



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

“SISTEMA PARA LA ADMINISTRACIÓN DE
UN ESTACIONAMIENTO PÚBLICO”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A

F R A N C I S C O L U N A L U Z



DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA.

Ciudad Universitaria

Mayo 2013

ÍNDICE

CAPÍTULO 1 ENTORNO DEL PROBLEMA.....	1
1.1 Introducción	1
1.2 Situación de la empresa	6
1.3 Conceptos básicos de administración	11
CAPÍTULO 2 MARCO TEÓRICO	14
2.1 Características, ventajas y desventajas del modelo de base de datos relacionales.....	14
Base de datos relacionales	15
Cardinalidad en las relaciones.....	19
Algebra relacional	20
Operaciones básicas.....	20
Operaciones no básicas	21
Las formas normales.....	22
Ventajas de las bases de datos relacionales.....	23
Desventajas de las bases de datos relacionales	24
2.2 Características, ventajas y desventajas del lenguaje de programación Visual C#.....	25
Desventajas	26
2.3 Características, ventajas y desventajas del motor de base de datos relacionales	
PostgreSQL	27
Ventajas de PostgreSQL.....	28
Desventajas de PostgreSQL.....	29
2.4 Características, ventajas y desventajas de la metodología de RUP y lenguaje UML.....	30
RUP (Rational Unified Process, Proceso Unificado de Rational)	30
UML (Unified Modeling Language, Lenguaje Unificado de Modelado)	31
Diagramas UML que se utilizan en RUP.....	34
Características de RUP	36
Ventajas de la metodología RUP	36
Desventajas de la metodología RUP	36
CAPÍTULO 3 ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA.....	38
3.1 Análisis del problema	38

3.2 Recopilación y análisis de la información	43
3.3 Requerimientos generales y particulares de la aplicación.....	54
Requerimientos generales	54
Requerimientos Particulares	55
Requerimientos de Seguridad	58
3.4 Planteamiento de la solución y posibles módulos.....	59
3.5 Justificación de la metodología y software a utilizar	63
CAPÍTULO 4 DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN.....	67
4.1 Modelado del Sistema.....	67
4.1.1 Diagrama de casos de uso	68
4.1.2 Diagrama de modelado de datos	80
4.1.3 Diagrama entidad – relación (DER)	82
4.1.4 Diagrama de clases	85
4.1.5 Diccionario de Datos.....	88
4.2 Implementación de Base de Datos	102
4.3 Diseño de interfaz de usuario	117
4.4 Generación de pruebas y mantenimiento.....	127
4.5 Generación de Reportes.....	137
CONCLUSIONES	143
BIBLIOGRAFÍA.....	145

ÍNDICE DE FIGURAS

Figura 1.1.1 Estacionamientos y pensiones en el D.F.	3
Figura 2.2.1 Estacionamiento público de autoservicio	7
Figura 1.3.2 Estacionamiento privado	8
Figura 1.2.4 Valet parking	10
Figura 2.1.1 Representación gráfica de una entidad	17
Figura 2.1.2 Diagrama de una entidad con un atributo	18
Figura 2.1.3 Diagrama de una relación entre dos entidades	18
Figura 2.1.4 Representación de una relación con cardinalidad uno a uno	19
Figura 2.1.5 Representación de una relación con cardinalidad Uno a Muchos	19
Figura 2.1.6 Representación de una relación con cardinalidad Muchos a Muchos	20
Figura 2.2.1 Logo del software Microsoft Visual Studio 2011	26
Figura 2.3.1 Logo de PostgreSQL	29
Figura 2.4.1 Disciplinas, fases e iteraciones del RUP	34
Figura 2.4.2 Logo del modelo RUP	37
Figura 3.1.1 Problemática de la falta de administración de los cajones de estacionamiento	39
Figura 3.1.2 Automóvil mal estacionado en el lugar destinado para personas con capacidades diferentes	41
Figura 3.1.3 Cajones para motocicletas	42
Figura 3.1.4 Robo de automóviles en estacionamientos públicos	43
Figura 3.2.1 Formulario cliente	44
Figura 3.2.2 Formulario vehículo	45
Figura 3.2.3 Formulario empleado	47
Figura 3.2.4 RFC	48
Figura 3.2.5 Ejemplo de factura	49
Figura 3.2.6 Ejemplo de un boleto de estacionamiento	53
Figura 3.2.7 Cuentas por pagar	53
Figura 3.3.1 Requerimiento particular, Menú	57
Figura 3.4.1 Modulo cajones	61
Figura 4.1.1.1 Diagrama general de casos de uso del sistema de administración de estacionamiento público	69
Figura 4.1.1.2 Diagrama de casos de uso registrar vehículo	70
Figura 4.1.1.3 Diagrama de secuencia, registro de vehículo	73
Figura 4.1.1.4 Caso de uso Registrar Salida	74

Figura	4.1.1.5 Diagrama de secuencia, registro salida	76
Figura	4.1.1.6 Caso de uso facturar	76
Figura	4.1.1.7 Caso de uso registrar pagos	77
Figura	4.1.1.8 Caso de uso administrar empleados	77
Figura	4.1.1.9 Diagrama de secuencia para el registro de empleado	79
Figura	4.1.2.1 Modelo de datos registro de pensión para clientes registrados	81
Figura	4.1.3.1 DER de las entidades Empleado, Privilegios, Referencia y Oficio.	83
Figura	4.1.3.2 DER de las entidades Tarifa_vehiculo, Tarifa_otros, Boletto, Cajon, Vehiculos_est, Marca y Color	84
Figura	4.1.3.3 DER de las entidades Empleado, Boletto, Cliente, Factura, Bitacora_cliente, Caja y Corte	85
Figura	4.1.4.1 Diagrama de clases contrato servicio por hora	87
Figura	4.2.1 Ejecución del instalador	102
Figura	4.2.2 Contraseña de la base de datos	103
Figura	4.2.3 Creación de base de datos y un usuario	104
Figura	4.2.4 Ingreso la terminal sql con nuevo role creado	106
Figura	4.2.5 Creación de una tabla en la terminal	106
Figura	4.2.6 Logo de pgAdminIII	107
Figura	4.2.7 Apariencia de la herramienta pgAdminIII	107
Figura	4.2.8 Password para conectarse al servidor	108
Figura	4.2.10 Bases de datos	108
Figura	4.2.11 Tablas	109
Figura	4.2.12 Creación de nueva tabla	110
Figura	4.2.13 Ventana new table	110
Figura	4.2.14 Ventana new columns	111
Figura	4.2.15 Columnas agregadas	112
Figura	4.2.16 Consulta en pgAdmin	113
Figura	4.2.18 Resultado de consulta en pgAdmin	114
Figura	4.2.19 Consulta con herramienta gráfica	115
Figura	4.2.20 Consulta sql generada	116
Figura	4.3.1 Estructura de despliegue de ventanas	118
Figura	4.3.2 Pagina web de Npgqsl	119
Figura	4.3.3 Ventana login de usuario	120
Figura	4.3.4 Ventana menú principal	121
Figura	4.3.5 Ventana cuota	122
Figura	4.3.6 Ventana nuevo ingreso	123
Figura	4.3.7 Ventana mapa estacionamiento	124

<i>Figura</i>	<i>4.3.8 Ventana nuevo ingreso</i>	<i>125</i>
<i>Figura</i>	<i>4.3.9 Ventana ajustes</i>	<i>126</i>
<i>Figura</i>	<i>4.3.10 Ventana Usuarios</i>	<i>127</i>
<i>Figura</i>	<i>4.4.1 Validación en ventana Login</i>	<i>128</i>
<i>Figura</i>	<i>4.4.2 Validación en ventana Login</i>	<i>128</i>
<i>Figura</i>	<i>4.4.3 Ventana login con datos correctos</i>	<i>129</i>
<i>Figura</i>	<i>4.4.4 Ventana menú principal</i>	<i>129</i>
<i>Figura</i>	<i>4.4.5 Ventana menú ajustes</i>	<i>130</i>
<i>Figura</i>	<i>4.4.6 Validación del campo No Boleto</i>	<i>131</i>
<i>Figura</i>	<i>4.4.7 Validación de campos en ventana Nuevo Registro</i>	<i>132</i>
<i>Figura</i>	<i>4.4.8 Validación del campo Cajón</i>	<i>132</i>
<i>Figura</i>	<i>4.4.9 Ventana por Hora</i>	<i>133</i>
<i>Figura</i>	<i>4.4.10 Restricción de caracteres</i>	<i>133</i>
<i>Figura</i>	<i>4.4.11 Verifica campos de registro del vehículo</i>	<i>134</i>
<i>Figura</i>	<i>4.4.12 Verifica si puede abrir la ventana de boleto por hora</i>	<i>134</i>
<i>Figura</i>	<i>4.5.1 Ventana vehículos</i>	<i>138</i>
<i>Figura</i>	<i>4.5.2 PDF de reporte de vehículos registrados por el servicio de hora</i>	<i>138</i>
<i>Figura</i>	<i>4.5.3 Ventana boletos</i>	<i>139</i>
<i>Figura</i>	<i>4.5.4 PDF reporte de boletos activos</i>	<i>139</i>
<i>Figura</i>	<i>4.5.5 Ventana estadísticas</i>	<i>140</i>
<i>Figura</i>	<i>4.5.6 Ventana clientes</i>	<i>141</i>
<i>Figura</i>	<i>4.5.8 Ventana empleados</i>	<i>142</i>

INDICE DE TABLAS

<i>Tabla 3.5.1 Tabla comparativa de bases de datos.....</i>	<i>64</i>
<i>Tabla 3.5.2 Características de Power Builder y Visual Studio.....</i>	<i>66</i>
<i>Tabla 4.1.1.1 Secuencia registrar vehículo.....</i>	<i>72</i>
<i>Tabla 4.1.1.2 Secuencia registrar salida.....</i>	<i>75</i>
<i>Tabla 4.1.1.3 Secuencia administrar empleado.....</i>	<i>78</i>
<i>Tabla 4.1.5.1 Referencias.....</i>	<i>89</i>
<i>Tabla 4.1.5.2 Marca.....</i>	<i>90</i>
<i>Tabla 4.1.5.3 Color.....</i>	<i>90</i>
<i>Tabla 4.1.5.4 Privilegios.....</i>	<i>91</i>
<i>Tabla 4.1.5.5 Empleados.....</i>	<i>92</i>
<i>Tabla 4.1.5.6 Oficios.....</i>	<i>93</i>
<i>Tabla 4.1.5.7 Tipo_vehiculos.....</i>	<i>93</i>
<i>Tabla 4.1.5.8 Cajones.....</i>	<i>94</i>
<i>Tabla 4.1.5.9 Tarifa_otros.....</i>	<i>95</i>
<i>Tabla 4.1.5.10 Tarifa_vehículos.....</i>	<i>95</i>
<i>Tabla 4.1.5.11 Vehículos_est.....</i>	<i>96</i>
<i>Tabla 4.1.5.12 Boletos.....</i>	<i>97</i>
<i>Tabla 4.1.5.13 Clientes.....</i>	<i>98</i>
<i>Tabla 4.1.5.14 Facturas.....</i>	<i>99</i>
<i>Tabla 4.1.5.15 Bitacora_cliente.....</i>	<i>100</i>
<i>Tabla 4.1.5.16 Cortes.....</i>	<i>101</i>
<i>Tabla 4.1.5.17 Caja.....</i>	<i>101</i>

CAPÍTULO 1 ENTORNO DEL PROBLEMA

1.1 Introducción

El incremento desmesurado de automóviles en la ciudad y los limitados espacios de aparcamiento han generado una gran oportunidad de emprender un negocio rentable y exitoso, la creación de estacionamientos; sin olvidar que se debe tener un control apropiado para el buen funcionamiento de éstos, de no ser así, se estaría limitando el progreso del establecimiento debido a que el cliente podría no encontrar satisfactorio el uso del servicio, si el lugar de aparcamiento no es sencillo de encontrar dentro del estacionamiento.

Un estacionamiento es de suma importancia en cualquier población; todo individuo que posea un automóvil y se dirija a su trabajo, de compras, a un aeropuerto, a dondequiera que vaya y haga uso de su auto se encontrará con la pregunta obligada ¿donde estacionarlo? y acto seguido buscará un lugar donde estacionar su vehículo, podría aparcar en alguna calle con la incertidumbre de que tan seguro está su coche, es por esta razón que un estacionamiento tiene gran relevancia en la vida diaria de una ciudad muy concurrida, porque una persona optaría por hacer uso de un establecimiento formal para estacionarse con el fin de cuidar sus intereses.

Hoy en día, encontrar de manera rápida un lugar para estacionarse en la ciudad es un gran problema. Aunado a esto, la gran cantidad de trabajos que una población en pleno desarrollo ofrece, ha originado un incremento en sus residentes que por consecuencia desemboca en una mayor cuantía de vehículos en uso.

Según la Secretaría del Medio Ambiente del Distrito Federal (SMADF) en su inventario de emisiones de contaminantes del 2008, ¹se estima que circulan más de 4.5 millones

¹Inventario de emisiones de contaminantes criterio de la ZMVM 2008 [consulta 16 de junio 2012] Disponible en:
http://www.sma.df.gob.mx/sma/links/download/biblioteca/2008ie_criterio/2008ie_criterio.pdf

de vehículos automotores en el Distrito Federal, de los cuales el 64% corresponde a unidades registradas en el Distrito Federal y el 36% restantes a unidades registradas en el Estado de México.

²De acuerdo con el Directorio Estadístico Nacional de Unidades Económicas (DENUE marzo 2012) del Instituto Nacional de Estadística y Geografía (INEGI) hay más de 11 mil 700 estacionamientos y pensiones en todo el país y 1,784 se encuentran en el Distrito Federal. En la Figura 1.1.1 se representa con puntos los establecimientos de estacionamiento y pensiones en el Distrito Federal, la mayoría se encuentra concentrada en el centro de la ciudad.

² INEGI [fecha de consulta 16 de junio 2012] de:
<http://www.inegi.org.mx/Sistemas/DENUE/Presentacion.aspx>



Figura 1.1.1 Estacionamientos y pensiones en el D.F.

Un estudio realizado por PROFECO³ más de la mitad (60%) de estos negocios ofrece el servicio de pensión, donde se puede guardar un automóvil varias horas durante días, semanas o meses, conforme al pago de una cuota periódica.

³ García Soto Carlos E. y Segovia Amadeo. Pensiones para autos *Brújula de Compra* [en línea] 12-Noviembre-2010 [fecha de consulta 4 de Junio 2012] Disponible en: http://www.profeco.gob.mx/encuesta/brujula/bruj_2010/bol183_pensiones.asp

El motivo para la realización del sistema para la administración de un estacionamiento público se basa en la falta de organización para la asignación de un espacio de estacionamiento; la forma de vida es más rápida en la ciudad de manera subsecuente la necesidad de conseguir un lugar seguro y rápido para que el conductor pueda realizar sus posteriores actividades aumenta.

Otras ventajas que tendrá este sistema son:

- Mejorar el servicio de estacionamientos, evitando la irritación de los usuarios en encontrar un lugar donde estacionarse rápidamente.
- Contar con estacionamientos que proporcionen mayor comodidad en el servicio.
- La reducción de los niveles de contaminación ocasionadas por la circulación de automóviles en recorridos innecesarios del estacionamiento.
- Conocer el estado de ocupación del estacionamiento al instante en que sea requerido por el operador.
- Proporcionar un buen espacio de estacionamiento al cliente, de acuerdo a sus necesidades, como personas con capacidades diferentes, el tamaño del vehículo, etc.

Para una mejor comprensión de cómo esta ordenada la información y el contenido de la presente tesis se dará a continuación una breve información de que tratará en cada capítulo.

En el primer capítulo Entorno del problema, se describe de forma general cada tipo de estacionamiento que se cita en el reglamento de estacionamientos del Distrito Federal, el fin para el cual fueron creados y la forma en que operan.

En el segundo capítulo Marco teórico está dedicado a los lenguajes de programación y la metodología con la que se propone el desarrollo y diseño del sistema informático.

Para cada uno de éstos se proporciona una breve introducción, conceptos utilizados en su utilización, sus características principales, sus ventajas y desventajas al ser utilizados.

En el tercer capítulo Análisis y planteamiento del problema, se documentan los problemas detectados en el estacionamiento público de autoservicio, los documentos y formularios necesarios para el funcionamiento correcto de éste, además se especifican los requisitos con los cuales debe contar el sistema.

En el último capítulo Diseño y construcción de la aplicación está dedicada a cada parte durante la fase de diseño y construcción del sistema, al inicio se especifican los diagramas que serán utilizados para la construcción del sistema, posteriormente se describe el proceso de creación del back-end y del front-end, finalizando con la presentación de casos del sistema en funcionamiento y la demostración de los reportes necesarios para la empresa.

1.2 Situación de la empresa

⁴Los estacionamientos se dividen en dos tipos según el reglamento de estacionamientos públicos del Distrito Federal (1991): Privados y Públicos, y por su tipo de servicio son de autoservicio o de acomodadores.

Los estacionamientos privados son áreas destinadas a unidades habitacionales, a instituciones o empresas siempre que el servicio otorgado sea gratuito.

Los estacionamientos públicos son los que están destinados a prestar el servicio al público de recepción, guarda, protección y devolución de vehículos, a cambio del pago por el uso de éste. A continuación se muestra el funcionamiento de forma general de los tipos de estacionamiento.

➤ Estacionamiento público de autoservicio

En este tipo de estacionamiento el conductor puede solicitar: a) servicio de estacionamiento por tiempo, la primera hora se cobra total y el siguiente tiempo de estancia se cobra por fracciones de quince minutos. b) servicio de pensión: el cliente puede dejar su automóvil estacionado por un rango de tiempo, puede ser: semanal o mensual. y se deberá escoger el horario: de día, de noche o todo el día (24 horas).

De acuerdo al servicio pedido por el cliente depende la tarifa, si se pide el servicio de pensión éste se paga por anticipado, si el servicio es por horas al día se paga antes de poder salir del estacionamiento con su vehículo.

En este tipo de estacionamientos el cliente es el que se encarga de buscar un lugar para poder dejar su automóvil, en algunos casos en el servicio de pensión se deja en un área específica para su guarda pero éste ya depende de la compañía que ofrece el servicio.

⁴ Reglamento de estacionamientos públicos del Distrito Federal pg.1 pg.2 Consultado en: <http://www.coyoacan.df.gob.mx/transparencia/art14/i/reglamentos/trim1-2009/ESTACIONAMIENTOS.pdf>

En la Figura 2.2.1 Estacionamiento público de autoservicio se muestra la forma de trabajo para este tipo de estacionamiento.

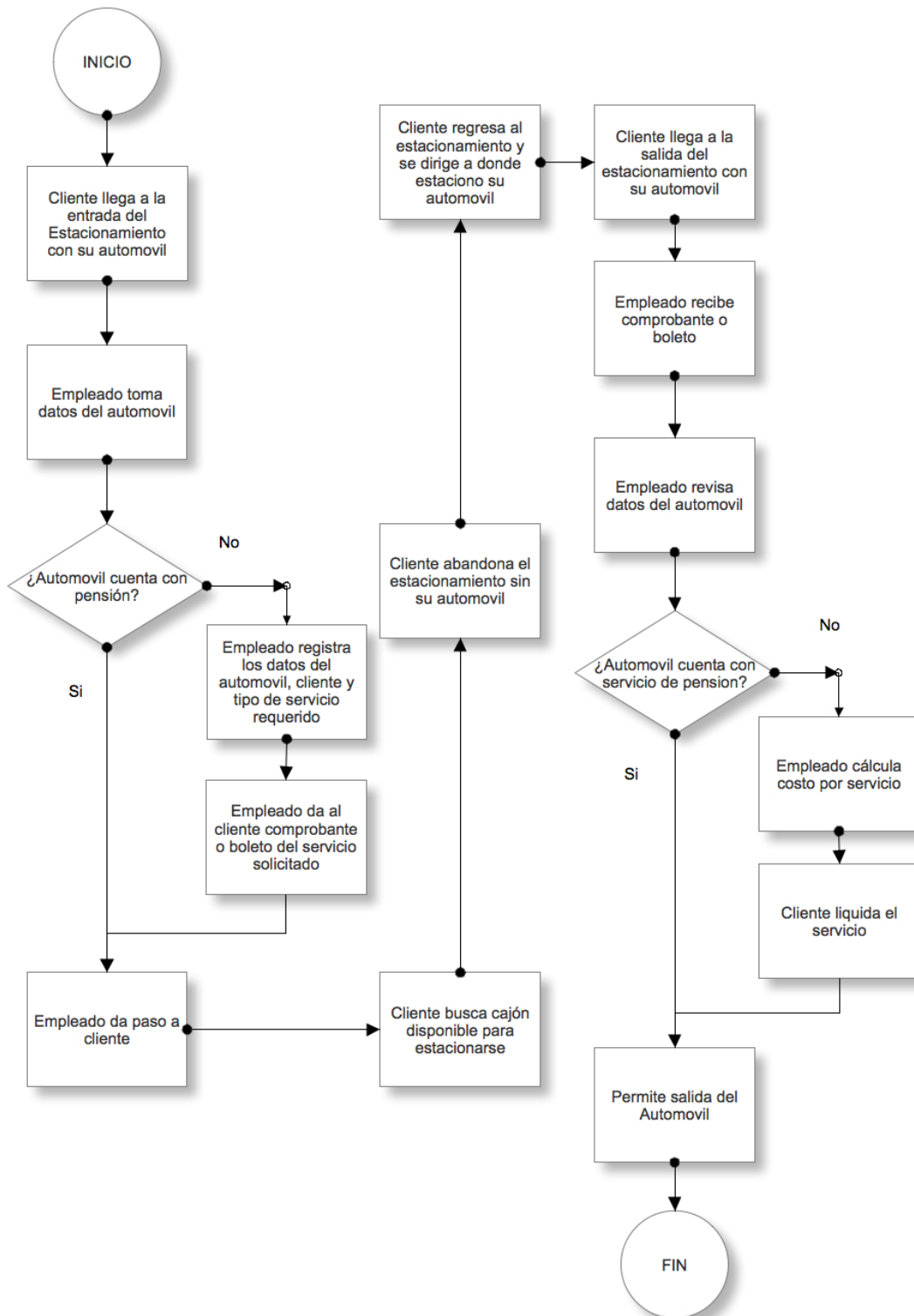


Figura 2.2.1 Estacionamiento público de autoservicio

➤ El estacionamiento privado

La forma de trabajo de estos estacionamientos es que el conductor puede tener acceso si trabaja en la empresa, no tiene que pagar ninguna cuota. A la entrada del estacionamiento se encuentra un vigilante el cual revisa si es trabajador de la empresa dejándolo acceder si lo es. Para esto primero se necesita una base de datos de los trabajadores de la empresa y que se le proporcione una identificación al trabajador. Cuando se accede al estacionamiento se toma como registro su hora de entrada y su hora de salida.

En este tipo de estacionamiento el trabajador debe buscar un cajón disponible para estacionar su automóvil. En la Figura 1.3.2 Estacionamiento privado se ve de forma general el flujo de trabajo.

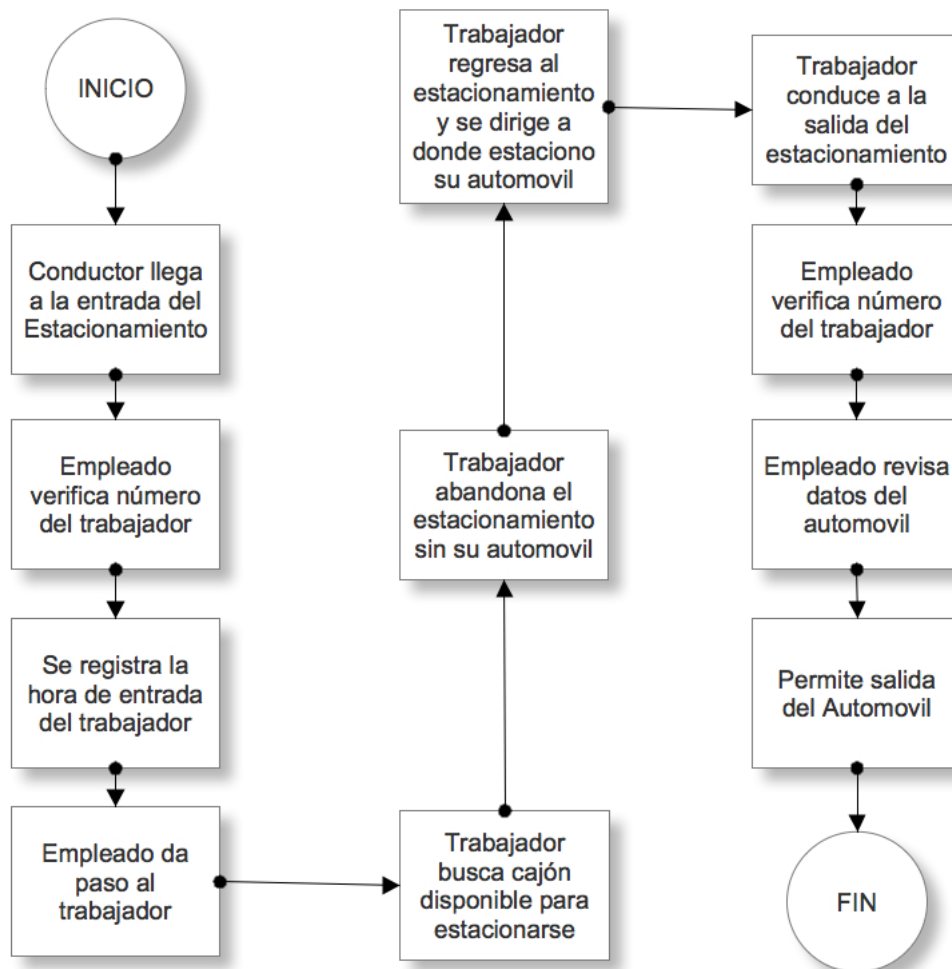


Figura 1.3.2 Estacionamiento privado

➤ El valet parking o estacionamiento restringido

En este tipo de estacionamiento el conductor no necesita estacionar el automóvil en el cajón, un empleado del estacionamiento llamado “acomodador” se encarga de recibir el automóvil junto con las llaves a la entrada del estacionamiento o establecimiento, hace entrega al cliente el recibo de recepción del vehículo y posteriormente busca y estaciona el automóvil en el cajón.

En este tipo de estacionamiento el conductor no invierte tiempo en la parte de búsqueda del cajón, y el estacionamiento tiene mayor seguridad al restringir el acceso a personas que no son empleados del estacionamiento.

Al regreso del cliente (antes conductor), éste presenta su comprobante al checador, el checador calcula el costo del servicio, el cliente paga por el tiempo que hizo uso del servicio de estacionamiento, el acomodador va en busca del automóvil del cliente para al fin entregarlo a su dueño para su posterior salida.

Por lo general este tipo de servicio se aplica por ejemplo en restaurantes.

En la Figura 1.2.4 Valet parking, se ve de forma general el flujo de trabajo.

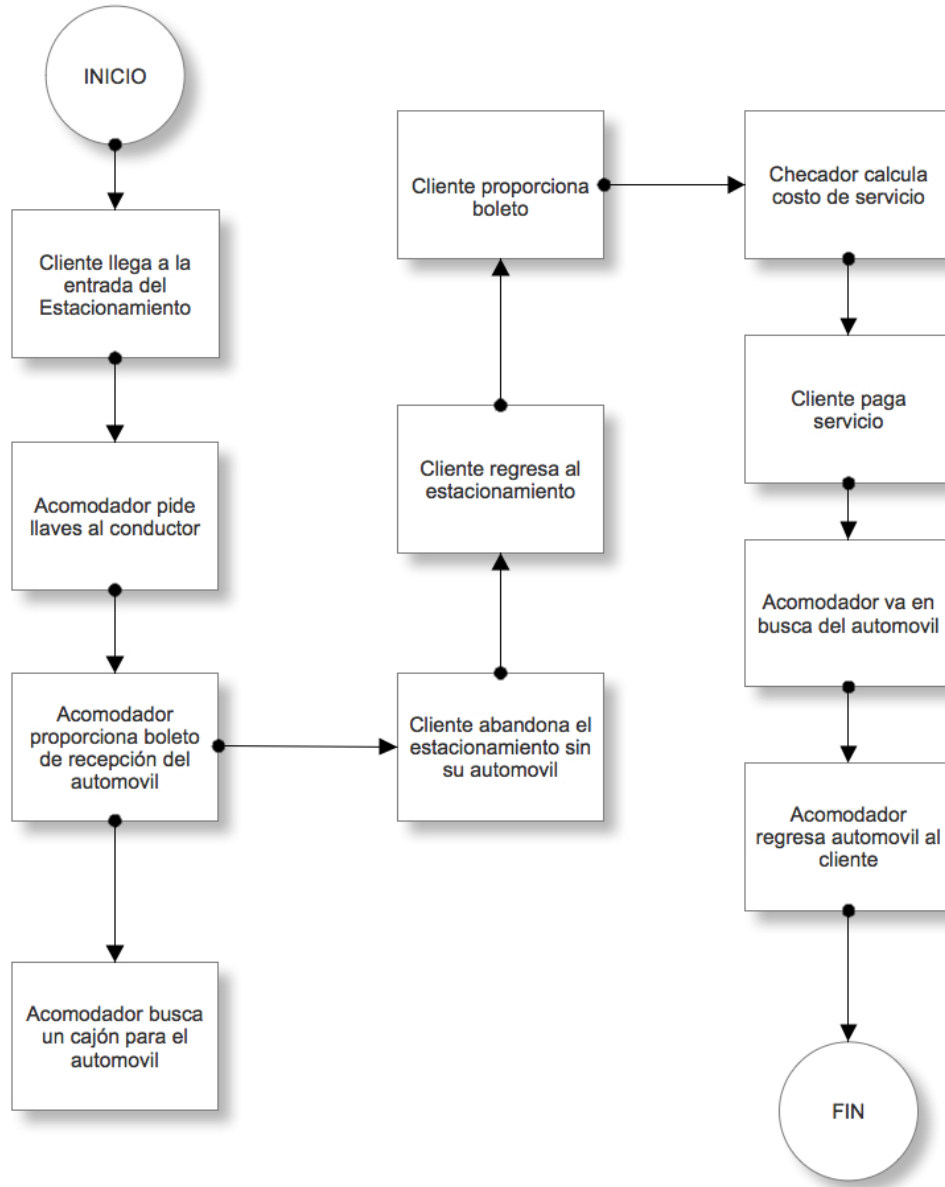


Figura 1.2.4 Valet parking

Éstos fueron algunos diagramas del funcionamiento de los estacionamientos más comunes.

Por mencionar, también existen estacionamientos automatizados, donde la forma de acomodar los autos se hace de forma mecánica aprovechando de mejor manera el espacio, tanto a lado a lado de cada auto como sobre y debajo de éstos mejorando la cantidad de entrada de autos y seguridad.

1.3 Conceptos básicos de administración

➤ Capital

Es la cantidad de recursos, bienes y valores disponibles para realizar una actividad y generar un beneficio económico.

➤ Ingreso

Es la cantidad que recibe la empresa por la venta de sus productos o servicios

➤ Activos

Son todos los bienes que la empresa posee, por ejemplo:

Efectivo: pagos efectuados por los clientes, aportes en efectivo, cobros de ingresos o rentas, dinero recibido.

Bancos: El dinero de la empresa depositado en las diferentes entidades financieras en cuentas corrientes o de ahorros, etc.

Cuentas por cobrar

➤ Pasivos

Son las deudas que la empresa se compromete a pagar, por cualquier concepto, algunos son:

Obligaciones financieras: es el valor de los préstamos que recibió la empresa.

Proveedores: son las deudas de la empresa que se hicieron por medio de crédito en mercancía.

Impuestos: Se registra el impuesto por valor agregado (IVA) que se aplica sobre las ventas del servicio, los gastos de operación, etc.

➤ Productividad

Es el indicador de eficiencia que relaciona la cantidad de producto utilizado con la cantidad de producción obtenida

➤ Competitividad

Se define como la capacidad de generar la mayor satisfacción de los consumidores al menor precio. Depende especialmente de la calidad del producto y del nivel de precios. Existen otros factores como la innovación, la calidad del servicio o la imagen corporativa.

➤ Póliza de Seguro

La póliza de seguro es el documento principal del contrato de seguro, en donde constan los derechos y obligaciones de las partes.

➤ Demanda

Es la estimación del uso de espacio disponible y la demanda de éstos.

La demanda total es la suma de los vehículos que han encontrado un lugar disponible, los vehículos que aun no se han podido estacionar y están en busca de un lugar, los que están estacionados ilegalmente y los que aun no han iniciado la búsqueda por la poca probabilidad de encontrar un lugar que se adecue a sus requerimientos.

Demanda básica: Ésta toma los autos ya estacionado y los que están en proceso de estacionarse, en maniobras de estacionamiento.

Demanda Ilegal: Ésta incluye además de la básica los que están mal estacionados.

Demanda Excedente: Son los vehículos que aun no encuentran un lugar para estacionarse.

Demanda Potencial: Son los vehículos que no van a estacionarse en el área ya que suponen que no encontrarán espacio.

CAPÍTULO 2 MARCO TEÓRICO

2.1 Características, ventajas y desventajas del modelo de base de datos relacionales

Las bases de datos son uno de los elementos más importantes de las empresas, gracias a éstas y a su correcto funcionamiento hace más sencillo llevar una mejor organización y mejor entendimiento de los datos de la compañía, ya sea de sus productos, sus empleados, clientes, etc.

Los elementos que conforman un sistema de información para poder, obtener, procesar, almacenar y presentar la información son:

- El contenido, es la información recopilada
- El equipo físico, la computadora que soporta el sistema de información
- El equipo lógico, el sistema operativo, el gestor de base de datos, el sistema de comunicaciones, etc.
- El administrador, es el encargado de asegurar los datos
- Los usuarios

No todo conjunto de estructuras de datos se le puede llamar una base de datos, ya que debe cumplir con una serie de reglas para que pueda asegurar, la independencia, la integridad y la seguridad de los datos:

- Integridad de toda la información de la organización
- Persistencia de los datos
- Accesibilidad simultanea para distintos usuarios
- Descripción unificada de los datos e independiente de los programas
- Independencia de los programas respecto a la representación física de los datos
- Definición de vistas parciales de los datos para distintos usuarios
- Mecanismos para controlar la integridad y la seguridad de los datos

Base de datos relacionales

⁵El objetivo de diseñar una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información. Una de las técnicas para diseñar esquemas es por medio de las formas normales.

Las bases de datos relacionales se basan en el modelo relacional el cual fue propuesto por Edgar Frank Codd en 1970 y ha sido el modelo de construcción para casi todos los SGBD comerciales, por ejemplo: Oracle, DB2, Microsoft SQL Server, etc.

- Su éxito reside en su sencillez, ya que se representan como un conjunto de tablas (filas y columnas) y en su fácil declaración del lenguaje de programación.
- Para este modelo su estructura básica es la de relación. Con ésta se puede representar los objetos y su relación entre ellos.

A continuación se listan conceptos para el mejor entendimiento de la base de datos relacional:

- Estructuras: Son objetos que almacenan o acceden a los datos de la base de datos, tablas, vistas, índices, reglas, constraints (restricciones), procedimientos almacenados, etc.
- Tabla: Es un objeto de almacenamiento de información la cual esta ordenada en forma de columnas y filas. Cada fila de la tabla es un registro y cada columna es un atributo. Todo valor en la tabla es atómico, que no puede ser divisible.
- Integridad: Se refiere a que la información de la base de datos sea valida y consistente.

⁵ Sanchez Jorge. Principios sobre Bases de Datos Relacionales, 2004 [fecha de consulta 4 de mayo 2012] Consultado en: <http://www.jorgesanchez.net/bd/bdrelacional.pdf>

- Acceso concurrente: Se refiere a que el manejador de la base de datos controla el acceso múltiple de varios usuarios a la base.
- Independencia física: Permite modificar el esquema físico sin tener que reescribir los programas de la aplicación.
- Independencia lógica: Permite modificar el esquema conceptual sin que se tenga que alterar el programa de la aplicación
- Facilidad de uso: Los usuarios tendrán un fácil acceso a los datos.
- Redundancia controlada: Los datos serán almacenados una sola vez, a menos que sea necesario para mantener las relaciones entre llaves primarias y llaves foráneas.
- Seguridad de Acceso: Se podrá especificar quienes podrán ver los datos, totales, parciales, o denegarle la base de datos.
- Operaciones: Son el resultado de utilizar operadores para definir o manipular las estructuras.
- Reglas de integridad: Gobiernan los tipos de acciones permitidas en los datos y la estructura de la base de datos. Protegen los datos y estructuras.
- Identificador único: No pueden existir dos columnas con el mismo nombre en la misma tabla y los valores almacenados en la misma columna corresponden al mismo tipo de dato.
- Clave única: Cada tabla tiene un campo o varios para poder identificar de manera única un registro, no existen registros con el mismo identificador.

- Clave primaria. Es una clave única que identifica a todos los demás atributos de la tabla, para poder especificar los datos que serán relacionados con las demás tablas. La forma de hacerlo es por medio de claves foráneas. Solo puede existir una clave primaria por tabla y ningún campo puede ser de valor NULO.
- Dominio: Un dominio describe un conjunto de valores que puede adoptar un atributo. Restringe el tipo de dato del atributo.
- Normalización: El modelo relacional pasa por un proceso de normalización, el objetivo de esto es que se diseñe un esquema que permita su utilización de forma eficiente.

⁶En el proceso de diseño de la base de datos se desarrolla el modelo lógico llamado Entidad-Relación (ER). Este modelo se implementa de manera gráfica, donde se muestra las relaciones entre los datos que se están modelando. El modelo ER está formado por:

- Entidades: Es un modelo lógico. En gramática es como un sustantivo. Véa figura 2. 1.1

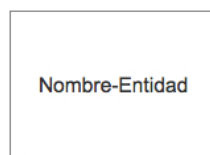


Figura 2.1.1 Representación gráfica de una entidad

- Atributos: Las características de la entidad, es información más detallada de la entidad. Véa figura 2.1.2

⁶ Celma Gimenez- Matilde Bases de Datos Relacionales, Ed, Prentice Hall. 2003 Pp. 40 - 44

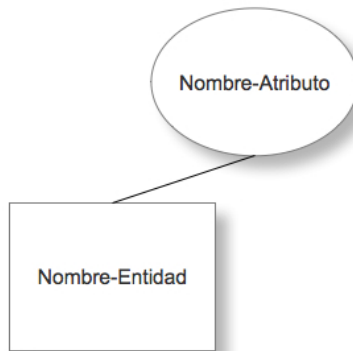


Figura 2.1.2 Diagrama de una entidad con un atributo

- Relaciones: Las relaciones entre las entidades representan la dependencia entre ellas. Véa figura 2.1.3



Figura 2.1.3 Diagrama de una relación entre dos entidades

Cardinalidad en las relaciones

⁷La cardinalidad representa el número de ocurrencia de una entidad en una relación:

- Una a una (1:1). Una sola entidad de A esta asociada a una entidad de B y viceversa. Véa figura 2.1.4



Figura 2.1.4 Representación de una relación con cardinalidad uno a uno

- Una a muchas (1:N) Una entidad de A esta asociada a una o varias entidades en B, en cambio una entidad B sólo puede estar asociada con una sola entidad de A. Véa figura 2.1.5



Figura 2.1.5 Representación de una relación con cardinalidad Uno a Muchos

⁷ Celma Gimenez- Matilde Bases de Datos Relacionales, Ed, Prentice Hall. 2003 Pg. 45

- Muchas a muchas(N:M). Una entidad en A esta asociada a cualquier cantidad de entidades en B; y una entidad en B está asociada a cualquier cantidad de entidades en A. Vea figura 2.1.6



Figura 2.1.6 Representación de una relación con cardinalidad Muchos a Muchos

Algebra relacional

Conjunto de operaciones que se aplican en las relaciones. Estas operaciones se utilizan como un paso intermedio de una consulta a la base de datos.

Operaciones básicas

➤ Selección (σ)

Permite seleccionar un subconjunto de tuplas de una relación(R), todas aquellas que cumplan la condición P.

➤ Proyección (Π)

Permite extraer columnas (atributos) de una relación, resultando un subconjunto vertical de atributos de la relación R

➤ Producto cartesiano (\times)

El producto cartesiano de dos relaciones se escribe de la siguiente manera:

$$R \times S$$

Entrega una relación, cuyo esquema corresponde a una combinación de todas la tuplas de R con cada una de las tuplas de S, y sus atributos corresponden a los de R seguidos por los de S.

- Unión (\cup)

R U S

Retorna el conjunto de tuplas que están en R, o en S, o en ambas, R y S deben ser uniones compatibles.

- Diferencia ($-$)

R-S

La diferencia entre dos relaciones, R y S, entrega todas a aquellas tuplas que estén en R pero no estén en S. R y S deben de ser uniones compatibles.

Operaciones no básicas

- Intersección (\cap)

La intersección de dos relaciones se puede especificar en función de los operadores básicos:

$$R \cap S = R - (R - S)$$

Como en la teoría de los conjuntos, corresponde al conjunto de todas las tuplas que están en R y en S, siendo R y S uniones compatibles.

- Unión Natural (\bowtie) Natural Join

Ésta permite reconstruir las tablas originales previas al proceso de normalización. Consiste en combinar la proyección, selección y producto cartesiano en una sola operación, donde la condición Θ es la igualdad Clave Primaria = Clave Externa (o Foránea), y la proyección elimina la columna duplicada (clave externa).

Expresada en las operaciones básicas queda:

$$R \bowtie S = \Pi_{A_1, A_2 \dots A_n}(\sigma_{\theta}(R \times S))$$

➤ División (/)

Supongamos que tenemos dos relaciones $A(x,y)$ y $B(y)$ donde el dominio de y en A y B , es el mismo.

El operador división A/B retorna los distintos valores de x tales que para todo valor y en B existe una tupla (x,y) en A

➤ Agrupación (G)

Permite agrupar conjuntos de valores en función de un campo determinado y hacer operaciones con otros campos.

Las formas normales

Primera forma normal (1FN)

Una tabla se encuentra en primera forma normal si impide que un atributo de una tabla pueda tomar más de un valor.

Se dice que un conjunto de atributos (Y) depende funcionalmente de otro conjunto de atributos (X) si para cada valor de X hay un único valor posible para Y . Simbólicamente se denota por $X \rightarrow Y$

Segunda forma normal (2FN)

Una tabla en 1NF está en 2FN si y solo si, dada una clave primaria y cualquier atributo que no sea un constituyente de la clave primaria, el atributo no clave depende de toda la clave primaria en vez de solo una parte de ella.

Tercera forma normal (3FN)

Una tabla se encuentra en 3FN si esta en 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave

Cuarta forma normal (4FN)

Una tabla se encuentra en 4FN si, y sólo si, para cada una de sus dependencias múltiples no funcionales $X \twoheadrightarrow Y$, siendo X una súper-clave que, X es o una clave candidata o un conjunto de claves primarias.

Quinta forma normal (5FN)

Una tabla esta en 5FN si esta en 4FN si, y sólo si, cada relación de dependencia se encuentra definida por las claves candidatas.

Ventajas de las bases de datos relacionales

- La información no tiene un orden dentro de la base
- Todos los valores son atómicos
- Se puede extraer cualquier información que contenga la base de datos, en cualquier momento.
- Evitan la redundancia de datos
- Se reduce en gran medida del riesgo de inconsistencia
- Facilita el intercambio de datos y la administración de los mismos

- Mantiene la integridad de los datos, gracias a las restricciones.
- El uso de una base de datos relacional bien diseñada puede reducir mucho la cantidad de datos que debe ingresar cada vez que se agrega un registro.
- Garantiza la integridad referencial, al eliminar un registro todos los registros relacionados dependientes también se eliminan. Se debe indicar por lo menos las llaves primarias y foráneas.

Desventajas de las bases de datos relacionales

- Si la base de datos es centralizada puede ocasionar la pérdida de información si existe una falla en el equipo.
- Si se abusa de los índices crece desmesuradamente y perjudica el rendimiento y el proceso de mantenimiento.

2.2 Características, ventajas y desventajas del lenguaje de programación Visual C#

⁸Visual Studio C#

C# es un lenguaje de programación que tiene su base en la familia de lenguaje C es similar a C, C++ o a Java, completamente orientado a objetos.

El objetivo principal de este lenguaje de programación es el de la simplicidad. Muchas características (o ausencias) contribuyen a la simplicidad. Simplifica y moderniza C++ en las áreas de las clases, los espacios de nombres, la sobrecarga de métodos y la manipulación de excepciones. Gran parte de la complejidad de C++ fue eliminada de C# con el fin de facilitar su uso y cometer menos errores.

Contribuyendo a la facilidad se destaca la eliminación de ciertas características de C++: desaparecen las macros, las plantillas y la herencia múltiple. Estas características crean más problemas que ventajas.

Unas de las características de C# se describen a continuación:

- No existen punteros. Se trabaja con código manipulado, donde no están autorizadas las operaciones inseguras, como la manipulación directa de la memoria.
- Suprime los distintos operadores a favor de uno solo: el punto (.). Lo único que tiene que entender el programador es la noción de nombres anidados.
- C# proporciona un sistema de tipos unificado. Este sistema le permite ver todos los tipos como un objeto, ya sea un tipo primitivo o una clase.

⁸ C# para Desarrolladores de Java Ed. MacGrawHill, España 2003 pg. 3 – pg. 8

- Nuevos tipos de datos, decimal, dirigido a cálculos financieros. Puede crear nuevos tipos adaptados específicamente a la aplicación.
- Está orientado a objetos: soporta todos los conceptos orientados a objetos, como el encapsulamiento, la herencia y el polimorfismo.
- No hay funciones, variables o constantes globales. Todo debe ser englobado en una clase, ya sea un miembro de una clase o un miembro estático. Esto hace que el código sea más legible y le ayuda a reducir los conflictos potenciales de los nombres.
- Recolector de basura automático

Desventajas

- Por defecto, el código funciona en modo seguro, donde no se permite que haya punteros. No obstante, si se necesita que haya punteros, podrá usarlos a través de códigos no seguros, y no hay control cuando se llama al código no seguro.
- Cuando se está en modo no seguro, el recolector de basura no puede tocar sus ubicaciones de memoria y moverlas como si lo hiciera en código manipulado.

En la figura 2.2.1 se muestra el logo de Visual Studio C#



Figura 2.2.1 Logo del software Microsoft Visual Studio 2011

2.3 Características, ventajas y desventajas del motor de base de datos relacionales PostgreSQL

⁹PostgreSQL es un Sistema de Gestión de Bases de Datos (DBMS) que incorpora el modelo relacional para sus bases de datos y es compatible con el lenguaje de consulta estándar ANSI-SQL: 2008. Es de código abierto, cuenta con 16 años de desarrollo activo y ha ganado su reputación por su fiabilidad.

Tiene muy buenas características de rendimiento. Funciona en casi cualquier plataforma UNIX, incluyendo los sistemas tipo UNIX, como FreeBSD, Linux, y MAC OS X. También se puede ejecutar en servidores Microsoft Windows NT/2000/2003.

Es completamente compatible con ACID, tiene soporte completo para llaves foráneas, uniones, vistas, triggers y procedimientos almacenados. Se incluye la mayor parte de los tipos de datos de SQL: 2008.

Soporta el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonido, o video.

¹⁰PostgreSQL puede ser utilizado desde casi cualquier lenguaje de programación importante, como C, C++, Perl, Python, Java, Tcl, PHP, Ruby, ODBC, entre otros.

PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales como DB2, Oracle o SQLServer.

Características de PostgreSQL

⁹ *PostgreSQL-es* [fecha de consulta 16 de junio 2012] en: http://www.postgresql.org.es/sobre_postgresql

¹⁰ Neil Matthew y Richard Stones. *Beginning Databases with PostgreSQL From Novice to Professional*, Segunda Edición Ed. Apress pg.1

- Su documentación es bastante amplia y detallada
- Es un sistema de gestión de bases de datos relacionales
- Soporte completo de ACID (Atomicity Consistency Isolation Durability)
- Soporta Triggers o disparadores, que son procedimientos que se lanzan automáticamente bajo determinadas circunstancias como cuando ocurren actualizaciones, inserciones o eliminaciones de registros en una tabla. Permiten establecer reglas de integridad y consistencia a nivel del servidor de base de datos.
- Posee soporte para vistas, que son un conjunto de registros, resultado de una consulta que se comporta como una tabla física para facilitar su manejo.
- Tiene soporte para todos los tipos de JOINS o uniones entre tablas, cumpliendo con los estándares de sintaxis SQL.
- Buena Seguridad. Gracias a la buena gestión de usuarios, grupos de usuarios y contraseñas, así como permisos asignados a cada uno de ellos mediante sentencias SQL.
- Soporta backups o respaldos en caliente (mientras trabaja el servidor PostgreSQL) y recuperación completa de las bases de datos.

A continuación se lista de forma separada las ventajas y desventajas de PostgreSQL.

Ventajas de PostgreSQL

Ventajas con respecto a los manejadores de Base de Datos comerciales

- Económico. No existe algún costo para su utilización.
- No es necesario tener una licencia
- Ahorro en costo de operación. Ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos comerciales conservando todas las características de estabilidad y rendimiento.
- Extensible. El código es abierto.
- Multiplataforma: Puede instalarse en varias plataformas como: Linux, Mac OS x, Solaris, Windows.

- Herramientas gráficas de diseño y administración de bases de datos. Como pgAdmin y pgAcces

Desventajas de PostgreSQL

- Soporte en línea: Existen foros oficiales donde pueden asesorarte pero no es de forma obligatoria
- Consume más recursos que MySQL, por lo que se necesita mayores características de hardware para ejecutarlo
- El software libre no tiene garantía

En la figura 2.3.1 se muestra el logo de PostgreSQL

PostgreSQL

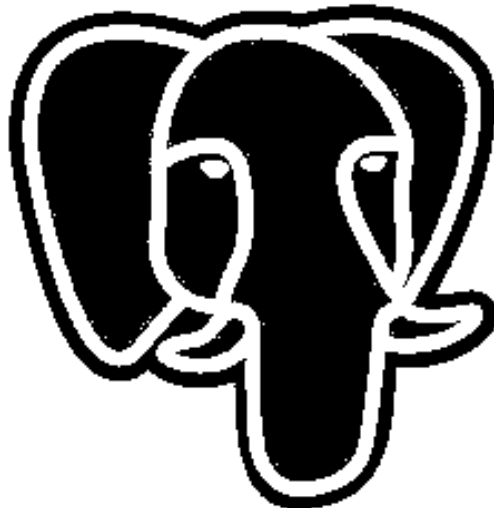


Figura 2.3.1 Logo de PostgreSQL

2.4 Características, ventajas y desventajas de la metodología de RUP y lenguaje UML

Hoy en día existe la necesidad de desarrollar software de alta calidad, las empresas desarrolladoras se preocupan por tener mejores procesos y obtener desarrollos en el menor tiempo posible. Es aquí donde entra la utilización de métodos para el desarrollo de sistemas, como RUP y el lenguaje de modelado UML.

RUP y UML ofrecen incrementos potenciales en la productividad y calidad en el desarrollo de sistemas. La utilización y administración con éxito de estas técnicas, pueden dar lugar a beneficios importantes para todas las personas involucradas en el desarrollo de sistemas.

RUP (Rational Unified Process, Proceso Unificado de Rational)

RUP es un conjunto de actividades y responsabilidades necesarias para transformar los requisitos de un sistema de software, provee de una disciplina asignando tareas y responsabilidades a los que participan en el desarrollo de un sistema orientado a objetos, el principal objetivo de RUP es asegurar la producción de software de alta calidad que satisfaga las necesidades de los usuarios.

Esta metodología esta basada en una integración de tres metodologías, Ivar Jacobs, Grady Booch y James Rumbaugh.

¹¹RUP está en continua actualización y mejoramiento, basado en experiencias ya que provee las mejores prácticas, es una guía que muestra cómo utilizar eficientemente el Lenguaje Unificado de Modelado (UML). No es rígido, es flexible, ya que es un *proceso* configurable y adaptable a cualquier desarrollo de software.

¹¹ Jiménez Luna Silvia y Soriano Monzalvo Claudia Elvira. "UML y RUP como elementos clave en el desarrollo de sistemas orientados a objetos". Tesis Licenciatura [Licenciado en Informática]. México DF. Universidad Nacional Autónoma de México. Facultad de Contaduría y Administración. 2002. 333 p.

UML (Unified Modeling Language, Lenguaje Unificado de Modelado)

Es un *lenguaje de modelado* y no un método. Permite mostrar el diseño y los requerimientos de un sistema, creando modelos precisos de desarrollo, lo que permite mejorar la comunicación con los usuarios y el equipo de trabajo, ya que es una herramienta útil para generar documentación detallada de la arquitectura del sistema a desarrollar.

RUP posee tres características que lo definen:

- Dirigido por casos de uso: La interacción de alguien o algo, ya sean personas u otros sistemas, con el sistema que se desarrolla. Un caso de uso es un requisito funcional y todo el conjunto de casos de uso describe la funcionalidad total del sistema.
Los casos de uso no sólo inician el proceso de desarrollo, también proporcionan un orden, a través de un serie de flujos de trabajo.
- Centrado en la arquitectura: Describe mediante diferentes vistas la arquitectura del sistema, permite percibir si el sistema es como lo desea el usuario o cliente, ya que incluye la plataforma, lo que proporciona un marco de trabajo para el diseño completo del sistema
- Iterativo e incremental: A través de esta característica, RUP divide de manera práctica el desarrollo de un sistema en partes más pequeñas o mini proyectos, cada uno de ellos es una iteración que resulta en un incremento, es decir, las iteraciones hacen referencia a los flujos de trabajo y los incrementos al crecimiento del producto.

El ciclo de vida está compuesto por fases y cada una de estas fases está compuesto por un número de iteraciones e incrementos que nos conducen a los siguientes criterios:

Fase de inicio

El criterio esencial es la viabilidad del proyecto, las actividades que se llevan a cabo son:

- Identificación y la reducción de riesgos para la viabilidad del desarrollo del sistema.
- Creación de una arquitectura candidata, a partir de ciertos requisitos
- Estimación de costos, esfuerzo, calendario y calidad del producto.

Fase de elaboración

El criterio esencial es la capacidad de construir el sistema dentro de un marco de trabajo, para lograr esto se debe:

- Identificar y reducir los riesgos que afectan de manera significativa la construcción del sistema
- Especificación de la mayoría de los casos de uso que representan la funcionalidad que ha de desarrollarse
- Realizar una estimación para justificar la inversión

Fase de Construcción

El criterio esencial es un sistema con operatividad inicial en el entorno del usuario, para ello se debe utilizar:

- Una serie de iteraciones con incrementos y entregas periódicas.

Fase de transición

El criterio esencial es un sistema que alcanza una operatividad final, que se logra a través de:

- La modificación del producto para subsanar problemas que no se identificaron en fases anteriores.
- La corrección de defectos.

RUP define disciplinas. Las disciplinas conllevan el flujo de trabajo, los cuales son una secuencia de pasos para la culminación de cada disciplina.

- Modelado de negocios: Consiste en entender el negocio
- Requerimientos: Define que debe de hacer el sistema
- Análisis y diseño: Como se realizan los requerimientos
- Implementación: Consiste en construir el sistema
- Pruebas: Lleva a cabo verificación, validación y retroalimentación
- Distribución: Consiste en la entrega del producto
- Administración del proyecto: define actividades para planear, presupuestar, asignar recursos y darle seguimiento al proyecto
- Administración de cambios y configuración: Consiste en realizar nuevas versiones y actualizaciones.
- Ambiente: En esta disciplina se realiza el mantenimiento.

En la figura 2.4.1 se representa la cantidad de tiempo que se invierte en cada disciplina. En iteraciones tempranas se invierte más tiempo en los requerimientos mientras que en las ultimas iteraciones se dedica más tiempo en la realización del proyecto.

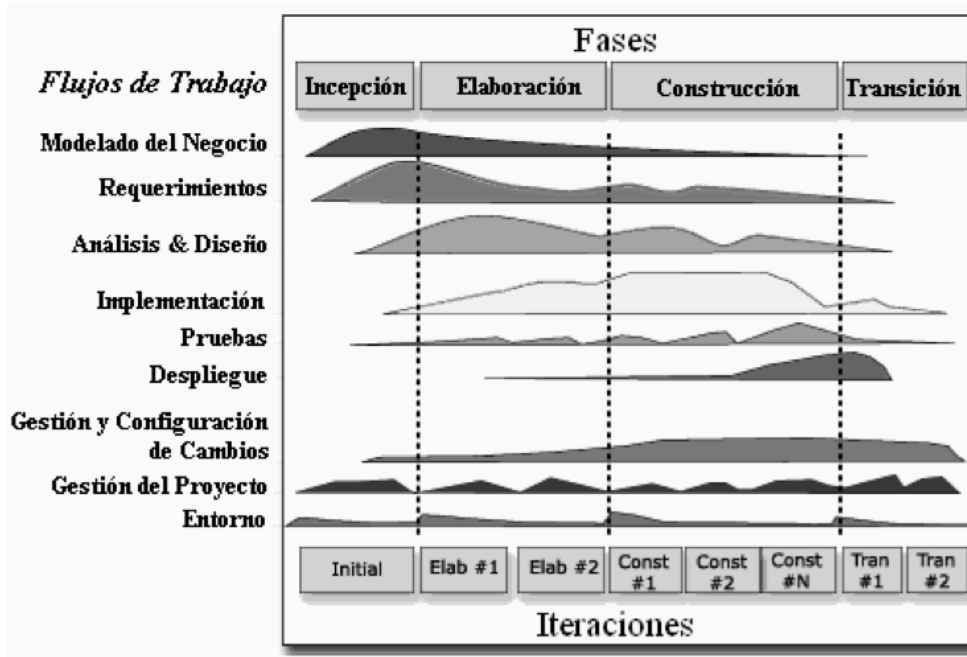


Figura 2.4.1 Disciplinas, fases e iteraciones del RUP¹²

Diagramas UML que se utilizan en RUP

El proceso de desarrollo de software debe ofrecer un conjunto de modelos que permitan expresar el producto desde cada una de las perspectivas de interés. Es aquí donde UML recomienda la utilización de nueve diagramas para representar las distintas vistas de un sistema.

A continuación se describe cada uno de éstos.

- a) Diagrama de casos de uso: Sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o sistemas. Ayuda a organizar y visualizar los requisitos del sistema.

¹²Philippe Kruchten, The Rational Unified Process An Introduction, Second Edition, Marzo 14 2000

- b) Diagrama de clases: muestra las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí.
- c) Diagrama de objetos: Contiene a los objetos y sus relaciones en un momento dado del tiempo. Puede ser considerado un caso especial de un diagrama de clases o de un diagrama de colaboraciones.
- d) Diagramas de comportamiento, dentro de estos diagramas se encuentran:
 - Diagrama de estados: Representa todos los estados que puede tomar un objeto particular durante su existencia y cómo va cambiando el estado del objeto, ya sea por el paso del tiempo, los mensajes recibidos, errores encontrados o condiciones verdaderas, también representa los eventos que producen dichos cambios de estado de los objetos de una clase.
 - Diagrama de actividades: Simplifica el Diagrama de Estados modelando el comportamiento mediante flujos de actividades.
 - Diagramas de interacción: Es una representación gráfica de interacciones entre objetos. Hay dos tipos de diagramas según la interacción que enfatizan:
 - Diagrama de secuencia: Muestra interacción entre objetos organizadas en secuencia temporal. En particular muestra los objetos participantes de la interacción y la secuencia de mensajes intercambiados.
 - Diagrama de colaboración: Muestra la interacción entre objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- e) Diagramas de implementación
 - Diagrama de componentes: Muestra la organización y la dependencia entre un conjunto de componentes.

- Diagrama de despliegue: Modela los elementos tangibles como: computadoras personales, servidores, impresora, procesadores, entre otros dispositivos, que pertenecen a los componentes físicos o hardware del sistema.

Características de RUP

- Utiliza UML
- Gramática bien definida
- Terminología usada para la definición de los procesos

Ventajas de la metodología RUP

- RUP ha ido madurando con el tiempo
- El uso de UML hace que se apegue a estándares de la industria
- Las bases se adaptan a la organización
- Existen herramientas que ayudan a implementar RUP en todas las áreas de una empresa
- Define actividades, roles y responsabilidades desde el jefe de proyectos hasta los analistas y desde los desarrolladores hasta el equipo de pruebas.

Desventajas de la metodología RUP

- En empresas donde hay sistemas híbridos donde no todo encaja en metodologías orientadas a objetos UML no es efectivo
- La sintaxis de modelación requiere de notaciones que no poseen los desarrolladores promedio
- Costosa. Comprar las herramientas y entrenar a los equipos.
- Para las empresas implica, licenciamiento, entrenamiento y consultoría
- Ciclo de vida completo

- No contempla el retiro del software

En la figura 2.4.2 se muestra el logo de RUP



Figura 2.4.2 Logo del modelo RUP

CAPÍTULO 3 ANÁLISIS Y PLANTEAMIENTO DEL PROBLEMA

3.1 Análisis del problema

La empresa no cuenta con algún sistema computacional o informático para la administración de su estacionamiento, por lo que se tendrá que investigar, recopilar información y diseñar el sistema para su desarrollo.

El tipo de estacionamiento es de autoservicio, por lo que el cliente tendrá que estacionar su automóvil dentro del establecimiento, con la condición de que deberá dejarlo en el cajón que le sea asignado.

En base a la experiencia de la empresa se han identificado ciertos problemas que le gustaría evitar.

A continuación se describen los problemas que se ha presentado dentro del estacionamiento.

En la figura 3.1.1, podemos observar que el cliente llega al estacionamiento público, pasa directamente con el empleado para que le proporcione su ticket de entrada con su respectiva hora de entrada, comienza el servicio de estacionamiento, posteriormente va en busca de un lugar para poder aparcar pero sin ningún éxito y tras varias rondas al estacionamiento no encuentra lugar disponible, ya sea porque ya hay alguien esperando el lugar que será desocupado o porque el lugar desocupado está muy angosto debido a que los autos estacionados a un lado están muy anchos.

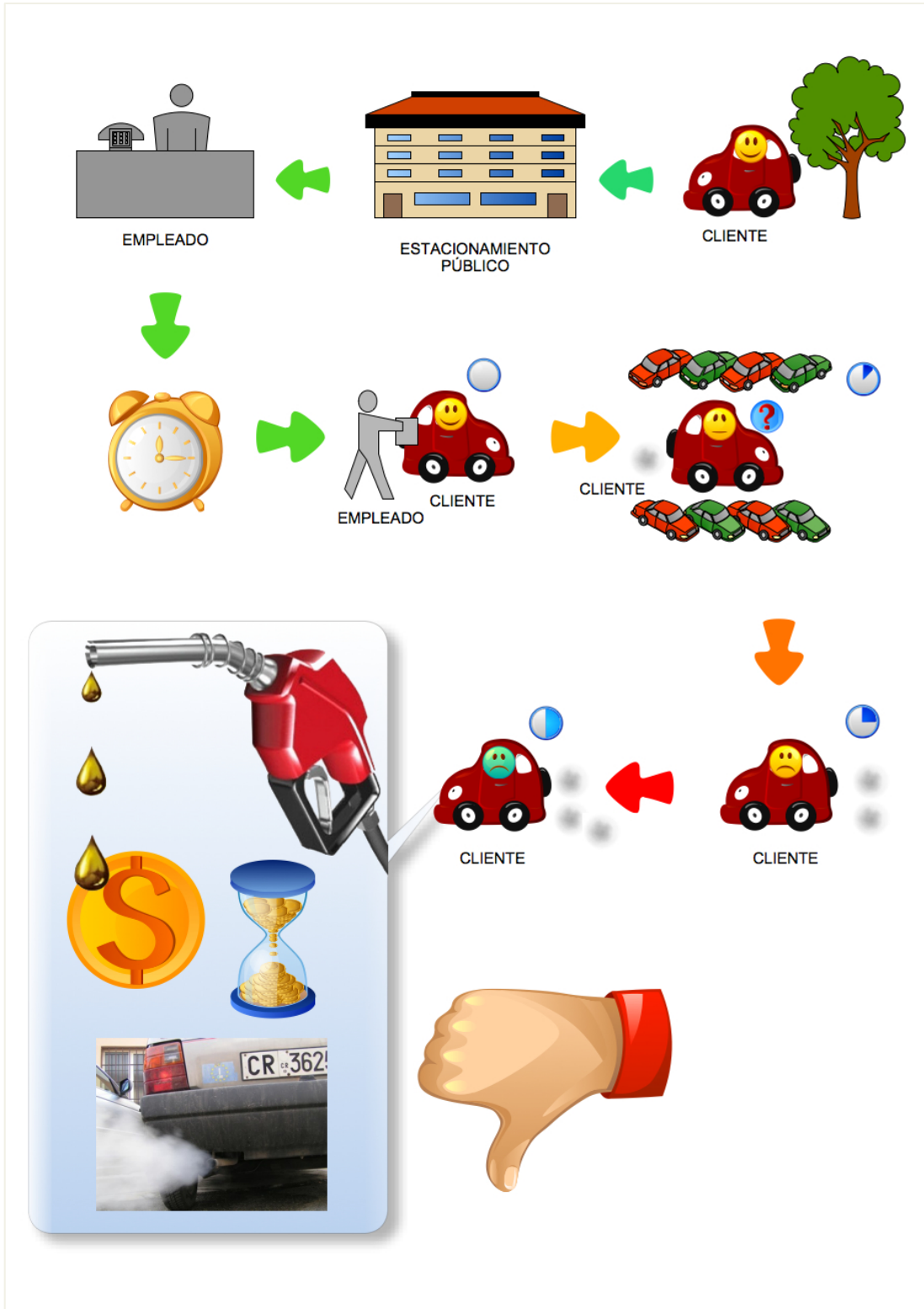


Figura 3.1.1 Problemática de la falta de administración de los cajones de estacionamiento

En este caso el cliente al no poder obtener pronto un lugar de estacionamiento se puede llegar a desesperar, frustrar o incluso enojar, debido a que al no encontrar rápidamente un lugar se pierde tiempo y dinero, ya que el servicio ya esta siendo cobrado, sin mencionar el gasto de gasolina y la creciente emisión de gases tóxicos que provocan los autos en la misma situación.

Esta problemática se presenta por la falta de administración en la cantidad de lugares que cuenta el estacionamiento o porque la misma empresa provoca una demanda excedente.

Otra situación que es común en nuestra sociedad es la falta de ética en el uso de lugares reservados que son destinados a las personas con capacidades diferentes. Los clientes al no poder encontrar un lugar o por la rapidez de estacionarlo dejan el auto en cajones que son destinados a estas personas, provocando así que las personas que tienen derecho a ocupar el lugar tengan que esperar a que se desocupe o bien, ir en busca de otro cajón el cual muy probablemente se encuentre más alejado de las salidas o rampas construidas para su comodidad. VÉR figura 3.1.2



Figura 3.1.2 Automóvil mal estacionado en el lugar destinado para personas con capacidades diferentes

El estacionamiento de la empresa cuenta con varios tamaños de cajones, esto se hizo considerando que no todos los autos son del mismo tamaño, las camionetas evidentemente ocupan más espacio. Otra de las razones de los distintos tamaños de cajones es para que existan cajones destinados al aparcamiento de motocicletas, ya que cuando ésta se estaciona ocupa el cajón que ocuparía un auto.



Figura 3.1.3 Cajones para motocicletas

Otro problema a considerar en este tipo de estacionamientos, es la incertidumbre de que se falsifiquen boletos o existan fugas de dinero debido a que sólo se toma como registro el ticket que se proporciona.

Cabe mencionar que el dueño de la empresa sólo permitirá el acceso a las personas que conducen el auto o que han sido registradas en el estacionamiento, por lo que las personas acompañantes esperan afuera del establecimiento.

Esto es por seguridad, ya que existe el robo de partes de automóviles, asaltos dentro del establecimiento e incluso accidentes. Ver figura 3.1.4



Figura 3.1.4 Robo de automóviles en estacionamientos públicos

3.2 Recopilación y análisis de la información

La recopilación de información es fundamental para el desarrollo del sistema a implementar, ya que de esto dependen los posibles módulos a desarrollar y que la empresa este de acuerdo con las actividades que realizará el sistema.

Para que haya una buena administración del estacionamiento es necesario tener un buen registro de los clientes, de los empleados y sus sueldos, de los autos que serán custodiados, de los pagos que se deben realizar para el mantenimiento del establecimiento, los pagos de servicios que son necesarios de empresas externas, entre otros. A continuación se describen elementos que son necesarios para la administración del estacionamiento.

Formulario cliente:

Para el registro de los clientes es necesario llenar un formulario. El cual sirve para comprobar la propiedad del vehículo aparcado, por si existe algún percance o para encuestas que necesite el dueño de la empresa. Este formulario va orientado a los clientes que contratan el servicio de pensión. Ver figura 3.2.1

FECHA REGISTRO				<input type="text"/>
DATOS CLIENTE				
NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO		
<input type="text"/>	<input type="text"/>	<input type="text"/>		
ENTIDAD FEDERATIVA	DELEGACION o MUNICIPIO	COLONIA		
<input type="text"/>	<input type="text"/>	<input type="text"/>		
CALLE	NUM. EXTERIOR	NUM.INTERIOR	C.P.	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
E-MAIL	TELEFONO	TEL. CELULAR		
<input type="text"/>	<input type="text"/>	<input type="text"/>		

Figura 3.2.1 Formulario cliente

Formulario vehículo:

Este registro es necesario para poder identificar los autos que se encuentran aparcados dentro del estacionamiento y a sus dueños. Este formulario va orientado a los clientes que piden el servicio de pensión por lo que todos los campos deber ser debidamente llenados. Ver figura 3.2.2

DATOS DEL VEHICULO

NOMBRE DEL DUEÑO	APELLIDO PATERNO	APELLIDO MATERNO
<input type="text"/>	<input type="text"/>	<input type="text"/>
PLACAS	MARCA	COLOR
<input type="text"/>	<input type="text"/>	<input type="text"/>

TIPO DE VEHICULO

AUTOMOVIL

CAMIONETA

MOTOCICLETA

ANOTACIONES

Figura 3.2.2 Formulario vehículo

El campo Anotaciones ayuda a tener registrado si el auto esta en mal estado cuando entro al estacionamiento.

Formulario empleado:

El registro de los empleados es muy importante, ya que estos son los que se encargaran de resguardo de los autom6viles, el acceso al establecimiento y el cobro de servicios, por lo que es necesario tener su informaci3n personal y la de un conocido como referencia. Ver Figura 3.2.3 Formulario empleado.

FECHA CONTRATACIÓN				<input type="text"/>
DATOS EMPLEADO				
NOMBRE(S)	APELLIDO PATERNO	APELLIDO MATERNO		
<input type="text"/>	<input type="text"/>	<input type="text"/>		
ENTIDAD FEDERATIVA	DELEGACION o MUNICIPIO	COLONIA		
<input type="text"/>	<input type="text"/>	<input type="text"/>		
CALLE	NUM. EXTERIOR	NUM.INTERIOR	C.P.	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
E-MAIL	TELEFONO	TEL. CELULAR		
<input type="text"/>	<input type="text"/>	<input type="text"/>		
CURP				
<input type="text"/>				
DATOS PERSONA DE REFERENCIA				
NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO		
<input type="text"/>	<input type="text"/>	<input type="text"/>		
ENTIDAD FEDERATIVA	DELEGACION o MUNICIPIO	COLONIA		
<input type="text"/>	<input type="text"/>	<input type="text"/>		
CALLE	NUM. EXTERIOR	NUM.INTERIOR	C.P.	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
E-MAIL	TELEFONO	TEL. CELULAR		
<input type="text"/>	<input type="text"/>	<input type="text"/>		

Figura 3.2.3 Formulario empleado

RFC:

El Registro Federal de Contribuyentes (RFC) se refiere a una clave que tiene toda persona física en México para realizar alguna actividad lícita que este obligada a pagar impuestos a toda persona moral sin excepción. A estas personas se les llama contribuyentes. Vea figura 3.2.4



Figura 3.2.4 RFC

Facturas:

La empresa proporciona facturas, las cuales los clientes requieren para aclarar sus impuestos. Este documento requiere datos que se especifica en el artículo 29-A del Código Fiscal de la Federación. Ver figura 3.2.5


Camilo Cruz Estrada			FACTURA	
RFC: CACE-830914-602			Folio: A 100	
ESPERANZA No. 16 COL. JACARANDAS CP 51530 TEL: 55 49 30 01				
DELEGACIÓN IZTACALCO MÉXICO DF.				
México DF a _____				
NOMBRE: _____				
DOMICILIO: _____			CP: _____	
COLONIA: _____		POBLACIÓN: _____		RFC CLIENTE: _____
CANTIDAD	DESCRIPCIÓN	P. UNITARIO	IMPORTE	
Pedimento No: _____				
Fecha Pedimento: _____				
Aduana: _____				
TOTAL CON LETRA			SUBTOTAL	
			IVA	
			TOTAL	
 <p>IMPRESO POR OSCAR JIMENEZ ESTRADA, JIE0700110147 CALLE LERDO No. 50 COL. GUERRERO, DEL MIGUEL HIDALGO, CP 08000, MÉXICO DF. TEL. 55899251. FECHA DE INCLUSIÓN DE LA AUTORIZACION EN LA PAGINA DE INTERNET DEL SAT: 8 DE MARZO DEL 2002. COMPROBANTES ELABORADOS EL 8 DE MARZO DEL 2004. LA REPRODUCCIÓN NO AUTORIZADA DE ESTE COMPROBANTE CONSTITUYE UN DELITO EN LOS TERMINOS DE LAS DISPOSICIONES FISCALES. VIGENCIA DEL 8 DE MARZO DE 2004 AL 8 DE MARZO DE 2006</p>				
NÚMERO DE APROBACIÓN DEL SISTEMA DE CONTROL DE IMPRESORES AUTORIZADOS: 223720				
"EFECTOS FISCALES AL PAGO"			"PAGO EN UNA SOLA EXHIBICIÓN"	

Figura 3.2.5 Ejemplo de factura

Los requisitos para las facturas que establece el artículo 29-A del Código Fiscal de la Federación son:

1. El nombre impreso, denominación o razón social, domicilio fiscal y clave del Registro Federal de Contribuyentes de quien los expida. Sí se tiene más de un local o establecimiento, se debe señalar el domicilio del local o establecimiento en el que se expidan los comprobantes.
2. Contener impreso el número de folio.
3. Lugar y fecha de expedición.
4. Clave del Registro Federal de Contribuyentes de la persona a favor de quien se expida.
5. Cantidad y clase de mercancías o descripción del servicio que amparen.

6. Valor unitario consignado en número e importe total señalado en número o en letra, así como el monto de los impuestos que en los términos de las disposiciones fiscales deban trasladarse desglosado por tasa de impuesto, en su caso.
7. Número y fecha del documento aduanero, así como la aduana por la cual se realizó la importación, tratándose de ventas de primera mano de mercancías de importación.
8. Fecha de impresión y datos de identificación del impresor autorizado.
9. Cuando se trate de enajenación de ganado, la reproducción del hierro de marcar de dicho ganado siempre que se trate de aquel que deba ser marcado.
10. Vigencia. Los comprobantes podrán ser utilizados en un plazo máximo de dos años, contados a partir de su fecha de impresión, este plazo se puede prorrogar cuando se cubran los requisitos que señale la autoridad fiscal de acuerdo a reglas de carácter general que al efecto se expidan. Transcurrido dicho plazo, se considerará que el comprobante queda sin efectos para las deducciones o acreditamientos previstos en las leyes fiscales. Respecto a la vigencia de dos años de los comprobantes y de conformidad con la regla I.2.10.17. de la Resolución Miscelánea Fiscal para 2010, se puede optar por considerar sólo el mes y año como fecha de impresión y la vigencia se calcula a partir del primer día del mes que se imprima en el comprobante y la numeración de los comprobantes continuará siendo consecutiva, incluyendo la de los comprobantes cancelados al término de su vigencia.
11. Señalar en forma expresa si el pago se hace en una exhibición.

También debe contener los siguientes requisitos establecidos en el artículo 39 del Reglamento del Código Fiscal de la Federación publicado en el Diario Oficial de la Federación el día 07 de diciembre de 2009.

- La cédula de identificación fiscal la cual en el caso de personas físicas podrá o no contener la CURP, reproducida en 2.75 cm. por 5 cm. con una resolución de 133 líneas/ 1200 dpi. Sobre la impresión de la cédula, no podrá efectuarse anotación alguna que impida su lectura.
- La leyenda: "la reproducción no autorizada de este comprobante constituye un delito en los términos de las disposiciones fiscales", con letra no menor de 3 puntos.
- El Registro Federal de Contribuyentes y nombre del impresor, así como la fecha en que se incluyó la autorización correspondiente en la página de Internet del Servicio de Administración Tributaria, con letra no menor de 3 puntos.
- El número de aprobación asignado por el **Sistema de integral de comprobantes fiscales**.
- En el caso de personas físicas que tributen conforme al Régimen de Actividades Empresariales y Profesionales, así como en el Régimen Intermedio; además, debe imprimir la leyenda "Efectos fiscales al pago" de conformidad con lo señalado en la fracción III del Art.133 de la Ley del Impuesto sobre la Renta.
- En caso de que sean emitidos por personas morales del Régimen Simplificado, los comprobantes deberán llevar además la leyenda: "Contribuyente del Régimen de Transparencia".

Boleto de estacionamiento:

La empresa expide boletos a los clientes que han dejado su automóvil en el establecimiento. Ver figura 3.2.6

El boleto debe tener impreso varios datos que se especifica en el Artículo 20 del Reglamento de Estacionamientos Públicos del Distrito Federal publicado en el Diario Oficial de la Federación el 27 de marzo de 1991, el cual se muestra a continuación:

¹³**Artículo 20.-** El boleto que entregue el estacionamiento al usuario deberá contener los siguientes datos:

- I.- Nombre o razón social y domicilio del prestador del servicio de estacionamiento;
- II.- Clave del Registro Federal de Contribuyentes;
- III.- Los números telefónicos para reportar quejas, tanto del propio estacionamiento, como los que para ese efecto establezca el Departamento del Distrito Federal;
- IV.- La clasificación del estacionamiento y, de acuerdo con ello, la tarifa aplicable;
- V.- Número de boleto;
- VI.- Forma en la que se responderá por los daños que hayan sufrido los vehículos durante el tiempo de guarda;
- VII.- Espacio para asentar la hora de entrada;
- VIII.- Espacio para apuntar la hora de salida;
- IX.- Espacio para anotar el número de placa.

¹³ Reglamento de estacionamientos públicos del Distrito Federal pg.6 Consultado en: <http://www.coyoacan.df.gob.mx/transparencia/art14/i/reglamentos/trim1-2009/ESTACIONAMIENTOS.pdf>



Desarrollos A y F Las Ánimas, S.A. de C.V.

R.F.C. DAF950320-B80

Carretera Federal Lázaro Cárdenas No. 271
Fracc. Las Ánimas, Xalapa, Ver. C.P. 91190

Calle Martín Mondalde No. 1422
Col. del Valle, México, D.F. C.P. 03100

Horario: 6:00 a 20:00 hrs. De Lunes a Sábado

MARCA: _____

PLACAS: _____

ENTRADA: _____

SALIDA: _____

IMPORTE \$ _____

Nº [REDACTED]

*ESTE COMPROBANTE NO TIENE VALOR FISCAL

*NO SE ENTREGARÁ NINGUNA UNIDAD SIN EL COMPROBANTE RESPECTIVO

ESTA EMPRESA NO SE HACE RESPONSABLE DE GOLPES OCASIONADOS POR TERCEROS U OTROS VEHÍCULOS. DAÑOS OCASIONADOS POR LA NATURALEZA O QUE VENGAN DEL EXTERIOR, POR INCENDIO, POR ROBO TOTAL O PARCIAL DE LA UNIDAD, ASÍ COMO POR OBJETOS DE VALOR DEJADOS DENTRO DE LA UNIDAD SIN QUE LA ADMINISTRACIÓN TENGA CONOCIMIENTO.

Figura 3.2.6 Ejemplo de un boleto de estacionamiento

Cuentas por pagar:

La empresa tiene que realizar pagos a otras empresas por diferentes servicios como telefonía, electricidad, agua, etc. En la figura 3.2.7 se muestra un ejemplo de cuentas por pagar a diferentes empresas.

NUMERO	215000				
NOMBRE	COMISION FEDERAL DE ELECTRICIDAD				
CONTRAREC.	FECHA	FACTURA		CANTIDAD	
	25/06/20		\$		109.00
	25/06/20		\$		676.00
		TOTAL	\$		785.00
NUMERO	215000				
NOMBRE	TELEFONOS DE MEXICO SA DE CV				
CONTRAREC.	FECHA	FACTURA		CANTIDAD	
	23/06/20		\$		8,592.00
	23/06/20		\$		1,366.00
		TOTAL	\$		9,958.00
NUMERO	215000				
NOMBRE	NELIDA MORENO ESPARZA				
CONTRAREC.	FECHA	FACTURA		CANTIDAD	
	MAYO		\$		2,850.00
		TOTAL	\$		2,850.00
NUMERO	215000				
NOMBRE	SECRETARIA DE FINANZAS DEL ESTADO DE JALISCO				
CONTRAREC.	FECHA	FACTURA		CANTIDAD	
	MAYO		\$		4,087.58
		TOTAL	\$		4,087.58
TOTAL			\$		17,680.58
			\$		17,680.58

Figura 3.2.7 Cuentas por pagar

3.3 Requerimientos generales y particulares de la aplicación

Este proceso tiene por objetivo comprender la necesidad del cliente, la innovación tecnológica que se busca, y colocarla en un marco descriptivo, definirlo en términos técnicos y/o científicos.

Se requiere un sistema, con interfaz amigable y moderna que le permita al operador administrar los lugares de estacionamiento y llevar un registro de los pagos realizados por los servicios de estacionamiento por hora, por pensión u otro servicio que ofrezca el estacionamiento.

A continuación se describe la solución propuesta, así como los requerimientos para la elaboración del Sistema para administración de un estacionamiento público.

Requerimientos generales

El sistema debe contar con los siguientes requerimientos:

- El sistema deberá ser instalado en cualquier computadora por alguna falla
- Desarrollar el sistema en un ambiente gráfico y amigable como son los ambientes de Microsoft para que no presente ningún inconveniente por parte de los usuarios.
- El administrador del sistema será el único que podrá tener acceso directo a la base de datos.
- Toda la información almacenada deberá ser validada por el sistema mismo.
- El sistema será desarrollado con herramientas de Microsoft por considerar que hay mayor soporte hacia estas, además de que la empresa posee licencias necesarias.
- El acceso al sistema será por medio de un nombre de usuario y contraseña para cada usuario, mismo que serán asignado por el administrador del sistema.

Requerimientos Particulares

El sistema debe contar con un menú que presente los módulos que lo integran, dicho menú debe contener los siguientes módulos.

➤ Disposición de Cajones:

- Consultar de manera gráfica la localización de todos los cajones (mapa del estacionamiento),
- Identificar qué tipo de vehículo puede contener el cajón: personas con capacidades diferentes, automóvil, camioneta o motocicleta.
- Verificar su disponibilidad: ocupado, disponible o reservado.

➤ Clientes:

El operador debe ser capaz de:

- Registrar los clientes que contratan el servicio de pensión
- Realizar el borrado de clientes
- Realizar la búsqueda de algún cliente.
- Modificar el registro de un cliente

➤ Servicios:

El operador debe ser capaz de:

- Consultar los tipos de servicio que se ofrecen
- Agregar más servicios
- Borrar servicios
- Modificar el costo del servicio

➤ Contabilidad:

El operador debe ser capaz de:

- Realizar el corte de caja
- Consultar las ganancias del estacionamiento
- Consultar los pagos que se realizaron
- Consultar los cobros que se realizaron
- Consultar la cantidad de vehículos que contrataron el servicio de estacionamiento.
- Consultar la cantidad de vehículos que contrataron cada uno de los servicios adicionales.
- Consultar la cantidad de vehículos que contrataron pensión.
- Consultar si algún vehículo se ha dejado fuera del tiempo de contratación.
- Registrar pagos de los servicios externos u otros pagos

➤ Facturación:

El operador debe ser capaz de:

- Registrar las facturas que se expidieron
- Consultar las facturas que se expidieron
- Cancelar las facturas si existe alguna equivocación

➤ Seguridad:

El operador debe ser capaz de:

- Dar permisos a funciones del sistema a usuarios que utilizan el sistema.
- Denegar permisos de funciones del sistema a usuarios que utilizan el sistema.

➤ Empleados:

El operador debe ser capaz de:

- Registrar los nuevos empleados
- Consultar la información del empleado
- Modificar la información del empleado
- Eliminar el registro de un empleado

En la Figura 3.3.1 Requerimiento particular, Menú, se muestra lo que espera el dueño de la empresa como menú principal del sistema.

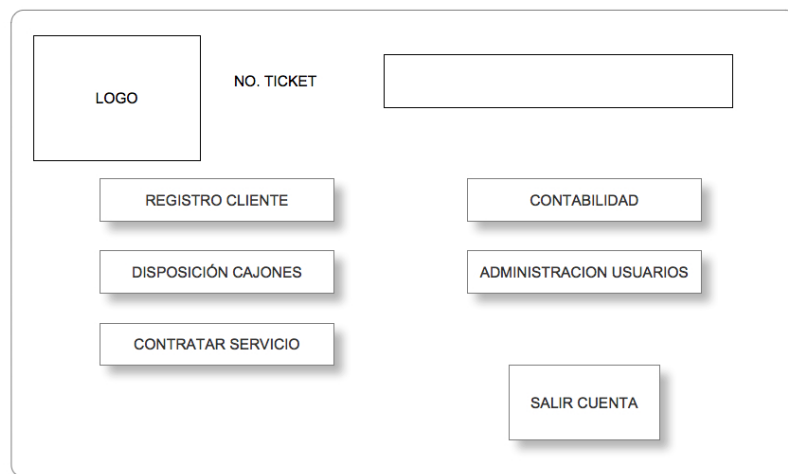


Figura 3.3.1 Requerimiento particular, Menú

Requerimientos de Seguridad

En esta sección se lista características necesarias en cuanto a seguridad de la información y avisos necesarios al operador.

- Seguridad en el acceso a la Información por el usuario
- Navegación del usuario de acuerdo a sus permisos
- Bitácora para el almacenamiento de movimientos cancelados o modificados.
- Conocer el usuario que realizo el servicio.
- Tener un registro de los empleados con la información necesaria de estos.
- Tener respaldos de la información.
- Aviso de automóviles que han estado en el estacionamiento por más de 30 días sin contar con el servicio de pensión como se especifica en el Artículo 25 del Reglamento de estacionamientos públicos del Distrito Federal publicado en el Diario Oficial de la Federación el 27 de marzo de 1991.

3.4 Planteamiento de la solución y posibles módulos

El sistema en base a la información recabada y atendiendo a la necesidad de la empresa se considera que el sistema deberá comprender los módulos de:

➤ **Acceso**

Este comprende el inicio del sistema donde el usuario obligatoriamente teclea su nombre y contraseña, éste modulo está ligado al modulo “Usuario”. Se encargará del acceso al sistema.

➤ **Usuarios**

En este módulo se administraran los usuarios que podrán utilizar el sistema: creación, modificación, borrado, búsqueda y además se especificará el tipo de permisos que tiene para la modificación o acceso a datos en el sistema, como la parte de contabilidad. Este modulo está ligado al modulo empleado.

➤ **Empleados**

En este modulo se administrarán todos las personas que trabajan en el estacionamiento.

Se hace registro de todos los datos que se encuentran en el formulario de empleado.

Se podrá hacer modificación de datos o eliminación del registro del empleado.

Consultar la lista completa de los empleados.

➤ **Clientes**

Este módulo se encargará de administrar los registros de los clientes como el registro con los datos del “formulario de cliente”, la modificación de datos, el borrado de registros y búsqueda de clientes.

➤ **Servicios**

En éste modulo administrarán los servicios que se ofrecen en la empresa: se registraran los nuevos servicios, se modificara la información de los servicios: costo o se podrá eliminar servicios.

➤ **Cajones**

Este módulo se encarga de desplegar en pantalla los espacios disponibles en el estacionamiento además de poder seleccionar el espacio que será ocupado por el cliente. Es modulo estará diseñado de forma visual (mapa del estacionamiento) y estará representado por símbolos para el tamaño del cajón, el tipo de cajón y su disponibilidad. Véa figura 3.4.1

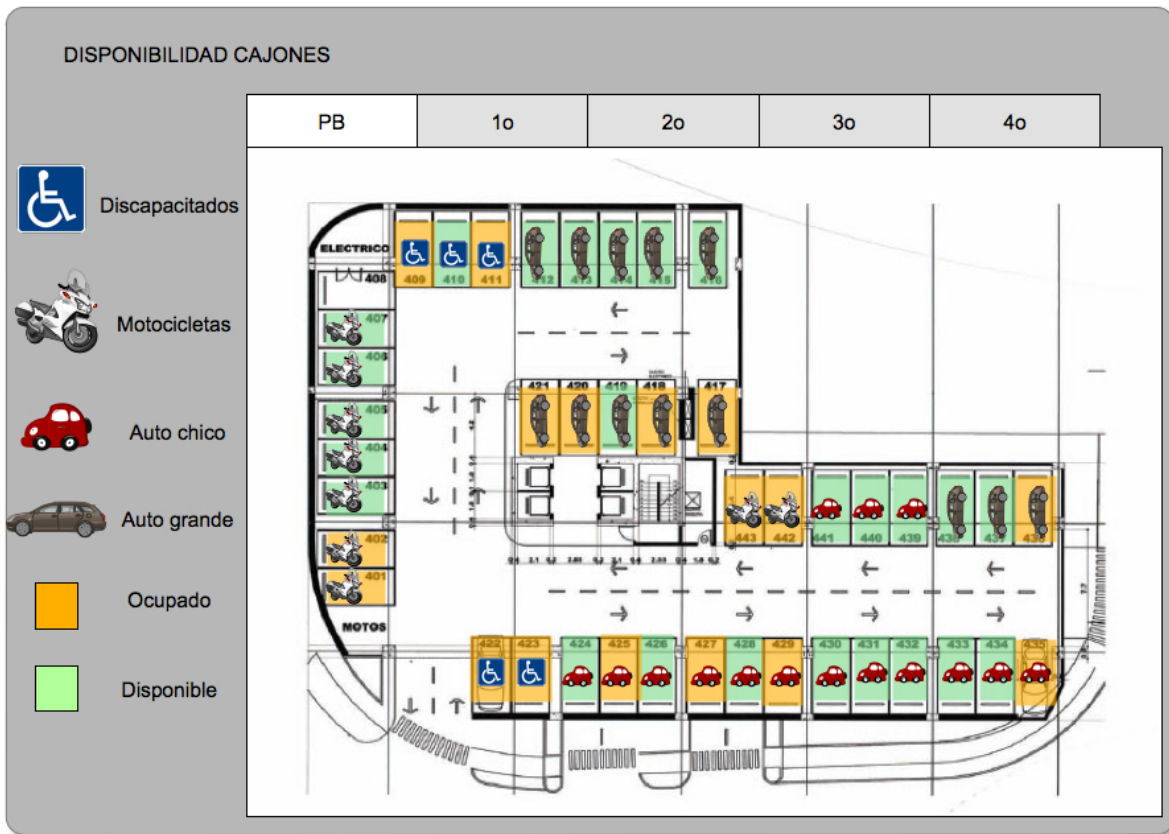


Figura 3.4.1 Modulo cajones

Caja

Éste modulo se encarga de administrar los registros de pagos o cobros que se han hecho, como el cobro de horas de estacionamiento, cobro por pensión, pago de algún servicio externo o devolución.

Corte

Éste modulo se encarga de calcular las ganancias que se tuvieron día con día.

➤ **Vehículos**

Este modulo se encarga de la administración de los vehículos que se encuentran dentro del estacionamiento. Permite consultar los datos de los vehículos, el tiempo que llevan estacionados.

➤ **Facturación**

Éste modulo se encargara de administrar las facturas que se expiden: se registraran los datos necesarios para la facturación, la cancelación de facturas puesto que no se puede hacer el borrado de éstas antes de haberlas reportado y eliminación después de un periodo de tiempo establecido.

➤ **Estadísticas**

Éste modulo se encarga de desplegar información referente a la cantidad de clientes que hicieron uso de los distintos servicios que ofrece el estacionamiento, en un día, una semana o un mes.

Debe recordarse que esta aplicación es de uso interno de la empresa.

3.5 Justificación de la metodología y software a utilizar

Se decidió hacer la programación orientada a objetos ya que se pueden crear las distintas partes del sistemas siendo cada uno independiente de los otros, además de que si en el futuro el cliente desea implementar más funciones al sistema estos serán más sencillos de integrarlos, evitando realizar muchos cambios en la programación ya realizada del sistema.

Para la elección del back-end se analizaron dos bases de datos, ambas son de distribución y utilización libre (gratuita).

En cuanto al gestor de base de datos se muestra en la tabla 3.5.1 las diferencia entre los gestores de datos: PostgreSQL y MySQL.

Base de datos	PostgreSQL	MySQL
Sistema de gestión de bases de datos	Objeto-relacional (ORDBMS)	Relacional
Licencia	BSD	GLP
Multihilo	No	Si
Software libre	Si	Si
Lenguajes de programación con los que puede ser utilizado	C, C++, Java, Python, PHP, etc.	C, C++, Java, PHP, etc.
Implementación del estándar SQL92/SQL99	Si	No
Gran escalabilidad	Si	Si
Soporta distintos tipos de datos	Si	Si
Velocidad a la hora de realizar las operaciones	Relativamente lenta	Mayor
Rollback's	Si	No
Subconsultas	Si	No
Transacciones	Si	Si
Seguridad	Mayor	Menor en la versión libre
Capacidad de comprobar la integridad referencial	Si	No
Viable para su uso con grandes bases de datos	Si	Si
Facilidad de configuración e instalación	Menor	Mayor

Tabla 3.5.1 Tabla comparativa de bases de datos

En el diseño de software el front-end es la parte del software que interactúa con el usuario, la idea general es que se encarga de recopilar los datos del usuario. En el sistema de este tema de tesis el front-end se realizará con el lenguaje de programación C# y su visualización será por medio de Windows Forms.

Esto nos da la ventaja de desarrollar el proyecto con los objetos ya predefinidos para utilizarse en el sistema operativo Windows ayudando a enfocarse más en el desarrollo de operatividad y validación de datos, que en la parte de desarrollo gráfico.

Para fines informativos se muestra en la tabla 3.5.2 las características de Visual Studio frente a las características de PowerBuilder, el cual es una herramienta de desarrollo orientada a objetos con los que se permite el desarrollo de aplicaciones y componentes para ser ejecutadas en arquitecturas cliente/servidor, distribuidas y Web.

Power Builder	Visual Studio
<p>Soporte e Integración con estándares de la Industria como: SOA, Web Services, AJAX.</p> <p>Mejoras con respecto a usabilidad e interface de usuario</p> <p>Soporte de <i>rebuilds</i> incrementales</p> <p>Soporte a implementaciones con Oracle 10g RAC.</p> <p>Flexibilidad para migrar existentes aplicaciones o crear nuevas aplicaciones para .NET WinForms, WebForms y Smart Clients.</p> <p>Completando la Integración con .NET ofrece: Soporte de Strong Named Assemblies; acceso a clases del tipo: static, primitive y enumerative; y soporte de IIS7</p> <p>Ofrece el método más rápido de acceso, manipulación y presentación de datos a través de los DataWindow. Power Builder 11 puede presentar gráficos de columnas, barras, líneas y áreas en 3D; ofrece características de gradientes y transparencia; soporte a archivos PNG; y, soporte para la edición de estilos de las columnas basadas en Rich Text Edit.</p> <p>Soporte nativo para Oracle 11g y SQL Server 2008</p> <p>PowerBuilder 11 viene con Application Server Plug-in, que permite desplegar las aplicaciones en servidores de aplicaciones de terceros, directamente desde el IDE de PowerBuilder</p>	<p>Creación de aplicaciones que usan las últimas tecnologías web con compatibilidad mejorada para AJAX, controles web y Microsoft AJAX Library</p> <p>Creación de aplicaciones web fácilmente con una interface de diseño mejorada y compatibilidad con los estándares de la industria.</p> <p>Conexión a cualquier origen de datos gracias a LINQ, un conjunto de extensiones de lenguaje para Visual Basic y Visual C#.</p> <p>La ejecución de pruebas unitarias está integrada en Visual Studio 2008 Professional Edition.</p> <p>Diseñadores integrado para Windows Presentation Foundation que garantizan interfaces amigables para los usuarios finales. Las interfaces de usuario creadas con Windows Presentation Foundation pueden interoperar con Windows Forms.</p> <p>Diseño de aplicaciones conectadas mediante los nuevos diseñadores visuales para Windows Communications Foundation y Windows Workflow Foundation</p> <p>Creación de soluciones basada en Microsoft Office que sean seguras, escalables y de fácil mantenimiento.</p> <p>Ambiente de trabajo colaborativo entre desarrolladores y diseñadores de las aplicaciones.</p>

Tabla 3.5.2 Características de Power Builder y Visual Studio

CAPÍTULO 4 DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

4.1 Modelado del Sistema

Para conseguir desarrollar software de manera profesional y obtener un producto de calidad, es completamente necesario seguir una metodología y no abordar los problemas de manera somera, con el propósito de obtener un modelo que represente adecuadamente la solución al mismo.

El modelado es la base del desarrollo de software de calidad. Los modelos se construyen para explicar el comportamiento del sistema a desarrollar, para comprender mejor el sistema en cuestión, para controlar el riesgo y en definitiva para poder atacar problemas que, sin el modelado su resolución sería imposible, tanto desde el punto de vista de los desarrolladores como desde el punto de vista del cliente, al cual, si finalmente se le entrega el producto del desarrollo se encontrará con infinidad de problemas, desde que no se cumplen las especificaciones hasta fallos que dejen inutilizable el sistema.

Un modelo representa a un sistema de software desde una perspectiva específica, cada diagrama permite observar un aspecto distinto del sistema, por lo tanto nuestro sistema estará documentado por medio de los diagramas que proporciona el Lenguaje Unificado de Modelado (UML) y basado en la Metodología RUP.

Los diagramas de UML con los que vamos a trabajar son los siguientes:

- Diagrama de casos de uso.
- Diagrama de modelado de datos
- Diagrama de clases
- Diccionario de datos

Para el modelado de la base de datos se usará el diagrama entidad – relación.

4.1.1 Diagrama de casos de uso

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.

A continuación se presenta una parte de los casos de uso que se realizan en el sistema de administración de estacionamiento público.

En la figura 4.1.1.1 se muestra de forma general la interacción de los usuarios con el sistema de administración de estacionamiento público

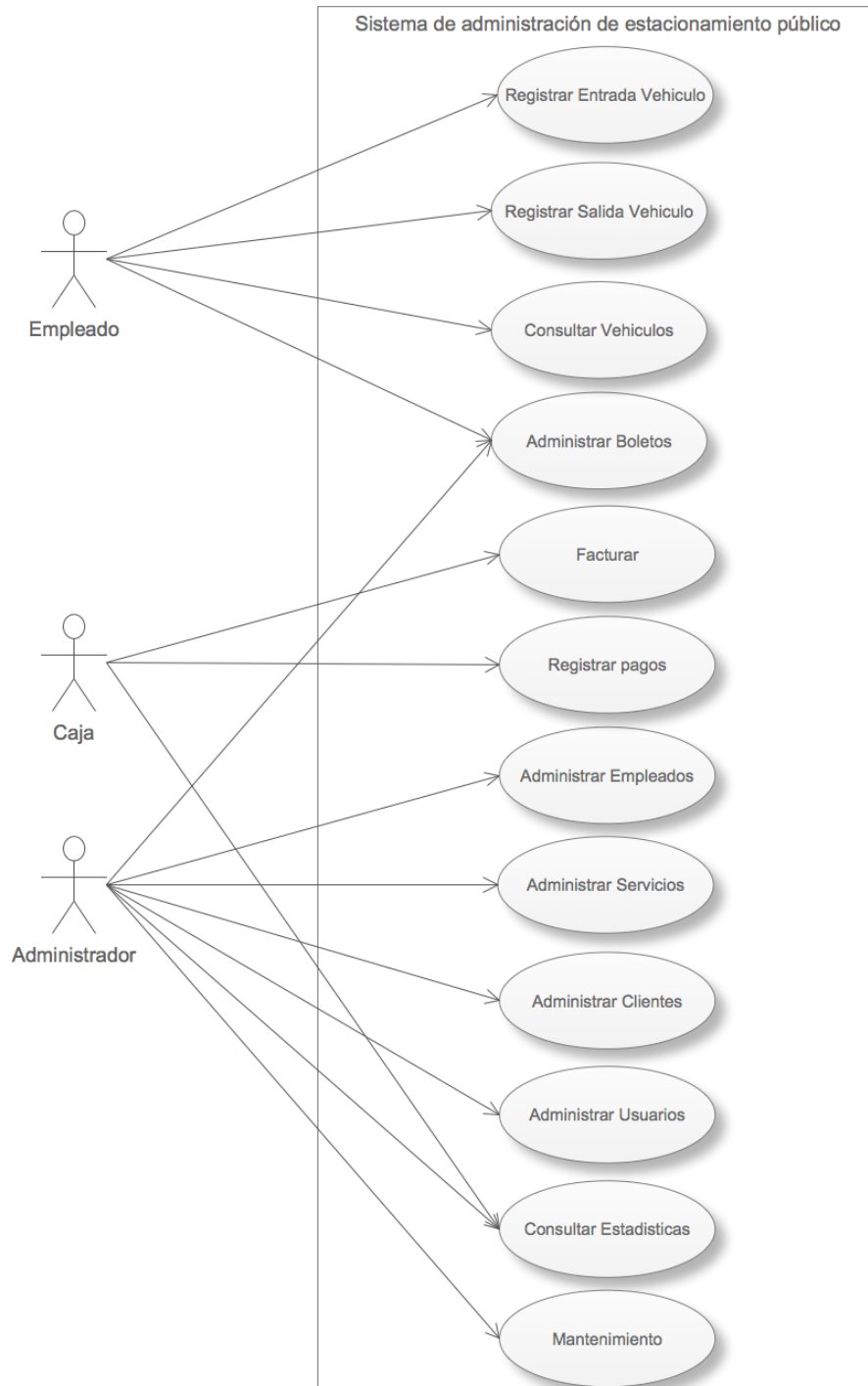


Figura 4.1.1.1 Diagrama general de casos de uso del sistema de administración de estacionamiento público

El modulo de registro de vehículos tiene por objetivo llevar el registro de los vehículos que entran al estacionamiento, la asignación de un lugar para aparcar y el registro de los servicios que contrata el cliente.

En la figura 4.1.1.2 se muestra los casos de uso de Registrar Vehículo que son realizados por el actor empleado

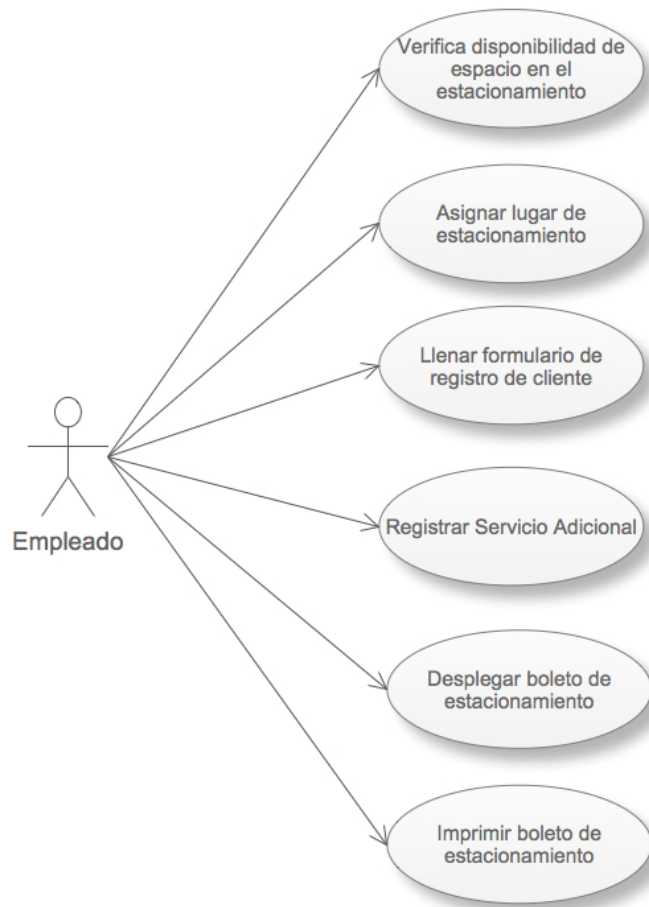


Figura 4.1.1.2 Diagrama de casos de uso registrar vehículo

En la tabla 4.1.1.1 vemos la secuencia de registrar vehículo:

Caso de Uso:		Registrar Vehículo		
Actores:		Empleado		
Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Ingreso al sistema proporcionando usuario y contraseña	2	Valida los datos y filtra información de las áreas correspondientes	
3	Da clic en el botón de “nuevo registro”	4	Muestra el formulario de datos de vehículo	
5	Completa los campos necesarios para el registro y da clic en el botón llamado “asignar cajón”	6	Muestra la ventana de los cajones. Los que están disponibles.	
7	Da clic en el botón del cajón deseado	8	Vuelve al formulario de vehículo completando el campo llamado “asignar cajón”	
9	Selecciona el servicio que desea el cliente (por hora , o por pensión) los cuales son botones	10.1	Botón “por hora”: se muestra la ventana de boleto con lo que se despliega la información correspondiente a los datos del vehículo con el servicio elegido.	Si no está completo los campos para el botón pensión no se permite el registro.
		10.2	Botón: “por pensión”: se muestra la ventana de “administración de clientes”	
11.1	Puede seleccionar un servicio adicional si el cliente así lo desea. Y da clic en el botón de “imprimir”	12.1	Se registra en la base de datos la información del vehículo e información necesaria. Se imprime el comprobante para el cliente	

11.2	Se hace la búsqueda del cliente, si no se encuentra se llena el formulario y da clic en el botón "registrar"	12.2	Valida la información necesaria, guarda los datos temporalmente, y muestra el formulario de pensión.	Si no están correctos no se hace el registro y no puede hacer el cambio de ventana a formulario de pensión.
13.1	Entrega el comprobante al cliente y permite el paso al vehículo			
13.2	El usuario introduce los datos necesario del formulario pensión y da clic en el botón "costo"	14.2	Valida la información introducida, calcula el costo, y despliega en pantalla los datos.	Si no están completos los datos no se puede dar clic en el botón "costo"
15.2	Pide al cliente anticipo para el servicio de pensión e introduce el monto al formulario. Da clic en el botón "registrar"	16.2	Valida el monto introducido y registra en la base de datos todos los datos proporcionados. Se imprime el comprobante del cliente	
17.2	Entrega el comprobante al cliente y permite el paso al vehículo			

Tabla 4.1.1.1 Secuencia registrar vehículo

En la figura 4.1.1.3 vemos el diagrama de secuencia para el registro del vehículo

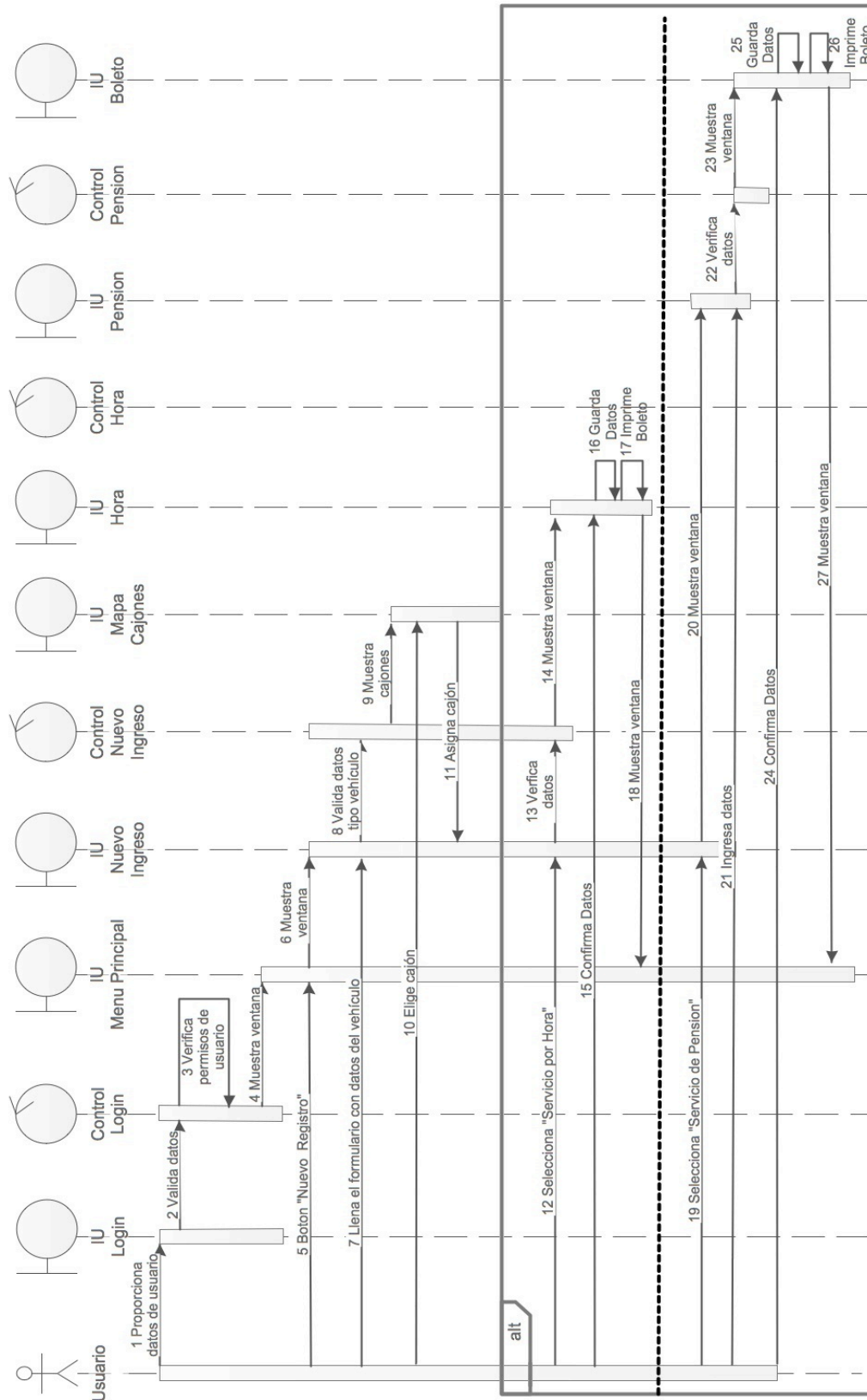


Figura 4.1.1.3 Diagrama de secuencia, registro de vehículo

El modulo Registrar Salida Vehículos se encarga de registrar la salida de los vehículos del estacionamiento y el cobro de los servicios contratados por el cliente. En la Figura 4.1.1.4 se muestra el caso de registrar salida

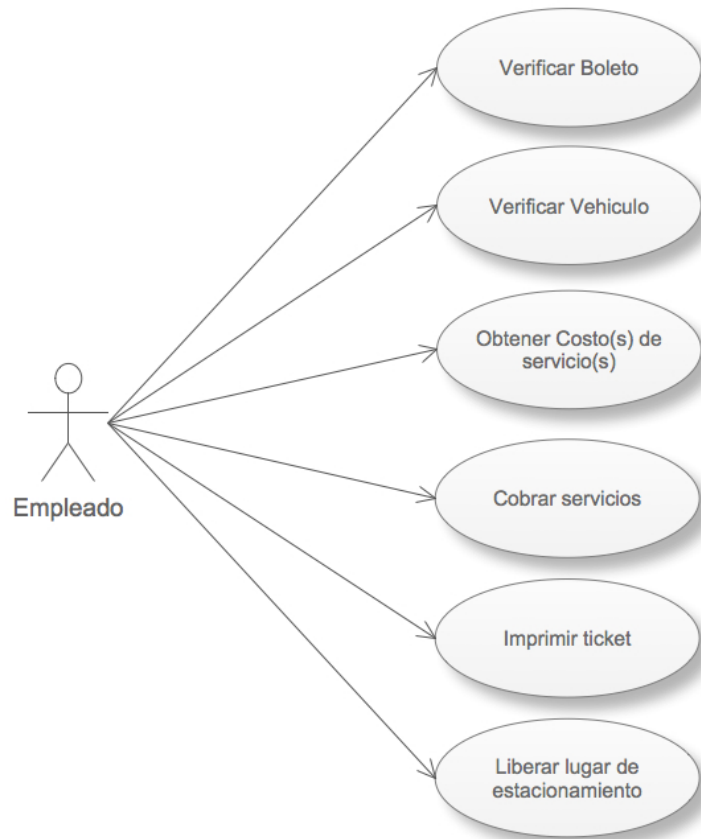


Figura 4.1.1.4 Caso de uso Registrar Salida

En la Tabla 4.1.1.2 vemos el diagrama de secuencia de Registrar Salida

Caso de Uso:		Registrar Salida		
Actores:		Empleado		
Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Ingreso al sistema proporcionando usuario y contraseña	2	Valida los datos y filtra información de las áreas correspondientes	
3	Recibe el comprobante del cliente. Introduce el numero de boleto en el campo llamado "No de boleto" y presiona el botón Enter	4	Valida el numero de boleto y despliega los datos correspondientes al tipo de servicio.	Se manda un mensaje si no existe el numero de boleto introducido
		4.1	Si es por pensión solo se muestra el tiempo que hace falta por terminar el contrato	
		4.2	Si es por hora se calcula el costo de servicios.	
5.1	Da clic en el botón "Salida". Permite la salida del vehículo.	6.1	Se registra la hora y fecha de salida del vehículo.	
5.2	Recibe el pago por los servicios. Da clic en el botón llamado "Pagado".	6.2	Imprime ticket. Se registra la fecha y hora de salida del vehículo. Cambia el estado del cajón utilizado de ocupado a libre.	
7.2	Entrega ticket al cliente y permite la salida del vehículo.			

Tabla 4.1.1.2 Secuencia registrar salida

En la figura 4.1.1.5 se muestra el diagrama de secuencia para el registro de salida del vehículo.

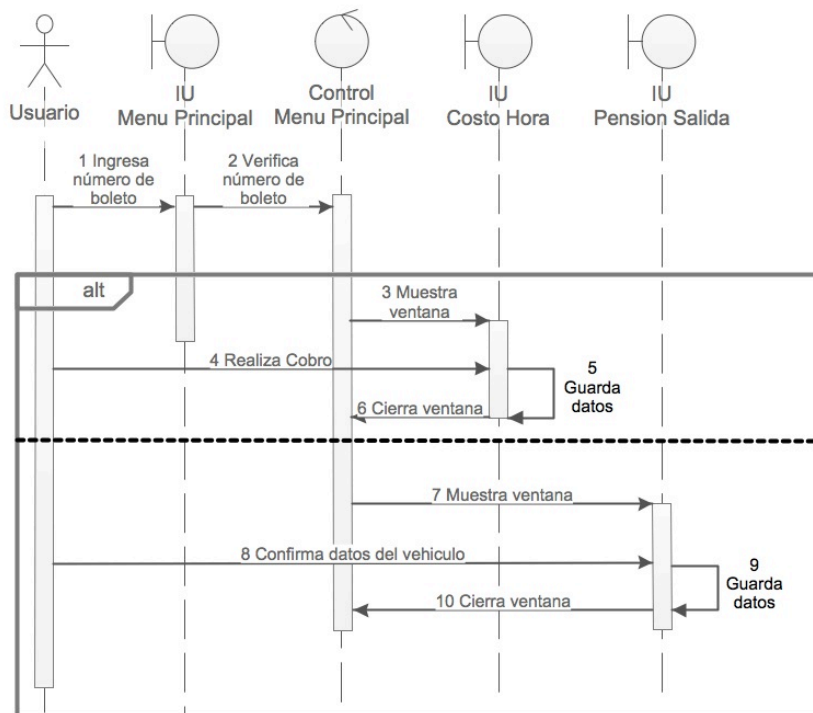


Figura 4.1.1.5 Diagrama de secuencia, registro salida

El modulo facturar se encarga de llevar el control de las facturas expedidas y la cancelación de éstas. En la figura 4.1.1.6 se muestra el caso de uso facturar.

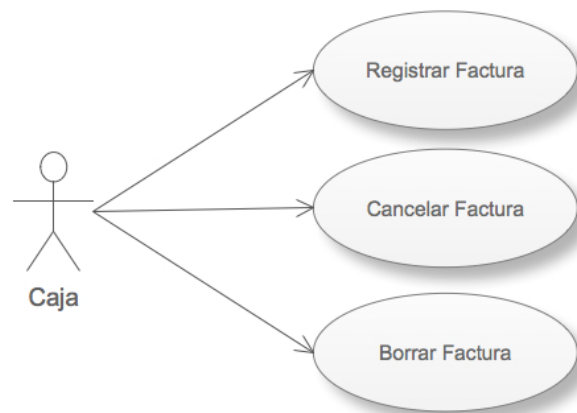


Figura 4.1.1.6 Caso de uso facturar

El modulo registrar pagos se encarga de registrar los pagos que se han realizado, el pago a empleados, el pago por servicios externos. También se encarga de registrar los cobros que se han hecho como: el cobro por pensión, el cobro por horas de estacionamiento y el cobro por servicios adicionales que ofrece el estacionamiento. También se realiza el corte de caja. En la figura 4.1.1.7 se muestra el caso de uso registrar pagos

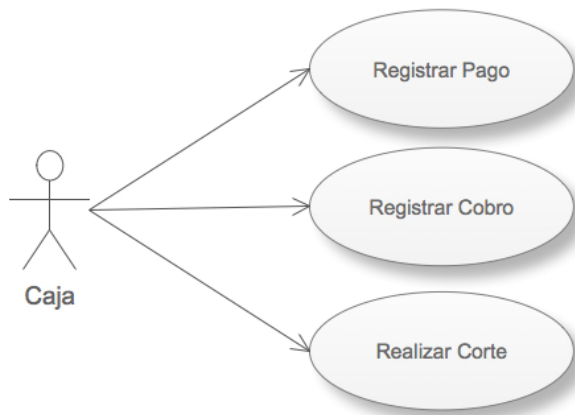


Figura 4.1.1.7 Caso de uso registrar pagos

El Modulo Administrar Empleados se encarga de llevar el control de las personas que son empleados del estacionamiento. Se hace el registro, la modificación de datos y el borrado de registros. En la figura 4.1.1.8 se muestra el caso de uso administrar empleados

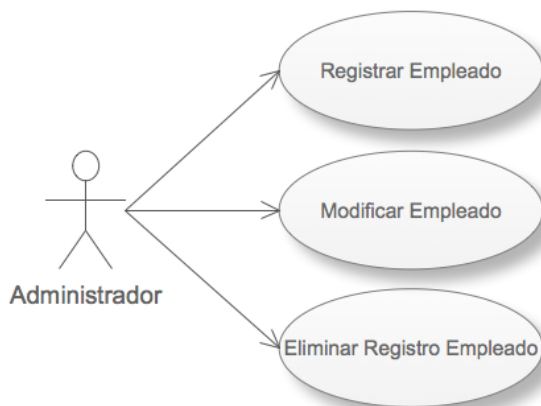


Figura 4.1.1.8 Caso de uso administrar empleados

En la tabla 4.1.1.3 se muestra el diagrama de secuencia del caso de uso administrar empleado donde solo se describe la opción de registrar empleado.

Caso de Uso:		Administrar Empleado		
Actores:		Administrador		
Actor		Sistema		
Paso	Acción	Paso	Acción	Excepción
1	Ingreso al sistema proporcionando usuario y contraseña	2	Valida los datos y filtra información de las áreas correspondientes	
3	Da clic en el botón "Ajustes"	4	Muestra la ventana de Ajustes	
5	Da clic en el botón llamado "Empleados"	6	Muestra la ventana de Empleados	
7	Selecciona el botón "Nuevo Empleado"	8	Muestra el formulario de nuevo empleados	
9	Llena el formulario de Empleados y da clic en el botón "Referencia"	10	Muestra la ventana de Persona de Referencia	
11	Llena el formulario de Persona de Referencia y da clic en el botón "Registrar"	12	Se registra en la base de datos la información de la persona de referencia y se le asigna este al nuevo empleado. Regresa a la ventana de Formulario de Empleado.	
13	Da clic en el botón "Guardar"	14	Guarda los datos del empleado con las asociaciones que fueron introducidas en el formulario.	

Tabla 4.1.1.3 Secuencia administrar empleado

En la figura 4.1.1.9 se muestra el diagrama de secuencia para el registro de empleados

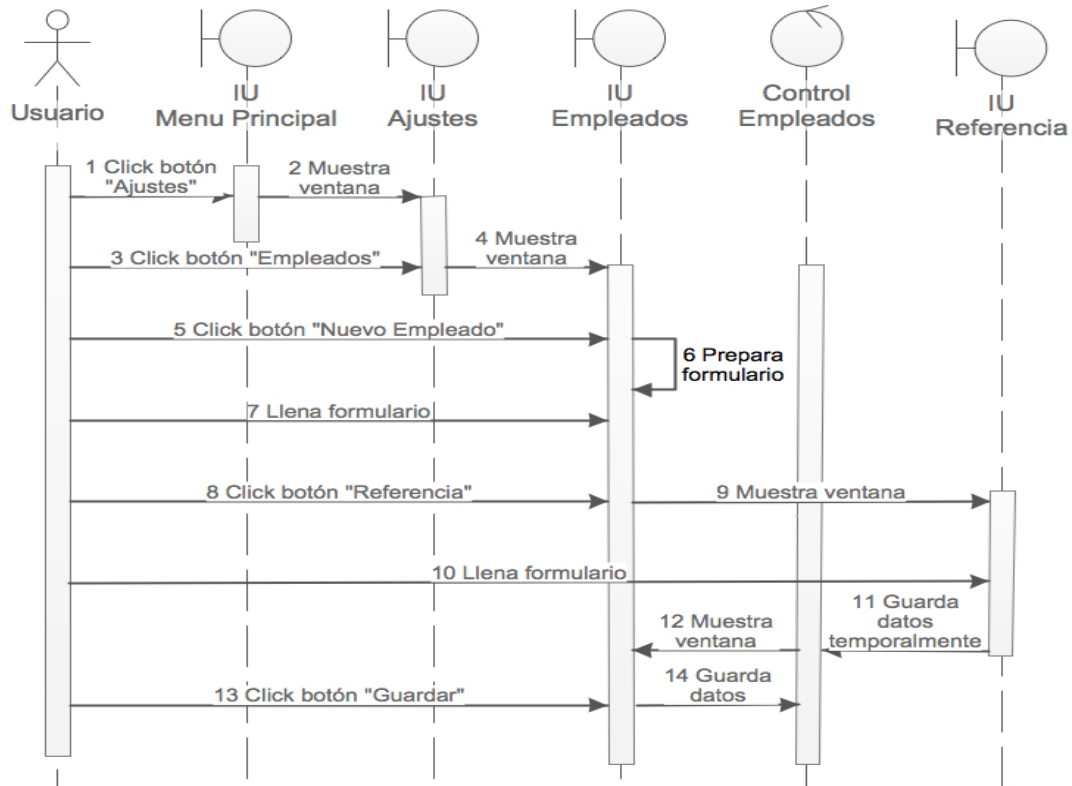


Figura 4.1.1.9 Diagrama de secuencia para el registro de empleado

El modulo consultar vehículos se encarga de mostrar al empleado los vehículos que están actualmente estacionados en el estacionamiento.

El modulo Administrar Boletos se encarga de llevar el control de todos los boletos expedidos, sirve para hacer modificaciones al boleto, como cancelaciones o cambio de servicios adicionales de un vehículo.

4.1.2 Diagrama de modelado de datos

Un modelo de datos es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia. Las características principales de un modelo de datos son:

- Analizar los aspectos de interés para una organización y la relación que tienen unos con otros.
- Descubrir y documentar los recursos de datos del negocio.
- Registra los requerimientos de datos de un proceso de negocio.

Tipos de modelado de datos básicamente son tres:

- **Conceptual:** muy general y abstracto, visión general del negocio/institución.
- **Lógico:** versión completa que incluye todos los detalles acerca de los datos.
- **Físico:** esquema que se implementara en un manejador de bases de datos (DBMS).

En la figura 4.1.2.1 se observa un diagrama de modelo de datos de la orden de servicio de pensión para un cliente ya registrado. Como se puede ver el diagrama el proceso que se sigue para realizar una orden de pensión de la empresa se requiere pasar por diferentes procesos. Inicialmente el cliente detallará la orden de servicio al empleado; a continuación el empleado registra los datos del vehículo, se le asigna un cajón y se le asigna el servicio que ordeno el cliente. El empleado genera el boleto con los datos proporcionados y hace el cobro del servicio. Se registra el pago del cliente, el cajón pasa de estar libre a ocupado, se registra el número de boleto al cliente y se crea un registro de entradas y salidas del vehículo del estacionamiento.

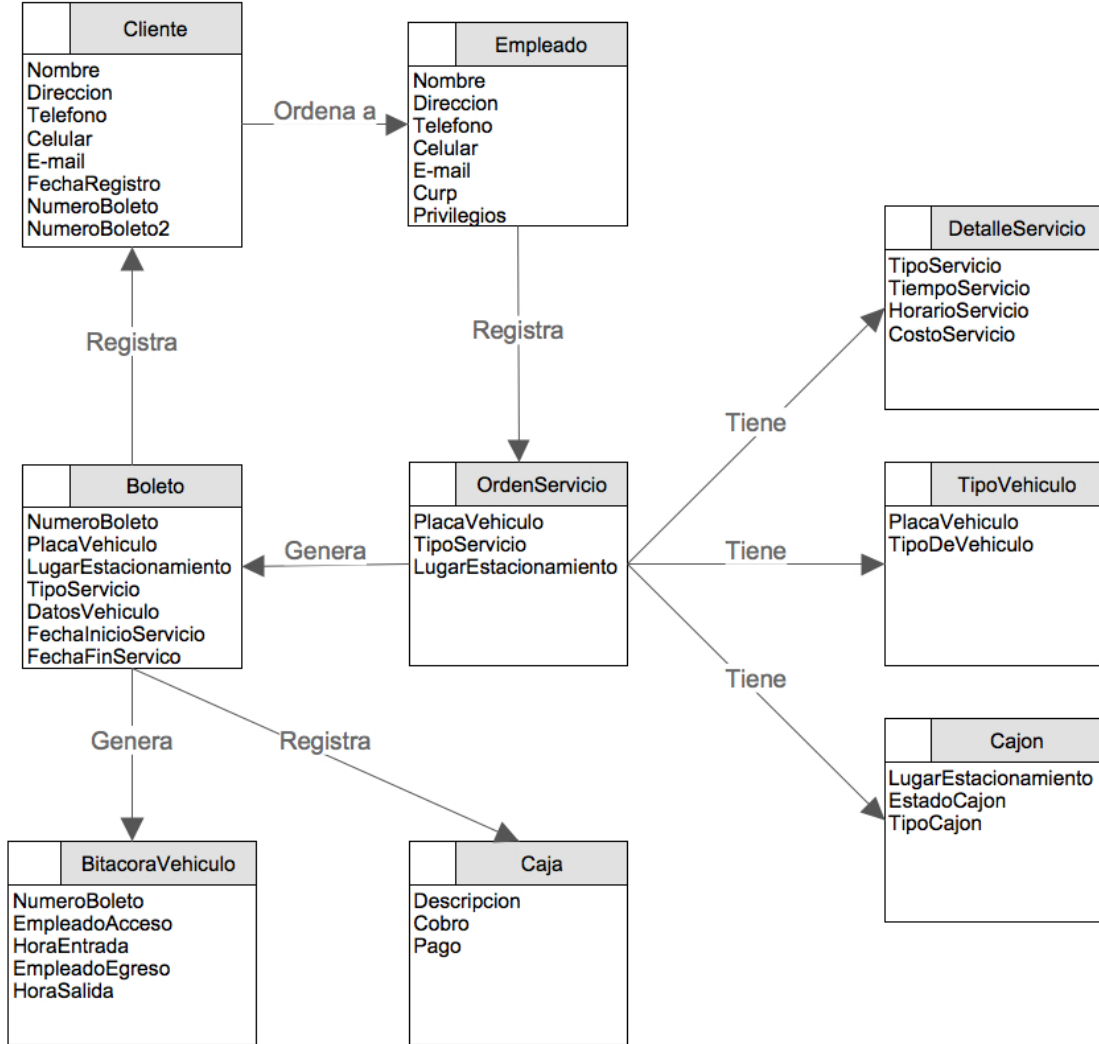


Figura 4.1.2.1 Modelo de datos registro de pensión para clientes registrados

4.1.3 Diagrama entidad – relación (DER)

La definición de la estructura y respuesta de una base de datos tiene que ver en buena medida a la modelación de los datos que suministra el mundo real.

El diagrama entidad-relación se utiliza para describir los datos y sus relaciones de manera conceptual y de visión final.

El modelo entidad-relación se compone por:

- **Entidades:** Todo lo que existe y es capaz de ser descrito (sustantivo).

- **Atributos:** Es una característica (adjetivo) de una entidad que puede hacer una de tres cosas:
 - Describir
 - Identificar
 - Relacionar

- **Relaciones:** La conexión que existe entre dos entidades (verbo).

- **Cardinalidad:** Número de ocurrencias que pueden existir entre un par de entidades.

- **Súper llave:** Conjunto de uno o más atributos que "juntos" identifican de manera única a una entidad

- **Llave candidata:** Es una súper llave mínima

- **Llave primaria:** La seleccionada para identificar a los elementos de un conjunto de entidades.

En la figura 4.1.3.1 se muestran las entidades, como se relacionan entre estas y los atributos que contiene cada una. A partir de estos diagramas se crearan las tablas en la base de datos con las reglas de relación.

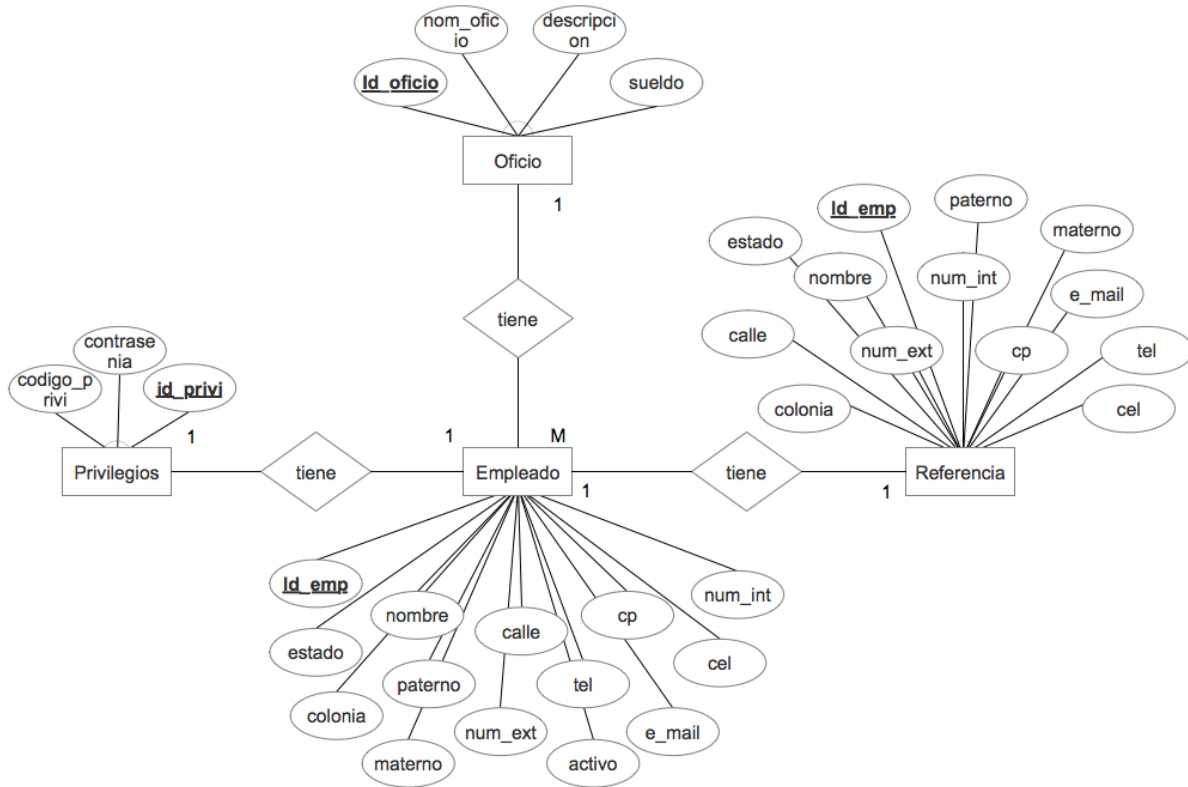


Figura 4.1.3.1 DER de las entidades Empleado, Privilegios, Referencia y Oficio.

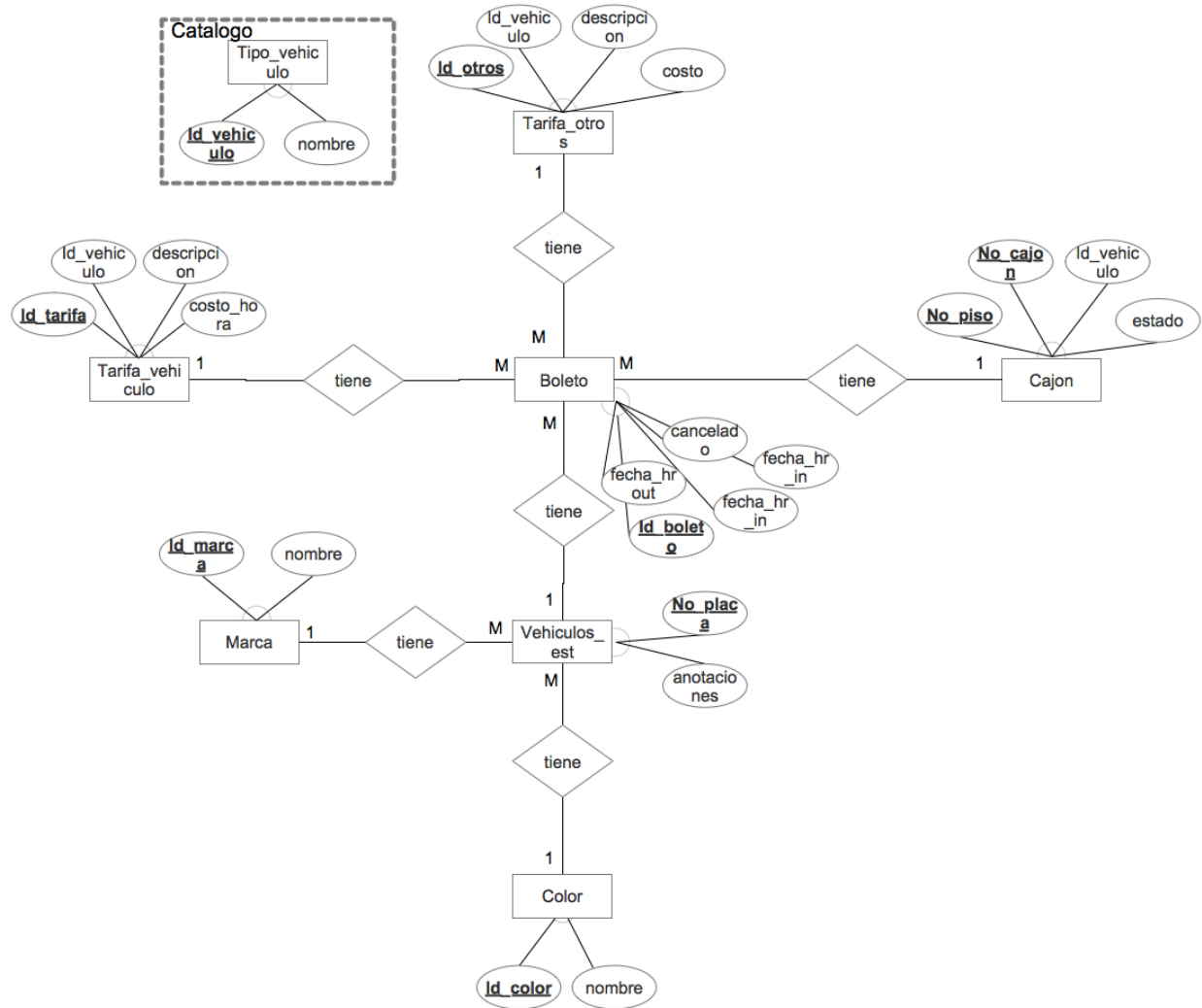


Figura 4.1.3.2 DER de las entidades Tarifa_vehiculo, Tarifa_otros, Boleto, Cajon, Vehiculos_est, Marca y Color.

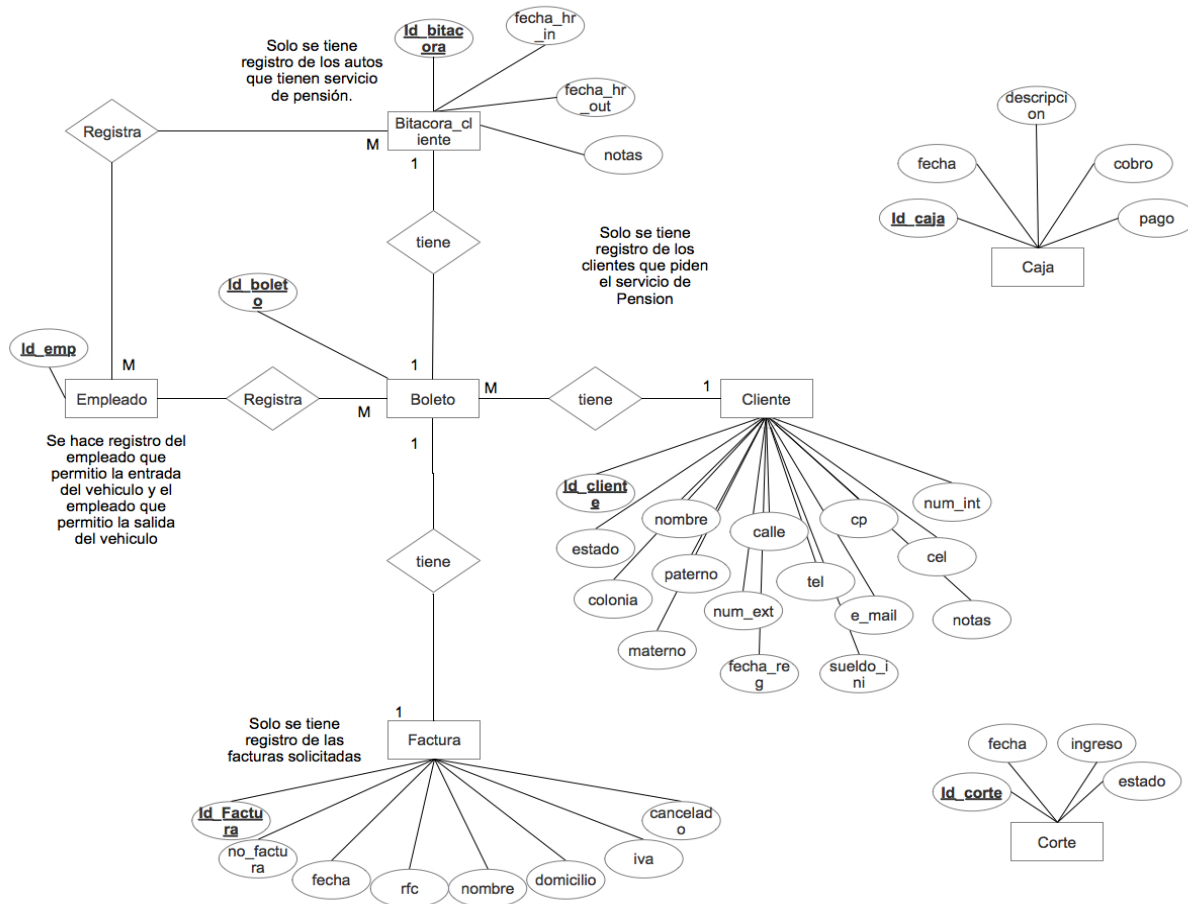


Figura 4.1.3.3 DER de las entidades Empleado, Boleto, Cliente, Factura, Bitacora_cliente, Caja y Corte

4.1.4 Diagrama de clases

El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir, los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación.

La clase define el ámbito de definición de un conjunto de objetos. Cada objeto pertenece a una clase. Los objetos se crean por instanciación de las clases. Cada clase se representa en un rectángulo con tres compartimientos:

- Nombre de la clase
- Atributos de la clase

- Operaciones de la clase

Los atributos de las clases son los siguientes:

- **Tipo:** puede llegar a depender del lenguaje de programación a utilizar.
- **Valor inicial:** valor que poseerá el atributo al crear un objeto.
- **Visibilidad:** está relacionado con el encapsulamiento.
- **Multiplicidad:** determinar si un atributo debe estar o no, y si posee un único valor o una lista de valores.
- **Ordenamiento:** especifica si el atributo determina alguna relación de orden dentro de la clase.
- **Capacidad de cambio:** permite definir atributos con valores constantes.
- **Modificadores:** un atributo puede ser de clase, derivado, volátil, transitorio.

Niveles de encapsulamiento:

(-) Privado: es el más fuerte. Esta parte es totalmente invisible desde fuera de la clase.

(#) Los atributos/operaciones protegidos están visibles para las clases amigas y para las clases derivadas de la original.

(+) Los atributos/operaciones públicos son visibles a otras clases.

En la figura 4.1.4.1 se muestran las clases que son necesarias para el registro de un vehículo el cual contrata el servicio de estacionamiento por hora.

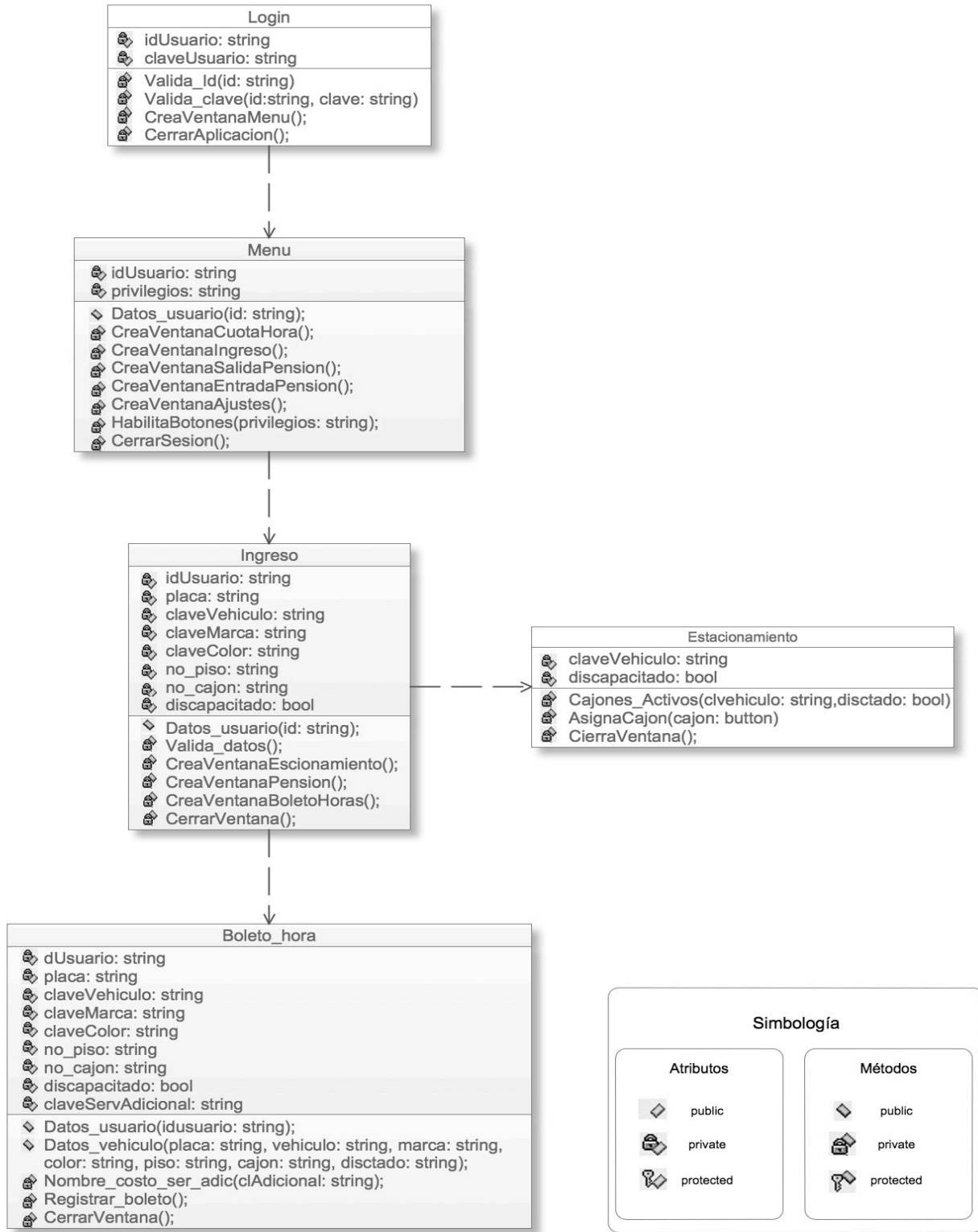


Figura 4.1.4.1 Diagrama de clases contrato servicio por hora

4.1.5 Diccionario de Datos

El Diccionario de Datos (DD) es una lista que describe todos los elementos del sistema, con definiciones precisas para los analistas. Se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño del proyecto.

Este documento es de gran importancia para tener registro de cómo están diseñadas las bases de datos, incluye información de diseño como, nombre de la tabla, y detalles de la composición como: nombre de campo, tipo de dato, tamaño de campo, además de una descripción de cada campo acerca del dato que va a representar y las relaciones que tienen cada una de ellas con los campos en común con las diferentes tablas.

De la tabla 4.1.5.1 a la tabla 4.1.5.17 se muestran los datos junto con su tipo de dato que se manejarán en la base de datos para su correcto funcionamiento e implementación en la base de datos.

REFERENCIAS

Nombre Original	Acrónimo	Tipo Dato	Llave Primaria	Tabla Asociada	Valor por Default	Descripción
Identificador de Referencia	Id_ref	integer	Si	CLIENTES	SERIAL	Identificador de la persona que es registrada como referencia del nuevo empleado
Nombre	Nombre	character varying(50)	No			
Apellido Paterno	Paterno	character varying(50)	No			
Apellido Materno	Materno	character varying(50)	No			
Calle	Calle	character varying(50)	No			
Numero exterior	Num_ext	integer	No			
Numero interior	Num_int	integer	No			
Código Postal	CP	integer	No			
Teléfono	Tel	character varying(15)	No			
Celular	Cel	character varying(15)	No			
E-mail	E_mail	character varying(50)	No			

Tabla 4.1.5.1 Referencias

MARCA

Nombre Original	Acronimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador Marca del Vehículo	Id_marca	integer	Si	VEHICULOS_EST	SERIAL	Identificador de las marcas registradas
Nombre Marca	Nombre	character varying(20)	No			Nombre de la marca del vehículo

Tabla 4.1.5.2 Marca

COLOR

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador del color	Id_color	integer	Si	VEHICULOS_EST	SERIAL	Identificador de los colores más usados
Nombre del color	Nombre	character varying(20)	No			Color del automóvil

Tabla 4.1.5.3 Color

PRIVILEGIOS

Nombre Original	Acronimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador de privilegios	Id_priv	integer	Si	EMPLEADOS	SERIAL	Identificador de los privilegios que se le asignan al usuario
Clave	Codigo_priv	character varying(10)	No			Secuencia de caracteres con los que se identifica el nivel de acceso a los módulos
Contraseña	Contrasenia	character varying(10)	No			Contraseña del usuario

Tabla 4.1.5.4 Privilegios

EMPLEADOS

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador or Empleado	Id_emp	integer	Si	BOLETO BITACOR A_CLIEN TE	SERIAL	Identificador del empleado
Nombre	Nombre	character varying(50)	No			
Apellido paterno	Paterno	character varying(50)	No			
Apellido materno	Materno	character varying(50)	No			
CURP	Curp	character varying(20)	No			
Calle	Calle	character varying(50)	No			
Numero exterior	Num_ext	integer	No			
Numero interior	Num_int	integer	No			
Código postal	CP	integer	No			
Teléfono	Tel	character varying(15)	No			
Celular	Cel	character varying(15)	No			
E-mail	E_mail	character varying(50)	No			
Identificador or de Referencia	Id_ref	integer	No			Identificador de la persona que se registro como referencia
Identificador or de Oficio	Id_oficio	integer	No			Identificador del tipo de trabajo que realiza
Identificador or de privilegios	Id_priv	integer	No			Identificador de privilegios y contraseña
Activo	Activo	boolean	No			Señala si la persona registrada esta actualmente trabajando.

Tabla 4.1.5.5 Empleados

OFICIOS

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador Oficio	Id_oficio	Integer	Si	EMPLEADOS	SERIAL	Identificador del trabajo a realizar
Nombre Oficio	Nom_oficio	character varying(15)	No			Nombre del trabajo
Descripción	Descripcion	text	No			Descripción del trabajo que se realiza
Sueldo	Sueldo	numeric(9,2)	No			Cantidad que se paga al empleado

Tabla 4.1.5.6 Oficios

TIPO_VEHICULOS

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador Vehículo	Id_vehiculo	integer	Si	TARIFA_VEHICULO TARIFA_OTROS CAJONES VEHICULOS_EST	SERIAL	Identificador del tipo de vehículos
Nombre Vehículo	Nombre	character varying(20)	No			Tipos de vehículos a los que puede acceder al estacionamiento

Tabla 4.1.5.7 Tipo_vehiculos

CAJONES

Nombre Original	Nombre	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Nivel del edificio	No_piso	integer	Si	BOLETO		Numero del piso en el que se encuentra
Numero de cajón	No_cajon	integer	Si	BOLETO		Numero de cajón
Identificador vehículo	Id_vehiculo	integer	No			Para que tipo de vehículo es el cajón: "Automóvil", "Camioneta", "Motocicleta" o "Discapacitado"
Estado	Estado	character varying(20)	No			Puede tomar los valores "Libre" y "Ocupado"

Tabla 4.1.5.8 Cajones

TARIFA_OTROS

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador otro servicio	Id_otros	integer	Si	BOLETO	SERIAL	Identificador de servicios adicionales
Identificador vehículo	Id_vehículo	integer	No			Identificador del tipo de vehículo para el cual es el servicio
Descripción	Descripción	character varying(50)	No			Descripción del servicio
Costo	Costo	numeric(9,2)	No			Costo del servicio

Tabla 4.1.5.9 Tarifa_otros

TARIFA_VEHÍCULOS

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador tarifa	Id_tarifa	integer	Si	BOLETO	SERIAL	Identificador de la tarifa por estacionamiento
Identificador vehículo	Id_vehículo	integer	No			Identificador del tipo de vehículo para el cual es la estancia
Tiempo de estancia	Tiempo	character varying(50)	No			Nombre del tiempo de estancia
Horario	Horario	character varying(50)	No			Nombre del horario de servicio
Costo	Costo	numeric(9,2)	No			Costo por el servicio

Tabla 4.1.5.10 Tarifa_vehículos

VEHÍCULOS_EST

Nombre Original	Acronimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Número de placa	No_placa	character varying(10)	Si	BOLETO		Número de placa del vehículo
Identificador vehículo	Id_vehiculo	integer	No			Identificador del tipo de vehículo
Identificador marca	Id_marca	Integer	No			Identificador de la marca del vehículo
Identificador color	Id_color	integer	No			Identificador del color del vehículo
Anotaciones	Anotaciones	text	No			Anotaciones sobre el estado del vehículo

Tabla 4.1.5.11 Vehículos_est

BOLETOS

Nombre Original	Acónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador boleto	Id_boleto	integer	Si	CLIENTES FACTURAS BITACOR A_CLIENTE	SERIAL	Folio del Boleto del presente año
Año	Año	integer	Si			Año en que se expide el boleto
Número de placa	No_Placa	character varying(10)	No			Numero de placa del vehículo
Número de piso	No_piso	integer	No			Identificador del piso donde se encuentra el cajón
Número de cajón	No_cajon	Integer	No			Identificador del cajón
Identificador del Operador de entrada	Operador_1	integer	No			Usuario que permitió la entrada al estacionamiento
Fecha y hora de entrada	Fecha_hr_in	timestamp	No			Fecha y hora de entrada del vehículo al estacionamiento
Identificador del operador de salida	Operador_2	integer	No			Usuario que permitió la salida del vehículo
Fecha y hora de salida	Fecha_hr_out	timestamp	No			Fecha y hora de salida del vehículo del estacionamiento
Identificador de servicio adicional	Id_otros	integer	No			Identificador del contrato de un segundo servicio
Identificador de la tarifa a cobrar	Id_tarifa	integer	No			Identificador del servicio contratado para estacionamiento
Cancelado	Cancelado	boolean	No			Identificador de cancelación de boleto (pagado)

Tabla 4.1.5.12 Boletos

CLIENTES

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador Cliente	Id_cliente	integer	Si		SERIAL	Identificador de cliente
Nombre	Nombre	character varying(50)	No			
Apellido paterno	Paterno	character varying(50)	No			
Apellido materno	Materno	character varying(50)	No			
Colonia	Colonia	character varying(50)	No			
Estado	Estado	character varying(100)	No			
Calle	Calle	character varying(50)	No			
Número exterior	Num_ext	integer	No			
Número interior	Num_int	integer	No			
Código postal	CP	integer	No			
Teléfono	Tel	character varying(15)	No			
Celular	Cel	character varying(15)	No			
E-mail	E_mail	character varying(50)	No			
Notas	Notas	text	No			Notas sobre el auto
Fecha de registro	Fecha_reg	Date	No			Fecha que se registro al cliente
Identificador de boleto	Id_boleto	integer	No			Identificador del boleto si es que tiene estacionado un vehículo
Identificador de segundo boleto	Id_boleto_2	integer	No			Identificador del 2° boleto si es que tiene estacionado un vehículo

Tabla 4.1.5.13 Clientes

FACTURAS

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador Factura	Id_factura	integer	Si	BOLETO	SERIAL	Identificador de facturas expedidas
Numero de factura	No_factura	character varying(50)	No			Folio de la factura que se entrego
Fecha de expedición	Fecha	Date	No			Fecha en que se expidió la factura
RFC	RFC	character varying(20)	No			
Nombre	Nombre	character varying(50)	No			
Domicilio	Domicilio	character varying(70)	No			
Ciudad	Ciudad	character varying(50)	No			
Identificador de boleto	Id_boleto	integer	No			Identificador del boleto por la que se está haciendo la factura
IVA	Iva	integer	No			IVA que se aplica
Cancelado	Cancelado	boolean	No			Identificador de cancelación de factura.

Tabla 4.1.5.14 Facturas

BITACORA_CLIENTE

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador de bitácora	Id_bitacora	integer	Si		SERIAL	Identificador del registro de bitácora
Identificador boleto	Id_boleto	integer	No			Vehículo que ingresa
Identificador operador entrada	Operador_1	integer	No			Usuario que permitió la entrada al estacionamiento
Fecha y hora de entrada	Fecha_hr_in	timestamp	No			Fecha y hora de entrada del vehículo al estacionamiento
Identificador operador salida	Operador_2	integer	No			Usuario que permitió la salida del estacionamiento
Fecha y hora de salida	Fecha_hr_out	timestamp	No			Fecha y hora de salida del vehículo del estacionamiento
Notas	Notas	text	No			Notas sobre el vehículo

Tabla 4.1.5.15 Bitacora_cliente

CORTES

Nombre Original	Nombre	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador de corte	Id_corte	integer	Si		SERIAL	Identificador del corte
Fecha registro	Fecha	Date	No		Now()	Fecha en que se hizo el corte
Monto	Monto	numeric(9,2)	No			Ingresos totales que se realizaron hasta la fecha de corte

Tabla 4.1.5.16 Cortes

CAJA

Nombre Original	Acrónimo	Tipo de Dato	Llave Primaria	Tabla Asociada	Valor por Default	Comentario
Identificador movimiento caja	Id_caja	integer	Si		SERIAL	Identificador de movimiento en caja
Fecha de registro	Fecha	Date	No			Fecha que se realizo el movimiento
Descripción	Descripción	text	No			Descripción de que fue el movimiento
Cobros	Cobros	numeric(9,2)	No			Cantidad que entro a caja
Pagos	Pagos	numeric(9,2)	No			Cantidad que salió de caja

Tabla 4.1.5.17 Caja

4.2 Implementación de Base de Datos

Instalación de PostgreSQL 8.2.4 en Windows

Lo primero que se tiene que hacer es entrar a la página de Postgresql <http://www.postgresql.org.es/descargas> seleccionamos el sistema operativo Windows que es el sistema operativo que utilizaremos, guardamos el archivo de instalación en una carpeta de nuestro disco duro.

Una vez que se tenga el archivo de instalación, se procede a ejecutar el programa de instalación.

El programa de instalación se identifica con el nombre de “postgresql-8.4.11-2-windows”.

En la figura 4.2.1 se muestra el instalador



Figura 4.2.1 Ejecución del instalador

Se pide una contraseña para la base de datos esta es del usuario postgres. En la figura 4.2.2 se muestra la pantalla donde se pide este dato

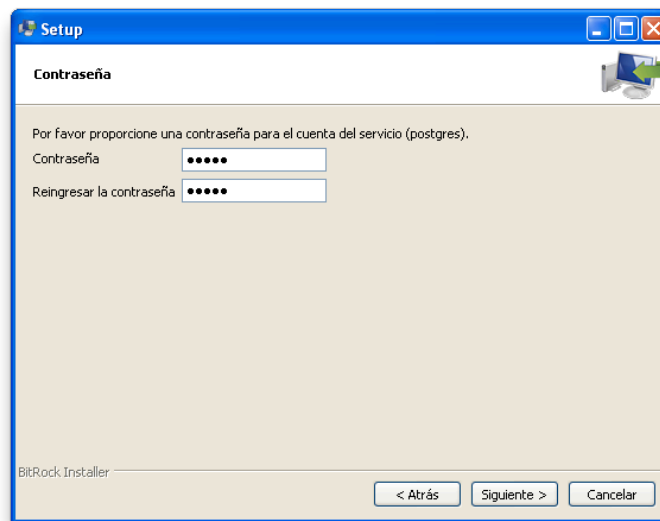


Figura 4.2.2 Contraseña de la base de datos

Se dejo el mismo puerto que se especifica por default

Puerto: 5432

Creación de la Base de Datos a través de la terminal sql:

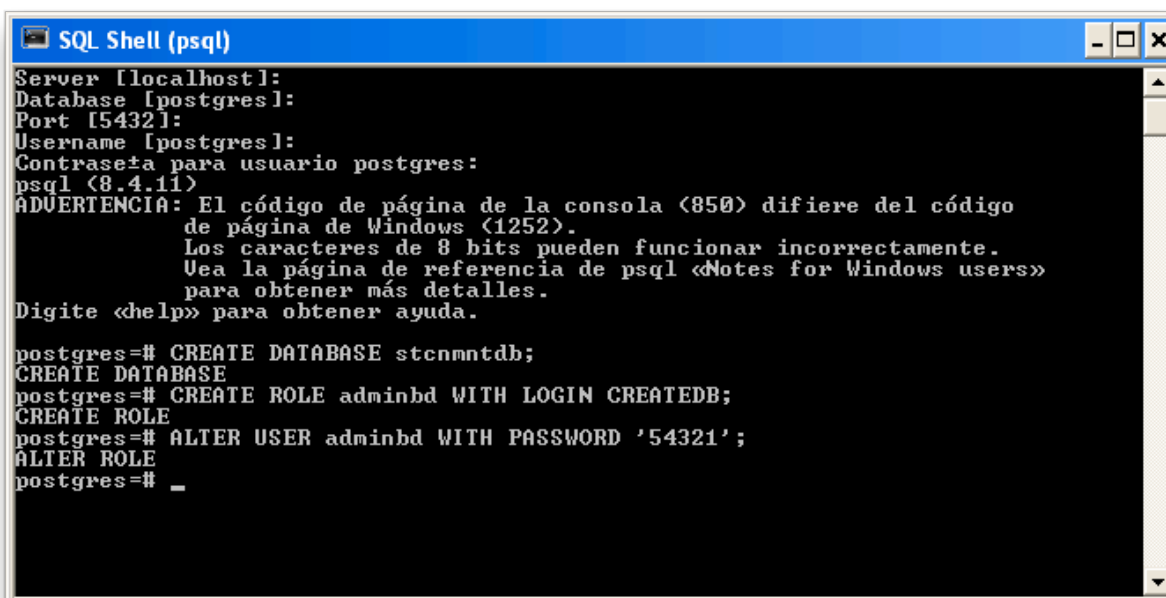
A continuación se describe como se crea una base de datos, un usuario y una tabla en PostgreSQL a través de la terminal.

- Primeramente ejecutamos SQLshell(psql) e introducimos los datos del usuario postgres. Los que se pidieron cuando se instalo PostgreSQL
- Una vez que estamos en la sesión creamos la base de datos con el comando:
CREATE DATABASE stcnmntdb;
Si la base se creó correctamente, da como resultado: CREATE DATABASE

- Creamos un nuevo usuario con la capacidad de poder logearse, con el comando: CREATE ROLE adminbd WITH LOGIN;
- Le asignamos un password con el comando: ALTER USER adminbd WITH PASSWORD '54321';

Con esta base de datos y usuario podremos crear las tablas y funciones necesarias para el correcto funcionamiento del sistema.

En la figura 4.2.3 se muestra como se crea la base de datos y un usuario.



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
psql (8.4.11)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

postgres=# CREATE DATABASE stcnmtdb;
CREATE DATABASE
postgres=# CREATE ROLE adminbd WITH LOGIN CREATEDB;
CREATE ROLE
postgres=# ALTER USER adminbd WITH PASSWORD '54321';
ALTER ROLE
postgres=# _
```

Figura 4.2.3 Creación de base de datos y un usuario

Creando una tabla por medio de la terminal.

Una vez creada la base de datos, procedemos a crear las diferentes tablas.

Se ingresa a la base de datos creada con el nuevo usuario para comprobar si fue creada con éxito. Véase figura 4.2.4

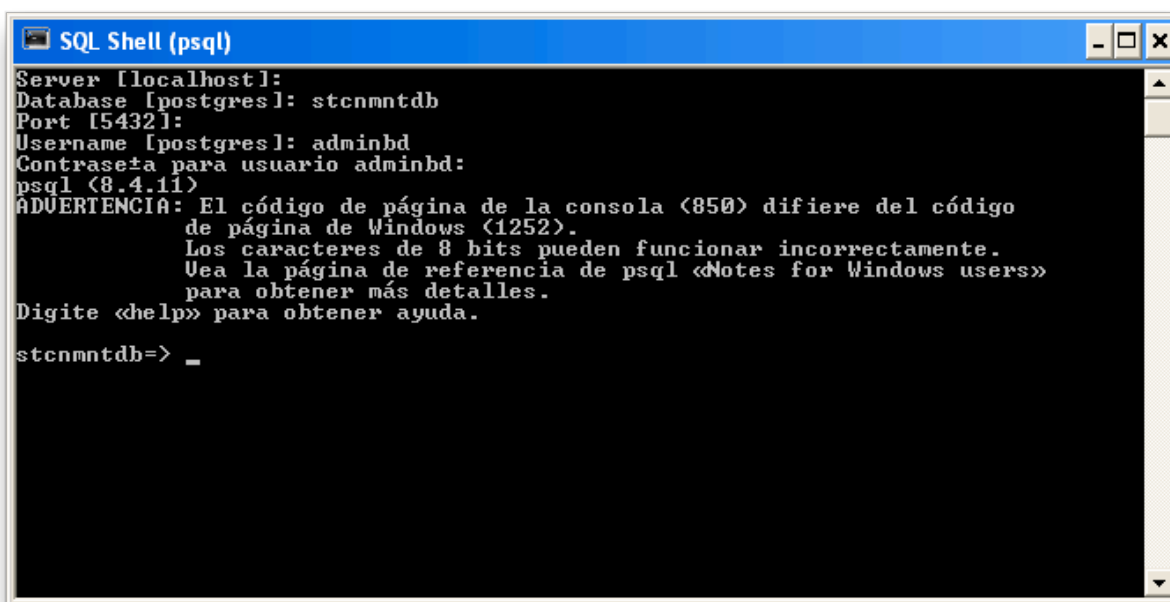


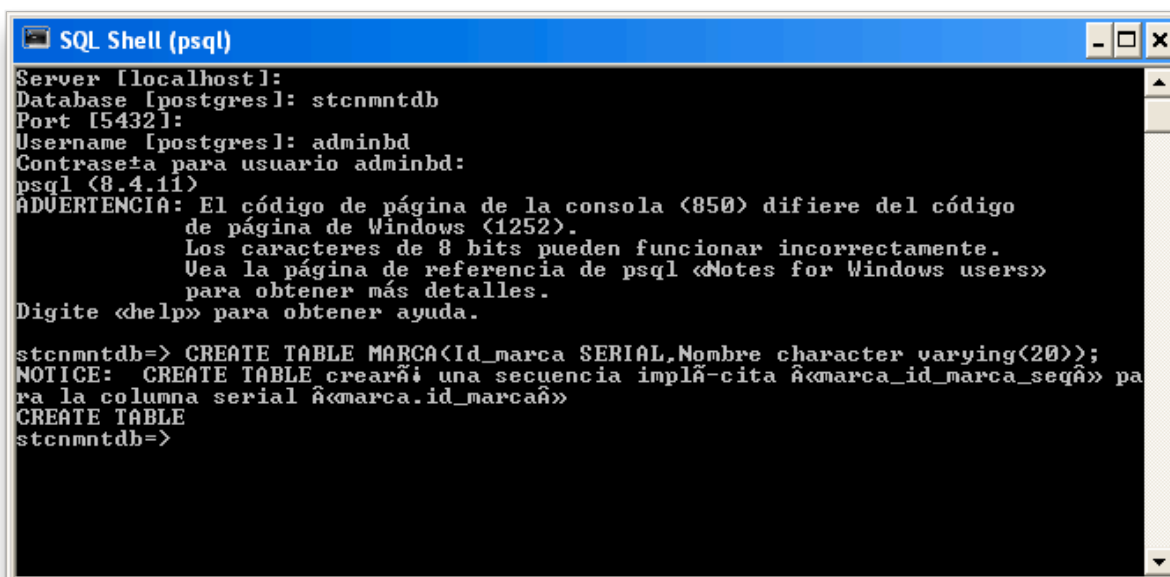
Figura 4.2.4 Ingreso la terminal sql con nuevo role creado

A continuación se muestra el código que se introdujo en la terminal para la creación de la tabla CLIENTES, en este se puede apreciar la creación de un índice en el atributo Id_cliente y de asignación de valores default.

```
CREATE TABLE CLIENTES(
Id_cliente SERIAL,
Nombre character varying(50),
Paterno character varying(50),
Materno character varying(50),
Calle character varying(50),
Num_ext integer,
Num_int integer,
CP integer,
Tel character varying(15),
```

Cel character varying(15),
E_mail character varying(50),
Notas text,
Fecha_reg Date,
Id_boleto integer default 0,
Id_boleto_2 integer default 0);

En la figura 4.2.5 se muestra la creación de la tabla MARCA en la terminal, dando como resultado que se creara una secuencia y que la tabla fue creada.



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]: stcnmntdb
Port [5432]:
Username [postgres]: adminbd
Contraseña para usuario adminbd:
psql (8.4.11)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

stcnmntdb=> CREATE TABLE MARCA(Id_marca SERIAL,Nombre character varying(20));
NOTICE: CREATE TABLE creará una secuencia implícita «marca_id_marca_seq» pa
ra la columna serial «marca.id_marca»
CREATE TABLE
stcnmntdb=>
```

Figura 4.2.5 Creación de una tabla en la terminal

El árbol de tablas:

Con la herramientas pgAdminIII podemos observar las tablas que fueron creadas en la base de datos.

Accedemos a la aplicación pgAdminIII. En las figuras 4.2.6 y 4.2.7 se aprecia el logotipo de pgAdminIII y la apariencia de la herramienta respectivamente.

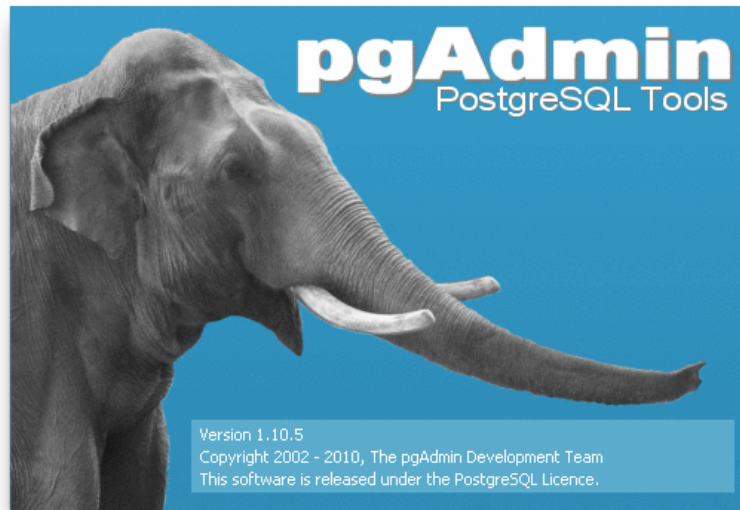


Figura 4.2.6 Logo de pgAdminIII

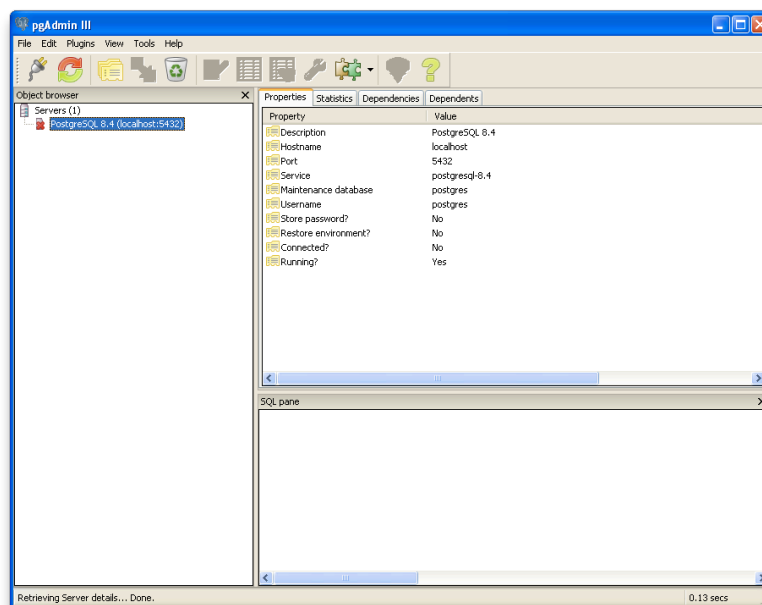


Figura 4.2.7 Apariencia de la herramienta pgAdminIII

Nos conectamos al servidor, en este caso PostgreSQL 8.4 (localhost), para ello necesitamos introducir nuestro password de postgres. Véase figura 4.2.9

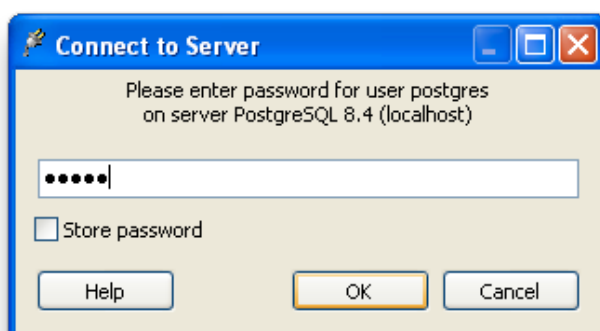


Figura 4.2.8 Password para conectarse al servidor

En la base de datos se muestra la base de datos que creamos desde la terminal (stcnmntdb), damos doble clic sobre esta para desplegar su contenido. Véase figura 4.2.10

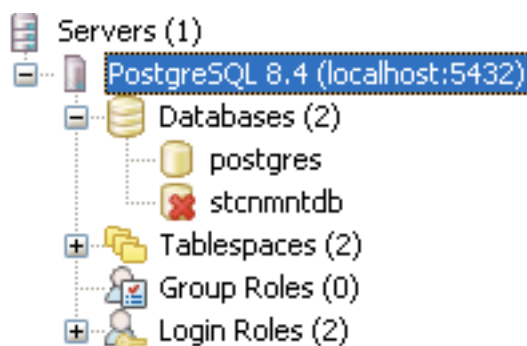


Figura 4.2.10 Bases de datos

Nos adentramos en el árbol de objetos hasta llegar a las tablas de la base de datos. En la figura se muestran todas las tablas que fueron creadas para el sistema de administración de estacionamiento público. Véase figura 4.2.11

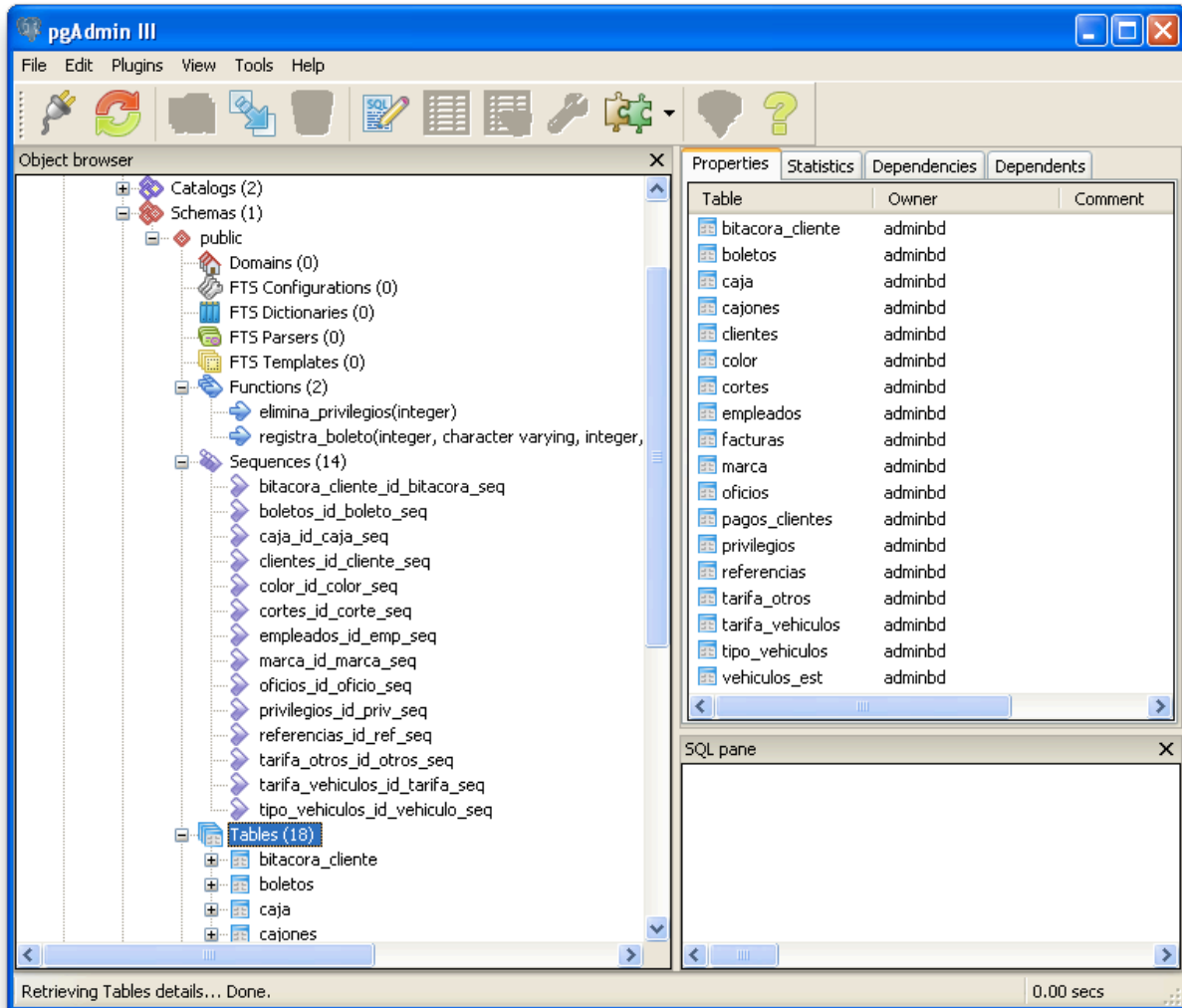


Figura 4.2.11 Tablas

Como se puede crear tablas en la terminal también se puede realizar en pgAdmin ya que es una herramienta gráfica para poder realizar las mismas operaciones de creación de bases de datos, tablas, funciones entre otros.

Creación de Tablas a través de PgAdminIII

Para la creación de tablas se necesita hacer clic derecho sobre el objeto Tables y seleccionar New Table... En la figura se muestra la apariencia del submenú que se despliega después de hacer clic derecho sobre Tables. Véase figura 4.2.12

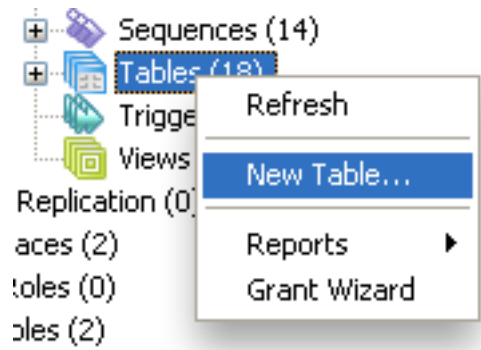


Figura 4.2.12 Creación de nueva tabla

Se abre una nueva ventana donde introducimos los datos de la tabla. En la figura se muestra la apariencia de la ventana New Table, asignando a la nueva table el nombre de “color”. Véase figura 4.2.13

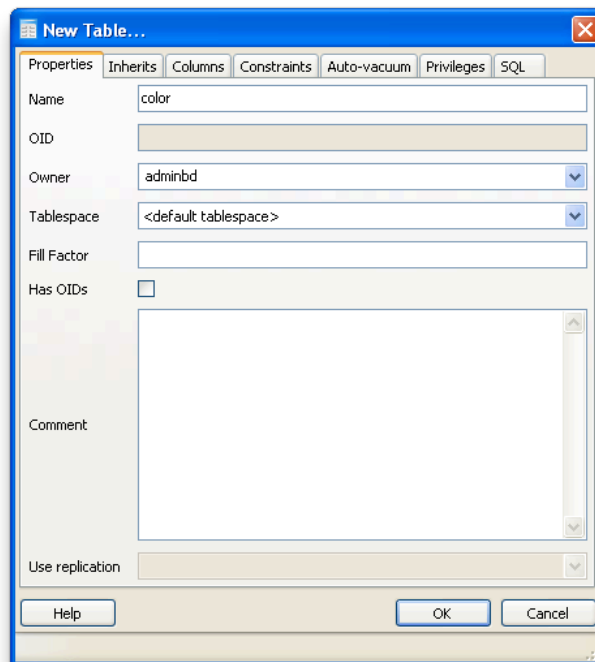


Figura 4.2.13 Ventana new table

Para agregar columnas a la tabla es necesario ir al tab “Columns” y posteriormente dar clic en el botón “Add” este desplegara una nueva ventana, llamada “New Column...” donde se le asigna un nombre, el tipo de datos, valor default, y si es nulo.

En la figura se muestra la apariencia de la ventana “New Column...”. Véase figura 4.2.14

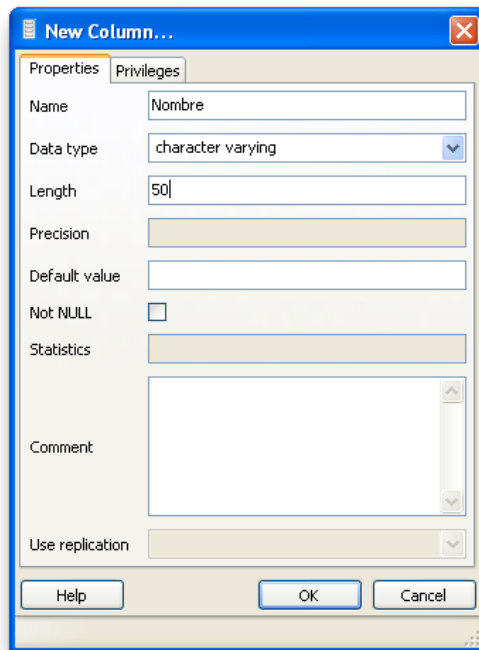


Figura 4.2.14 Ventana new columns

En la figura 4.2.15 se muestra las columnas agregadas.

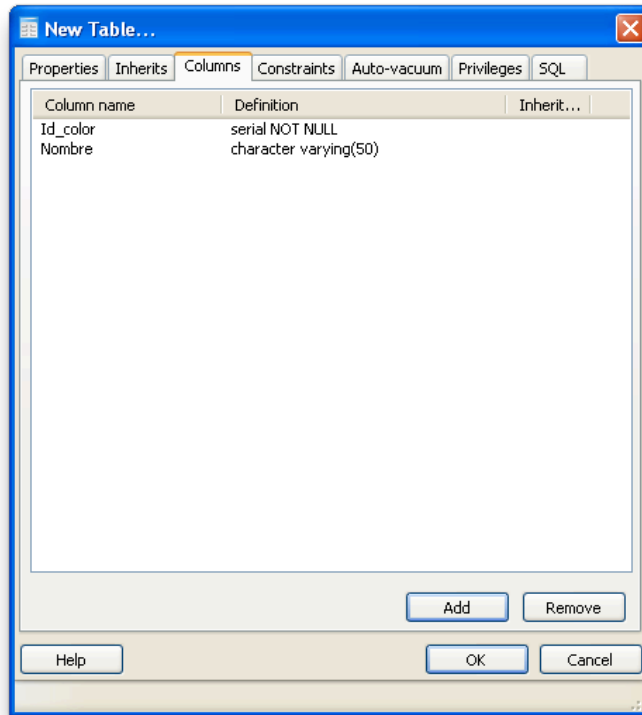


Figura 4.2.15 Columnas agregadas

Consultas a la base de datos:

Una vez que se crearon todas las tablas y se asignaron los valores default se implementan las consultas a la base de datos. Estas consultas muestran la información necesaria al usuario que opera el sistema.

Las siguientes consultas se realizaron después de haber agregado la información correspondiente en cada tabla.

A continuación se muestra una sentencia sql con la que se muestran los datos de los vehículos que están estacionados actualmente en el establecimiento.

```
select e.id_boleto,b.no_placa,a.nombre as "Vehiculo",  
c.nombre as "Marca",  
d.nombre as "Color"
```

```

from vehiculos_est as b inner join tipo_vehiculos as a on (b.id_vehiculo = a.id_vehiculo)
join marca as c on (b.id_marca = c.id_marca) join color as d on (b.id_color=d.id_color)
join boletos as e on (b.no_placa = e.no_placa) where e.id_tarifa=1 or e.id_tarifa=8

```

En la figura 4.2.16 se muestra dentro de pgAdmin el resultado de esta consulta.

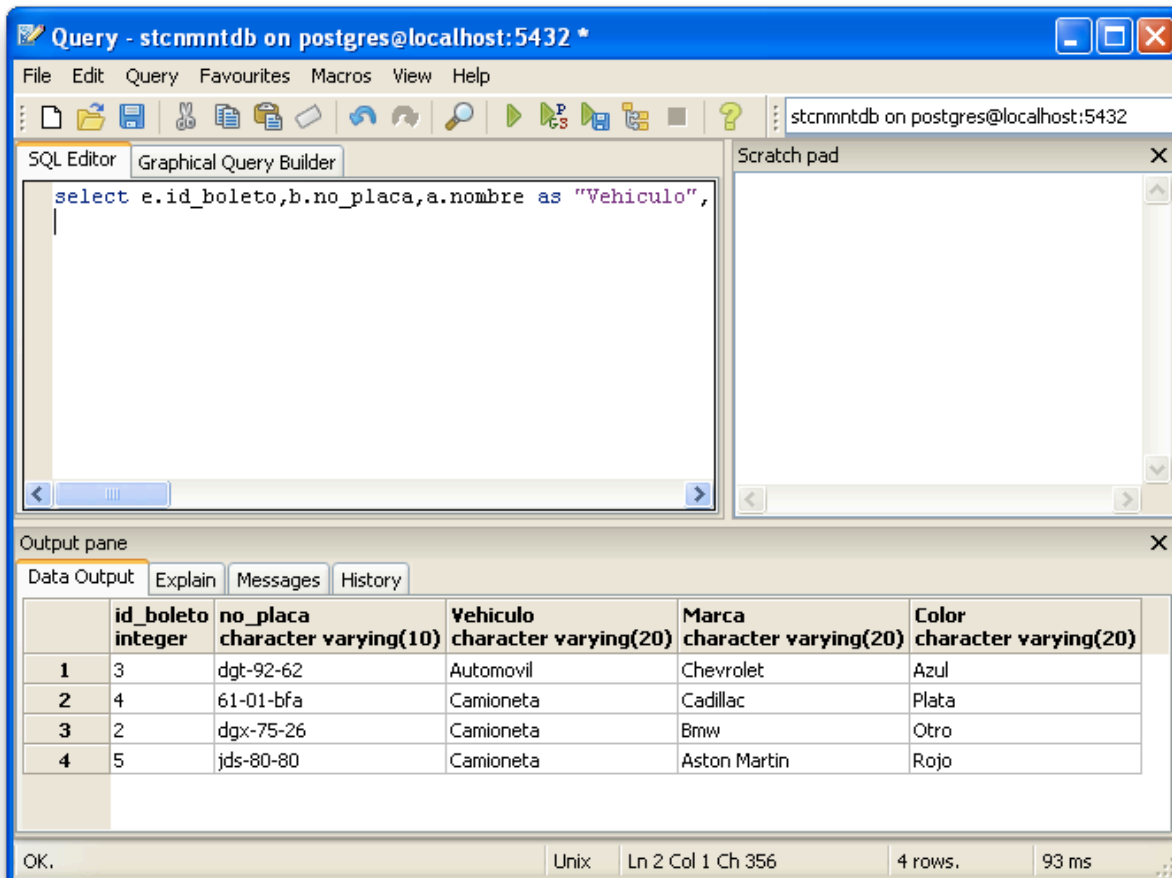


Figura 4.2.16 Consulta en pgAdmin

Otra consulta que se realiza es la de los boletos que aun son validos, para la salida de vehículo, o acceso o estacionamiento para los vehículos que tienen el servicio de pensión. A continuación se muestra:

```

select a.id_boleto as "Numero de Boleto",
a.no_placa as "Placas",

```

a.no_piso as "Piso",
a.no_cajon as "Cajon",
c.tiempo as "Tiempo",
c.horario as "Horario",
d.descripcion as "Servicio Adicional",
b.nombre||' '||b.paterno||' '||b.materno as "Operador Entrada",
a.fecha_hr_in as "Hora de Entrada" from boletos as a inner join empleados as b on
(a.operador_1=b.id_emp)join tarifa_vehiculos as c on (a.id_tarifa = c.id_tarifa) join
tarifa_otros as d on (a.id_otros = d.id_otros) where a.cancelado=false

En la figura 4.2.17 se muestra el resultado de esta consulta:

The screenshot shows the pgAdmin interface with a SQL query in the editor and its results displayed in the Output pane. The query is a complex join involving tables 'boletos', 'empleados', 'tarifa_vehiculos', and 'tarifa_otros'. The results table has 10 columns: 'Numero de Boleto', 'Placas', 'Piso', 'Cajon', 'Tiempo', 'Horario', 'Servicio Adicional', 'Operador Entrada', and 'Hora de Entrada'.

	Numero de Boleto integer	Placas character varying(10)	Piso integer	Cajon integer	Tiempo character varying(50)	Horario character varying(50)	Servicio Adicional character varying(50)	Operador Entrada text	Hora de Entrada timestamp without time zone
1	3	dgt-92-62	0	8	1 Hora	No Aplica	Lavado	Francisco Luna Luz	2012-06-19 22:18:39.562
2	2	dgx-75-26	0	7	1 Hora	No Aplica	Ningun Servicio	Francisco Luna Luz	2012-06-19 22:14:59.703
3	6	ybu-80-80	0	11	1 Semana	Nocturno	Ningun Servicio	Francisco Luna Luz	2012-06-19 00:00:00
4	5	jds-80-80	0	10	1 Hora	No Aplica	Lavado	Francisco Luna Luz	2012-06-19 22:46:02.14
5	4	61-01-bfa	0	9	1 Hora	No Aplica	Encerado	Francisco Luna Luz	2012-06-19 22:26:51.75

Figura 4.2.18 Resultado de consulta en pgAdmin

Las consultas se pueden realizar a través de la herramienta gráfica que ofrece pgAdmin. En las figura 4.2.19 y 4.2.20 se aprecia la captura de pantalla en el momento en que se realiza una consulta para obtener los datos de los vehículos que cuentan con algún servicio activo.

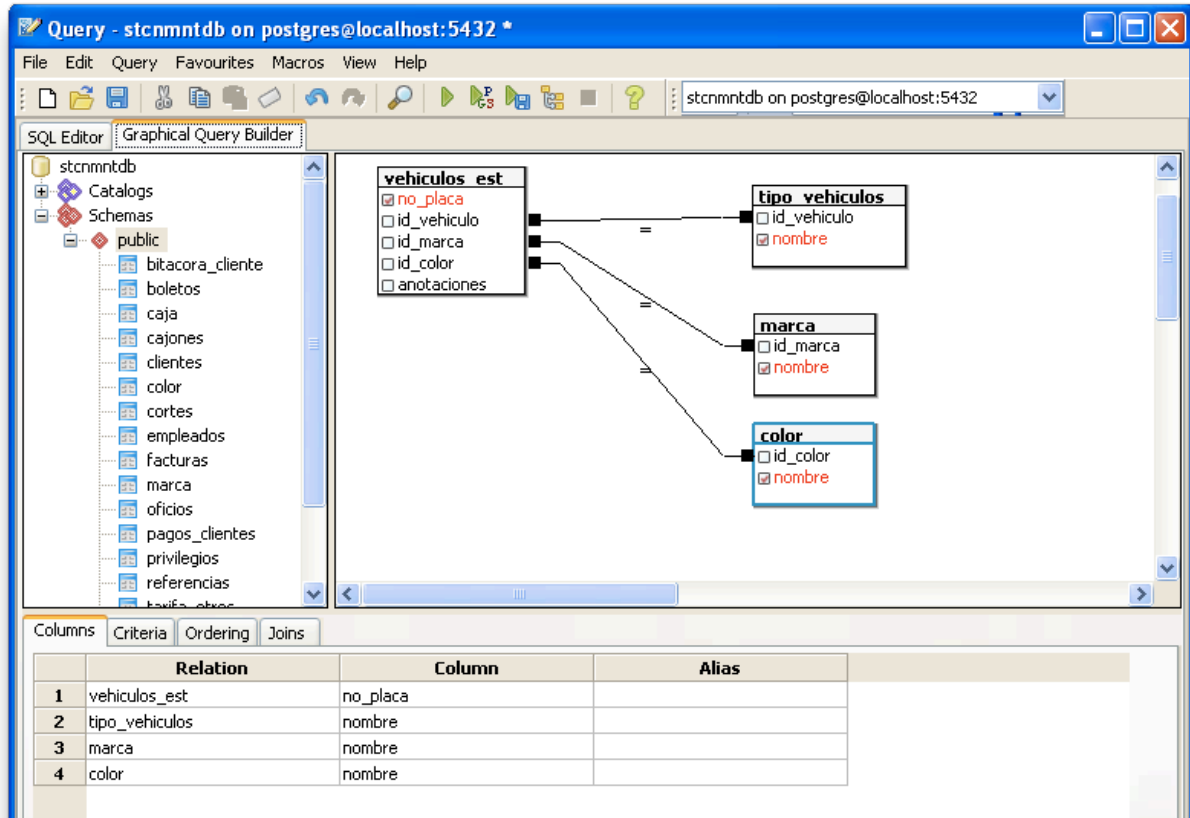


Figura 4.2.19 Consulta con herramienta gráfica

Query - stcnmtdb on postgres@localhost:5432 *

File Edit Query Favourites Macros View Help

stcnmtdb on postgres@localhost:5432

SQL Editor Graphical Query Builder Scratch pad

```

SELECT
vehiculos_est.no_placa,
tipo_vehiculos.nombre,
marca.nombre,
color.nombre
FROM
public.vehiculos_est,
public.tipo_vehiculos,
public.marca,
public.color
WHERE
vehiculos_est.id_vehiculo = tipo_vehiculos.id_vehiculo AND
vehiculos_est.id_marca = marca.id_marca AND
vehiculos_est.id_color = color.id_color;

```

Output pane

Data Output Explain Messages History

	no_placa character varying(10)	nombre character varying(20)	nombre character varying(20)	nombre character varying(20)
1	3456	Camioneta	Aston Martin	Blanco
2	jds-80-80	Camioneta	Aston Martin	Rojo
3	ybu-80-80	Automovil	Audi	Verde
4	tyhd-5466	Motocicleta	Audi	Gris
5	dgx-75-26	Camioneta	Bmw	Otro
6	61-01-bfa	Camioneta	Cadillac	Plata
7	dgt-92-62	Automovil	Chevrolet	Azul
8	zxcvbn	Camioneta	Citroen	Verde
9	456-mnb	Automovil	Lincoln	Rojo

OK. Unix Ln 15 Col 1 Ch 340 9 rows. 16 ms

Figura 4.2.20 Consulta sql generada

4.3 Diseño de interfaz de usuario

Para realizar la interfaz de usuario se pensó en la creación de una aplicación que siguiera el flujo de trabajo de los empleados que usarán el sistema pero a la vez solo se tenga permitido el acceso a cada parte del sistema dependiendo del tipo de usuario que accede.

Con esta información se decidió crear la aplicación con una estructura jerárquica, la cual es empleada en sitios corporativos e institucionales, ya que la mayoría de los usuarios encuentran esta estructura fácil de entender. Una organización jerárquica también impone al constructor de la interfaz una eficaz disciplina en su análisis de contenidos, ya que las jerarquías funcionan si el material está bien organizado.

El fin es conseguir un orden de menús y ventanas que el usuario perciba como natural.

En la figura 4.3.1 se muestra la estructura jerárquica de despliegue de ventanas

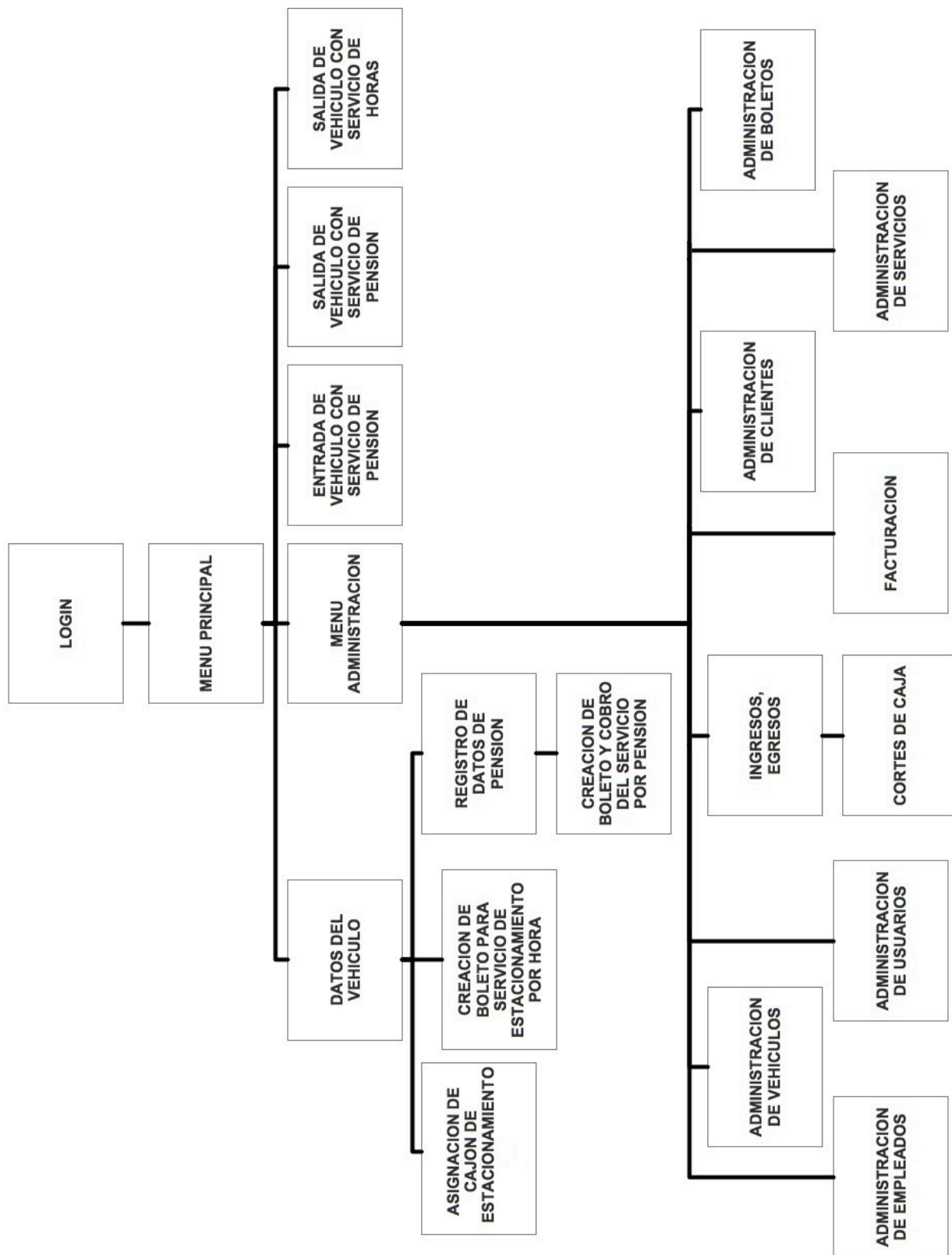


Figura 4.3.1 Estructura de despliegue de ventanas

Con este diagrama y la información recabada en la fase de requerimiento y diseño se dispone a crear la aplicación.

Npgsql

Antes de realizar la programación de la interfaz de usuario, fue necesario descargar el proveedor de datos Npgsql. (<http://pgfoundry.org/projects/npgsql/>). La apariencia del sitio web se muestra en la Figura 4.3.2

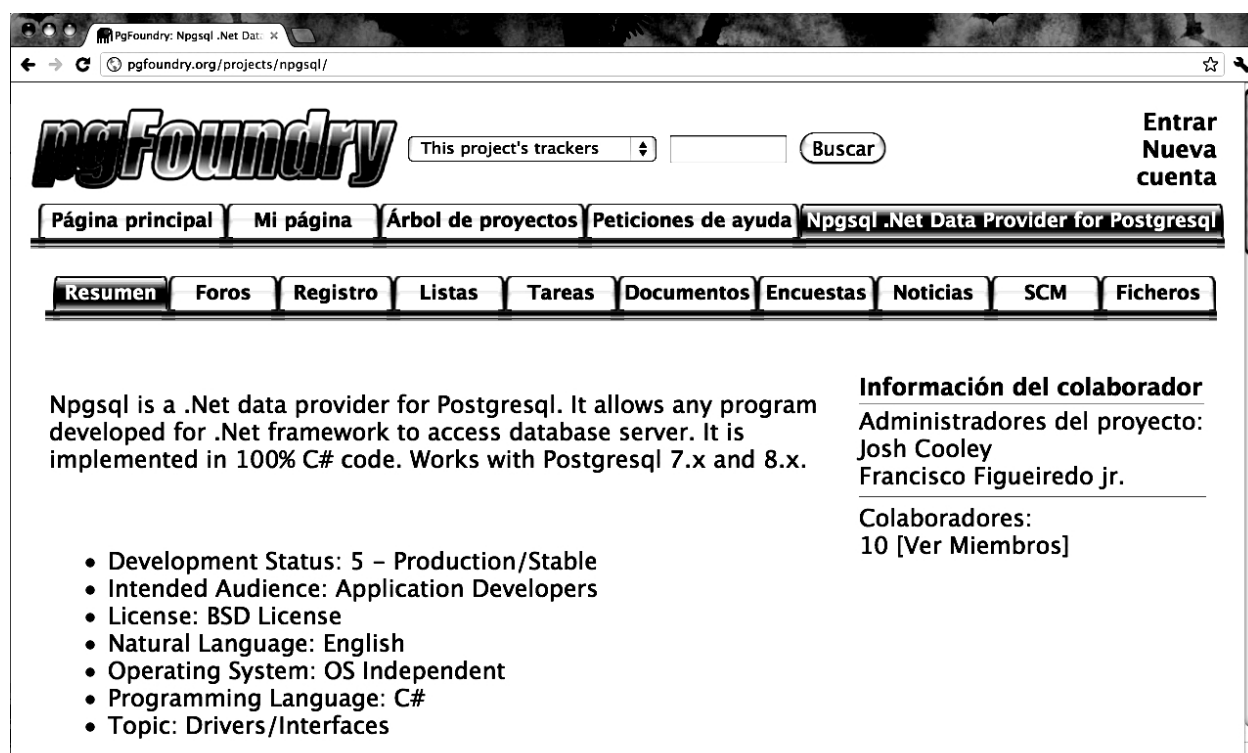


Figura 4.3.2 Pagina web de Npgsql

¹⁴Npgsql es una implementación de código abierto de un proveedor de datos .NET para C#. Se llevo a cabo completamente en C# y proporciona una interfaz directa con el protocolo de red que utiliza PostgreSQL. Permite el acceso a bases de datos

¹⁴ Neil Matthew y Richard Stones. *Beginning Databases with PostgreSQL From Novice to Professional*, Segunda Edición Ed. Apress pg.539

PostgreSQL, tanto local como a través de la red. Puede soportar todo tipo de proyectos, desde consola hasta Windows Forms.

La interfaz del sistema, desarrollado en Visual C#, para operar en ambiente Windows, posee las propiedades de éste y en consecuencia funciona por medio de una serie de ventanas, menús y submenús que facilita la operación del mismo.

Las ventanas son interfaces que permiten al usuario operar la información de la base de datos, en tanto que los menús y submenús son el medio por el que se llega a dichas ventanas.

Como parte del sistema se implementó la ventana Login de Usuario como lo muestra la Figura 4.3.3, la cual contiene los campos Id. del Empleado y Contraseña para la validación del usuario.

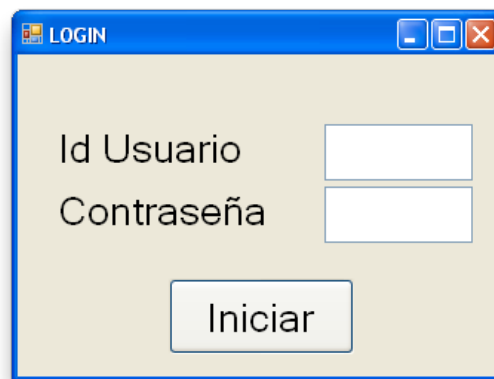


Figura 4.3.3 Ventana login de usuario

La segunda ventana que se despliega después de ser validado el usuario se llama Menú Principal, esta ventana es muy simple pero cumple con las necesidades del empleado que se encarga de la entrada y salida de vehículos. Véase figura 4.3.4



Figura 4.3.4 Ventana menú principal

A partir de esta ventana se desprenden varios caminos de acuerdo a la función que se quiere realizar.

El sistema cuenta con varias ventanas con las que se realizan distintas funciones en los sistemas, como registro de vehículos al acceder al estacionamiento, la administración de clientes, la administración de empleados, entre otros.

Por ejemplo el campo No. boleto tiene múltiples funciones por lo cual despliega distintas ventanas dependiendo del número del boleto.

Al introducir el número de boleto en el sistema este despliega la ventana que corresponde al tipo de servicio que se registro para el número de boleto introducido. Algunas de estas ventanas son:

Desplegar la ventana de cuota, esta ventana se muestra a los vehículo que ordenaron el servicio de estacionamiento por hora. Esta despliega la información referente a tiempo de estancia y el costo que se deberá cobrar pos los servicios. Véase figura 4.3.5

The image shows a software window titled "Cuota" with a blue title bar and standard Windows window controls. The window contains several input fields and a button. The fields are arranged as follows:

- No. Boleto:** A text box containing the number "6".
- Hora Entrada:** A text box containing "13/06/2012 01:15:04 p.m.".
- Hora Actual:** A text box containing "13/06/2012 01:15:19 p.m.".
- Cuota:** A text box containing "\$ 16".
- Serv. Adicional Cuota:** An empty text box.
- Total:** A text box containing "\$ 16".
- Cobrar:** A button located at the bottom right of the window.

Figura 4.3.5 Ventana cuota

Desplegar la ventana Entrada Pensión, esta ventana se muestra cuando se introduce un número de boleto que cuenta con el servicio de pensión. Muestra si tiene permitido el acceso en ese momento, dependiendo del horario que haya ordenado y se registra en la bitácora del cliente la entrada del vehículo.

En botón registrar de la ventana menú principal tiene la función de obtener los datos de los vehículos que no cuentan con servicio de estacionamiento. Véase figura 4.3.6

Nuevo Ingreso de Vehículo

Usuario Francisco Luna Luz Volver a Menu

No. Placa mdw-512-GD

Tipo de Vehículo Camioneta Discapitados Por Hora

Marca Dodge

Color Plata Pension

Asignar Cajón Nivel: PB Cajon:10

Figura 4.3.6 Ventana nuevo ingreso

Aquí el operador (usuario) introduce los datos del vehículo el cual se usará para el registro del servicio que ha solicitado el cliente (conductor).

En esta ventana se encuentran tres botones principales: uno es el de Asignar Cajón, este botón despliega el mapa del estacionamiento. La función de esta venta es la de mostrar gráficamente los cajones que pueden ser asignados al vehículo y al dar clic sobre alguno de estos se le asigna temporalmente y vuelve a la ventana nuevo Ingreso. Véase figura 4.3.7

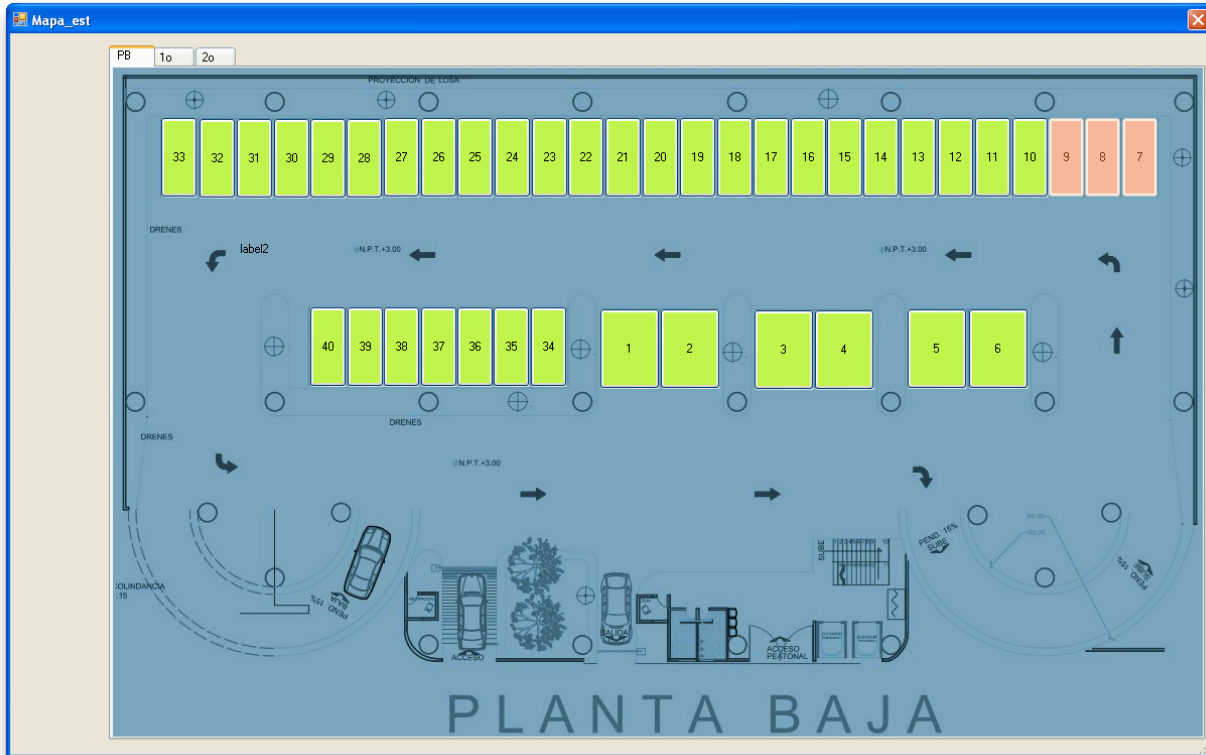


Figura 4.3.7 Ventana mapa estacionamiento

Los botones verdes son los cajones el cual el vehículo puede estacionarse, en cuanto a los rojos, ya están asignados a otros vehículos.

Posteriormente el empleado deberá elegir entre los botones Por hora, que es el servicio que se cobra por cuartos de hora o el botón Pensión que es el servicio de pensión.

La ventana Por_hora, la función de esta ventana es la de mostrar los datos y que el empleado compruebe si están bien los datos, al presionar el botón registrar, se imprime el boleto de estacionamiento con los datos necesario y hace el registro en el sistema. Véase figura 4.3.8.

The screenshot shows a software window titled "Por_hora" with a blue title bar and a close button in the top right corner. The window contains several input fields for recording vehicle entry data:

- No. Placa:** mdw-512-
- Vehiculo:** Camionet
- Marca:** Dodge
- Color:** Plata
- No. Boleto:** 6
- Lugar Cajon:** Nivel: PB Cajon:10
- Hora de Entrada:** 13/06/2012 01:14:15
- Servicio Adicional ?**: Lavado (selected in a dropdown menu)

A "Registrar" button is located in the bottom right corner of the window.

Figura 4.3.8 Ventana nuevo ingreso

La ventana Ajustes, esta ventana muestra otras funciones mas orientadas a la administraci3n de dinero o informaci3n del sistema. Es un men3 a estas funciones. V3ase figura 4.3.9

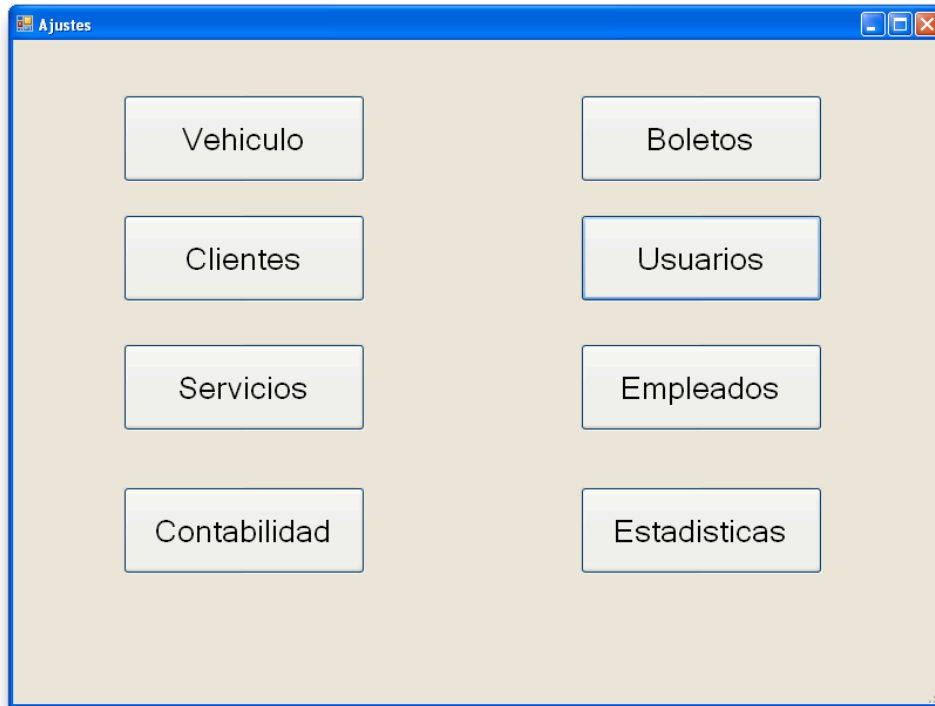


Figura 4.3.9 Ventana ajustes

El acceso a estas ventanas y sus funciones dependen de la clave y password de usuario. Para esto existe la ventana Usuarios.

La ventana Usuarios se encarga de administrar los empleados que pueden utilizar el sistema, ya que un usuario forzosamente debe ser un empleado para poder darlo de alta como usuario. En esta ventana se le asigna el acceso a ventanas o deniega, también se le asigna un password. La figura 4.3.10 muestra la ventana de Usuarios

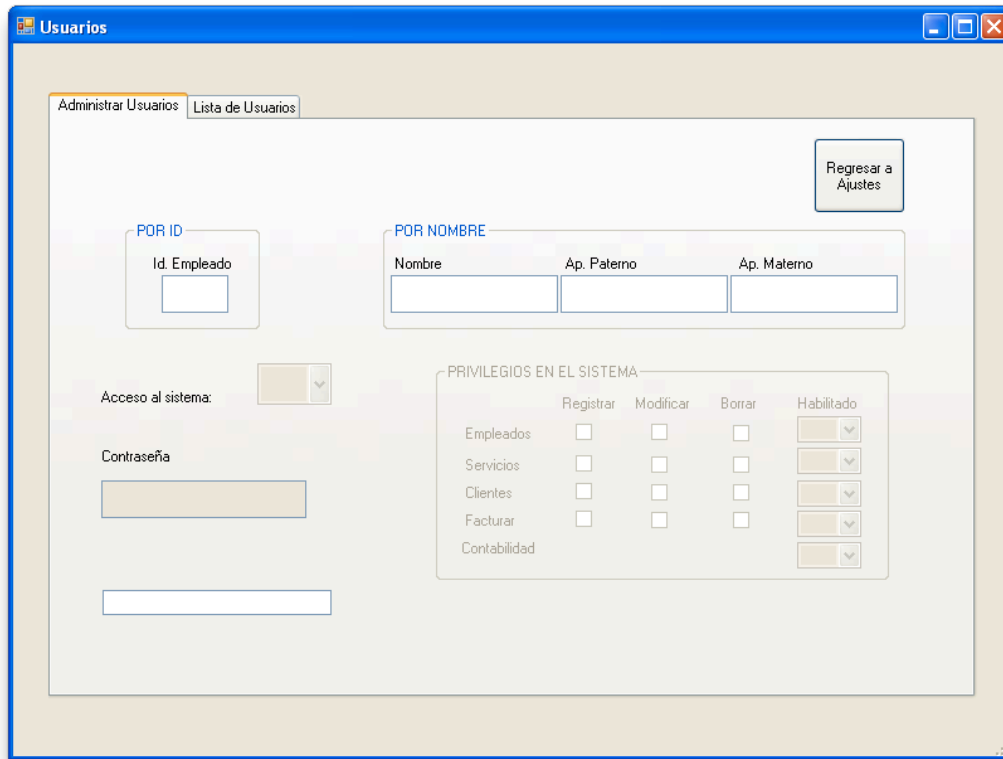


Figura 4.3.10 Ventana Usuarios

4.4 Generación de pruebas y mantenimiento

Las pruebas del software constituyen una parte vital del desarrollo del sistema, estos procesos permiten verificar y revelar la calidad de un producto de software. Se realizan con el propósito de descubrir defectos. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

A continuación se presentan los tipos de pruebas y las que se realizaron al sistema:

Prueba de caja negra:

La funcionalidad esperada es si el usuario ha introducido correctamente los datos al sistema para poder acceder o para poder ir al siguiente paso del proceso de registro.

En la figura se muestra la ventana de Login pero no se proporciono ningún dato para acceder por lo que se manda un mensaje de error.

Validación de usuario en la ventana Login.

En esta prueba se verifica que al no contener datos en los campos se despliega la ventana de error. En la figura 4.4.1 se muestra la prueba de validación.

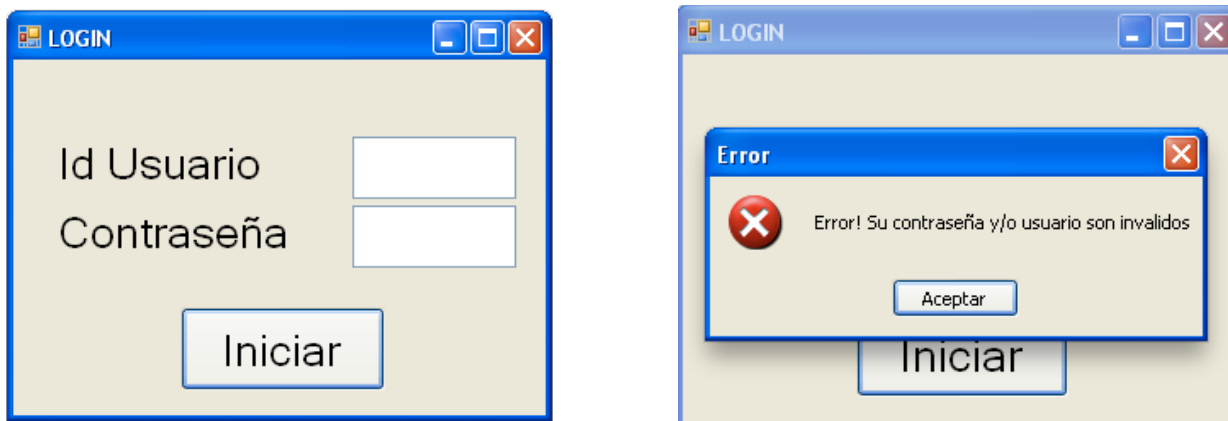


Figura 4.4.1 Validación en ventana Login

En la figura 4.4.2 se muestra la introducción de un Id de usuario incorrecto, también no cuenta con la contraseña por lo que manda el mismo mensaje de error.

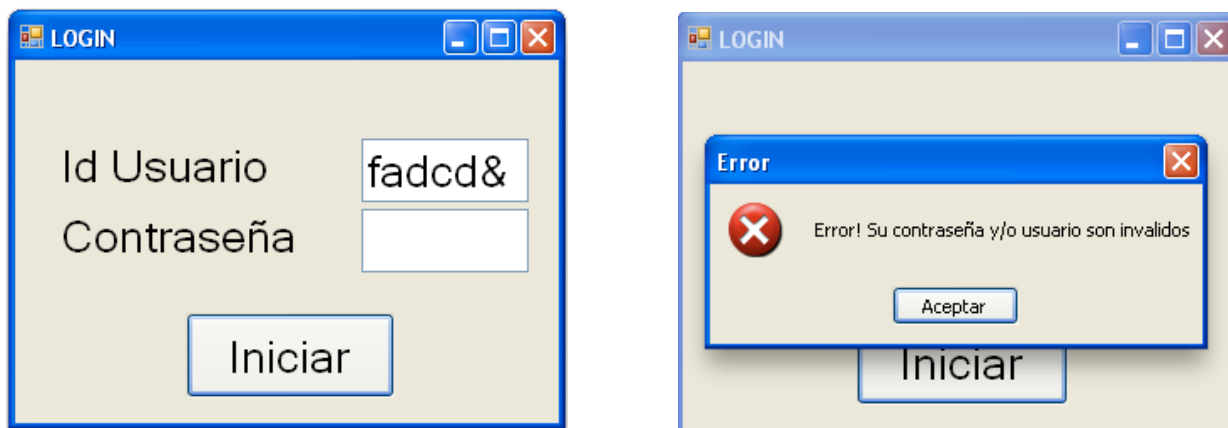


Figura 4.4.2 Validación en ventana Login

Al colocar correctamente los datos como se muestra en la Figura 4.4.3 que son los datos de un Usuario activo se despliegue la pantalla de Menú Principal (Figura 4.4.4) donde se muestra en la parte superior el nombre del usuario quien ingresos al sistema.

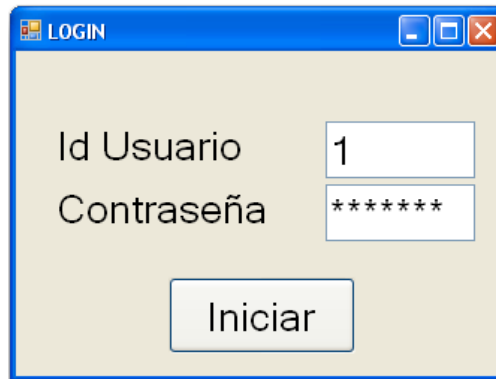


Figura 4.4.3 Ventana login con datos correctos



Figura 4.4.4 Ventana menú principal

Gracias a esta validación el sistema muestra solo las funciones que el usuario tiene permitido. En la figura se muestra el menú Ajustes donde se puede apreciar que está realizando la parte de seguridad, ya que solo están activos los botones a los que puede

acceder el usuario. En este caso (figura 4.4.5) el usuario que ingreso al sistema es el empleado que tiene como id usuario el valor de dos.

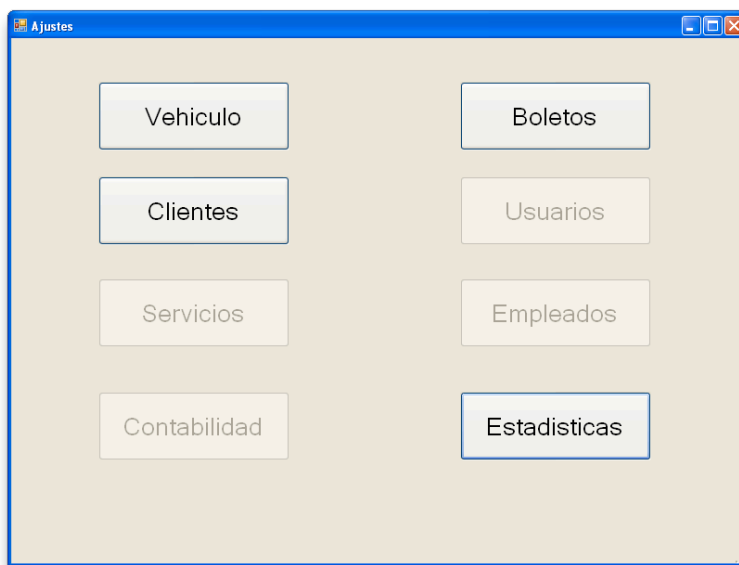


Figura 4.4.5 Ventana menú ajustes

La validación del campo No. Boleto

Lo que se hace es ingresar el numero de boleto, si el número de boleto o la información introducida no es válida se manda un mensaje al usuario. Esto se muestra en la figura 4.4.6, donde se introdujo un número de boleto el cual ha sido invalidado por el sistema, ya que este ya fue cobrado.

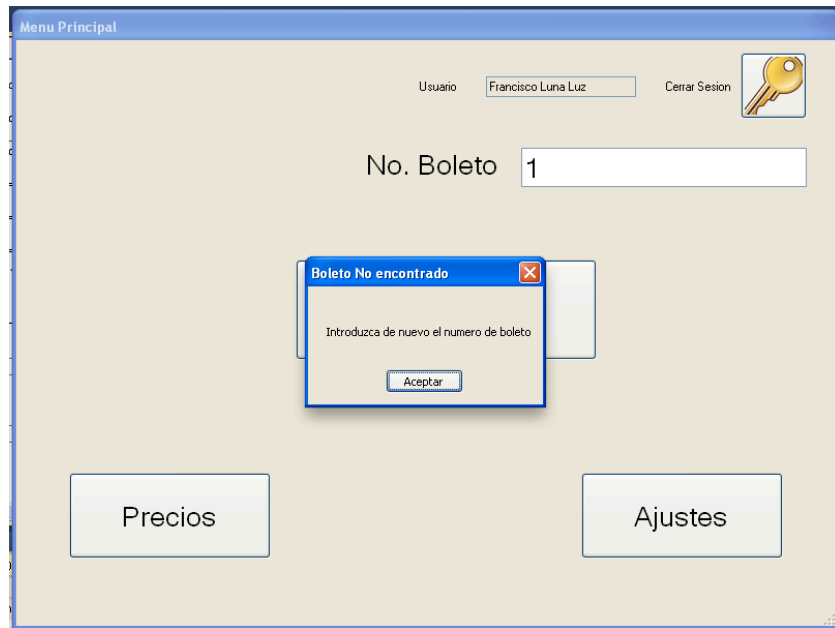


Figura 4.4.6 Validación del campo No Boleto

Validación de datos en registro de vehículo.

En este formulario se tiene que validar que todos los campos tengan asignado un valor antes de poder seguir con el registro, ya sea para el servicio de estacionamiento por hora o para el servicio de pensión. En la figura 4.4.7 se muestra un mensaje de error ya que el campo No Placa no contiene el dato.

En la figura 4.4.8 se muestra un mensaje de error ya que el campo Cajón ya que aun no se le ha asignado un lugar de estacionamiento

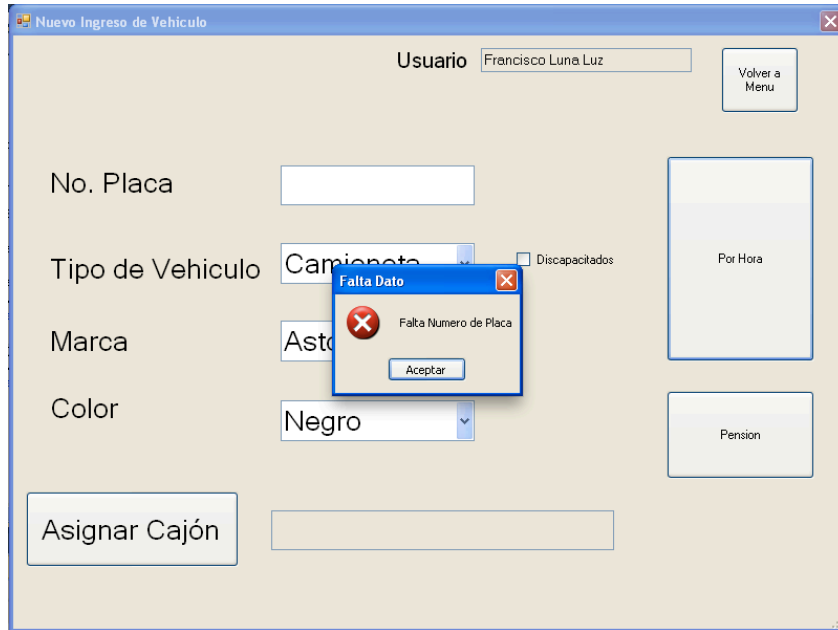


Figura 4.4.7 Validación de campos en ventana Nuevo Registro

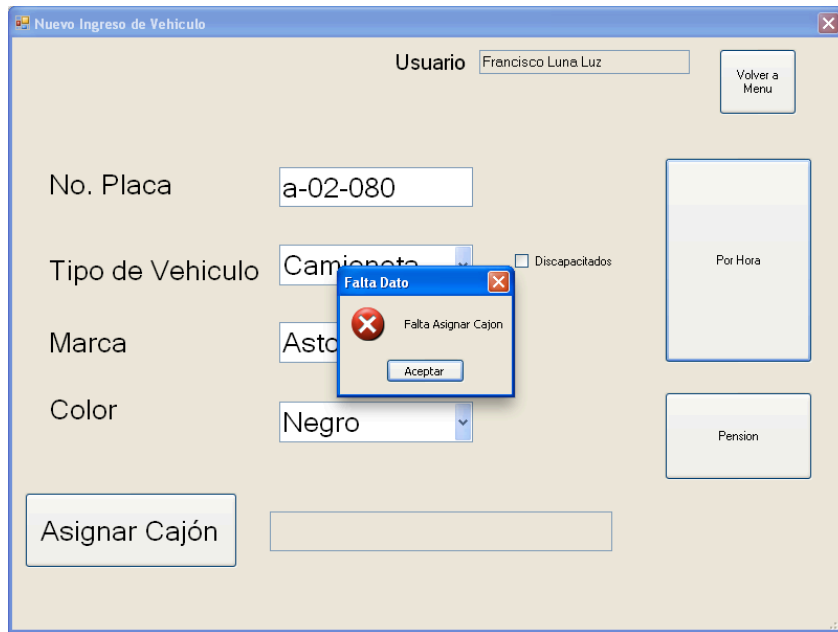
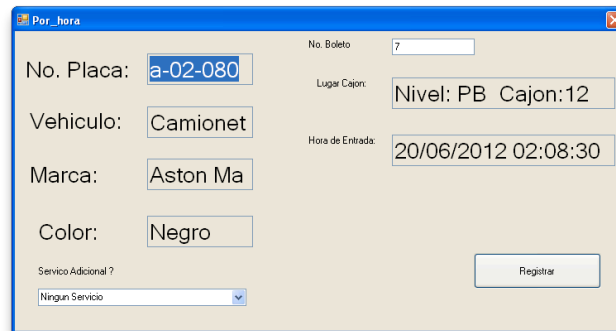


Figura 4.4.8 Validación del campo Cajón

Al tener totalmente los datos del formulario se puede pasar al siguiente paso, como ir a la ventana de servicio por hora, la cual muestra los datos ingresados para solo registrar y hacer la impresión del ticket. En la figura 4.4.9 se muestra que al tener todos los

campos completos se despliega la ventana llamada Por Hora donde se muestran los datos que fueron introducidos por el operador.



The screenshot shows a Windows application window titled 'Por_hora'. It contains several input fields and a button. The fields are: 'No. Placa:' with the value 'a-02-080', 'No. Boleto:' with the value '7', 'Vehiculo:' with the value 'Camionet', 'Lugar Cajon:' with the value 'Nivel: PB Cajon:12', 'Marca:' with the value 'Aston Ma', 'Hora de Entrada:' with the value '20/06/2012 02:08:30', and 'Color:' with the value 'Negro'. There is also a dropdown menu for 'Servicio Adicional?' with the selected option 'Ningun Servicio'. A 'Registrar' button is located at the bottom right of the window.

Figura 4.4.9 Ventana por Hora

Prueba de Caja Blanca:

Son pruebas estructurales. Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones.

Las Figuras 4.4.10 , 4.4.11 y 4.4.12 se muestra el código para la validación de datos del vehículo. Si estos están correctos podrá ingresar a la siguiente ventana.

```
//Solo permite Numeros,Letras y el guion '-' en el campo placas
private void txtInput_KeyPress(object sender, System.Windows.Forms.KeyPressEventArgs e)
{
    char keyChar;
    keyChar = e.KeyChar;

    if (!Char.IsDigit(keyChar) && !Char.IsLetter(keyChar) && keyChar != 45 && keyChar != 8)
    {
        e.Handled = true;
    }
}
```

Figura 4.4.10 Restricción de caracteres


```

private bool valida_campos_llenos() {

    if (textBox1.Text == "")
    {
        MessageBox.Show("Falta Numero de Placa", "Falta Dato", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }

    if (comboBox1.SelectedItem == null) {
        MessageBox.Show("Falta Asignar el Tipo de vehiculo", "Falta Dato", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    if (comboBox2.SelectedItem == null)
    {
        MessageBox.Show("Falta Asignar la Marca de Vehiculo ", "Falta Dato", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    if (comboBox3.SelectedItem == null)
    {
        MessageBox.Show("Falta Asignar el Color de Vehiculo", "Falta Dato", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    if (textBox_cajon.Text == "")
    {
        MessageBox.Show("Falta Asignar Cajon", "Falta Dato", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }

    return true;
}

```

Figura 4.4.11 Verifica campos de registro del vehículo

```

private void boton_hora_Click(object sender, EventArgs e)
{
    if (valida_campos_llenos() == true)
    {
        Por_hora frmPor_hora = new Por_hora();
        //Envia datos que se han elegido e introducido
        frmPor_hora.CerrarVentanaDelegateCallback = new CerrarVentanaDelegate(this.CerrarVentanaCallback);
        frmPor_hora.Datos_Registrados(textBox1.Text, comboBox1.SelectedItem.ToString(), comboBox2.SelectedItem.ToString());
        frmPor_hora.Datos_sesion(id_del_usuario, comboBox3.SelectedIndex + 1, comboBox2.SelectedIndex + 1);

        //Muestra el form de Por_hora
        frmPor_hora.ShowDialog();
    }
    else {

    }
}

```

Figura 4.4.12 Verifica si puede abrir la ventana de boleto por hora

Otras pruebas que se realizan a los sistemas son:

Pruebas de Estrés

Son utilizadas normalmente para someter a la aplicación al límite de su funcionamiento, mediante la ejecución de un número de usuarios muy superior al esperado.

Prueba de Regresión

Es volver a ejecutar un subconjunto de pruebas que se han llevado a cabo anteriormente para asegurarse de que los cambios no han propagado efectos no deseados.

Prueba Unitaria

Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada módulo individualmente funciona correctamente.

Prueba de integración

Se lleva a cabo durante el proceso de construcción del sistema una vez aprobadas las pruebas unitarias, involucra un número creciente de módulos y terminan probando el sistema en conjunto.

Pruebas Alfa

Estas pruebas consisten en invitar al cliente (quien ordenó el sistema) a probar el sistema, se trabaja en un entorno controlado y el cliente siempre está acompañado de un experto para ayudarle a utilizar el sistema y analizar los resultados.

Pruebas Beta

Estas pruebas son posteriores a las pruebas alfa y se desarrollan fuera de un entorno controlado. El cliente utiliza el sistema sin ninguna ayuda y trata de encontrar fallos para poder reportarlos al desarrollador.

Mantenimiento al Sistema

Esta fase tiene como objetivo principal corregir, mejorar y optimizar el software después de su entrega al usuario final.

Ya que se concluyo el sistema y que se han superado las pruebas e implementación solo queda el mantenimiento del sistema. A continuación se describen los tipos de mantenimiento aplicables a este sistema.

➤ Preventivo

El mantenimiento preventivo se realiza a partir del momento en el que se inicia a programación y se encuentra un defecto pero que aun no se presenta como falla en el sistema. Esto permite realizar las modificaciones necesarias y corregir el defecto antes de que se produzca el fallo.

➤ Correctivo

El mantenimiento correctivo tiene lugar cuando ocurre una falla o avería, se pone a consideración y se busca la causa de la falla, haciendo las correcciones necesarias.

➤ Perfectivo

Este tipo de mantenimiento se refiere a los cambios en la especificaciones, normalmente debido a cambios en los requerimientos del sistema. Puede cubrir desde

un cambio de formato de impresión de documentos hasta la incorporación de un nuevo módulo funcional.

➤ Adaptativo

Este tipo de mantenimiento se presenta cuando se realizan cambios en una porción del sistema y se requiere hacer cambio en otras secciones del programa para que pueda ser implementado correctamente. El mantenimiento adaptativo se hace para agregar un parámetro extra y no para corregir los defectos, es decir, permite adaptar el sistema a medida que evoluciona. Los cambios pueden ser como el cambio de sistema operativo a uno más moderno.

4.5 Generación de Reportes

Dentro del sistema el usuario puede generar documentos, estos documentos les son útiles para la toma de decisiones o el estado del estacionamiento en cuanto a ingresos o de vehículos en resguardo.

En el sistema de estacionamiento público se generan varios reportes los cuales se listan a continuación:

➤ Reporte de Vehículos

El venta Vehículos, esta ventana lista los vehículos por servicio (por hora, por pensión o ambos) además puede generar documentos pdf para su impresión. Esto sirve para tener un inventario de los vehículos que quedan en resguardo, y llevar un mejor control de los vehículos. Esto se muestra en las Figuras 4.5.1 y 4.5.2.

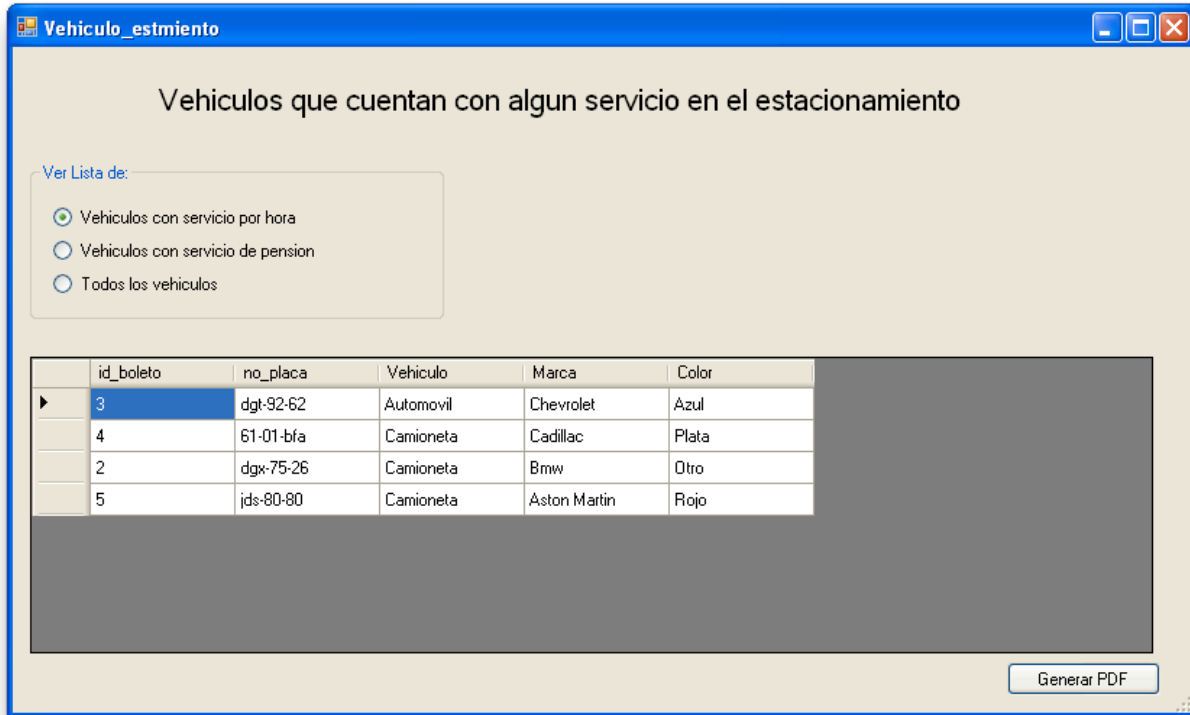


Figura 4.5.1 Ventana vehículos

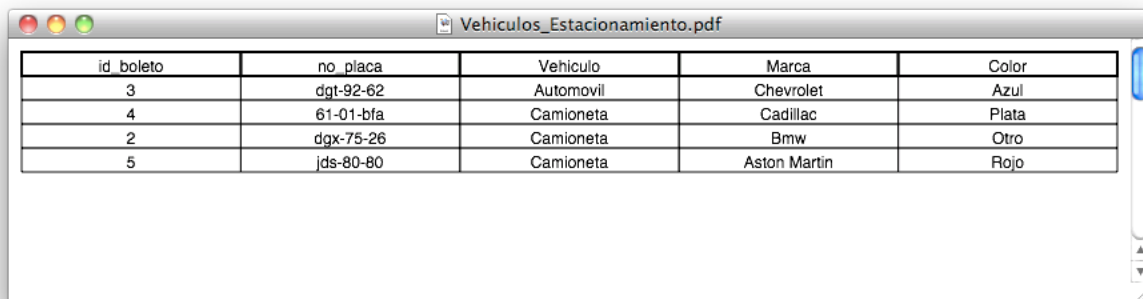


Figura 4.5.2 PDF de reporte de vehículos registrados por el servicio de hora

➤ Reporte de boletos

En la ventana boletos se lista todos los boletos que han sido pagados, o que aun están activos, como los boletos que no se cobran por el servicio de hora o los boletos que están asignados al servicio de pensión y no ha terminado su tiempo de contrato. Esta

ventana muestra datos muy importantes, ya que concentra la mayoría de la información, como el operador que atendió al cliente, la hora de entrada del vehículo, el o los servicios que contrato y el cajón que se le asigno entre otros.

Esta lista sirve al operador cuando el usuario pierde su boleto. Al igual que en la lista de vehículos puede generarse un pdf de estas listas. Esto se muestra en las figuras 4.5.3 y 4.5.4.

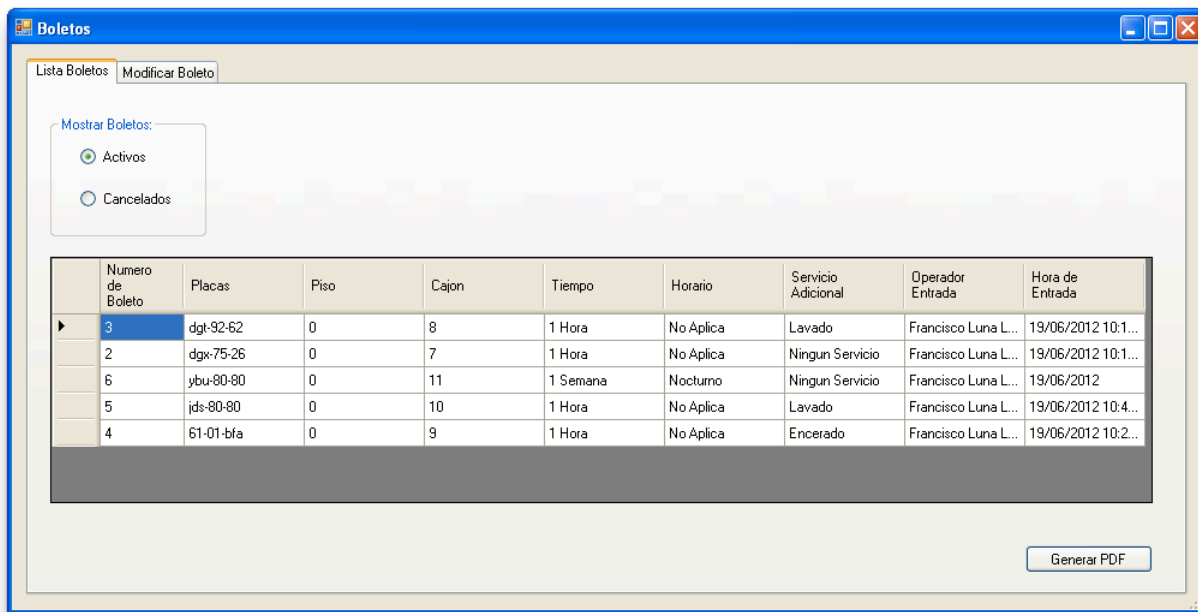


Figura 4.5.3 Ventana boletos

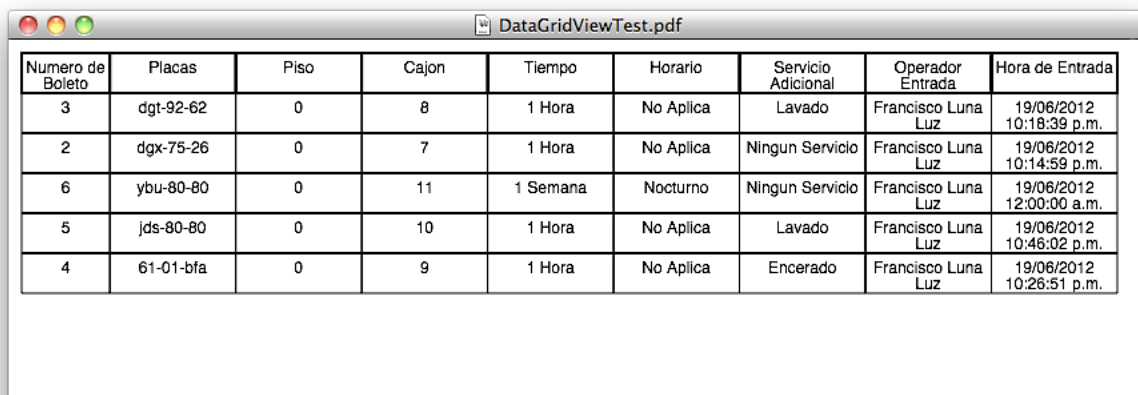


Figura 4.5.4 PDF reporte de boletos activos

➤ Reporte de estadísticas

La ventana Estadísticas muestra gráficas donde se representa la cantidad de personas que han contratado cada uno de los servicios, los cuales son: estacionamiento por hora, estacionamiento por pensión, lavado del vehículo y encerado del vehículo.

Este reporte le servirá al dueño para tener un mejor entendimiento de los servicios que más se piden en determinado tiempo, puede elegir el día, una semana o un mes determinado para el despliegue de la grafica. En la figura 4.5.5 se muestra la ventana Estadísticas.

También se puede generar un documento pdf para mayor comodidad del usuario.

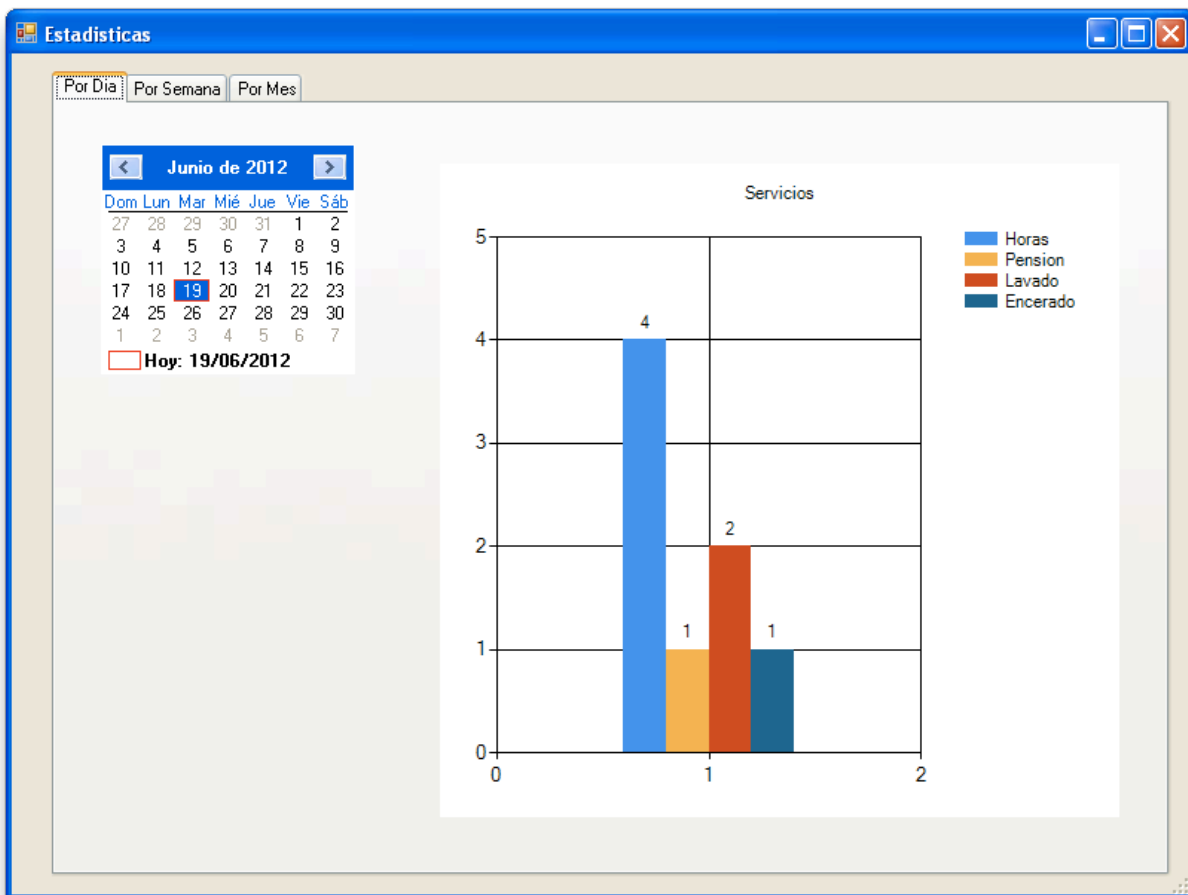


Figura 4.5.5 Ventana estadísticas

➤ Reporte de clientes

La ventana clientes además de realizar modificaciones a los registros de clientes muestra toda la información de todos los clientes registrados. En la figura 4.5.6 se muestra la ventana Clientes.



Figura 4.5.6 Ventana clientes

➤ Reporte de usuarios

La ventana Usuarios, además de dar de alta los empleados como usuarios, muestra una lista de todos los empleados que pueden utilizar el sistema. En la figura 4.5.7 se muestra la ventana Usuarios.

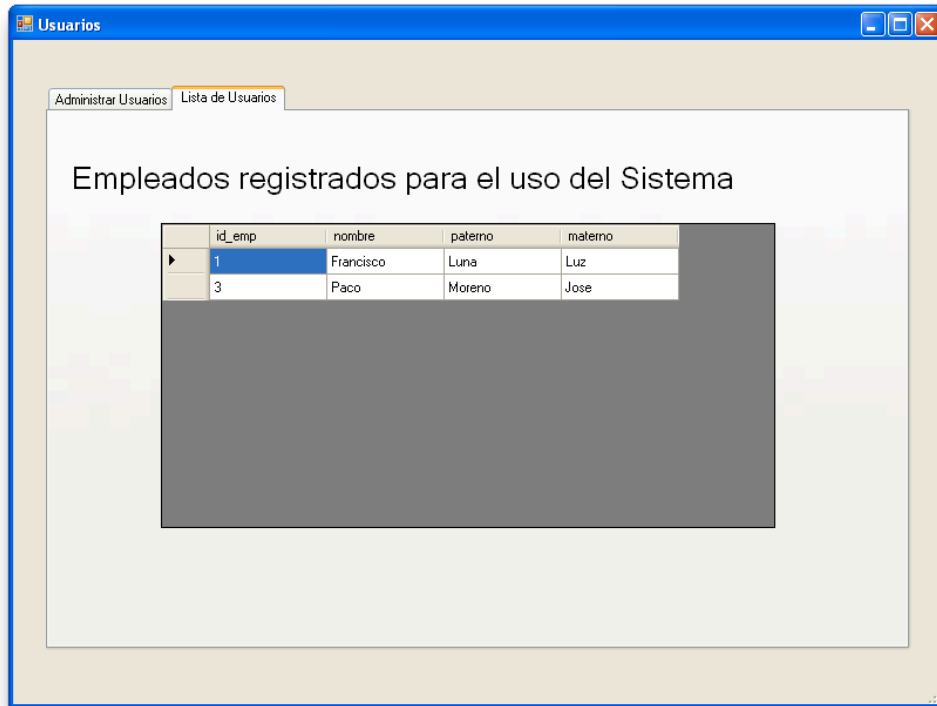


Figura 4.5.8 Ventana empleados

CONCLUSIONES

- Se desarrollo un sistema de software a fin de proveer al usuario una interfaz amigable para administrar su estacionamiento y hacer más eficiente la contabilidad de la empresa. Automatizando la asignación de lugares, realizando la reservación de espacios, la administración de horarios del servicio de pensión, el cálculo de tarifas, la administración de los clientes, la administración de los empleados, la administración de pagos y cobros, la administración de los vehículos, la administración de boletos y la administración de facturas.
- La importancia de crear y desarrollar un sistema es principalmente conocer las necesidades o problemática que presenta el cliente para poder brindarle la solución óptima a dichas necesidades.
- El sistema logra agilizar el análisis de información permitiendo simplicidad y rapidez en las consultas.
- Gracias a la creación del sistema se evitará almacenar una gran cantidad de papeles y se permitirá reducir el trabajo del personal al llevar el proceso de registro de la información a mano, dando como resultado la obtención de información precisa y actualizada en el momento que se requiere.
- Se cubren las necesidades de seguridad y respaldo de la información importante dentro de la base de datos.
- La implementación del sistema causará buen impacto en los usuarios ya que uno de los objetivos del sistema es el de registrar, almacenar y organizar de una manera rápida y eficiente toda la información de los vehículos, cobros y pagos de la empresa.

- La implementación de la metodología adecuada se utilizó con el fin de satisfacer las necesidades del cliente y nos guió durante todas las fases de su elaboración para concluir con éxito el proyecto.
- La utilización de herramientas visuales reduce el tiempo de desarrollo agilizando el proceso de entrega de la solución informática, además de que crea una interfaz amigable para el usuario que hace más fácil el manejo del sistema generando un menor tiempo de capacitación y por lo tanto, reducción en los gastos de la compañía.
- Respecto a los resultados esperados, se cubrieron satisfactoriamente las expectativas de desarrollo y desempeño del sistema, ya que se logró consolidar un sistema completo, práctico y funcional, listo para ser usado.

BIBLIOGRAFÍA

1. Mannino Michael V., *Administración de Bases de Datos. Diseño y Desarrollo de Aplicaciones*, 3a. Ed., Editorial: McGraw Hill.
2. Roger S. Pressman *Ingeniería del software*. Quinta edición 2010
3. Celma Giménez- Matilde Bases de Datos Relacionales, Ed, Prentice Hall. 2003
4. *C# para Desarrolladores de Java* Ed. McGraw-Hill, España 2003
5. Rueda Chacón Julio Cesar. Aplicación de la metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar J2EE. Tesis [Licenciatura]. Guatemala. Universidad de San Carlos de Guatemala. 2004 p. 133
6. Wille Christoph *C#* Pearson Madrid 2001
7. Neil Matthew y Richard Stones. *Beginning Databases with PostgreSQL From Novice to Professional*, Segunda Edición Ed. Apress
8. Teorey Toby J. *Database Modeling & Design* , Ed. Morgan Kauffman Publishers Third Edition 1999,
9. M. Ricardo Catherine, *Database Systems: Principles, design, and implementation*, Ed. Macmillan Publishing Company. 1990
10. Kruchten Philippe, *The Rational Unified Process an intrduction*, 2000
11. Inventario de emisiones de contaminantes criterio de la ZMVM 2008 [consulta 16 de junio 2012] Disponible en:
http://www.sma.df.gob.mx/sma/links/download/biblioteca/2008ie_criterio/2008ie_

criterio.pdf

12. <http://www.inegi.org.mx/Sistemas/DENUE/Presentacion.aspx>
13. García Soto Carlos E. y Segovia Amadeo. Pensiones para autos *Brújula de Compra* [en línea] 12-Noviembre-2010 [fecha de consulta 4 de Junio 2012] Disponible en:
http://www.profeco.gob.mx/encuesta/brujula/bruj_2010/bol183_pensiones.asp
14. Reglamento de estacionamientos públicos del Distrito Federal [en línea] Consultado en:
<http://www.coyoacan.df.gob.mx/transparencia/art14/i/reglamentos/trim1-2009/ESTACIONAMIENTOS.pdf>
15. Sánchez Jorge. Principios sobre Bases de Datos Relacionales, 2004 [fecha de consulta 4 de mayo 2012] Consultado en:
<http://www.jorgesanchez.net/bd/bdrelacional.pdf>
16. *PostgreSQL-es* [fecha de consulta 16 de junio 2012] en:
http://www.postgresql.org.es/sobre_postgresql
17. Diagrama de clase [en línea] [fecha de consulta 17 de junio 2012] de:
http://cadiit.anahuac.mx/~sac/download/52/ITI5002/p/DIAGRAMADECLASE_20090223.ppt
18. Modelo de clases [Fecha de consulta 17 de junio 2012] Consultado en:
<http://personales.dcc.uchile.cl/~psalinas/uml/modelo.html>
19. Modelo Entidad-Relación [Fecha de consulta 17 de junio 2012] Disponible en:
<http://gva1.dec.usc.es/~antonio/docencia/2004basesdedatos/teoria/ModeloEntidadRelacion.pdf>
20. Modelo de Clases [Fecha de consulta 17 de junio 2012] Disponible en:
<http://personales.dcc.uchile.cl/~psalinas/uml/modelo.htm>

21. Procesos de desarrollo: RUP, XP y FDD [Fecha de consulta 17 de junio 2012]
Disponible en:
<http://www.willydev.net/descargas/articulos/general/cualxpfddrup.pdf>

22. Modelado de Datos [Fecha de consulta 17 de junio 2012] Disponible en:
<http://ict.udlap.mx/people/carlos/is341/bases02.html>

➤ Bases de Datos Relacionales [fecha de consulta 20 de abril 2012] Disponible en:
<http://usuarios.multimania.es/cursosgbd/UD4.htm>

➤ Bases de Datos Relacionales [fecha de consulta 20 de abril 2012] Disponible en:
<http://www.jorgesanchez.net/bd/bdrelacional.pdf>

➤ RUP [fecha de consulta 6 de mayo 2012] Disponible en :
<http://jackopc.blogspot.mx/2007/05/metodologias-rup-y-xp-procesos-de.html>