

Desarrollo de Software
SD-08: Laboratorio: Declaración de clases (100 minutos)

Laboratorio 4

Lanzamiento de la Actividad

- **Abra** el proyecto SD-08-1: ClassesLab-Individual en VS code siguiendo las instrucciones mencionadas en la Guía de Programación Local TIFC.
- **Examine** el fragmento de código de ejemplo proporcionado.

Ejemplo de fragmento de código

```
exportar clase Jugador {  
  constructor() {  
    //los datos van aquí  
  }  
  
  Método() {  
    //el código va aquí  
  }  
}
```

Este programa JavaScript:

- Defina una **clase** con el nombre **Jugador** que se utilizará para construir objetos del mismo tipo.
 - El método **constructor()** se llama cuando se construye un nuevo objeto.
 - Cada objeto construido construye sus **propias** propiedades siguiendo la plantilla en el constructor.
 - Otros métodos definidos en la clase se convierten en **métodos compartidos** para la clase.
 - Cada objeto construido **hereda** estos métodos de la clase.

Nota: Hay una palabra en este código que aún no se le ha explicado: **exportar**. **Aprenderás más en la próxima lección.**

Desarrollo de Software
SD-08: Laboratorio: Declaración de clases (100 minutos)

Tareas

Las clases son una sintaxis alternativa para definir una **plantilla** para construir objetos. Al igual que las funciones de constructor, se pueden usar para construir objetos con su **propio** conjunto de propiedades y métodos. Al igual que los prototipos, también permiten que los objetos los **hereden**. Los **métodos del constructor** de clases se utilizan para crear una instancia de una clase. **Los métodos compartidos** son heredados por cada instancia.

¡Hoy, definirá clases, inicializará objetos con propiedades y actuará sobre esas propiedades con métodos!

Para cada una de estas tareas, construirá y repetirá su solución para la tarea anterior.

Tarea 1: Está trabajando en un código que se utilizará en un videojuego. Se le ha pedido que cree una clase de objeto. Estos se utilizarán para definir a los jugadores en ese juego. Cada jugador puede elegir su propio nombre, y estos objetos se utilizarán para almacenarlos.

1. Modifique la clase de jugador para que acepte un "**nombre**" de jugador en un argumento.
 - La clave de esta propiedad en el objeto resultante **debe** ser "nombre" - ¡recuerde, **los computadores son muy literales!**

Tarea 2: Ahora se le ha pedido que mejore su código, para que los objetos del jugador puedan definir tanto un nombre como un número de nivel.

1. Modifique la clase Jugador para que acepte una cadena de nombre de jugador y un número de "**nivel**" en dos argumentos separados.
 - La clave de esta propiedad en el objeto resultante **debe** estar "nivelada", ¡recuerde que **los computadores son muy literales!**

Tarea 3: Ahora se le ha pedido que incluya un método que enviará una cadena a la consola anunciando un aumento de nivel.

1. Modifique la clase Jugador para que acepte una cadena de nombre de jugador y un número de nivel en dos argumentos separados.

Desarrollo de Software
SD-08: Laboratorio: Declaración de clases (100 minutos)

2. Luego, defina un método de objeto compartido info() que **imprimirá** la siguiente cadena, reemplazando los dos marcadores de posición:
 - `<nombre> ¡ha alcanzado el nivel <nivel>!`
 - Un jugador llamado **Tara** en el nivel **6** debería dar como resultado "`¡Tara ha alcanzado el nivel 6!`" impreso en la consola.

Tarea 4: Se le ha pedido que incluya un método para subir de nivel a un jugador, aumentando su número de nivel en uno.

1. Modifique la clase Jugador para que acepte una cadena de nombre de jugador y un número de nivel en dos argumentos separados.
2. A continuación, defina un método de objeto compartido info() que generará la siguiente cadena:
 - `<nombre> ¡ha alcanzado el nivel <nivel>!`
3. Finalmente, defina un segundo método de objeto compartido llamado levelUp () que **incrementará** el nivel del Jugador.

Tareas Adicionales

Si ha completado las tareas anteriores, pruebe las siguientes tareas adicionales para **experimentar** más a fondo.

- **Tarea 5:** Experimente con permitir al jugador subir de nivel en función de los puntos de experiencia ganados.
 - Un punto de experiencia es un **número**. Una subida de nivel debería producirse cuando un jugador gana suficientes puntos de experiencia.
 - Intenta añadir un método que permita a un jugador ganar un número determinado de puntos de experiencia.
 - ¿Cuántos puntos de experiencia deberían dar lugar a una subida de nivel? ¿Cómo puede controlar esta cifra?
- **Tarea 6:** Experimente con la posibilidad de añadir objetos construidos del jugador a una **matriz** de miembros del grupo.
 - ¿Cómo debe identificarse en el código una matriz de miembros del partido?
 - Intenta añadir métodos para añadir o eliminar objetos jugadores de un grupo determinado.
- **Tarea 7:** Experimente con permitir que el jugador tenga un inventario de objetos.
 - Prueba a añadir métodos para añadir o eliminar elementos de un inventario.
 - ¿Cómo puede controlar la cantidad de cada artículo? ¿Qué **estructura de datos** necesitaría para ello?

Desarrollo de Software
SD-08: Laboratorio: Declaración de clases (100 minutos)

Puntuación

El instructor puntuará las tareas de codificación al final de la sesión.

Sugerencias

- La palabra clave “this” se utiliza en los constructores para permitir que los objetos contruidos tengan sus propiedades **propias**.
- Puede hacer referencia a las propiedades propias de un objeto en código, como en un método de **objeto compartido**.
- Incrementar significa añadir 1 a un número. Hay muchas formas de **incrementar** el valor de una variable o propiedad de un objeto.