

# UNIVERSIDAD TECMILENIO

**Mauricio Estrada De la Garza - AL02976904**

**Luis Roberto Reyes Buxadé - AL02922638**

**Jhonn Cristopher Yap Garcia - AL02939706**

**23/11/2023**

**Programación Orientada a Objetos**

**Reto Final**

**Reto Final: Implementación FINAL de los conceptos fundamentales de Programación en Python.**

**Objetivo de la actividad:**

**El alumno desarrollará un programa en el cual se implementa los conceptos:**

- 1. Uso correcto de variables y manipulación de datos.**
- 2. Uso correcto de datos de entrada / salida (input/print).**
- 3. Creación y uso correcto de colecciones.**
- 4. Creación y uso correcto de Funciones.**
- 5. Implementación y uso correcto de excepciones**
- 6. Uso correcto de condicionales.**
- 7. Uso correcto de ciclos.**
- 8. Uso correcto de flujos de ejecución.**
- 9. Diseño y creación de clases.**
- 10. Creación de objetos en base a las clases diseñadas.**
- 11. Manejo de constructores.**
- 12. Manejo de Herencia**
- 13. Interacción con el usuario.**
- 14. Documentación precisa y concreta dentro del programa.**

**Requerimientos para la actividad:**

**Cualquier IDE que verifique la sintaxis de python.**

**Instrucciones para el alumno:**

**Un juego es una excelente manera de aplicar todos esos conceptos. El objetivo es crear un juego de combate de héroes. Se podría tener diferentes**

**clases de héroes** (clases en términos de personajes, no de programación), cada uno con **habilidades únicas y estadísticas**. Podrían **combatir entre ellos** o **contra enemigos generados aleatoriamente**.

#### Clases:

1. **Héroe:** Una clase base que contendrá los atributos y métodos comunes a todos los héroes. Podrías definir aquí características como nombre, salud, ataque, defensa, etc.
2. **Clases Específicas de Héroe:** Derivadas de la clase base Héroe, podrías tener clases específicas como **Guerrero, Mago, Arquero**, cada una con **habilidades y características únicas**.

#### Funciones o Métodos:

- **Combate:** Puedes tener métodos en la clase Héroe para simular el combate entre dos héroes o entre un héroe y un enemigo.
- **Interacción con el usuario:** Utiliza la entrada y salida de datos para que el usuario pueda seleccionar acciones para su héroe, como atacar, defender, usar habilidades, etc.

#### Librerías:

- **Random:** Úsala para generar enemigos aleatorios, determinar el éxito de los ataques, la cantidad de daño infligido, etc.

#### Conceptos a calificar:

Todos los marcados en el objetivo de la actividad.

1. Todos los marcados en el objetivo de la actividad.
2. Manual de usuario, explicación detalla del juego
3. Reflexión: Discutir sobre la flexibilidad de objetos y clases en python versus otros lenguajes de programación.

#### Entregables:

- **Código Fuente**, archivos con extensión .py (Recuerda, cualquier otro tipo de archivo no es permitido, i.e. .pyd, .class, .zip, .jar, etc)

- **Compilación sin errores.**
- **Cumplimiento de Especificaciones.**
- **Reporte con las pantallas de ejecución respectivas por cada rubro.**
- **!!!! Reflexión individual por integrante del equipo. !!!!!**

**Importante:** Solo se revisa la última entrega

**\* Sin reporte y/o código fuente, ningún porcentaje es considerado ⇒  
Calificación NE**

**Rúbrica:**

- **Presentación/Demo presencial 15% y 100%\***
- **FrameWork 70%**
- **Reporte 15% y 100%\* (Incluida una reflexión individual por integrante del equipo)**

**Restricción:** Equipos de no más de cuatro personas, no menos de dos. 😊

***Condiciones para ser elegibles a una calificación:***

1. **Reporte de ejecución, con las conclusiones del ejercicio, es decir, cual es más eficiente, cuál es el menos eficiente, tablas comparativas entre los algoritmos.**
2. **Código en archivo de texto con extensión .java**
3. **En Equipos, no mas de cuatro, no menos de dos.**
4. **Presentación/Demo final el 1° de Diciembre (último día de clases) (presentador escogido al azar)**

**CÓDIGO:**

```
1 #Reto Final
2
3 #Juego del Señor de los Anillos
4
5 ...
6
7 Modos de Juego:
8
9 PvP entre personajes: Tipo Mortal Kombat
10
11 Survival: Enfrentamiento con Orcos (Tipo COD Zombies)
12
13 ...
14
15
16
17 import random
18 import time
19
20 class Heroe:
21     def __init__(self, nombre, salud, ataque, defensa):
22         self.nombre = nombre
23         self.salud = salud
24         self.ataque = ataque
25         self.defensa = defensa
26
27     def recibir_ataque(self, danio):
28         self.salud -= danio
29         print(f'{self.nombre} ha recibido {danio} puntos de daño. \nVida restante: {self.salud}')
30
31 class Arquero(Heroe):
32     def __init__(self, nombre, salud=100, ataque=20, defensa=10):
33         super().__init__(nombre, salud, ataque, defensa)
```

El Señor de los Anillos (GOTV)

Presiona 1 para comenzar  
Presiona 2 para salir

Selecciona una opcion: 2  
PS C:\Users\Wau>

```
class Arquero(Heroe):
    def __init__(self, nombre, salud=100, ataque=20, defensa=10):
        super().__init__(nombre, salud, ataque, defensa)

class Mago(Heroe):
    def __init__(self, nombre, salud=100, ataque=25, defensa=5):
        super().__init__(nombre, salud, ataque, defensa)

class Guerrero(Heroe):
    def __init__(self, nombre, salud=100, ataque=30, defensa=15):
        super().__init__(nombre, salud, ataque, defensa)

def line():
    print("_____")

def space():
    print("\n\n")

def charactersPVP():
    print("Selecciona dos personajes para la pelea:")
    print("1. Gandalf")
    print("2. Saruman")
    print("3. Legolas")
    print("4. Tauriel")
    print("5. Gimli")
    print("6. Aragorn")

    personajes_elegidos = []
    while len(personajes_elegidos) < 2:
```

Señor de los Anillos (GOTV)

Presiona 1 para comenzar  
Presiona 2 para salir

Selecciona una opcion: 2  
C:\Users\Wau>

```
61 personajes_elegidos = []
62 while len(personajes_elegidos) < 2:
63     opcion = int(input(f"Elige el personaje (len(personajes_elegidos) + 1): "))
64
65     if opcion == 1:
66         personajes_elegidos.append(Mago("Gandalf"))
67     elif opcion == 2:
68         personajes_elegidos.append(Mago("Saruman"))
69     elif opcion == 3:
70         personajes_elegidos.append(Arquero("Legolas"))
71     elif opcion == 4:
72         personajes_elegidos.append(Arquero("Tauriel"))
73     elif opcion == 5:
74         personajes_elegidos.append(Guerrero("Gimli"))
75     elif opcion == 6:
76         personajes_elegidos.append(Guerrero("Aragorn"))
77     else:
78         print("Selecciona una opción válida.")
79
80 return personajes_elegidos
81
82 > ...
83 > ...
84 > ...
122
123 def modePVP():
124     print("¿Que comience la pelea!")
125     jugadores = charactersPVP()
126
127     space()
128
129     jugador1, jugador2 = jugadores[0], jugadores[1]
130     while jugador1.salud > 0 and jugador2.salud > 0:
```

El Señor de los Anillos (GOTV)

Presiona 1 para comenzar  
Presiona 2 para salir

Selecciona una opción: 2  
PS C:\Users\Maya> [ ]

- Python
- Python
- Python
- Python
- Python
- Python
- Python
- Python
- Python
- Python

```
space()
jugador1, jugador2 = jugadores[0], jugadores[1]
while jugador1.salud > 0 and jugador2.salud > 0:
    input("Presiona Enter para que el Jugador 1 ataque...")
    danio_jugador1 = max(0, jugador1.ataque - jugador2.defensa)
    jugador2.salud -= danio_jugador1
    print(f"{jugador1.nombre} ataca a {jugador2.nombre} y hace {danio_jugador1} puntos de daño.")
    print(f"Vida de {jugador2.nombre}: {jugador2.salud}")

    if jugador2.salud <= 0:
        print(f"{jugador1.nombre} ha ganado la pelea.")
        break

    input("Presiona Enter para que el Jugador 2 ataque...")
    danio_jugador2 = max(0, jugador2.ataque - jugador1.defensa)
    jugador1.salud -= danio_jugador2
    print(f"{jugador2.nombre} ataca a {jugador1.nombre} y hace {danio_jugador2} puntos de daño.")
    print(f"Vida de {jugador1.nombre}: {jugador1.salud}")

    if jugador1.salud <= 0:
        print(f"{jugador2.nombre} ha ganado la pelea.")

space()
print("Quieres jugar otra vez?")
print("-Presiona 1 para continuar \n-Presiona 2 para salir al Menu")
gameOption = int(input("Elige una opcion: "))

if gameOption == 1:
    modePVP()
elif gameOption == 2:
    gameModes()
```

El Señor de los Anillos (GOTV)

Presiona 1 para comenzar  
Presiona 2 para salir

Selecciona una opcion: 2  
PS C:\Users\Wau> [ ]

Ln 24, Col 29 Spaces: 4 UTF-8 CRLF Python 3.11.2 64-bit 10:55 a. m. 01/12/2023

```
127 space()
128 jugador1, jugador2 = jugadores[0], jugadores[1]
129 while jugador1.salud > 0 and jugador2.salud > 0:
130     input("Presiona Enter para que el Jugador 1 ataque...")
131     danio_jugador1 = max(0, jugador1.ataque - jugador2.defensa)
132     jugador2.salud -= danio_jugador1
133     print(f"{jugador1.nombre} ataca a {jugador2.nombre} y hace {danio_jugador1} puntos de daño.")
134     print(f"Vida de {jugador2.nombre}: {jugador2.salud}")
135
136     if jugador2.salud <= 0:
137         print(f"{jugador1.nombre} ha ganado la pelea.")
138         break
139
140     input("Presiona Enter para que el Jugador 2 ataque...")
141     danio_jugador2 = max(0, jugador2.ataque - jugador1.defensa)
142     jugador1.salud -= danio_jugador2
143     print(f"{jugador2.nombre} ataca a {jugador1.nombre} y hace {danio_jugador2} puntos de daño.")
144     print(f"Vida de {jugador1.nombre}: {jugador1.salud}")
145
146     if jugador1.salud <= 0:
147         print(f"{jugador2.nombre} ha ganado la pelea.")
148
149 space()
150 print("Quieres jugar otra vez?")
151 print("-Presiona 1 para continuar \n-Presiona 2 para salir al Menu")
152 gameOption = int(input("Elige una opcion: "))
153
154 if gameOption == 1:
155     modePVP()
156 elif gameOption == 2:
157     gameModes()
158
159
```

El Señor de los Anillos (GOTV)

Presiona 1 para comenzar  
Presiona 2 para salir

Selecciona una opcion: 2  
PS C:\Users\Wau> [ ]

Ln 24, Col 29 Spaces: 4 UTF-8 CRLF Python 3.11.2 64-bit 10:55 a. m. 01/12/2023

```
163 def personajesOrcos():
164     print("Selecciona dos personajes para la pelea:")
165     print("1. Gandalf")
166     print("2. Saruman")
167     print("3. Legolas")
168     print("4. Tauriel")
169     print("5. Gimli")
170     print("6. Aragorn")
171
172     personajes_elegidos = []
173     while len(personajes_elegidos) < 1:
174         opcion = int(input(f"Elige el personaje {len(personajes_elegidos) + 1}: "))
175
176         if opcion == 1:
177             personajes_elegidos.append(Mago("Gandalf"))
178         elif opcion == 2:
179             personajes_elegidos.append(Mago("Saruman"))
180         elif opcion == 3:
181             personajes_elegidos.append(Arquero("Legolas"))
182         elif opcion == 4:
183             personajes_elegidos.append(Arquero("Tauriel"))
184         elif opcion == 5:
185             personajes_elegidos.append(Guerrero("Gimli"))
186         elif opcion == 6:
187             personajes_elegidos.append(Guerrero("Aragorn"))
188         else:
189             print("Selecciona una opción válida.")
190
191     return personajes_elegidos
192
193 def generar_orcos(cantidad):
194     orcos = []
195     for _ in range(cantidad):
196         orco = Orco()
197         orcos.append(orco)
198     return orcos
199
200 def jugar():
201     print("¡Bienvenido al juego de El Señor de los Anillos!")
202     print("Presiona 1 para comenzar")
203     print("Presiona 2 para salir")
204
205     while True:
206         opcion = int(input("Selecciona una opción: "))
207         if opcion == 1:
208             # Comenzar el juego
209             personajes_elegidos = personajesOrcos()
210             orcos = generar_orcos(5)
211             # Aquí iría la lógica de la pelea
212         elif opcion == 2:
213             # Salir del juego
214             break
215         else:
216             print("Opción no válida. Intenta de nuevo.")
217
218 if __name__ == "__main__":
219     jugar()
```

The image shows a Python IDE with a dark theme. The main editor window contains a Python script for a game titled 'El Señor de los Anillos (GOTY)'. The script is divided into several functions:   
1. `generar_orcos(cantidad)`: A function that generates a list of orcs. It uses `random.randint(5, 15)` to determine the number of orcs and `random.randint(5, 15)` to determine the damage each orc can deal.   
2. `enfrentar_orcos(jugador, orcos)`: A function that simulates a battle. It prints a message to prepare for battle, then enters a loop where the player can choose to fight orcs. If the player chooses to fight, the player's health is reduced by the damage of the orcs. If the player's health reaches 0, the game ends.   
3. `juego_orcos()`: The main game loop. It prints a message to prepare for the mission, creates a player character, and enters a loop where the player can choose to fight orcs. If the player chooses to fight, the `enfrentar_orcos` function is called.   
The terminal window at the bottom shows the output of the script. It displays the title 'El Señor de los Anillos (GOTY)', a prompt to press 1 to start or 2 to exit, and a selection screen where the player chooses an option. The terminal also shows the player's health and the damage of the orcs.   
On the right side of the IDE, there is a sidebar with a file explorer showing the project structure, which includes a `src` folder and a `main.py` file. The status bar at the bottom indicates the current line is 24, column 29, and the file is encoded in UTF-8 with CRLF line endings.

```
221 ronda = 1
222 while jugador.salud > 0:
223     print(f"\n--- Ronda {ronda} ---")
224     cantidad_orcos = random.randint(1, 5) # Número aleatorio de orcos por ronda
225     orcos = generar_orcos(cantidad_orcos)
226     resultado = enfrentar_orcos(jugador, orcos)
227     if not resultado:
228         break
229     ronda += 1
230     time.sleep(2) # Pausa entre rondas
231
232     space()
233     print("Quieres jugar otra vez?")
234     print("-Presiona 1 para continuar \n-Presiona 2 para salir al Menu")
235     gameOption = int(input("Elige una opcion: "))
236
237     if gameOption == 1:
238         juego_orcos()
239     elif gameOption == 2:
240         gameModes()
241
242 #Modos de Juego
243 def gameModes():
244
245     global gameOn
246     space()
247     print("Selecciona un modo de juego: ")
248     space()
249     print("-PVP (Presiona 1) \n-Survival (Presiona 2) \n-Regresar al menu principal (Presiona 3)")
250     space()
251     mode = int(input("Elige una opcion: "))
252
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

El Señor de los Anillos (GOTV)

Presiona 1 para comenzar  
Presiona 2 para salir

Selecciona una opcion: 2  
PS C:\Users\Maui> |

Ln 24, Col 29 Spaces: 4 UTF-8 CRLF Python 3.11.2 64-bit Go Live



```
252
253     if mode == 1:
254         space()
255         print("Bienvenido a PVP")
256         space()
257         modePVP()
258
259
260     elif mode == 2:
261         space()
262         print("Bienvenido a Survival!")
263         space()
264         juego_orcos()
265
266
267     elif mode == 3:
268         mainMenu()
269
270
271
272 # Menu Principal del Juego
273 def mainMenu():
274
275     global gameOn
276
277     line()
278     space()
279     print("El Señor de los Anillos (GOTV)")
280     space()
281     print("Presiona 1 para comenzar")
282     print("Presiona 2 para salir ")
283     space()
284
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

-----

El Señor de los Anillos (GOTV)

Presiona 1 para comenzar  
Presiona 2 para salir

Selecciona una opcion: 2  
PS C:\Users\Wau>

Ln 24, Col 29 Spaces: 4 UTF-8 CRLF Python 3.11.2 64-bit Go Live

```
284     space()
285
286     start = int(input("Selecciona una opcion: "))
287
288     if start == 1:
289         space()
290         gameModes()
291
292     elif start == 2:
293         gameOn = False
294
295
296 #Codigo MAIN
297
298 gameOn = True
299
300 while(gameOn):
301     mainMenu()
302
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

-----

El Señor de los Anillos (GOTV)

Presiona 1 para comenzar  
Presiona 2 para salir

Selecciona una opcion: 2  
PS C:\Users\Wau>

Ln 24, Col 29 Spaces: 4 UTF-8 CRLF Python 3.11.2 64-bit Go Live

PRUEBAS:

```
El Señor de los Anillos (GOTV)

Presiona 1 para comenzar
Presiona 2 para salir

Selecciona una opción: 1
```

```
Selecciona un modo de juego:

-PVP (Presiona 1)
-Survival (Presiona 2)
-Regresar al menu principal (Presiona 3)

Elige una opción: 1
```

```
[Que comience la pelea!
Selecciona dos personajes para la pelea:
1. Gandalf
2. Saruman
3. Legolas
4. Tauriel
5. Gimli
6. Aragorn
Elige el personaje 1: 1
Elige el personaje 2: 2

Presiona Enter para que el Jugador 1 ataque...
```

```
Presiona Enter para que el Jugador 1 ataque...
Gandalf ataca a Saruman y hace 20 puntos de daño.
Vida de Saruman: 80
Presiona Enter para que el Jugador 2 ataque...
Saruman ataca a Gandalf y hace 20 puntos de daño.
Vida de Gandalf: 80
Presiona Enter para que el Jugador 1 ataque...
Gandalf ataca a Saruman y hace 20 puntos de daño.
Vida de Saruman: 60
Presiona Enter para que el Jugador 2 ataque...
Saruman ataca a Gandalf y hace 20 puntos de daño.
Vida de Gandalf: 60
Presiona Enter para que el Jugador 1 ataque...
```

```
Presiona Enter para que el Jugador 2 ataque...
Saruman ataca a Gandalf y hace 20 puntos de daño.
Vida de Gandalf: 40
Presiona Enter para que el Jugador 1 ataque...
Gandalf ataca a Saruman y hace 20 puntos de daño.
Vida de Saruman: 20
Presiona Enter para que el Jugador 2 ataque...
Saruman ataca a Gandalf y hace 20 puntos de daño.
Vida de Gandalf: 20
Presiona Enter para que el Jugador 1 ataque...
Gandalf ataca a Saruman y hace 20 puntos de daño.
Vida de Saruman: 0
Gandalf ha ganado la pelea.
```

```
Gandalf ha ganado la pelea.

Quieres jugar otra vez?
-Presiona 1 para continuar
-Presiona 2 para salir al Menu
Elige una opción: 2

Selecciona un modo de juego:
```

```
-PVP (Presiona 1)
-Survival (Presiona 2)
-Regresar al menu principal (Presiona 3)

Elige una opción: 2

Bienvenido a Survival!
```

```
Selecciona dos personajes para la pelea:
1. Gandalf
2. Saruman
3. Legolas
4. Tauriel
5. Gimli
6. Aragorn
Elige el personaje 1: 5

--- Ronda 1 ---
¡Prepárate para la batalla contra los orcos!
Presiona Enter para enfrentar al siguiente orco...
```

```
--- Ronda 1 ---
¡Prepárate para la batalla contra los orcos!
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 8 puntos de daño.
Vida restante: 92
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 12 puntos de daño.
Vida restante: 80
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 11 puntos de daño.
Vida restante: 69

--- Ronda 2 ---
¡Prepárate para la batalla contra los orcos!
```

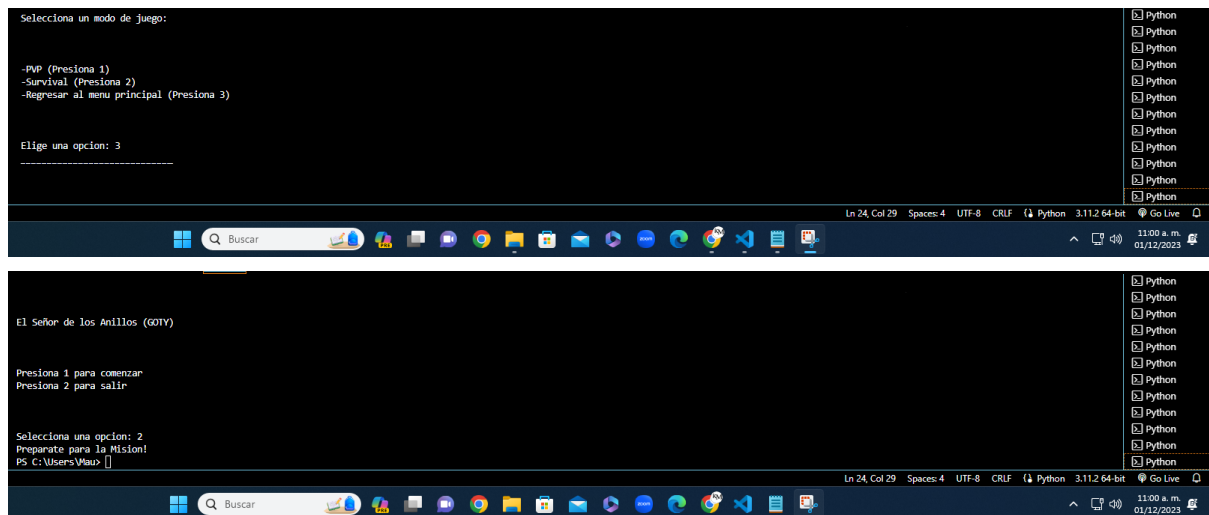
```
--- Ronda 2 ---
¡Prepárate para la batalla contra los orcos!
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 5 puntos de daño.
Vida restante: 64
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 13 puntos de daño.
Vida restante: 51
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 5 puntos de daño.
Vida restante: 46
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 10 puntos de daño.
Vida restante: 36
```

```
--- Ronda 3 ---
¡Prepárate para la batalla contra los orcos!
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 7 puntos de daño.
Vida restante: 14
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 6 puntos de daño.
Vida restante: 8
Presiona Enter para enfrentar al siguiente orco...
Gimli ha recibido 10 puntos de daño.
Vida restante: -2
¡El jugador ha sido derrotado!
```

```
Vida restante: -2
¡El jugador ha sido derrotado!

Quieres jugar otra vez?
-Presiona 1 para continuar
-Presiona 2 para salir al Menu
Elige una opción: 2

Selecciona un modo de juego:
```



## REPORTE:

La temática del juego es del señor de los anillos, implementamos a los personajes principales de la saga y además hay 2 modos de juego, PVP que es un modo de juego sobre pelea tipo Mortal Kombat donde el jugador tiene que elegir 2 personajes del juego y el modo Survival, donde el jugador elegirá un personaje del juego y se enfrenta a una ola de orcos donde tendrá que sobrevivir por rondas, algo similar a los zombies de COD. Lo primero que hicimos fue importar las librerías random y time, mas que todo se usarán para el modo de juego de Survival donde se generarán orcos aleatoriamente en cierto tiempo determinado. La clase principal es la de Héroe, los atributos son (nombre, salud, ataque y defensa) y hay un método para recibir ataques, donde mientras el jugador reciba más daño, va disminuyendo su salud que es 100 en todas las subclases, luego está la subclase de arquero donde se incluyen los personajes de Legolas y Tauriel (Tauriel es personaje de las películas del Hobbit, pero igual decidimos incluirla para no solamente tener a Legolas como único Arquero) en la subclase de Mago estarán Gandalf y Saruman. Y por último en la clase de guerrero estarán Gimli y Aragorn. Hicimos 1 función para elegir personajes en cada modo de juego, aquí va la diferencia: para el modo PVP, el jugador tiene que elegir 2 personajes, aparecen todos los personajes en pantalla y se guardarán en una lista, mientras no se elijan 2 personajes no se procederá a nada por el ciclo WHILE, entonces una vez el usuario elija un personaje se agrega a la lista de personajes elegidos, y ya que se elijan todos los personajes, ahora sí se puede proceder a lo siguiente, y para el modo survival únicamente es de elegir 1 solo personaje, y listo! pero están codificados de la misma forma tal cual. Luego así se codificó el modo PVP, hicimos una variable llamada "jugadores" donde equivale a la función de elegir personajes en PVP, entonces la variable jugadores es igual a la lista de personajes elegidos en la función, entonces luego las variables de jugador 1 y 2, es igual a la posición 0 y 1 de la lista, o sea los 2 personajes que el usuario eligió y guardo en la lista. entonces el juego está codificado de esta forma, con un ciclo while y un AND, mientras la salud del jugador 1 y 2 sea mayor a 0, con un input el usuario le presiona enter para continuar el combate y se va disminuyendo la salud de los jugadores, ademas que sale la cantidad de daño que los jugadores se han infligido y cuanta salud les

queda, con una condicional mientras la salud de cualquiera de los jugadores es menor a 0, pierde la pelea, solo uno de los 2 es ganador. y al final hay una opción para volver a jugar, si el usuario quiere volver a jugar, presiona 1 y como mandamos a llamar a llamar otra vez la función del juego, se vuelve a repetir el modo. de lo contrario lo lleva al menú de los modos de juego. Luego para el modo Survival tuvimos que hacer 3 funciones, la primer función es para generar orcos donde pasamos un parámetro llamado "cantidad" creamos una lista de orcos, y luego con un ciclo FOR pasando el parámetro de la función, el daño que harían los orcos sería de 5 a 15 puntos de vida, y agregamos esa variable a la lista de orcos, y al final de la función que nos regresa la lista de orcos. Después la segunda función es para enfrentar orcos, pasamos como parámetros la variable jugador y orcos, entonces con un ciclo FOR mandando a llamar el daño de orcos y la variable "orcos" el jugador presiona la tecla enter por la función input, y después mandamos a llamar el método de recibir ataque pasándole el parámetro de daño de orcos. y con una condicional, mientras la salud del jugador sea menor a 0, entonces el jugador pierde y se regresa un False, además que hay lapso de tiempo a esperar de 1 segundo para los ataques de los orcos. y regresa un True. Ya la última función es del juego principal de los orcos, para el tema de elegir personajes lo codificamos igual que en el modo PVP, luego definimos la variable ronda como 1, con un ciclo WHILE mientras la salud del jugador fuese mayor a 0, seguiria corriendo el juego. Se refleja la ronda actual. la variable "cantidad orcos" es lo que estará generando los orcos aleatoriamente, de 1 a 5 orcos por ronda. y después Definimos la variable orcos que fuera igual a la función de generar orcos pasándole como parámetro la variable "cantidad orcos", y las rondas van subiendo 1 a 1. y el tiempo entre rondas es de 2 segundos. Al igual que en el modo PVP, hay una opción para seguir jugando o no. La función de los modos de juego está codificado de esta forma, mandamos a llamar el ciclo WHILE del main en el programa, ahí pusimos como dato de salida los diferentes modos que puede jugar el usuario, y con las condicionales, el primer caso es para jugar PVP y el segundo caso para jugar Survival, la tercera opción es para ir al menú principal del juego. Ya únicamente la función del menú principal dice el título del juego, y las opciones para empezar a jugar o salir del juego. Ya únicamente en el main definimos la variable booleana para el ciclo While, y mandamos a llamar la funcion del menu principal, de simplemente 3 lineas de codigos hay detras cientos de lineas de codigo literalmente!

## CONCLUSIÓN:

**Mau:** Estuvo bastante complicado hacer este reto, pero fue divertido hacer un programa sobre un juego sencillo a simple vista, la mayoría de gente lo vería como algo muy simple pero no saben que aun asi toma su tiempo codificarlo, ya veo porque los videojuegos indies o los AAA son bastantes complicados de hacer.

**Luis:** Estuvo divertido hacerlo de temática del señor de los anillos, a mi se me ocurrió esa idea ya que me gustan mucho las películas. Además también me gusta jugar Mortal Kombat y los zombies de COD, quisimos hacer algo que estuviera acorde al proyecto y qué mejor idea que hacerlo de esta forma.

**Cris:** Pienso lo mismo que Mau, la verdad si fue algo complicado hacer este proyecto, pero gracias a que trabajamos en equipo todo se pudo hacer más simple, parte de la estructura del código está más organizado ya que estuve viendo cómo podemos conectar todas las funciones en unas simples líneas de código.