

	Juego	Mapa	Entidad
+	+SALVAR.txt +CARGAR.txt +DEFECHAS.txt +GOLAZOS.txt Controlador_Vistas: ControladorVistas +mapa, nivel, actual: Mapa +nivel, actual: Nivel +alabara_sprites: SpritesFactory +alabara_entidades: EntidadesFactory +vistas: Vistas +atempo_resistente: int +ascore: Scoreo +Jugador()	+mi_juego: Juego +mario: Mario +entidades: List<Entidad> +Magoo(pango: Juego) +get_posicion_jugador(): int +get_entidad(n): List<Entidad> +get_mario(): Mario +agregar_entidad(entidad: Entidad): void +agregar_mario(mario: Mario): void +resetear_mapa(): void +mover_jugador(direccion: int): void +nivel_completo(): boolean +chequear_colision(): void	+mi_mapa: Mapa +posicion_en_y: double +posicion_en_x: double +vida: bool +sprite: Sprite +Entidad(): double, y: double, sprite: Sprite) +get_posicion_en_x(): float +get_posicion_en_y(): float +get_vida_bool(): int +get_imagen_representativa(): Sprite +set_posicion_en_x(): void +set_posicion_en_y(): void +set_puntaje_a_sumar(): void +consumir(material: Material): void +es_consumible(): boolean +es_me_gustada(): boolean +matr_ai_hay_colision(mario: Mario): boolean +matr_ai_hay_colision(emergo: Emergo): boolean +matr_ai_hay_colision(bolaFuego: BolaDeFuego): boolean +desarmado(): void +es_desarmable(): boolean +registrador_observer(Observer Observer): void

```

classDiagram
    class Sonido {
        -estado: SonidoState
        +Sonido(estado: SonidoState)
        +prender_sonido() void
        +apagar_sonido() void
        +cambiar_estado(nuevo_estado: SonidoState) void
    }
    class OfiMusicState {
        -sound: Sound
    }
    class OnMusicState {
        -sound: Sound
        +OnMusicState(sound: Sound)
        +prender_sonido() void
        +apagar_sonido() void
    }
    class Builder {
        +nivel(Builder: Builder)
        +get_posicion_inicial_jugador_x() double
        +get_posicion_inicial_jugador_y() double
        +get_tempo_restante() int
        +get_numero_de_nivel() int
    }
    class Ranking {
        -ranking: Map<String, Integer>
        +Ranking()
        +agregar_jugador(String nombre, int puntos) void
        +get_sort_ranking() Map<String, Integer>
    }
    class ObservadorEntidades {
        -entidad: ObservadorEntidadLogica
        +actualizar() void
    }
    class ObservadorJugador {
        -entidad: ObservadorEntidadJugador
        +actualizar_nombre() void
        +actualizar_posicion_tamano() void
    }
    class ObservadorGráfico {
        -entidad: ObservadorEntidadGráfico
        +actualizar() void
        +actualizar_imagen() void
        +actualizar_posicion_tamano() void
    }
    class EntidadLogica {
        +get_x() double
        +get_y() double
        +get_estado() State
    }
    class EntidadJugador {
        +get_puntaje_actual() int
        +get_puntaje_acumulado() int
        +get_monedas() int
    }
    class EntidadEstado {
        +get_estado() State
    }
    class EntidadNivel {
        +cargar_nivel(archivo: InputStream, generador: EntidadesFactory, m: Mapa, Mapa: Nivel, jugador: ObservadorEntidadLogica, String: String, valor: String, valor: generador, EntidadesFactory: generador, Mapa: Mapa) void
    }
    class EntidadMapa {
        +get_mapa() Mapa
    }
    Sonido --> OfiMusicState
    Sonido --> OnMusicState
    OfiMusicState --> OnMusicState
    Builder --> Ranking
    ObservadorEntidades --> ObservadorJugador
    ObservadorEntidades --> ObservadorGráfico
    ObservadorJugador --> ObservadorGráfico
    EntidadLogica --> EntidadJugador
    EntidadLogica --> EntidadEstado
    EntidadLogica --> EntidadMapa
    EntidadJugador --> EntidadEstado
    EntidadJugador --> EntidadMapa
    EntidadEstado --> EntidadMapa
    EntidadNivel --> EntidadMapa
    
```

Sonido

- estado: SonidoState
- +Sonido(estado: SonidoState)
- +prender_sonido() void
- +apagar_sonido() void
- +cambiar_estado(nuevo_estado: SonidoState) void

OfiMusicState

- sound: Sound

OnMusicState

- sound: Sound
- +OnMusicState(sound: Sound)
- +prender_sonido() void
- +apagar_sonido() void

Builder

- +nivel(Builder: Builder)
- +get_posicion_inicial_jugador_x() double
- +get_posicion_inicial_jugador_y() double
- +get_tempo_restante() int
- +get_numero_de_nivel() int

Ranking

- ranking: Map<String, Integer>
- +Ranking()
- +agregar_jugador(String nombre, int puntos) void
- +get_sort_ranking() Map<String, Integer>

ObservadorEntidades

- entidad: ObservadorEntidadLogica
- +actualizar() void

ObservadorJugador

- entidad: ObservadorEntidadJugador
- +actualizar_nombre() void
- +actualizar_posicion_tamano() void

ObservadorGráfico

- entidad: ObservadorEntidadGráfico
- +actualizar() void
- +actualizar_imagen() void
- +actualizar_posicion_tamano() void

EntidadLogica

- +get_x() double
- +get_y() double
- +get_estado() State

EntidadJugador

- +get_puntaje_actual() int
- +get_puntaje_acumulado() int
- +get_monedas() int

EntidadEstado

- +get_estado() State

EntidadNivel

- +cargar_nivel(archivo: InputStream, generador: EntidadesFactory, m: Mapa, Mapa: Nivel, jugador: ObservadorEntidadLogica, String: String, valor: String, valor: generador, EntidadesFactory: generador, Mapa: Mapa) void

EntidadMapa

- +get_mapa() Mapa

[illegible]