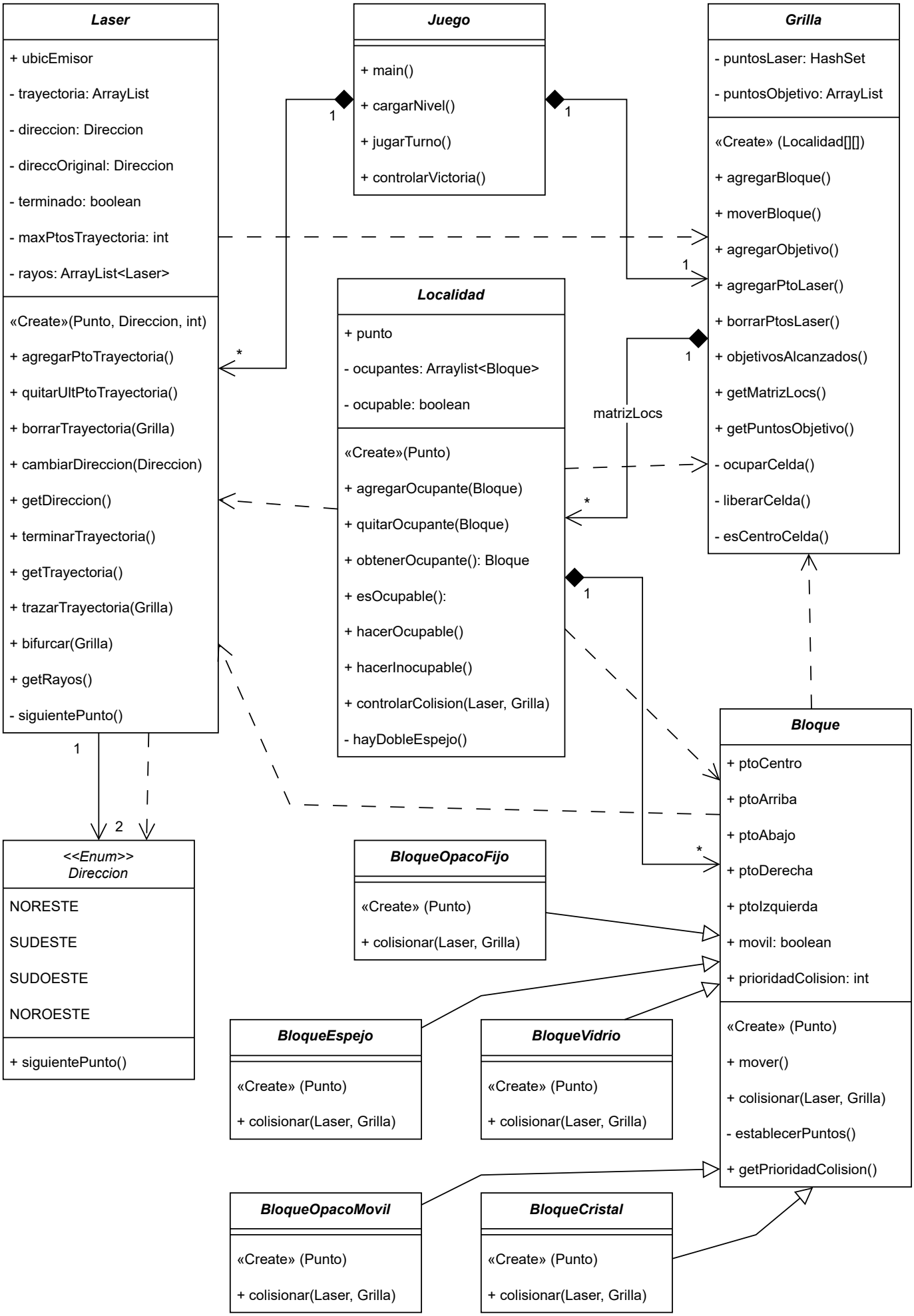
Laser
+ ubicEmisor
- trayectoria: ArrayList
- direccion: Direccion
- direccOriginal: Direccion
- terminado: boolean
- maxPtosTrayectoria: int
- rayos: ArrayList<Laser>
«Create»(Punto, Direccion, int)
+ agregarPtoTrayectoria()
+ quitarUltPtoTrayectoria()
+ borrarTrayectoria(Grilla)
+ cambiarDireccion(Direccion)
+ getDireccion()
+ terminarTrayectoria()
+ getTrayectoria()
+ trazarTrayectoria(Grilla)
+ bifurcar(Grilla)
+ getRayos()
- siguientePunto()
Juego
+ main()
+ cargarNivel()
+ jugarTurno()
+ controlarVictoria()
Grilla
- puntosLaser: HashSet
- puntosObjetivo: ArrayList
«Create» (Localidad[][])
+ agregarBloque()
+ moverBloque()
+ agregarObjetivo()
+ agregarPtoLaser()
+ borrarPtosLaser()
+ objetivosAlcanzados()
+ getMatrizLocs()
+ getPuntosObjetivo()
- ocuparCelda()
- liberarCelda()
- esCentroCelda()
Localidad
+ punto
- ocupantes: Arraylist<Bloque>
- ocupable: boolean
«Create»(Punto)
+ agregarOcupante(Bloque)
+ quitarOcupante(Bloque)
+ obtenerOcupante(): Bloque
+ esOcupable():
+ hacerOcupable()
+ hacerInocupable()
+ controlarColision(Laser, Grilla)
- hayDobleEspejo()
matrizLocs
Bloque
+ ptoCentro
+ ptoArriba
+ ptoAbajo
+ ptoDerecha
+ ptoIzquierda
+ movil: boolean
+ prioridadColision: int
«Create» (Punto)
+ mover()
+ colisionar(Laser, Grilla)
- establecerPuntos()
+ getPrioridadColision()
<<Enum>>
Direccion
NORESTE
SUDESTE
SUDOESTE
NOROESTE
+ siguientePunto()
BloqueOpacoFijo
«Create» (Punto)
+ colisionar(Laser, Grilla)
BloqueEspejo
«Create» (Punto)
+ colisionar(Laser, Grilla)
BloqueVidrio
«Create» (Punto)
+ colisionar(Laser, Grilla)
BloqueOpacoMovil
«Create» (Punto)
+ colisionar(Laser, Grilla)
BloqueCristal
«Create» (Punto)
+ colisionar(Laser, Grilla)
1
*
2