

# **MUNDO ARTIFICIAL ONLINE**

Carlos Daniel Vargas Walteros

Ingeniería de sistemas

Gestión de proyectos

Universidad de pamplona

2024

## INFORMACIÓN

Con base a mis conocimientos de TypeScript y JavaScript realice un proyecto piloto acerca de un juego online aplique lo que entiendo de programación para simular como seria las estadísticas en ese juego y mundo donde hay muchos jugadores aplicando ciertas opciones ya sea de personalización o simplemente acciones en sí.



### EXPLICACIÓN DEL CODIGO:

Archivo: HISTORIAL\_JUGADAS.ts

#### CLASE JUGADOR

Primero realice la clase y función principal llamada jugador, esto la cual hacer que al crear al nuevo jugador esta función lo clasifique como un jugador principal con parámetros de variables nombre, rol, edad, sexo.

Esto la cual va hacer que los jugadores personalicen sus personajes a sus gustos.

```
class Jugador {  
  nombre: string;  
  rol: string;  
  edad: number;  
  sexo: string;  
  
  constructor(n: string, r: string, e: number, a: string) {  
    this.nombre = n;  
    this.rol = r;  
    this.edad = e;  
    this.sexo = a;  
  }  
}
```

## PROPIEDADES

- ( mostrarInfo )

Esta propiedad muestra la información de los jugadores ya guardadas en jugador.

```
mostrarInfo() {  
  document.write(" <br> - El jugador " + this.nombre + " de rol: " + this.rol + " y tiene edad " + this.edad + " con genero : " + this.sexo + "<br>");  
}
```

- ( atacar )

Esta propiedad ataca a otros jugadores y cuantas veces lo atacara.

```
atacar(AT: number, J: string) {  
  document.write(" <br> - El jugador " + this.nombre + " atacara " + AT + " veces a jugador: " + J + "<br>");  
}
```

- ( dormir )

Esta propiedad simplemente es para que el jugador descanse ósea dormir en cuantos minutos.

```
dormir(nm: number) {  
  document.write("<br> - JUGADOR " + this.nombre + " duerme " + nm + " minutos " + "<br>");  
}
```

- ( esconderse )

Esta propiedad es para que el jugador se esconda por ciertos minutos

```
esconderse(EC: string, min: number) {  
  document.write(" <br> - JUGADOR " + this.nombre + " se escondio en: " + EC + " por " + min + " minutos " + "<br>");  
}
```

- ( mostrarDatos )

Esta propiedad es mas complicado de desarrollar ya que su objetivo es mostrar los datos de jugadores pero con un diseño en general esto toma las variables: nombre , edad , rol , sexo.

Mas de todo su parámetro mostrara una imagen 100x100 ya que en su parámetro se pondrá el link de la imagen o gif del jugador pero este enlace tendrá que ser online o de otra ruta.

Clasificara en formato HTML la etiqueta <div> ya que es para dividir en cuadrícula las imágenes tipo contenedor.

```

mostrarDatos(L: string) {
  const contenedor = document.createElement('div');
  contenedor.className = 'jugador';
  contenedor.innerHTML = `
    <br> NOMBRE: ${this.nombre} <br>
    ROL: ${this.rol} <br> |
    EDAD: ${this.edad} <br>
    GENERO: ${this.sexo} <br>
    <br>
    
    <br>
  `;
  document.body.appendChild(contenedor);
}

```

NOMBRE: SANCHEZ  
 ROL: CIVIL  
 EDAD: 18  
 GENERO: M



NOMBRE: CHILY  
 ROL: TRAFICANTE  
 EDAD: 26  
 GENERO: F



NOMBRE: MANUEL  
 ROL: CIVIL  
 EDAD: 21  
 GENERO: MTF



NOMBRE: JAVI  
 ROL: REY  
 EDAD: 15  
 GENERO: M



>

## CLASE TITLES

Esta clase es simplemente para poner títulos ósea solo strings, se hizo por el cual se miraría más organizado los títulos en general, solo lleva la propiedad ( mostrar ) que funciona para mostrar los títulos

```
class Titles {
    TT: string;

    constructor(C: string) {
        this.TT = C;
    }

    mostrar() {
        document.write("<br><br>" + this.TT + "<br><br>");
    }
}
```

## INFORMACION Y ACCIONES

Esto es simplemente los resultados.

```
// TITULOS
T1 = new Titles("---[ INFORMACION JUGADORES ]---");
T2 = new Titles("---[ HISTORIAL DE JUGADAS ]---");
T3 = new Titles("---[ SKIN DE JUGADORES ]---");
T4 = new Titles("---[ DATOS DE JUGADOR ]---");

// JUGADORES
p1 = new Jugador("SANCHEZ", "CIVIL", 18, "M");
p2 = new Jugador("CHILY", "TRAFICANTE", 26, "F");
p3 = new Jugador("MANUEL", "CIVIL", 21, "MTF");
p4 = new Jugador("JAVI", "REY", 15, "M");

// Mostrar información y acciones
T1.mostrar();
p1.mostrarInfo();
p2.mostrarInfo();
p3.mostrarInfo();
p4.mostrarInfo();

T2.mostrar();
p1.atacar(9, "CHILY");
p2.dormir(90);
p1.dormir(200);
p3.atacar(3, "SANCHEZ");
p1.escondese("un árbol", 5);
p4.dormir(1000);
p4.escondese("en su camuflaje", 9);

// Título adicional
T4.mostrar();

// Mostrar datos con imágenes
p1.mostrarDatos("https://i.postimg.cc/KvSxcBVL/SANCHES.png");
p2.mostrarDatos("https://i.postimg.cc/pLPvzxPL/CHILY.png");
p3.mostrarDatos("https://i.postimg.cc/Vk48FTNB/MANUEL.png");
p4.mostrarDatos("https://i.postimg.cc/ZqMzGdpw/JAVI.png");
```

ARCHIVO: SKIN.ts

INFORMACIÓN: Es para poder personalizar las skins de los jugadores, es diferente a mostrarinfo de la clase jugador pero se vincula con el archivo "HISTORIAL\_JUGADAS.ts" para poner vincular a los jugadores con las variables de nuevos jugadores

#### CLASE JUGADORSKIN

Todos sus parámetros son de tipo string, como parámetros tenemos a jugador, color\_piel, traje, accesorio.

jugador: esta variable es una instancia de jugador, es solo llamar la variable del nuevo jugador

color\_piel: color de piel del jugador

traje: esto no va concordar con rol puesto es una ropa así que puede ser cualquiera

accesorio: el arma del jugador

Solo tendrá la función de ( mostrarInfo ) para mostrar los resultados escogidos.

```
mostrarInfo() {  
  document.write("<br> - El jugador de nombre " + this.jugador.nombre + " de color de piel " + this.color_piel + " con traje de: " + this.traje + " de arma " + this.accesorio + "<br>");  
}
```

#### ENLACE DEL PROYECTO:

Este proyecto este comprimido tiene los documentos TS Y JS además de su HTML

Link:

[https://drive.google.com/file/d/1ZvbOF\\_E4Wyp1umywON8icM1k4b6ZtRKB/view?usp=sharing](https://drive.google.com/file/d/1ZvbOF_E4Wyp1umywON8icM1k4b6ZtRKB/view?usp=sharing)