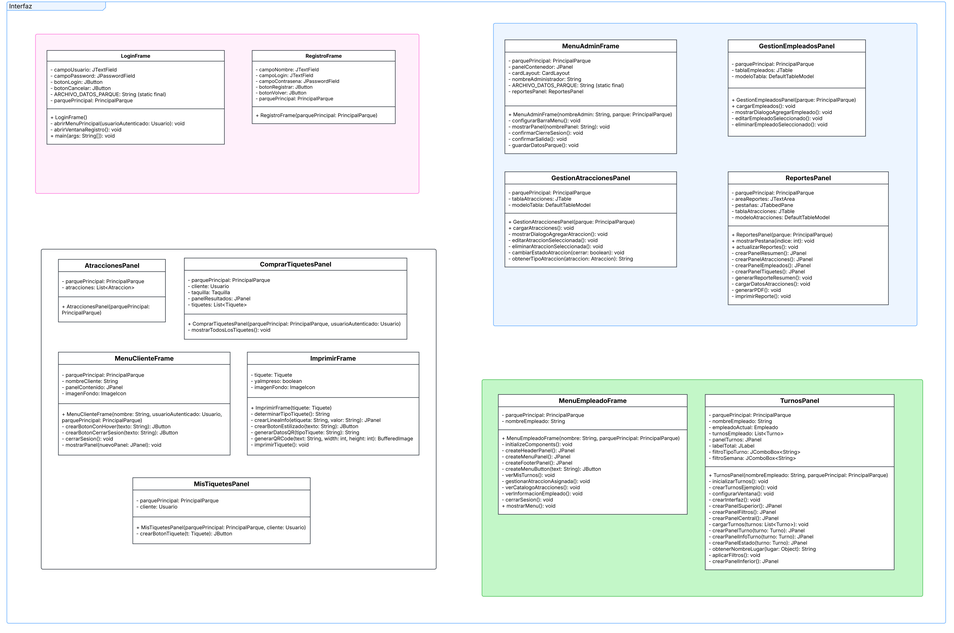
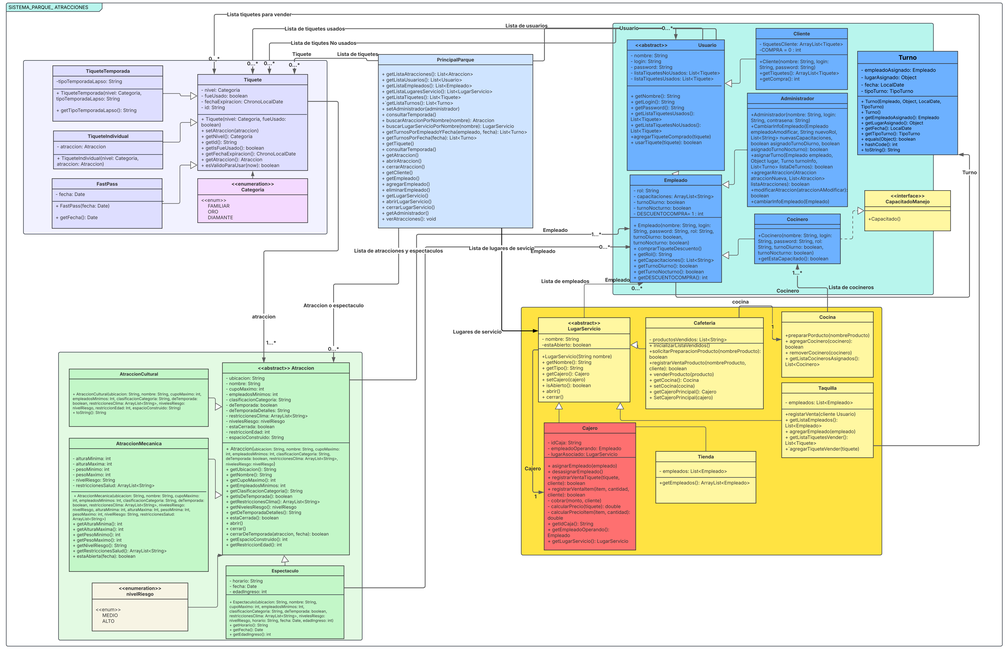
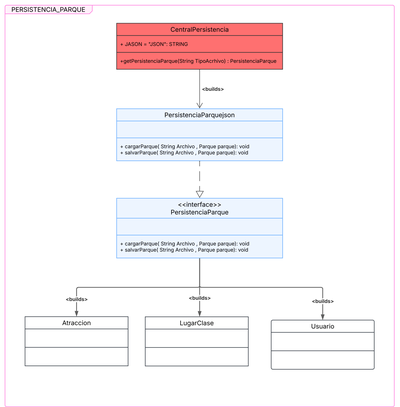

Interfaz
LoginFrame
- campoUsuario: JTextField
- campoPassword: JPasswordField
- botonLogin: JButton
- botonCancelar: JButton
- ARCHIVO_DATOS_PARQUE: String (static final)
- parquePrincipal: PrincipalParque
+ LoginFrame()
- abrirMenuPrincipal(usuarioAutenticado: Usuario): void
- abrirVentanaRegistro(): void
+ mainArgs: String[]): void
RegistroFrame
- campoNombre: JTextField
- campoLogin: JTextField
- campoContraseña: JPasswordField
- botonRegistrar: JButton
- botonVolver: JButton
- parquePrincipal: PrincipalParque
+ RegistroFrame(parquePrincipal: PrincipalParque)
MenuAdminFrame
- parquePrincipal: PrincipalParque
- panelContenedor: JPanel
- cardLayout: CardLayout
- nombreAdministrador: String
- ARCHIVO_DATOS_PARQUE: String (static final)
- reportesPanel: ReportesPanel
+ MenuAdminFrame(nombreAdmin: String, parque: PrincipalParque)
- configurarBarraMenu(): void
- mostrarPanel(nombrePanel: String): void
- cerrarSesionYVolverAlLogin(): void
- confirmarCierreDeSesión(): void
- guardarDatosParque(): void
GestionEmpleadosPanel
- parquePrincipal: PrincipalParque
- tablaEmpleados: JTable
- modeloTabla: DefaultTableModel
+ GestionEmpleadosPanel(parque: PrincipalParque)
+ cargarEmpleados(): void
- mostrarDialogoAgregarEmpleado(): void
- editarEmpleadoSeleccionado(): void
- eliminarEmpleadoSeleccionado(): void
GestionAtraccionesPanel
- parquePrincipal: PrincipalParque
- tablaAtracciones: JTable
- modeloTabla: DefaultTableModel
+ GestionAtraccionesPanel(parque: PrincipalParque)
+ cargarAtracciones(): void
- mostrarDialogoAgregarAtraccion(): void
- editarAtraccionSeleccionada(): void
- eliminarAtraccionSeleccionada(): void
- cambiarEstadoAtraccion(cerrar: boolean): void
- obtenerTipoAtraccion(atraccion: Atraccion): String
ReportesPanel
- parquePrincipal: PrincipalParque
- areaReportes: JTextArea
- pestañas: JTabbedPane
- tablaAtracciones: JTable
- modeloAtracciones: DefaultTableModel
+ ReportesPanel(parque: PrincipalParque)
+ mostrarPestana(indice: int): void
+ actualizarReportes(): void
- crearPanelResumen(): JPanel
- crearPanelAtracciones(): JPanel
- crearPanelEmpleados(): JPanel
- crearPanelTiquetes(): JPanel
- generarReporteResumen(): void
- cargarDatosAtracciones(): void
- generarPDF(): void
- imprimirReporte(): void
AtraccionesPanel
- parquePrincipal: PrincipalParque
- atracciones: List<Atraccion>
+ AtraccionesPanel(parquePrincipal: PrincipalParque)
ComprarTiquetesPanel
- parquePrincipal: PrincipalParque
- cliente: Usuario
- taquilla: Taquilla
- panelResultados: JPanel
- tiquetes: List<Tiquete>
+ ComprarTiquetesPanel(parquePrincipal: PrincipalParque, usuarioAutenticado: Usuario)
- mostrarTodosLosTiquetes(): void
MenuClienteFrame
- parquePrincipal: PrincipalParque
- nombreCliente: String
- panelContenido: JPanel
- imagenFondo: ImageIcon
+ MenuClienteFrame(nombre: String, usuarioAutenticado: Usuario, parquePrincipal: PrincipalParque)
- crearBotonOmnivoreDesde(texto: String): JButton
- crearBotonCentrarSeleccion(texto: String): JButton
- cerrarSesion(): void
- mostrarPanel(nuevoPanel: JPanel): void
ImprimirFrame
- tiquete: Tiquete
- yaImpreso: boolean
- imagenFondo: ImageIcon
+ ImprimirFrame(tiquete: Tiquete)
- determinarTipoTiquete(): String
- crearLineaInfo(etiqueta: String, valor: String): JPanel
- crearBotonOmnivoreDesde(texto: String): JButton
- generarDatosQRTipoTiquete(tipo: String): String
- generarQRCode(text: String, width: int, height: int): BufferedImage
- imprimirTiquete(): void
MisTiquetesPanel
- parquePrincipal: PrincipalParque
- cliente: Usuario
+ MisTiquetesPanel(parquePrincipal: PrincipalParque, cliente: Usuario)
- crearBotonTiquete(t: Tiquete): JButton
MenuEmpleadoFrame
- parquePrincipal: PrincipalParque
- nombreEmpleado: String
+ MenuEmpleadoFrame(nombre: String, parquePrincipal: PrincipalParque)
- inicializaComponentes(): void
- createHeaderPanel(): JPanel
- createMenuPanel(): JPanel
- createFooterPanel(): JPanel
- crearBotonMenuEmpleado(texto: String): JButton
- verMisTurnos(): void
- gestionarAtraccionAsignada(): void
- verCatalogoAtracciones(): void
- verInformacionEmpleado(): void
- cerrarSesion(): void
- mostrarMenu(): void
TurnosPanel
- parquePrincipal: PrincipalParque
- nombreEmpleado: String
- turnoAsignado: Turno
- turnosEmpleado: List<Turno>
- panelTurnos: JPanel
- labelTotal: JLabel
- filtroTipoTurno: JComboBox<String>
- filtroSemana: JComboBox<String>
+ TurnosPanel(nombreEmpleado: String, parquePrincipal: PrincipalParque)
- inicializarTurnos(): void
- crearTurnosEjemplo(): void
- configurarVentanaEs(): void
- crearInterfaz(): void
- crearPanelSuperior(): JPanel
- crearPanelFiltros(): JPanel
- crearPanelCentral(): JPanel
- crearPanelTarjetasTurnos(): void
- crearPanelTurnoDurno(turno: Turno): JPanel
- crearPanelInfoTurnoDurno(turno: Turno): JPanel
- crearPanelEstadoDurno(turno: Turno): JPanel
- obtenerNombreLugarSegur(lugar: Object): String
- aplicarFiltros(): void
- crearPanelInferior(): JPanel


SISTEMA_PARQUE_ATRACCIONES
Lista tiquetes para vender
Lista de tiquetes usados
Lista de tiquetes No usados
Lista de tiquetes
Lista de usuarios
Tiquete
TiqueteTemporada
- tipoTemporada:Lapso: String
+ TiqueteTemporada(nivel: Categoria, tipoTemporada:Lapso: String
+ getTipoTemporada:Lapso(): String
TiqueteIndividual
- atraccion: Atraccion
+ TiqueteIndividual(nivel: Categoria, atraccion: Atraccion)
FastPass
- fecha: Date
+ FastPass(fecha: Date)
+ getFecha(): Date
Tiquete
- nivel: Categoria
- fueUsado: boolean
- fechaExpiracion: ChronoLocalDate
- id: String
+ TiqueteIndividual: Categoria, fueUsado: boolean
+ getAtraccion(atraccion)
+ getNivel(): Categoria
+ getId(): String
+ getFueUsado(): boolean
+ getFechaExpiracion(): ChronoLocalDate
+ getAtraccion(): Atraccion
+ estaValidoParaUsar(nivel: boolean
<<enumeration>>
Categoria
+ <enum>>
FAMILIAR
ORO
DIAMANTE
PrincipalParque
+ getListaAtracciones(): List<Atraccion>
+ getListaUsuarios(): List<Usuario>
+ getListaEmpleados(): List<Empleado>
+ getListaLugaresServicio(): List<LugarServicio>
+ getListaTiquetes(): List<Tiquete>
+ getNombre(): String
+ setNombre(nombre: String)
+ consultarTemporada()
+ buscarAtraccionPorNombre(nombre): Atraccion
+ buscarLugarServicioPorNombre(nombre): LugarServicio
+ getTurnosPorEmpleadoEnFecha(empleado, fecha): List<Turno>
+ getTurnosPorFechaDeHoy(fecha): List<Turno>
+ getNombre()
+ consultarTemporada()
+ getAtraccion()
+ abrirAtraccion()
+ cerrarAtraccion()
+ getNombre()
+ getEmpleado()
+ agregarEmpleado()
+ eliminarEmpleado()
+ getLugarServicio()
+ agregarLugarServicio()
+ cerrarLugarServicio()
+ getAdministrador()
+ verAtracciones(): void
Usuario
- nombre: String
- login: String
- password: String
- listaTiquetesNoUsados: List<Tiquete>
- listaTiquetesUsados: List<Tiquete>
+ getNombre(): String
+ getLogin(): String
+ getPassword(): String
+ getListaTiquetesUsados(): List<Tiquete>
+ getListaTiquetesNoUsados(): List<Tiquete>
+ agregarTiqueteComprado(tiquete)
+ usarTiqueteEsquiado: boolean
Cliente
- tiquetesCliente: ArrayList<Tiquete>
- COMPRA = 2: int
+ Cliente(nombre: String, login: String, password: String)
+ getTiquetes(): ArrayList<Tiquete>
+ getComprar(): int
Administrador
+ Administrador(nombre: String, login: String, contraseña: String)
+ cambiarInfoEmpleado(Empleado empleadoAnodiFuer, String nuevoRol, List<String> nuevosCapacitaciones, boolean nuevoTurnoNocturno): boolean
asignarIdaTurnoNocturno(): boolean
+ asignarTurnoEmpleado(empleado, Object logar, Turno turnoInfo, List<Turno> listaDeTurnos): boolean
+ modificarAtraccion(atraccion, atraccionNueva, List<Atraccion> listaAtracciones): boolean
+ modificarAtraccion(atraccionAModificar): boolean
+ cambiarInfoEmpleado(Empleado)
Turno
- empleadoAsignado: Empleado
- lugarAsignado: Object
- fecha: LocalDate
- tipoTurno: TipoTurno
+ Turno(Empleado, Object, LocalDate, TipoTurno)
+ Turno()
+ getEmpleadoAsignado(): Empleado
+ getLugarAsignado(): Object
+ getFecha(): LocalDate
+ getTipoTurno(): TipoTurno
+ equals(Object): boolean
+ hashCode(): int
+ toString(): String
Empleado
- rol: String
- capacitaciones: ArrayList<String>
- turnoDiurno: boolean
- turnoNocturno: boolean
- DESCUENTODOMINA = 1: int
+ Empleado(nombre: String, login: String, password: String, rol: String, turnoDiurno: boolean, turnoNocturno: boolean)
+ comprarTiqueteDescuentado()
+ getRol(): String
+ getCapacitaciones(): List<String>
+ getTurnoDiurno(): boolean
+ getTurnoNocturno(): boolean
+ getDESCUENTODOMINA(): int
Cocinero
+ Cocinero(nombre: String, login: String, password: String, rol: String, turnoDiurno: boolean, turnoNocturno: boolean)
+ getEstaCapacitado(): boolean
<<interface>>
CapacitadoManejo
+ isCapacitado()
Turno
Empleado
Lista de atracciones y espectaculos
Lista de lugares de servicio
Lista de empleados
Lista de cocineros
Lugares de servicio
Atraccion o espectaculo
atraccion
cocina
Cocinero
Empleado
Usuario
Cliente
Administrador
Empleado
Cocinero
<<abstract>>
LugarServicio
- nombre: String
- estaAbierto: boolean
+ LugarServicio(String nombre)
+ getNombre(): String
+ getTipo(): String
+ getCajero(): Cajero
+ setCajero(cajero)
+ isAbierto(): boolean
+ abrir()
+ cerrar()
Cafeteria
- productosVendidos: List<String>
+ Cafeteria(productosVendidos)
+ solicitarPreparacionProducto(nombreProducto): boolean
+ registrarVentaProductos(nombreProducto, int cantidad)
+ venderProducto(producto)
+ getCocina(): Cocina
+ setCocina(cocina)
+ getCajeroPrincipal(): Cajero
+ setCajeroPrincipal(cajero)
Cocina
+ prepararPorductos(nombreProducto)
+ getCocineros(cocineros): boolean
+ removeCocinero(cocineros)
+ getListaCocinerosAsignados(): List<Cocinero>
Taquilla
- empleados: List<Empleado>
+registrarVenta(cliente Usuario)
+ getListaEmpleados()
+ setEmpleados()
+ agregarEmpleado(empleado)
+ getListaTiquetesVender():List<Tiquete>
+ agregarTiqueteVender(tiquete)
Cajero
- isCaja: String
- empleadoOperando: Empleado
- lugarAsociado: LugarServicio
+ asignarEmpleado(empleado)
+ desasignarEmpleado()
+ registrarVentaTiquete(tiquete, cliente): boolean
+ registrarVentaItem(item, cantidad, cliente): boolean
+ cobrar(monto, cliente)
+ calcularPrecioTiquete(): double
+ calcularPrecioItem(item, cantidad): double
+ getIsCaja(): String
+ getEmpleadoOperando()
+ setEmpleado()
+ getLugarServicio(): LugarServicio
Tienda
- empleados: List<Empleado>
+getEmpleados(): ArrayList<Empleado>
AtraccionCultural
+ AtraccionCultural(atraccion: String, nombre: String, cupoMaximo: int, edadMinima: int, alturaMinima: int, pesoMaximo: int, tipoTemporada: boolean, restriccionesClima: ArrayList<String>, nivelRiesgo: nivelRiesgo, restriccionEdad: int, espacioConstruido: String)
+ toString(): String
AtraccionMecanica
- alturaMinima: int
- alturaMaxima: int
- pesoMinimo: int
- pesoMaximo: int
- edadMinima: int
- restriccionesSalud: ArrayList<String>
+ AtraccionMecanica(atraccion: String, nombre: String, cupoMaximo: int, empleadoMinimo: int, clasificacionCategoria: String, deTemporada: boolean, restriccionesClima: ArrayList<String>, nivelesRiesgo: nivelRiesgo, alturaMinima: int, alturaMaxima: int, edadMinima: int, pesoMaximo: int, pesoMinimo: int, nivelRiesgo: String, restriccionesSalud: ArrayList<String>)
+ getAlturaMinima(): int
+ getAlturaMaxima(): int
+ getPesoMinimo(): int
+ getPesoMaximo(): int
+ getNivelRiesgo(): String
+ getRestriccionesSalud(): ArrayList<String>
+ estaAptoParaSalud(fecha): boolean
<<enumeration>>
nivelRiesgo
+ <enum>>
MEDIO
ALTO
<<abstract>> Atraccion
- ubicacion: String
- nombre: String
- cupoMaximo: int
- empleadosMinimos: int
- clasificacionCategoria: String
- deTemporada: boolean
- restriccionesClima: String
- restriccionesClima: ArrayList<String>
- nivelesRiesgo: nivelRiesgo
- estaCerrada: boolean
- restriccionEdad: int
- espacioConstruido: String
+ Atraccion(Ubicacion: String, nombre: String, cupoMaximo: int, empleadosMinimos: int, clasificacionCategoria: String, deTemporada: boolean, restriccionesClima: ArrayList<String>, nivelesRiesgo: nivelRiesgo)
+ getUbicacion(): String
+ getNombre(): String
+ getCupoMaximo: int
+ getEmpleadosMinimos: int
+ getClasificacionCategoria(): String
+ getDeTemporada: boolean
+ getRestriccionesClima: ArrayList<String>
+ getNivelesRiesgo(): nivelRiesgo
+ getDeTemporadaDetalle(): String
+ estaCerrada(): boolean
+ abrir()
+ cerrar()
+ cerrarDeTemporadaEstaAtraccion, fecha): boolean
+ getEspacioConstruido(): int
+ getRestriccionEdad(): int
Espectaculo
- horario: String
- fecha: Date
- edadEsperaso: int
+ Espectaculo(ubicacion: String, nombre: String, cupoMaximo: int, empleadosMinimos: int, clasificacionCategoria: String, deTemporada: boolean, restriccionesClima: ArrayList<String>, nivelesRiesgo: nivelRiesgo, horario: String, fecha: Date, edadEsperaso: int)
+ getHorario(): String
+ getFecha(): Date
+ getEdadEsperaso(): int


PERSISTENCIA_PARQUE
CentralPersistencia
+ JASON = "JSON": STRING
+getPersistenciaParque(String TipoArchivo) : PersistenciaParque
<builds>
PersistenciaParquejson
+ cargarParque(String Archivo , Parque parque): void
+ salvarParque(String Archivo , Parque parque): void
<<interface>>
PersistenciaParque
+ cargarParque(String Archivo , Parque parque): void
+ salvarParque(String Archivo , Parque parque): void
<builds>
<builds>
<builds>
Atraccion
LugarClase
Usuario