

## @ Mano DAMADEPICAS: int CORAZONES: int F OTRAS: int cartasEnMesa: int IS contadorlugadas: int o numeroJugada: int jugadores: Jugador[] a cartas/ugadas: Carta[] primeraCartalugada: Carta o jugadorPerdedor: Jugador Viano(jugadores: Jugador[]) o tirarCartaEnMesa(turnoProx: int, cartaEnJuego: Carta, puedeTirarOtraCarta: boolean, corazonesRotos: boolean): boolean primeraCarta(): boolean o determinarPerdedor(): int contarPuntos(posPerdedorJugadas: int): void

puntajeCarta(carta: Carta): int

reiniciarContadorlugadas0: void

## 1 Observable o notificar(evento: Object): void agregarObservador(observador: Observador): void O Juego cantCartasRepartidas. int cantCartasIntercambio: int puntajeMaximo: int cantlugadores: int mazo: Mazo jugadores: Jugador[] o ronda: int ugadas: List<Mano> a turno: int direction: Direction o cartaAJugar: Carta poslugadorGanador: int o corazonesRotos: boolean observadores: List < Observador > S Juego() o iniciarluego(): void cantidadDeJugadoresValida(): boolean tiroCorazones(): void tiroCartalncorrecta(): void puntajeMaximoActual(): int determinarGanador(): void pnmercarta2Trebol(jugada: Mano): void o cartasPosiblesAJugar(): String o jugarCaita(i: int): void repartirCartas(): void agregarlugadores(nombre: String): boolean o reemplazarJugadores(nombre: String, posicion: int): boolean pasajeDeCartas(): void ntercambioDeCartas(valor: int): void buscarJugador(jugador: Jugador): int otorgarCartasJugadores(cartasPasaje: ArrayList < Carta[]>): void ⊕ jugarCartaPasaje(i: int): void ⊖ cartasPosiblesAJugarPasaje(): String o getJugadorActual(): Jugador getCartasEnMesa(): String getPerdedorJugada(): String getNumeroJugada(): String getPuntajes(): String getGanadorJuego(): String

a netRondary String

## 

FIN PASAJE DE CARTAS: EventosObservador

(a) Direccion

## Observador ar.edu.unlu.corazones.observer actualizar(evento: Object, observado: Observable): voi Controlador modelo: Juego vista: IVista Controlador(modelo: Juego, vista: IVista) comenzarJuego(): void mostrarCartasPosiblesATirar(): String jugadorActual(): Jugador cartaJugada(i: int): void cartasEnMesa(): String perdedorJugada(): String numeroJugada(): String puntajesJugadores(): String andorJuego(): String anadorJuego(): String

o numeroRonda(): String

o direccionPasaje(): String

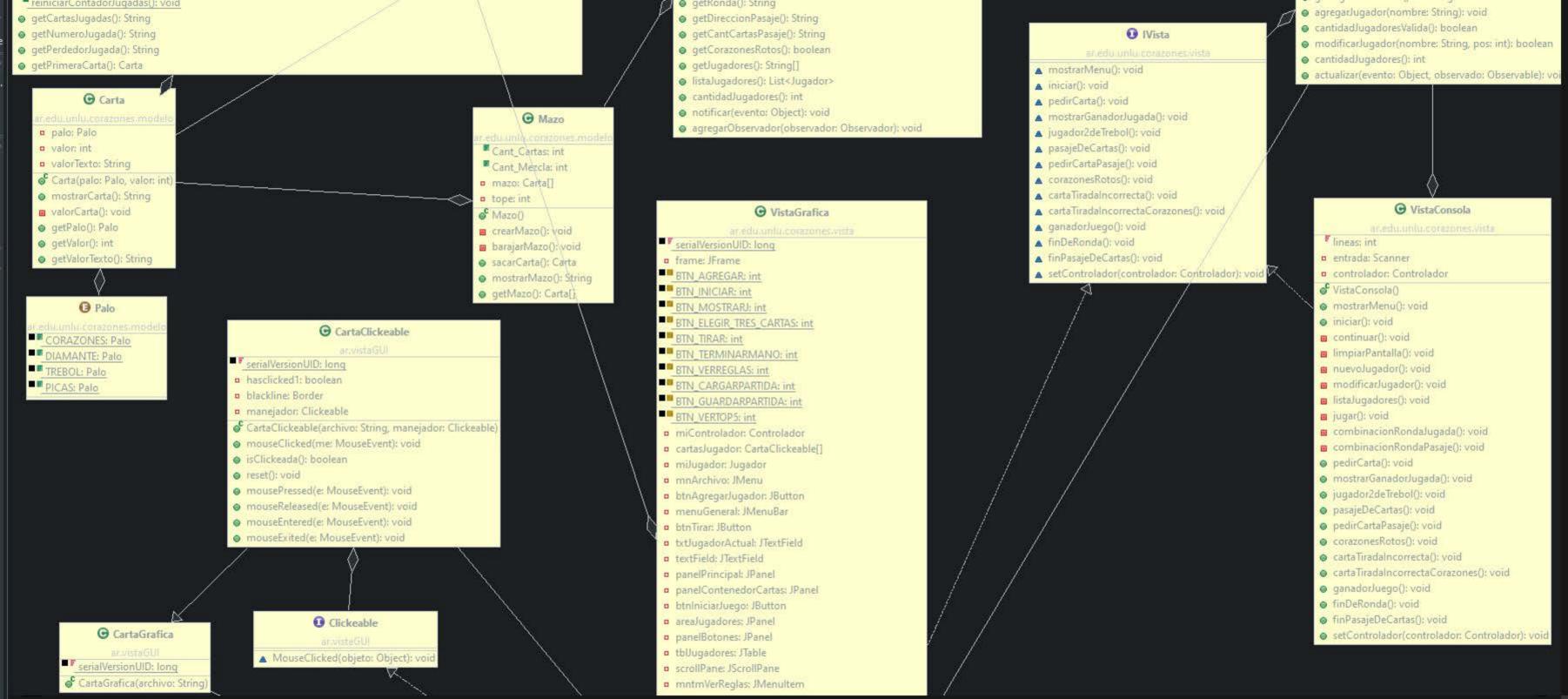
→ listaJugadores(): String[]

o cartalugadaPasaje(i: int): void

o isCorazonesRotos(): boolean

getJugadoresAsList(): List< Jugador>

o mostrarCartasPosiblesATirarPasaje(): String



```
mntmCargarPartida: JMenultem
mntmGuardarPartida: JMenultem
mntmVerTop: JMenultem
mnReglas; JMenu
mnSalin JMenultem
□ JLmensajes; JLabel
panelCartas: JPanel
a tablaTop: TablaTop
mesa: JPanel
gbc_descarte: GridBagConstraints

    txtMensajes: JTextField

a cartaReverso: CartaGrafica
gbc_reverso: GridBagConstraints
txtJugador: JTextField
panelJugadorOeste: JPanel
panelJugadorNorte: JPanel
panelJugadorEste: JPanel
panelJugadorSur: JPanel
C VistaGrafica()
o initialize(): void
seleccionarAccion(action: int): void
o buildListalugadores(): void
posicionCartaClickeada(): int
tiraCarta(): void
actualizarCartas(): void
datosJugador(jugadores: List<Jugador>): TableModel
o darFormatoJugadores(jugadores: List<Jugador>): void
actualizaPuntaje(): void
agregarJugador(): void
o setControlador(controlador: Controlador): void
o reiniciarMano(): void
o finJuego(): void
o notificarMensaje(mensaje: String): void
o notificarError(error: String): void
o juegolniciado(): void
mostrarCartas(): void
o turno(); void
getlmage(archivo: String); CartaGrafica
getClickedImage(archivo: String): CartaClickeable
manejaCartaClickeada(): CartaClickeable
MouseClicked(objeto: Object): void
o verTop(): void
o cargaJuego(): void
o iniciar(); void
o cambiaTurno(): void
o limpiaTurno(): void
o deshabilitaBotones(): void

    mostrarCartasJugadorNoActual(): void
```

```
o notificarMensaje(mensaje: String): void
notificarError(error: String): void
@ juegolniciado(): void
mostrarCartas(): void
o turno(): void
getimage(archivo: String): CartaGrafica
getClickedImage(archivo: String): CartaClickeable
manejaCartaClickeada(): CartaClickeable

→ MouseClicked(objeto: Object): void

o verlop(): void
@ cargaJuego(): void
o iniciar(): void
a cambia Turno(): void
@ limpiaTurno(): void
e deshabilitaBotones(): void
a mostrarCartasJugadorNoActual(): void
e muestraMesa(): void
@ mostrarMenu(): void
o comenzarJuego(): void
@ descarteVacio(): void
pedirCarta(): void
mostrarGanadorJugada(): void

    jugador?deTrebol(): void

pasajeDeCartas(): void
pedirCartaPasaje(): void
o corazonesRotos(): void
a cartaTiradalncorrecta(): void
o ganadorJuego(): void
@ finDeRonda(): void
o finPasajeDeCartas(): void
```