



Proyecto Final – Manual Técnico

Ochoa Herrera Rodrigo Alejandro
CUCEI 2020-A | SSPBD

Índice

Introducción	3
Antecedentes	4
Definición del Caso de Estudio	5
Requerimientos	7
Diagrama ER	8
Diccionario de Datos	9
Script SQL	11
Conclusión	12
Bibliografía	13
Anexos	13

Introducción

Las bases de datos son una solución muy popular cuando de guardar información se trata. Nos permiten almacenar grandes cantidades de datos, de manera distintas de organización según nuestras necesidades. Una de las principales ventajas de utilizar bases de datos en una aplicación digital es la persistencia de datos centralizados, es decir, todos los datos son almacenados en una ubicación centralizada y no en dispositivos independientes. Con este esquema de centralización, podemos desarrollar aplicaciones con una gran escalabilidad.

La aplicación de esta tecnología en la vida diaria de un negocio puede facilitar mucho el trabajo si se emplea de la manera correcta. Por esto, en este proyecto decidí enfocar el desarrollo de la aplicación a un sistema que nos permite administrar entradas y salidas de un estacionamiento, así como diferentes usuarios que puedan acceder como empleados a la plataforma.

Antecedentes

La idea del proyecto nace al ver las deficiencias del sistema de cobro que maneja la mayoría de estacionamientos pequeños (no automatizados) en la zona centro de Guadalajara. En este sistema se genera un ticket manualmente cada vez que un vehículo entra, registrando la hora de ingreso. Al salir, se calcula el tiempo de estancia, así como el cobro (utilizando herramientas manuales como la calculadora, tablas de precios o realizando los cálculos mentalmente).

Las principales deficiencias que noto en este sistema es que no se tiene un control automatizado de ingresos/egresos, ni se puede tener un control de la información en caso de que se tengan varias sucursales.

Actualmente, las soluciones digitales que se tienen en estacionamientos más modernizados son tecnologías que funcionan, pero su aspecto visual no es el más amigable con el usuario.

Definición del Caso de Estudio

El principal objetivo de este proyecto es el desarrollo de una plataforma para la gestión de estacionamientos, proporcionándole acceso a los empleados en sitio para el registro de ingresos y egresos de vehículos, así como el acceso a la información de cada uno de los estacionamientos a un usuario administrador.

Como se mencionó en los antecedentes, las soluciones similares que existen en el mercado se llegan a sentir como tecnologías anticuadas al no tener una buena interfaz y experiencia de usuario. En este desarrollo, además de la robustez del programa, se enfatizará en el cuidado de la UI/UX de todo el programa.

Como requerimientos funcionales, se piden los siguientes puntos:

- Capacidad de generar tickets de ingreso/egreso de vehículos
- Capacidad de administrar uno o varios estacionamientos
- Capacidad de administrar cuentas de empleados y administradores
- Sistema de inicio de sesión (Autorización y Autenticación)
- Método de recuperación de cuentas (SMS/Email)
- Capacidad para agregar, editar y modificar:
 - Usuarios (Empleados y Administradores)
 - Estacionamientos
 - Vehículos

El programa deberá automatizar todo el proceso de ingreso y egreso de vehículos, generando tickets para cada situación. En el caso del ingreso, se pedirán los datos del vehículo y el modo en el que esta siendo estacionado (pensión/normal). Se generará el ticket de forma automática con la fecha actual del sistema. Para el ticket de egreso, se calculará el total a pagar a la fecha actual del sistema y el modo en el que ingresó el vehículo (precio pensión/normal).

Para el registro de entidades, es obligatorio que el programa pueda almacenar los siguientes campos de información. Campos que el desarrollador considere apropiados para la aplicación son opcionales, pero la información que se requiere para cada entidad es la siguiente:

- *Usuarios.* El programa debe tener la posibilidad de almacenar la siguiente información para cada cuenta de usuario.
 - Nombre de Usuario
 - Contraseña
 - Nombres
 - Apellidos
 - Email
 - Puesto en la Empresa
 - Salario Mensual
 - Permisos de Acceso (Administrador/Empleado)
 - Foto de Perfil

- *Estacionamientos.* El programa debe tener la posibilidad de almacenar la siguiente información para cada estacionamiento.
 - Nombre de Estacionamiento
 - Dirección Física
 - Teléfono
 - Capacidad de Lugares
 - Precio de Estacionamiento
 - Precio de Pensión

- *Vehículos.* El programa debe tener la posibilidad de almacenar la siguiente información para cada vehículo ingresado.
 - Placa
 - Marca
 - Modelo
 - Color

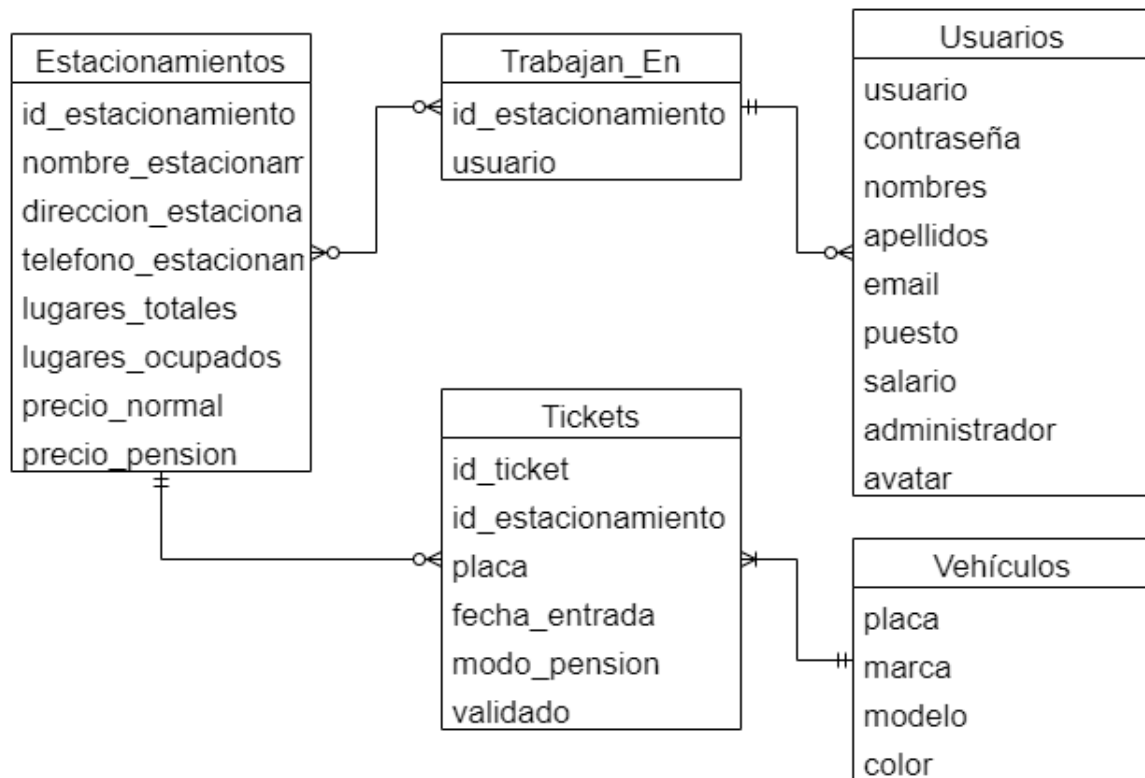
Requerimientos

Para reducir los costos de implementación de la plataforma, se requiere que el sistema pueda ejecutarse en sistemas de cómputo domésticos/comerciales, es decir, que no se requiera de hardware muy potente para ejecutar el programa. Los requisitos mínimos que se piden para la correcta ejecución son:

- Procesador de Dos Núcleos (1.0 GHz)
- 512 MB de Almacenamiento
- 2 GB de RAM
- Conexión a Internet*

*La conexión a internet únicamente se requiere para la recuperación de contraseña por medio de email, y para establecer la foto de perfil de cada cuenta de usuario.

Diagrama ER



Diccionario de Datos

Tabla Vehículos				
Nombre del Campo	Tipo de Dato	Tamaño	Descripción	Clave
placa	VARCHAR	10	Placa del Vehículo	PK
marca	VARCHAR	50	Marca del Vehículo	NN
modelo	VARCHAR	50	Modelo del Vehículo	NN
color	VARCHAR	25	Color del Vehículo	NN

Tabla Estacionamientos				
Nombre del Campo	Tipo de Dato	Tamaño	Descripción	Clave
id_estacionamiento	BIGSERIAL		Generado Automáticamente	PK
nombre_estacionamiento	VARCHAR	100	Nombre del Estacionamiento	NN
direccion_estacionamiento	VARCHAR	200	Dirección del Estacionamiento	NN
telefono_estacionamiento	VARCHAR	20	Teléfono del Estacionamiento	NN
lugares_totales	INTEGER		Capacidad de Lugares	NN
lugares_ocupados	INTEGER		Autos estacionados	NN
precio_normal	DECIMAL		Precio de estacionamiento	NN
precio_pension	DECIMAL		Precio de pensión	NN

Tabla Usuarios				
Nombre del Campo	Tipo de Dato	Tamaño	Descripción	Clave
usuario	VARCHAR	20	Nombre de Usuario	PK
contraseña	VARCHAR	20	Contraseña de Usuario	NN
nombres	VARCHAR	50	Nombres del Usuario	NN
apellidos	VARCHAR	50	Apellidos del Usuario	NN
email	VARCHAR	100	Email registrado. Sirve para la recuperación de contraseña.	NN
puesto	VARCHAR	100	Puesto dentro de la empresa	NN
salario	DECIMAL		Salario mensual	NN
administrador	BOOLEAN		Define si tiene permisos de administrador	NN
avatar	VARCHAR	200	Almacena la URL de la foto de perfil cargada con el API imgBB	

Tabla Tickets				
Nombre del Campo	Tipo de Dato	Tamaño	Descripción	Clave
id_ticket	BIGSERIAL		Generado Automáticamente	PK
id_estacionamiento	SERIAL		Referencia al estacionamiento donde se generó el ticket	FK
placa	VARCHAR	10	Referencia del vehículo que generó el ticket	FK
fecha_entrada	TIMESTAMP WITHOUT TIME ZONE		Almacena la fecha en el formato DD/MM/AAAA HH:MM:SS	NN
modo_pension	BOOLEAN		Define si se ingresó el vehículo en modo pensión o en modo normal	NN
validado	BOOLEAN		Define si el ticket ya fue cobrado o sigue pendiente	NN

Tabla Trabajan_En				
Nombre del Campo	Tipo de Dato	Tamaño	Descripción	Clave
id_estacionamiento	SERIAL		Referencia al estacionamiento donde trabaja el usuario	FK
usuario	VARCHAR	20	Referencia al usuario	FK

Script SQL

```
CREATE TABLE vehiculos
(
    placa VARCHAR(10) PRIMARY KEY,
    marca VARCHAR(50) NOT NULL,
    modelo VARCHAR(50) NOT NULL,
    color VARCHAR(25) NOT NULL
);

CREATE TABLE estacionamientos
(
    id_estacionamiento BIGSERIAL PRIMARY KEY,
    nombre_estacionamiento VARCHAR(100) NOT NULL,
    direccion_estacionamiento VARCHAR(200) NOT NULL,
    telefono_estacionamiento VARCHAR(20) NOT NULL,
    lugares_totales INTEGER NOT NULL,
    lugares_ocupados INTEGER NOT NULL,
    precio_normal DECIMAL NOT NULL,
    precio_pension DECIMAL NOT NULL
);

CREATE TABLE usuarios
(
    usuario VARCHAR(20) PRIMARY KEY,
    contraseña VARCHAR(20) NOT NULL,
    nombres VARCHAR(50) NOT NULL,
    apellidos VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL,
    puesto VARCHAR(100) NOT NULL,
    salario DECIMAL NOT NULL,
    administrador BOOLEAN NOT NULL,
    avatar VARCHAR(200)
);

CREATE TABLE tickets
(
    id_ticket BIGSERIAL PRIMARY KEY,
    id_estacionamiento SERIAL REFERENCES estacionamientos (id_estacionamiento) ON DELETE CASCADE,
    placa VARCHAR(10) REFERENCES vehiculos (placa) ON UPDATE CASCADE ON DELETE CASCADE,
    fecha_entrada TIMESTAMP WITHOUT TIME ZONE NOT NULL,
    modo_pension BOOLEAN NOT NULL,
    validado BOOLEAN NOT NULL
);

CREATE TABLE trabajan_en
(
    id_estacionamiento SERIAL REFERENCES estacionamientos (id_estacionamiento) ON DELETE CASCADE,
    usuario VARCHAR(20) REFERENCES usuarios (usuario) ON UPDATE CASCADE ON DELETE CASCADE
);
```

Conclusión

El desarrollo de la aplicación fue mucho más complejo de lo que inicialmente tenía contemplado. Algunas operaciones tardaron más en desarrollarse que otras debido al poco conocimiento de algunas sentencias y funcionamiento general de las bases de datos. Simples detalles como olvidar definir las sentencias ON UPDATE ON DELETE en el script SQL en las llaves foráneas hizo que el desarrollo de las operaciones CRUD de todas las entidades se retrasaran.

Además de aprender cosas sobre SQL y bases de datos, desarrolle otras habilidades en el manejo de API's para complementar el funcionamiento del programa. Como, por ejemplo, el almacenar imágenes en la base de datos es un proceso un tanto complicado, además de que aumenta considerablemente su peso. Por eso decidí implementar otro método el cual almacena las imágenes en la nube utilizando un API, y guardando únicamente el URL en la base de datos.

Para la funcionalidad de recuperación de contraseñas, utilicé otro API para el envío de correos electrónicos, con esto, fue posible enviar al usuario un código de recuperación por medio del correo electrónico registrado en su cuenta.

En resumen, siento que este proyecto fue un verdadero reto personal para mí, en el cual siento que el resultado final es una plataforma de calidad, con funcionalidades añadidas que aportaron a mi conocimiento sobre otras tecnologías.

Bibliografía

Connolly, T. M., & Begg, C. E. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson Education.

Pérez, González y Moreno (2014)). Manejo de bases de datos con MySQL y PostgreSQL. México: TRAUCO

Ramez, Elmasri & Shamkant B., Navathe (2011). Sistemas de bases de datos. México: Pearson.

Silverschatz, Abraham (2011). Fundamentos de Bases de Datos. México. McGrawn Hill

Documentación Oficial de PostgreSQL

Disponible en: www.postgresql.org/docs/12/index.html

Documentación Oficial de SendGrid

Disponible en: <https://sendgrid.com/docs/for-developers/sending-email/v3-csharp-code-example/>

Anexos

Repositorio GitHub del Proyecto: <https://github.com/ROALOGH/parkDB>