Sistema de Gestión de Parque de Atracciones

Nicolas Cano Cifuentes 30/05/2025

Diseño y programación O.O.

Entrega 3

# Tabla de Contenido

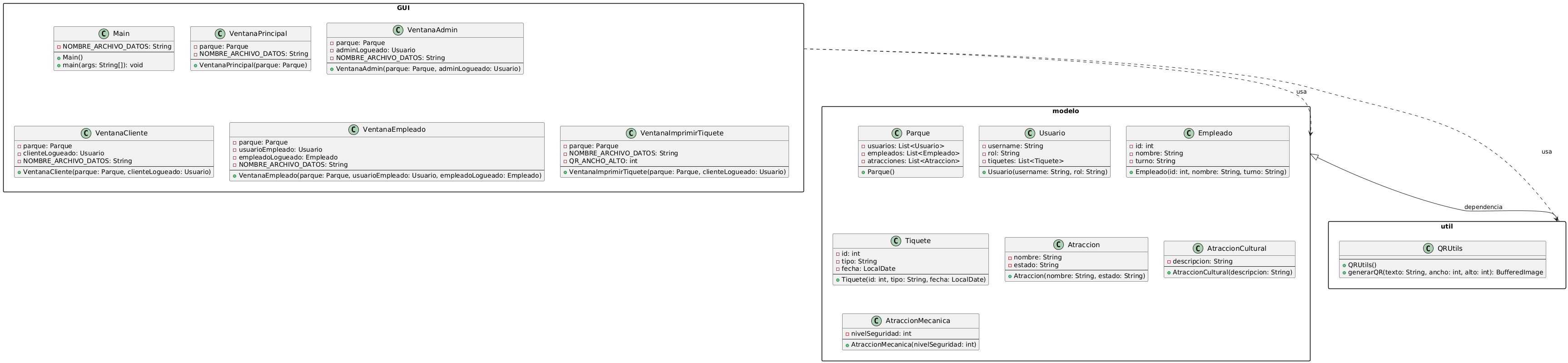
1. Descripción General del Sistema
2. Diagrama de Paquetes
3. Glosario de Clases y Responsabilidades
4. Diagrama de Clases de Alto Nivel
5. Diagrama de Clases Completo
6. Diagramas de Secuencia
7. Supuestos de Diseño
8. Reglas de Negocio
9. Consideraciones de Extensibilidad y Mantenibilidad
10. Anexos

# 1. Descripción General del Sistema

Este documento describe el diseño del sistema de información para la gestión de un parque de atracciones. El objetivo principal es facilitar la administración de los diferentes procesos asociados al parque, tales como el registro de usuarios, gestión de empleados, administración de atracciones y venta de tiquetes, a través de una interfaz gráfica amigable y segura.  
  
El sistema está estructurado en capas, separando de manera lo más clara posible la lógica de dominio (modelo de datos y reglas de negocio) de la interfaz de usuario. Para este diseño se promueve la mantenibilidad, facilita futuras ampliaciones y asegura que los cambios en una capa tengan mínimo impacto en las demás. El uso de una arquitectura modular y orientada a objetos contribuye a que el sistema sea robusto y escalable.

# 2. Diagrama de Paquetes

El sistema está dividido en tres paquetes principales:  
- GUI: Contiene las clases de la interfaz gráfica del usuario (ventanas principales, etc.).  
- modelo: Incluye la lógica de negocio y entidades clave (usuarios, empleados, atracciones, tiquetes).  
- util: Incluye clases de utilidad, como la generación de códigos QR.  
  
Cada paquete contiene exclusivamente elementos de su área de responsabilidad, promoviendo la independencia entre componentes.



# 3. Glosario de Clases y Responsabilidades

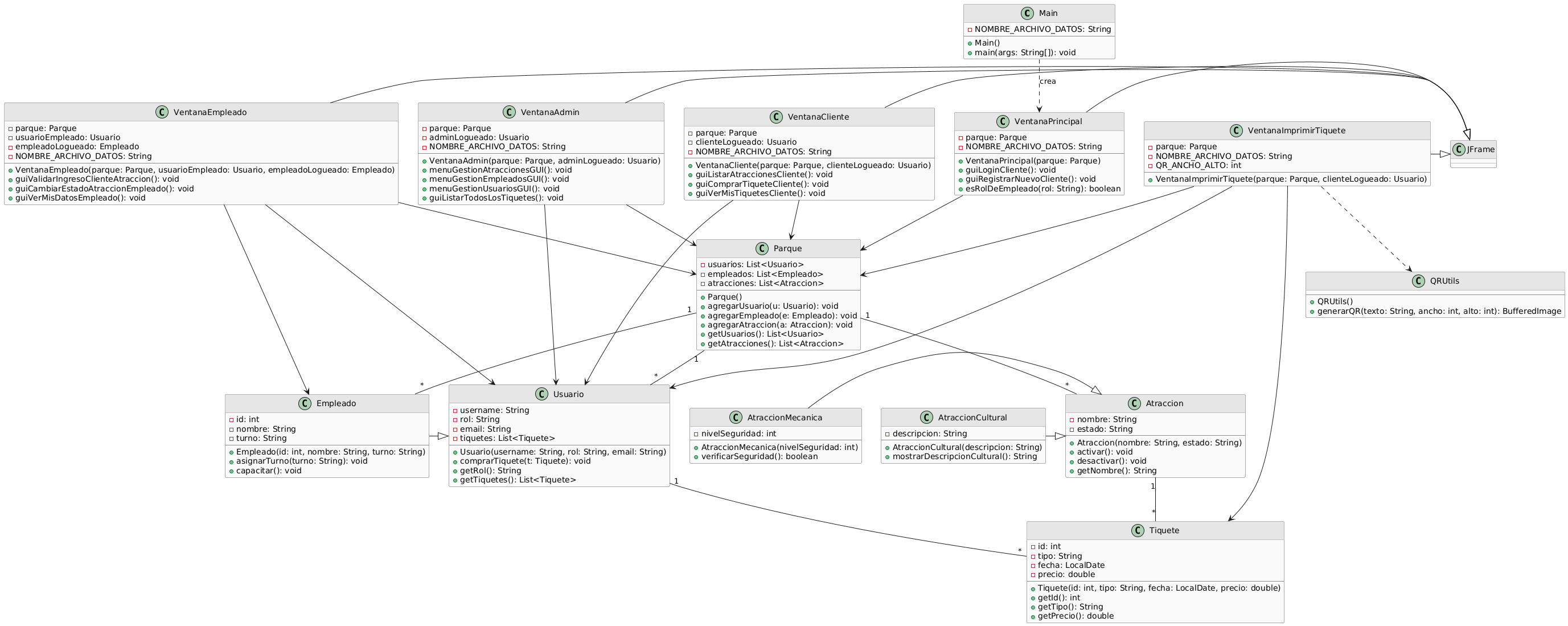
|  |  |  |
| --- | --- | --- |
| Clase | Paquete | Responsabilidad principal |
| Main | GUI | Punto de entrada, inicialización de la aplicación y la ventana principal |
| VentanaPrincipal | GUI | Ventana principal, acceso a módulos de cliente, empleado y administrador |
| VentanaAdmin | GUI | Ventana de administración, gestión de empleados, atracciones y usuarios |
| VentanaCliente | GUI | Ventana para clientes, compra y visualización de tiquetes |
| VentanaEmpleado | GUI | Ventana para empleados, validación de tiquetes y gestión de atracciones |
| VentanaImprimirTiquete | GUI | Ventana para visualizar e imprimir tiquetes |
| QRUtils | util | Clase utilitaria para generar códigos QR |
| Parque | modelo | Gestión central de usuarios, empleados y atracciones |
| Usuario | modelo | Representa a un usuario, maneja tiquetes y roles |
| Empleado | modelo | Subtipo de Usuario, maneja turnos y capacitación |
| Tiquete | modelo | Representa un tiquete comprado por un usuario |
| Atraccion | modelo | Representa una atracción, con sus estados |
| AtraccionCultural | modelo | Subtipo de Atracción, añade información cultural |
| AtraccionMecanica | modelo | Subtipo de Atracción, añade nivel de seguridad |

# 4. Diagrama de Clases de Alto Nivel

Este diagrama muestra cómo las principales clases de la interfaz gráfica se relacionan con los elementos centrales del dominio del sistema. Se ilustra la conexión directa entre los componentes visuales y la lógica de negocio, sin entrar en el detalle completo de atributos y métodos internos.

# 5. Diagrama de Clases Completo

En este apartado se presenta el diagrama UML que abarca todas las clases relevantes del sistema, con sus principales atributos, métodos y relaciones. Este diagrama refleja la arquitectura final implementada, incluyendo herencia, asociaciones y dependencias entre los diferentes componentes.

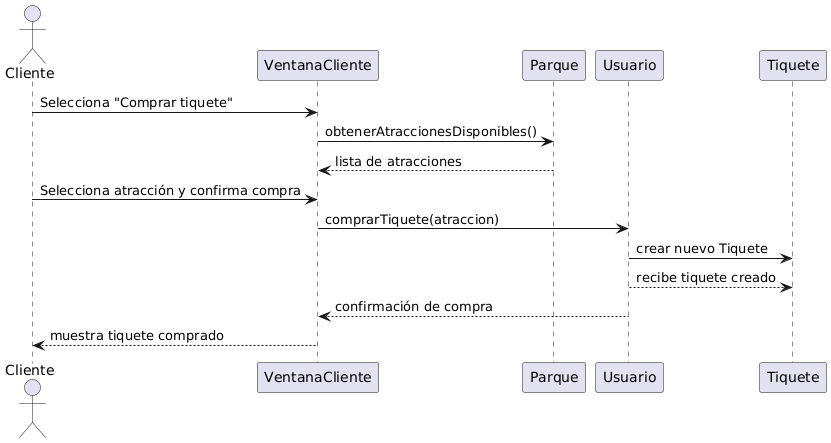


# 6. Diagramas de Secuencia

A continuación, se presentan los diagramas de secuencia para los procesos más importantes del sistema. Cada diagrama ilustra la interacción entre los actores y los objetos clave del sistema.

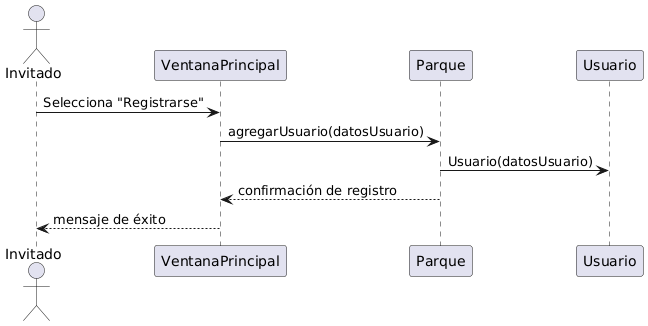
## 6.1. Compra de Tiquete por un Cliente

Cuando un cliente desea comprar un tiquete, la interacción comienza en la VentanaCliente, donde selecciona la opción de compra. El sistema obtiene las atracciones disponibles desde el objeto Parque. El cliente elige una atracción y confirma la compra. Entonces, el Usuario crea un nuevo Tiquete y el sistema notifica al cliente mostrando la confirmación de la compra.



## 6.2. Registro de Usuario Nuevo

Este flujo cubre el registro de un nuevo usuario. Un invitado accede a la VentanaPrincipal y solicita el registro. La ventana recopila los datos y solicita al Parque agregar el nuevo usuario. El Parque crea un nuevo objeto Usuario y confirma el registro, permitiendo al usuario autenticarse posteriormente.



## 6.3. Validación de Acceso a Atracción por Empleado

Cuando un empleado valida el acceso de un cliente a una atracción, accede a la VentanaEmpleado y selecciona la opción correspondiente. Se verifica en el Usuario si el cliente posee un tiquete válido para la atracción. El sistema también comprueba que la atracción esté activa. El resultado se muestra al empleado.

## 

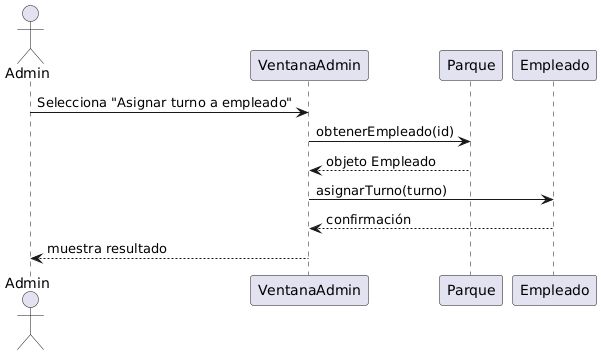
## 6.4. Cambio de Estado de Atracción por Administrador

El administrador puede activar o desactivar una atracción desde la VentanaAdmin. El proceso implica seleccionar la atracción, cambiar su estado mediante una llamada a la lógica del dominio y finalmente mostrar el resultado de la operación.

## 

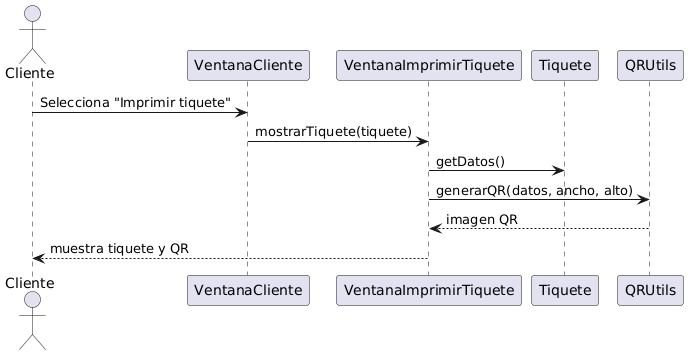
## 6.5. Asignar Turno a Empleado

Este proceso es gestionado por un administrador, quien desde la VentanaAdmin asigna un turno a un empleado. Se obtiene el empleado desde el Parque, se asigna el turno y se confirma la operación.



## 6.6. Impresión o Visualización de Tiquete

El cliente puede imprimir o visualizar un tiquete. Desde la VentanaCliente accede a la VentanaImprimirTiquete, donde se consulta la información del tiquete, se genera el código QR usando QRUtils y finalmente se muestra la información al cliente.



# 7. Supuestos de Diseño

En esta sección mostraremos algunos supuestos de diseño que nos ayudaran a tener un mejor entendimiento de cómo se estructuro el código y se pensó para algunos casos particulares para su funcionamiento.

- El sistema asume que todos los usuarios tienen un nombre de usuario único.  
- Los datos persistentes se cargan desde un archivo local al iniciar la aplicación.  
- Los empleados no pueden cambiar su propio rol ni gestionar sus propios turnos.  
- Solo los administradores pueden modificar la información de atracciones y empleados.  
- El sistema confía en las validaciones realizadas en la interfaz gráfica antes de pasar datos al modelo.

-El administrador es el único que puede editar tanto empleados y sus cargos como también atracciones.

-El sistema espera que cuando se comienza a usar el programa se cree un administrador.

# 8. Reglas de Negocio

- Un usuario puede comprar varios tiquetes, pero solo para atracciones activas.  
- Los tiquetes tienen una única fecha de uso y no son transferibles.  
- Las atracciones mecánicas requieren que el nivel de seguridad esté por encima de un umbral mínimo para estar activas.  
- Un empleado puede operar solo las atracciones asignadas a su perfil y dentro de su turno.  
- Un usuario con rol administrador puede crear y eliminar empleados, atracciones y usuarios.

# 9. Consideraciones de Extensibilidad y Mantenibilidad

El sistema ha sido diseñado de forma modular y orientada a objetos, siguiendo principios de bajo acoplamiento y alta cohesión. La separación clara entre los paquetes GUI y modelo permite desarrollar nuevas interfaces (por ejemplo, una versión web o móvil) reutilizando la lógica de negocio ya implementada. Gracias al uso de herencia en clases como Atraccion, es posible agregar nuevas categorías de atracciones sin modificar el código existente, solo extendiendo la jerarquía.  
  
Además, la centralización de utilidades como la generación de códigos QR en un paquete específico (util) facilita la reutilización y mantenimiento de estas funciones.

# 10. Anexos

La aplicación se puede testear ya que funciona normal, sin embargo, agregare algunas capturas de su funcionamiento.

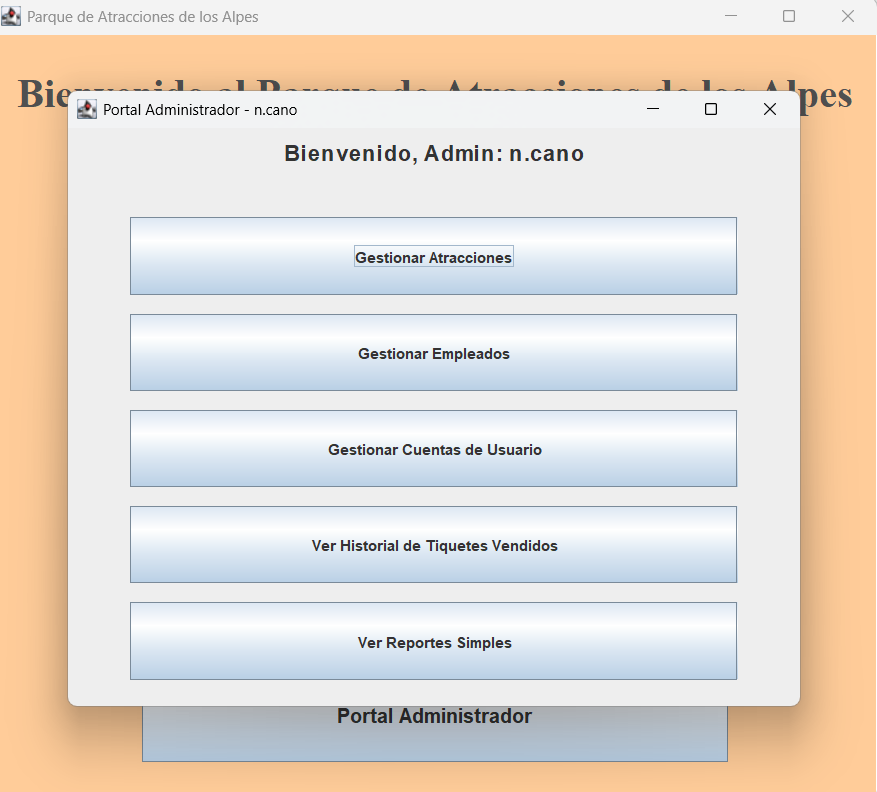
Funcionamiento 1



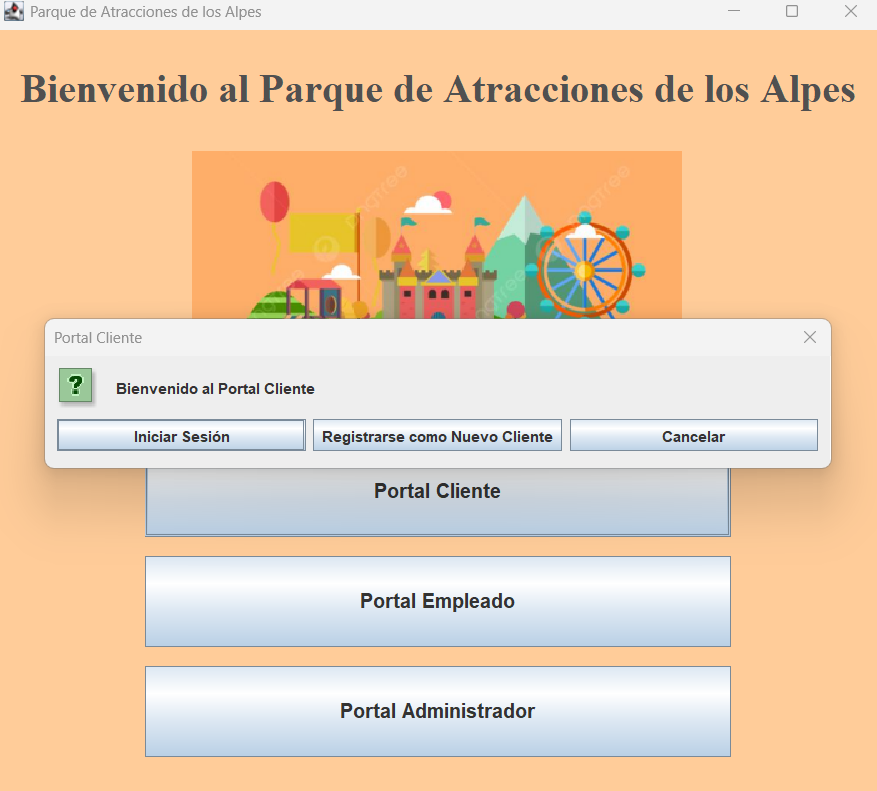
Funcionamiento 2



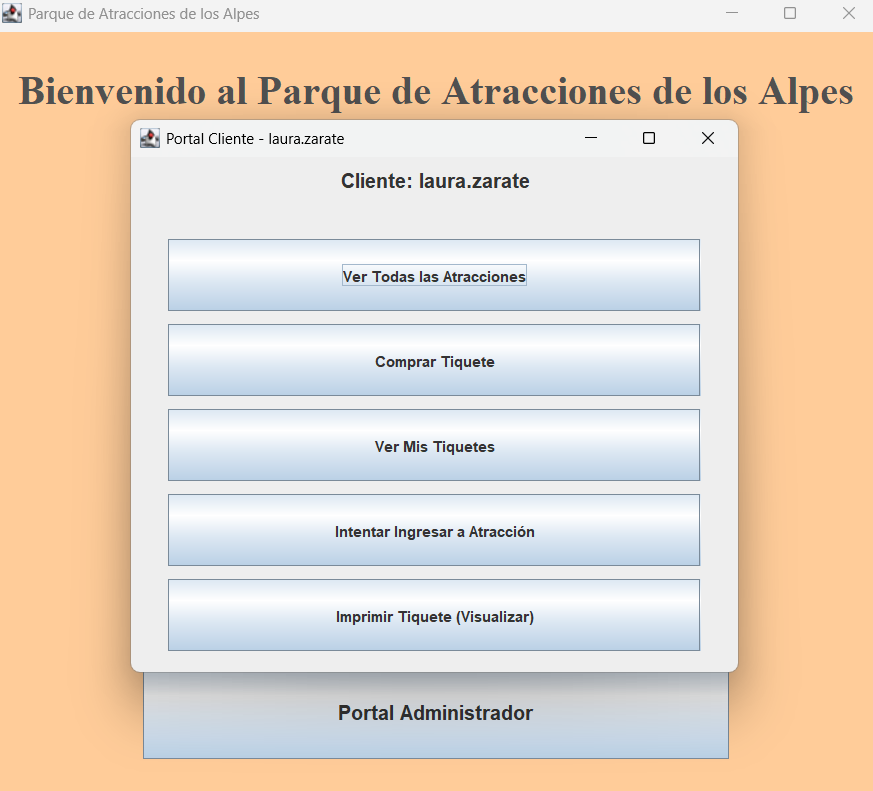
Funcionamiento 3.



Funcionamiento 4:



Funcionamiento 5



Funcionamiento 6



Otro tema para considerar es que para probar el programa se debe ejecutar desde el main, pedirá que se cree un administrador, de ahí se puede usar el menú para crear una cuenta cliente, luego de crear una cuenta de cliente se tiene que hacer la operación de comprar un tiquete de cualquier manera, esto con el objetivo de tener un ticket y poder imprimirlo con su respectivo QR en la opción de imprimir ticket en el menú de cliente.

Como Anexo hay que resaltar que se usó librerías externas para la creación de códigos QR, siendo estas las librerías de QR Gen y Zxing.

<https://github.com/kenglxn/QRGen>

<https://github.com/zxing/zxing>