

UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Tema: Herramientas para el manejo de errores



PRESENTA: (Jared Isaías Monje Flores)

Docente: (Michelle Emmanuel López Franco)

Materia: (Computación Tolerante a Fallas)

El manejo de errores se refiere a los procedimientos de respuesta y recuperación de las condiciones de error presentes en la aplicación de software. En otras palabras, es el proceso compuesto por anticipación, detección y resolución de errores de aplicación, errores de programación o errores de comunicación. El manejo de errores ayuda a mantener el flujo normal de ejecución del programa.

Instrucciones **try** y **catch** (bloques try-catch): En lenguajes de programación que admiten bloques de código try y catch, puedes envolver el código propenso a errores dentro de un bloque try. Si ocurre una excepción dentro del bloque try, el control se traslada al bloque catch correspondiente donde puedes manejar el error de manera adecuada.

```
31 try:
32     # Código propenso a errores
33 except TipoDeExcepcion as e:
34     # Manejo del error (puede incluir impresiones de mensaje, registro de errores, etc.)
35
```

Sentencias **finally**: Algunos lenguajes de programación también permiten el uso de la sentencia finally después de un bloque try y opcionalmente después de los bloques catch. El código dentro del bloque finally se ejecuta independientemente de si se produjo una excepción o no. Se utiliza comúnmente para realizar acciones de limpieza.

```
31 try:
32     # Código propenso a errores
33 except TipoDeExcepcion as e:
34     # Manejo del error
35 finally:
36     # Acciones que se ejecutan sin importar si hubo una excepción o no
37
```

Lanzamiento de excepciones personalizadas: Puedes definir tus propias clases de excepción personalizadas para manejar situaciones específicas. Esto te permite

tener un control más preciso sobre los tipos de errores que pueden ocurrir en tu programa.

```
31 class MiErrorPersonalizado(Exception):
32     pass
33 try:
34     if condicion_error:
35         raise MiErrorPersonalizado("Ocurrió un error personalizado")
36 except MiErrorPersonalizado as e:
37     # Manejo del error personalizado
```

Retorno de códigos de error: En lenguajes que no usan excepciones, a menudo se utilizan códigos de retorno especiales para indicar errores. Por ejemplo, una función puede devolver **-1** para indicar un error y **0** para indicar éxito.

Lanzamiento de excepciones: Para lanzar una excepción en C++, se utiliza la palabra clave **throw**.

```
30 #include <iostream>
31 using namespace std;
32
33 int main() {
34     try {
35         int divisor = 0;
36         if (divisor == 0) {
37             throw runtime_error("División por cero");
38         }
39         int resultado = 10 / divisor;
40         cout << "Resultado: " << resultado << endl;
41     }
42     catch (const runtime_error& e) {
43         cout << "Error: " << e.what() << endl;
44     }
45     return 0;
46 }
```

window.onerror: Este evento global permite capturar y manejar errores no controlados en todo el contexto de la página web. Puede ser útil para registrar errores que ocurran en cualquier parte del código.

```
30 window.onerror = function(message, source, lineno, colno, error) {  
31     // Manejo de errores no controlados  
32 };
```

Manejo de errores en funciones asincrónicas: Si estás utilizando funciones asincrónicas con **async/await**, puedes utilizar un bloque try-catch dentro de la función para capturar errores.

```
34 async function miFuncion() {  
35     try {  
36         const resultado = await algunaPromesa();  
37         console.log(resultado);  
38     }  
39     catch (error) {  
40         console.error("Error:", error);  
41     }  
42 }
```

Evitar inyecciones SQL PHP

```
// Conexión a la base de datos usando PDO  
$pdo = new PDO("mysql:host=localhost;dbname=mydatabase", "username", "password");  
  
// Entrada del usuario  
$usuario_input = $_POST['usuario'];  
  
// Consulta preparada para evitar inyección SQL  
$query = "SELECT * FROM usuarios WHERE username = :usuario";  
$stmt = $pdo->prepare($query);  
$stmt->bindParam(':usuario', $usuario_input);  
$stmt->execute();  
  
$resultado = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

Python con SQLAlchemy

```
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from models import Usuario # Supongamos que tenemos un modelo Usuario definido

# Conexión a la base de datos usando SQLAlchemy
engine = create_engine("mysql://username:password@localhost/mydatabase")
Session = sessionmaker(bind=engine)
session = Session()

# Entrada del usuario
usuario_input = request.form['usuario']

# Consulta usando un ORM para evitar inyección SQL
usuario = session.query(Usuario).filter_by(username=usuario_input).first()
```

Estos ejemplos ilustran cómo usar consultas parametrizadas o consultas preparadas para evitar la inyección SQL en diferentes lenguajes. En lugar de concatenar directamente los valores del usuario en las consultas, se utilizan marcadores de posición (como :usuario en el caso de PDO) que se enlazan a los valores reales después. Esto asegura que los valores del usuario no sean interpretados como comandos SQL maliciosos.

En resumen, el uso adecuado de herramientas para el manejo de errores permite a los desarrolladores identificar, diagnosticar y corregir problemas en el código, mejorando la calidad y la estabilidad del software. Al anticiparse a posibles fallos y proporcionar información detallada sobre los errores, estas herramientas permiten a los programadores crear aplicaciones más resilientes, facilitando la detección y solución de problemas para ofrecer una experiencia más satisfactoria tanto para los usuarios como para los desarrolladores.

Referencias:

es, Q. (2023). *¿Qué es el manejo de errores? - definición de techopedia - Software 2023. Icy Science.*

<https://es.theastrologypage.com/error-handling>

Kamunya, T. (2023, July 19). *Las 11 mejores herramientas de seguimiento de errores para equipos de desarrollo modernos.* Geekflare.
<https://geekflare.com/es/bug-tracking-tools/>

Wikiwand - Manejo de excepciones. (2022). Wikiwand; Wikiwand.

https://www.wikiwand.com/es/Manejo_de_excepciones