

Explicación Funcionalidad:

**Método mostrarSquad(int numHumanos)** Este método lo utilizo para mostrar la información sobre los personajes que están en el juego. Primero, me aseguro de que el número de jugadores no sea mayor que la cantidad de personajes disponibles ni menor que 0. Y una vez comprobado que es siempre porque esta determinado en la clase CrearPersonaje(), muestro un mensaje indicando que los equipos ya están listos. Luego, recorro la lista de personajes y los agrupo en squads, mostrando sus características en un mensaje donde salen todas las squads.

**Método Batalla()** Aquí simulo el proceso de la batalla. Lo primero que hago es elego aleatoriamente el primer enemigo de entre los tres posibles squads enemigos. Después, llamo al método mostrarEnfrentamiento() para mostrar un mensaje con los detalles del enfrentamiento. Luego, simulo la batalla entre mi equipo (Squad 1) y el primer equipo enemigo, realizo rondas en las que ambos equipos atacan. Al final de la primera batalla, determino si mi equipo ganó o ha perdido y dependiendo de eso, paso a la fase final de la batalla entre la squad de bots restantes, llamando al método batallaFinal().

**Método JugadoresVivos(int inicio, int fin)** Este método lo utilizo para verificar si hay jugadores vivos dentro de un rango de personajes. Si al menos uno de los jugadores tiene vida mayor que cero, devuelvo true; si no, devuelvo false. Este método lo utilizo sobre todo para ver si algún equipo ha sido derrotado.

**Método batallaFinal(boolean miEquipoGana, int primerEnemigo)** Este método maneja la fase final de la batalla. Si mi equipo ganó la primera batalla, ahora se enfrenta al equipo ganador de los bots restantes. Primero, determino cuáles dos squads de bots se enfrentarán. Luego, simulo esa batalla con el método batallaEntreBots(). Finalmente, determino al ganador de la batalla entre bots y paso a simular la batalla final entre el quipo ganador de la batalla 1 y el equipo ganador de los bots de la batalla 2.

**Método batallaEntreBots(int squad1, int squad2)** En este método, simulo una batalla entre dos squads de bots. Los turnos de ataque se alternan entre los dos squads hasta que uno de ellos sea derrotado. Cada turno, los personajes de un squad atacan a los personajes del otro squad. El método devuelve el número del squad que ganó la batalla.

**Método actualizarCaracteristicasSquad(int ganadorSquad)** Este método me permite actualizar las características de los personajes del squad ganador de la

batalla entre bots. Aunque aún no está completamente implementado, la idea es que los personajes del equipo ganador reciban alguna mejora en sus características, como vida o daño.

**Método mostrarEnfrentamiento(int squadEnemigo, int inicioEnemigo, int finEnemigo)** Este método lo uso para mostrar la información de los dos equipos que están luchando en un enfrentamiento: mi equipo y el equipo enemigo.

**Método mostrarEstadoRonda(int inicio1, int fin1, int inicio2, int fin2)** Aquí muestro el estado de los dos squads después de cada ronda de combate.

**Método turnoEquipoJugador(int inicioEnemigo, int finEnemigo)** Este método simula los turnos de ataque de los jugadores. Cada jugador ataca a un enemigo aleatorio dentro del rango de personajes enemigos especificado. Si el personaje tiene vida, el daño se resta de la vida del enemigo.

**Método turnoEquipoEnemigo(int inicioEnemigo, int finEnemigo)** Aquí simulo los turnos de ataque de los enemigos. Al igual que con los jugadores, cada enemigo ataca a un jugador aleatorio dentro del rango de personajes enemigos dado. Si el enemigo tiene vida, el daño se resta de la vida del jugador atacado.

### **Problemas:**

Al principio hice el código en base a varias ArrayList pero me di cuenta que era una tontería ya que tenido la gestión en otra clase entonces lo quite y lo en base al ArraList de la clase Crearpersonaje , he tenido batante problemas para la habilidad ya que intenta llamar al método que tiene cada personaje pero no me funcionaba bien se me hacía un bucle infinito entonces me intente adaptar sacando el número del jugador para identificar a quien iba la habilidad después me encontré muchos problemas para apartar mi quipo d ellos de mas botos y lo hice con dos variables haciendo operaciones y sacando la posición y tuve fallos a la hora de actualizarEstadisticas y la hora de crear la segunda batalla simulada porque no me funcionaba bien se me creaba un bucle infinito. Y para terminar en la gestión de rounds también me hice mucho lio porque no se ponían las características donde quería pero bueno lo arregle identicando la posición y mostrando el mensaje correctamente.