

Ejercicios excepciones

1. **División entre cero:** Crea una función que divida dos números e implementa un manejo de excepciones para evitar divisiones entre cero.
2. **Conversión de cadena a entero:** Escribe una función que convierta una cadena a un número entero. Maneja la excepción si la conversión no es posible.
3. **Acceso a índice en una lista:** Implementa una función que reciba una lista y un índice. Maneja la excepción cuando el índice está fuera de rango.
4. **Abrir archivo inexistente:** Crea una función que intente abrir un archivo que no existe y maneja la excepción correspondiente.
5. **Ingreso de número válido:** Escribe un programa que pida al usuario un número y maneje la excepción si el usuario introduce algo que no sea un número.
6. **Manejo de múltiples excepciones:** Crea una función que maneje múltiples tipos de excepciones (p. ej., `ZeroDivisionError`, `ValueError`) y muestre un mensaje adecuado para cada una.
7. **Función que lanza una excepción:** Implementa una función que lanza una excepción de tipo `ValueError` si el valor recibido no es positivo.
8. **Raising manual de excepciones:** Crea una función que lance manualmente una excepción personalizada cuando se cumpla una condición específica.
9. **Ingreso de nombre no vacío:** Implementa un programa que pida al usuario su nombre y maneje una excepción si el nombre está vacío.
10. **Acceso a clave en diccionario:** Escribe una función que reciba un diccionario y una clave, y maneja la excepción si la clave no existe.
11. **Validar edad:** Crea una función que solicite una edad y maneje las excepciones si el valor ingresado no es válido (no numérico o negativo).
12. **Manejo de excepciones anidadas:** Implementa una función que tenga varios niveles de operaciones que puedan lanzar excepciones. Maneja las excepciones en los diferentes niveles.
13. **Intentos limitados de entrada:** Crea una función que permita al usuario ingresar un número, con un límite de intentos. Si después de varios intentos falla, lanza una excepción personalizada.
14. **Excepción al calcular raíz cuadrada:** Escribe una función que calcule la raíz cuadrada de un número, pero maneja la excepción si el número es negativo.

15. Cerrar archivo en manejo de excepciones: Crea una función que abra un archivo, realice alguna operación y, aunque haya una excepción, asegúrate de que el archivo se cierre correctamente utilizando **finally**.
16. Tipo de dato incorrecto: Implementa una función que maneje la excepción **TypeError** si se pasa un tipo de dato incorrecto a la función.
17. Reintento en caso de excepción: Crea una función que intente realizar una operación, y si ocurre una excepción, vuelve a intentarlo varias veces antes de lanzar una excepción final.
18. Excepciones con **try** y **else**: Escribe una función que utilice un bloque **try/except/else** para manejar excepciones, donde **else** se ejecuta solo si no hay excepciones.
19. Registro de errores en archivo: Crea un programa que intente realizar varias operaciones, maneje las excepciones, y guarde los mensajes de error en un archivo de texto.
20. Excepciones personalizadas: Define una clase de excepción personalizada y úsala para manejar una condición específica dentro de un programa.