

AplicacionGestionEventos
- gestor: GestorEventos<Evento> - vistaListaEventos: ListView<Evento>
+ start(escenarioPrincipal: Stage): void throws IOException, ClassNotFoundException + convertirAFecha(fechaStr: String): LocalDate - actualizarListaEventos(): void - limpiarCampos(campos: TextField...): void - mostrarAlerta(titulo: String, contenido: String): void - mostrarVentanaEstadisticas(estadisticas: List<Map<String, Object>): void + main(args: String[]): void

Consola
+ main(args: String[]): void

GestorEventos
- listaEventos: List<T> - ARCHIVO_BINARIO: String
+ GestorEventos() + cargarDesdeBinario(): void + actualizar(eventoActualizado: Evento): void + guardarEnBinario(): void + obtenerEstadisticas(fileName: String): List<Map<String, Object> + add(evento: Evento): boolean + remove(evento: Evento): boolean + getAll(): List<T> + setAll(items: List<T>): void + ordenarEventosPorFecha(): void + ordenarPorUbicacion(): void + ordenarPorPrecioAscendente(ascendente: boolean): void + ordenarPorPrecioDescendente(descendente: boolean): void + filtrarEventos(filtro: FiltroEvento): List<T> + iterator(): Iterator<T> + writeBinary(fileName: String, object: List<Evento>): boolean + readBinary(fileName: String): List<Evento> + writeJSON(fileName: String, object: List<?>): boolean + readJSON<R>(fileName: String, typeToken: TypeToken<R>): R + generarEstadisticasAsistencia(): boolean + mostrarEstadisticas(fileName: String): void + exportarEventosCSV(nombreDirectorio: String, nombreArchivo: String): boolean + mostrarContenidoCSV(nombreDirectorio: String, nombreArchivoCSV: String): boolean + buscarEventos(criterio: int, valor: Object): List<Evento> + mostrarTodosLosEventos(): void + modificarEvento(codigo: String, opcionModificar: int, nuevoValor: String): boolean + registrarEvento(codigo: String, titulo: String, fecha: LocalDate, capacidadMaxima: int, organizador: String, ubicacion: String, precio: int, asistentes: int, tipoEvento: int, tema: String, panelistas: List<String>, artistaPrincipal: String, generoMusical: String): boolean + eliminarEvento(codigo: String): void

«FunctionalInterface» FiltroEvento
+ test(evento: Evento): boolean

«interface» Repository<T>
+ add(item: T): boolean throws CapacidadExcedidaException + remove(item: T): boolean throws EventoNoEncontradoException + getAll(): List<T> + setAll(items: List<T>): void

«interface» Serializer<T>
+ writeBinary(fileName: String, object: T): boolean throws IOException + readBinary(fileName: String): T throws Exception + writeJSON(fileName: String, object: List<?>): boolean throws IOException + <R> readJSON(fileName: String, typeToken: TypeToken<R>): R throws IOException

PrecioComparator
- ascendente: boolean
+ PrecioComparator(ascendente: boolean) + compare(e1: Evento, e2: Evento): int

UbicacionComparator
+ compare(e1: Evento, e2: Evento): int

«abstract» Evento
- serialVersionUID: long - codigo: String - titulo: String - fecha: LocalDate - organizador: String - capacidadMaxima: int - tipoEvento: TipoEvento - ubicacion: String - precio: double - asistentes: int
+ Evento(codigo: String, titulo: String, fecha: LocalDate, organizador: String, capacidadMaxima: int, tipoEvento: TipoEvento, ubicacion: String, precio: double, asistentes: int) + getCodigo(): String + setCodigo(codigo: String): void + getTitulo(): String + setTitulo(titulo: String): void + getFecha(): LocalDate + setFecha(fecha: LocalDate): void + getOrganizador(): String + setOrganizador(organizador: String): void + getCapacidadMaxima(): int + setCapacidadMaxima(capacidadMaxima: int): void + getTipoEvento(): TipoEvento + setTipoEvento(tipoEvento: TipoEvento): void + getUbicacion(): String + setUbicacion(ubicacion: String): void + getPrecio(): double + setPrecio(precio: double): void + getAsistentes(): int + setAsistentes(asistentes: int): void + getCapacidadRestante(): int + getTipo(): TipoEvento + estaLleno(): boolean + mostrarDetalles(): void + compareTo(otroEvento: Evento): int + toString(): String

«enum» TipoEvento
+ CONCIERTO + CONFERENCIA

Conferencia
- tema: String - panelistas: List<String>
+ Conferencia(codigo: String, titulo: String, fecha: LocalDate, organizador: String, capacidadMaxima: int, ubicacion: String, precio: double, asistentes: int, tema: String, panelistas: List<String>) + getTema(): String + getPanelistas(): List<String> + setTema(tema: String): void + setPanelistas(panelistas: List<String>): void + cantidadPanelistas(): int + estaLleno(): boolean + toString(): String + mostrarDetalles(): void

Concierto
- artistaPrincipal: String - generoMusical: String
+ Concierto(codigo: String, titulo: String, fecha: LocalDate, organizador: String, capacidadMaxima: int, ubicacion: String, precio: double, asistentes: int, artistaPrincipal: String, generoMusical: String) + getArtistaPrincipal(): String + getGeneroMusical(): String + setArtistaPrincipal(artistaPrincipal: String): void + setGeneroMusical(generoMusical: String): void + esMusicaClasica(): boolean + estaLleno(): boolean + toString(): String + mostrarDetalles(): void