Materia: Computación Tolerante a Fallas.

NRC: 179961

Maestra: Lopez Franco, Michael Emanuel

Aula: X-02

Sección: D06

Alumno: Zashuvath López Moreno, Ethan Israel

Código: 216493953

Índice

[Introducción 3](#_Toc147323572)

[Objetivo: 3](#_Toc147323573)

[Contenido 3](#_Toc147323574)

[Conclusión 5](#_Toc147323575)

[Bibliografía 5](#_Toc147323576)

# Introducción

Realizaremos la modificación de un código proporcionado por el profesor, el cual fue elaborado por <https://gist.github.com/lauralorenz>

## Objetivo:

Realizar modificaciones al código de Laura Lorenz

# Contenido

Código base proporcionado por el profesor:

|  |
| --- |
|  |
|  |

Código modificado

|  |
| --- |
|  |
|  |
|  |

**Cambio 1: Configuración de la URL de la API y el nombre de la base de datos:** Aquí se establece la URL de la API de donde se obtendrán los datos y el nombre de la base de datos SQLite donde se almacenarán los datos extraídos.

**Cambio 2: Configuración de registro:** Se configura el sistema de registro (logging) para registrar mensajes informativos a nivel de INFO. Esto permitirá la creación de registros que ayuden en la depuración y el seguimiento del flujo de trabajo.

**Cambio 3: Función para realizar solicitudes HTTP de manera segura:** Se define una función llamada **‘safe\_request’** que se utiliza para realizar solicitudes HTTP a la API de manera segura. La función captura excepciones de solicitudes HTTP y registra errores en caso de que ocurran.

**Cambio 4: Uso de la función safe\_request para solicitudes HTTP:** En la función **‘get\_complaint\_data()’**, se utiliza la función **‘safe\_request’** para realizar la solicitud HTTP a la API. Esto asegura que cualquier error en la solicitud sea manejado adecuadamente.

**Cambio 5: Manejo de errores al interactuar con la base de datos:** En la función **‘store\_complaints(parsed)’**, se utiliza un bloque **‘try...except’** para manejar errores al interactuar con la base de datos SQLite. Si ocurre un error al crear la tabla o al insertar datos, se registrará un mensaje de error.

# Conclusión

Las modificaciones que se realizaron en el código son para mejorar la robustez y la seguridad del flujo de trabajo ETL (Extract, Transform and Load), además de añadir configuraciones para la API y el registro de errores, una función segura para las solicitudes HTTP y la implementación de manejadores de errores para las interacciones con SQLite.

# Bibliografía

JSONPlaceholder - fake online REST API for developers. (s. f.). <https://jsonplaceholder.cypress.io/>

Prefect. (2020, 17 abril). Getting started with Prefect (PyData Denver) [Vídeo]. YouTube. <https://www.youtube.com/watch?v=FETN0iivZps>

PyData.(2019, 4 enero). Task failed successfully - Jeremiah Lowin [Vídeo]. YouTube. <https://www.youtube.com/watch?v=TlawR_gi8-Y>LauraLorenz’s gists.

(s. f.). Gist. <https://gist.github.com/lauralorenz>