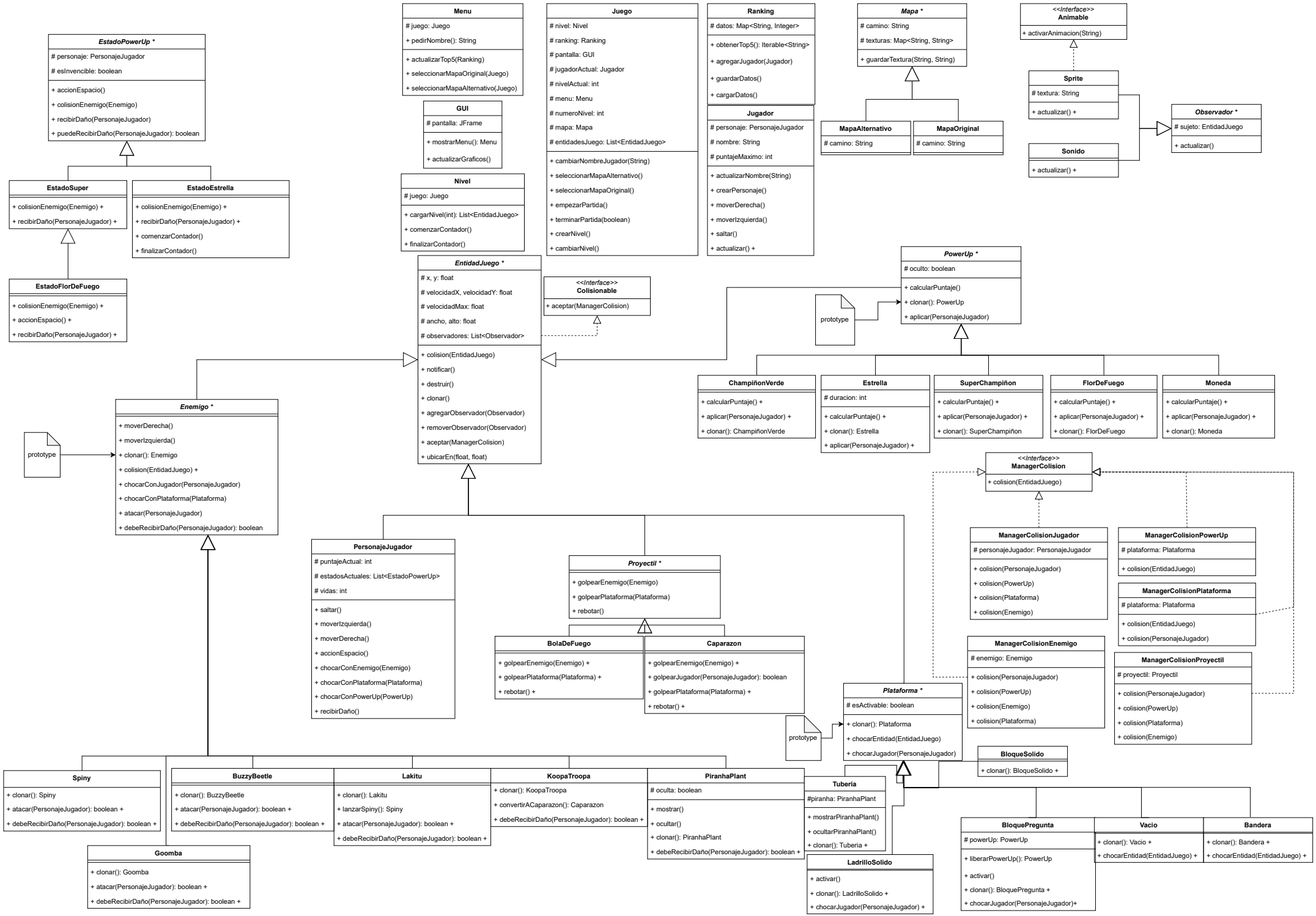
EstadoPowerUp *
personaje: PersonajeJugador
esInvencible: boolean
+ accionEspacio()
+ colisionEnemigo(Enemigo)
+ recibirDaño(PersonajeJugador)
+ puedeRecibirDaño(PersonajeJugador): boolean
EstadoSuper
+ colisionEnemigo(Enemigo) +
+ recibirDaño(PersonajeJugador) +
EstadoEstrella
+ colisionEnemigo(Enemigo) +
+ recibirDaño(PersonajeJugador) +
+ comenzarContador()
+ finalizarContador()
EstadoFlorDeFuego
+ colisionEnemigo(Enemigo) +
+ accionEspacio() +
+ recibirDaño(PersonajeJugador) +
Menu
juego: Juego
+ pedirNombre(): String
+ actualizarTop5(Ranking)
+ seleccionarMapaOriginal(Juego)
+ seleccionarMapaAlternativo(Juego)
GUI
pantalla: JFrame
+ mostrarMenu(): Menu
+ actualizarGraficos()
Nivel
juego: Juego
+ cargarNivel(int): List<EntidadJuego>
+ comenzarContador()
+ finalizarContador()
Juego
nivel: Nivel
ranking: Ranking
pantalla: GUI
jugadorActual: Jugador
nivelActual: int
menu: Menu
numeroNivel: int
mapa: Mapa
entidadesJuego: List<EntidadJuego>
+ cambiarNombreJugador(String)
+ seleccionarMapaAlternativo()
+ seleccionarMapaOriginal()
+ empezarPartida()
+ terminarPartida(boolean)
+ crearNivel()
+ cambiarNivel()
Ranking
datos: Map<String, Integer>
+ obtenerTop5(): Iterable<String>
+ agregarJugador(Jugador)
+ guardarDatos()
+ cargarDatos()
Jugador
personaje: PersonajeJugador
nombre: String
puntajeMaximo: int
+ actualizarNombre(String)
+ crearPersonaje()
+ moverDerecha()
+ moverIzquierda()
+ saltar()
+ actualizar() +
Mapa *
camino: String
texturas: Map<String, String>
+ guardarTextura(String, String)
MapaAlternativo
camino: String
MapaOriginal
camino: String
<<Interface>>
Animable
+ activarAnimacion(String)
Sprite
textura: String
+ actualizar() +
Sonido
+ actualizar() +
Observador *
sujeto: EntidadJuego
+ actualizar()
EntidadJuego *
x, y: float
velocidadX, velocidadY: float
velocidadMax: float
ancho, alto: float
observadores: List<Observador>
+ colision(EntidadJuego)
+ notificar()
+ destruir()
+ clonar()
+ agregarObservador(Observador)
+ removerObservador(Observador)
+ aceptar(ManagerColision)
+ ubicarEn(float, float)
<<Interface>>
Colisionable
+ aceptar(ManagerColision)
PowerUp *
oculto: boolean
+ calcularPuntaje()
+ clonar(): PowerUp
+ aplicar(PersonajeJugador)
prototype
ChampiñonVerde
+ calcularPuntaje() +
+ aplicar(PersonajeJugador) +
+ clonar(): ChampiñonVerde
Estrella
duracion: int
+ calcularPuntaje() +
+ clonar(): Estrella
+ aplicar(PersonajeJugador) +
SuperChampiñon
+ calcularPuntaje() +
+ aplicar(PersonajeJugador) +
+ clonar(): SuperChampiñon
FlorDeFuego
+ calcularPuntaje() +
+ aplicar(PersonajeJugador) +
+ clonar(): FlorDeFuego
Moneda
+ calcularPuntaje() +
+ aplicar(PersonajeJugador) +
+ clonar(): Moneda
Enemigo *
+ moverDerecha()
+ moverIzquierda()
+ clonar(): Enemigo
+ colision(EntidadJuego) +
+ chocarConJugador(PersonajeJugador)
+ chocarConPlataforma(Plataforma)
+ atacar(PersonajeJugador)
+ debeRecibirDaño(PersonajeJugador): boolean
prototype
PersonajeJugador
puntajeActual: int
estadosActuales: List<EstadoPowerUp>
vidas: int
+ saltar()
+ moverIzquierda()
+ moverDerecha()
+ accionEspacio()
+ chocarConEnemigo(Enemigo)
+ chocarConPlataforma(Plataforma)
+ chocarConPowerUp(PowerUp)
+ recibirDaño()
Proyectil *
+ golpearEnemigo(Enemigo)
+ golpearPlataforma(Plataforma)
+ rebotar()
BolaDeFuego
+ golpearEnemigo(Enemigo) +
+ golpearPlataforma(Plataforma) +
+ rebotar() +
Caparazon
+ golpearEnemigo(Enemigo) +
+ golpearJugador(PersonajeJugador): boolean
+ golpearPlataforma(Plataforma) +
+ rebotar() +
<<Interface>>
ManagerColision
+ colision(EntidadJuego)
ManagerColisionJugador
personajeJugador: PersonajeJugador
+ colision(PersonajeJugador)
+ colision(PowerUp)
+ colision(Plataforma)
+ colision(Enemigo)
ManagerColisionPowerUp
plataforma: Plataforma
+ colision(EntidadJuego)
ManagerColisionPlataforma
plataforma: Plataforma
+ colision(EntidadJuego)
+ colision(PersonajeJugador)
ManagerColisionEnemigo
enemigo: Enemigo
+ colision(PersonajeJugador)
+ colision(PowerUp)
+ colision(Enemigo)
+ colision(Plataforma)
ManagerColisionProyectil
proyectil: Proyectil
+ colision(PersonajeJugador)
+ colision(PowerUp)
+ colision(Plataforma)
+ colision(Enemigo)
Plataforma *
esActivable: boolean
+ clonar(): Plataforma
+ chocarEntidad(EntidadJuego)
+ chocarJugador(PersonajeJugador)
prototype
BloqueSolido
+ clonar(): BloqueSolido +
Spiny
+ clonar(): Spiny
+ atacar(PersonajeJugador): boolean +
+ debeRecibirDaño(PersonajeJugador): boolean +
BuzzyBeetle
+ clonar(): BuzzyBeetle
+ atacar(PersonajeJugador): boolean +
+ debeRecibirDaño(PersonajeJugador): boolean +
Lakitu
+ clonar(): Lakitu
+ lanzarSpiny(): Spiny
+ atacar(PersonajeJugador): boolean +
+ debeRecibirDaño(PersonajeJugador): boolean +
KoopaTroopa
+ clonar(): KoopaTroopa
+ convertirACaparazon(): Caparazon
+ debeRecibirDaño(PersonajeJugador): boolean +
PiranhaPlant
oculta: boolean
+ mostrar()
+ ocultar()
+ clonar(): PiranhaPlant
+ debeRecibirDaño(PersonajeJugador): boolean +
Goomba
+ clonar(): Goomba
+ atacar(PersonajeJugador): boolean +
+ debeRecibirDaño(PersonajeJugador): boolean +
Tuberia
#piranha: PiranhaPlant
+ mostrarPiranhaPlant()
+ ocultarPiranhaPlant()
+ clonar(): Tuberia +
LadrilloSolido
+ activar()
+ clonar(): LadrilloSolido +
+ chocarJugador(PersonajeJugador) +
BloquePregunta
powerUp: PowerUp
+ liberarPowerUp(): PowerUp
+ activar()
+ clonar(): BloquePregunta +
+ chocarJugador(PersonajeJugador)+
Vacio
+ clonar(): Vacio +
+ chocarEntidad(EntidadJuego) +
Bandera
+ clonar(): Bandera +
+ chocarEntidad(EntidadJuego) +