UTNMDP
Regional Mar del Plata

Actividad Práctica 5 – Promesas + Async Await

1. Crear una promesa que se resuelva después de 2 segundos con el mensaje "Proceso completado". Usar .then() y .catch() para mostrar el resultado en consola.

2. Hacer una función que reciba un número por parámetro: si el número es par, la promesa se resuelve mostrando el mensaje "El número es par"; Si el número es impar, la promesa se rechaza con el mensaje “El número es impar”. Usar .then() y .catch().

3. Crear una promesa que devuelva un número. Encadenar .then() para:
   a. Sumarle 5
   b. Multiplicar por 2
   c. Mostrar el resultado final

4. Hacer una función que simule “cargar datos” con setTimeout (por ejemplo, tarda 3 segundos en resolverse). Mostrar "Cargando..." antes de la promesa y "Datos cargados" al resolverla.

5. Reutilizar el ejercicio 1, pero en lugar de .then(), crear una función async y usar await para esperar la promesa. Mostrar el mensaje en consola.

6. Reutilizar el ejercicio 2 (número par/impar), pero ahora implementarlo usando async/await, y capturar el error con try/catch.
   - Recordá que cuando trabajamos con .then() se usa .catch() para manejar errores.
   - En cambio, cuando usamos async/await, los errores de una promesa rechazada se capturan con try/catch, igual que las excepciones en código sincrónico.

   Dicho esto, Implementar la función de verificación usando async/await y capturar el error con try/catch.

7. Crear 3 funciones que devuelvan promesas con un setTimeout distinto (por ejemplo: 1s, 2s y 3s). Llamarlas en una función async usando await para ejecutarlas en orden y mostrar los resultados.

8. Tomar las 3 funciones del ejercicio anterior, pero ahora ejecutarlas con Promise.all (investigar). Mostrar todos los resultados juntos cuando se resuelvan.