

FITXA D'ACTIVITAT		
CURS: 2022-23	DATA: 13/01/2023 Nombre: Daniel Bernaus Pérez	GRUP: 1DAW/1DAM/1ASIX
PROFESSOR/A:	Javier Salvador Carmen Quintás	
MÒDUL/UF:	MP05/UF1-M16/UF2	
TÍTOL		
ACTIVITAT:	Pr4: Refactorización de código	

1. Documento PDF explicando todos los cambios realizados a nivel de código/funciones

A continuación, mostraremos imágenes con los cambios realizados al código

```
int heroHP = 350;
```

Daniel Bernaus

Cambiamos el atributo de vida del héroe para que la pelea se alargue un poco más.

```
void checkEnemys(string name, int& eneHp, bool& enemyLive) {
    if (eneHp <= 0) {
        cout << "Te has cargado al enemigo " << name << " \n";
        eneHp = 0;
        enemyLive = false;
    }
    else {
        cout << "El enemigo " << name << " tiene " << eneHp << " puntos de vida\n";
        enemyLive = true;
    }
}
```

Daniel Bernaus

Cambiamos la llamada a la función por valor, a una función de llamada "Void" de nombre "checkEnemys" con parámetros por referencia, introduciendo atributos como:

- string** para los nombres de los enemigos. Este atributo lo pasamos con una función de llamada por valor, ya que los nombres de los enemigos no cambian.
- atributo **int** con el símbolo & indicando que va a ser por referencia la vida de los enemigos.
- Booleano** que también pasaremos por referencia para indicar si el enemigo está vivo o muerto.

Ahora que tenemos nuestra función por referencia, podemos llamarla desde nuestro main y el resultado queda tal que así:

FITXA D'ACTIVITAT		
CURS: 2022-23	DATA: 13/01/2023 Nombre: Daniel Bernaus Pérez	GRUP: 1DAW/1DAM/1ASIX
PROFESSOR/A:	Javier Salvador Carmen Quintás	
MÒDUL/UF:	MP05/UF1-M16/UF2	
TÍTOL		
ACTIVITAT:	Pr4: Refactorización de código	

```
int main()
{
    gameStart();
    while (heroIsAlive && (enemyIsAlive1 || enemyIsAlive2)) {
        //ELEGIMOS ENEMIGO AL QUE ATACAR
        enemyChooosed = chooseEnemy();
        //ELEGIMOS ATAQUE Y CAPTURAMOS EL VALOR DEL DANYO QUE HAREMOS
        heroDamage = chooseAttack();

        if (enemyChooosed == 1) {
            //ATACAMOS AL ENEMIGO 1
            heroAttack(enemyName1, heroDamage, enemyHP1);
            checkEnemy(enemyName1, enemyHP1, enemyIsAlive1);
        }
        else {
            //ATACAMOS AL ENEMIGO 2
            heroAttack(enemyName2, heroDamage, enemyHP2);
            checkEnemy(enemyName2, enemyHP2, enemyIsAlive2);
        }

        //ATACA EL ENEMIGO 1 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive1 && heroIsAlive) {
            enemyAttack(enemyName1, 30 + rand() % 20);
        }
        //ATACA EL ENEMIGO 2 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive2 && heroIsAlive) {
            enemyAttack(enemyName2, 30 + rand() % 20);
        }
    }
}
```

Daniel Bernaus

Dentro de nuestro main, pondremos el nombre de nuestra función, además de pasar los valores de cada uno de nuestros enemigos.

Estos atributos se encuentran antes del main para poderlos pasar luego por referencia, tanto como el enemigo 1 o el enemigo 2 como es el caso.

```
#include <iostream>

using namespace std;

//Atributos del enemigo
int enemyHP1 = 100;
string enemyName1 = "Zombie";
bool enemyIsAlive1 = true;

//Atributos del enemigo 2
int enemyHP2 = 150;
string enemyName2 = "Creeper";
bool enemyIsAlive2 = true;
```

Daniel Bernaus

Seguimos introduciendo valores por referencia, y así optimizar nuestro código

FITXA D'ACTIVITAT		
CURS: 2022-23	DATA: 13/01/2023 Nombre: Daniel Bernaus Pérez	GRUP: 1DAW/1DAM/1ASIX
PROFESSOR/A:	Javier Salvador Carmen Quintás	
MÒDUL/UF:	MP05/UF1-M16/UF2	
TÍTOL		
ACTIVITAT:	Pr4: Refactorización de código	

```
void heroAttack(string name,int damage, int& Life) {
    cout << "Atacas al enemigo " << name << " y le haces un danyo de " << damage << " \n";
    Life = Life - damage;
}
```

Daniel Bernaus

Llamaremos a esta función desde nuestro main con un void. Dentro de la función dejaremos los atributos por valor tales como el nombre de los enemigos y el daño que hace nuestro héroe. Pasaremos por referencia la vida de los enemigos como muestra la imagen.

```
int main()
{
    gameStart();
    while (heroIsAlive && (enemyIsAlive1 || enemyIsAlive2)) {
        //ELEGIMOS ENEMIGO AL QUE ATACAR
        enemyChosed = chooseEnemy();
        //ELEGIMOS ATAQUE Y CAPTURAMOS EL VALOR DEL DANYO QUE HAREMOS
        heroDamage = chooseAttack();

        if (enemyChosed == 1) {
            //ATACAMOS AL ENEMIGO 1
            heroAttack(enemyName1, heroDamage, enemyHP1);
            checkEnemy(enemyName1, enemyHP1, enemyIsAlive1);
        }
        else {
            //ATACAMOS AL ENEMIGO 2
            heroAttack(enemyName2, heroDamage, enemyHP2);
            checkEnemy(enemyName2, enemyHP2, enemyIsAlive2);
        }

        //ATACA EL ENEMIGO 1 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive1 && heroIsAlive) {
            enemyAttack(enemyName1, 30 + rand() % 20);
        }
        //ATACA EL ENEMIGO 2 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive2 && heroIsAlive) {
            enemyAttack(enemyName2, 30 + rand() % 20);
        }
    }
}
```

Daniel Bernaus

Dentro del main haremos los cambios introduciendo el nombre de nuestra función. Introduciendo los nombres del enemigo, el daño del héroe y la vida del enemigo después de recibir los ataques.

FITXA D'ACTIVITAT		
CURS: 2022-23	DATA: 13/01/2023 Nombre: Daniel Bernaus Pérez	GRUP: 1DAW/1DAM/1ASIX
PROFESSOR/A:	Javier Salvador Carmen Quintás	
MÒDUL/UF:	MP05/UF1-M16/UF2	
TÍTOL		
ACTIVITAT:	Pr4: Refactorización de código	

```
void enemyAttack(string name,int damage) {
    heroHP = heroHP - damage;
    if (heroHP > 0) {
        cout << "El enemigo " << name << " te ha atacado y te quedan " << heroHP << " puntos de vida\n";
    }
    else {
        cout << "El enemigo " << name << " te ha atacado y te ha matado. El mundo esta condenado. \n";
        heroIsAlive = false;
    }
}
```

Daniel Bernaus

Como observamos en la imagen no hay ningún valor por referencia, pero si que hemos cambiado la llamada a la función por un tipo void. Dentro de la función especificamos los parámetros string nombre y daño de los enemigos, permitiendo ahorrar líneas de código. El resultado queda así dentro de nuestro main:

```
int main()
{
    gameStart();
    while (heroIsAlive && (enemyIsAlive1 || enemyIsAlive2)) {
        //ELEGIMOS ENEMIGO AL QUE ATACAR
        enemyChosed = chooseEnemy();
        //ELEGIMOS ATAQUE Y CAPTURAMOS EL VALOR DEL DANYO QUE HAREMOS
        heroDamage = chooseAttack();

        if (enemyChosed == 1) {
            //ATACAMOS AL ENEMIGO 1
            heroAttack(enemyName1, heroDamage, enemyHP1);
            checkEnemys(enemyName1, enemyHP1, enemyIsAlive1);
        }
        else {
            //ATACAMOS AL ENEMIGO 2
            heroAttack(enemyName2, heroDamage, enemyHP2);
            checkEnemys(enemyName2, enemyHP2, enemyIsAlive2);
        }

        //ATACA EL ENEMIGO 1 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive1 && heroIsAlive) {
            enemyAttack(enemyName1, 30 + rand() % 20);
        }
        //ATACA EL ENEMIGO 2 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive2 && heroIsAlive) {
            enemyAttack(enemyName2, 30 + rand() % 20);
        }
    }
}
```

Daniel Bernaus

Llamando a nuestra función por referencia, el código nos queda más simplificado y más legible.