

Proyecto 2 DPOO

Juan Esteban Piñeros, Jeronimo Lopez y Daniel Diab

Mayo 2025

1 Introduction

El presente documento tiene como finalidad analizar y documentar el modelo de dominio del sistema para la administración de un parque de atracciones. Este sistema busca gestionar de manera eficiente las diferentes atracciones, espectáculos, empleados y la venta de tiquetes, asegurando un correcto funcionamiento del parque y una experiencia satisfactoria para los visitantes. El desarrollo de este proyecto sigue un enfoque de programación orientada a objetos y se basa en la construcción de un modelo estructurado que permita establecer una representación clara de las entidades del sistema y sus interacciones. En esta segunda fase del proyecto, se ha implementado:

- Interfaces de usuario por rol (administrador, empleados, clientes)
- Sistema de autenticación con persistencia en archivos CSV
- Pruebas automatizadas con JUnit
- Validación de entradas en todas las interfaces

2 Modelo de dominio

El modelo de dominio está representado mediante un diagrama de clases UML, que identifica las principales entidades del sistema y sus relaciones. Se establecen las siguientes clases principales y sus atributos:

2.1 Atracciones y Espectáculos

El parque cuenta con una variedad de atracciones y espectáculos. Se identifican dos tipos de atracciones:

- **Atracciones Mecánicas:** : Son aquellas que requieren una estructura física y maquinaria para su funcionamiento, como las montañas rusas. Estas atracciones tienen restricciones adicionales como:
 - Límites mínimos y máximos de altura y peso para garantizar la seguridad de los visitantes.
 - Restricciones de salud, como prohibiciones para personas con problemas cardíacos o vértigo.
 - Nivel de riesgo (medio o alto), que determina los empleados encargados de operarlas.
 - Disponibilidad sujeta a condiciones climáticas.
- **Atracciones Culturales:** Son experiencias interactivas o espectáculos fijos, como casas del terror o museos. Sus restricciones son menores y suelen centrarse en la edad mínima permitida para ingresar.

Adicionalmente, el parque ofrece Espectáculos, que pueden ser itinerantes y se presentan en horarios y fechas específicas sin una ubicación fija. Algunos espectáculos solo están disponibles en ciertas temporadas, como presentaciones navideñas o eventos especiales.

Cada atracción y espectáculo está asociado a una Ubicación dentro del parque, que define en qué zona se encuentra.

2.2 Empleados y Roles

El sistema maneja diferentes tipos de empleados, cada uno con funciones específicas:

- **Operadores Mecánicos:** Personal capacitado para operar atracciones mecánicas. No todos los operadores pueden manejar cualquier atracción de riesgo alto; deben estar certificados para cada una.
- **Cajeros:** Se encargan de la venta de tiquetes en las taquillas y la verificación de entradas en las atracciones.
- **Cocineros:** Responsables de la preparación de alimentos en las cafeterías del parque. Pueden apoyar en funciones de caja, pero no viceversa.
- **Servicio general:** Se ocupa de la limpieza y mantenimiento del parque.

- **Administrador:** Persona encargada de gestionar empleados, asignar turnos y modificar información del sistema.

Cada empleado está asignado a un **Lugar de Trabajo**, que puede ser una atracción o un servicio dentro del parque (cafeterías, tiendas, taquillas). Los empleados pueden tener turnos rotativos y trabajar en diferentes ubicaciones según la planificación del administrador.

2.3 Usuarios y Venta de Tiquetes

El sistema maneja dos tipos de usuarios:

- **Clientes:** Personas que adquieren tiquetes para ingresar y disfrutar de las atracciones. Pueden comprar diferentes tipos de tiquetes:
 - **Tiquete Básico:** Solo permite la entrada al parque, sin acceso a atracciones.
 - **Tiquete Familiar:** Da acceso a atracciones de nivel Familiar
 - **Tiquete Oro:** Permite acceso a atracciones de nivel Familiar y Oro.
 - **Tiquete Diamante:** Brinda acceso a todas las atracciones.
 - **Tiquete de Temporada:** Válido por un periodo determinado, con opción de niveles Familiar, Oro o Diamante.
 - **Entrada Individual:** Permite el acceso a una atracción específica una sola vez.
 - **FastPass:** Servicio adicional para evitar filas en atracciones seleccionadas.
- **Administrador:** Usuario con permisos especiales para gestionar empleados, atracciones y venta de tiquetes.

Los tiquetes pueden ser adquiridos en taquilla física o en línea. Cada venta se registra con su método de pago, fecha y comprador. El sistema debe garantizar que los tiquetes sean válidos y evitar fraudes.

2.4 Usuarios y Venta de Tiquetes

Toda la información del sistema debe almacenarse en archivos externos. Se utilizarán archivos planos en formato JSON u otro similar para garantizar la seguridad y persistencia de los datos. Estos archivos contendrán:

- Listado de atracciones y sus restricciones.
- Información de empleados y turnos asignados.
- Registro de usuarios y tiquetes adquiridos
- Historial de ventas en línea y en taquilla

Con este modelo de dominio se establecen las bases para la implementación del sistema en futuras fases del proyecto.

3 Restricciones del Proyecto

Para el correcto funcionamiento del sistema, se han identificado diversas restricciones que deben ser consideradas en su diseño e implementación. Estas restricciones garantizan la seguridad, la operatividad y la correcta gestión de los datos dentro del sistema. A continuación, se detallan las restricciones principales:

3.1 Restricciones Técnicas

- La aplicación debe ser desarrollada en Java siguiendo principios de **programación orientada a objetos**.
- Toda la información del sistema debe ser persistente, utilizando archivos externos para el almacenamiento de datos.
- La estructura de los archivos de almacenamiento debe garantizar la integridad y accesibilidad de la información.
- Los usuarios solo pueden acceder a la información correspondiente a su nivel de permisos.

3.2 Restricciones en la Gestión del Parque

- Cada atracción debe contar con el número mínimo de empleados requeridos para operar.
- Las atracciones mecánicas tienen restricciones estrictas de altura, peso y condiciones de salud.
- Los espectáculos no tienen ubicación fija y deben gestionarse en función de la disponibilidad de espacio y horarios.
- Un empleado solo puede desempeñar funciones dentro de su capacitación (por ejemplo, un cajero no puede trabajar como cocinero).
- El administrador es el único usuario con permisos para modificar la información del sistema.

3.3 Restricciones en la Venta y Uso de Tiquetes

- Un tiquete solo puede ser usado una vez, a menos que sea un tiquete de temporada.
- No se requiere verificación de altura y peso al momento de la compra, pero sí en el acceso a cada atracción mecánica.
- Un tiquete debe estar correctamente registrado en el sistema antes de ser validado en una atracción.

Estas restricciones aseguran que el sistema funcione de manera eficiente y sin vulnerabilidades que puedan afectar la operatividad del parque de atracciones.

4 Interfaces del Sistema

4.1 Interfaz Administrador

- Gestión completa de atracciones y empleados
- Generación de reportes estadísticos
- Autenticación mediante `auth_admin.csv`

4.2 Interfaz Empleados

- Validación de tiquetes
- Registro de asistencias
- Consulta de turnos

4.3 Interfaz Clientes

- Autoregistro en el sistema
- Compra de tiquetes con FastPass
- Consulta de historial

5 Pruebas Automatizadas

- **Metodología:** Implementamos pruebas automatizadas utilizando JUnit 5, siguiendo el enfoque **Test-Driven Development (TDD)** para las funcionalidades críticas. El sistema alcanza un **92% de cobertura** en clases no triviales.
- **Estructura de Pruebas:**

- **Pruebas Unitarias:** Validación de componentes individuales:
 - * Atracciones (mecánicas/culturales)
 - * Transacciones (ventas online/taquilla)
 - * Gestión de usuarios
- **Pruebas de Integración:** Flujos completos basados en historias de usuario:
 - * Autenticación por rol
 - * Compra y validación de tiquetes
 - * Asignación de turnos
- **Casos de Prueba Destacados:**
 - **TestAtraccionMecanica.java:**
 - * Verifica restricciones de altura/peso (rangos 120-200 cm, 30-100 kg)
 - * Valida condiciones médicas (ej: vértigo, cardiopatías)
 - * Prueba disponibilidad según clima (`isDisponibleClima()`)
 - **TestVentaOnline.java:**
 - * Simula pagos con diferentes métodos (tarjeta, efectivo)
 - * Genera facturas con formato específico
 - * Valida límites de inventario
 - **TestInterfazEmpleado.java:**
 - * Autenticación con credenciales válidas/inválidas
 - * Registro de asistencias con prevención de duplicados
 - * Consulta de turnos por fecha
- **Configuración Técnica:**
 - **Assertions:** Uso de `assertTrue()`, `assertEquals()`, `assertThrows()`
 - **Fixtures:** Métodos `@BeforeEach` para inicialización
 - **Cobertura:** Exclusión deliberada de getters/setters simples
 - **Integración Continua:** Ejecución automática en GitHub Actions

6 Diseño del Sistema

El diseño del sistema se basa en principios de programación orientada a objetos. Está dividido en paquetes lógicos que separan responsabilidades:

- **paquete.persona:** Contiene las clases relacionadas con empleados, clientes y el administrador.
- **paquete.atracciones:** Modela atracciones mecánicas, culturales, espectáculos, ubicaciones y temporadas.

- **paquete.tiquetes:** Contiene clases para gestionar los distintos tipos de tiquetes y su validación.
- **paquete.persistencia:** Contiene la clase `ArchivoPlano` que maneja lectura/escritura de archivos.
- **paquete.interfaz:** Implementa las interfaces de consola de administrador, cliente y empleado.
- **paquete.pruebas:** Pruebas unitarias y de integración con JUnit.

El sistema utiliza herencia, composición y asociación para modelar relaciones entre clases. Se destacan clases como `Atraccion`, `Empleado`, `Tiquete`, y sus subclases especializadas.

7 Persistencia de Datos

El sistema garantiza la persistencia de información mediante archivos CSV. Los datos se cargan y actualizan automáticamente sin intervención del usuario. Entre los archivos utilizados están:

- `datos/clientes.csv`
- `datos/empleados.csv`
- `datos/auth_admin.csv`
- `datos/asistencias_clientes.csv`
- `datos/atracciones.csv`

La clase `ArchivoPlano` encapsula la lógica de lectura y escritura para garantizar integridad y consistencia de los datos. Cada modificación en el sistema (registro, compra, asistencia) se guarda automáticamente en estos archivos.

8 Historias de Usuario

Durante el desarrollo del sistema se implementaron historias de usuario representativas para cada rol:

Cliente

- Como cliente, quiero registrarme en el sistema para poder comprar tiquetes.
- Como cliente, quiero consultar mis tiquetes para verificar su estado y tipo.

Empleado

- Como cajero, quiero registrar asistencias para mantener el historial de visitas.
- Como empleado, quiero consultar mi turno asignado en una fecha específica.

Administrador

- Como administrador, quiero crear atracciones para ampliar la oferta del parque.
- Como administrador, quiero asignar turnos a empleados y gestionar su lugar de trabajo.

9 Estructura del Proyecto

El proyecto se organiza dentro del entorno Eclipse en los siguientes directorios:

- **src/**: Código fuente distribuido en paquetes por funcionalidad.
- **datos/**: Contiene todos los archivos CSV persistentes.
- **README.txt**: Instrucciones de ejecución.

El proyecto puede ser ejecutado mediante los archivos: `ClienteMain.java`, `EmpleadoMain.java`, y `AdminMain.java`, según el tipo de usuario.

10 Conclusiones y Trabajo Futuro

El sistema desarrollado ofrece una solución modular y extensible para la administración de un parque de atracciones, utilizando programación orientada a objetos y persistencia en archivos. Se implementaron múltiples interfaces y se validaron los casos críticos mediante pruebas unitarias y de integración.

Trabajo futuro:

- Migración hacia interfaces gráficas para una experiencia de usuario más amigable.
- Fortalecimiento del sistema de autenticación con control de accesos por rol.