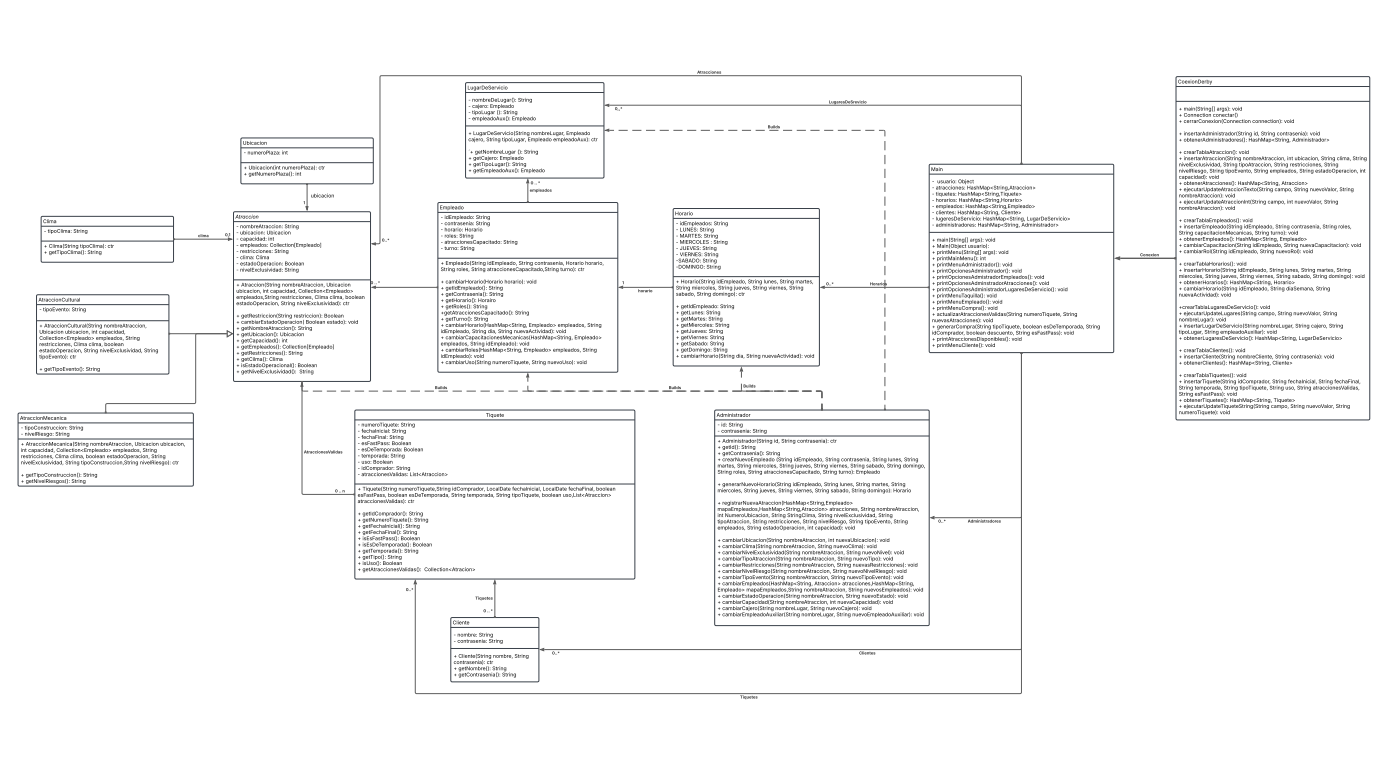
Pablo Medina Forero – 202323276

Juan David Suárez - 202223801

Sebastián Barbosa – 202316512

Proyecto 3:

1. Modelo UML:



1. Descripcion del código:

Clase 1: VentanaAgregarAdministrador

Responsabilidad: Crear una interfaz para agregar administradores al sistema.

Atributos:

• JTextField txtId: Captura el ID del administrador.

• JPasswordField txtContrasenia: Captura la contraseña.

Métodos:

• public VentanaAgregarAdministrador(): Constructor que define la ventana y sus componentes.

• btnAgregar.addActionListener(...): Verifica campos y llama a conexion.insertarAdministrador(id, contrasenia).

• btnCancelar.addActionListener(...): Cierra la ventana con dispose().

Clase 2: VentanaAgregarEmpleado

Responsabilidad: Registrar nuevos empleados y sus horarios.

Atributos:

• JTextField txtId, txtContrasenia, txtRoles, txtCapacitacion, txtTurno: Capturan datos del nuevo empleado.

• JTextField[] camposDias: Captura la actividad para cada día de la semana.

Métodos:

• public VentanaAgregarEmpleado(JFrame ventanaAnterior): Constructor de la ventana.

• btnAgregar.addActionListener(new ActionListener() {...}): Valida datos y ejecuta:

• conexion.insertarEmpleado(...)

• conexion.insertarHorario(...)

• btnCancelar.addActionListener(...): Retorna a la ventana anterior y cierra actual.

Clase 3: VentanaAgregarLugarDeServicio

Responsabilidad: Crear un nuevo lugar de servicio.

Atributos:

• JTextField txtNombre, txtCajero, txtTipo, txtEmpleadoAuxiliar: Capturan datos del lugar.

Métodos:

• public VentanaAgregarLugarDeServicio(): Constructor de la interfaz.

• btnAgregar.addActionListener(...): Llama a conexion.insertarLugarDeServicio(...).

• btnCancelar.addActionListener(...): Cierra la ventana.

⸻

Clase 4: VentanaCambiarCapacitaciones

Responsabilidad: Mostrar y actualizar capacitaciones de un empleado.

Atributos:

• JTextField txtIdEmpleado, txtNuevaCapacitacion

• JTextArea txtCapacitacionesActuales

• HashMap<String, Empleado> mapaEmpleados

Métodos:

• public VentanaCambiarCapacitaciones(JFrame ventanaAnterior)

• btnVerActuales.addActionListener(...): Muestra capacitaciones actuales desde el mapa.

• btnActualizar.addActionListener(...): Llama a conexion.cambiarCapacitacion(...).

• btnCancelar.addActionListener(...): Vuelve a la ventana anterior.

⸻

Clase 5: VentanaCambiarHorario

Responsabilidad: Permite cambiar la actividad de un empleado para un día.

Atributos:

• JTextField txtIdEmpleado, txtNuevaActividad

• JComboBox comboDias

Métodos:

• public VentanaCambiarHorario(JFrame ventanaAnterior)

• btnCambiar.addActionListener(...): Llama a conexion.cambiarHorario(...) después de validar.

• btnCancelar.addActionListener(...): Vuelve atrás.

⸻

Clase 6: VentanaCambiarInformacionAtraccion

Responsabilidad: Actualizar un campo específico de una atracción.

Atributos:

• JComboBox comboCampos

• JTextField txtNuevoValor, txtNombreAtraccion

Métodos:

• public VentanaCambiarInformacionAtraccion(JFrame parent)

• btnActualizar.addActionListener(...): Usa:

• ConexionDerby.ejecutarUpdateAtraccionTexto(...)

• ConexionDerby.ejecutarUpdateAtraccionInt(...)

Clase 7: VentanaCambiarRol

Responsabilidad: Mostrar y actualizar el rol de un empleado.

Atributos:

• JTextField txtIdEmpleado, txtNuevoRol

• JTextArea txtRolActual

• HashMap<String, Empleado> mapaEmpleados

Métodos:

• public VentanaCambiarRol(JFrame ventanaAnterior)

• btnVerRol.addActionListener(...): Muestra el rol actual.

• btnActualizar.addActionListener(...): Llama a conexion.cambiarRol(...).

• btnCancelar.addActionListener(...): Vuelve atrás.

Clase 8: VentanaDetallesEmpleado

Responsabilidad: Mostrar los datos de un empleado y su horario.

Atributos:

• Empleado empleado

• Horario horario

Métodos:

• public VentanaDetallesEmpleado(Empleado empleado, Horario horario)

• btnCerrar.addActionListener(...): Cierra la ventana.

Clase 9: VentanaModificarLugarDeServicio

Responsabilidad: Modificar información de un lugar de servicio.

Atributos:

• JTextField txtNombre, txtNuevoValor

• JComboBox comboCampos

Métodos:

• public VentanaModificarLugarDeServicio()

• btnActualizar.addActionListener(...): Llama a conexion.ejecutarUpdateLugares(...).

• btnCancelar.addActionListener(...): Cierra ventana.

Clase 10: VentanaOpcionesAtracciones

Responsabilidad: Menú de acciones para atracciones.

Métodos:

• public VentanaOpcionesAtracciones()

• btnRegistrar.addActionListener(...): Abre VentanaRegistrarAtraccion.

• btnModificar.addActionListener(...): Abre VentanaCambiarInformacionAtraccion.

• btnVerAtracciones.addActionListener(...): Abre VentanaVerAtracciones.

Clase 11: VentanaOpcionesEmpleado

Responsabilidad: Menú de acciones para empleados.

Métodos:

• public VentanaOpcionesEmpleado(JFrame ventanaAnterior)

• btnCrear.addActionListener(...): Abre VentanaAgregarEmpleado.

• btnCambiarHorario.addActionListener(...)

• btnCambiarCapacitaciones.addActionListener(...)

• btnCambiarRoles.addActionListener(...)

• btnVerEmpleados.addActionListener(...)

Clase 12: VentanaOpcionesLugaresDeServicio

Responsabilidad: Menú para lugares de servicio.

Métodos:

• public VentanaOpcionesLugaresDeServicio()

• btnInsertar.addActionListener(...): Abre VentanaAgregarLugarDeServicio.

• btnModificar.addActionListener(...): Abre VentanaModificarLugarDeServicio.

⸻

Clase 13: VentanaRegistrarAtraccion

Responsabilidad: Registrar nueva atracción.

Atributos:

• Muchos JTextField para datos de atracción.

Métodos:

• public VentanaRegistrarAtraccion(JFrame parent, HashMap<String, Empleado> mapaEmpleados)

• btnAgregar.addActionListener(...): Llama a registrarNuevaAtraccion(...).

• private void registrarNuevaAtraccion(...): Valida y usa conexion.insertarAtraccion(...).

⸻

Clase 14: VentanaVerEmpleados

Responsabilidad: Mostrar lista y detalles de empleados.

Métodos:

• public VentanaVerEmpleados()

• btnVerDetalles.addActionListener(...): Llama a VentanaDetallesEmpleado.

⸻

Clase 15: Ventana

Responsabilidad: Pantalla de bienvenida principal.

Métodos:

• public static void main(String[] args)

• btnAdmin.addActionListener(...): Abre VentanaAdministrador.

• btnEmpleado.addActionListener(...): Abre VentanaEmpleado.

• btnCliente.addActionListener(...): Abre VentanaCliente.

⸻

Clase 16: VentanaAdministrador

Responsabilidad: Panel del administrador tras autenticación.

Métodos:

• public VentanaAdministrador()

• btnGestionAtracciones.addActionListener(...)

• btnGestionEmpleados.addActionListener(...)

• btnGestionAdmins.addActionListener(...)

• btnGestionLugaresDeServicio.addActionListener(...)

Clase 17: VentanaCliente

Responsabilidad: Interfaz del cliente autenticado.

Métodos:

• public VentanaCliente()

Clase 18: VentanaEmpleado

Responsabilidad: Panel de opciones para empleados.

Métodos:

• public VentanaEmpleado()

Clase 19: VentanaLogin

Responsabilidad: Login para administradores, clientes o empleados.

Atributos:

• boolean autenticado

• HashMap<String, Usuario> mapa

Métodos:

• public VentanaLogin(JFrame parent, String tipoInicio)

• btnIngresar.addActionListener(...): Lógica de autenticación.

• public boolean mostrarDialogo()

Clase 20: VentanaVerAtracciones

Responsabilidad: Mostrar lista y detalles de atracciones.

Atributos:

• HashMap<String, Atraccion> mapaAtracciones

Métodos:

• public VentanaVerAtracciones(HashMap<String, Atraccion> mapaAtracciones)

• btnVerDetalles.addActionListener(...)

• private void mostrarInformacion(String nombreAtraccion)

# Clase 21: VentanaHorarioEmpleado

Responsabilidad: Mostrar el turno y el horario detallado de un empleado.

* Atributos:
* • Empleado empleado: Información del empleado.
* • Horario horario: Datos de su horario.
* Métodos:
* • public VentanaHorarioEmpleado(Empleado empleado, Horario horario): Constructor que inicializa la ventana con el turno y horario semanal.
* • btnCerrar.addActionListener(...): Cierra la ventana al hacer clic.

# Clase 22: VentanaId

Responsabilidad: Pedir el ID del empleado para mostrar su horario.

* Atributos:
* • JTextField txtId: Campo para ingresar el ID.
* • HashMap<String, Empleado> mapaEmpleados: Mapa de empleados traído desde la base de datos.
* Métodos:
* • public VentanaId(): Constructor que pide el ID y, si existe, muestra su horario usando VentanaHorarioEmpleado.

# Clase 23: VentanaOpcionesComprarTiquetesEmpleados

Responsabilidad: Permitir a empleados comprar tiquetes personalizados.

* Atributos:
* • JComboBox para temporada, tipo de tiquete y fast pass.
* • JTextField txtNombre, txtAtracciones: Capturan nombre del cliente y atracciones seleccionadas.
* Métodos:
* • public VentanaOpcionesComprarTiquetesEmpleados(): Constructor que configura la interfaz.
* • btnComprar.addActionListener(...): Valida campos, filtra atracciones y llama a conexion.insertarTiquete(...).

# Clase 24: VentanaIdCliente

Responsabilidad: Permitir al cliente ingresar su ID para ver sus tiquetes.

* Atributos:
* • JTextField txtId: Entrada del ID.
* • HashMap<String, Cliente> mapaClientes: Clientes registrados.
* Métodos:
* • public VentanaIdCliente(HashMap<String, Tiquete> todosTiquetes): Si el cliente existe, filtra y abre VentanaOpccionVerTiquetes con sus tiquetes.

# Clase 25: VentanaOpccionVerTiquetes

Responsabilidad: Mostrar lista de tiquetes de un cliente y sus detalles.

* Atributos:
* • HashMap<String, Tiquete> mapaTiquetes: Mapa con los tiquetes del cliente.
* Métodos:
* • public VentanaOpccionVerTiquetes(...): Construye la lista.
* • btnVerDetalles.addActionListener(...): Muestra detalles con mostrarInformacion(...).

# Clase 26 VentanaOpcionesComprarTiqueteClientes

Responsabilidad: Permitir a los clientes comprar tiquetes personalizados.

* Métodos:
* • public VentanaOpcionesComprarTiqueteClientes(): Constructor y lógica similar a la clase de empleados.

# Clase 27: Ventana

Responsabilidad: Pantalla de bienvenida principal con acceso a los roles.

* Métodos:
* • public static void main(String[] args): Inicializa la app.
* • Botones para abrir: VentanaAdministrador, VentanaEmpleado, VentanaCliente, VentanaCrearCuenta.

# Clase 28: VentanaAdministrador

Responsabilidad: Panel para administradores luego del login.

* Métodos:
* • public VentanaAdministrador(): Verifica credenciales y muestra opciones para gestionar recursos.

# Clase 29: VentanaCliente

Responsabilidad: Panel de cliente autenticado.

* Métodos:
* • public VentanaCliente(): Login y opciones como comprar entradas y ver tiquetes.

# Clase 30: VentanaEmpleado

Responsabilidad: Panel para empleados autenticados.

* Métodos:
* • public VentanaEmpleado(): Login y acceso a funciones como ver horarios o comprar tiquetes.

# Clase 31: VentanaLogin

Responsabilidad: Autenticación para usuarios.

* Atributos:
* • boolean autenticado: Resultado del login.
* • HashMap<String, Usuario> mapa: Usuarios cargados según tipo.
* Métodos:
* • public VentanaLogin(JFrame parent, String tipoInicio): Constructor que muestra campos de login.
* • public boolean mostrarDialogo(): Devuelve si fue exitoso.

# Clase 32: VentanaVerAtracciones

Responsabilidad: Mostrar lista de atracciones y sus detalles.

Atributos:

* • HashMap<String, Atraccion> mapaAtracciones: Todas las atracciones.
* Métodos:
* • public VentanaVerAtracciones(...): Constructor de la interfaz.
* • btnVerDetalles.addActionListener(...): Muestra detalles con mostrarInformacion(...).