

«interface» ControladorGrafico
+ registrarEntidad(p:PlataformaMovil): ObserverPlataformaMovil + registrarEntidadEstatica(e: EntidadEstatica): void + registrarEntidad(e: EntidadDinamica): void + registrarEntidad(j: Jugador): void + setModoJuego(modo:String): void

«interface» ControladorDeVista
+ cambiarInicioJuego(): void + cambiarPantallaRanking(): void + cambiarPantallaModo(): void + cambiarPantallaGameOver(): void + cambiarPantallaTransicion(): void + cambiarPantallaPrincipal(): void + cambiarPantallaFinalJuego(): void

<table border="1"> <thead> <tr> <th>ConstantesDificultad</th></tr> </thead> <tbody> <tr> <td> + POTENCIA_DISPARO_AUMENTADA: int=2 + POTENCIA_DISPARO_NORMAL: int=1 + VELOCIDAD_DISPARO: int=15 + ALTURA_MAX_SALTO: int=72 + SALUD_DEMONIO_ROJO: int=2 + SALUD_TROLL_AMARILLO: int=1 + SALUD_RANA_DE_FUEGO: int=1 + SALUD_MOGHERA: int=3 + SALUD_KAMAKICHI: int=3 + VELOCIDAD_DEMONIO_ROJO: int=1 + VELOCIDAD_TROLL_AMARILLO: int=2 + VELOCIDAD_RANA_DE_FUEGO: int=1 + VELOCIDAD_MOGHERA: int=8 + VELOCIDAD_KAMAKICHI: int=8 + VELOCIDAD_CALABAZA: int=1 + VELOCIDAD_FANTASMA: int=1 + VELOCIDAD_BOLADENIEVE: int=2 + VELOCIDAD_MAXIMA_BOLADENIEVE: int=15 + VELOCIDAD_BOLADEFUEGO: int=6 + VELOCIDAD_ENTIDAD_DINAMICA_X: int=8 + VELOCIDAD_ENTIDAD_DIANMICA_Y: int=4 + VELOCIDAD_PLATAFORMA_MOVIL: int=1 + VELOCIDAD_JUGADOR: int=5 + VELOCIDAD_SALTO: int=4 + TOPE TIEMPO CONTRARRELOJ: int=120 + CONDICION_SUPERVIVENCIA: int=5000; + VIDAS_INICIO: int=3 + EFECTO_BOTELLA_AZUL: int=10 </td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>ConstantesVistas</th></tr> </thead> <tbody> <tr> <td> + VENTANA_ALTO: int=660 + VENTANA_ANCHO: int=832 + PANEL_ALTO: int=608 + PANEL_ANCHO: int=832 + PANEL_JUEGO_ANCHO: int=768 + PANEL_JUEGO_ALTO: int=600 </td></tr> </tbody> </table>	ConstantesDificultad	+ POTENCIA_DISPARO_AUMENTADA: int=2 + POTENCIA_DISPARO_NORMAL: int=1 + VELOCIDAD_DISPARO: int=15 + ALTURA_MAX_SALTO: int=72 + SALUD_DEMONIO_ROJO: int=2 + SALUD_TROLL_AMARILLO: int=1 + SALUD_RANA_DE_FUEGO: int=1 + SALUD_MOGHERA: int=3 + SALUD_KAMAKICHI: int=3 + VELOCIDAD_DEMONIO_ROJO: int=1 + VELOCIDAD_TROLL_AMARILLO: int=2 + VELOCIDAD_RANA_DE_FUEGO: int=1 + VELOCIDAD_MOGHERA: int=8 + VELOCIDAD_KAMAKICHI: int=8 + VELOCIDAD_CALABAZA: int=1 + VELOCIDAD_FANTASMA: int=1 + VELOCIDAD_BOLADENIEVE: int=2 + VELOCIDAD_MAXIMA_BOLADENIEVE: int=15 + VELOCIDAD_BOLADEFUEGO: int=6 + VELOCIDAD_ENTIDAD_DINAMICA_X: int=8 + VELOCIDAD_ENTIDAD_DIANMICA_Y: int=4 + VELOCIDAD_PLATAFORMA_MOVIL: int=1 + VELOCIDAD_JUGADOR: int=5 + VELOCIDAD_SALTO: int=4 + TOPE TIEMPO CONTRARRELOJ: int=120 + CONDICION_SUPERVIVENCIA: int=5000; + VIDAS_INICIO: int=3 + EFECTO_BOTELLA_AZUL: int=10	ConstantesVistas	+ VENTANA_ALTO: int=660 + VENTANA_ANCHO: int=832 + PANEL_ALTO: int=608 + PANEL_ANCHO: int=832 + PANEL_JUEGO_ANCHO: int=768 + PANEL_JUEGO_ALTO: int=600	<table border="1"> <thead> <tr> <th>ConstantesNiveles</th></tr> </thead> <tbody> <tr> <td> + NIVEL1: String="SnowBros/src/Niveles/nivel1.txt" + NIVEL2: String="SnowBros/src/Niveles/nivel2.txt" + NIVEL3: String="SnowBros/src/Niveles/nivel3.txt" + NIVEL4: String="SnowBros/src/Niveles/nivel4.txt" + OLEADA1: String="SnowBros/src/Niveles/oleada1.txt" + OLEADA2: String= "SnowBros/src/Niveles/oelada2.txt" </td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>ConstantesTiempos</th></tr> </thead> <tbody> <tr> <td> + TIEMPO_CALABAZA: int=60 + TIEMPO_BOLA_DE_FUEGO: int=10 + TIEMPO_BOLA_DE_NIEVE: int=10 + TIEMPO_DISPARO_JUGADOR: int=12 + TIEMPO_VIDA_POWER_UP: int=10 + TIEMPO_EFECTO_POWER_UP: int=7 + COOLDOWN_DISPARO: long=300 + TIEMPO_LAPSO_INVENCIBILIDAD: long=100 + TIEMPO_CREACION_FANTASMA: int=15 + TIEMPO_CONGELADO_ENEMIGOS: int=10 + TIEMPO_CAI DA_DISPARO: int=800 </td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>ConstantesEntidades</th></tr> </thead> <tbody> <tr> <td> + AIRE: int=20000 + SUELO: int=20001 + ARRIBA: int=15000 + ARRIBA_IZQUIERDA: int=15001 + ARRIBA_DERECHA: int=15002 + IZQUIERDA: int=15003 + DERECHA: int=15004 + ABAJO: int=15005 + QUIETO: int=0 + MARGEN_HIT_BOX: int=1 + X_CALABAZA: int=320 + Y_CALABAZA: int=120 + X_MOGHERA: int=730 </td></tr> </tbody> </table>	ConstantesNiveles	+ NIVEL1: String="SnowBros/src/Niveles/nivel1.txt" + NIVEL2: String="SnowBros/src/Niveles/nivel2.txt" + NIVEL3: String="SnowBros/src/Niveles/nivel3.txt" + NIVEL4: String="SnowBros/src/Niveles/nivel4.txt" + OLEADA1: String="SnowBros/src/Niveles/oleada1.txt" + OLEADA2: String= "SnowBros/src/Niveles/oelada2.txt"	ConstantesTiempos	+ TIEMPO_CALABAZA: int=60 + TIEMPO_BOLA_DE_FUEGO: int=10 + TIEMPO_BOLA_DE_NIEVE: int=10 + TIEMPO_DISPARO_JUGADOR: int=12 + TIEMPO_VIDA_POWER_UP: int=10 + TIEMPO_EFECTO_POWER_UP: int=7 + COOLDOWN_DISPARO: long=300 + TIEMPO_LAPSO_INVENCIBILIDAD: long=100 + TIEMPO_CREACION_FANTASMA: int=15 + TIEMPO_CONGELADO_ENEMIGOS: int=10 + TIEMPO_CAI DA_DISPARO: int=800	ConstantesEntidades	+ AIRE: int=20000 + SUELO: int=20001 + ARRIBA: int=15000 + ARRIBA_IZQUIERDA: int=15001 + ARRIBA_DERECHA: int=15002 + IZQUIERDA: int=15003 + DERECHA: int=15004 + ABAJO: int=15005 + QUIETO: int=0 + MARGEN_HIT_BOX: int=1 + X_CALABAZA: int=320 + Y_CALABAZA: int=120 + X_MOGHERA: int=730
ConstantesDificultad											
+ POTENCIA_DISPARO_AUMENTADA: int=2 + POTENCIA_DISPARO_NORMAL: int=1 + VELOCIDAD_DISPARO: int=15 + ALTURA_MAX_SALTO: int=72 + SALUD_DEMONIO_ROJO: int=2 + SALUD_TROLL_AMARILLO: int=1 + SALUD_RANA_DE_FUEGO: int=1 + SALUD_MOGHERA: int=3 + SALUD_KAMAKICHI: int=3 + VELOCIDAD_DEMONIO_ROJO: int=1 + VELOCIDAD_TROLL_AMARILLO: int=2 + VELOCIDAD_RANA_DE_FUEGO: int=1 + VELOCIDAD_MOGHERA: int=8 + VELOCIDAD_KAMAKICHI: int=8 + VELOCIDAD_CALABAZA: int=1 + VELOCIDAD_FANTASMA: int=1 + VELOCIDAD_BOLADENIEVE: int=2 + VELOCIDAD_MAXIMA_BOLADENIEVE: int=15 + VELOCIDAD_BOLADEFUEGO: int=6 + VELOCIDAD_ENTIDAD_DINAMICA_X: int=8 + VELOCIDAD_ENTIDAD_DIANMICA_Y: int=4 + VELOCIDAD_PLATAFORMA_MOVIL: int=1 + VELOCIDAD_JUGADOR: int=5 + VELOCIDAD_SALTO: int=4 + TOPE TIEMPO CONTRARRELOJ: int=120 + CONDICION_SUPERVIVENCIA: int=5000; + VIDAS_INICIO: int=3 + EFECTO_BOTELLA_AZUL: int=10											
ConstantesVistas											
+ VENTANA_ALTO: int=660 + VENTANA_ANCHO: int=832 + PANEL_ALTO: int=608 + PANEL_ANCHO: int=832 + PANEL_JUEGO_ANCHO: int=768 + PANEL_JUEGO_ALTO: int=600											
ConstantesNiveles											
+ NIVEL1: String="SnowBros/src/Niveles/nivel1.txt" + NIVEL2: String="SnowBros/src/Niveles/nivel2.txt" + NIVEL3: String="SnowBros/src/Niveles/nivel3.txt" + NIVEL4: String="SnowBros/src/Niveles/nivel4.txt" + OLEADA1: String="SnowBros/src/Niveles/oleada1.txt" + OLEADA2: String= "SnowBros/src/Niveles/oelada2.txt"											
ConstantesTiempos											
+ TIEMPO_CALABAZA: int=60 + TIEMPO_BOLA_DE_FUEGO: int=10 + TIEMPO_BOLA_DE_NIEVE: int=10 + TIEMPO_DISPARO_JUGADOR: int=12 + TIEMPO_VIDA_POWER_UP: int=10 + TIEMPO_EFECTO_POWER_UP: int=7 + COOLDOWN_DISPARO: long=300 + TIEMPO_LAPSO_INVENCIBILIDAD: long=100 + TIEMPO_CREACION_FANTASMA: int=15 + TIEMPO_CONGELADO_ENEMIGOS: int=10 + TIEMPO_CAI DA_DISPARO: int=800											
ConstantesEntidades											
+ AIRE: int=20000 + SUELO: int=20001 + ARRIBA: int=15000 + ARRIBA_IZQUIERDA: int=15001 + ARRIBA_DERECHA: int=15002 + IZQUIERDA: int=15003 + DERECHA: int=15004 + ABAJO: int=15005 + QUIETO: int=0 + MARGEN_HIT_BOX: int=1 + X_CALABAZA: int=320 + Y_CALABAZA: int=120 + X_MOGHERA: int=730											

EstadosDePatron

- + SALTAR: int=16002
- + MOVERIZQUIERDA: int=16003
- + MOVERDERECHA: int=16004
- + ESPERAR: int=16005
- + MOVERARRIBA: int=16006
- + MOVERABAJO: int=16007

```
# nivel: Nivel
# puntajeJuga
# tiempo: Temp
# reiniciarJuga
# jugador.Jug
# ranking: Rank
# controladorG
# fabricaSprite
# fabricaEntid
# parserActual
- tiempoAntes/
- tiempoAntes/
# hiloColision:
# hiloGlobal: T
# juegoEnEjec
# nombreJuga
# modoDeJueg
- ultimoDispar
# dominio: Stri
# gestor: Gest
# jefeCreado: I
- oleadas: int
- habilitarCond
```

Ranking
topClasico: Map<String, Integer>
topSupervivencia: Map<String, Integer>
topContraTiempo: Map<String, Integer>
+ setJugador(nombre: String, puntaje: int, modo: String): void
- actualizarRanking(mapa: LinkedHashMap<String, Integer>, nombre: String, puntaje: int): void

Juego

```
dor: int
porizador
ador: boolean
ador
king
Grafica: GUI
os: FabricaSprites
ades: FabricaEntidades
l: ParserNivel
AparicionCalabaza: Temporizador
AparicionJefeContrarreloj: Temporizador
Thread
Thread
cucion: boolean
idor: String
go: String
ro: long
ing
orEntidad
boolean

dicionClasico: boolean
forGrafico(miControlador: GUI): void
{
    void
    o(): void
    l(): void
    i: void
    l(): void
    dicionDelModo(): boolean
    intaje(puntajeASumar: int): void
    : boolean): void
    dor(): GUI
    al(): Nivel
    int
    ): Jugador
    al(): Nivel
    ctual(): int
    Temporizador
    ): Ranking
    ): Jugador
ugador(nombre: String): void
dominio: String): void
el: Nivel): void
ntidades(): FabricaEntidades
evers(): void
erverJugadro(): void
serverEntidad(entidad: EntidadDinamica):

eversEntidades(entidades: List<?
d>): void
eversPlataformas(entidades: List<?
d>): void
Jp(p: PowerUp): void
las(): void
xtra(): void
```

GestorEntidad

```
- juego: Juego
- nivel: Nivel

+ getFabrica(): FabricaEntidades
+ agregarPowerUp(p: PowerUp): void
+ actualizarPuntaje(i: int): void
+ crearBolaDeNieve(posX: int, posY: int,
puntaje: int): void
+ crearBolaDeFuego(posX: int, posY: int,
puntaje: int): BolaDeFuego
+ crearFantasma(posX: int, posY): void
+ setNivel(n: Nivel): void
```

Launcher

```
+ main()
```

Sonido

<<Interface>> Observer
+ actualizar(): void + actualizarBolaDeNieve(b: boolean):void + borrarObserver(): void

ObserverEntidad
entidadObservada: EntidadDinamica # plataformaObservada: PlataformaMovil - imagen: JLabel
+ actualizar(): void # actualizarImagen(): void # actualizarPosicionTamaño(): void # actualizarPosicionTamañoPlataforma(): void + actualizarBolaDeNieve(activo: boolean): void + borrarObserver(): void + getImagen(): JLabel

ObserverPlataformaMovil
+ actualizar(): void # actualizarImagen(): void

PanelPantallaTransicion
imagenFondo: JLabel # textoNivel: JLabel # miControladorGrafico: GUI
+ mostrarNivel(): void - agregarImagenFondo(): void - agregarTextoNivel(): void - iniciarTemporizadorTransicion(): void - refrescar(): void

PanelPantallaGameOver
imagenFondo: JLabel # botonSalir: JButton # botonRanking: JButton # miControladorGrafico: GUI
+ agregarImagenFondo(): void - agregarBotonSalir() - agregarBotonRanking(): void - transparentarBoton(boton: JButton): void - registrarOyenteBotonSalir(): void - registrarOyenteBotonRanking(): void

```
+ setModoJuego(modo: String): void
+ setDominio(dominio: String): void
```

```
+ cambiarPantallaJuego(): void
+ cambiarPantallaDominio(): void
```

GUI
<pre># ventana: JFrame # panelPantallaPrincipal: PanelPantallaPrincipal # panelPantallaJuego: PanelPantallaJuego # panelPantallaModo: PanelPantallaModo # panelPantallaDominio: PanelPantallaDominio # panelPantallaTransicion: PanelPantallaTransicion # panelPantallaFinDelJuego: PanelPantallaFinDelJuego # panelPantallaRanking: PanelPantallaRanking # panelPantallaGameOver: PanelPantallaGameOver # miJuego: Juego - movDerecha: boolean - movIzquierda: boolean - disparando: boolean - empujando: boolean - juegoIniciado: boolean # dominio: String # modoDeJuego: String - configurarVentana(): void # registrarOyenteVentana(): void - addWindowKeyListener(): void + moverJugadorSegunInput(): void - cambiarPantalla(panel: JPanel): void + cambiarPantallaTransicion(): void + cambiarPantallaGameOver(): void + cambiarPantallaFinDelJuego(): void + cambiarPantallaPrincipal(): void + cambiarPantallaModo(): void + cambiarPantallaDominio(): void + cambiarPantallaRanking(): void + cambiarInicioJuego(): void + setDominio(dominio: String): void + setModoJuego(modo: String): void + setNombre(modo: String): void + registrarEntidad(entidad: EntidadDinamica): void + registrarEntidadEstatica(e: EntidadEstatica): void + registrarEntidad(jugador: Jugador): void + registrarEntidad(entidad: PlataformaMovil): ObserverPlataformaMovil + setJugadorVisibleGraficamente(): void + setJugadorInvisibleGraficamente(): void + setMoverIzq(b: boolean): void + setMoverDer(b: boolean): void + setEmpujando(b: boolean): void + refrescar(): void + getJuego(): Juego + getPantalla(): PanelPantallaJuego + getVentana(): JFrame</pre>

PanelPantallaPrincipal
<pre>+ serialVersionUID: long = 1L # imagenFondo: JLabel # botonIniciar: JButton # botonRanking: JButton # miControladorGrafico: GUI # agregarImagenFondo(): void # agregarBotonIniciar(): void - agregarBotonRanking(): void - registrarOyenteBotonIniciar(): void - registrarOyenteBotonRanking(): void - transparentarBoton(boton: JButton): void</pre>

PanelPantallaDominio
<pre># imagenFondo: JLabel # botonOriginal: JButton # botonAlternativo: JButton # miControladorGrafico: GUI + agregarImagenFondo(): void + agregarBotonOriginal(): void + agregarBotonAlternativo(): void + registrarOyenteBotonOriginal(): void + registrarOyenteBotonAlternativo(): void + transparentarBoton(boton: JButton): void</pre>

PanelPantallaModo
<pre># imagenFondo: JLabel # botonClasico: JButton # botonContrarreloj: JButton # botonSupervivencia: JButton # miControladorGrafico: GUI + agregarImagenFondo(): void + agregarBotonClasico(): void + agregarBotonSupervivencia(): void + agregarBotonContrarreloj(): void + registrarOyenteBotonClasico(): void + registrarOyenteBotonContrarreloj(): void + registrarOyenteBotonSupervivencia(): void + transparentarBoton(boton: JButton): void</pre>

+ Y_MOGHERA: int=70
+ X_KAMAKICHI: int=
+ Y_KAMAKICHI: int=

**<<Interface>>
InterfazEliminar**

+ eliminarPlataforma(p: Plataforma): void
+ eliminarPared(p: Pared): void
+ eliminarJugador(j: Jugador): void
+ eliminarEnemigo(e: Enemigo): void
+ eliminarPowerUp(p: PowerUp): void
+ eliminarObstaculo(o: Obstaculo): void
+ eliminarBolaDeFuego(b: BolaDeFuego): v
+ eliminarBolaDeNieve(b: BolaDeNieve): vo
+ eliminarDisparo(d: DisparoJugador): void
+ eliminarSilueta(s: Silueta): void
+ eliminarFantasma(f: Fantasma): void
+ eliminarCalabaza(): void
+ eliminarPMovil(p: PlataformaMovil): void
+ eliminarResbaladizo(s: SueloResbaladizo)

**<<Interface>>
LocalizadorJugador**

+ getJugadorX(): void

**<<Interface>>
Visitante**

+ aceptar():

PanelPantallaJuego

panelPanelPantallaJuego: JPanel
panelInformacion: JPanel
imagenFondoJuego: JLabel
labelPuntaje: JLabel
labelNivel: JLabel
labelTiempo: JLabel
panelVidas: JLabel
miControladorGrafico: ControladorGrafico
agregarPanelInformacion(): void
agregarPanelJuego(): void
+ mostrarPlataformas(): void
+ mostrarObstaculos() : void
+ mostrarEntidades(entidades: List<T>, ajustarZOrder: boolean): void

PanelPantallaFinDelJuego

imagenFondo: JLabel
botonSalir: JButton
botonNuevaPartida: JButton
botonRanking: JButton
miControladorGrafico: GUI
+ agregarImagenFondo(): void
+ agregarBotonSalir(): void
+ agregarBotonRanking(): void
+ agregarBotonNuevaPartida(): void
+ registrarOyenteBotonSalir(): void
+ registrarOyenteBotonNuevaPartida(): void
+ registrarOyenteBotonRanking(): void
+ transparentarBoton(botón: JButton): void

PanelPantallaRank

imagenFondo: JLabel
botonSalir: JButton
miControladorGrafico: GUI
+ agregarImagenFondo(): void
+ agregarBotonSalir(): void
+ registrarOyenteBotonSalir(): void
+ transparentarBoton(botón: JButton): void
+ agregarRankings(): void
+ agregarColumnaRanking(título: String, LinkedHashMap<String, Integer>, x: filaInicialY: int, separacion: int): void
+ actualizaRanking(ranking: Ranking): void

```
+ getTopClasico(): LinkedHashMap<String, Integer>
+ getTopSupervivencia(): LinkedHashMap<String, Integer>
+ getTopContrarreloj(): LinkedHashMap<String, Integer>
- ordenar(mapeo: LinkedHashMap<String, Integer>): void
```

<<Interface>>
Visitador

```
+ visitar(e:Enemigo): void
+ visitar(p:Plataforma): void
+ visitar(e: PlataformaMovil): void
+ visitar(e: PlataformaQuebradiza): void
+ visitar(b: BotellaAzul): void
+ visitar(b: BotellaRoja): void
+ visitar(b: BotellaVerde): void
+ visitar(f: Fruta): void
+ visitar(v: VidaExtra): void
+ visitar(e: Pared): void
+ visitar(e: ParedDestructible): void
+ visitar(e: Escalera): void
+ visitar(e:SueloResbaladizo): void
+ visitar(e:TrampaPinchos): void
+ visitar(e: Jugador): void
```

Nivel

```
# sprite: Sprite
# jugador: Jugador
# numeroNivel: int
# silueta: Silueta
# listaEnemigos: LinkedList<Enemigo>
# listaPowerUps: LinkedList<PowerUp>
# listaObstaculos: LinkedList<Obstaculo>
# listaPlataformas: LinkedList<Plataformas>
# listaDisparoJugador: LinkedList<DisparoJugador>
# listaBolaDeNieve: LinkedList<BolaDeNieve>
# listaBolaDeFuego: LinkedList<BolaDeFuego>
# listaParedes: LinkedList<Pared>
# listaPMovil: LinkedList<PlataformaMovil>
# calabaza: Calabaza
# listaFantasmas: LinkedList<Fantasma>
# listaResbaladizos: LinkedList<SueloResbaladizo>
```

```
+ setNivel(nivel: int): void
+ getEnemigos(): LinkedList<Enemigo>
+ getPowerUps(): LinkedList<PowerUp>
+ getObstaculos(): LinkedList<Obstaculo>
+ getParedes(): LinkedList<Pared>
+ getPlataformas(): LinkedList<Plataformas>
+ getPlataformasMovil(): LinkedList<PlataformaMovil>
+ getDisparos(): LinkedList<DisparoJugador>
+ getBolasDeNieve(): LinkedList<BolaDeNieve>
+ getBolasDeFuego(): LinkedList<BolaDeFuego>
+ getFantasmas(): LinkedList<Fantasma>
+ getCalabaza(): Calabaza
+ getResbaladizos(): LikedList<SueloResbaladizo>
+ getNivelActual(): int
+ eliminarPlataforma(p: Plataforma): void
+ eliminarJugador(j: Jugador): void
+ eliminarEnemigo(e: Enemigo): void
+ eliminarPowerUp(p: PowerUp): void
+ eliminarObstaculo(o: Obstaculo): void
+ eliminarEntidadesEstaticas(): void
+ eliminarPared(p: Pared): void
+ eliminarSilueta(s: Silueta): void
+ eliminarBolaDeFuego(b: BolaDeFuego): void
+ eliminarDisparo(b: DisparoJugador): void
+ eliminarBolaDeNieve(b: BolaDeNieve): void
+ eliminarFantasma(f: Fantasma): void
+ eliminarCalabaza(): void
+ eliminarPMovil(p: PlataformaMovil): void
+ eliminarResbaladizo(s: SueloResbaladizo): void
+ agregarJugador(j: Jugador): void
+ agregarPlataforma(p: Plataforma): void
+ agregarPlataformaMovil(p: PlataformaMovil): void
+ agregarObstaculo(o: Obstaculo): void
+ agregarPared(p: Pared): void
+ agregarPowerUp(p: PowerUp): void
+ agregarEnemigo(e: Enemigo): void
+ agregarDisparo(b: DisparoJugador): void
+ agregarBolaDeNieve(b: BolaDeNieve): void
+ agregarBolaDeFuego(b: BolaDeFuego): void
+ agregarFantasma(f: Fantasma): void
+ agregarCalabaza(c: Calabaza): void
+ agregarResbaladizo(s: SueloResbaladizo): void
+ efectoBotellaVerde(): void
+ getJuego(): Juego
+ getJugadorX(): int
+ getJugador(): Jugador
+ getSilueta(): Silueta
+ notificarObservers(): void
```

```
+ eliminarPowerUp()
+ eliminarEntidad()
void
+ eliminarPlataforma(PlataformaQuebradiza)
+ getModoDeJuego()
+ setModoDeJuego()
+ crearCalabaza()
+ crearKamakiri()
+ crearMogher()
+ crearBolaDeFuego()
+ puntajeEmpujarBola()
+ crearBolaDeFuego()
+ crearFantasma()
+ shootJugador()
+ empujarBola()
+ invencibilidad()
+ estaCerca(entidad)
entidadEvaluar()
void
```

void
void

<<Interface>>
Juego

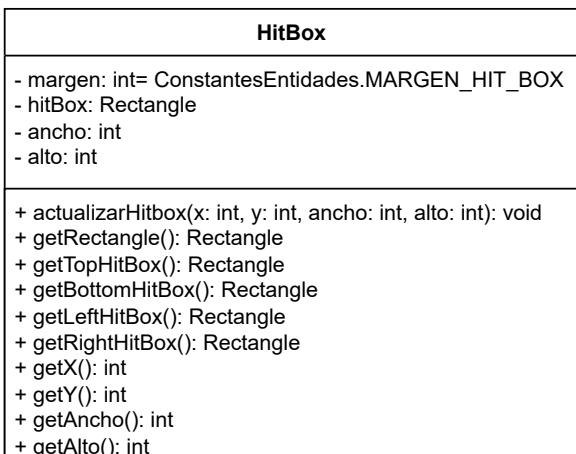
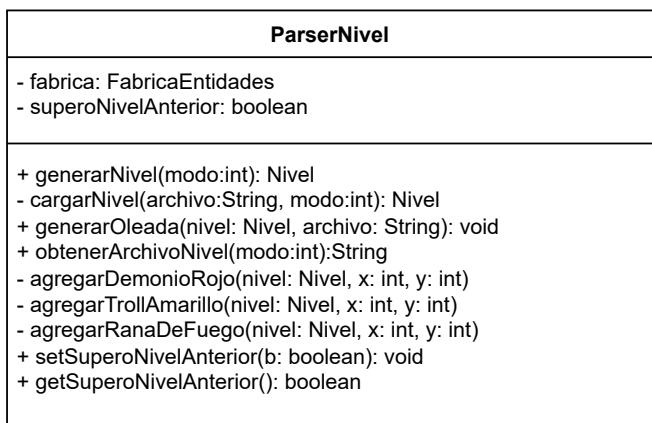
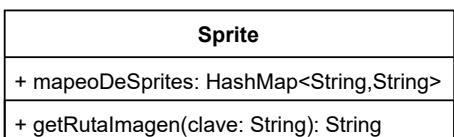
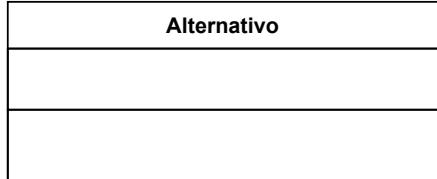
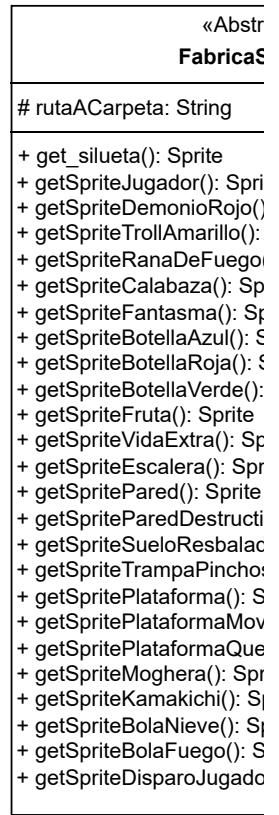
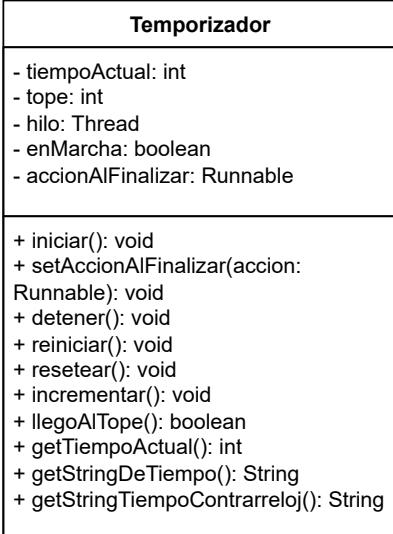
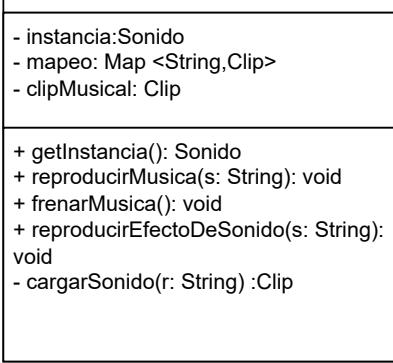
void
Juego, mapeo:
int, tituloY: int,
void

void

```

erUp(p: PowerUp): void
dadGraficamente(d: EntidadDinamica):
    formaQuebradiza(p:
        bradiza): void
    Juego(): String
    luego(modo: String): void
    za(): void
    ichi(): void
    ra(): void
    Nievez(posX: int, posY: int,
        do: int): void
    Fueguez(posX: int, posY: int, direccion: int):
        ma(posX: int, posY: int): void
        or(): void
        DeNieve(): void
        d(): void
        ntidadPrincipal: Entidad,
        da: Entidad, umbralDeRango: double);

```



```

`act>
Sprites

ite
): Sprite
Sprite
(): Sprite
rite
rite
Sprite
Sprite
: Sprite

rite
rite
:
ible(): Sprite
dizo(): Sprite
s(): Sprite
Sprite
il(): Sprite
ebradiza(): Sprite
rite
rite
Sprite
or(): Sprite

```

FabricaEntidades
<ul style="list-style-type: none"> - fabrica: FabricaSprite - gestor : GestorEntidad <ul style="list-style-type: none"> + asignarGestor(e: Enemigo): void + getSilueta(): Silueta + getJugador(x: int, y: int): Jugador + getDisparoJugador(x: int, y: int): DisparoJugador + getDemonioRojo(x: int, y: int): DemonioRojo + getTrollAmarillo(x: int, y: int): TrollAmarillo + getRanaDeFuego(x: int, y: int): RanaDeFuego + getCalabaza(x: int, y: int): Calabaza + getFantasma(x: int, y: int): Fantasma + getBotellaAzul(x: int, y: int): BotellaAzul + getBotellaRoja(x: int, y: int): BotellaRoja + getBotellaVerde(x: int, y: int): BotellaVerde + getFruta(x: int, y: int): Fruta + getVidaExtra(x: int, y: int): VidaExtra + getEscalera(x: int, y: int): Escalera + getPared(x: int, y: int): Pared + getParedDestructible(x: int, y: int): ParedDestructible + getSueloResbaladizo(x: int, y: int): SueloResbaladizo + getTrampaPinchos(x: int, y: int): TrampaPinchos + getPlataforma(x: int, y: int): Plataforma + getPlataformaMovil(x: int, y: int): PlataformaMovil + getPlataformaQuebradiza(x: int, y: int): PlataformaQuebradiza + getBolaDeNieve(x: int, y: int): BolaDeNieve + getBolaDeFuego(x: int, y: int): BolaDeFuego + getMoghera(x: int, y: int): Moghera + getKamakichi(x: int,y: int): Kamakichi

Original


```

sociedad, void
- creaLabelDesdeEntidad(entidad: Entidad): JLabel
+ incorporarEntidad(entidad: EntidadDinamica): void
+ incorporarEntidad(entidad: PlataformaMovil):
ObserverPlataformaMovil
+ incorporarEntidadEstatica(e: EntidadEstatica): JLabel
+ incorporarEntidadJugador(jugador: Jugador): void
+ incorporarSilueta(silueta: Entidad): void
+ actualizarLabelsVidas(j: Jugador): void
+ actualizarLabelsPuntaje(puntajeASumar: int): void
+ actualizarLabelsTiempo(): void
+ actualizarLabelTiempoContrarreloj(): void
+ eliminarLabel(l: JLabel): void
+ refrescar(): void

```

<<Abstra
EntidadDin

```

# direccion : int
# dirAnterior: int
# velocidad: int
# velocidadY: int
# saltando: boolean
# enAire: boolean
- bajandoPlataforma: bool
- alturaSalto: int
# alturaMax: int
# tocandoSuelo: int
# velocidadSalto: int
# ultimoSalto: long
# observer: ObserverEntidad
- miNivel: Nivel

```

```

+ getSaltando(): boolean
+ setDireccion(direccion: int)
+ setSaltando(salto: boolean)
+ setBajarPlataforma(b: boolean)
+ mover(): void
+ setTocandoSuelo(estado: boolean)
+ generarClaveEstado(): int
+ getTocandoSuelo(): int
+ getObserver(): ObserverEntidad
+ verificarSuelo(): void
+ actualizarCaida(): void
+ iniciarSalto(): void
+ actualizarSalto(): void
# visitarPlataforma(e: Plataforma)
# verificarPared(p: Pared)
+ seguirCaminando(e: Entidad)
+ eliminar(): void
+ setNivel(n: Nivel): void
+ posicionarHitbox(y: int)

```

BolaDeFuego

```

# tiempo: Temporizador
- tiempoCaida: int
- decrementos: int
- caida: boolean
- spriteJefeKamakichi: boolean=false
- spriteJefeMoghera: boolean=false

```

```

+ mover(): void
+ getTimer(): Temporizador
- subida(): void
- decrementarY(): void
+ setDireccion(direccionDisparo: int): void
+ kamakichiBomba(): boolean
- actualizarBolaMoghera(): void
+ setSpriteKamakichi(b: boolean): void
+ setSpriteMoghera(b: boolean): void

```

BolaDeNieve

```

-estaActiva: boolean
- tiempo: Temporizador
- velocidadMaxima: int
- reboto: boolean
# puntajeEmpujado: int
# gestor: GestorEntidad

```

```

+ mover(): void
+ getEstado(): boolean
+ getTimer(): Temporizador
+ disparar(): int
- terminarRebotar(): void
- aparecerPowerUp(): void
+ empujar(): void
+ getPuntajeEmpujado(): int
+ setPuntajeEmpujado(punto: int)

```

```
+ terminoElNivel(): boolean
+ eliminarse(): void
```

*ct>>
iamica*

lean

dad

int): void
ean): void
olean):

o: int): void
String

rEntidad

taforma): void
l: void
tidad): void

int

Silueta

```
+ generarClaveEstado(): String
+ eliminar(): void
```

*<<Abstract>>
Entidad*

```
# sprite: Sprite
# posX: int
# posY: int
# alto: int
# ancho: int
# hitbox: HitBox
# nivel: InterfazEliminar

+ setPosX(x: int): void
+ setPosY(y: int): void
+ getPosX(): int
+ getPosY(): int
+ getHitbox(): HitBox
+ getSprite(): Sprite
+ eliminar(): void
+ setNivel(l: Nivel): void
```

o

DisparoJugador

Jugador

```
# tiempoEfecto: Temporizador
# potenciaDisparo: int
# velocidad: int
# tiempoCaida: int
# aumentos: int
# efectoBotella: boolean
```

```
- vidaDelInicio: int=3
# miJuego: Juego
# vidas: int
# nombre: String
# disparoAumentado: boolean
# posXOriginal: int
# posYOriginal: int
# habilitadoASubirEscalera: boolean
# invencible: boolean
# bolaEmpujada: BolaDeNieve
# chocoParedIzq: boolean=false
# chocoParedDer: boolean=false
```

or

```
+ mover(): void
+ setDireccion(direccion: int, dirAnt: int): void
- caida(): void
+ incrementarY(): void
+ setPotenciaAumentada(efectoActivo: boolean): void
+ getDaño(): int
+ eliminar(): void
```

```
+ mover(direccionAMover: int):void
+ disparar(): DisparoJugador
+ setVelocidad(v: int): void
+ setVelocidadBase(v): void
```

DemonioRojo

```
# puntajeCongelado: int = 150
# puntajeEmpujado: int = 300
# contadorAnimacion: int=0
- tiempoEstimado: long=0
- estado: int=
```

int
intaje: int):

<<Abstract>>
Enemigo

```
# salud: int
# temporizadoCongelado: Temporizador
# accionMovimientoEnemigo: Temporizador
# rodando: boolean
# muerto: boolean
# congelado: boolean
# detenido: boolean
# tiempoEstado: long=0
# inicioCongelado: long=0
# estado:
int=EstadoDePatron.MOVERDERECHA
# activarJefe: boolean
# jefeDerrotado: boolean
# gestor: GestorEntidad
```

```
+ detener(): void
+ avanzar(): void
+ patronDeMovimiento(): void
+ getSalud(): int
# verificarPared(p: Pared): void
+ recibirDaño(daño: int): void
+ generarClaveEstado(): String
+ actualizarTiempoCongelado(): void
+ setCongelado(b: boolean): void
+ getCongelado(): boolean
+ getRodando(): boolean
+ getMuerto(): muerto
+ getDerrotado(): boolean
+ getInicioCongelado(): long
+ congelar(): void
+ descongelar(): void
+ convertirEnBola(): void
+ eliminar()
```

TrollAmarillo

```
# puntajeCongelado: int = 300
# puntajeEmpujado: int = 500
- tiempoEstado: long
```

```
+ patronDeMovimiento():
```

RanaDeFuego

```
# puntajeCongelado: int = 150
# puntajeEmpujado: int = 300
# lanzandoBolas: boolean
# direccionDisparo: int
- holan anzada: boolean=false
```

Fantasma

```
- xJugador: int
- yJugador: int
```

```
+ setPosicionJugador(x: int, y:int): void
+ patronDeMovimiento(): void
```

Calab

```
# tiempoCreacionFanta
- tiempoEstado: long=0
- estado: int=
EstadosDePatron.MOV
```

```

<<Abstract>>
EntidadEstatica

# miLabel: JLabel

+ generarClaveEstado(): String
+ setMiLabel(j: JLabel): void
+ getMiLabel(): void
+ borrarLabel(): void

```

```

<<Abstract>>
Obstaculo

+eliminar():void

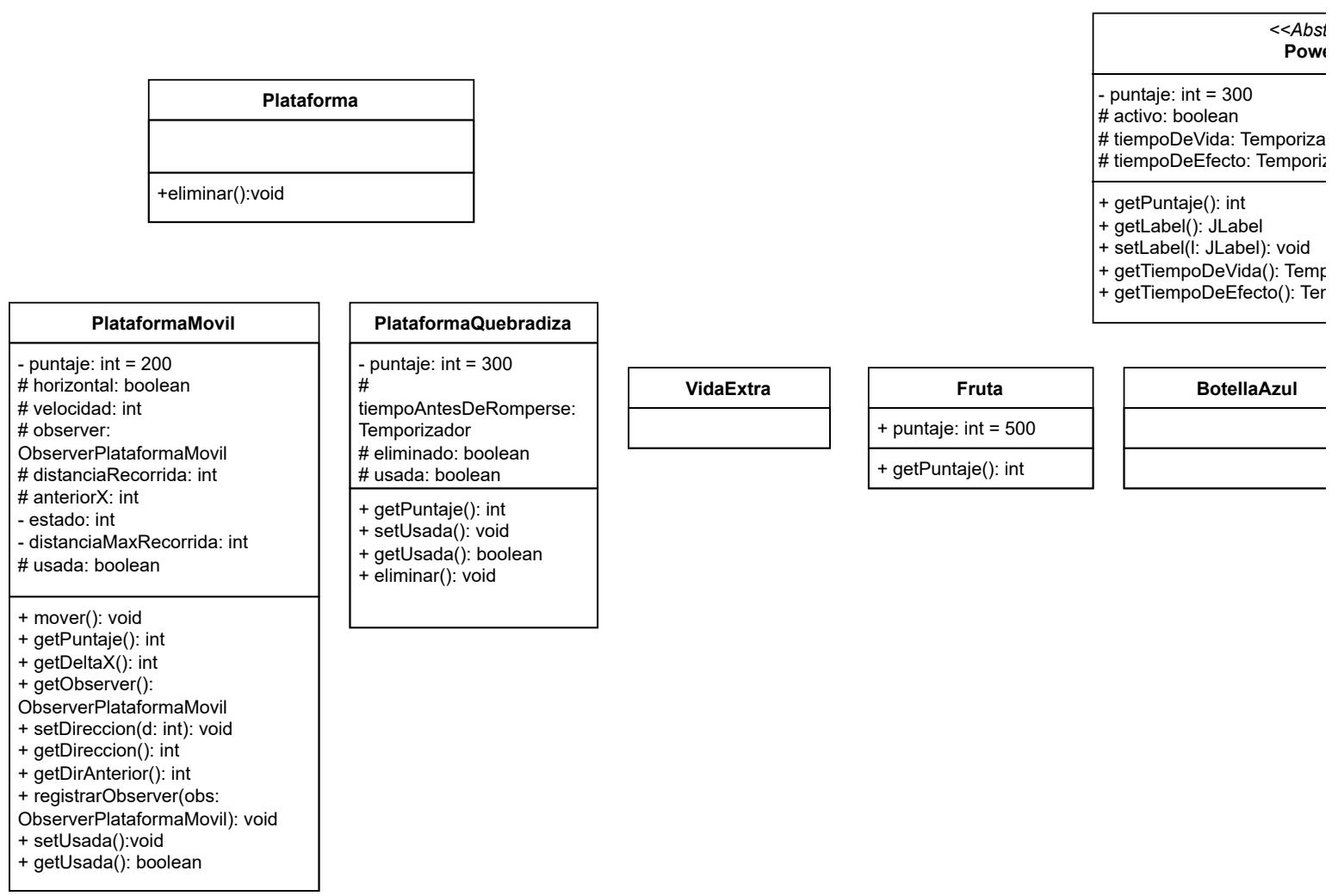
```

TrampaPinchos	SueloResbaladizo	Escalera	Pared	ParedDestructible
				+ destruirse(): void + eliminarPared(): void + getPuntaje(): int

aza
isma: Temporizador
'ERIZQUIERDA

Moghera
puntajeCongelado: int = 1000
puntajeEmpujado: int = 5000
- altoJefe: int
congelado: boolean
- holal anzada: boolean

Kamakichi
puntajeCongelado: int = 1000
puntajeEmpujado: int = 5000
activarDisparo: boolean
patronDisparo: boolean



tract>>
erUp

idor
zador

orizador
mporizador

BotellaRoja

BotellaVerde

+ eliminar(): void

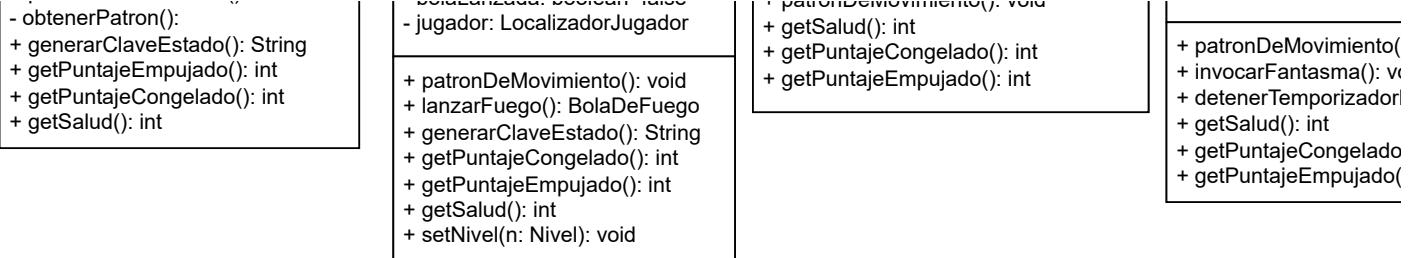
+ setGestor(g: GestorEntidad): void
+ eliminar(): void

dad): void

```
+ setVelocidadBase(): void  
+ setEmpujando(b: BolaDeNieve): void  
+ actualizarPuntaje(puntaje: int): void  
+ getVidas(): int  
+ getJuego(): Juego  
+ getTiempo(): int  
+ getNombre(): String  
+ setNombre(nombre: String): void  
+ getVelocidad(): int  
+ getBolaEmpujada(): BolaDeNieve  
+ setDireccion(direccionActual: int): void  
+ setJuego(j: Juego): void  
+ getChocoParedIzq: boolean  
+ getChocoParedDer: boolean  
+ setChocoPared(b: boolean): void  
+ chocaLadoPlataforma(ladoplataforma: boolean): void  
+ perderVida(): void  
+ iniciarInvencibilidad(): void  
+ getInvencibilidad(): boolean  
+ setVidas(): void  
- efectoBotellaAzul(b: BotellaAzul): void  
- efectoBotellaRoja(b: BotellaRoja): void  
+ getDisparoAumentado(): boolean  
- efectoBotellaVerde(b: BotellaVerde): void  
- efectoPowerUp(b: PowerUp): void  
+ estaHabilitadoASubirEscalera(): boolean  
+ eliminar(): void
```

```
Estado: ...  
EstadosDePatron.MOVERIZQUII  
  
+ patronDeMovimiento(): void  
+ getPuntajeCongelado(): int  
+ getPuntajeEmpujado(): int  
+ generarClaveEstado(): String  
+ getSalud(): int
```

ERDA



) : void
oid
DeFantasma(): void
) : int
(: int

solidarizada: boolean
- recorrido: int

+ patronDeMovimiento(): void
+ lanzarFuego(): BolaDeFuego
+ setActivo(t: boolean): void
+ convertirEnBola(): void
+ getPuntajeCongelado(): int
+ getPuntajeEmpujado(): int
+ getSalud(): int
+ setNivel(n: Nivel): void
+ posicionarHitbox(y: int): int
+ seguirCaminando(e: Entidad): void
+ eliminar(): void

+ verificarSuelo(): void
+ patronDeMovimiento(): void
+ dispararBomba(): void
+ setActivo(t: boolean): void
+ agregarBomba(posXInicial: int,
posYInicial: int, direccionDisparo:
int): void
+ convertirEnBola(): void
+ getPuntajeCongelado(): int
+ getPuntajeEmpujado(): int
+ getSalud(): int
+ generarClaveEstado(): String
+ eliminar(): void

