**Lucas Donado Cortes 202320724**

**Diego Bernal 202011546**

Proyecto 1

|  |
| --- |
| 1. **Contexto y Análisis del Problema** |
| El sistema a desarrollar tiene como propósito gestionar las operaciones de un parque de diversiones que cuenta con diversos tipos de atracciones (mecánicas y culturales) y espectáculos en vivo, además de un conjunto amplio de empleados que cubren distintos roles, y un sistema de venta de tiquetes para los visitantes. |
| * **Atracciones y Espectáculos**   **Atracciones Mecánicas:**   * Construcciones a gran escala (por ejemplo, montañas rusas). * Restringidas por límites de altura, peso y posibles contraindicaciones de salud (vértigo, problemas cardíacos, discapacidades, etc.). * Cada atracción mecánica requiere un número mínimo de empleados capacitados y puede tener diferentes niveles de riesgo (medio o alto).   **Atracciones Culturales:**   * Usualmente en espacios temáticos o salas de exhibición. * Con menores restricciones, enfocadas principalmente en la edad mínima de ingreso en ciertos casos (por ejemplo, castillos de terror).   **Espectáculos:**   * Pueden realizarse en cualquier lugar del parque y en horarios específicos. * Algunos pueden ser de temporada o de un solo día (como un show de fin de año). * Se consideran “tercerizados”, por lo que no requieren empleados del parque para su operación. * Todas las atracciones (mecánicas o culturales) tienen un cupo máximo de personas y un nivel de exclusividad asociado (Familiar, Oro o Diamante). Algunas atracciones y espectáculos pueden verse afectadas por el clima. * **Empleados y Gestión de Roles** Se deben manejar distintos tipos de empleados (cajeros, cocineros, servicio general, operadores de atracciones, etc.).  Los cajeros pueden trabajar en taquillas (vendiendo tiquetes y revisándolos en las atracciones) o en lugares de servicio (cafeterías, tiendas).   Los cocineros solo pueden trabajar en cafeterías, pero un cocinero podría cubrir momentáneamente un puesto en caja si así se requiere.  Los operadores de atracciones mecánicas deben contar con la capacitación adecuada para operar atracciones de riesgo medio o alto.   * Las de riesgo alto solo pueden ser operadas por empleados específicamente entrenados en esa atracción.   Cada día que el parque abre, hay dos turnos de trabajo (apertura y cierre); un mismo empleado puede tomar ambos turnos (horas extra).  Los empleados pueden consultar su asignación diaria. El administrador del parque es quien gestiona los turnos y la información de los empleados.   * **Venta de Tiquetes**   Existen diversas modalidades de tiquetes: Básico, Familiar, Oro, Diamante, Temporada, Individual y FastPass.  Los tiquetes Básico solo dan acceso al parque sin incluir atracciones.  Los tiquetes Familiar, Oro y Diamante permiten el acceso a atracciones de su mismo nivel y de niveles inferiores (por ejemplo, un tiquete Oro permite acceder a atracciones Familiares y Oro).  Los tiquetes Temporada permiten uso ilimitado dentro de un rango de fechas (semanal, mensual, estacional, anual) y ofrecen un descuento significativo respecto a comprar tiquetes sueltos.  Los tiquetes Individual dan acceso a una atracción específica, una sola vez, en cualquier momento en que esté disponible.  El FastPass puede comprarse para un día específico, independientemente del tiquete base que se tenga, permitiendo acceso prioritario.  Para evitar fraudes, el sistema debe registrar si un tiquete ya fue usado.   * **Persistencia y Restricciones Técnicas**   Toda la información (atracciones, empleados, tiquetes, etc.) debe almacenarse en archivos planos o binarios, en una carpeta separada del código fuente.  Solo la aplicación tiene acceso de lectura/escritura a dicha carpeta (no se contemplan ataques maliciosos directos).  Todos los usuarios del sistema (empleados, administrador, clientes) deben autenticarse con login y password. |

|  |
| --- |
| 1. **Requerimientos Funcionales** |
| * **Gestión de Atracciones y Espectáculos**  Registrar, modificar y eliminar atracciones (mecánicas y culturales) con sus respectivos límites, cupo máximo y nivel de exclusividad.  Registrar, modificar y eliminar espectáculos, indicando fecha, horario y ubicación temporal.  Consultar información de atracciones y espectáculos, incluyendo restricciones de acceso (edad, altura, peso, contraindicaciones, clima).  Controlar el número mínimo de empleados asignados a cada atracción y gestionar la disponibilidad según la operación del día. * **Administración de Empleados** Registrar y modificar la información de los empleados (datos personales, rol, certificaciones, etc.). Asignar turnos de apertura y cierre a cada empleado, especificando su lugar de trabajo (atracción, cafetería, tienda o taquilla). Validar que un empleado cuente con la capacitación adecuada para operar una atracción de riesgo medio o alto. Permitir a los empleados consultar sus turnos y lugares de asignación. Registrar la asistencia de los empleados a sus turnos. * **Venta e Inventario de Tiquetes**  Registrar la venta de tiquetes en línea y en taquillas. Soportar múltiples tipos de tiquetes (Básico, Familiar, Oro, Diamante, Temporada, Individual y FastPass). Aplicar descuentos especiales a empleados.  Validar el uso de un tiquete (para prevenir fraudes) y marcarlo como usado.. Llevar un inventario o historial de ventas y permitir generar reportes básicos de ventas. * **Control de Acceso a las Atracciones**  Verificar en tiempo real si un tiquete es válido para ingresar a una atracción (exclusividad, disponibilidad, uso previo).. Restringir la entrada a atracciones mecánicas según altura, peso y contraindicaciones.. Restringir la entrada a atracciones culturales por edad, si aplica.  Registrar el uso de un tiquete (especialmente en atracciones con cupo máximo). * **Operación de Temporada y Clima** Marcar atracciones y espectáculos de temporada con rango de fechas disponible.  Deshabilitar una atracción o espectáculo si las condiciones climáticas lo impiden. * **Autenticación y Seguridad** Permitir la creación y administración de credenciales (login y password) para cada usuario. Verificar la identidad de los usuarios antes de permitir la ejecución de operaciones sensibles (gestión de empleados, administración de atracciones, etc.). |

|  |
| --- |
| 1. **Requerimientos No Funcionales** |
| * **Persistencia de Datos**   Toda la información debe almacenarse en archivos (planos o binarios), en una carpeta distinta del código fuente.  La estructura de los archivos (cuántos, cómo se dividen los datos) debe diseñarse cuidadosamente para facilitar la lectura/escritura y evitar inconsistencias.  Debe haber mecanismos básicos de respaldo y restauración de datos (por ejemplo, copias de seguridad periódicas).   * **Plataforma y Lenguaje**   La aplicación debe implementarse en Java.  Inicialmente no se requiere interfaz gráfica, pero sí programas de demostración que muestren la lógica y funcionalidades.   * **Seguridad y Autenticación**   Se requiere un login y password para cada usuario (empleado, administrador o cliente).  El sistema debe validar las credenciales y permitir la administración de contraseñas (cambio, recuperación, etc.).  No se contemplan ataques maliciosos directos a los archivos, pero se asume que el acceso está restringido al sistema.   * **Rendimiento y Escalabilidad**   El sistema debe ser capaz de manejar un número considerable de atracciones, empleados y tiquetes, sin un tiempo de respuesta excesivo.  El diseño debe permitir la evolución a una arquitectura más compleja (por ejemplo, integración con base de datos) si se requiere en proyectos futuros.   * **Usabilidad y Extensibilidad**   Aunque en la primera etapa no se desarrollará una interfaz de usuario sofisticada, la aplicación debe diseñarse de manera que sea fácil de mantener y escalar (por ejemplo, para agregar una interfaz de consola o gráfica).  El sistema debe permitir la adición de nuevos tipos de tiquetes o modalidades de atracciones con un mínimo de cambios en la lógica base.   * **Confiabilidad** El sistema debe evitar pérdidas de información y manejar con cuidado las transacciones de venta y actualización de datos.   Si se produjera un fallo, la recuperación de los datos almacenados debe ser posible sin comprometer la consistencia del sistema.   * **Conformidad con Normativas** Deben respetarse las regulaciones relacionadas con la operación de atracciones mecánicas (niveles de riesgo, límites de altura y peso) y la protección de datos de los usuarios.  De ser necesario, el sistema debe permitir ajustes regionales o legales (por ejemplo, diferentes políticas de privacidad). |

|  |
| --- |
| 1. **Restricciones del proyecto** |
| * **Restricciones técnicas** * Lenguaje obligatorio: La aplicación debe desarrollarse en Java. * Persistencia obligatoria: Toda la información debe ser persistente. Debe almacenarse en archivos (planos o binarios). Los archivos deben estar en una carpeta separada del código fuente. Autenticación obligatoria: Todos los usuarios del sistema deben tener login y contraseña. * **Restricciones del dominio del problema**   **Atracciones:**   * + - * Cada atracción tiene:       * - Cupo máximo de personas. -Número mínimo de empleados para operar.       * Las atracciones mecánicas tienen:   Restricciones de altura y peso.  Posibles restricciones médicas (vértigo, problemas cardiacos, discapacidades). - Un nivel de riesgo (medio o alto).   * + - * Las atracciones culturales pueden restringirse solo por edad.       * El funcionamiento de atracciones y espectáculos puede estar restringido por condiciones climáticas.       * Nivel de exclusividad para atracciones (Familiar, Oro, Diamante).       * Espectáculos:   - No tienen ubicación fija.   * + - * + - No tienen restricciones por nivel de tiquete.   **Empleados:**   * + Solo empleados capacitados pueden operar atracciones mecánicas.   -Para riesgo alto, solo empleados entrenados específicamente en esa atracción.   * Un empleado puede estar capacitado en varias atracciones, pero no intercambiarlas arbitrariamente. * Cocineros pueden cubrir caja, pero empleados comunes no pueden cubrir cocina. * Espectáculos no requieren empleados del parque: están tercerizados. * Cada lugar de servicio debe tener mínimo un cajero. * Un empleado puede estar asignado a diferentes roles/lugares según el turno y día. * Turnos diarios: dos por día (apertura y cierre).   **Tiquetes:**   * + - * + Tipos de tiquete: Básico, Familiar, Oro, Diamante, de temporada, individuales.         + Restricciones:   + Tiquetes Básico no permiten entrar a ninguna atracción.   + Restricciones por nivel de exclusividad según el tiquete.   + Verificaciones (edad, peso, altura) no se hacen en la venta, sino al ingreso.   + FastPass solo se puede comprar para días específicos, sin importar tipo de tiquete.   + Debe poder detectarse si un tiquete ya fue usado.   - Empleados tienen descuentos al comprar tiquetes.   * **Restricciones de uso/funcionalidad** * El administrador es el único autorizado para modificar datos de atracciones y empleados. * Los empleados y clientes solo pueden consultar cierta información. * El sistema debe permitir que empleados consulten sus tareas y lugares asignados. * Empleados pueden trabajar ambos turnos (horas extra). * Los cajeros de taquilla son responsables de asociar las ventas de tiquetes con los compradores. |

|  |
| --- |
| 1. **Programas de Prueba** |
| * **Prueba de Gestión de Atracciones**   Demostrará: Creación de atracciones mecánicas y culturales con sus respectivas restricciones  Verificará: Que las restricciones de altura/peso/edad se aplican correctamente  Mostrará: Listado de atracciones disponibles según condiciones climáticas y temporada   * **Prueba de Gestión de Empleados**   Demostrará: Asignación de empleados a turnos en diferentes lugares  Verificará: Que las reglas de capacitación para atracciones de riesgo se cumplen  Mostrará: Listado de empleados y sus asignaciones diarias   * **Prueba de Venta de Tiquetes**   Demostrará: Creación de diferentes tipos de tiquetes (Familiar, Oro, Diamante, Temporada)  Verificará: Que los niveles de exclusividad se aplican correctamente  Mostrará: Cálculo de precios con y sin descuento de empleado   * **Prueba de Validación de Acceso**   Demostrará: Validación de tiquetes contra atracciones específicas  Verificará: Que las restricciones de exclusividad, altura, peso y edad se aplican  Mostrará: Mensajes de error cuando no se cumplen las condiciones de acceso   * **Prueba de Persistencia de Datos**   Demostrará: Carga y guardado de datos en archivos  Verificará: Que todos los objetos principales se guardan y recuperan correctamente  Mostrará: Estado del sistema antes y después de operaciones de guardado/carga |

|  |
| --- |
| 1. **Documentación de la persistencia** |
| * El módulo de persistencia funciona mediante un sistema organizado en capas que permite guardar y recuperar los datos del parque de atracciones en archivos JSON. La estructura se compone de una \*\*interfaz base (IPersistencia)\*\* que define las operaciones esenciales (guardar y cargar), una \*\*clase abstracta (CorePersistencia)\*\* que implementa la lógica común (como la conversión a JSON y manejo de archivos), y clases específicas (como PersistenciaTiquetes o PersistenciaAtracciones) que se especializan en cada entidad del sistema. Cuando la aplicación necesita guardar datos (por ejemplo, una lista de tiquetes), la clase correspondiente convierte los objetos a formato JSON y los almacena en archivos separados dentro de la carpeta `/data`, respetando así el requisito de mantener los datos separados del código fuente.  ºPara recuperar la información, el proceso se invierte: las clases de persistencia leen los archivos JSON, reconstruyen los objetos Java correspondientes y los devuelven a la aplicación. El sistema incluye \*\*validaciones\*\* para asegurar la integridad de los datos (como verificar que los objetos tengan la estructura correcta antes de guardarlos) y manejo de errores para casos como archivos corruptos o permisos insuficientes. Esta arquitectura modular permite que cada componente (tiquetes, atracciones, empleados) tenga su propio manejo de persistencia, pero compartiendo la lógica común de serialización y acceso a archivos, lo que facilita el mantenimiento y el cumplimiento de los requisitos del proyecto. |

|  |
| --- |
| 1. **Asignación de Responsabilidades y Justificaciones** |
| Clase ParqueAtracciones  Responsabilidades:   * Gestionar el catálogo completo de atracciones (mecánicas y culturales) y espectáculos * Administrar el registro de empleados y usuarios del sistema * Controlar el inventario de tiquetes vendidos * Sincronizar los datos con el sistema de persistencia * Servir como interfaz principal para operaciones administrativas   Justificación:  Centraliza la coordinación del sistema, lo cual permite un punto de entrada claro y facilita la interacción con los diferentes módulos (tiquetes, usuarios, atracciones).  Cumple con el principio de **Alta Cohesión**, ya que agrupa funcionalidades relacionadas directamente con la operación del parque.  La responsabilidad de sincronización con persistencia evita duplicidad de lógica en otros componentes, manteniendo SRP.  Clases de Atracciones  Atracción (Abstracta)  Responsabilidades:   * Mantener información base de todas las atracciones (ID, nombre, capacidad, ubicación) * Gestionar el estado operativo y mantenimiento * Validar acceso básico según nivel de exclusividad * Proveer interfaz común para operaciones de registro y actualización   Justificación:  La abstracción permite reutilizar lógica común (nombre, capacidad, ubicación), evitando duplicación de código.  Las clases AtracciónMecánica y AtracciónCultural implementan comportamientos específicos, respetando el **Principio de Abierto/Cerrado** (OCP): se pueden extender sin modificar la clase base.  Se cumple el **Principio de Sustitución de Liskov**: cualquier objeto de tipo Atracción puede ser tratado de manera uniforme, sin importar si es mecánica o cultural.  AtracciónMecánica  Responsabilidades:   * Gestionar parámetros de seguridad (límites altura/peso) * Controlar nivel de riesgo y contraindicaciones médicas * Validar restricciones específicas para visitantes * Manejar operaciones de inicio/detención con controles de seguridad   AtracciónCultural  Responsabilidades:   * Administrar restricción de edad mínima * Gestionar información temática de la atracción * Controlar eventos culturales (inicio/finalización) * Validar ingreso según edad del visitante   Jerarquía de Usuarios  Usuario (Abstracto)  Responsabilidades:   * Gestionar información básica de autenticación (login, password) * Proveer funcionalidad base para cambio de contraseña * Servir como clase padre para tipos específicos de usuarios   Justificación:    La clase abstracta Usuario permite generalizar operaciones comunes como autenticación, mientras que Cliente, Empleado y Administrador manejan lógica propia.   Fomenta el uso del Polimorfismo: operaciones sobre usuarios pueden ejecutarse sin conocer su tipo concreto.  Facilita escalabilidad del sistema (por ejemplo, permitir diferentes interfaces o permisos según el tipo de usuario).    Cliente  Responsabilidades:   * Mantener historial de compras de tiquetes * Realizar compras de nuevos tiquetes * Consultar información de sus compras anteriores   Empleado  Responsabilidades:   * Gestionar información laboral (cargo, certificaciones) * Controlar asignación de turnos y disponibilidad * Registrar asistencia a turnos asignados * Proveer funcionalidad base para todos los tipos de empleados   Administrador  Responsabilidades:   * Gestionar el registro completo de usuarios * Realizar operaciones CRUD sobre cualquier entidad del sistema * Servir como superusuario con acceso total   Tipos Especializados de Empleados  EmpleadoServicio  Responsabilidades:  Manejar operaciones de venta en puntos de servicio  Generar facturas para transacciones Gestionar servicios generales cuando no está asignado a ubicación específica   Justificación:   Distinguir entre EmpleadoAtracción y EmpleadoServicio permite encapsular lógica específica y necesaria para operaciones seguras o de venta.  Sigue el **SRP**: cada clase maneja una única responsabilidad.  Permite que futuros tipos de empleados puedan ser fácilmente añadidos sin afectar la jerarquía actual (OCP).  EmpleadoAtracción  Responsabilidades:   * Controlar operación segura de atracciones mecánicas * Verificar tiquetes de visitantes * Gestionar capacitaciones específicas para atracciones de alto riesgo   Lugares de Servicio  Cafetería  Responsabilidades:   * Gestionar preparación de alimentos (operaciones de cocina) * Controlar venta de productos alimenticios * Mantener información de ubicación y menú   Justificación:   Separar cada tipo de lugar permite encapsular operaciones específicas como cocina, ventas o tiquetería.  Cumple con SRP al tener una única responsabilidad operativa.  Refleja fielmente las restricciones de dominio: los empleados no se comparten entre atracciones y servicios.  Tienda  Responsabilidades:   * Administrar venta de productos comerciales * Gestionar inventario de productos disponibles   Taquilla  Responsabilidades:   * Controlar venta física de tiquetes * Mantener registro de tiquetes vendidos/disponibles * Gestionar transacciones en punto de venta   Jerarquía de Tiquetes  Tiquete (Base Abstracta)  Responsabilidades:   * Mantener información común de todos los tiquetes (ID, estado de uso) * Gestionar lista de atracciones accesibles * Controlar aplicación de descuentos * Proveer interfaz común para operaciones de compra * Justificaicion:   Uso de una clase abstracta permite centralizar información común (estado de uso, ID, atracciones permitidas).   Las especializaciones (Normal, Temporada, FastPass) permiten modelar correctamente reglas de negocio: accesos, fechas, descuentos.  Aplica OCP y Liskov, ya que los tiquetes se pueden tratar de forma polimórfica sin alterar la lógica general del sistema.  TiqueteNormal  Responsabilidades:   * Implementar lógica de tiquetes estándar (Familiar, Oro, Diamante) * Gestionar acceso básico a atracciones según nivel * Controlar estado de uso (utilizado/no utilizado)   TiqueteTemporada  Responsabilidades:   * Administrar rango de fechas de validez * Controlar acceso ilimitado durante período contratado * Gestionar descuentos por compra de temporada   TiqueteFastPass  Responsabilidades:   * Manejar fecha específica de validez * Controlar privilegios de acceso rápido * Gestionar integración con sistema de filas   Clases de Soporte  Turno  Responsabilidades:   * Gestionar información horaria (inicio/fin) * Mantener datos de ubicación temporal asignada * Controlar tipo de turno (apertura/cierre) * Administrar puesto específico asignado   Justificación:  Turno separa la lógica temporal del sistema, permitiendo modelar disponibilidad y asignación. Producto centraliza datos comunes para cafetería y tienda, fomentando reutilización. Estas clases cumplen el SRP y refuerzan la modularidad del sistema.  Producto  Responsabilidades:   * Mantener información básica de productos/platos * Gestionar precios y nombres de items * Servir como base para productos de cafetería/tienda   Enumeraciones  PuestoAsignado  Responsabilidad:   * Definir los posibles roles laborales en el parque * Establecer categorías claras para asignación de personal * Justificación:   Las enumeraciones garantizan integridad de datos y facilitan validación.   Mejoran legibilidad del código y previenen errores por valores arbitrarios.  Son consistentes con los requisitos del dominio (distintos niveles de acceso, funciones laborales fijas).  TipoTiquete  Responsabilidad:   * Especificar niveles de exclusividad disponibles * Definir jerarquía de acceso a atracciones   Esta asignación de responsabilidades cumple con los principios SOLID, especialmente con el Principio de Responsabilidad Única (SRP), |