



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TEMA:

Promesas

PRESENTADO POR:

Hernández Miranda Rafael Francisco

GRUPO:

10B

MATERIA:

Gestión del Proceso de Desarrollo de Software

PROFESOR:

Ray Brunett Parra Galaviz

FECHA:

25/01/2025.

Introducción

Una Promise (promesa en castellano) es un objeto que representa la terminación o el fracaso de una operación asíncrona. Dado que la mayoría de las personas consumen promises ya creadas, esta guía explicará primero cómo consumirlas, y luego cómo crearlas.

Esencialmente, una promesa es un objeto devuelto al cual se adjuntan funciones callback, en lugar de pasar callbacks a una función.

Las promesas funcionan de la siguiente manera:

- Se crea una promesa con el constructor Promise
- La promesa puede estar en estado pendiente, cumplido o rechazado
- Si la promesa se cumple, se llama al método resolve
- Si la promesa se rechaza, se llama al método reject
- Se asocian acciones con la promesa mediante los métodos then, catch o finally

Los métodos then, catch y finally se usan para manejar los resultados de las promesas.

Métodos de promesas

- **then()**: Se usa para manejar el resultado exitoso de una promesa
- **catch()**: Se usa para manejar errores que puedan ocurrir durante la ejecución de la promesa
- **finally()**: Se usa para ejecutar una función después de que la promesa se resuelva o se rechace

En este ejemplo, Utilizaremos el api de Pokémon

```
code.js > catch() callback
1  function obtenerPokemon(nombre) {
2      return new Promise((resolve, reject) => {
3          fetch(`https://pokeapi.co/api/v2/pokemon/${nombre}`)
4              .then((response) => {
5                  if (!response.ok) {
6                      throw new Error("No se encontró el Pokémon");
7                  }
8                  return response.json();
9              })
10             .then((data) => resolve(data))
11             .catch((error) => reject(error.message));
12     });
13 }
14
15 // Uso de la promesa para obtener información de Pikachu
16 obtenerPokemon("pikachu")
17     .then((pokemon) => {
18         console.log(`Nombre: ${pokemon.name}`);
19         console.log(`Altura: ${pokemon.height}`);
20         console.log(`Peso: ${pokemon.weight}`);
21     })
22     .catch((error) => {
23         console.error("Error:", error);
24     });
25
```

Conclusión

Las promesas son una herramienta fundamental en el manejo de programación asíncrona en JavaScript, ya que permiten escribir código más limpio y estructurado. Su uso ha sido ampliamente adoptado en frameworks como React y Node.js, facilitando el desarrollo de aplicaciones web modernas.

Referencias

Promesas en JavaScript: qué son y cómo se crean. (s/f). Arsys. Recuperado el 5 de febrero de 2025, de <https://www.arsys.es/blog/promesas-javascript>

Promise. (s/f). MDN Web Docs. Recuperado el 5 de febrero de 2025, de https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

(S/f). Pokeapi.co. Recuperado el 5 de febrero de 2025, de <https://pokeapi.co/api/v2/pokemon/>