

Nombre: Miguel Alejandro Santiago Pérez

No. de Matrícula.: zap362

Materia: Fundamentos de Programación

Grupo: Dev1

Turno: Matutino

Carrera: Desarrollo de Software Interactivo y Videojuegos

Tema: ENUM/CLASS/STRUCT

No: R.10

Fecha propuesta: 28/04/2021

Fecha de Entrega: 05/05/2021

Escuela: Instituto Universitario Amerike

Plantel Zapopan

Calle: Calle Montemorelos

No: 3503

Colonia: Rinconada de la Calma

C.P.: 45080

Teléfono: 3336326100

Ciudad: Zapopan



Firma del alumno (a)

Firma de revisión fecha

| 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.) | |

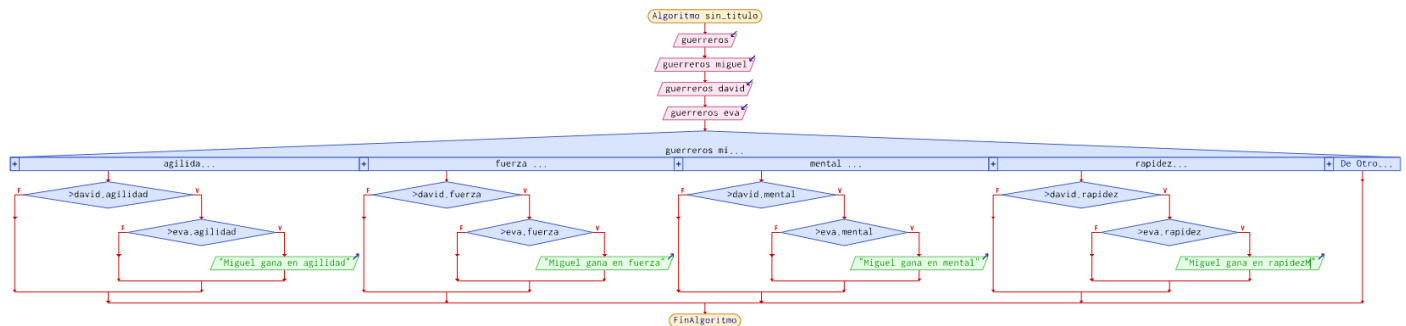
Índice

| | |
|------------------|------------------------------|
| Pagina 3 | Teoría y Cálculos |
| Paginas 4 – 13.. | Diagrama y Código |
| Pagina 14..... | Observaciones y conclusiones |

Teoría Al compilar el código de forma aleatoria se asignarán valores a las habilidades y después comprobaremos quien es el ganador

Cálculos Se agrega una variable de entero en forma de powerup que agrega un numero aleatorio entre 0 y 5 a los puntos de habilidad de los guerreros

Diagramas



```
#include <iostream>
```

```
#include <time.h>
```

```
using namespace std;
```

```
struct Guerreros
```

```
{
```

```
    int fuerza;
```

```
    int rapidez;
```

```
    int agilidad;
```

```
    int mental;
```

```
};
```

```
enum PowerUp;
```

```
int main()
```

```
{
```

```
    srand(time(NULL));
```

```
    int powerup = 0;
```

```
    Guerreros miguel;
```

```
    Guerreros david;
```

```
    Guerreros eva;
```

```
    Guerreros jesus;
```

```
    Guerreros antimo;
```

```
    powerup = rand() % 5;
```

```
    //Habilidades Miguel
```

```
    miguel.agilidad = rand() % 10 + powerup;
```

```
miguel.fuerza = rand() % 10 + powerup;
miguel.mental = rand() % 10 + powerup;
miguel.rapidez = rand() % 10 + powerup;
cout << "*****Habilidades Miguel*****" << endl;
cout << endl;
cout << "Agilidad = " << miguel.agilidad << endl;
cout << "Fuerza = " << miguel.fuerza << endl;
cout << "Mental = " << miguel.mental << endl;
cout << "Rapidez = " << miguel.rapidez << endl;
cout << endl;
cout << "*****" << endl;
cout << endl;
```

```
//Habilidades David
david.agilidad = rand() % 10 + powerup;
david.fuerza = rand() % 10 + powerup;
david.mental = rand() % 10 + powerup;
david.rapidez = rand() % 10 + powerup;
cout << "*****Habilidades David*****" << endl;
cout << endl;
cout << "Agilidad = " << david.agilidad << endl;
cout << "Fuerza = " << david.fuerza << endl;
cout << "Mental = " << david.mental << endl;
cout << "Rapidez = " << david.rapidez << endl;
cout << "*****" << endl;
cout << endl;
```

```
//Habilidades Eva
eva.agilidad = rand() % 10 + powerup;
eva.fuerza = rand() % 10 + powerup;
eva.mental = rand() % 10 + powerup;
eva.rapidez = rand() % 10 + powerup;
cout << "*****Habilidades Eva*****" << endl;
cout << endl;
cout << "Agilidad = " << eva.agilidad << endl;
cout << "Fuerza = " << eva.fuerza << endl;
cout << "Mental = " << eva.mental << endl;
cout << "Rapidez = " << eva.rapidez << endl;
cout << "*****" << endl;
cout << endl;
```

```
//Habilidades Jesus
jesus.agilidad = rand() % 10 + powerup;
jesus.fuerza = rand() % 10 + powerup;
jesus.mental = rand() % 10 + powerup;
jesus.rapidez = rand() % 10 + powerup;
```

```
cout << "*****Habilidades Jesus*****" << endl;
cout << endl;
cout << "Agilidad = " << jesus.agilidad << endl;
cout << "Fuerza = " << jesus.fuerza << endl;
cout << "Mental = " << jesus.mental << endl;
cout << "Rapidez = " << jesus.rapidez << endl;
cout << "*****" << endl;
cout << endl;
```

```
//Habilidades Antimo
```

```
antimo .agilidad = rand() % 10 + powerup;
antimo.fuerza = rand() % 10 + powerup;
antimo.mental = rand() % 10 + powerup;
antimo.rapidez = rand() % 10 + powerup;
cout << "*****Habilidades Antimo*****" << endl;
cout << endl;
cout << "Agilidad = " << antimo.agilidad << endl;
cout << "Fuerza = " << antimo.fuerza << endl;
cout << "Mental = " << antimo.mental << endl;
cout << "Rapidez = " << antimo.rapidez << endl;
cout << "*****" << endl;
cout << endl;
```

```
system("pause");
```

```
system("cls");
```

```
//Comprobacion Habilidades Miguel
```

```
//Comprobacion Agilidad
```

```
if (miguel.agilidad > david.agilidad)
{
    if (miguel.agilidad > eva.agilidad)
    {
        if (miguel.agilidad > jesus.agilidad)
        {
            if (miguel.agilidad > antimo.agilidad)
            {
                cout << "Miguel gana en agilidad" << endl;
            }
        }
    }
}
//Fin Agilidad
```

```
//Comprobacion Fuerza
```

```
if (miguel.fuerza > david.fuerza)
{
```

```
if (miguel.fuerza > eva.fuerza)
{
    if (miguel.fuerza > jesus.fuerza)
    {
        if (miguel.fuerza > antimo.fuerza)
        {
            cout << "Miguel gana en fuerza" << endl;
        }
    }
}
//Fin Fuerza

//Comprobacion Mental
if (miguel.mental > david.mental)
{
    if (miguel.mental > eva.mental)
    {
        if (miguel.mental > jesus.mental)
        {
            if (miguel.mental > antimo.mental)
            {
                cout << "Miguel gana en mental" << endl;
            }
        }
    }
}
//Fin mental

//Comprobacion Rapidez
if (miguel.rapidez > david.rapidez)
{
    if (miguel.rapidez > eva.rapidez)
    {
        if (miguel.rapidez > jesus.rapidez)
        {
            if (miguel.rapidez > antimo.rapidez)
            {
                cout << "Miguel gana en rapidez" << endl;
            }
        }
    }
}
//Fin Rapidez
```

//Comprobacion Habilidades David

```
//Comprobacion Agilidad
if (david.agilidad > miguel.agilidad)
{
    if (david.agilidad > eva.agilidad)
    {
        if (david.agilidad > jesus.agilidad)
        {
            if (david.agilidad > antimo.agilidad)
            {
                cout << "David gana en agilidad" << endl;
            }
        }
    }
}
// Fin Agilidad
```

```
//Comprobacion Fuerza
if (david.fuerza > miguel.fuerza)
{
    if (david.fuerza > eva.fuerza)
    {
        if (david.fuerza > jesus.fuerza)
        {
            if (david.fuerza > antimo.fuerza)
            {
                cout << "David gana en fuerza" << endl;
            }
        }
    }
}
//Fin Fuerza
```

```
//Comprobacion Mental
if (david.mental > miguel.mental)
{
    if (david.mental > eva.mental)
    {
        if (david.mental > jesus.mental)
        {
            if (david.mental > antimo.mental)
            {
                cout << "David gana en mental" << endl;
            }
        }
    }
}
```



```
}  
//Fin Mental  
  
//Comprobacion Rapidez  
if (david.rapidez > miguel.rapidez)  
{  
    if (david.rapidez > eva.rapidez)  
    {  
        if (david.rapidez > jesus.rapidez)  
        {  
            if (david.rapidez > antimo.rapidez)  
            {  
                cout << "David gana en rapidez" << endl;  
            }  
        }  
    }  
}  
}  
//Fin Rapidez  
  
//Comprobacion Habilidades Eva  
  
//Comprobacion Agilidad  
if (eva.agilidad > miguel.agilidad)  
{  
    if (eva.agilidad > david.agilidad)  
    {  
        if (eva.agilidad > jesus.agilidad)  
        {  
            if (eva.agilidad > antimo.agilidad)  
            {  
                cout << "Eva gana en agilidad" << endl;  
            }  
        }  
    }  
}  
//Fin Agilidad  
  
//Comprobacion Fuerza  
if (eva.fuerza > miguel.fuerza)  
{  
    if (eva.fuerza > david.fuerza)  
    {  
        if (eva.fuerza > jesus.fuerza)  
        {  
            if (eva.fuerza > antimo.fuerza)  
            {  
                cout << "Eva gana en fuerza" << endl;  
            }  
        }  
    }  
}
```

```
        cout << "Eva gana en fuerza" << endl;
    }
}
}
//Fin Fuerza
```

```
//Comprobacion Mental
if (eva.mental > miguel.mental)
{
    if (eva.mental > david.mental)
    {
        if (eva.mental > jesus.mental)
        {
            if (eva.mental > antimo.mental)
            {
                cout << "Eva gana en mental" << endl;
            }
        }
    }
}
//Fin Mental
```

```
//Comprobacion Rapidez
if (eva.rapidez > miguel.rapidez)
{
    if (eva.rapidez > david.rapidez)
    {
        if (eva.rapidez > jesus.rapidez)
        {
            if (eva.rapidez > antimo.rapidez)
            {
                cout << "Eva gana en rapidez" << endl;
            }
        }
    }
}
//Fin Rapidez
```

```
//Comprobacion Habilidades Jesus
```

```
//Comprobacion Agilidad
if (jesus.agilidad > miguel.agilidad)
{
    if (jesus.agilidad > david.agilidad)
    {
```

```
    if (jesus.agilidad > eva.agilidad)
    {
        if (jesus.agilidad > antimo.agilidad)
        {
            cout << "Jesus gana en agilidad" << endl;
        }
    }
}
//Fin Agilidad
```

```
//Comprobacion Fuerza
if (jesus.fuerza > miguel.fuerza)
{
    if (jesus.fuerza > david.fuerza)
    {
        if (jesus.fuerza > eva.fuerza)
        {
            if (jesus.fuerza > antimo.fuerza)
            {
                cout << "Jesus gana en fuerza" << endl;
            }
        }
    }
}
//Fin Fuerza
```

```
//Comprobacion Mental
if (jesus.mental > miguel.mental)
{
    if (jesus.mental > david.mental)
    {
        if (jesus.mental > eva.mental)
        {
            if (jesus.mental > antimo.mental)
            {
                cout << "Jesus gana en mental" << endl;
            }
        }
    }
}
//Fin Mental
```

```
//Comprobacion Rapidez
if (jesus.rapidez > miguel.rapidez)
{
```

```
if (jesus.rapidez > david.rapidez)
{
    if (jesus.rapidez > eva.rapidez)
    {
        if (jesus.rapidez > antimo.rapidez)
        {
            cout << "Jesus gana en rapidez" << endl;
        }
    }
}
//Fin Rapidez

//Comprobacion Habilidades Antimo

//Comprobacion Agilidad
if (antimo.agilidad > miguel.agilidad)
{
    if (antimo.agilidad > david.agilidad)
    {
        if (antimo.agilidad > eva.agilidad)
        {
            if (antimo.agilidad > jesus.agilidad)
            {
                cout << "Antimo gana en agilidad" << endl;
            }
        }
    }
}
//Fin Agilidad

//Comprobacion Fuerza
if (antimo.fuerza > miguel.fuerza)
{
    if (antimo.fuerza > david.fuerza)
    {
        if (antimo.fuerza > eva.fuerza)
        {
            if (antimo.fuerza > jesus.fuerza)
            {
                cout << "Antimo gana en fuerza" << endl;
            }
        }
    }
}
//Fin Fuerza
```

```
//Comprobacion Mental
if (antimo.mental > miguel.mental)
{
    if (antimo.mental > david.mental)
    {
        if (antimo.mental > eva.mental)
        {
            if (antimo.mental > jesus.mental)
            {
                cout << "Antimo gana en mental" << endl;
            }
        }
    }
}
//Fin Mental

//Comprobacion Rapidez
if (antimo.rapidez > miguel.rapidez)
{
    if (antimo.rapidez > david.rapidez)
    {
        if (antimo.rapidez > eva.rapidez)
        {
            if (antimo.rapidez > jesus.rapidez)
            {
                cout << "Antimo gana en rapidez" << endl;
            }
        }
    }
}
//Fin Rapidez
}
```

Observaciones

Falto agregar algún tipo de comprobación en caso de empates ya que en esas situaciones no hay algo que se imprima

Conclusiones

Veo mucho mas sencillo el uso de Struct