Autores:

Santiago Campo González – 202325582

Juan José Cocomá Lozano – 202322061

Juan Pablo Ussa Caicedo – 202314574

1. Estructura General del Sistema

El sistema está organizado de manera modular en los siguientes paquetes:

* model: Contiene las clases que representan las entidades del dominio, como Cliente, Empleado, Tiquete, FastPass, Atracción, Espectáculo, entre otras.
* service: Encapsula la lógica de negocio a través de clases gestoras como GestorClientes, GestorTiquetes, GestorEmpleados, GestorTurnos y GestorAtracciones.
* persistence: Maneja la lectura y escritura de datos desde y hacia archivos .txt, respetando el principio de separación de responsabilidades.
* app: Incluye las interfaces de usuario por consola para los diferentes roles del sistema: ClienteApp, EmpleadoApp y AdministradorApp.
* test: Contiene pruebas unitarias escritas con JUnit para validar el comportamiento esperado de los distintos componentes.

2. Requerimientos Funcionales

Administración de Atracciones y Espectáculos:

* Registrar, editar y eliminar atracciones mecánicas y culturales.
* Registrar, editar y eliminar eventos o espectáculos.
* Modificar la disponibilidad de las atracciones según la temporada.
* Calcular y mostrar la disponibilidad considerando condiciones climáticas y personal mínimo.
* Validar el cumplimiento de restricciones físicas como altura, peso y salud.

Venta de Tiquetes:

* Registrar y vender tiquetes de tipo Básico, Familiar, Oro y Diamante.
* Registrar y vender tiquetes de temporada (semanal, mensual, estacional, anual).
* Registrar y vender tiquetes individuales para atracciones específicas.
* Registrar y vender FastPass asociados a fechas puntuales.
* Registrar compras con descuento realizadas por empleados.
* Validar el uso de tiquetes para evitar fraudes y usos múltiples no permitidos.

Gestión de Empleados:

* Añadir, modificar y eliminar empleados.
* Asignar empleados a atracciones y espectáculos mediante turnos.
* Validar credenciales y roles para el acceso a funciones específicas.
* Mostrar a cada empleado su calendario de turnos desde la interfaz correspondiente.
* Administrar username y contraseña de cada empleado.

3. Requerimientos No Funcionales

El sistema debe estar completamente desarrollado en el lenguaje de programación Java.

La persistencia de datos debe implementarse mediante archivos de texto (.txt) ubicados en una carpeta dedicada fuera de la estructura principal del proyecto.

El sistema debe incluir pruebas automatizadas que validen las funcionalidades principales.

Debe existir una clara separación entre la lógica de negocio y la presentación.

4. Separación de Lógica e Interfaz

Cada aplicación (ClienteApp, EmpleadoApp, AdministradorApp) sirve como interfaz de usuario por consola y se comunica exclusivamente con los gestores del paquete service. No se permite la implementación de lógica de negocio directamente en estas interfaces, asegurando una arquitectura desacoplada y mantenible.

5. Persistencia

El subsistema de persistencia se encarga de almacenar y recuperar los siguientes elementos a través de archivos de texto:

* Clientes (clientes.txt)
* Empleados (empleados.txt)
* Tiquetes y FastPass (tiquetes.txt, fastpass.txt)
* Atracciones (atracciones.txt)
* Turnos (turnos.txt)
* Registros de ventas (ventas.txt)

Todos los datos se cargan automáticamente al inicio de cada aplicación correspondiente, garantizando que el sistema opere con la información más reciente.

6. Validaciones del Sistema

La lógica del sistema incluye validaciones automáticas como:

* Restricciones físicas para el ingreso a atracciones (altura, peso, salud).
* Validación de uso único de tiquetes y FastPass.
* Autenticación por credenciales con verificación de roles para funcionalidades restringidas.
* Validación de disponibilidad de atracciones según condiciones meteorológicas, temporada y asignación de empleados.

7. Modelo de Clases (Resumen Descriptivo)

Se entrega un diagrama UML completo en formato PDF/PNG como complemento gráfico a este documento.

Entidades Principales:

* Cliente, Empleado, Turno
* Tiquete (abstracta) con subclases TiqueteIndividual y TiqueteTemporada
* ElementoParque (abstracta) con subclases AtraccionMecanica, AtraccionCultural, Espectaculo
* FastPass, asociado a un Tiquete

Gestores de Lógica de Negocio:

* GestorClientes
* GestorEmpleados
* GestorTiquetes
* GestorTurnos
* GestorAtracciones

Relaciones Clave:

* Empleado es autenticado mediante GestorEmpleados.
* Tiquete es gestionado por GestorTiquetes, que maneja su registro, validación y uso.
* FastPass se utiliza junto con el tiquete correspondiente y se valida su uso por fecha.

8. Pruebas

Las pruebas unitarias se encuentran en el paquete test e incluyen:

* TiqueteTest, FastPassTest
* GestorEmpleadosTest, GestorClientesTest

Estas pruebas validan casos críticos identificados en las historias de usuario y buscan garantizar la calidad y estabilidad del sistema a lo largo del tiempo.

Historia de los usuarios

Cliente

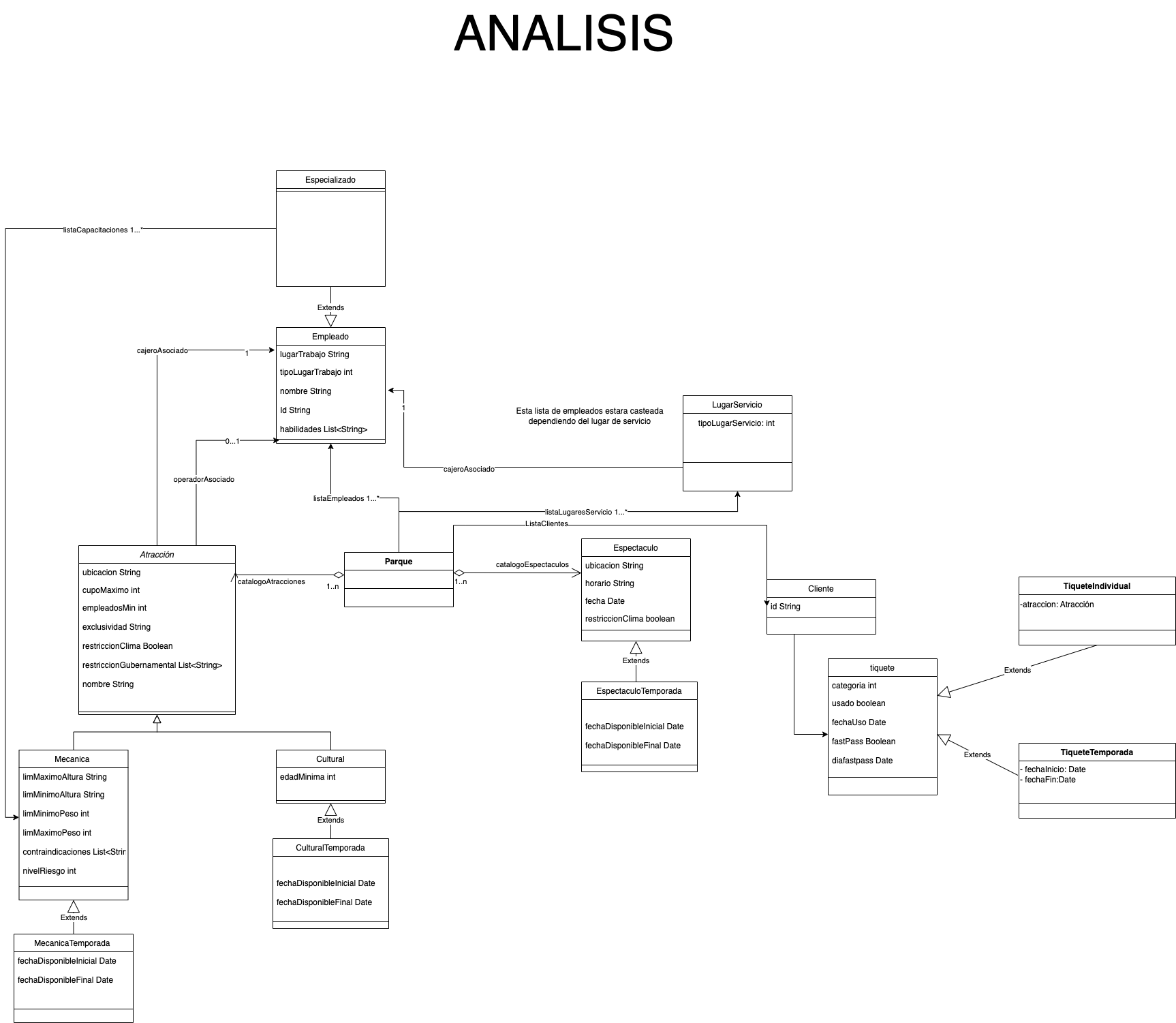
Como cliente quiero poder ingresar mi nombre al tiquete y obtener un codigo para usasrlo. de modo que pueda entrar al parque de forma correcta. Y si quiero fast pass podre ingresar la opcion sin problema y podre acceder a las filas rapidas.

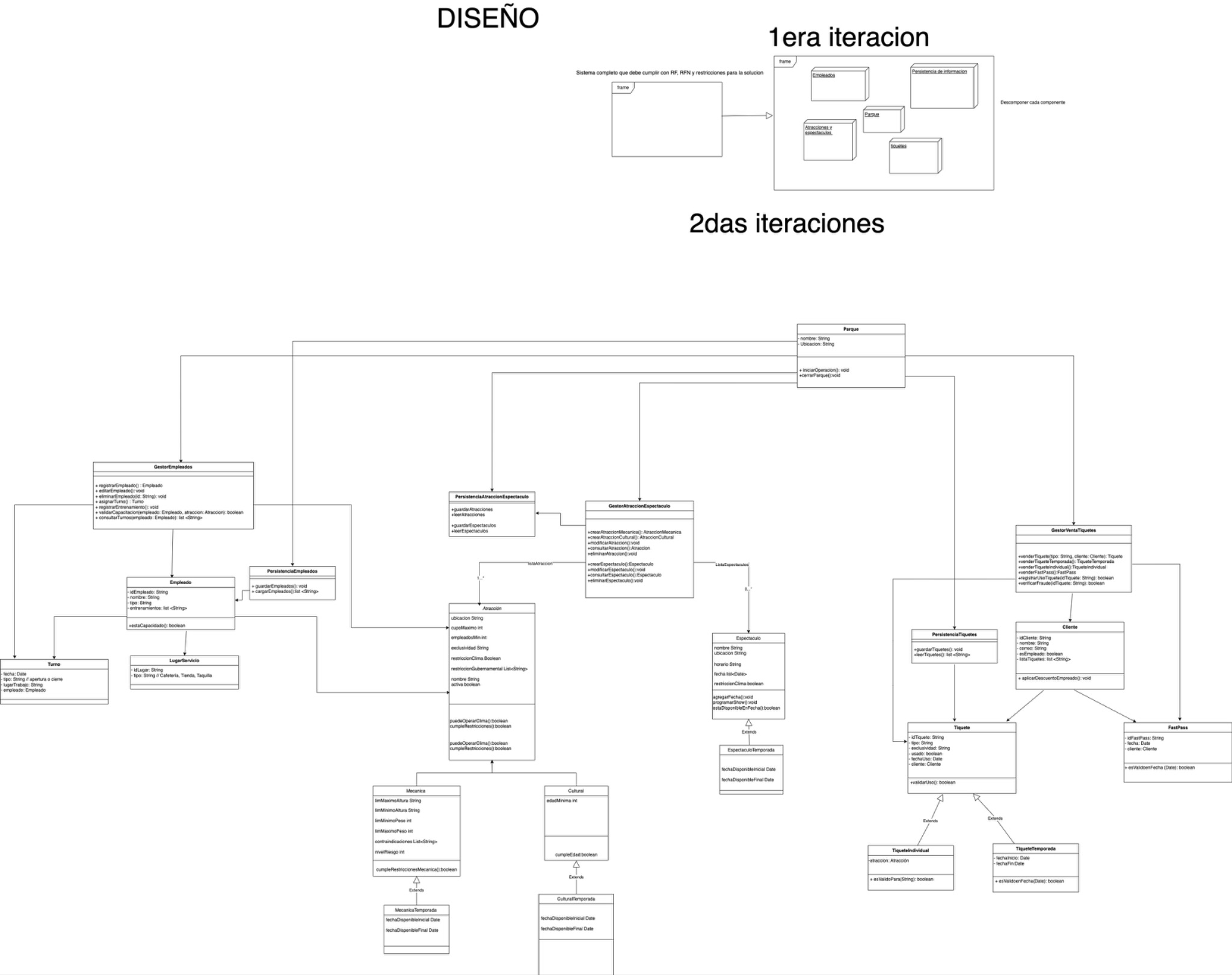
Empleado

Como empleado quiero poder ingresar mi id y contraseña y aparte poder ver mis turnos asignados para cada atraccion y dia

Admin

Podre ingresar mis credenciales para gestionar el parque. tambien podre actualizar e ingresar nuevos empleados o su informacion. Aparte podre agregar nuevas atracciones y espectaculos sin problema





<https://acortar.link/9rqtqN>