

SPRINT FINAL MÓDULO

Plan Formativo: Desarrollo de Aplicaciones Full Stack JavaScript Trainee

Nivel de Dificultad: Medio / Alto

Módulo 4: Programación avanzada en JavaScript**APRENDIZAJES ESPERADOS****Aprendizaje Esperado 1**

Utilizar los conceptos fundamentales de la programación Orientada a Objetos acorde al lenguaje JavaScript para resolver un problema simple.

Aprendizaje Esperado 2

Utilizar las nuevas funcionalidades de la especificación ES6+ para la implementación de un algoritmo JavaScript que resuelve un problema planteado.

Aprendizaje Esperado 3

Reconocer los elementos fundamentales del Domain Object Model y los mecanismos para la manipulación de elementos en un documento HTML y la herencia de clases

Aprendizaje Esperado 4

Utilizar elementos de programación asíncrona para resolver un problema simple distinguiendo los diversos mecanismos para su implementación acorde al lenguaje JavaScript.

Aprendizaje Esperado 5

Utilizar el objeto XHR y la API Fetch para el consumo de una API externa y su procesamiento acorde al lenguaje JavaScript.

OBJETIVO DEL EJERCICIO

El/la estudiante debe implementar soluciones en base a los aprendizajes esperados, una problemática y darle solución en base al uso y manejo de variables y distintos tipos de datos: number, boolean, string además de objetos y array. También debe implementar soluciones en base a el estándar de JavaScript ECMAScript v6 usando módulos export - import, let-const, arrow function, clases, el método map y los templates literales.

Además, debe conjugar lo antes mencionado plasmando el uso de funciones según sea necesario.

Debe utilizar funciones constructoras en base al principio de la Programación orientada a Objetos (POO) y reconociendo los elementos fundamentales del Domain Object Model y los mecanismos para la manipulación de elementos en un documento HTML y la herencia de clases

Finalmente, el/la estudiante debe utilizar elementos de programación asíncrona para resolver un problema simple distinguiendo los diversos mecanismos para su implementación acorde al lenguaje JavaScript, como la utilización del objeto XHR y la API fetch para el consumo de una API externa y su procesamiento.

PROBLEMA

Se requiere desarrollar un programa con JavaScript que se conecte a la API de entrenamiento POKE API. Una vez extraído ese archivo JSON el landing deben visualizar los distintos Pokemones en un card de Bootstrap en donde se visualicen sus atributos y su imagen respectiva

Este card debe tener la funcionalidad que al hacer click sobre él se debe visualizar en una modal la información del Pokémon mostrando sus poderes en una Gráfico de torta.

Para mostrar los card se deben generar 2 html, uno para búsqueda por ID y otro por búsqueda masiva

El primero de ellos debe contar con un input para hacer una búsqueda y filtrar los pokemones por nombre y/o ID

En el segundo HTML, al apretar un botón **Cargar** se deben mostrar los primeros 20 pokemones en el Landing. Luego de apretarlo se debe crear un segundo botón **Cargar Más** que muestre los siguientes 20 pokemones en pantalla así sucesivamente. Deben crear un tercer botón llamado **Borrar todo** que reinicie la secuencia y deje nuevamente los 20 primeros pokemones

Para poder trabajar el proyecto pueden usar los siguientes recursos:

Documentación PokeAPI

- <https://pokeapi.co/>

Librería para generar gráficos

- <https://www.chartjs.org/docs/latest/>

REQUERIMIENTOS

1. Crear una función asíncrona para obtener los pokemones que retorne una promesa después de un (2) segundo utilizando setTimeout. El mensaje a retornar debe en caso de try debe ser un mensaje que indique: "Información Enviada". **(4 puntos)**
2. Dentro de un bloque Try/Catch, utilizar el método fetch mediante la instrucción await para recibir el valor directamente de la promesa. **(5 puntos)**
3. Utilizar un método de manejo del DOM para visualizar la información **(4 puntos)**
4. Mostrar la información del Pokémon en un card **(4 punto)**
5. Generar un HTML con la función búsqueda por nombre **(5 puntos)**
6. Generar un modal con un gráfico dinámico para los poderes del Pokémon **(5puntos)**
7. Generar un HTML con la función cargar **(3 puntos)**

Requerimiento opcional

1. Generar un tercer HTML que permita hacer competir a 2 pokemones, similar al ejercicio de piedra, papel o tijera

El trabajo debe ser subido en formato zip, rar o archivo de texto con URL del repositorio

RECURSOS BIBLIOGRÁFICOS

- <https://www.freecodecamp.org/espanol/news/como-usar-async-await-para-escribir-un-codigo-mejor-en-javascript/>
- https://www.w3schools.com/ai/ai_chartjs.asp
- <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/try..catch>