

Felipe Alejandro Jiménez Castillo  
215671386



**Universidad de Guadalajara - CUCEI**  
**Computación Tolerante a Fallas**  
Herramientas para el Manejo de Errores

---

## Contenido

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Desarrollo</b>	<b>2</b>
<b>3</b>	<b>Conclusión</b>	<b>3</b>
<b>4</b>	<b>Bibliografía</b>	<b>4</b>

# 1 Introducción

El manejo de errores hace referencia a la respuesta y recuperación presentes en las condiciones de error que se encuentra un programa o software, en otras palabras es el proceso compuesto por la anticipación, detección y resolución de diferentes tipos de errores en una aplicación y/o programa. El manejo de errores nos ayuda a mantener el flujo de ejecución adecuado de un programa. En el día a día, muchas aplicaciones enfrentan numerosos retos de diseño cuando se consideran técnicas de manejo de errores, es por esto, que es todo una ciencia y arte conocer dichas técnicas para una implementación concisa.

# 2 Desarrollo

Una herramienta o software de manejo de errores, es aquel programa que nos ayuda, como desarrolladores, a identificar, rastrear y corregir el origen de un error. Estas herramientas pueden ser muy útiles para proyectos grandes o complejos, ya que nos ayudan a administrar y priorizar los errores, así como rastrear su origen y progreso. Con esto en mente una herramienta de manejo de errores debería tener al menos una de las siguientes características: Registro (ser capaz de identificar y denotar), Trazabilidad (ser capaz de rastrear), Priorizar (señalar la gravedad) y Gestionar (asignar-seleccionar).

## 1. Jira

- (a) Si bien *Jira* es un software para la gestión de proyectos, en esta se puede generar un estado de aplicación y realizar un seguimiento, asignación, priorización y gestión de los errores encontrados en nuestro software.

## 2. LSP

- (a) *LSP (Language Server Protocol)* es un servicio, creado por Microsoft, que agrega características como auto completado, ir a las definiciones, documentación, referencias y más, sobre el curso en el proceso de programación, funcionando como un protocolo de comunicación entre el IDE o editor y un servidor de lenguaje, lo que nos proporciona retroalimentación en tiempo real de errores, definiciones incorrectas, y más.

## 3. Visual Studio Code

- (a) Si bien *Visual Studio Code* es un editor de código hecho y derecho, este cuenta con una extensa cantidad de plugins que hacen del proceso de desarrollo un entorno más cómodo, con este podemos tener herramientas de detección de errores pre-runtime, así como análisis léxico, sintáctico y semántico de la mayoría de lenguajes en el mundo, por lo que podríamos considerarlo como una herramienta muy potente.

### 3 Conclusión

Si bien el proceso de programación es un proceso lógico de mucha complejidad, el mundo moderno nos ha guiado al desarrollo de herramientas que nos ayudan a priorizar los retos lógicos y poder tener, de segunda mano, un asistente "inteligente" para solucionar errores, menores o mayores, que podamos dejar atrás o simplemente hayamos olvidado. Es aquí donde las herramientas de manejo de errores nos dan la mano para ser parte de nuestro proceso y habilitar una mejora a nuestros productos de software y, finalmente, mejorar la calidad del proceso de desarrollo.

## 4 Bilbiografia

1. - Atlassian. (n.d.). Jira software - features. Atlassian. Retrieved August 27, 2023, from <https://www.atlassian.com/software/jira/features>
2. Fault tolerance. (n.d.). Learning Center; Imperva Inc. Retrieved August 15, 2023, from <https://www.imperva.com/learn/availability/fault-tolerance/>
3. Extensions, L. M. A. (n.d.). Visual Studio Code - code editing. Redefined. Visualstudio.com. Retrieved August 27, 2023, from <https://code.visualstudio.com/>
4. Gunnell, M., Haqshanas, R., & Cooling, S. (2017, May 1). Error Handling. Techopedia. <https://www.techopedia.com/definition/16626/error-handling>
5. Official page for Language Server Protocol. (n.d.). Github.Io. Retrieved August 27, 2023, from <https://microsoft.github.io/language-server-protocol/>