

TÉCNICA DE TOLERANCIA A FALLOS IMPLEMENTADA

1. MANEJO DE ERRORES Y EXCEPCIONES EN PHP

En el proyecto, se ha implementado una técnica básica pero efectiva de tolerancia a fallos mediante el manejo adecuado de errores y excepciones en el backend (PHP). Esto permite que el sistema siga funcionando o informe al usuario de manera clara si ocurre algún problema inesperado, en lugar de mostrar mensajes de error técnicos o dejar la aplicación en un estado inconsistente.

¿Cómo se implementa?

Uso de bloques try-catch:

Todas las operaciones críticas, como la conexión a la base de datos y las consultas SQL, están envueltas en bloques try-catch. Si ocurre un error (por ejemplo, la base de datos no responde), el sistema captura la excepción y muestra un mensaje amigable al usuario, evitando que la aplicación se detenga abruptamente.

Mensajes de error personalizados:

En lugar de mostrar mensajes de error técnicos, el sistema informa al usuario con mensajes claros como "No se pudo completar la operación. Por favor, inténtelo más tarde."

Registro de errores (*logging*):

Los errores capturados pueden ser registrados en un archivo de log para que el administrador pueda revisarlos y tomar acciones correctivas sin afectar la experiencia del usuario.

Ejemplo de código:

```
try {
    $conexion = new PDO("mysql:host=localhost;dbname=clientes", "usuario",
"contraseña");
    // Operaciones con la base de datos...
} catch (PDOException $e) {
    // Registrar el error en un archivo log
    error_log($e->getMessage(), 3, "errores.log");
    // Mostrar mensaje amigable al usuario
    echo "Ocurrió un problema al conectar con la base de datos. Intente nuevamente
más tarde.";
}
```

2. VALIDACIÓN DE DATOS

Otra técnica de tolerancia a fallos es la validación exhaustiva de los datos tanto en el frontend (**JavaScript**) como en el backend (**PHP**). Esto previene que datos incorrectos o maliciosos lleguen a la base de datos y causen fallos.

3. RESPALDO DE LA BASE DE DATOS (OPCIONAL)

Como recomendación adicional, se puede programar un respaldo periódico de la base de datos, para poder restaurar la información en caso de un fallo grave.

4. CONCLUSIÓN

El proyecto implementa tolerancia a fallos principalmente mediante el manejo de errores y excepciones en PHP, mensajes amigables al usuario y registro de errores para su posterior análisis. Esto asegura que el sistema sea más robusto y confiable ante problemas inesperados.