HINTS

CONSEJOS CONCEPTUALES

- El uso de promesas y `async await` es considerado como un nivel intermedio - avanzado dentro de la programación con `JavaScript`, debido a que no es un concepto fácil de entender a primera vista, y la sintaxis también puede resultar algo confusa.

- Siempre es buena idea comenzar a crear tus propias promesas, pues así podrás ver cómo funcionan "por dentro", y cuando empieces a utilizarlas en otras librerías, ya entenderás cómo funciona su flujo.

- Recuerda siempre que una promesa solo tiene dos caminos: o se resuelve, o se rechaza. Para obtener la respuesta de una, debemos utilizar el método `then()` (se pueden emplear tantos `then()` como quieras en cadena).

- Cuando una promesa es rechazada y no existe el control de esta respuesta, obtenemos un error de `runtime`, y nuestro programa se caerá; para esto es que debemos utilizar el método `catch()`.

- Siempre se utiliza un solo método `catch()`, ya sea después de un único `then()`, o una cadena de éstos.

- Es recomendable practicar primero con promesas, ya que esto nos prepara para entender de mejor forma la sintaxis `async-await`.

- Generalmente, `async-await` se define como "Syntactic sugar", o Azúcar sintáctico o de sintaxis, debido a que está basado en promesas, y no es más que una forma fácil de escribirlas. Por lo tanto, funcionan igual, pero se escriben distinto, en donde `async-await` hace ver el código de una manera más "síncrona".

- Siempre debes leer la documentación de las apis a las que quieras acceder, pues muchas veces encontrarás algunas que son pagadas, u otras que permiten cuentas gratuitas, pero debes obtener una clave o "key", la cual debes utilizar para realizar la consulta a la api, por ejemplo, algunas de las apis de Google.