

**CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENERIAS**

**Aldo Guillermo Román Del Muro**

**INCO**

**217554557**

**Herramientas para el manejo de errores.**

**Computación Tolerante a fallas.**

**DR. MICHEL EMANUEL LOPEZ FRANCO.**

Introducción:

El manejo correcto de errores y excepciones es importante para que el nodo de integración funcione correctamente. Debe considerar cómo y cuándo la extensión definida por el usuario debe manejar errores y excepciones.

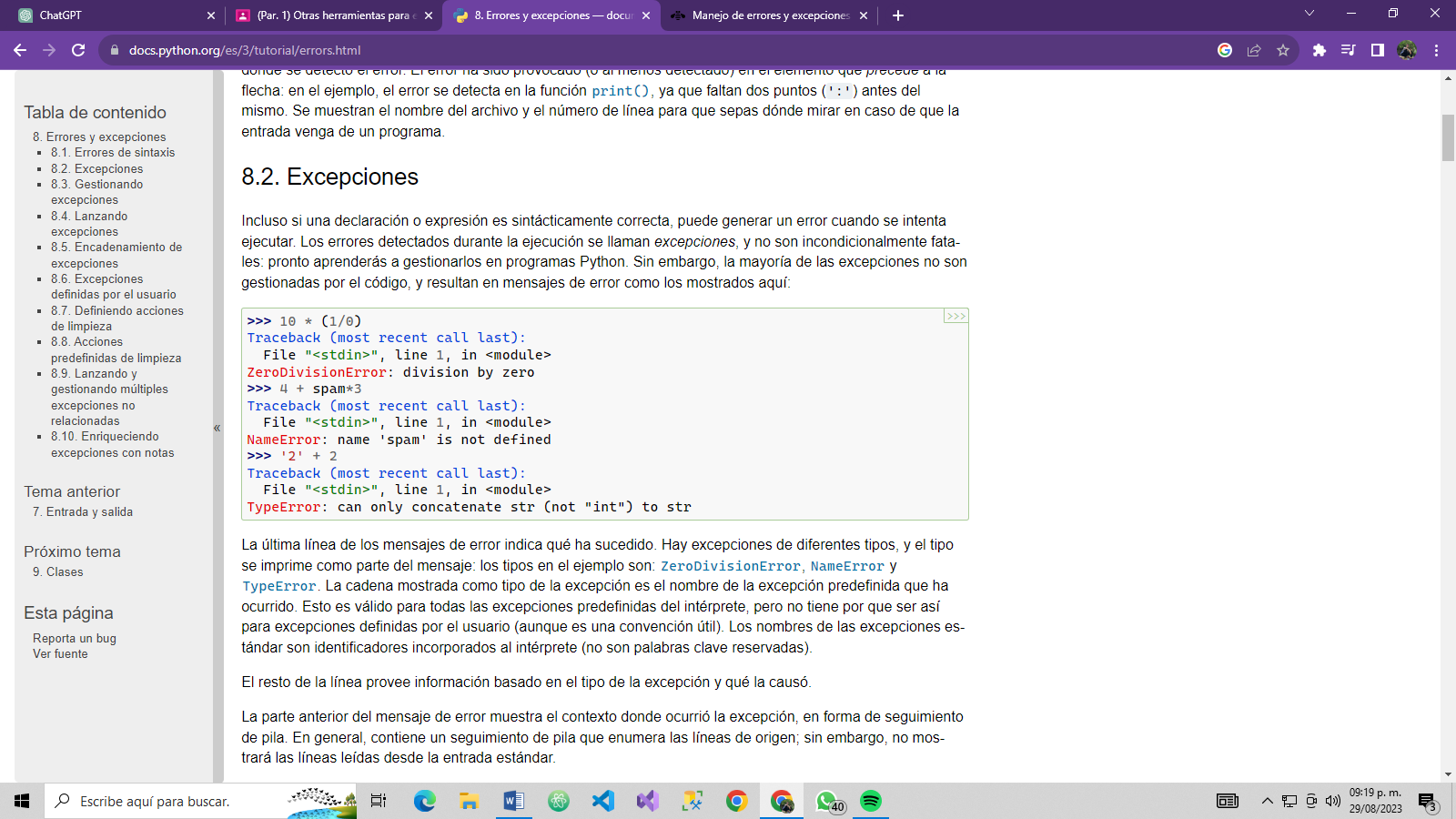
Desarrollo:

Se realizó una investigación del manejo de errores en distintos programas y encontré uno que me pareció particularmente bueno, es el manejo de errores y excepciones en Python.

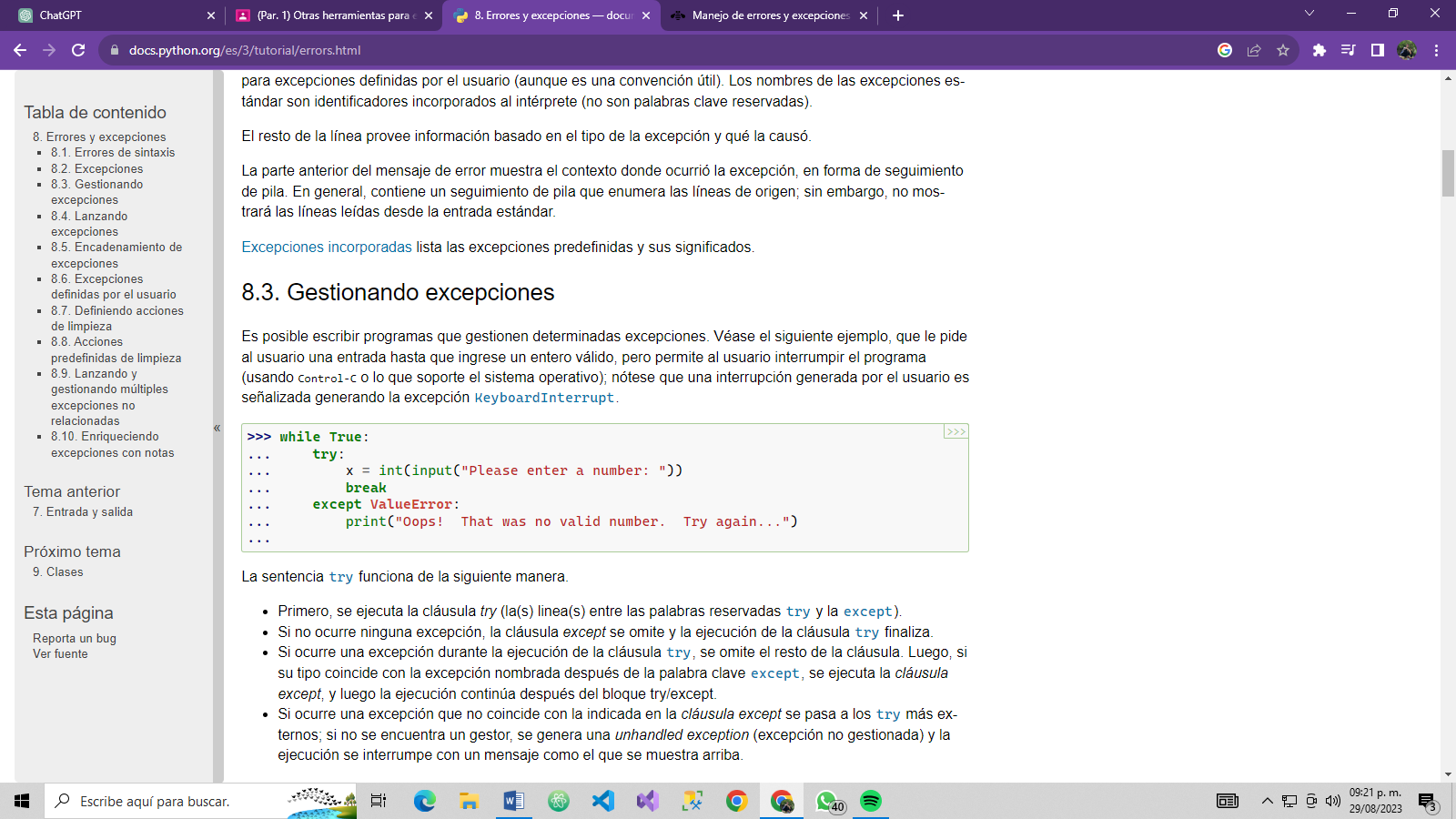
Veamos un poco de cómo funciona:

Hasta el momento solo conocíamos dos tipos de errores el de sintaxis y el de excepciones.

Incluso cuando una sentencia es correctamente escrita puede soltar error dentro del momento. Los errores destacados son conocidos como excepciones y no son incondicionalmente fatales.



Ahora veamos cómo se gestionan estas excepciones. Es posible escribir programas que gestionen excepciones como en el siguiente ejemplo:



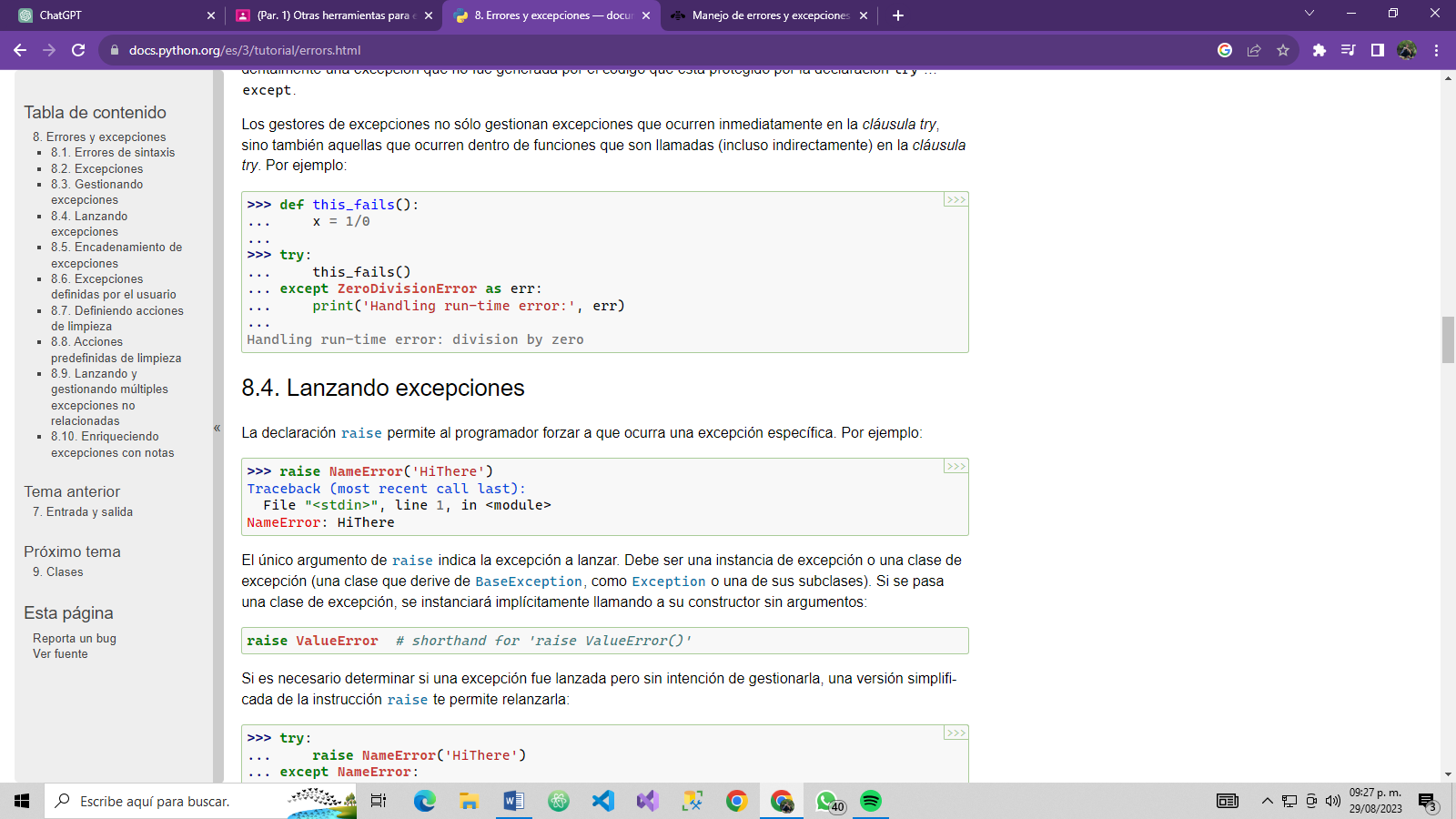
Donde se le solicita al usuario que introduzca un número y en caso de que ocurra algo que no está planeado por así decirlo no romperá el programa, sino que volverá a solicitar el numero hasta que el usuario lo ingrese de manera correcta por así decirlo.

La sentencia try funciona de la siguiente manera:

* Primero se ejecuta la sentencia try, si no ocurre ninguna excepción durante la ejecución try se omitirá el resto de la cláusula.
* En caso de detectar una excepción irá directamente a lo que establece la sentencia except.

Una declaración try puede tener más de una sentencia except para especificar diferentes excepciones.

También existe la sentencia raise lo cual permite al usuario lanzar excepciones específicas.



En conclusión:

Es importante tener en cuenta un buen manejo de errores dentro de nuestro software para así hacer más “amigable” el software para nuestro usuario.

No solo eso, sino que también nos ayuda a tener una mejor calidad de software con herramientas que nos ayudan a identificar, gestionar y corregir errores de manera más eficiente. Nos facilita la colaboración para poder trabajar de manera colectiva más sencillo y no tener problemas al momento de manejar distintas versiones. También nos ayuda a tener una mejor escalabilidad como un manejo de código más limpio y libre de errores a manera de que escala el proyecto y “evoluciona”.

# Bibliografía

* *8. Errores y excepciones*. (s. f.). Python documentation. https://docs.python.org/es/3/tutorial/errors.html
* *IBM documentation*. (s. f.-b). https://www.ibm.com/docs/es/integration-bus/10.0?topic=extension-errors-exception-handling