Usuario **Partida** Carta ID: int ID: int ID: int Nombre: string • ID host: int Tipo: enum(ID partida: int • Nombre: string Acción. · Contraseña: string o Defensa, 108 Rol partida: enum(o Contagio. 。 La Cosa. . Mínimo jugadores: int Obstáculo. o Infectado, · Máximo jugadores: int Pánico 1...N Humano · Cantidad iugadores: int • Nombre: string • ID vecino iza: int • Estado: enum(Descripción: string • ID vecino der: int o Espera, Jugando, 1 1 • Cartas: int[4] Terminada • obtener ID() Cuarentena: bool • obtener tipo() • Puerta trancada vec izg: bool • ID jugador actual: int • obtener nombre() Acción actual: enum(• Puerta trancada vec der: bool obtener descripcion() Alzar. Jugar validar alzar() Desechar validar desechar() Intercambiar obtener ID() validar jugar() obtener nombre() validar intercambiar() obtener ID partida() • Sentido ronda: enum(• generar efecto() Izquierda. crear() Derecha asignar partida() sacar partida() 4 obtener rol() asignar rol() obtener ID() obtener ID vecinos() obtener ID host() asignar vecinos() • obtener nombre() obtener ID cartas() obtener contraseña() sacar carta() • obtener rango jugadores() asignar carta() obtener cantidad jugadores() • obtener lista jugadores() • esta en cuarentena() obtener estado() • tiene puerta trancada() • obtener ID jugador actual() • obtener accion actual() enviar mensaje() obtener sentido ronda() · recibir mensaje() crear() asignar host() • unir usuario() sacar usuario() iniciar() terminar() eliminar() obtener paso actual() notificar paso actual() • preparar siguiente paso() validar jugada() efectuar_jugada() validar fin partida() mostrar ganadores()