





Launch
+ main()

#

er

Juego
nivel: Nivel







<i>&lt;&lt;Interface&gt;&gt;</i> <b>Observer</b>
+ actualizar(): void + actualizar(direccion: int, estadoAereo: int): void

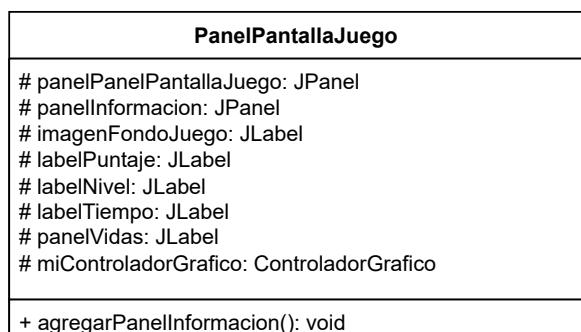
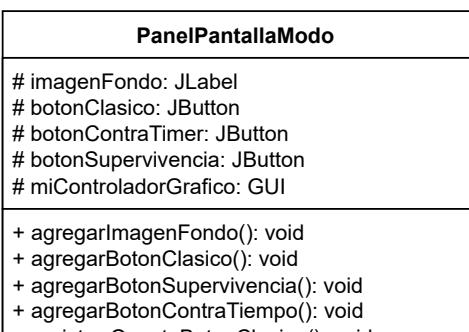
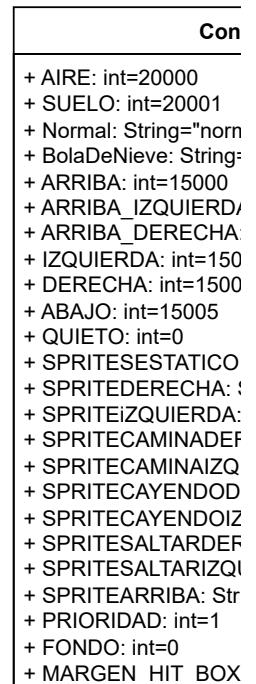
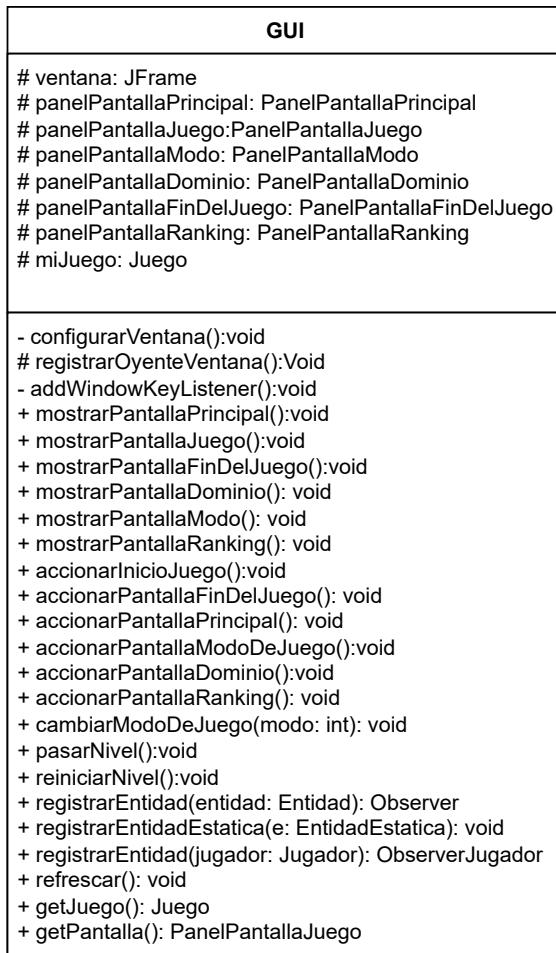
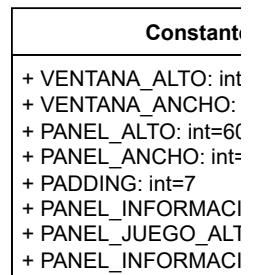
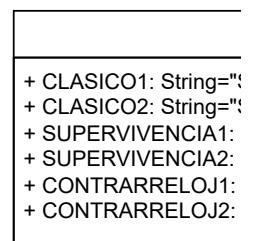
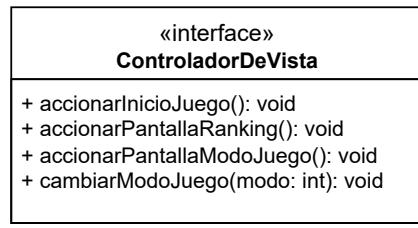
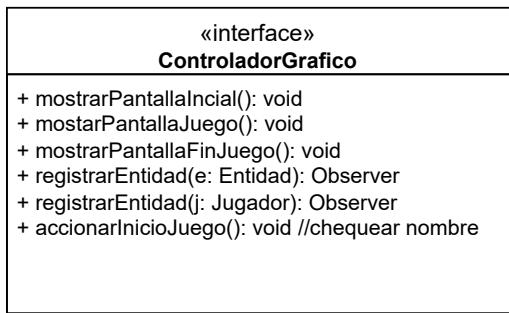
<b>ObserverGrafico</b>
# entidadObservada: Entidad
+ actualizar(): void # actualizarImagen(): void # actualizarPosicionTamano(): void + actualizar(direccion: int, estadoAereo: int): void

<b>ObserverEntidad</b>
+ actualizar(direccion: int, estadoAereo: int): void

<b>ObserverJugador</b>
# jugadorObservado: Jugador  + actualizar(direccion: int, estadoAereo: boolean): void # actualizarImagen(direccion: int, estadoAereo: boolean): void

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

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



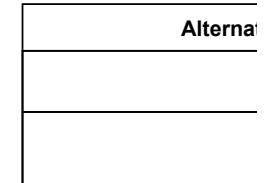
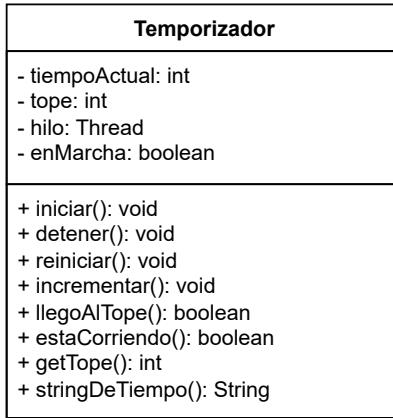
<h3>ConstantesNiveles</h3> <pre>SnowBros/src/Niveles/clasico 1.txt" SnowBros/src/Niveles/clasico 2.txt" String="SnowBros/src/Niveles/supervivencia 1.txt" String="SnowBros/src/Niveles/supervivencia 2.txt" String="SnowBros/src/Niveles/contrarreloj 1.txt" String="SnowBros/src/Niveles/contrarreloj 2.txt"</pre>	<h3>Ranking</h3> <pre># topClasico: Map&lt;String, Integer&gt; # topSupervivencia: Map&lt;String, Integer&gt; # topContraTiempo: Map&lt;String, Integer&gt;  + setJugador(nombre: String, puntaje: int, modo: String): void + ordenarTopClasico(): void + ordenarTopSupervivencia(): void + ordenarTopContrarreloj(): void + getTopClasico(): LinkedHashMap&lt;String, Integer&gt; + getTopSupervivencia(): LinkedHashMap&lt;String, Integer&gt; + getTopContrarreloj(): LinkedHashMap&lt;String, Integer&gt; - ordenar(mapeo: LinkedHashMap&lt;String, Integer&gt;): void</pre>
<h3>VariablesVistas</h3> <pre>t=660 int=832 08 =832  ION_ANCHO: int=768 FO: int=600 ION_ANCHO: int=70</pre>	<h3>ConstantesTiempos</h3> <pre>+ TIEMPO_CALABAZA: int=120 + 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=10 + TIEMPO_CONGELADO_ENEMIGOS: int=10 + TOPE_TIEMPO_JUEGO: int=120 + TIEMPO_SALTO: int=4</pre>
<h3>ConstantesEntidades</h3> <pre>nal" ="BolaDeNieve"  A: int=15001 .: int=15002 )03 )4  : String="estatico" String="derecha" : String="izquierda" RECHA: String="avanzarDerecha" UIERDA: String="avanzarIzquierda" ERECHA: String="caerDerecha" QUIERDA: String="caerIzquierda" RECHA: String="saltandoDerecha" UIERDA: String="saltandoIzquierda" ing="subiendo"  : int=5</pre>	<h3>ConstantesDificultad</h3> <pre>+ POTENCIA_DISPARO_AUMENTADA: int=20 + POTENCIA_DISPARO_NORMAL: int=10 + VELOCIDAD_DISPARO: int=15 + SALUD_DEMONIO_ROJO: int=30 + SALUD_TROLL_AMARILLO: int=30 + SALUD_RANA_DE_FUEGO: int=30 + SALUD_MOGHERA: int=70 + SALUD_KAMAKICHI: int=70 + VELOCIDAD_DEMONIO_ROJO: int=8 + VELOCIDAD_TROLL_AMARILLO: int=8 + VELOCIDAD_RANA_DE_FUEGO: int=8 + VELOCIDAD_MOGHERA: int=8 + VELOCIDAD_KAMAKICHI: int=8 + VELOCIDAD_CALABAZA: int=8 + VELOCIDAD_FANTASMA: int=8 + VELOCIDAD_PLATAFORMA_MOVIL: int=4 + VELOCIDAD_JUGADOR: int=8 + VELOCIDAD_SALTO_JUGADOR: int=64 + VELOCIDAD_CAIDA: int=100 + EFECTO_BOTELLA_AZUL: int=10</pre>
<h3>FinDelJuego</h3> <pre>utton GUI  : void : void id rtida(): void</pre>	<h3>PanelPantallaRanking</h3> <pre># imagenFondo: JLabel # botonSalir: JButton # miControladorGrafico: GUI  + agregarImagenFondo(): void + agregarBotonSalir(): void + registrarOyenteBotonSalir(): void + transparentarBoton(boton: JButton): void</pre>

```

    : puntajeJugador: int
    : tiempo: Temporizador
    : reiniciarJugador: boolean
    : jugador: Jugador
    : sonido: Sonido
    : ranking: Ranking
    : controladorGrafica: GUI
    : fabricaSprites: FabricaSprites
    : fabricaEntidades: FabricaEntidades
    : parser: ParserNivel

    · setControladorGrafico(miControlador: GUI): void
    · iniciar(): void
    · reiniciarNivel(): void
    · superarNivel(): void
    · actualizarPuntaje(puntajeASumar: int): void
    · getPuntaje(): int
    · getJugador(): Jugador
    · getNivelActual(): Nivel
    · getTiempoActual(): int
    · getTimer(): Temporizador
    · getRanking(): Ranking
    · getModoDeJuego(): int
    · registrarObservers(): void
    registrarObserverJugador(): void
    · registrarObserversEntidades(entidades:
    .inkedList<Entidad>): void
    · crearPowerUps(): void
    · crearNuevoPowerUp(): void
    · getModoDeJuego(): int
    · shootJugador(): void
    · estaCercaEntidad(entidadPrincipal: Entidad,
    entidadEvaluada: Entidad, umbralDeRango: double):
    void
    · estaCercaPlataforma(entidadPrincipal: Entidad,
    entidadEvaluada: Entidad, umbralDeRango: double):
    void

```

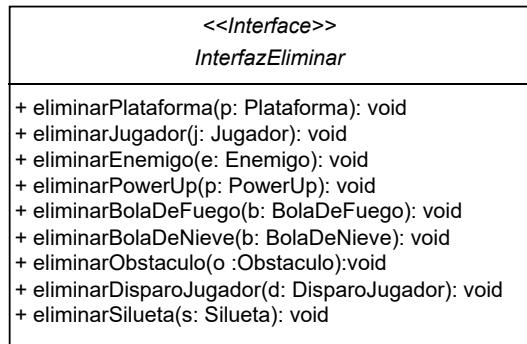


# rutaACarpeta:

```

+ getSpriteSiluet
+ getSpriteJugac
+ getSpriteDemc
+ getSpriteTrolla
+ getSpriteRana
+ getSpriteCalat
+ getSpriteFanta
+ getSpriteBotell
+ getSpriteBotell
+ getSpriteBotell
+ getSpriteFrutai
+ getSpriteVidaE
+ getSpriteEscal
+ getSpriteParec
+ getSpriteParec
+ getSpriteSuelo
+ getSpriteTramj
+ getSpritePlatal
+ getSpritePlatal
+ getSpritePlatal
+ getSpriteMogh
+ getSpriteKama
+ getSpriteBolaA
+ getSpriteBolaF
+ getSpriteDispa

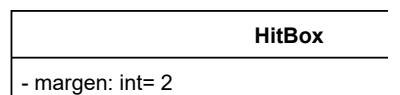
```



emigo>  
werUp>  
bstaculos>  
'lataformas>  
st<DisparoJugador>  
BolaDeNieve>  
<BolaDeFuego>

emigo>  
werUp>  
bstaculo>  
'lataformas>  
aros>  
<BolaDeNieve>  
<BolaDeFuego>

na): void  
>id  
void  
: void  
>eFuego): void  
eNieve): void



tivo

Original

«Abstract»
<b>FabricaSprites</b>
String
ta(): Sprite
dor(): Sprite
onioRojo(): Sprite
amarillo(): Sprite
DeFuego(): Sprite
aza(): Sprite
asma(): Sprite
aAzul(): Sprite
aRoja(): Sprite
aVerde(): Sprite
(): Sprite
Extra(): Sprite
lera(): Sprite
d(): Sprite
Destructible(): Sprite
Resbaladizo(): Sprite
paPinchos(): Sprite
forma(): Sprite
formaMovil(): Sprite
formaQuebradiza(): Sprite
era(): Sprite
akichi(): Sprite
nieve(): Sprite
Uego(): Sprite
roJugador(): Sprite

<b>FabricaEntidades</b>
- fabrica: FabricaSprite
+ getSilueta(): Silueta
+ getJugador(x: int, y: int): Jugador
+ 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
+ getPlataformaMoviles(x: int, y: int): PlataformaMovil
+ getPlataformaQuebradiza(x: int, y: int): PlataformaQuebradiza
+ getMoghera(x: int, y: int): Moghera
+ getKamakichi(x: int, y: int): Kamakichi

<b>ParserNivel</b>
- fabrica: FabricaEntidades
+ generarNivel(modo:int): Nivel
- cargarNivel(archivo:String, modo:int): Nivel
+ obtenerArchivoNivel(modo:int):String







- registrarUyenteBotonRanking(): void  
- transparentarBoton(jButton: JButton): void

...

```
+ registrarOyenteBotonClasico(): void
+ registrarOyenteBotonContraTiempo(): void
+ registrarOyenteBotonSupervivencia(): void
+ transparentarBoton(boton: JButton): void
```

```
+ agregarPanelJuego(): void
+ mostrarPlataformas(): void
+ incorporarEntidad(entidad: Entidad): Observer
+ incorporarEntidadEstatica(e: EntidadEstatica): void
+ incorporarEntidadJugador(jugador: Jugador): Observer
+ incorporarSilueta(silueta: Entidad): Observer
+ actualizarInformacionJugador(j: Jugador): void
+ actualizarLabelsVidas(i: int): void
+ actualizarLabelsPuntaje(j: Jugador): void
+ actualizarLabelsTiempo(j: Jugador): void
+ centrarEnJugador(posXJugador: int): void
+ agregarLabelsEditablesTiempo(): void
+ agregarLabelsEditablesPuntaje(): void
+ agregarLabelsEditablesVidas(): void
+ decorarLabelsTiempo(): void
+ decorarLabelsPuntaje(): void
+ decorarLabelsVidas(): void
+ eliminarLabel(l: JLabel): void
```

```
+ agregarBotonInvadir(): void
+ registrarOyenteBotonR(): void
+ registrarOyenteBotonS(): void
+ registrarOyenteBotonN(): void
+ transparentarBoton(bot): void
```

E1

```
# direccion : int
# dirAnterior: int
# observer: Observer
# estadoAereo : int
# velocidad: int
```

```
+ getDireccion()
+ getDirAnterior()
+ setDireccion()
+ mover(): void
+ actualizar(direccion: int): void
+ registrarObserver(observer: Observer)
+ establecerEstadoAereo(estadoAereo: int)
+ generarClaveEntidad(): String
+ getEstadoAereo(): int
+ getObserver(): Observer
+ notificarObserver(observer: Observer)
```

### BolaDeFuego

```
# tiempo: Temporizador
+ getTimer(): Temporizador
+ mover(): void
+ setDireccion(direccionDisparo: int): void
```

### Bola

```
-estaActiva: boolean
- tiempo: Temporizador
+ getEstado(): boolean
+ getTimer(): Temporizador
+ mover(): void
+ rebotar(): void
```

```

    inicio(): void
    alir(): void
    uevaPartida(): void
    on: JButton): void

```

```

+ agregarRankings(): void
+ agregarColumnaRanking(titulo: String, mapeo:
LinkedHashMap<String, Integer>, x: int, tituloY: int,
filaInicialY: int, separacion: int): void

```

```

+ eliminarObstaculo(o : Obstaculo)
+ eliminarSilueta(s: Silueta): void
+ eliminarDisparoJugador(d: DisparoJugador): void
+ agregarJugador(j: Jugador): void
+ agregarPlataforma(p: Plataforma)
+ agregarObstaculo(o: Obstaculo)
+ agregarPowerUp(p: PowerUp)
+ agregarEnemigo(e: Enemigo)
+ agregarDisparo(b: DisparoJugador)
+ agregarBolaDeNieve(b: BolaDeNieve)
+ agregarBolaDeFuego(b: BolaDeFuego)
+ efectoBotellaVerde(): void
+ getJugador(): Jugador
+ getSilueta(): Silueta

```

### <<Abstract>> EntidadDinamica

```

    :
    server
    Boolean

    :
    int
    (): int
    : void

    :accion: int, estadoAereo: int):
    rver(observer: Observer): void
    adoAereo(estado: int): void
    Estado(d: int): String
    eo(): int
    : Observer
    ver(): void

```

### <<Interface>> Visitador

```

+ visitar(e:Enemigo): void
+ visitar(p:Plataforma): 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

```

### BalaDeNieve

```

ean
izador

```

```

olean
nporizador

```

### DisparoJugador

```

# tiempoEfecto: Temporizador
# potenciaDisparo: int
# velocidad: int
# efectoBotella: boolean

+ mover(): void
+ setEfectoBotellaAzul(efecto:
boolean): void
+ activarDisparo(): void

```

### Jugador

```

- vidaDelInicio: int=3
# miJuego: Juego
# tocandoSuelo: int
# puntajeJugador: int
# velocidad: int
# vidas: int
# nombre: String
# disparoAumentado: boolean
# posXOriginal: int
# posYOriginal: int
# timerCaida: Timer

+ mover(direccionAMover: int):void
+ verificarSuelo(): void
+ iniciarCaida(): void
+ actualizarCaida(): void
+ iniciarSalto(): void
+ disparar():
+ actualizarPuntaje(puntaje: int): void
+ generarClaveEstado(direccionAnterior: int): String
+ getVidas(): int
+ getTiempo(): int
+ getDireccion(): int
+ getNombre(): String
+ getDireccionAnterior(): int
+ getPuntaje(): int
+ getVelocidad(): int
+ setDireccion(direccionActual: int): void
+ setJuego(j: Juego): void
+ nroPartidas(): void

```

### DemonioRojo

```

# puntajeCongelado: int = 150
# puntajeEmpujado: int = 300
# contadorAnimacion: int

+ generarClaveEstado(): String

```

```

    o):void
    paroJugador): void
    id
na): void
o): void
: void
void
ador): void
eNieve): void
eFuego): void

```

```

    " Poses "
# posY: int
# hitbox: HitBox
# miNivel: InterfazEliminar

+ setPosX(x: int): void
+ setPosY(y: int): void
+ getPosX(): int
+ getPosY(): int
+ getHitbox(): HitBox
+ getSprite(): Sprite
+ eliminar():void
+ setNivel(InterfazEliminar);void

```

```

+ mapeoDeSprites: HashMap<String, String>
+ getRutaImagen(clave: String): String

```

```

- hitBox: Rectangle
+ ancho: int
+ alto: int

+ actualizarHitbox(x: int, y: int, ancho: int)
+ getRectangle(): Rectangle
+ getTopHitBox(): Rectangle
+ getBottomHitBox(): Rectangle
+ getLeftHitBox(): Rectangle
+ getRightHitBox(): Rectangle
+ getX(): int
+ getY(): int
+ getAncho(): int
+ getAlto(): int

```

```

<<Interface>>
Visible

+ aceptar():void

```

**Silueta**

```

<<Abstract>>
Enemigo

# salud: int
# temporizadoCongelado: Temporizador
# accionMovimientoEnemigo: Temporizador
# rodando: boolean
# muerto: boolean
# congelado: boolean

+ mover(): void
+ recibirDaño()
+ getCongelado():boolean
+ getRodando(): boolean
+ getMuerto(): boolean
+ convertirEnBola(): void
+ agregarPowerUp(b: boolean): void
+ getPuntajeCongelado(): int
+ getPuntajeEmpujado(): int

```

**TrampaPiso**

```

}

```

**TrollAmarillo**

```

# puntajeCongelado: int = 300
# puntajeEmpujado: int = 500

+ generarClaveEstado(): String
+ saltar(): void

```

**RanaDeFuego**

```

# puntajeCongelado: int = 150
# puntajeEmpujado: int = 300
# lanzandoBolas: boolean
# direccionDisparo: int

+ lanzarFuego(): BolaDeFuego
+ generarClaveEstado(): String

```

**Fantasma**

**Calabaza**

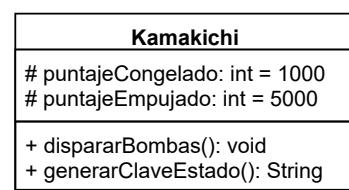
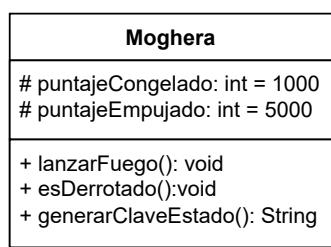
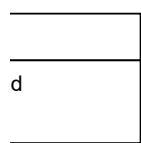
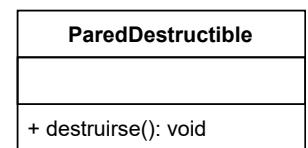
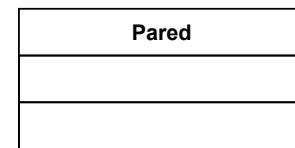
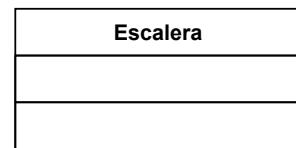
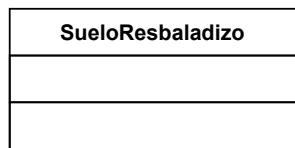
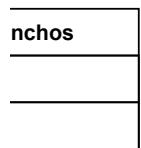
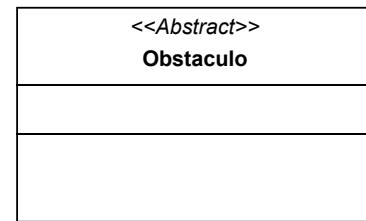
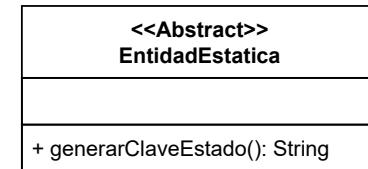
```

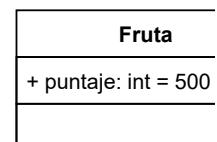
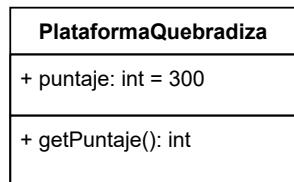
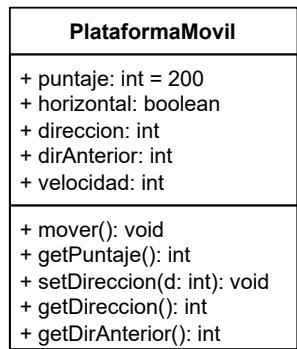
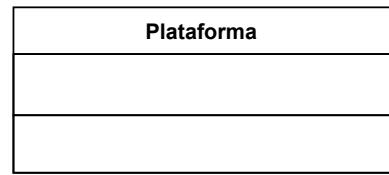
+ invocarFantasma(): void

```



int, alto: int): void

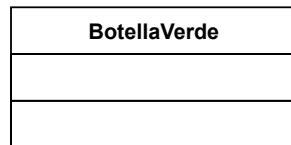
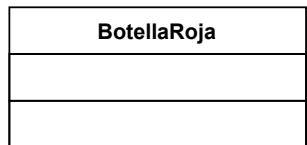
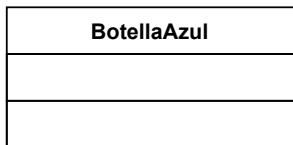




```
<<Abstract>>
PowerUp

# puntos: int = 300
# activo: boolean
# tiempoDeVida: Temporizador
# tiempoDeEfecto: Temporizador
# label: JLabel

+ getPuntaje(): int
+ getLabel(): JLabel
+ setLabel(l: JLabel): void
```







```
+ perderVida(): void  
+ setVidas(): void  
+ efectoBotellaAzul(): void  
+ efectoBotellaRoja(): void  
+ efectoBotellaVerde(): void
```







