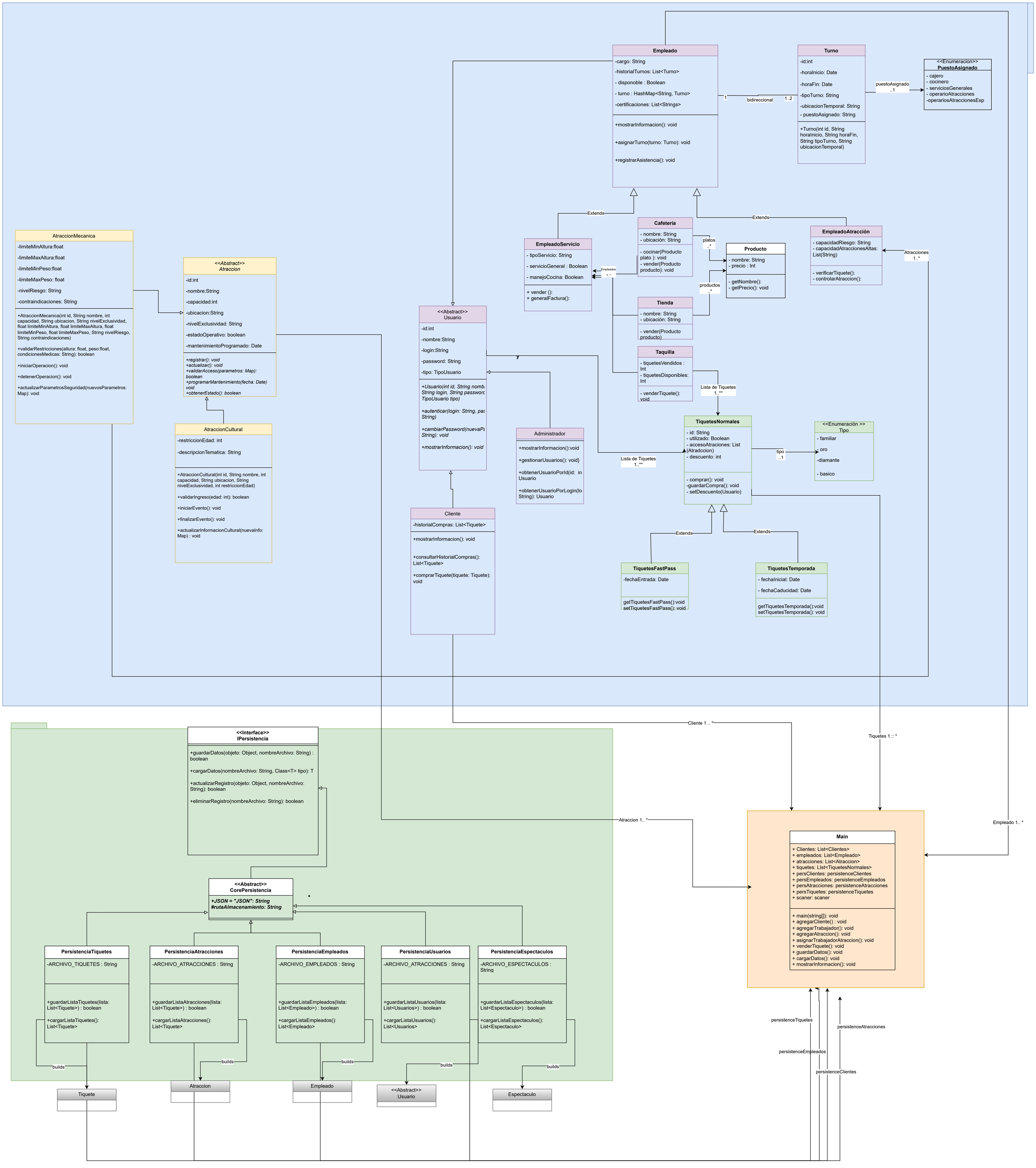
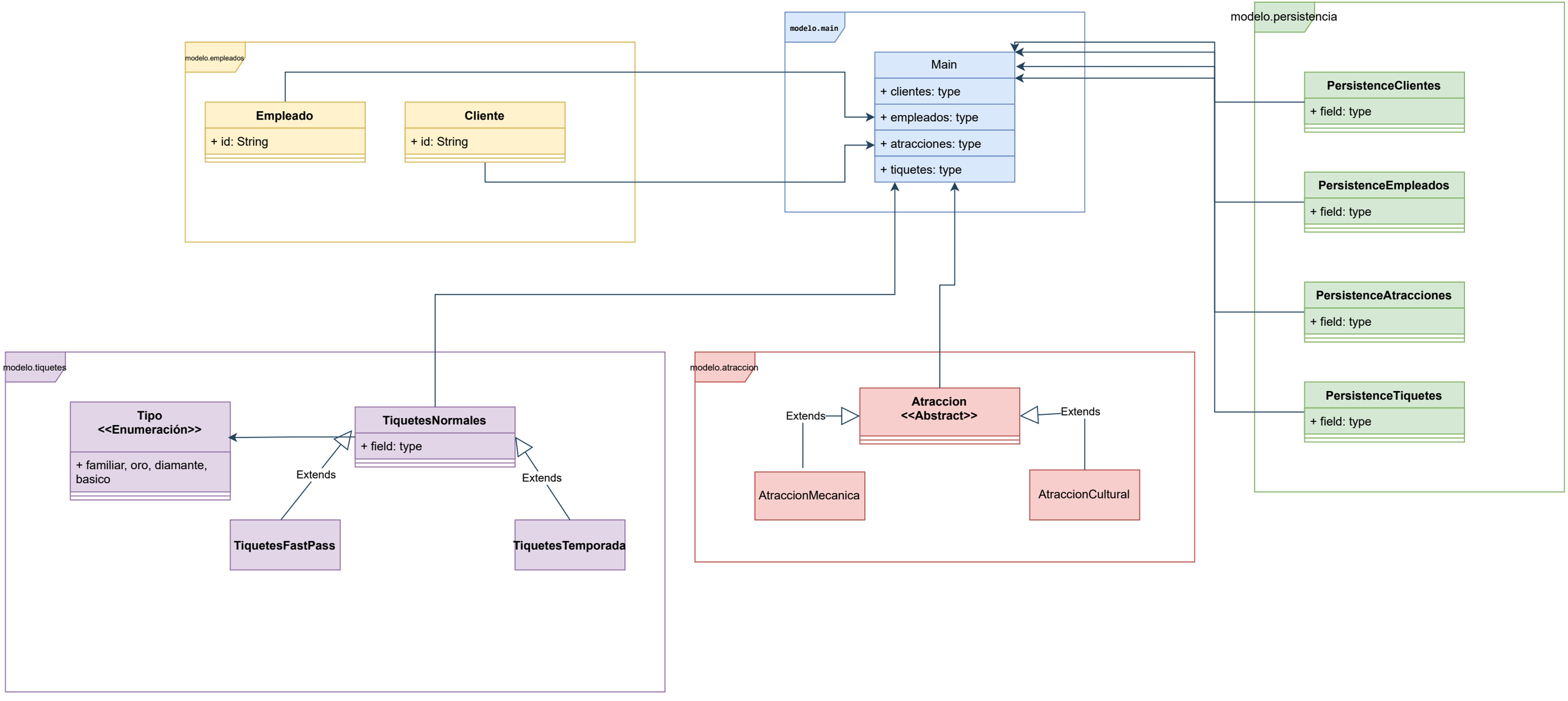

Empleado
-cargo: String
-historialTurnos: List<Turno>
- disponoble : Boolean
- turno : HashMap<String, Turno>
-certificaciones: List<Strings>
+mostrarInformacion(): void
+asignarTurno(turno: Turno): void
+registrarAsistencia(): void
1
bidireccional
1..2
Turno
-id:int
-horaInicio: Date
-horaFin: Date
-tipoTurno: String
-ubicacionTemporal: String
- puestoAsignado: String
+Turno(int id, String horaInicio, String horaFin, String tipoTurno, String ubicacionTemporal)
puestoAsignado
..1
<<Enumeracion>>
PuestoAsignado
- cajero
- cocinero
- serviciosGenerales
- operarioAtracciones
-operariosAtraccionesEsp
Extends
Extends
AtraccionMecanica
-limiteMinAltura:float
-limiteMaxAltura:float
-limiteMinPeso:float
-limiteMaxPeso: float
-nivelRiesgo: String
-contraindicaciones: String
+AtraccionMecanica(int id, String nombre, int capacidad, String ubicacion, String nivelExclusividad, float limiteMinAltura, float limiteMaxAltura, float limiteMinPeso, float limiteMaxPeso, String nivelRiesgo, String contraindicaciones)
+validarRestricciones(altura: float, peso:float, condicionesMedicas: String): boolean
+iniciarOperacion(): void
+detenerOperacion(): void
+actualizarParametrosSeguridad(nuevosParametros: Map): void
<<Abstract>>
Atraccion
-id:int
-nombre:String
-capacidad:int
-ubicacion:String
-nivelExclusividad: String
-estadoOperativo: boolean
-mantenimientoProgramado: Date
+registrar(): void
+actualizar(): void
+validarAcceso(parametros: Map): boolean
+programarMantenimiento(fecha: Date) void
+obtenerEstado(): boolean
AtraccionCultural
-restriccionEdad: int
-descripcionTematica: String
+AtraccionCultural(int id, String nombre, int capacidad, String ubicacion, String nivelExclusividad, int restriccionEdad)
+validarIngreso(edad: int): boolean
+iniciarEvento(): void
+finalizarEvento(): void
+actualizarInformacionCultural(nuevaInfo: Map) : void
<<Abstract>>
Usuario
-id:int
-nombre:String
-login:String
-password: String
-tipo: TipoUsuario
+Usuario(int id, String nomb String login, String passwor TipoUsuario tipo)
+autenticar(login: String, pas String)
+cambiarPassword(nuevaP String): void
+mostrarInformacion(): void
Administrador
+mostrarInformacion():void
+gestionarUsuarios(): void}
+obtenerUsuarioPorId(id: in Usuario
+obtenerUsuarioPorLogin(lo String): Usuario
Cliente
-historialCompras: List<Tiquete>
+mostrarInformacion(): void
+consultarHistorialCompras(): List<Tiquete>
+comprarTiquete(tiquete: Tiquete): void
EmpleadoServicio
- tipoServicio: String
- servicioGeneral : Boolean
- manejoCocina: Boolean
+ vender ():
+ generalFactura():
Empleados
1..*
Cafetería
- nombre: String
- ubicación: String
- cocinar(Producto plato): void
- vender(Producto producto): void
platos
..*
Producto
- nombre: String
- precio : Int
- getNombre():
- getPrecio(): void
productos
..*
Tienda
- nombre: String
- ubicación: String
- vender(Producto producto)
Taquilla
- tiquetesVendidos : Int
- tiquetesDisponibles: Int
- venderTiquete(): void
Lista de Tiquetes
1..**
EmpleadoAtracción
- capacidadRiesgo: String
- capacidadAtraccionesAltas: List(String)
- verificarTiquete():
- controlarAtraccion():
Atracciones
1..*
Lista de Tiquetes
1..**
TiquetesNormales
- id: String
- utilizado: Boolean
- accesoAtraciones: List (Atradccion)
- descuento: int
- comprar(): void
-guardarCompra(): void
- setDescuento(Usuario)
tipo
..1
<<Enumeración >>
Tipo
- familiar
- oro
-diamante
- basico
Extends
Extends
TiquetesFastPass
-fechaEntrada: Date
getTiquetesFastPass():void
setTiquetesFastPass(): void
TiquetesTemporada
- fechaInicial: Date
- fechaCaducidad: Date
getTiquetesTemporada():void
setTiquetesTemporada(): void
Cliente 1 .. *
Tiquetes 1 :: *
Atraccion 1.. *
Empleado 1.. *
<<Interface>>
IPersistencia
+guardarDatos(objeto: Object, nombreArchivo: String) : boolean
+cargarDatos(nombreArchivo: String, Class<T> tipo): T
+actualizarRegistro(objeto: Object, nombreArchivo: String): boolean
+eliminarRegistro(nombreArchivo: String): boolean
<<Abstract>>
CorePersistencia
+JSON = "JSON": String
#rutaAlmacenamiento: String
PersistenciaTiquetes
-ARCHIVO_TIQUETES : String
+guardarListaTiquetes(lista: List<Tiquete>) : boolean
+cargarListaTiquetes(): List<Tiquete>
PersistenciaAtracciones
-ARCHIVO_ATRACCIONES : String
+guardarListaAtracciones(lista: List<Tiquete>) : boolean
+cargarListaAtracciones(): List<Tiquete>
PersistenciaEmpleados
-ARCHIVO_EMPLEADOS : String
+guardarListaEmpleados(lista: List<Empleado>) : boolean
+cargarListaEmpleados(): List<Empleado>
PersistenciaUsuarios
-ARCHIVO_ATRACCIONES : String
+guardarListaUsuarios(lista: List<Usuarios>) : boolean
+cargarListaUsuarios(): List<Usuarios>
PersistenciaEspectaculos
-ARCHIVO_ESPECTACULOS : String
+guardarListaEspectaculos(lista: List<Espectaculo>) : boolean
+cargarListaEspectaculos(): List<Espectaculo>
builds
builds
builds
builds
Tiquete
Atraccion
Empleado
<<Abstract>>
Usuario
Espectaculo
Main
+ Clientes: List<Clientes>
+ empleados: List<Empleado>
+ atracciones: List<Atraccion>
+ tiquetes: LIst<TiquetesNormales>
+ persClientes: persistenceClientes
+ persEmpleados: persistenceEmpleados
+ persAtracciones: persistenceAtracciones
+ persTiquetes: persistenceTiquetes
+ scaner: scaner
+ main(string[]): void
+ agregarCliente() : void
+ agregarTrabajador(): void
+ agregarAtraccion(): void
+ asignarTrabajadorAtraccion(): void
+ venderTiquete(): void
+ guardarDatos(): void
+ cargarDatos(): void
+ mostrarInformacion(): void
persistenceTiquetes
persistenceAtracciones
persistenceEmpleados
persistenceClientes


modelo.empleados
Empleado
+ id: String
Cliente
+ id: String
modelo.main
Main
+ clientes: type
+ empleados: type
+ atracciones: type
+ tiquetes: type
modelo.persistencia
PersistenceClientes
+ field: type
PersistenceEmpleados
+ field: type
PersistenceAtracciones
+ field: type
PersistenceTiquetes
+ field: type
modelo.tiquetes
Tipo
<<Enumeración>>
+ familiar, oro, diamante, basico
TiquetesNormales
+ field: type
Extends
Extends
TiquetesFastPass
TiquetesTemporada
modelo.atraccion
Atraccion
<<Abstract>>
Extends
Extends
AtraccionMecanica
AtraccionCultural