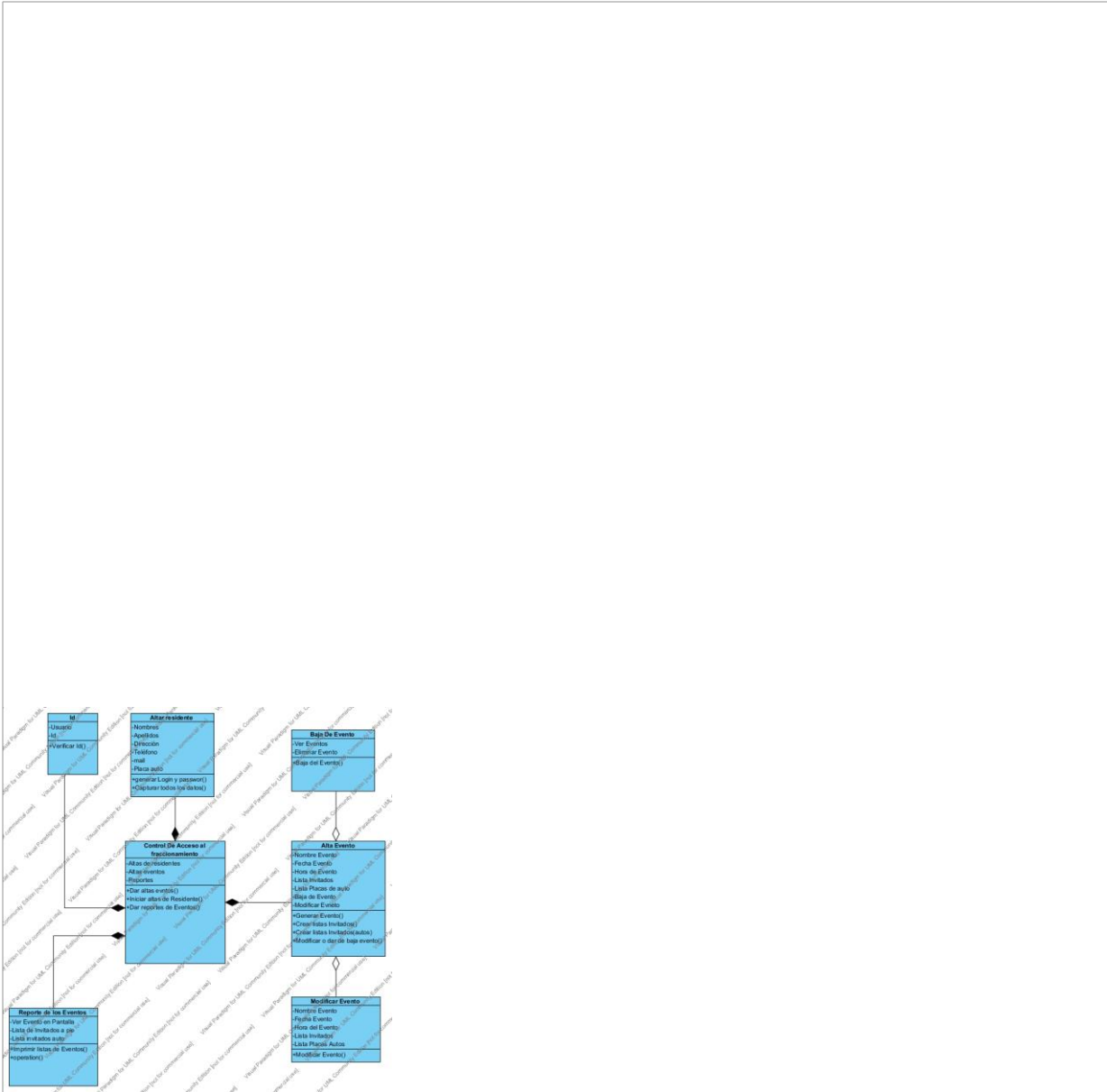


Figura 2.3 Diseño del diagrama de clases

2.7. DESARROLLO DE LOS DIAGRAMAS DE SECUENCIA

Cada uno de los diagramas de secuencia muestra el trabajo que realiza el sistema. En la figura 2.4 se muestra de forma general la actuación del administrador y los procesos que él puede realizar.

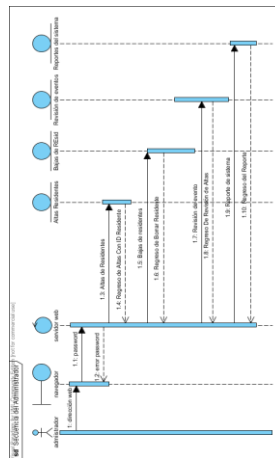


Figura 2.4 Diseño Diagrama de secuencia para el administrador. Caso de uso

Uno de los actores es el residente, es el único que puede dar de alta sus eventos, además de modificarlos o eliminarlos del sistema, figura 2.5

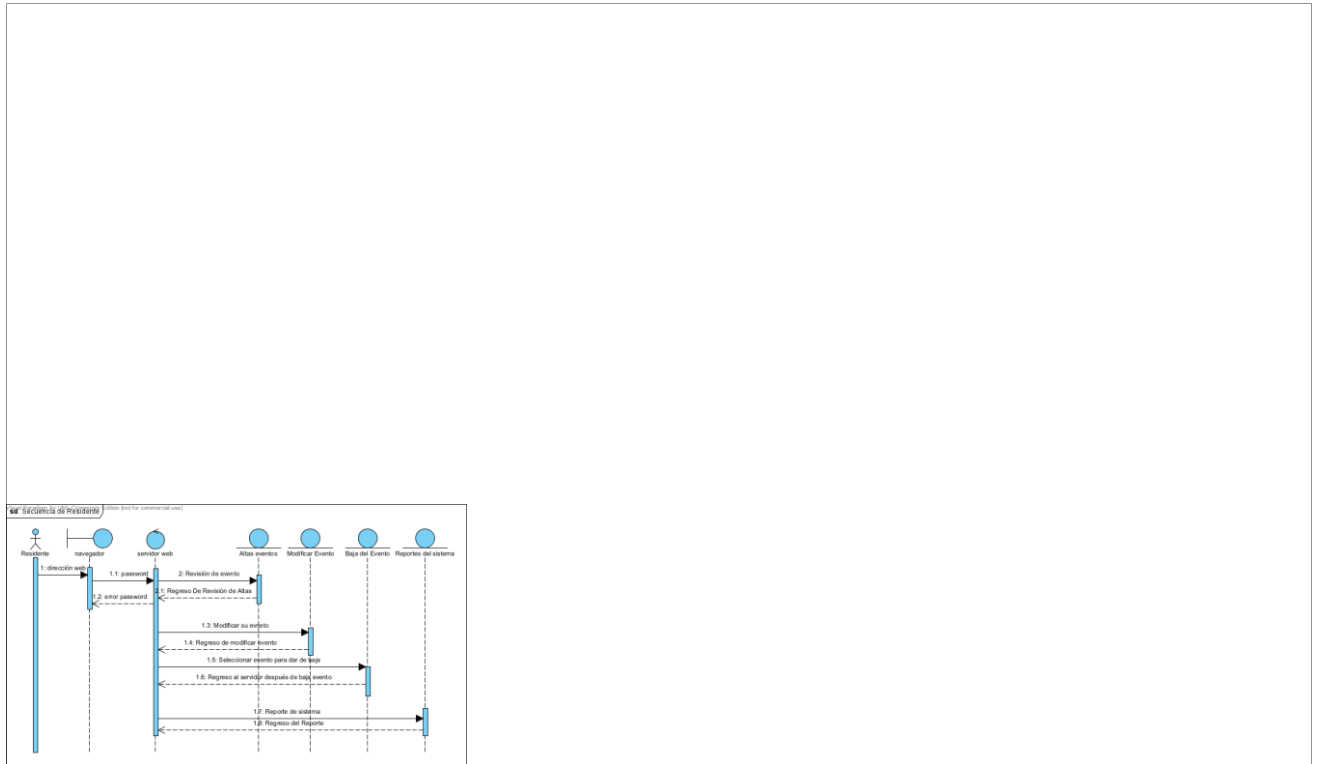


Figura 2.5 Diagrama de secuencia para el Residente.

En la figura 2.6 se muestra la actuación de la vigilancia la cuál puede obtener las listas de invitados ya sea a pie o la lista de los invitados en automóviles para los eventos posible del día.

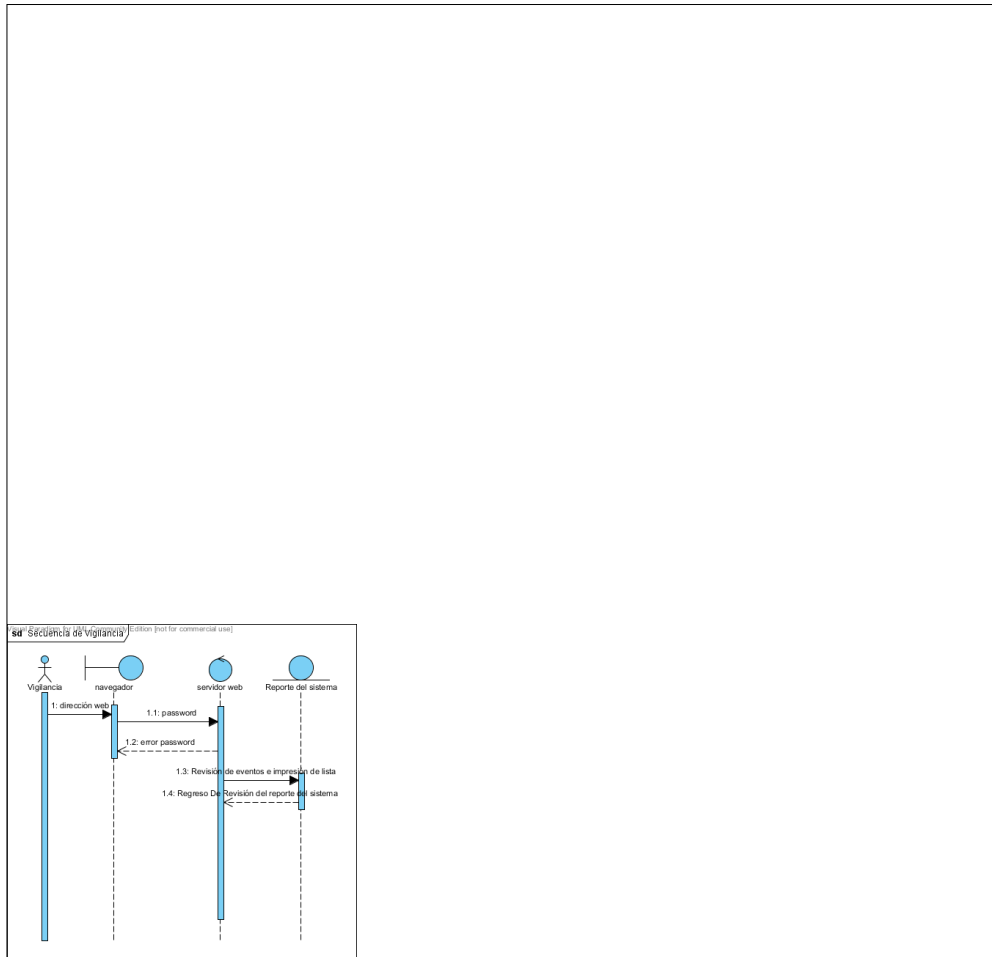


Figura 2.6 Diagrama de secuencia para la vigilancia.

2.8. DESARROLLO DE LA BASE DE DATOS

Una de las partes muy importante del sistema sin duda es la base de datos donde toda la información se pretende almacenar.

La modelación de datos es el proceso de crear una representación lógica de la estructura de una base de datos. La modelación es la tarea más importante en el desarrollo de las bases y sus aplicaciones [11].

Se debe contemplar como elementos claves de una E-R a entidades, atributos, identificadores y relaciones.

2.8.1 NORMALIZACIÓN

La normalización es una técnica para diseñar la estructura lógica de los datos de un sistema de información. Las ventajas de la normalización son evitar anomalías en inserción, modificación y borrado de datos, mejorar la independencia de datos y no establecer la restricción artificiales en la estructura de los datos.

Un concepto fundamental en la normalización es la dependencia funcional, que dice, una dependencia funcional es una relación entre atributos de una misma relación [12].

2.8.2 PRIMERA FORMA NORMAL(1FN)

Una relación está en primera forma normal si, y sólo si, todos los dominios de la misma contienen valores atómicos, es decir, no hay grupos repetitivos. Si se ve la relación gráficamente como una tabla, estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna.

Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos. En la primera, se repiten los atributos con un solo valor para cada valor del grupo repetitivo. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales. La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban. [12].

2.8.3 SEGUNDA FORMA NORMAL(2FN)

Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo no primo (que no está en la clave primaria) es completamente dependiente de la clave primaria.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen).

2.8.4 TERCERA FORMA NORMAL(3FN)

Una relación está en segunda forma normal si, y sólo si, está en 1FN y, además, cada atributo no primo (que no está en la clave primaria) es completamente dependiente de la clave primaria.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos que son funcionalmente dependientes y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen) [12].

Después de establecer la forma de normalizar una base de datos se implementan las tablas que darán origen a la base de datos de nuestro sistema que se muestran a continuación.

RESIDENTE

| | | | | | | | |
|------------|---------|---------|--------|-----------|-------|------|----------|
| <u>RFC</u> | PATERNO | MATERNO | NOMBRE | DIRECCIÓN | EMAIL | TIPO | PASSWORD |
|------------|---------|---------|--------|-----------|-------|------|----------|

EVENTO

| | | | | | | |
|------------|------------------|-----------|-------|---------|---------|--------|
| <u>RFC</u> | <u>NUMEVENTO</u> | NOMEVENTO | FECHA | HORAINI | HORAFIN | STATUS |
|------------|------------------|-----------|-------|---------|---------|--------|

INVITADO

| | | | | | |
|------------------|--------|-------|-------|-------|--------|
| <u>NOMEVENTO</u> | NOMBRE | MARCA | PLACA | COLOR | IMAGEN |
|------------------|--------|-------|-------|-------|--------|

TELÉFONO

| | | | |
|------------|------|-----|---------|
| <u>RFC</u> | CASA | CEL | TRABAJO |
|------------|------|-----|---------|

2.8.5 MODELO ENTIDAD-RELACIÓN

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los

atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido [12] la figura 2.7 se muestra la simbología par el modelo Entidad-Relación.

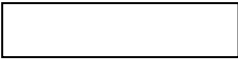
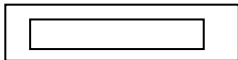
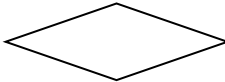

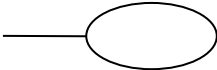

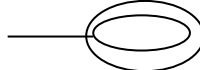

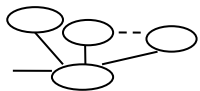
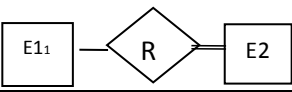
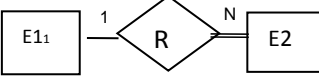
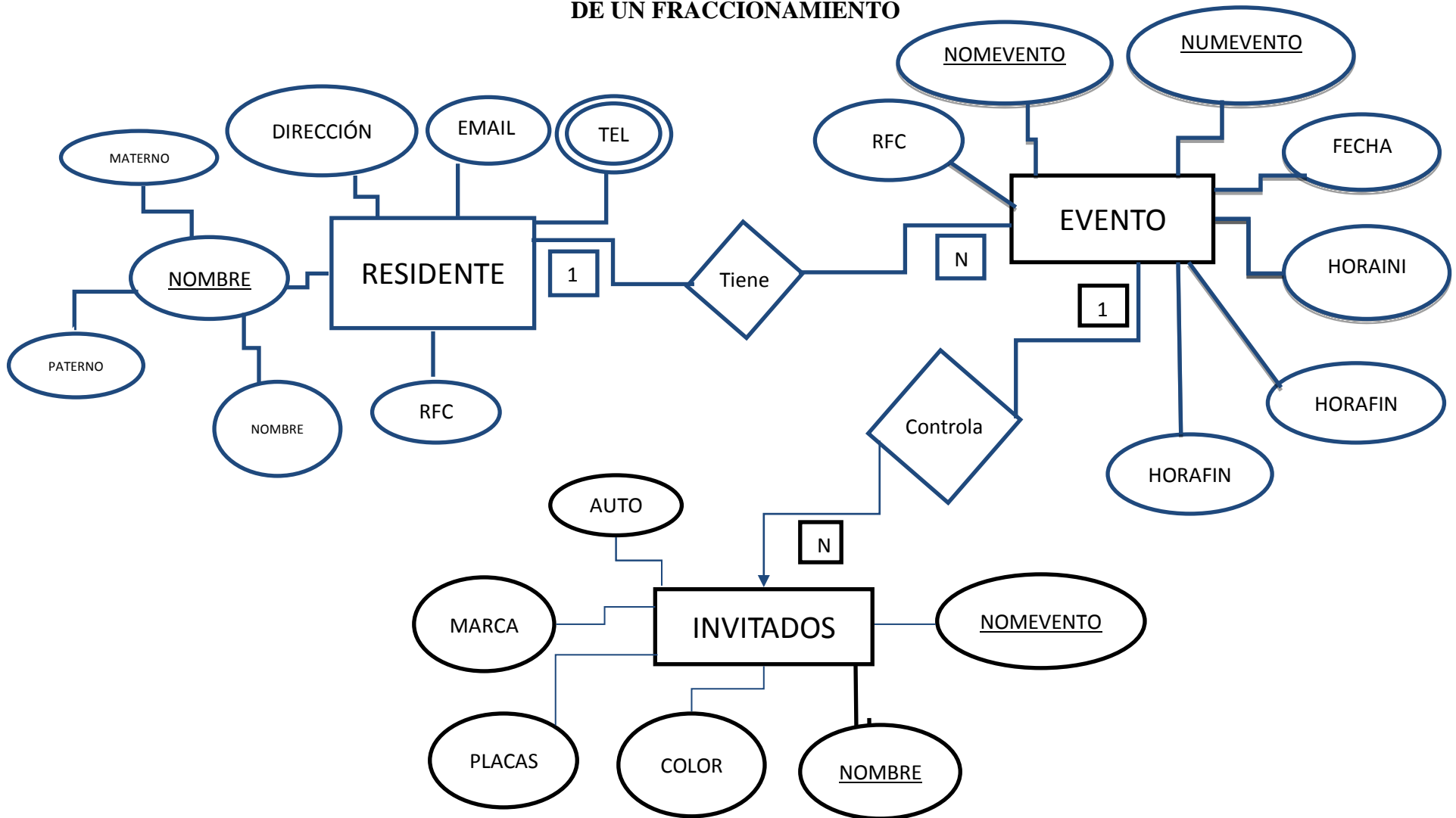
| | |
|---|--------------------------------|
|  | ENTIDAD |
|  | ENTIDAD DÉBIL |
|  | INTERRELACIÓN |
|  | INTERRELACIÓN DÉBIL |
|  | ATRIBUTO |
|  | ATRIBUTO LLAVE |
|  | ATRIBUTO MULTIVALUADO |
|  | ATRIBUTO DERIVADO |
|  | ATRIBUTO COMPUESTO |
|  | TOTAL PARTICIPACIÓN DE E2 EN R |
|  | RAZÓN DE CARDINALIDAD 1:N |

Figura 2.7 Simbología para diagrama E-R.

2.8.6 DIAGRAMA DE ENTIDAD-RELACIÓN

SISTEMA DE CONTROL DE ACCESO DE INVITADOS
DE UN FRACCIONAMIENTO



2.8.7 IMPLEMENTACIÓN DE LA BASE DE DATOS (CONTROL)

En esta sección se mostrará la base de datos (control) y cada una de las tablas que le componen, la implementación se hizo con wampserver.

La figura 2.8 Muestra la Base de Datos (control) junto con cada una de sus tablas.

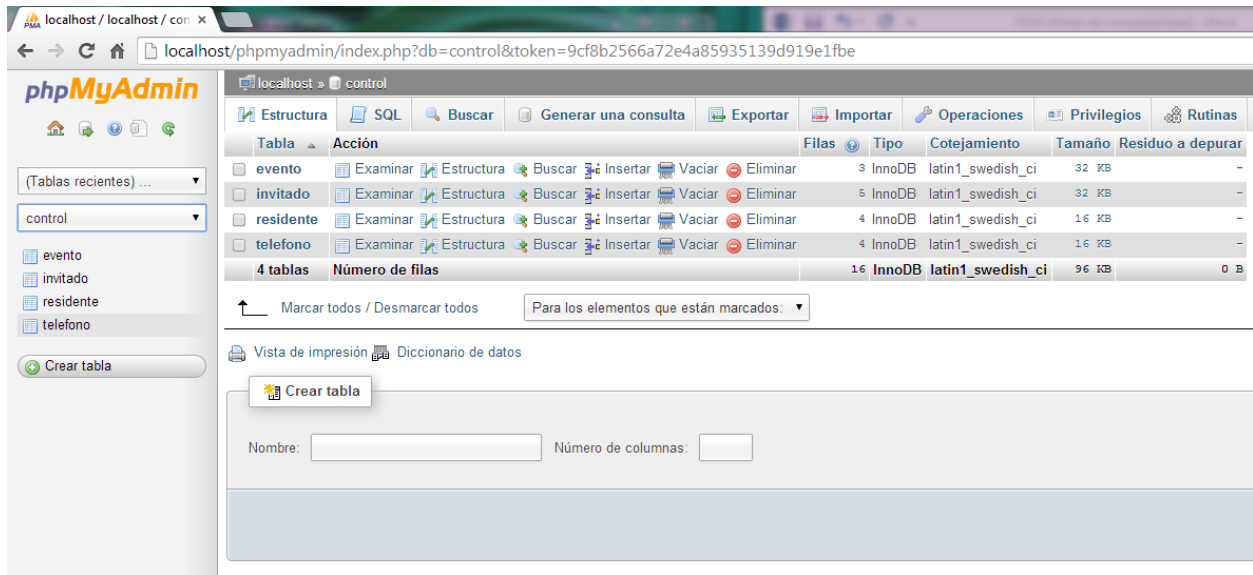


Figura 2.8 Base de Datos del Sistema (control).

La figura 2.9 Muestra la tabla evento con sus atributos junto con la estructura.

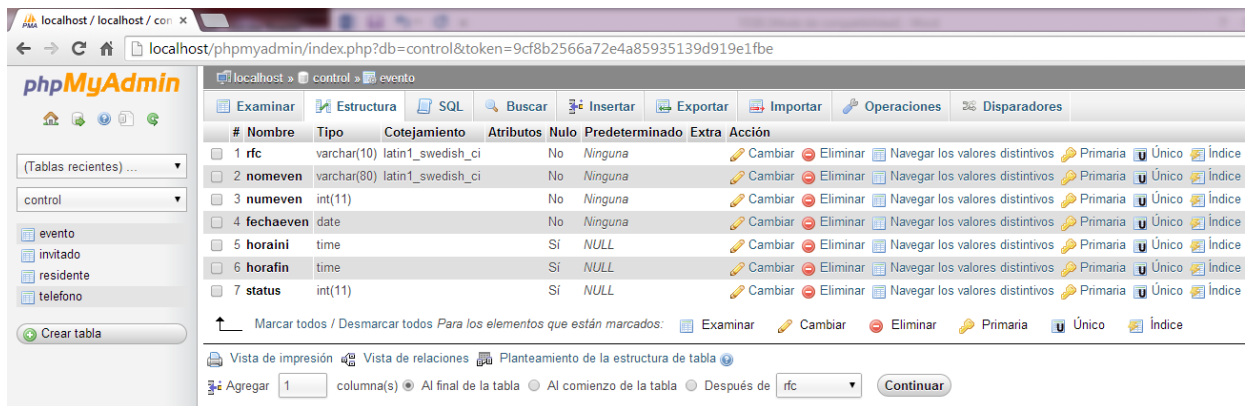


Figura 2.9 Tabla Evento del Sistema (control).

La figura 2.10 Muestra la tabla invitado con atributos y estructura.

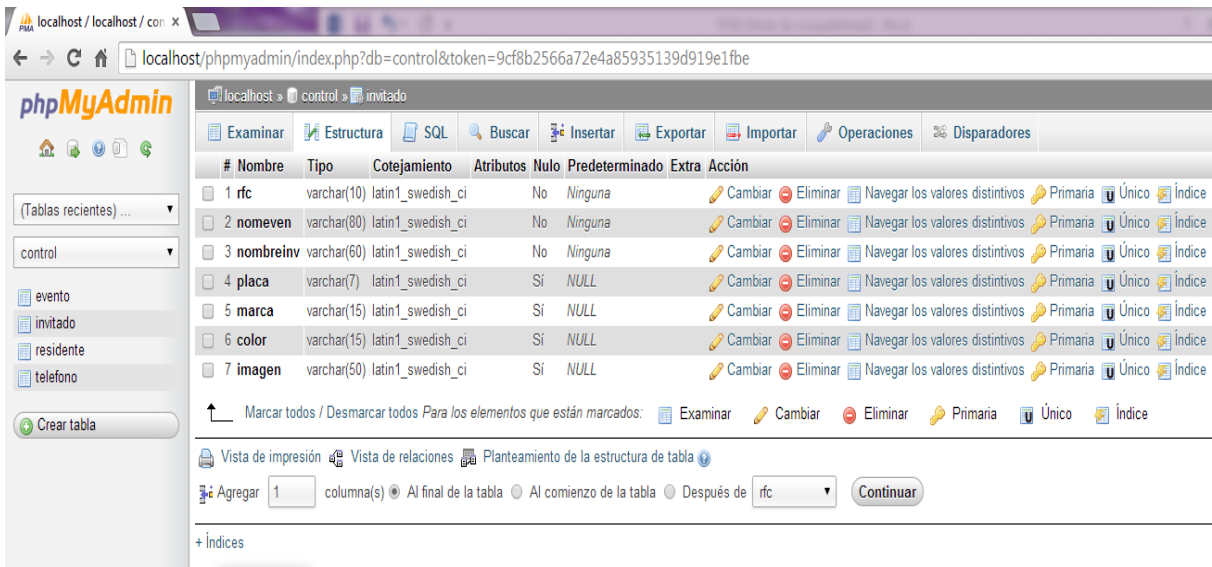


Figura 2.10 Tabla Invitado del Sistema (control).

La figura 2.11 Muestra la tabla residente con atributos y estructura.

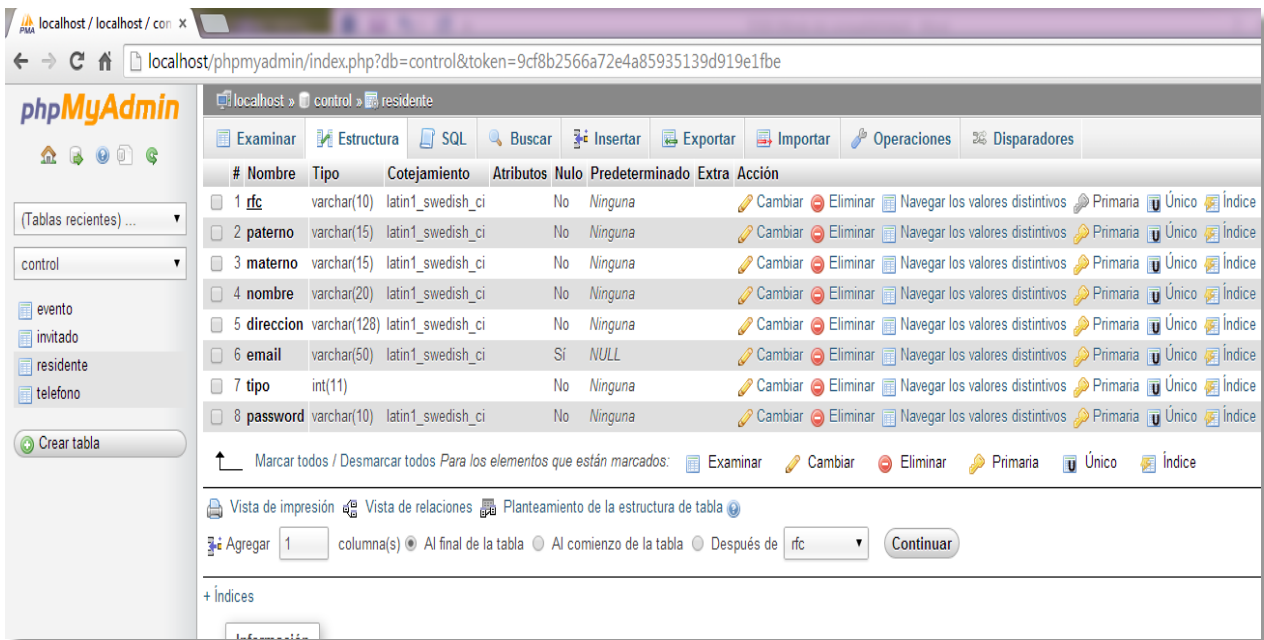


Figura 2.11 Tabla Residente del Sistema (control).

Finalmente en la figura 2.12 se muestra la tabla teléfono con sus atributos y estructura.

localhost / localhost / con x

localhost/phpmyadmin/index.php?db=control&token=9cf8b2566a72e4a85935139d919e1f8e

phpMyAdmin

control » telefono

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Disparadores

| # | Nombre | Tipo | Cotejamiento | Atributos | Nulo | Predeterminado | Extra | Acción |
|--------------------------|-----------|-------------|-------------------|-----------|------|----------------|-------|---|
| <input type="checkbox"/> | 1 rfc | varchar(10) | latin1_swedish_ci | | No | Ninguna | | Cambiar Eliminar Navegar los valores distintivos P U Índice |
| <input type="checkbox"/> | 2 casa | varchar(10) | latin1_swedish_ci | | Sí | NULL | | Cambiar Eliminar Navegar los valores distintivos P U Índice |
| <input type="checkbox"/> | 3 cel | varchar(10) | latin1_swedish_ci | | Sí | NULL | | Cambiar Eliminar Navegar los valores distintivos P U Índice |
| <input type="checkbox"/> | 4 trabajo | varchar(10) | latin1_swedish_ci | | Sí | NULL | | Cambiar Eliminar Navegar los valores distintivos P U Índice |

↑ Marcar todos / Desmarcar todos Para los elementos que están marcados: Examinar Cambiar Eliminar **P** **U** Índice

Vista de impresión Vista de relaciones Planteamiento de la estructura de tabla

Agregar 1 columna(s) Al final de la tabla Al comienzo de la tabla Después de rfc Continuar

+ Índices

Figura 2.12 Tabla Teléfono del Sistema (control).

CAPÍTULO III IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

Hoy día se debe establecer una infraestructura de procesamiento de la información, que cuente con los elementos requeridos para proveer información adecuada, exacta y oportuna, se requiere de un modelo que reúna todas las características necesarias para ésta infraestructura, independientemente del tamaño y complejidad de las operaciones.

Un elemento básico e imprescindible es un modelo Cliente/Servidor, ya que es una de las herramientas más utilizadas, importante y poderosa para el almacenamiento de información en una base de datos como servidor.

En este capítulo se mostrará la implementación del sistema, así como las pruebas del funcionamiento, tomando como base el diseño que se obtuvo al crear la base de datos, además se mencionaran las herramientas que se utilizaron para la implementación de dicho sistema.

3.1. LENGUAJE HTML

Para la implementación de nuestro sistema utilizamos el lenguaje HTML y nos apoyamos para la edición en el programa DreamWeaver 8.

3.2. WAMPSEVER

WampServer es un entorno de desarrollo web para Windows, que permite crear aplicaciones web con Apache, PHP y una base de datos MySQL. Al lado, PhpMyAdmin que permite administrar fácilmente las bases de datos.

3.2.1 APACHE

Se estableció como punto de publicación un servidor Web Apache que nos brinda un servicio potente, flexible y que además nos permite optimizar el recurso del servidor y obtener mejores resultados en los tiempos de procesamientos. Permite la integración de forma natural con el lenguaje PHP.

3.2.2 MySQL Y PhpMyAdmin

Para el manejo de la base de datos se usó MySQL y PhpMyAdmin, ya que es una forma muy eficaz de administrar bases de datos.

3.2.3 PHP

PHP es un lenguaje de programación de código abierto, sencillo de aprender, PHP es un lenguaje interpretado, no compilado, que permite hacer cambios rápidos y con una sintaxis suave. PHP se programa línea por línea y de arriba hacia abajo y aunque PHP no es un lenguaje orientado a objetos implementa las características que permiten definir las clases.

3.3. HOJAS DE ESTILO

Las hojas de estilo representan un avance importante para los diseñadores de páginas web, al darles un mayor rango de posibilidades para mejorar la apariencia de las páginas.

Para este sistema se utilizaron hojas de estilo de “Free CSS Templates”[13].

3.4. PHOTOSHOP E ILLUSTRATOR

Para el diseño de grafico e imágenes que se utilizaron dentro del sistema la mejor forma de hacerlo fue utilizando Photoshop e Illustrator.

photoshop es uno de los programas más utilizados en el ambiente gráfico, en el área fotográfica, ilustradores, artistas plásticos, etc. Este programas pertenece al grupo de edición de imágenes, también llamado programa de mapas bits, que tiene un entorno basado en píxeles.

Illustrator es el nombre oficial que tiene uno de los programas más populares de la familia Adobe, y que se trata esencialmente de una aplicación de creación y manipulación vectorial en forma de taller de arte que trabaja sobre un área de dibujo, que también se le conoce como mesa de trabajo. El trabajo principal que se puede realizar es la creación artística de dibujos y pinturas para Ilustración.

3.5. IMPLEMENTACIÓN DEL SISTEMA

Finalmente aquí se mostrara el trabajo que realiza el sistema y se mostrará la interacción de cada uno de los actores y las actividades que pueden desempeñar cada uno de ellos.

3.5.1. PAGINA PRINCIPAL

Al ingresar al sistema se puede hacer desde cualquier navegador y lo primero que se presenta en esta interfaz es la página de inicio como se muestra en la figura 3.1 En este sistema lo primero que se pide es que los actores se identifiquen al introducir el nombre del usuario y contraseña (logueo), hay que mencionar que el nombre del usuario es su registro federal de contribuyente.

Posteriormente se muestra como medida de seguridad (figura 3.2) un recaptcha para tratar de evitar violaciones al sistema, posteriormente y dependiendo del actor es sistema lo ingresa al área que le corresponde.

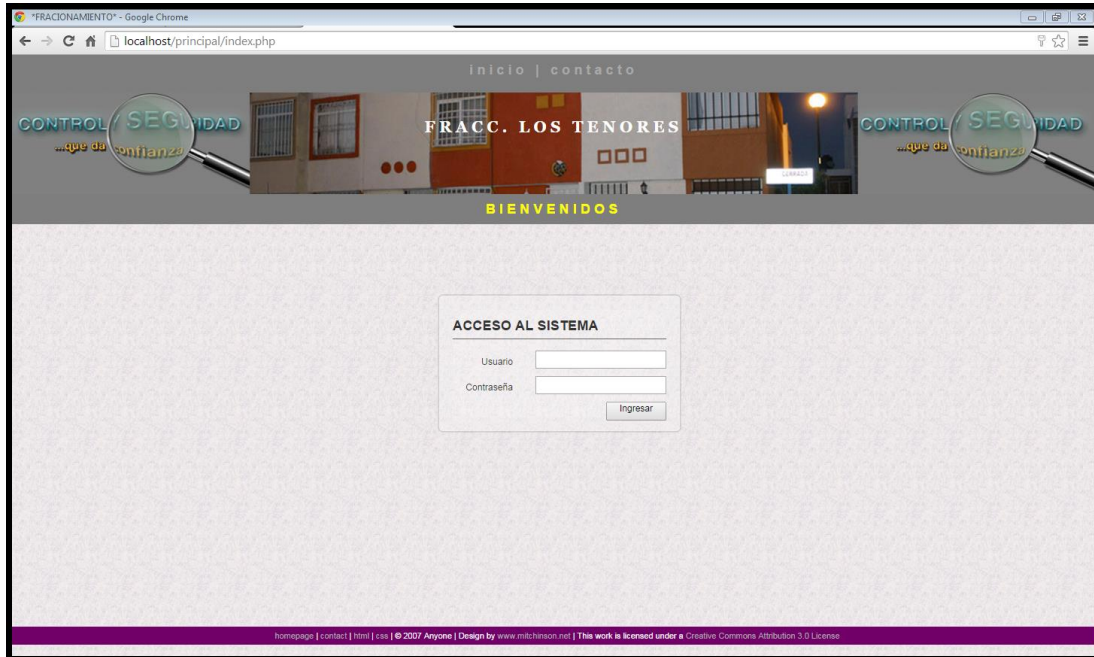


Figura 3.1 Página principal del Sistema (control).

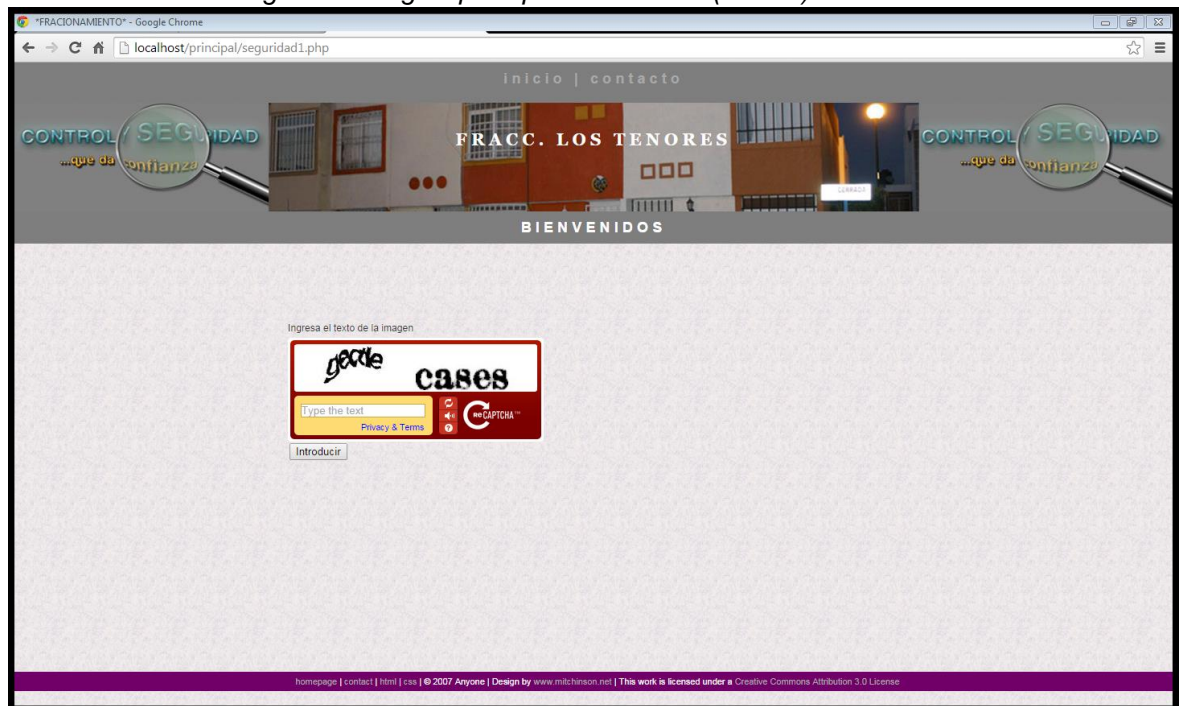


Figura 3.2 Página de seguridad de Sistema (recaptcha).

3.5.2. ADMINISTRADOR

En esta ventana (figura3.3) se le da la bienvenida al administrador donde él puede consultar a los usuarios, generar las cuentas de los residentes y vigilantes, eliminar cuentas, modificar cuentas, ver eventos del fraccionamiento, Imprimir listas de invitados etc.

Además en la ventana de bienvenida existe información correspondiente al uso del sistema, es decir, la explicación que le ayudará a entender y mantener el sistema de control del fraccionamiento y la actuación que cada actor que utiliza el programa tiene en el mismo.

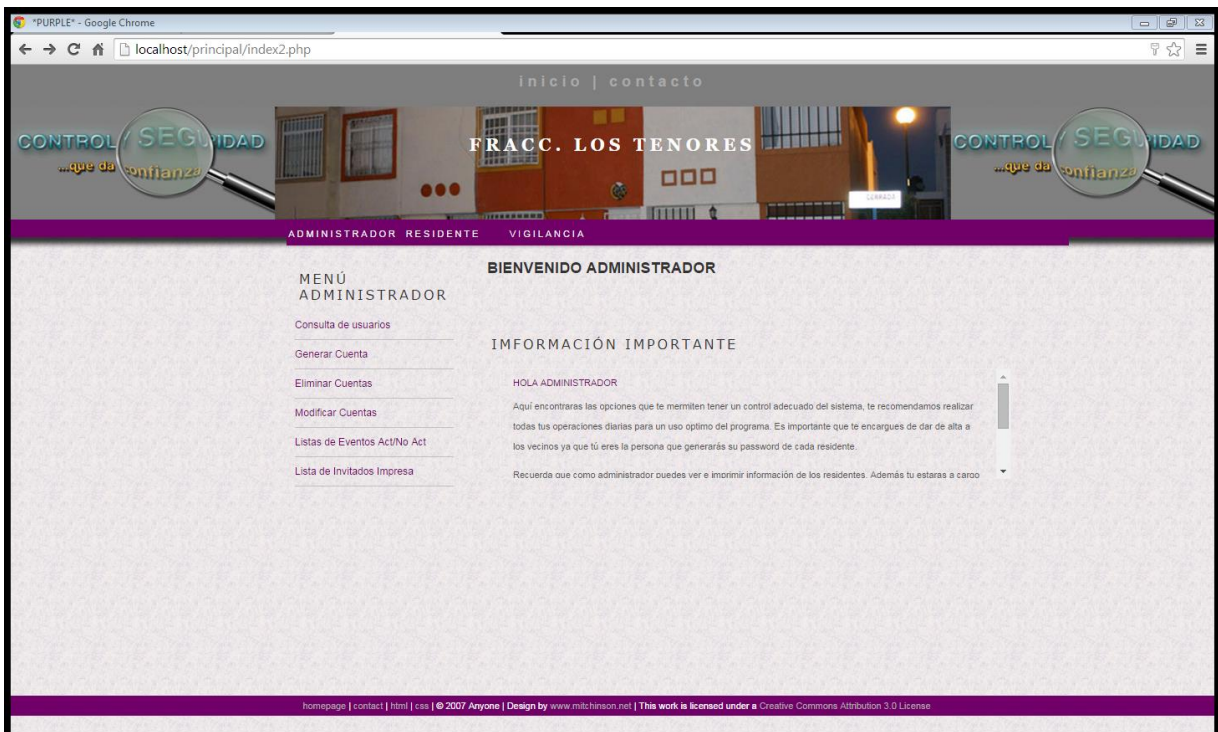


Figura 3.3 Página del administrador del Sistema.

3.5.3. CONSULTA DE USUARIOS

En esta sección de consulta (figura 3.4) el administrador puede visualizar a todos los usuarios del sistema que incluye a los residentes y vigilancia.

The screenshot shows a web interface for 'FRACC. LOS TENORES'. At the top, there are links for 'inicio | contacto'. Below this is a header with the text 'FRACC. LOS TENORES' and a navigation bar with 'ADMINISTRADOR', 'RESIDENTE', and 'VIGILANCIA'. The main content area is titled 'CONSULTAR LA BASE DE DATOS DEL FRACCIONAMIENTO'. On the left, there is a 'MENÚ ADMINISTRADOR' with options: 'Consulta de usuarios', 'Generar Cuenta', 'Eliminar Cuentas', 'Modificar Cuentas', 'Listas de Eventos Act/No Act', and 'Lista de Invitados Impresa'. The main area displays 'MOSTRANDO INFORMACIÓN' followed by a table of user data.

| RFC | PATERNO | MATERNO | NOMBRES | DIRECCIÓN |
|------------|-----------|----------|-------------|---|
| robj030810 | Gonzalez | Vivaldo | Elias | Fray Juan de rivas # 3222 col. 3 cruces |
| robj030816 | RODRIGUEZ | ROSAS | OSCAR EFREN | JOSÉ CARRERAS #22 LOS TENORES |
| robj640829 | RODRIGUEZ | BERMUDEZ | JUAN OSCAR | JOSÉ CARRERAS #22 LOS TENORES |

Figura 3.4 Página de consultas de Sistema.

3.5.4. GENERAR CUENTA

En esta ventana (figura 3.5) el administrador se encarga de dar de alta a los residentes y los elementos de vigilancia asignados al fraccionamiento.

La información que se capturan son datos personales como nombre dirección, teléfonos, etc. Además el administrador establece el tipo de usuario y su contraseña para el ingreso al sistema, recordar que sólo él puede dar de alta o baja a los residentes y vigilantes.

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ ADMINISTRADOR

- Consulta de usuarios
- Generar Cuenta
- Eliminar Cuentas
- Modificar Cuentas
- Listas de Eventos Act/No Act
- Lista de Invitados Impresa

GENERAR LA ALTA DE LOS REDIDENTES Y VIGILANTES

RFC:

Paterno:

Materno:

Nombres:

Dirección:

Tel. Casa:

Tel. Cel:

Tel. Trabajo:

Email:

Tipo de usuario:

Password:

Figura 3.5 Página para generar cuentas del Sistema.

3.5.5. ELIMINAR CUENTA

El administrador en esta parte del sistema (figura 3.6) se encarga de dar de baja a los residentes o elementos de vigilancia, permitiendo controlar a los residen que se mantienen al corriente en sus cuotas dentro del fraccionamiento y eliminar a los vigilantes que no laboren más en el fraccionamiento.

En la figura 3.7 se muestra la ventana que confirme la eliminación del usuario del sistema

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ ADMINISTRADOR

- Consulta de usuarios
- Generar Cuenta
- Eliminar Cuentas
- Modificar Cuentas
- Listas de Eventos Act/No Act
- Lista de Invitados Impresa

DEPURAR LA BASE DE DATOS

ELIMINACIÓN DEL RESIDENTE DE LA BASE DE DATOS

| RFC | PATERNO | MATERNO | NOMBRE | ELIMINAR |
|------------|-----------|----------|-------------|----------|
| robj030810 | Gonzalez | Vivaldo | Elias | X |
| robj030816 | RODRIGUEZ | ROSAS | OSCAR EFREN | X |
| robj640829 | RODRIGUEZ | BERMUDEZ | JUAN OSCAR | X |

Figura 3.6 Página que muestra residentes a dar de baja.



Figura 3.7 Página que confirma la baja del residente.

3.5.6. MODIFICAR CUENTA

En caso que el administrador requiera actualizar o modificar algún dato de cualquier residente él podrá en cualquier momento modificar una cuenta, en la figura 3.8 muestra a los residentes y vigilantes que se pueden modificar.

Sólo se debe seleccionar el elemento y el sistema mostrará los datos del residente que podrán ser modificados, y aceptar la actualización (figura 3.9).



Figura 3.8 Página que muestra residentes a modificar.



Figura 3.9 Página que muestra datos del residente a modificar.

3.5.7. MOSTRAR INVITADOS

El administrador podrá visualizar en pantalla la lista de invitados de cualquier evento registrado, el sistema le mostrará una ventana (figura 3.10) donde seleccionará el evento que mostrará la lista de invitados. En la figura 3.11 se muestra la lista de invitados del evento seleccionado que no se puede modificar ya que es uso exclusivo del residente.

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ ADMINISTRADOR

- Consulta de usuarios
- Generar Cuenta
- Eliminar Cuentas
- Modificar Cuentas
- Listas de Eventos Act/No Act
- Lista de Invitados Impresa

LISTA DE INVITADOS POR EVENTO EN PANTALLA

SELECCIONE EL EVENTO PARA MOSTRAR

LISTA DE INVITADOS

| RFC | NOMBRE EVENTO | FECHA EVENTO | HORA EVENTO | MOSTRAR LISTA |
|------------|-------------------|--------------|-------------|---------------|
| robj030816 | BAUTIZO | 2014-10-25 | 17:00:00 | ✓ |
| robj030816 | BODA PATY Y OSCAR | 2015-09-18 | 17:30:00 | ✓ |
| robj640829 | BAUTIZO OSCAR | 2015-08-16 | 17:30:00 | ✓ |

Figura 3.10 Página que muestra los eventos.



Figura 3.11 Página que muestra a los invitados del eventos.

3.5.8. IMPRESIÓN DE LISTAS DE INVITADOS (ADMINISTRADOR)

En esta parte del menú el administrador tendrá acceso a las listas de todos los eventos del fraccionamiento (figura 3.12) con la intención de solventar cualquier problema que se pudiera suscitar en la caseta de vigilancia, el sistema muestra los eventos y posteriormente la lista de invitados (figura 3.13) previo a la impresión.



Figura 3.12 Página que muestra los eventos a imprimir.

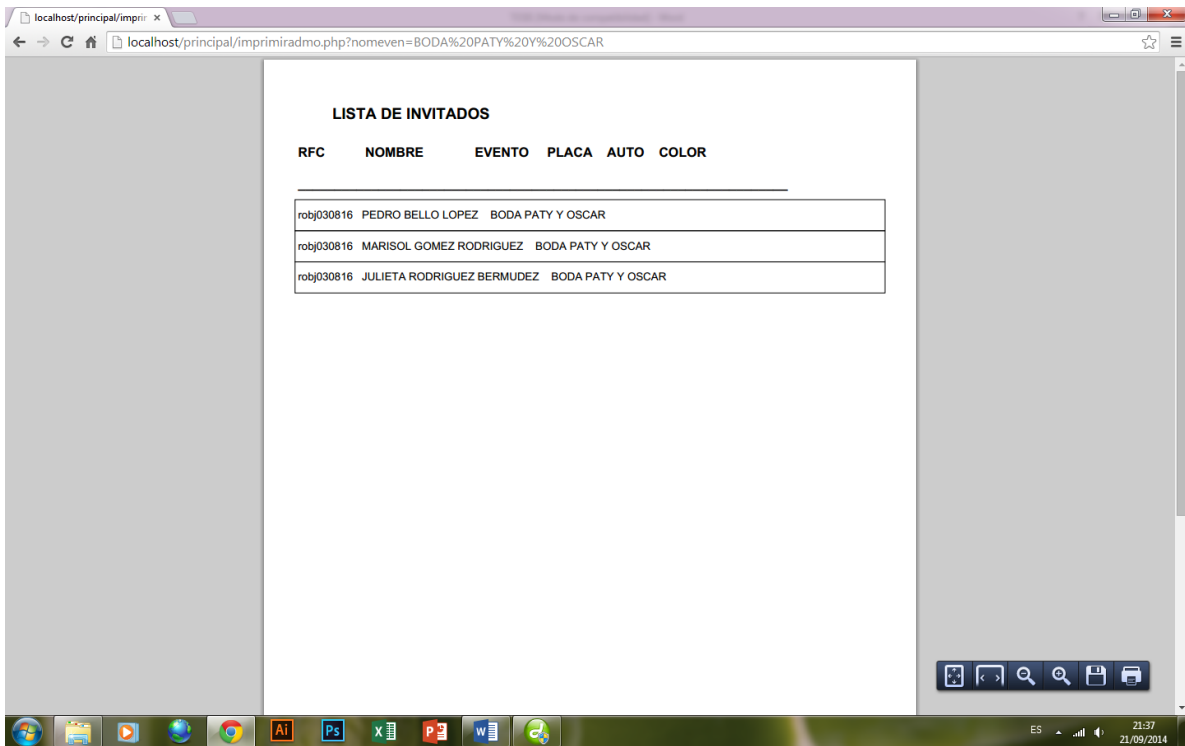


Figura 3.12 Página que muestra la impresión de invitados.

3.5.9. RESIDENTES

El residente después de loguearse como cualquiera de los tres actores, el sistema le muestra la ventana asignada a estos usuarios (figura 3.13), donde se muestra el menú con las opciones correspondiente. El residente podrá dar de alta su evento, modificarlo, darlo de baja, dar de alta sus invitados por evento e imprimir sus listas.

En esta ventana principal también se muestran las indicaciones generales del uso del sistema.

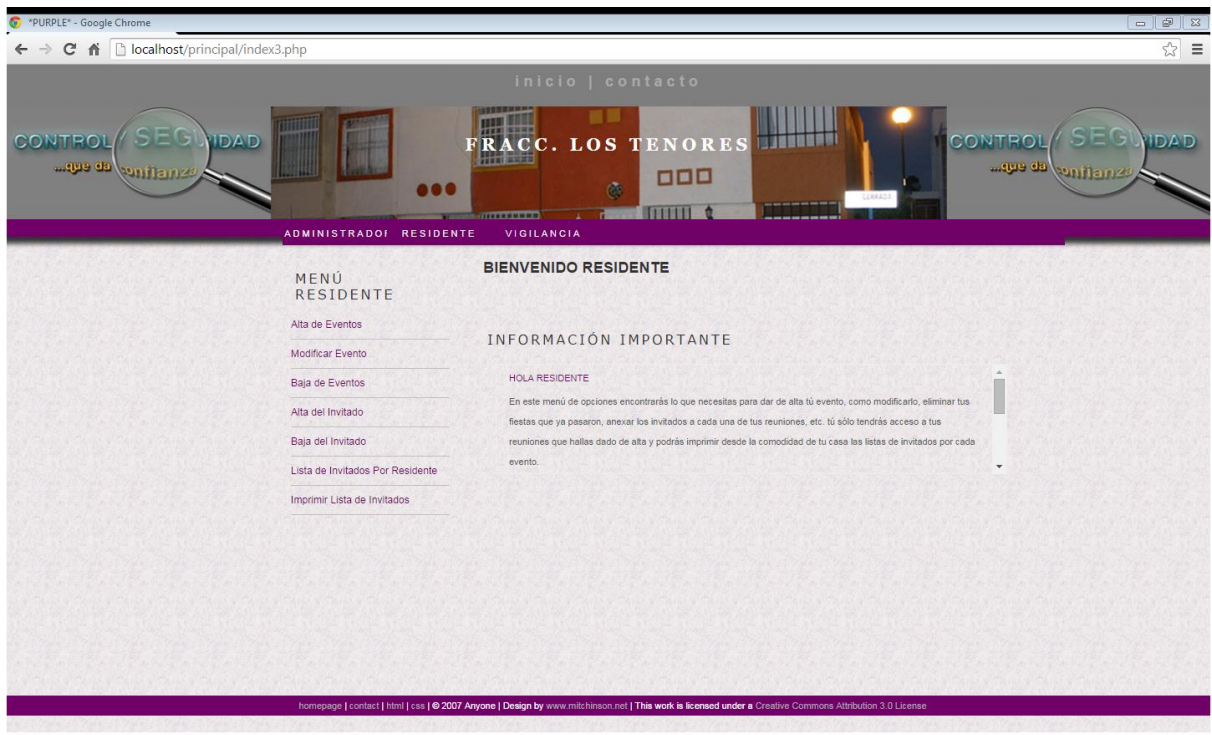


Figura 3.13 Página principal del residente.

3.5.10. ALTA DE EVENTO

El residente tendrá que identificar su evento, en la siguiente ventana (figura 3.14) se muestra los datos que identifican a su evento como son su rfc, nombre del evento, fecha del evento, hora de inicio, etc.

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ RESIDENTE

- Alta de Eventos
- Modificar Evento
- Baja de Eventos
- Alta del Invitado
- Baja del Invitado
- Lista de Invitados Por Residente
- Imprimir Lista de Invitados

ALTA DE SU EVENTOS

SU RFC:

Nombre Evento:

Fecha Evento:

Hora Inicio:

Hora Fin:

Estado:

Guardar

Figura 3.14 Página para dar de alta los eventos.

3.5.11. MODIFICAR EVENTO

El residente puede tener más de un evento disponible para realizar por lo que al seleccionar la opción “modificar evento” el sistema le muestra en ventana

(figura 3.15) los eventos dados de alta. Al seleccionar uno de los eventos se mostrara la ventana (figura 3.16) donde se visualizan los datos del evento seleccionado para su respectiva modificación.

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ RESIDENTE

- Alta de Eventos
- Modificar Evento
- Baja de Eventos
- Alta del Invitado
- Baja del Invitado
- Lista de Invitados Por Residente
- Imprimir Lista de Invitados

SELECCIÓN DEL EVENTO

ACTUALIZAR LOS DATOS DEL EVENTO

| RFC | NOMBRE EVENTO | FECHA EVENTO | HORA EVENTO | ACTUALIZAR |
|------------|-------------------|--------------|-------------|------------|
| robj030816 | BAUTIZO | 2014-10-25 | 17:00:00 | |
| robj030816 | BODA PATY Y OSCAR | 2015-09-18 | 17:30:00 | |

Figura 3.15 Página de selección de evento.



Figura 3.16 Página para modificar evento.

3.5.12. ELIMINAR EVENTO

En esta parte del menú de los residentes se da de baja a los eventos ya existentes, al seleccionar la opción “baja de evento” se muestra la ventana (figura 3.17) que lista los eventos dados de alta por el residente. Después de la selección se debe confirmar la baja del evento para eliminar de forma automática el evento como se muestra en la siguiente ventana (figura 3.18).



Figura 3.17 Página de selección de evento para dar de baja.



Figura 3.18 Página que muestra el evento que fue eliminado.

3.5.13. ALTA DE INVITADOS

Cuando el residente ha dado de alta su evento (figura 3.19), ahora puede ingresar todos los invitados. El residente primero registrara su rfc, seguido del nombre del evento, posteriormente dará nombre del invitado, placa de su auto, marca del auto y el color.

El nombre del evento lo coloca al invitado en la fiesta marcada con fecha y hora como se muestra (figura 3.20).

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ RESIDENTE

- Alta de Eventos
- Modificar Evento
- Baja de Eventos
- Alta del Invitado
- Baja del Invitado
- Lista de Invitados Por Residente
- Imprimir Lista de Invitados

GENERAR LA LISTA DE LOS INVITADOS DE TU EVENTO

TÚ RFC: robj030816

NOMBRE EVENTO: BODA PATY Y OSCAR

NOMBRE INVITADO: CARLOS RODRIGUEZ

PLACA AUTO: TXT2340

MARCA AUTO: JEEP

COLOR: AZUL

Guardar

Figura 3.19 Página de alta del invitado.



Figura 3.20 Página donde se integra al invitado al evento.

3.5.14. BAJA DE INVITADOS

El residente puede eliminar a sus invitados del evento al que fue asignado, el sistema mostrará la lista de eventos (figura 3.21) donde seleccionará el evento para eliminar a la persona. En la figura 3.22 se muestra la lista de invitados y se selecciona al invitado a dar de baja previa confirmación.

Final mente se muestra la lista de invitados (figura 3.23) indicando la ausencia de la persona dada de baja.

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ RESIDENTE

- Alta de Eventos
- Modificar Evento
- Baja de Eventos
- Alta del Invitado
- Baja del Invitado
- Lista de Invitados Por Residente
- Imprimir Lista de Invitados

LISTA DE INVITADOS POR EVENTO

SELECCIONE EL EVENTO PARA MOSTRAR LISTA DE INVITADOS Y DAR DE BAJA

| RFC | NOMBRE EVENTO | FECHA EVENTO | HORA EVENTO | MOSTRAR LISTA |
|------------|-------------------|--------------|-------------|-------------------------------------|
| robj030816 | BODA PATY Y OSCAR | 2015-09-18 | 17:30:00 | <input checked="" type="checkbox"/> |
| robj030816 | BAUTIZO OSCAR | 2015-08-16 | 10:00:00 | <input checked="" type="checkbox"/> |

Figura 3.21 Página que muestra los eventos.

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ RESIDENTE

- Alta de Eventos
- Modificar Evento
- Baja de Eventos
- Alta del Invitado
- Baja del Invitado
- Lista de Invitados Por Residente
- Imprimir Lista de Invitados

MOSTRANDO INVITADOS A ELIMINAR

ELIMINACIÓN DEL INVITADO DEL EVENTO

| | | | |
|------------|----------------------------|-------------------|-------------------------------------|
| robj030816 | MARISOL GOMEZ RODRIGUEZ | BODA PATY Y OSCAR | <input checked="" type="checkbox"/> |
| robj030816 | JULIETA RODRIGUEZ BERMUDEZ | BODA PATY Y OSCAR | <input checked="" type="checkbox"/> |
| robj030816 | CARLOS RODRIGUEZ | BODA PATY Y OSCAR | <input checked="" type="checkbox"/> |
| robj030816 | ALFREDO PEREZ | BODA PATY Y OSCAR | <input checked="" type="checkbox"/> |

Figura 3.22 Página que muestra a los invitado a dar de baja del evento.



Figura 3.23 Página donde se muestra la baja.

3.5.15. LISTA DE INVITADOS POR EVENTO

En esta opción del menú del residente (figura 3.24) se muestra la lista de los eventos, se selecciona uno de ellos y en la figura 3.25 se muestra la lista de los invitados del evento.



Figura 3.24 Página donde se muestran y los eventos



Figura 3.25 Página donde se muestran la lista de los invitados.

3.5.16. IMPRIMIR LISTAS DE INVITADOS

Cada residente puede imprimir sus listas de invitados previa selección de su evento. En la figura 3.26 se muestra la lista de sus reuniones sociales para seleccionar el que desea imprimir.

El sistema le muestra en pantalla (figura3.27) la lista de sus invitados previo a la impresión.

inicio | contacto

FRACC. LOS TENORES

ADMINISTRADOR RESIDENTE VIGILANCIA

MENÚ RESIDENTE

- Alta de Eventos
- Modificar Evento
- Baja de Eventos
- Alta del Invitado
- Baja del Invitado
- Lista de Invitados Por Residente
- Imprimir Lista de Invitados

SELECCIONE EL EVENTO PARA MOSTRAR LISTA DE INVITADOS

| RFC | NOMBRE EVENTO | FECHA EVENTO | HORA EVENTO | IMPRIMIR LISTA |
|------------|-------------------|--------------|-------------|----------------|
| robj030816 | BODA PATY Y OSCAR | 2015-09-18 | 17:30:00 | |
| robj030816 | BAUTIZO OSCAR | 2015-08-16 | 10:00:00 | |

Figura 3.26 Página donde se muestran la lista de los eventos a imprimir.

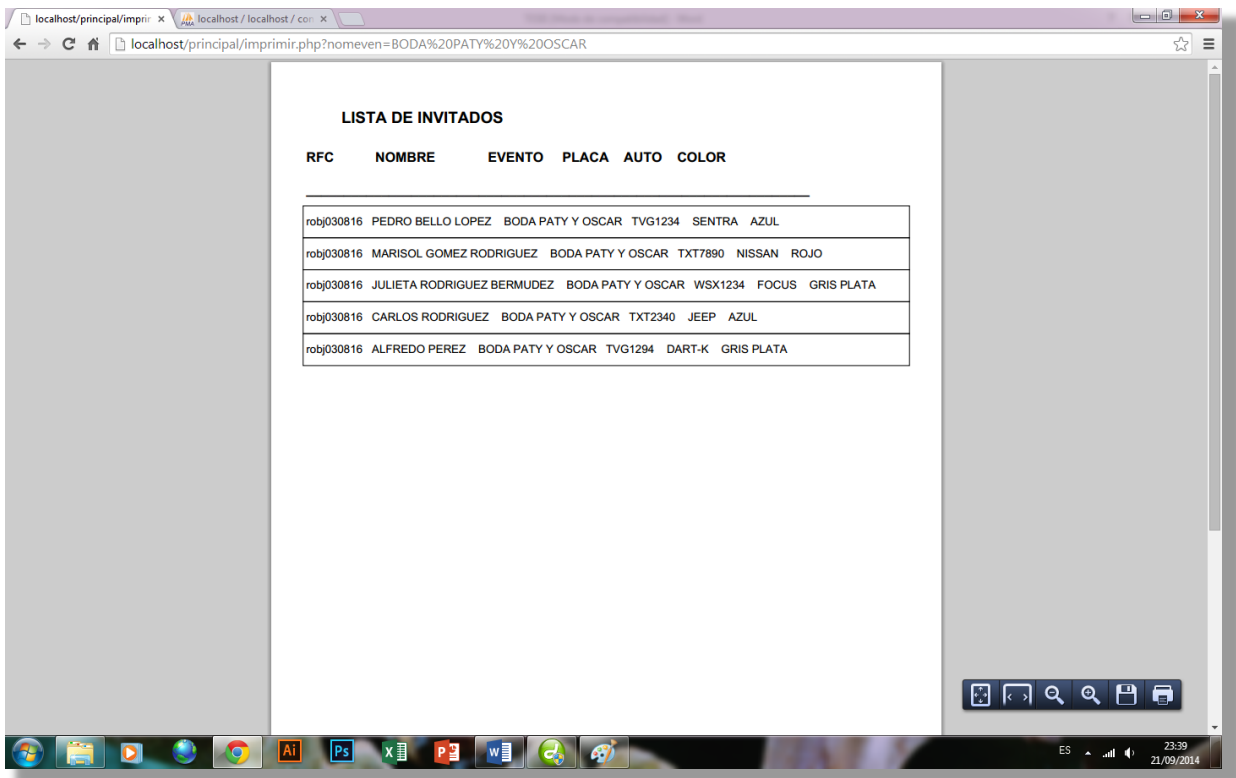


Figura 3.27 Página que muestran la lista de invitados a imprimir.

3.5.17. VIGILANCIA

En la última parte del sistema que se refiere a la vigilancia se muestra el siguiente menú (figura 3.28), la actuación de la vigilancia sólo se limita a poder consultar la información de los evento que están dados de alta.

El vigilante en turno puede ver la información en pantalla del evento seleccionado y en caso de ser necesario podrá imprimir las listas de la reunión social asignada a la fecha marcada.

En esta misma página se muestra información que le ayudará a los elementos de seguridad a utilizar el sistema y agilizar la entrada de los invitados al fraccionamiento.

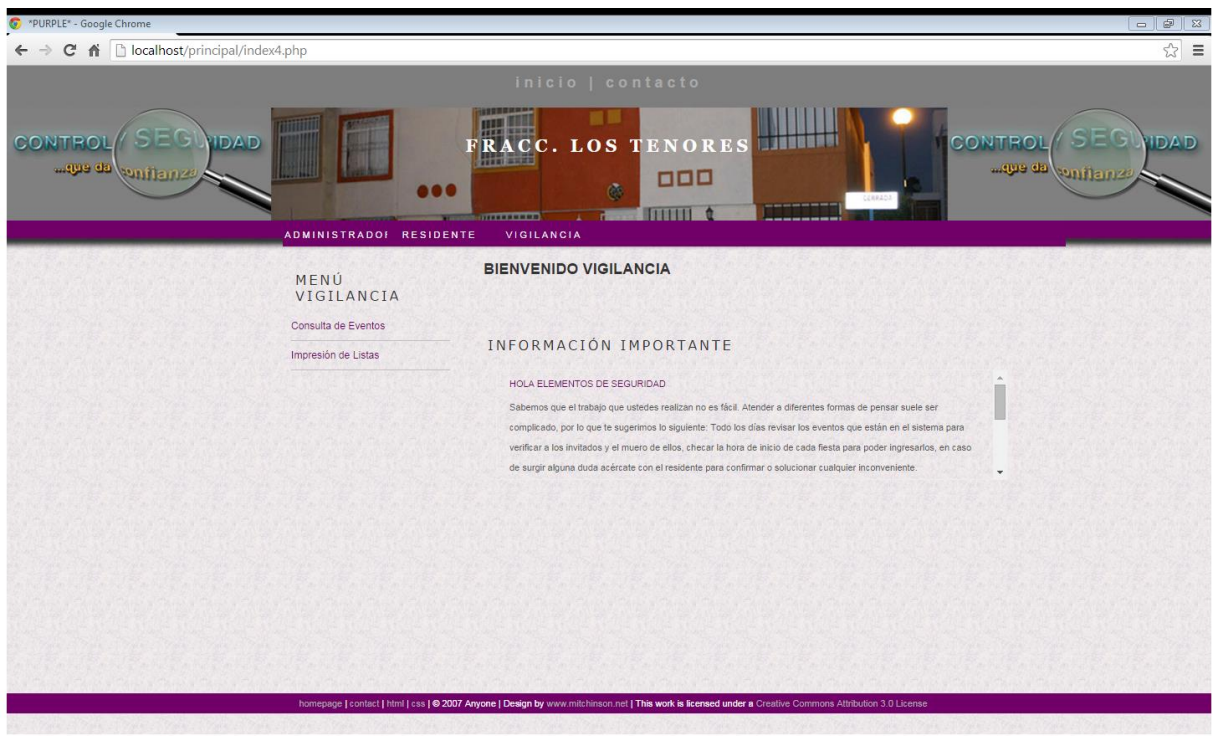


Figura 3.28 Página del Menú de vigilancia.

3.5.18. CONSULTA EVENTO (VIGILANCIA)

En la siguiente ventana (figura 3.29) se muestra la consulta de los eventos del fraccionamiento, el vigilante puede seleccionar alguno de ellos para ver la lista de los invitados.



Figura 3.29 Página donde se muestran la lista de los eventos (vigilancia).

3.5.19. IMPRIMIR LISTA DE INVITADOS (VIGILANCIA)

El vigilante podrá seleccionar el evento a imprimir (figura 3.30) para apoyarse en la recepción de los invitados. Y en la figura 3.31 se muestra la ventana que contiene la lista del evento seleccionado previo a la impresión.



Figura 3.29 Página donde se muestran la lista de los eventos a imprimir (vigilancia).

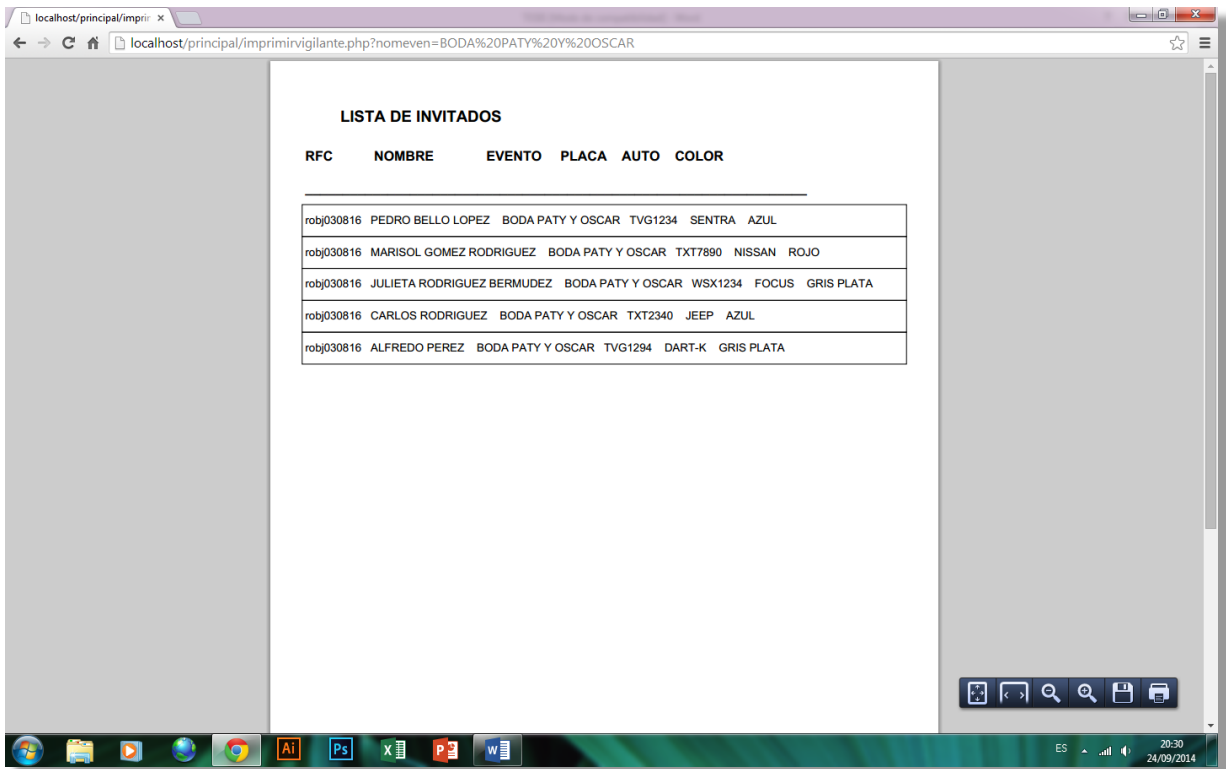


Figura 3.29 Página donde se muestran la lista de los invitados a imprimir (vigilancia).

CONCLUSIONES

El sistema para la Web se desarrolló y se implementó con Software libre, se utilizaron diferentes programas, para la base principal fue StarWampServer que proporciona Apache, Php, y MySQL siendo uno de los programas más confiables en el desarrollo de sistemas para la Web.

Los objetivos del sistema se cumplieron y se desarrolló satisfactoriamente una aplicación para el control de acceso de invitados a un fraccionamiento, el cual se puede aplicar a los fraccionamientos de la ciudad de Puebla como un paso inicial.

Gracias al lenguaje unificado de modelado (ULM) Se utilizó un método que nos permitiera diseñar nuestro sistema, que entre algunos de los beneficios que se obtuvieron son: Mejor tiempo para el diseño, modelar sistema y utilizar conceptos orientados a objetos, etc.

Las pruebas Realizadas hasta ahora son satisfactorias, y están basadas en los requerimientos de los casos de uso y esto nos garantiza que el sistema para la web funciona al 100%.

TRABAJO A FUTURO

Los objetivos se plantearon para el desarrollo del sistema y se cumplió con este trabajo de tesis y además se realizó satisfactoriamente, sin embargo, siempre hay cosas nuevas que se pueden implementar. A continuación se plantean algunos trabajos a futuro para mejorar el sistema.

- Crear una aplicación móvil para celulares.
- Crear un chat que pueda comunicar a los residentes del fraccionamiento.
- La ampliación del sistema a eventos masivos como estadios, teatros o estacionamientos privados.

BIBLIOGRAFÍA

- [1] Hernán Beati, Php Creación De Páginas Web Dinámicas, Alfaomega, México, 2011.
- [2] SponaHelma "Programación de Bases de Datos con Mysql y PHP", Alfaomega, México.
- [3] Eric J, Braude. "Ingeniería de software. Una perspectiva orientada a objetos". Alfaomega, México, 2007.
- [4] Kendall & Kendall, "Análisis y Diseño de Sistemas", Pearson, México, 2011.
- [5] Fowler Martin "ULM Gota A Gota", Pearson Educación, México.
- [6] SponaHelma "Programación de Bases de Datos con Mysql y PHP", Alfaomega, México.
- [7] Pavón Puertas Jacobo, "Creación De Un Portal Con Php Y Mysql", Alfaomega Grupo Editor S.A, México.
- [8] <http://www.icorp.com.mx>
- [9] <http://www.dvr.mx/>
- [10] Programa Visual Paradigm For ULM. V10.2(Licencia BUAP)
- [11] Procesamiento de bases de datos 8ª Edición David M. Kroenke. Pearson, México 2003
- [12] D.M.J.S. García, Diseño De Bases De Datos, Puebla 2012
- [13] Free CSS Templates www.free-css.com
- [14] Ramez Elmasri, Shamkant B. Navathe Fundamentos de Sistemas Base de Datos 5º Edición. Pearson 2007, Impreso en España.

APÉNDICE A

CREACIÓN DE LA BASE DATOS Y LAS TABLAS

Este es el script que se utilizó en la consola de MySQL para crear la base de datos y las tablas, al final del script se insertar los datos de los tres actores (Administrador, Residente y Vigilancia), además de sus generales se definen su nombre de usuario y su password.

```
DROP DATABASE IF EXISTS control;  
CREATE DATABASE control;  
use control;  
DROP TABLE IF EXISTS residente;  
CREATE TABLE residente(  
    rfc VARCHAR(10) NOT NULL PRIMARY KEY,  
    paterno VARCHAR(15) NOT NULL,  
    materno VARCHAR(15) NOT NULL,  
    nombre VARCHAR(20) NOT NULL,  
    direccion VARCHAR(128) NOT NULL,  
    email VARCHAR(50),  
    tipo INT NOT NULL,
```

```
password VARCHAR(10) NOT NULL);
```

```
describe residente;
```

```
DROP TABLE IF EXISTS telefono;
```

```
CREATE TABLE telefono(rfc VARCHAR(10) NOT NULL PRIMARY KEY, casa  
VARCHAR(10),
```

```
cel VARCHAR(10), trabajo VARCHAR(10),
```

```
FOREIGN KEY(rfc) REFERENCES residente(rfc) ON DELETE CASCADE ON  
UPDATE CASCADE);
```

```
describe telefono;
```

```
DROP TABLE IF EXISTS evento;
```

```
CREATE TABLE evento(  
rfc VARCHAR(10) NOT NULL,
```

```
nomeven VARCHAR(80) NOT NULL,
```

```
numeven INT NOT NULL,
```

```
fechaeven DATE NOT NULL,
```

```
horaini TIME,
```

```
horafin TIME,
```

```
status INT,
```

```
FOREIGN KEY(rfc) REFERENCES residente(rfc) ON DELETE CASCADE ON  
UPDATE CASCADE);
```

```
describe evento;
```

```
DROP TABLE IF EXISTS invitado;
```

```
CREATE TABLE invitado(  
rfc VARCHAR(10) NOT NULL,
```

```
rfc VARCHAR(10) NOT NULL,
```

```
nomeven VARCHAR(80) NOT NULL,  
nombreinv VARCHAR(60) NOT NULL,  
placa VARCHAR(7),  
marca VARCHAR(15),  
color VARCHAR(15),  
imagen VARCHAR(50),  
FOREIGN KEY(rfc) REFERENCES residente(rfc) ON DELETE CASCADE ON UPDATE CASCADE);
```

describe invitado;

```
INSERT INTO residente (rfc,paterno,materno,nombre,direccion,tipo,password)  
VALUES( 'robj640829','Rodríguez','Bermúdez','Juan Oscar','José Carreras # 22 Los  
Tenores','0','oscar2943');
```

```
INSERT INTO residente (rfc,paterno,materno,nombre,direccion,tipo,password)  
VALUES( 'robj030816','Rodríguez','Rosas','Oscar Efrén','José Carreras # 22 Los  
Tenores','2','efren2943');
```

```
INSERT INTO residente (rfc,paterno,materno,nombre,direccion,tipo,password)  
VALUES( 'robj030810','Gonzalez','Vivaldo','Elias','Fray Juan de rivas # 3222 col. 3  
cruces','1','elias2943');
```

```
INSERT INTO telefono (rfc) VALUES ( 'robj030810');
```

```
INSERT INTO telefono (rfc) VALUES ( 'robj640829');
```

```
INSERT INTO telefono (rfc) VALUES ( 'robj030816');
```

APÉNDICE B