



**¡Les damos la
bienvenida!**

¿Comenzamos?

Esta clase va a ser

- grabada

Clase 02. DESARROLLO AVANZADO DE BACKEND

Nuevas funcionalidades de los lenguajes ECMAScrip

Objetivos de la clase



Conocer las características de ECMAScript



Aplicar los conceptos incorporados en el desarrollo backend



MAPA DE CONCEPTOS



JavaScript y ECMAScript

Punto de partida: ES6

ECMAScript 6

Desde ES5 en 2009, hubo un largo período de tiempo en el cual no hubo grandes actualizaciones en los estándares del lenguaje. Hasta ECMAScript 6 en 2015, el cual supuso una revolución en el lenguaje debido a sus grandes cambios.

La salida de ES6 marcó un antes y un después en la historia del lenguaje, ya que a partir de éste se lo comenzó a considerar una implementación “moderna”.

JavaScript y ES7



Ejemplo en vivo

- ✓ Utilización del operador exponencial y manejo de array con includes.

JavaScript y ES8



Ejemplo en vivo

- ✓ Utilización de los métodos `Object.values`, `Object.keys` y `Object.entries`

JavaScript y ES9



Ejemplo en vivo

- ✓ Utilización básica de operador rest y spread en objetos.



Para pensar

Hemos notado muchas cosas que se han implementado a lo largo de ECMAScript.

- ✓ ¿Es obligatorio aprender cada nueva implementación para mantenerse en el mundo laboral?
- ✓ ¿El uso de una nueva implementación implica el reemplazo de lo viejo?



Break

¡10 minutos y volvemos!

Javascript y ES10



Ejemplo en vivo

- ✓ Validación de cadena con trim
- ✓ Aplanado de Array con múltiple anidación

JavaScript y ES11



Ejemplo en vivo

- ✓ Explicación de asignación de variable a partir de un nullish (??) para entender su diferencia con el operador OR (||)
- ✓ Explicación de una propiedad privada en una clase.

Registrador de tickets de eventos

¿Cómo lo hacemos? **Se creará una clase que permitirá llevar una gestión completa de usuarios que deseen acceder a dichos eventos.**

- ✓ Definir clase TicketManager, el cual tendrá un arreglo de eventos que iniciará vacío
- ✓ La clase debe contar con una variable privada "precioBaseDeGanancia", la cual añadirá un costo adicional al precio de cada evento.
- ✓ Debe contar con el método "getEventos" El cual mostrará los eventos guardados.
- ✓ Debe contar con el método "agregarEvento" El cual recibirá los siguientes parámetros:
 - nombre
 - lugar
 - precio (deberá agregarse un 0.15 del valor original)
 - capacidad (50 por defecto)
 - fecha (hoy por defecto)

El método deberá crear además el campo id autoincrementable y el campo "participantes" que siempre iniciará con un arreglo vacío.

Registrador de tickets de eventos

- ✓ Debe contar con un método "agregarUsuario" El cual recibirá:
 - id del evento (debe existir, agregar validaciones)
 - id del usuario

El método debe evaluar que el evento exista y que el usuario no haya estado registrado previamente (validación de fecha y capacidad se evitará para no alargar el reto)

Si todo está en orden, debe agregar el id del usuario en el arreglo "participantes" de ese evento.

- ✓ Debe contar con un método "ponerEventoEnGira" El cual recibirá:
 - id del evento
 - nueva localidad
 - nueva fecha

El método debe copiar el evento existente, con una nueva localidad, nueva fecha, nuevo id y sus participantes vacíos (Usar spread operator para el resto de las propiedades)

¿Preguntas?

Muchas gracias.