

Nombre: Angel Jacinto \_\_\_\_\_ No. de Matrícula.: \_\_ZAP  
534 \_\_\_\_\_

Materia: \_Fundamentos de programación \_\_\_\_\_ Grupo: DSI 1 SEMESTRE Turno: M\_\_\_\_\_

Carrera: Ingeniería en desarrollo de software interactivos y videojuegos \_\_\_\_\_

Tema: practica 5 No: R.1

Fecha propuesta: martes, 12 de marzo de 2022 Fecha de Entrega: domingo 8 de mayo de 2022

Escuela: Amerike Plantel GDL

Calle: \_Montemorelos\_\_\_\_\_ No: 3503 Colonia: Rinconada de la Calma. \_\_\_\_\_C.P.: 45080\_\_\_\_\_

Teléfono: 3336326100

Ciudad: Guadalajara



\_\_\_\_\_**Angel Jacinto**\_\_\_\_\_

Qué se evalúa:	10 pts.	7 pts.	4pts.	Pts.
Entrega electrónica	Es en tiempo y forma al iniciar la clase. (1 pts.)	Después de 30 minutos de iniciada la clase. (.7 pts.)	Al minuto 40. (Posteriormente ya no se reciben) (.4pts.)	
Del formato.	Cumple con todos los elementos solicitados. (1 pts.)	No cumple con dos elementos solicitados. (.7 pts.)	No cumple con tres o más elementos solicitados. (.4pts.)	
La ortografía.	Tiene dos errores ortográficos. (1 pts.)	Tiene de tres a cuatro errores ortográficos. (.7 pts.)	Tiene cinco o más errores ortográficos. (.4pts.)	
Del tema y objetivo.	La teoría y ejemplos corresponden al tema tratado. (1 pts.)	La teoría o ejemplos no corresponden al tema tratado. (.7 pts.)	La teoría y ejemplos no corresponden al tema tratado. (.4pts.)	
El programa y los cálculos.	Los parámetros y componentes corresponden al 100% de lo planeado. (1 pts.)	El programa arroja un error o componente no corresponden al 100% de lo planeado. (7 pts.)	El programa arroja dos errores o componentes no corresponden al 100% de lo calculado. (.4pts.)	
Diagramas.	Los diagramas a bloques, de flujo y esquemáticos son acorde al de la práctica y siguen una secuencia lógica. (1 pts.)	Los diagramas a bloques, o de flujo o esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.7 pts.)	Los diagramas a bloques, de flujo y esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.4pts.)	
La tabla de valores.	Los valores calculados y medidos presentan una desviación máxima del 10%. (1 pts.)	Los valores calculados y medidos presentan una desviación máxima del 15%. (.7 pts.)	Los valores calculados y medidos presentan una desviación máxima del 20%. (.4pts.)	
Las observaciones y conclusiones.	Son específicas y congruentes con la práctica. (1 pts.)	Las observaciones o conclusiones son específicas y congruentes con la práctica. (.7 pts.)	Las observaciones y las conclusiones no son específicas y congruentes con la práctica. (.4pts.)	
Bibliografía.	Es acorde al (los) tema (s) tratado (s) y está completa (1 pts.)	Es acorde a algún (os) tema (s) tratado (s), le falta algún elemento que la conforman (.7 pts.)	No es acorde al (los) tema (s) tratado (s), le faltan 2 elementos que la conforma (.4pts.)	
Fuentes de consulta.	Es acorde al (los) tema (s) tratado (s) (1 pts.)	Es acorde a algún (os) tema (s) tratado (s) (.7 pts.)	Es acorde a algún (los) tema (s) tratado (s) (.4pts.)	

```
#include<iostream>;
#include<cstdlib>;
#include<ctime>;
#include<string>;

using namespace std;
int main()
{
    srand(time(nullptr));

    int P_lifePoints = 100;
    int CH_LifePoints = 100;
    int PICKACHU = 1;
    int CHARIZARD = 2;
    int selection = PICKACHU || CHARIZARD;
    cout << "you're about to battle, choose a character" << endl;
    cout << "1. Pikachu 2.Charizard " << endl << endl << endl;
    cin >> selection;

    switch (selection)
    {
    case 1:
        cout << "you've choosen Pickachu as pokemon" << endl;
        break;

    case 2:
        cout << "you've choosen Charizard as pokemon" << endl;
        break;

    }
    if (selection == 1)
    {
        cout << "|_____|" << endl;
        cout << "LET'S BATTLE" << endl << endl << endl;

        // PICKACHU////////////////////////////////////
        do
        {
            int PICKACHU_RANDOM = rand()%50;
            int PICKACHU_TURN;
            cout << "                " << "--PICKACHU'S TURN--" << "                " << endl;
            cout << "PICKACHU: " << "[" << P_lifePoints << "]" << " CHARIZARD: [" << CH_LifePoints << "]"
            << endl;
```

```

cout << "1-[ATACAR]. 2-[DEFENDER]." << endl;
cin >> PICKACHU_TURN;
cout << endl << endl;

switch (PICKACHU_TURN)
{
case 1:
cout << endl << endl;
cout << "Pickachu ataca" << " " << " " << " " << PICKACHU_RANDOM << endl;
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << " " << "CHARIZARD: [" << (CH_LifePoints-
PICKACHU_RANDOM) << "]" << endl;
cout << endl << endl;
cout << " " << endl;
CH_LifePoints = (CH_LifePoints - PICKACHU_RANDOM);
break;

case 2:
cout << endl << endl;
cout << " " << endl;
cout << "Pikachu defiende" << " " << " " << "+" << PICKACHU_RANDOM << endl;
cout << "PICKACHU: " << "[" << (P_lifePoints+PICKACHU_RANDOM) << "]" << " " << "CHARIZARD: [" << CH_LifePoints << "]" << endl;
cout << endl << endl;
cout << " " << endl;
P_lifePoints = (P_lifePoints + PICKACHU_RANDOM);
break;

}
cout << endl << endl << endl;
system("pause");
system("CLS");
//CHARIZARD////////////////////////////////////

int CHARIZARD_RANDOM = rand() % 50+26;
int CHARIZARD_TURN = 0;
cout << " " << " " << "--CHARIZARD'S TURN--" << " " << endl;
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << " " << "CHARIZARD: [" << CH_LifePoints << "]" << endl;
cout << "1-[ATACAR]. 2-[DEFENDER]." << endl;
cin >> CHARIZARD_TURN;
cout << endl << endl;

```

```

switch (CHARIZARD_TURN)
{
case 1:
cout << endl;
cout << "Charizard ataca " << " " << "-" << CHARIZARD_RANDOM << endl;
cout << "PICKACHU: " << "[" << (P_lifePoints - CHARIZARD_RANDOM) << "]" << " CHARIZARD: [" << CH_LifePoints << "]" << endl;
cout << endl;
cout << " " << endl;
P_lifePoints = (P_lifePoints - CHARIZARD_RANDOM);
break;

case 2:
cout << endl;
cout << "Charizard defiende" << " " << "+" << CHARIZARD_RANDOM << endl;
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << " CHARIZARD: [" << (CH_LifePoints + CHARIZARD_RANDOM) << "]" << endl;
cout << endl;
cout << " " << endl;
CH_LifePoints = (CH_LifePoints + CHARIZARD_RANDOM);
break;

}

cout << endl << endl << endl;
system("pause");
system("CLS");

}while(CH_LifePoints >= 0 &&P_lifePoints>=0);
{
cout << "LAST PLAYER WIN" << endl;
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << " CHARIZARD: [" << CH_LifePoints << "]" << endl;
}

}

// CHARIZARD////////////////////////////////////
if (selection == 2)

```

```

{
cout << "|_____|" << endl;
cout << "LET'S BATTLE" << endl << endl << endl;

do
{
int PICKACHU_RANDOM = rand() % 50;
int PICKACHU_TURN;
cout << "          " << "--CHARIZARD'S TURN--" << "          " << endl;
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << "  CHARIZARD: [" << CH_LifePoints << "]"
<< endl;
cout << "1-[ATACAR]. 2-[DEFENDER]." << endl;
cin >> PICKACHU_TURN;
cout << endl << endl;

switch (PICKACHU_TURN)
{
case 1:
cout << "          " << endl;
cout << "Charizard ataca" << "          " << "-" << PICKACHU_RANDOM << endl;
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << "  CHARIZARD: [" << (CH_LifePoints -
PICKACHU_RANDOM) << "]" << endl;
cout << "          " << endl;
cout << "          " << endl;
CH_LifePoints = (CH_LifePoints - PICKACHU_RANDOM);
break;

case 2:
cout << "          " << endl;
cout << "Charizard defiende" << "          " << "+" << PICKACHU_RANDOM << endl;
cout << "PICKACHU: " << "[" << (P_lifePoints + PICKACHU_RANDOM) << "]" << "  CHARIZARD:
[" << CH_LifePoints << "]" << endl;
cout << "          " << endl;
cout << "          " << endl;
P_lifePoints = (P_lifePoints + PICKACHU_RANDOM);
break;

}
cout << endl << endl << endl;

```

```

system("pause");
system("CLS");
//CHARIZARD////////////////////////////////////

int CHARIZARD_RANDOM = rand() % 50 + 26;
int CHARIZARD_TURN = 0;
cout << "                " << "--PICKACHU'S TURN--" << "                " << endl;
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << " CHARIZARD: [" << CH_LifePoints << "]"
<< endl;
cout << "1-[ATACAR]. 2-[DEFENDER]." << endl;
cin >> CHARIZARD_TURN;
cout << endl << endl;
switch (CHARIZARD_TURN)
{
case 1:
cout << "                " << endl;
cout << "Pickachu ataca " << "                " << "-" << CHARIZARD_RANDOM << endl;
cout << "PICKACHU: " << "[" << (P_lifePoints - CHARIZARD_RANDOM) << "]" << " CHARIZARD: ["
<< CH_LifePoints << "]" << endl;
cout << "                " << endl;
cout << "                " << endl;
P_lifePoints = (P_lifePoints - CHARIZARD_RANDOM);
break;

case 2:
cout << "                " << endl;
cout << "Pickachu defiende" << "                " << "+" << CHARIZARD_RANDOM << endl;
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << " CHARIZARD: [" << (CH_LifePoints +
CHARIZARD_RANDOM) << "]" << endl;
cout << "                " << endl;
cout << "                " << endl;
CH_LifePoints = (CH_LifePoints + CHARIZARD_RANDOM);
break;

}

cout << endl << endl << endl;
system("pause");
system("CLS");

```

```
} while (CH_LifePoints >= 0 && P_lifePoints >= 0);  
{  
cout << "LAST PLAYER WIN" << endl;  
cout << "PICKACHU: " << "[" << P_lifePoints << "]" << "  CHARIZARD: [" << CH_LifePoints << "]"  
<< endl;  
}  
  
}  
return 0;  
}
```

## Observaciones

Al principio tuve dificultades porque intente crear una AI para que pueda retar a el jugador pero al final no funciono, entonces lo cree de 2 jugadores. Un jugador elige al principio cual Pokémon quiere utilizar y de allí empieza todo.

## Conclusiones

**Se establecen variables como siempre. Para las decisiones creé un sistema de un switch dentro de un do-while loop y dentro de una sentencia if.**

**Al principio hay un switch para la toma de decisión del Pokémon, seguido por una sentencia if de cada Pokémon (pickachu, charizard), seguido con un do-while loop la cual es la base de las decisiones, seguido por sentencias switch, las cuales son las decisiones. Al final el jugador que acabe con al vida de su enemigo es el que gana.**