

Launched
+ main()

<<Interface>> Observer
+ actualizar(): void + actualizar(direccion: int, estadoAereo: int): void

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

ObserverEntidad
+ actualizar(direccion: int, estadoAereo: int): void

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

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

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

«interface» ControladorGrafico
+ mostrarPantallaIncial(): void + mostarPantallaJuego(): void + mostrarPantallaFinJuego(): void + registrarEntidad(e: Entidad): Observer + registrarEntidad(j: Jugador): Observer + accionarInicioJuego(): void //chequear nombre

«interface» ControladorDeVista
+ accionarInicioJuego(): void + accionarPantallaRanking(): void + accionarPantallaModoJuego(): void + cambiarModoJuego(modos: int): void

+ CLASICO1: String="1" + CLASICO2: String="2" + SUPERVIVENCIA1: + SUPERVIVENCIA2: + CONTRARRELOJ1: + CONTRARRELOJ2:
--

Constant
+ VENTANA_ALTO: int + VENTANA_ANCHO: + PANEL_ALTO: int=60 + PANEL_ANCHO: int= + PADDING: int=7 + PANEL_INFORMACI + PANEL_JUEGO_ALT + PANEL_INFORMACI

GUI
ventana: JFrame # panelPantallaPrincipal: PanelPantallaPrincipal # panelPantallaJuego:PanelPantallaJuego # panelPantallaModo: PanelPantallaModo # panelPantallaDominio: PanelPantallaDominio # panelPantallaFinDelJuego: PanelPantallaFinDelJuego # panelPantallaRanking: PanelPantallaRanking # miJuego: Juego
- configurarVentana():void # registrarOyenteVentana():Void - addWindowKeyListener():void + mostrarPantallaPrincipal():void + mostrarPantallaJuego():void + mostrarPantallaFinDelJuego():void + mostrarPantallaDominio(): void + mostrarPantallaModo(): void + mostrarPantallaRanking(): void + accionarInicioJuego():void + accionarPantallaFinDelJuego(): void + accionarPantallaPrincipal(): void + accionarPantallaModoDeJuego():void + accionarPantallaDominio():void + accionarPantallaRanking(): void + cambiarModoDeJuego(modos: int): void + pasarNivel():void + reiniciarNivel():void + registrarEntidad(entidad: Entidad): Observer + registrarEntidadEstatica(e: EntidadEstatica): void + registrarEntidad(jugador: Jugador): ObserverJugador + refrescar(): void + getJuego(): Juego + getPantalla(): PanelPantallaJuego

Con
+ AIRE: int=20000 + SUELO: int=20001 + Normal: String="norm + BolaDeNieve: String: + ARRIBA: int=15000 + ARRIBA_IZQUIERDA/ + ARRIBA_DERECHA + IZQUIERDA: int=150 + DERECHA: int=1500 + ABAJO: int=15005 + QUIETO: int=0 + SPRITESESTATICO + SPRITEDERECHA: : + SPRITEIZQUIERDA: + SPRITECAMINADEF + SPRITECAMINAIZQ + SPRITECAYENDOD + SPRITECAYENDOIZ + SPRITESALTARDEF + SPRITESALTARIZQI + SPRITEARRIBA: Str + PRIORIDAD: int=1 + FONDO: int=0 + MARGEN_HIT_BOX

PanelPantallaModo
imagenFondo: JLabel # botonClasico: JButton # botonContraTimer: JButton # botonSupervivencia: JButton # miControladorGrafico: GUI
+ agregarImagenFondo(): void + agregarBotonClasico(): void + agregarBotonSupervivencia(): void + agregarBotonContraTiempo(): void

PanelPantallaJuego
panelPanelPantallaJuego: JPanel # panelInformacion: JPanel # imagenFondoJuego: JLabel # labelPuntaje: JLabel # labelNivel: JLabel # labelTiempo: JLabel # panelVidas: JLabel # miControladorGrafico: ControladorGrafico
+ agregarPanelInformacion(): void

PanelPantalla
imagenFondo: JLabel # botonReinicio: JButton # botonSalir: JButton # botonNuevaPartida: JB # miControladorGrafico: C
+ agregarImagenFondo(): + agregarBotonReinicio(): + agregarBotonSalir(): vo + agregarBotonNuevaPar

Ranking
<pre># topClasico: Map<String, Integer> # topSupervivencia: Map<String, Integer> # topContraTiempo: Map<String, Integer></pre>
<pre>+ setJugador(nombre: String, puntaje: int, modo: String): void + ordenarTopClasico(): void + ordenarTopSupervivencia(): void + ordenarTopContrarreloj(): void + getTopClasico(): LinkedHashMap<String, Integer> + getTopSupervivencia(): LinkedHashMap<String, Integer> + getTopContrarreloj(): LinkedHashMap<String, Integer> - orderar(mapeo: LinkedHashMap<String, Integer>): void</pre>

ConstantesTiempos
+ 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

Constantes	Dificultad
+ 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

Nivel
sprite: Sprite # jugador: Jugador # numeroNivel: int # silueta: Silueta # listaEnemigos: LinkedList<Enemigo> # listaPowerUps: LinkedList<PowerUp> # listaObstaculos: LinkedList<Obstaculo> # listaPlataformas: LinkedList<Plataforma> # listaDisparoJugador: LinkedList<Disparo> # listaBolaDeNieve: LinkedList<BolaDeNieve> # listaBolaDeFuego: LinkedList<BolaDeFuego>
+ getEnemigos(): LinkedList<Enemigo> + getPowerUps(): LinkedList<PowerUp> + getObstaculos(): LinkedList<Obstaculo> + getPlataformas(): LinkedList<Plataforma> + getDisparos(): LinkedList<Disparo> + getBolasDeNieve(): LinkedList<BolaDeNieve> + getBolasDeFuego(): LinkedList<BolaDeFuego> + getNivelActual(): int + eliminarPlataforma(p: Plataforma): void + eliminarJugador(j: Jugador): void + eliminarEnemigo(e: Enemigo): void + eliminarPowerUp(p: PowerUp): void + eliminarBolaDeFuego(b: BolaDeFuego): void + eliminarBolaDeNieve(b: BolaDeNieve): void

```

PanelPantallaRanking
# imagenFondo: JLabel
# botonSalir: JButton
# miControladorGrafico: GUI

+ agregarImagenFondo(): void
+ agregarBotonSalir(): void
+ registrarOyenteBotonSalir(): void
+ transparentarBoton(boton: JButton): void
+ ...

```

! 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

Sonido
- instancia:Sonido
- mapeo: Map <String,Clip>
+ getInstancia(): Sonido
+ reproducir(s: String): void
- cargarSonido(r: String) :Clip

Temporizador
- tiempoActual: int
- tope: int
- hilo: Thread
- enMarcha: boolean
+ iniciar(): void
+ detener(): void
+ reiniciar(): void
+ incrementar(): void
+ llegoAlTope(): boolean
+ estaCorriendo(): boolean
+ getTope(): int
+ stringDeTiempo(): String

Alternativa

rutaACarpeta:
+ getSpriteSilueta
+ getSpriteJugador
+ getSpriteDemoc
+ getSpriteTrollA
+ getSpriteRana
+ getSpriteCalab
+ getSpriteFanta
+ getSpriteBotell
+ getSpriteBotell
+ getSpriteBotell
+ getSpriteFruta
+ getSpriteVidaE
+ getSpriteEscal
+ getSpriteParec
+ getSpriteParec
+ getSpriteSuelo
+ getSpriteTram
+ getSpritePlataf
+ getSpritePlataf
+ getSpritePlataf
+ getSpriteMogh
+ getSpriteKama
+ getSpriteBolaM
+ getSpriteBolaF
+ getSpriteDispa

emigo>

werUp>

bstaculos>

'lataformas>

st<DisparoJugador>

BolaDeNieve>

<BolaDeFuego>

emigo>

werUp>

bstaculo>

'lataformas>

aros>

<BolaDeNieve>

!<BolaDeFuego>

na): void

oid

void

: void

leFuego): void

eNieve): void

<<Interface>> InterfazEliminar
+ eliminarPlataforma(p: Plataforma): void
+ eliminarJugador(j: Jugador): void
+ eliminarEnemigo(e: Enemigo): void
+ eliminarPowerUp(p: PowerUp): void
+ eliminarBolaDeFuego(b: BolaDeFuego): void
+ eliminarBolaDeNieve(b: BolaDeNieve): void
+ eliminarObstaculo(o :Obstaculo):void
+ eliminarDisparoJugador(d: DisparoJugador): void
+ eliminarSilueta(s: Silueta): void

<<Abstract>> Entidad
sprite: Sprite
nosX: int

Sprite

HitBox
- margen: int= 2

tivo

Original

«Abstract»
FabricaSprites
String
ta(): Sprite dor(): Sprite onioRojo(): Sprite \marillo(): Sprite iDeFuego(): Sprite oaza(): Sprite asma(): Sprite laAzul(): Sprite laRoja(): Sprite laVerde(): Sprite (): Sprite Extra(): Sprite lera(): Sprite d(): Sprite dDestructible(): Sprite oResbaladizo(): Sprite paPinchos(): Sprite forma(): Sprite formaMovil(): Sprite formaQuebradiza(): Sprite iera(): Sprite akichi(): Sprite Nieve(): Sprite -uego(): Sprite aroJugador(): Sprite

FabricaEntidades
- 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): PlatafornaMovil + getPlataformaQuebradiza(x: int, y: int): PlataformaQuebradiza + getMoghera(x: int, y: int): Moghera + getKamakichi(x: int,y: int): Kamakichi

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




```
- registrarUyenteBotonKanking(): void  
- transparentarBoton(boton: JButton): void
```

```
- transparentarBoton(boton: 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

+ agregarBotonReactivarEn

+ registrarOyenteBotonRe

+ registrarOyenteBotonSa

+ registrarOyenteBotonNi

+ transparentarBoton(bot

Entidad
direccion : int
dirAnterior: int
observer: Obs
estadoAereo :
velocidad: int
+ getDireccion()
+ getDirAnterior
+ setDireccion():
+ mover(): void
+ actualizar(dire
void
+ registrarObser
+ establecerEst
+ generarClavef
+ getEstadoAere
+ getObserver():
+ notificarObser

BolaDeFuego	Bola
# tiempo: Temporizador	-estaActiva: boole
+ getTimer(): Temporizador	- tiempo: Tempor
+ mover(): void	+ getEstado(): bo
+ setDireccion(direccionDisparo: int): void	+ getTimer(): Terr
	+ mover(): void
	+ rebotar(): void

<pre> + agregarRankings(): void + agregarColumnaRanking(titulo: String, mapeo: LinkedHashMap<String, Integer>, x: int, tituloY: int, filaInicialY: int, separacion: int): void </pre>	<pre> + eliminarObstaculo(o: Obstaculo): void + eliminarSilueta(s: Silueta): void + eliminarDisparoJugador(d: DisparoJugador): void + agregarJugador(j: Jugador): void + agregarPlataforma(p: Plataforma): void + agregarObstaculo(o: Obstaculo): void + agregarPowerUp(p: PowerUp): void + agregarEnemigo(e: Enemigo): void + agregarDisparo(b: DisparoJugador): void + agregarBolaDeNieve(b: BolaDeNieve): void + agregarBolaDeFuego(b: BolaDeFuego): void + efectoBotellaVerde(): void + getJugador(): Jugador + getSilueta(): Silueta </pre>
---	--

<<Abstract>> EntidadDinamica	<<Interface>> Visitador
<pre> + EntidadDinamica() + server() + Boolean() + int + (): int + void + accion: int, estadoAereo: int): void + server(observer: Observer): void + estadoAereo(estado: int): void + Estado(d: int): String + estado(): int + server(): Observer + server(): void </pre>	<pre> + 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 </pre>

Disparo	Jugador	Jugador	DemonioRojo
Disparo # 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(): void + 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 + ordenarVidas(): 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(): void + 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 + ordenarVidas(): void	DemonioRojo # puntajeCongelado: int = 150 # puntajeEmpujado: int = 300 # contadorAnimacion: int + generarClaveEstado(): String

o):void l paroJugador): void id na): void o): void : void void ador): void eNieve): void leFuego): void	// posY: int # 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> + getRutalmagen(clave: String): String	- hitBox: Rectangle + ancho: int + alto: int + actualizarHitbox(x: int, y: int, ancho: i + getRectangle(): Rectangle + getTopHitBox(): Rectangle + getBottomHitBox(): Rectangle + getLeftHitBox(): Rectangle + getRightHitBox(): Rectangle + getX(): int + getY(): int + getAncho(): int + getAlto(): int
---	--	--	---

<<Interface>> Visitable
+ 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

TrampaPi

	<table><tr><td>TrollAmarillo</td></tr><tr><td># puntajeCongelado: int = 300 # puntajeEmpujado: int = 500</td></tr><tr><td>+ generarClaveEstado(): String + saltar(): void</td></tr></table>	TrollAmarillo	# puntajeCongelado: int = 300 # puntajeEmpujado: int = 500	+ generarClaveEstado(): String + saltar(): void	<table><tr><td>RanaDeFuego</td></tr><tr><td># puntajeCongelado: int = 150 # puntajeEmpujado: int = 300 # lanzandoBolas: boolean # direccionDisparo: int</td></tr><tr><td>+ lanzarFuego(): BolaDeFuego + generarClaveEstado(): String</td></tr></table>	RanaDeFuego	# puntajeCongelado: int = 150 # puntajeEmpujado: int = 300 # lanzandoBolas: boolean # direccionDisparo: int	+ lanzarFuego(): BolaDeFuego + generarClaveEstado(): String	<table><tr><td>Fantasma</td></tr><tr><td></td></tr></table>	Fantasma		<table><tr><td>Calabaza</td></tr><tr><td>+ invocarFantasma(): voi</td></tr></table>	Calabaza	+ invocarFantasma(): voi
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(): voi														



int, alto: int): void

<<Abstract>> EntidadEstatica
+ generarClaveEstado(): String

<<Abstract>> Obstaculo

nchos

SueloResbaladizo

Escalera

Pared

ParedDestructible
+ destruirse(): void

d

Moghera
puntajeCongelado: int = 1000 # puntajeEmpujado: int = 5000
+ lanzarFuego(): void + esDerrotado():void + generarClaveEstado(): String

Kamakichi
puntajeCongelado: int = 1000 # puntajeEmpujado: int = 5000
+ dispararBombas(): void + generarClaveEstado(): String

Plataforma

PlataformaMovil
+ puntaje: int = 200 + horizontal: boolean + direccion: int + dirAnterior: int + velocidad: int
+ mover(): void + getPuntaje(): int + setDireccion(d: int): void + getDireccion(): int + getDirAnterior(): int

PlataformaQuebradiza
+ puntaje: int = 300
+ getPuntaje(): int

VidaExtra

Fruta
+ puntaje: int = 500

<<Abstract>> PowerUp
puntos: int = 300 # activo: boolean # tiempoDeVida: Temporizador # tiempoDeEfecto: Temporizador # label: JLabel
+ getPuntaje(): int + getLabel(): JLabel + setLabel(l: JLabel): void

	BotellaAzul	BotellaRoja	BotellaVerde


```
+ perUerviva(). void  
+ setVidas(): void  
+ efectoBotellaAzul(): void  
+ efectoBotellaRoja(): void  
+ efectoBotellaVerde(): void
```


