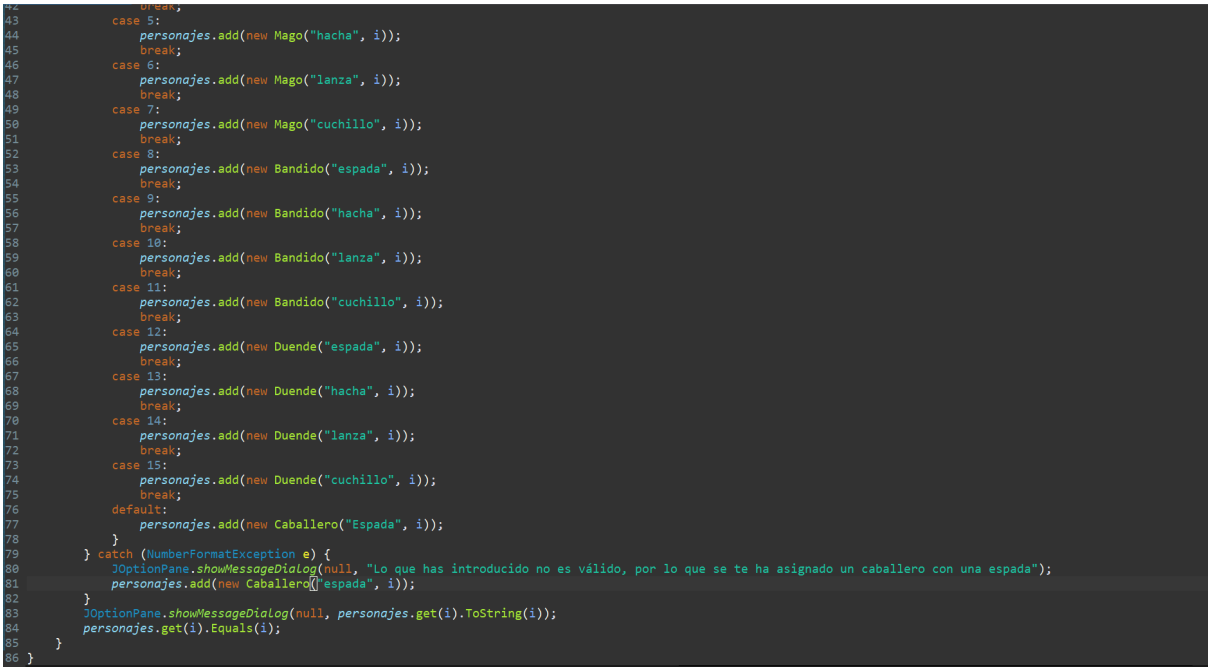
MEMORIA TRABAJO PROGRAMACIÓN RUBÉN BESTEIRO

En este trabajo yo hice 5 clases, una clase llamada CrearPersonajes, que sirve para crear a los personajes antes de que empiece la batalla, y 4 clases para los personajes, una llamada Personaje que tiene todos los métodos y las variables comunes, y una clase para cada tipo de personaje, que son Caballero, Mago, Bandido y Duende. Como esto por sí solo es menos trabajo que lo que hizo el resto, también me dediqué a arreglar bugs y a ayudar a los demás con mis conocimientos de programación.

CrearPersonajes:

```java
package CurroFinal;
import java.util.ArrayList;
import javax.swing.JOptionPane;


public class CrearPersonajes {                  // Todo esto es hecho por Rubi
    public static ArrayList<Personaje> personajes = new ArrayList<>();
    public static int numHumanos;

    public static void CreacionDePersonajes(int a) {
        numHumanos = a;     // Esto se usa para que la habilidad del mago detecte si la está usando el jugador o un bot
        int clase;

        for (int i = 0; i < numHumanos; i++) {      // Esto es para que los humanos escojan su clase
            clase = Integer.parseInt(JOptionPane.showInputDialog(null, "Jugador " + i + ", elige personaje\n" + "0-3: Caballero\n" + "4-7: Mago\n" + "8-11: Bandido\n"
            switchClase(clase, i);
        }

        for (int i = numHumanos; i < 16; i++) {     // Y esto es para que la máquina escoja su clase aleatoriamente
            clase = (int) (Math.random() * 16);
            switchClase(clase, i);
        }
    }

    static void switchClase(int clase, int i) {
        try {
            switch(clase) {
            case 0:
                personajes.add(new Caballero("espada", i));
                break;
            case 1:
                personajes.add(new Caballero("hacha", i));
                break;
            case 2:
                personajes.add(new Caballero("lanza", i));
                break;
            case 3:
                personajes.add(new Caballero("cuchillo", i));
                break;
            case 4:
                personajes.add(new Mago("espada", i));
                break;
            case 5:
                personajes.add(new Mago("hacha", i));
```



```
                break;
            case 5:
                personajes.add(new Mago("hacha", i));
                break;
            case 6:
                personajes.add(new Mago("lanza", i));
                break;
            case 7:
                personajes.add(new Mago("cuchillo", i));
                break;
            case 8:
                personajes.add(new Bandido("espada", i));
                break;
            case 9:
                personajes.add(new Bandido("hacha", i));
                break;
            case 10:
                personajes.add(new Bandido("lanza", i));
                break;
            case 11:
                personajes.add(new Bandido("cuchillo", i));
                break;
            case 12:
                personajes.add(new Duende("espada", i));
                break;
            case 13:
                personajes.add(new Duende("hacha", i));
                break;
            case 14:
                personajes.add(new Duende("lanza", i));
                break;
            case 15:
                personajes.add(new Duende("cuchillo", i));
                break;
            default:
                personajes.add(new Caballero("Espada", i));
            }
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Lo que has introducido no es válido, por lo que se te ha asignado un caballero con una espada");
            personajes.add(new Caballero("espada", i));
        }
        JOptionPane.showMessageDialog(null, personajes.get(i).ToString(i));
        personajes.get(i).Equals(i);
    }
}
```

Esta clase tiene 2 variables principales, que son el ArrayList de personajes que es del supertipo personaje, y un int que es el número de jugadores humanos, que según si le das al botón de solo, dúo o squad vale 1, 2 o 4.

Luego viene la parte donde se asignan los personajes que es en la función CreacionDePersonajes. Dentro contiene 2 bucles for, uno es para los jugadores humanos donde se les pide elegir, y el otro es para los bots donde eligen al azar. Cuando a cada jugador o bot le toca escoger, se llama a una función que es switchClase, donde hay un switch con todas las combinaciones de personajes y herramientas. Esto lo metí para no tener que reutilizar el código del switch.

Personaje:

```java
package CurroFinal;
import javax.swing.JOptionPane;


public class Personaje {
    private boolean habilidadActivada = false;      // Este booleano lo usarán las habilidades que afectan al siguiente turno
    private boolean bolaDeFuego = false;            // Si en el turno anterior un mago usó su habilidad contra el jugador, esto es true

    private String clase;       // Todas estas cosas van a ir cambiando según el personaje
    private String herramienta;
    private int vida;
    private int daño;

    void ataque(int numJugador, int i) {
        if (!this.bolaDeFuego && !(CrearPersonajes.personajes.get(numJugador).getClase() == "duende" && CrearPersonajes.personajes.get(numJugador).getHabilidadActivad
            try {           // Esto se ejecuta cuando el jugador ataca
                CrearPersonajes.personajes.get(numJugador).setVida(CrearPersonajes.personajes.get(numJugador).getVida() - this.getDaño());
                JOptionPane.showMessageDialog(null, "El jugador " + i + " (" + this.getClase() + ") ha atacado al jugador " + numJugador + " con un daño de " + this.g
            } catch (NullPointerException e) {
                JOptionPane.showMessageDialog(null, "El jugador " + i + " (" + this.getClase() + ") ha intentado atacar a un cadáver, lo cual no ha surtido efecto");
            }
        } else if (this.bolaDeFuego) {          // Si el jugador está quemado, el ataque no surte efecto y recibes daño
            this.setVida(this.getVida()-30);
            JOptionPane.showMessageDialog(null, "El jugador " + i + " (" + this.getClase() + ") ha intentado atacar estando quemado. Le queda " + this.getVida() + " d
            this.setBolaDeFuego(false);
        } else /*if (CrearPersonajes.personajes.get(numJugador).getClase() == "duende" && CrearPersonajes.personajes.get(numJugador).getHabilidadActivada())*/ {
            JOptionPane.showMessageDialog(null, "El jugador " + i + " (" + this.getClase() + ") ha intentado atacar a un duende usando su escudo, por lo que su ataque
        }
        this.habilidadActivada = false;
    }

    void ataquePeroSinJOptionPanes(int numJugador) {        // Metido a petición de Manu
        if (!this.bolaDeFuego && !(CrearPersonajes.personajes.get(numJugador).getClase() == "duende" && CrearPersonajes.personajes.get(numJugador).getHabilidadActivad
            try {           // Esto se ejecuta cuando el jugador ataca
                CrearPersonajes.personajes.get(numJugador).setVida(CrearPersonajes.personajes.get(numJugador).getVida() - this.getDaño());
            } catch (NullPointerException e) {
                // Nada
            }
        } else if (this.bolaDeFuego) {          // Si el jugador está quemado, el ataque no surte efecto y recibes daño
            this.setVida(this.getVida()-30);
            this.setBolaDeFuego(false);
        } else if (CrearPersonajes.personajes.get(numJugador).getClase() == "duende" && CrearPersonajes.personajes.get(numJugador).getHabilidadActivada()) {
            // Nada
        }
    }
```

```java
    public void habilidad(int i) {
        // Esto está vacío porque cada clase tiene su propia habilidad
    }


    // GETTERS Y SETTERS

    public String getClase() {
        return this.clase;
    }

    public void setClase(String clase) {
        this.clase = clase;
    }

    public String getHerramienta() {
        return this.herramienta;
    }

    public void setHerramienta(String herramienta) {
        this.herramienta = herramienta;
    }

    public int getVida() {
        return this.vida;
    }

    public void setVida(int vida) {
        this.vida = vida;
    }

    public int getDaño() {
        return daño;
    }

    public void setDaño(int daño) {
        this.daño = daño;
    }

    public boolean getHabilidadActivada() {
        return this.habilidadActivada;
    }

    public void setHabilidadActivada(boolean habilidadActivada) {
```

```
    public void setHabilidadActivada(boolean habilidadActivada) {
        this.habilidadActivada = habilidadActivada;
    }

    public boolean getBolaDeFuego() {
        return this.bolaDeFuego;
    }

    public void setBolaDeFuego(boolean bolaDeFuego) {
        this.bolaDeFuego = bolaDeFuego;
    }

    // OTROS MÉTODOS

    public String ToString(int posicion) {
        return ("El jugador " + posicion + " es un " + this.clase + " (" + this.vida + " de vida) cuya herramienta es " + this.herramienta + " (" + this.daño + " de d
    }

    public void Equals(int posicion) {
        for (int i = 0; i < (posicion-1); i++) {
            if (this.getClase() == CrearPersonajes.personajes.get(i).getClase() && this.getHerramienta() == CrearPersonajes.personajes.get(i).getHerramienta()) {
                JOptionPane.showMessageDialog(null, "El personaje que se acaba de crear es igual al que hay en la posición " + i);
                return;
            }
        }
    }
}
```

Esta es la clase de la que heredan las clases de los 4 tipos de personaje. Las variables comunes son todas privadas y tienen getters y setters. Las variables son los booleanos habilidadActivada (que es true cuando el jugador le da al botón de habilidad y lo usan el caballero y el duende porque sus habilidades tienen efecto el turno siguiente) y bolaDeFuego (que es true cuando un mago usa su habilidad contra ti), los strings clase y herramienta (que por ejemplo un caballero con una espada será clase = “caballero” y herramienta = “espada”), y los ints vida y daño (la vida la determina la clase escogida y el daño la herramienta). Acto seguido viene el método del ataque básico, donde se le pasa como argumento el jugador al que se le va a atacar. El método hace daño según que herramienta tenga el que ataca, excepto si la víctima está muerta (NullPointerException), el jugador tiene su booleano bolaDeFuego = true (en este caso no ataca y recibe 30 puntos de daño, solamente puede usar su habilidad) o es un duende que tiene su habilidad (bloqueo) activada. También hay un método igual pero sin los JOptionPanes porque Manu me pidió que lo metiese. También tiene un método habilidad que está vacío para que las subclases lo sobrescriban.

Después van los getters y los setters y por último los métodos ToString y Equals, éstos se llaman cuando los personajes se crean y el ToString le dicen al jugador la clase y la herramienta de cada personaje en el momento de su creación y el Equals notifica al usuario de que ya se ha creado un personaje igual al que acaba de meter, no te lo impide ni nada. El CompareTo no está porque no se me ocurrió ninguna forma de implementarlo que tuviese sentido.

Caballero, Mago, Bandido y Duende:

```java
package CurroFinal;
import javax.swing.JOptionPane;

public class Caballero extends Personaje {

    public Caballero(String herramienta, int i) {
        this.setClase("caballero");
        this.setVida(100);
        this.setHerramienta(herramienta);

        switch(herramienta) {
        case "hacha":
            this.setDaño(50);
            break;
        case "espada":
            this.setDaño(40);
            break;
        case "lanza":
            this.setDaño(30);
            break;
        case "cuchillo":
            this.setDaño(20);
        }
    }

    public void ataque(int numJugador, int i) {     // El ataque básico del caballero es distinto según si el ataque es o no cargado
        if (!this.getBolaDeFuego() && !(CrearPersonajes.personajes.get(numJugador).getClase() == "duende" && CrearPersonajes.personajes.get(numJugador).getHabilidadAct
            try {
                int dañoCargado = this.getDaño() * 2;
                if (this.getHabilidadActivada()) {          // Esto se ejecuta si el ataque está cargado (hace daño doble)
                    CrearPersonajes.personajes.get(numJugador).setVida(CrearPersonajes.personajes.get(numJugador).getVida() - dañoCargado);
                    JOptionPane.showMessageDialog(null, "El jugador " + i + " (caballero) ha atacado al jugador " + numJugador + " con un daño de " + (this.getDaño() *
                    this.setHabilidadActivada(false);
                } else {                                    // Esto se ejecuta si no está cargado
                    CrearPersonajes.personajes.get(numJugador).setVida(CrearPersonajes.personajes.get(numJugador).getVida() - this.getDaño());
                    JOptionPane.showMessageDialog(null, "El jugador " + i + " (caballero) ha atacado al jugador " + numJugador + " con un daño de " + this.getDaño() +
                }
            } catch (NullPointerException e) {
                JOptionPane.showMessageDialog(null, "Has intentado atacarle a un cadáver, por lo que se te saltará el turno");
            }
        } else if (this.getBolaDeFuego()) {         // Si el jugador está quemado, el ataque no surte efecto y recibes daño
            this.setVida(this.getVida()-30);
            JOptionPane.showMessageDialog(null, "Has intentado atacar estando quemado, por lo que recibes 30 de daño. Te queda " + this.getVida() + " de vida");
        } else if (CrearPersonajes.personajes.get(numJugador).getClase() == "duende" && CrearPersonajes.personajes.get(numJugador).getHabilidadActivada()) {
            JOptionPane.showMessageDialog(null, "Has elegido atacarle a un duende usando su escudo, por lo que tu ataque ha sido en vano");
        }
    }

    public void habilidad(int i) {
        JOptionPane.showMessageDialog(null, "El jugador " + i + " (caballero) ha cargado su ataque y hará el doble de daño en el siguiente turno");
        this.setHabilidadActivada(true);
        this.setBolaDeFuego(false);
    }
}
```

(Pongo solo la clase Caballero porque ponerlas todas ocuparía demasiado)

Estas son las clases que se usan para cada tipo de personaje. En su constructor se les asigna el nombre de la clase y la vida según a qué clase se haya llamado, y la herramienta según cuál sea el segundo argumento del constructor mediante un switch. Todos los personajes tienen un método habilidad que sobrescribe al método vacío de la clase Personaje, el del caballero hace que el ataque que hagas en el siguiente turno sea un ataque cargado que haga el doble de daño, el del mago lanza una bola de fuego que hace daño e impide a la víctima atacar en el siguiente turno y recibe daño si lo intenta, el del bandido hace un poco de daño a todos los jugadores que no estén muertos (incluyendo a los miembros de tu dúo o squad) y el del duende hace que saque un escudo que le proteja totalmente de daño en lo que queda

de turno. El único que tiene un método de ataque que sobrescriba al de la clase Personaje es el caballero porque su ataque es distinto según si está cargado o no.

Luego el resto de cosas las hicieron los demás, Jaime hizo el menú y el modo solo porque es el más fácil de hacer, Manu hizo el modo dúo y Antonio hizo el modo squad, pero no entraré en detalles porque eso saldrá en sus memorias.