

## **Parque Salitre Mágico: Aplicación CRUD**

Me complace presentar la aplicación que desarrollé para "Parque Salitre Mágico", un parque de atracciones que ahora cuenta con una innovadora solución MVC. Este proyecto, en el que trabajé de manera individual, integra un completo sistema CRUD diseñado con principios de Programación Orientada a Objetos (POO) y respaldado por el manejo eficiente de GIT y bases de datos relacionales y/o no relacionales.

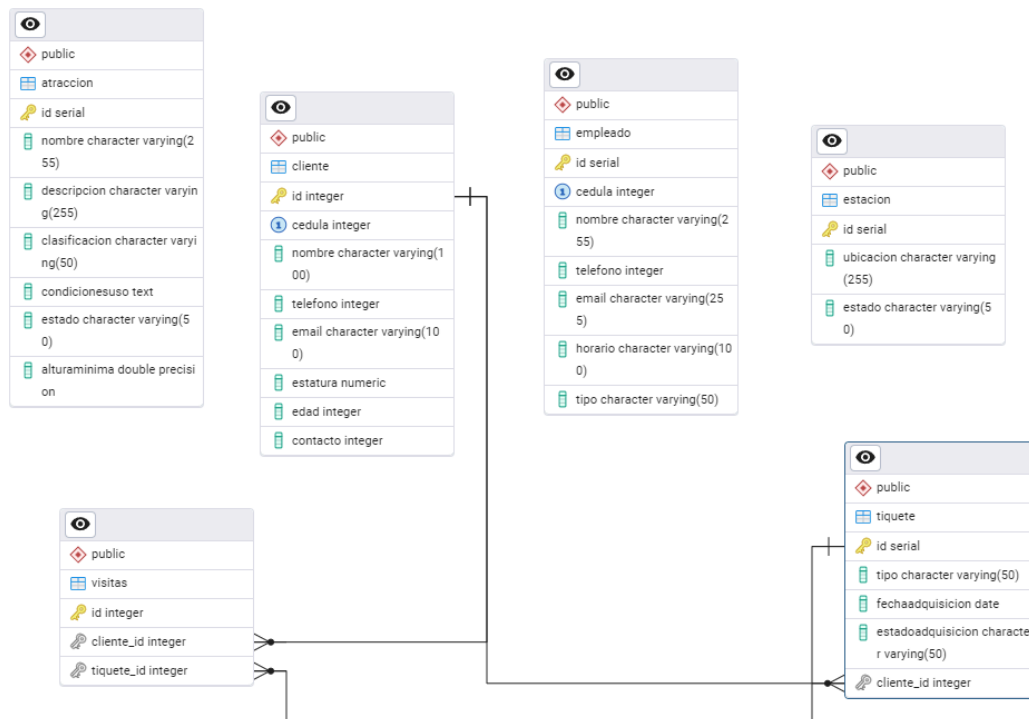
La aplicación permite gestionar y analizar información clave sobre el uso de las atracciones del parque. Entre sus funcionalidades destacan la administración de los tres tipos de pasaportes disponibles y el control de acceso a las atracciones, considerando las restricciones de estatura. Este proyecto no solo optimiza la experiencia de los visitantes, sino que también proporciona al parque herramientas valiosas para la toma de decisiones basada en datos.

### **Tecnologías usadas**

- **Base de datos relacional:** PostgreSQL
- **Lenguaje de programación:** Java
- **Patrón de diseño y paradigma de programación:** MVC y POO
- **Control de versiones:** GIT y GitHub
- **Ejecución:** La aplicación se ejecuta en la interfaz gráfica de escritorio integrada en el IDE, en este caso se utilizó **NetBeans**

## Diagrama Entidad-Relación

Se creó la tabla visitas con el objetivo de brindar al personal de publicidad información sobre los clientes que más visitan el parque.



## Tablas y Atributos

### 1. atraccion:

- id (serial, clave primaria)
- nombre (varchar(255))
- descripcion (varchar(255))
- clasificacion (varchar(50))
- condicionesuso (text)
- estado (varchar(50))
- alturaminima (double precision)

### 2. cliente:

- id (serial, clave primaria)
- cedula (integer)
- nombre (varchar(100))
- telefono (integer)
- email (varchar(100))
- estatura (numeric)

- edad (integer)
- contacto (integer)

### 3. empleado:

- id (serial, clave primaria)
- cedula (integer)
- nombre (varchar(255))
- telefono (integer)
- email (varchar(255))
- horario (varchar(100))
- tipo (varchar(50))

### 4. estacion:

- id (serial, clave primaria)
- ubicacion (varchar(255))
- estado (varchar(50))

### 5. tiquete:

- id (serial, clave primaria)
- tipo (varchar(50))
- fechaadquisicion (date)
- estadoadquisicion (varchar(50))
- cliente\_id (integer, clave foránea que referencia a cliente.id)

### 6. visitas:

- id (serial, clave primaria)
- cliente\_id (integer, clave foránea que referencia a cliente.id)
- tiquete\_id (integer, clave foránea que referencia a tiquete.id)

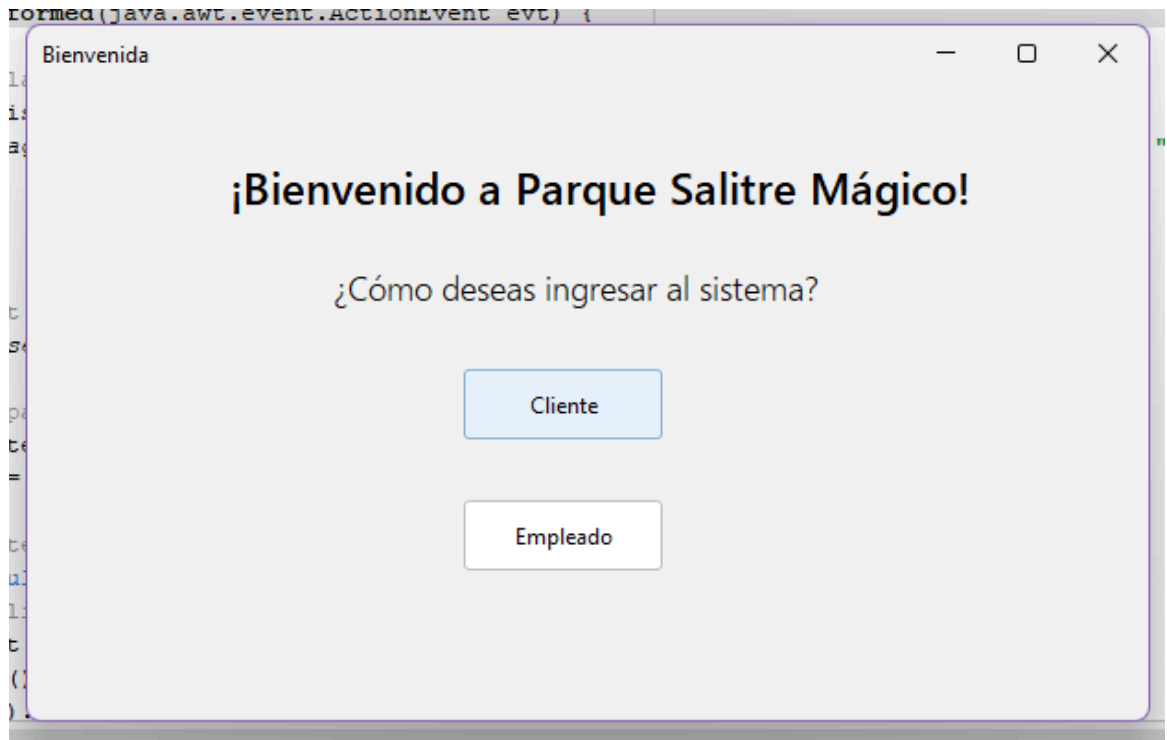
## Relaciones

Las relaciones entre las tablas se representan con líneas que conectan las claves foráneas con las claves primarias correspondientes.

- **cliente** tiene uno o más tiquetes. Esto se refleja mediante la clave foránea **cliente\_id** en la tabla tiquete.
- **tiquete** tiene una relación con visitas. La tabla visitas tiene la clave foránea **tiquete\_id** que referencia a **tiquete.id**.
- **cliente** tiene una relación con visitas. La tabla visitas tiene la clave foránea **cliente\_id** que referencia a **cliente.id**.

## Funcionalidades Parque de Diversiones

1. La aplicación cuenta con una ventana de inicio, en la cual el usuario escoge si ingresa como **Cliente** o como **Empleado**



2. Se creó la clase **Persona** con los atributos id, cedula, nombre, telefono y email. De esta clase se hereda la clase **Cliente**, con los atributos estatura, edad y contacto. Si el cliente es menor de edad, debe ingresar el número de cédula de un contacto, en caso de no hacerlo, el sistema mostrará un mensaje de advertencia y lanzará una excepción. Si el cliente ya ha estado antes en el parque, sólo deberá ingresar su cédula. Al hacer click en el botón “**Validar datos**” el sistema hace una consulta para obtener los datos de este o registrarlo en caso de que no haya estado antes en el parque, se le asignará un **Tiquete** con estado **Activo**, los tiquetes cuentan con 3 tipos (Estándar, Premium y Exclusivo) y se mostrará un mensaje con el número de tiquete asignado. Cuando el cliente hace click en el botón “**Ingresar al Parque**” se registrará una **Visita** con el **Tiquete** y el **Cliente** asociados. Sólo se mostrarán las **Estaciones** que ya han sido habilitadas por un empleado **Administrativo**.

Por ejemplo, este cliente es menor de edad

The screenshot shows a web form titled "Ingreso al Parque" with a message dialog box overlaid. The dialog box, titled "Message", contains an information icon and the text: "Si eres menor de edad, debes ingresar el número de cédula de tu acompañante". The form has two main sections: "Ingresa tus datos" and "Escoge una estación". The "Ingresa tus datos" section includes fields for Cédula (2839873), Nombre (Julian Molina), Teléfono (3446443), Email (julian@example.com), Estatura (1.50), Edad (12), and Contacto. A note states: "\*Obligatorio si es menor de 18 años". The "Escoge una estación" section has a dropdown menu set to "Punto Central". The "Escoge el tipo de tiquete" section has a dropdown menu set to "Estandar". There are buttons for "Validar Datos", "Ingresar al Parque", and "Atrás".

Message

Si eres menor de edad, debes ingresar el número de cédula de tu acompañante

OK

Atrás

Ingresa tus datos

Si ya has estado antes en el parque sólo ingresa tu cédula

Cédula 2839873

Nombre Julian Molina

Teléfono 3446443

Email julian@example.com

Estatura 1.50

Edad 12

Contacto

\*Obligatorio si es menor de 18 años

Validar Datos

Escoge una estación

Punto Central

Escoge el tipo de tiquete

Estandar

Ingresar al Parque

Al añadir un contacto, el cliente se registra correctamente

The screenshot shows the "Ingreso al Parque" form after successful registration. The "Contacto" field now contains the number "8737474". The "Validar Datos" button is highlighted. The "Escoge el tipo de tiquete" section now displays a message: "¡Gracias por visitar el Parque Salitre Mágico Julian Molina! Se te asignó el tiquete número 16 úsalo para ingresar a las atracciones." The "Ingresar al Parque" button is also visible.

Ingreso al Parque

Atrás

Ingresa tus datos

Si ya has estado antes en el parque sólo ingresa tu cédula

Cédula 2839873

Nombre Julian Molina

Teléfono 3446443

Email julian@example.com

Estatura 1.50

Edad 12

Contacto 8737474

\*Obligatorio si es menor de 18 años

Validar Datos

Escoge una estación

Punto Central

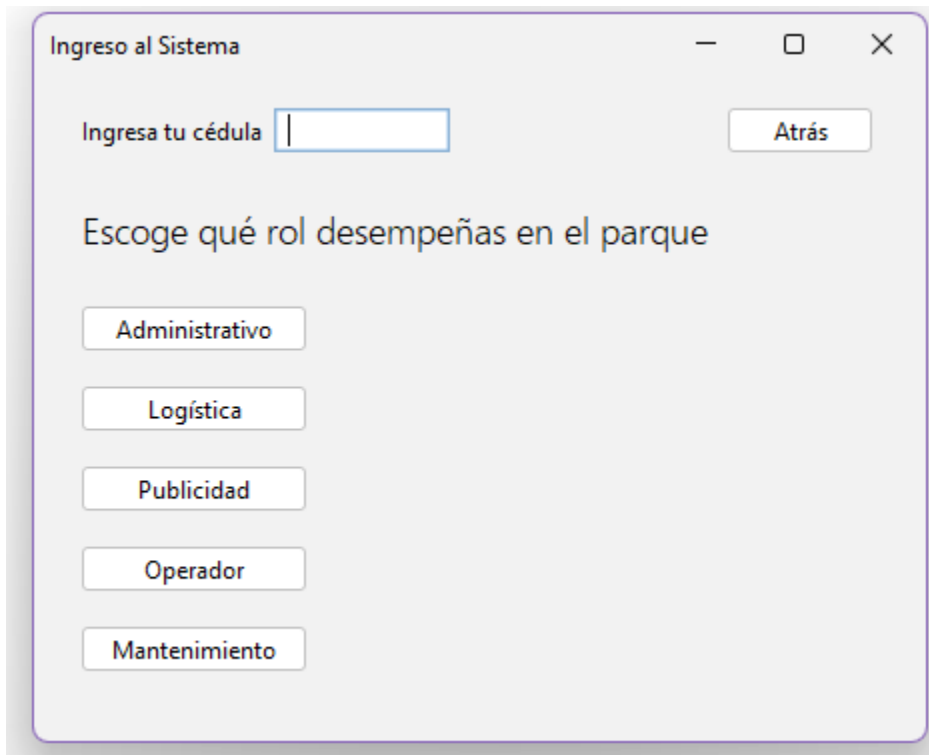
Escoge el tipo de tiquete

Estandar

¡Gracias por visitar el Parque Salitre Mágico Julian Molina! Se te asignó el tiquete número 16 úsalo para ingresar a las atracciones.

Ingresar al Parque

3. El parque tiene 5 tipos de empleados (administrativos, logística, publicidad, operadores y mantenimiento). Se creó una clase **Empleado** que se hereda de la clase **Persona**, esta tiene los atributos horario y tipo. El sistema verifica que al crear un nuevo empleado, este pertenezca a alguna de las áreas mencionadas. También, al ingresar al área de empleados, este debe proporcionar su cédula y el sistema hace una consulta a la base de datos para verificar que el empleado pertenece al área a la que desea ingresar. Los empleados **Administrativos** tienen acceso a la base de datos de **Clientes**, **Empleados**, **Estaciones** y **Atracciones**. Los empleados de **Publicidad** tienen acceso a la base de datos de **Visitas**. Los empleados **Operadores** tienen acceso a la base de datos de **Tiquetes**. Los empleados de **Mantenimiento** tienen acceso a la base de datos de **Atracciones**, pero solo pueden modificar el **estado** (Habilitada, Daño, Mantenimiento). Para los empleados de **Logística** no se asignó un área particular, pues sólo se encargan de entregar los tiquetes, sin embargo, el sistema también valida su cédula.



Ingreso al Sistema

Ingresa tu cédula

Atrás

Escoge qué rol desempeñas en el parque

Administrativo

Logística

Publicidad

Operador

Mantenimiento

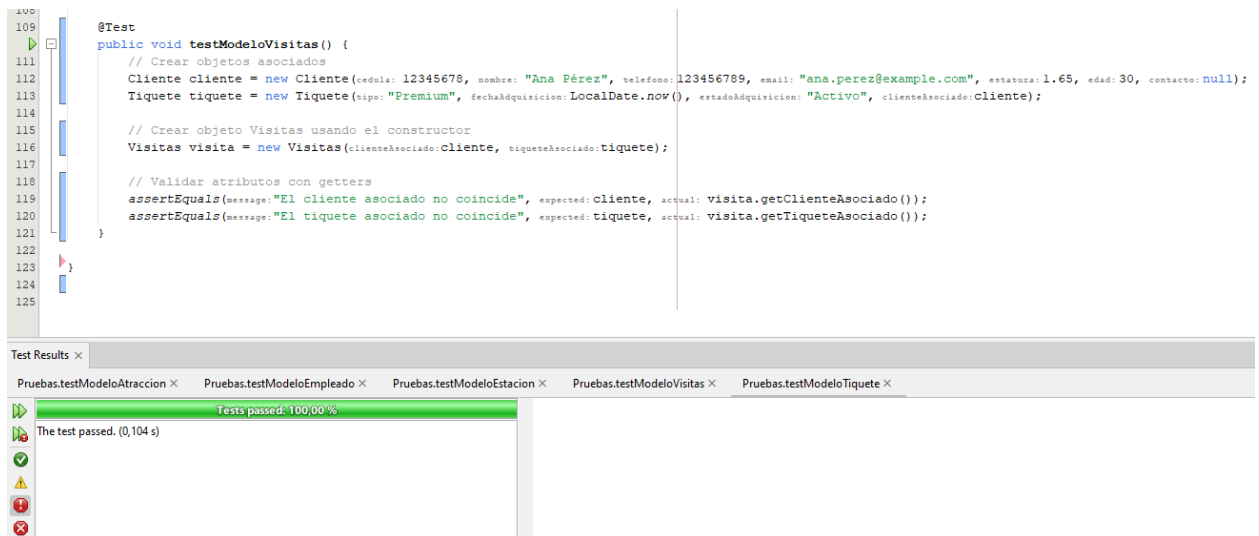
(La vista para cada área será abordada en el video explicativo)

4. Para ingresar a las **Atracciones** el cliente deberá proporcionar el número de ticket, el sistema hace una consulta para obtener la estatura del cliente asociado a este ticket y mostrará un mensaje si el cliente no cumple con la altura mínima. Sólo se mostrarán las atracciones con el estado **Habilitada**, el cual puede modificar un empleado de **Mantenimiento**. Si el cliente no cumple con la altura mínima, se mostrará un mensaje de advertencia y se notificará a un empleado **Operador**. Los empleados **Administrativos** recibirán notificaciones de las atracciones visitadas, con el objetivo de conocer su popularidad. Si el cliente hace click en el botón “**Salir del Parque**” se actualizará el estado del ticket a **Usado**.

The screenshot displays a software window titled "Ingreso a Atracciones". At the top, there is a text input field labeled "Ingresa el número de tu ticket" containing the value "16", and a button labeled "Atrás". Below this, the text "Escoge" is visible. A modal dialog box titled "Advertencia" is centered on the screen, featuring a yellow warning icon and the message "No cumples con la altura mínima requerida para esta atracción." with an "OK" button. In the background, a section titled "Montaña" is partially visible, showing a description "Una montaña rusa de alta velocidad.", a warning "No apto para personas con problemas de corazón", the word "Adrenalina", and a minimum height requirement "Altura mínima" of "1.6". At the bottom of the main window, there are buttons labeled "Ingresar" and "Salir del parque".

5. Adicional a todo lo anteriormente explicado, se usó persistencia de datos usando archivos **.txt** para notificar **alertas** y la **popularidad** de las atracciones, ya que no fue considerada necesaria la creación de una tabla en la base de datos para estas.

6. Finalmente, se crearon 5 pruebas unitarias usando JUnit, estas prueban el correcto funcionamiento de los modelos de las clases de la aplicación.



```
109
110
111 @Test
112 public void testModeloVisitas() {
113     // Crear objetos asociados
114     Cliente cliente = new Cliente(cedula: 12345678, nombre: "Ana Pérez", telefono: 123456789, email: "ana.perez@example.com", estatura: 1.65, edad: 30, contacto: null);
115     Tiquete tiquete = new Tiquete(tipo: "Premium", fechaAdquisicion: LocalDate.now(), estadoAdquisicion: "Activo", clienteAsociado: cliente);
116
117     // Crear objeto Visitas usando el constructor
118     Visitas visita = new Visitas(clienteAsociado: cliente, tiqueteAsociado: tiquete);
119
120     // Validar atributos con getters
121     assertEquals(message: "El cliente asociado no coincide", expected: cliente, actual: visita.getClienteAsociado());
122     assertEquals(message: "El tiquete asociado no coincide", expected: tiquete, actual: visita.getTiqueteAsociado());
123 }
124
125
```

Test Results x

Pruebas.testModeloAtraccion x Pruebas.testModeloEmpleado x Pruebas.testModeloEstacion x Pruebas.testModeloVisitas x Pruebas.testModeloTiquete x

Tests passed: 100,00 %

The test passed. (0,104 s)

**Link Video Explicativo:** <https://www.youtube.com/watch?v=5wS5WNXUjzI>

**Link del repositorio:** <https://github.com/Heixuzu/ParqueSalitreMagico>

Para concluir, la aplicación desarrollada para "Parque Salitre Mágico" es una herramienta integral que optimiza la gestión de las atracciones y mejora la experiencia de los visitantes. Este proyecto ha sido una valiosa oportunidad para aplicar mis conocimientos en desarrollo de software, integrando principios de Programación Orientada a Objetos, gestión de bases de datos y control de versiones con GIT en la creación de una solución MVC. A través de esta prueba, he logrado demostrar mis habilidades y obtener una visión más profunda de cómo las soluciones tecnológicas pueden mejorar la gestión de sistemas complejos.



