

| Plan Formativo Full Stack Python | |
|--|---------------------------------|
| Módulo | Programación Avanzada en Python |
| Tema | Errores y Excepciones |
| Nivel de Dificultad | Alta |
| Ejecución | Grupal |
| Duración | 60 Min |
| Código Ejercicio | E.3.5.G1 |
| Intención del aprendizaje o aprendizaje esperado: | |
| <ul style="list-style-type: none"> • Aprender las causas que generan gran número de excepciones posibles de encontrar en Python. • Manejo de excepciones con mensajes y acciones personalizadas. • Lograr continuidad de ejecución en programas frente a ocurrencia de errores. • Trabajo en grupo de desarrollo convergente con Git. | |
| Planteamiento del Problema: | |
| <p>En este ejercicio tomaremos el código final del ejercicio E.3.4.G1 y lo modificaremos para generar errores y formas de manejarlos. Consideremos el listado jerárquico de errores en Python que se muestra más abajo (Jerarquía de Excepciones Python).</p> <p>Parte 1) Generación de Excepciones</p> <p>Genere un programa principal que en base a preguntas al usuario en base a input() funcione en un loop infinito que consiste en un menú de opciones que permite crear condominios y modificar condominios existentes en base a manipulación de atributos y uso de sus métodos.</p> <p>Realice modificaciones al código de tal forma que induzca la aparición de diversidad de excepciones en su funcionamiento de la jerarquía mostrada en el anexo de este ejercicio. Debe generar al menos las siguientes cantidades y tipos de excepciones:</p> <pre>1xStopIteration, 1xAithmeticError, 1xAttributeError, 1xImportError, 1xLookupError, 2xSyntaxError, 2xTypeError, 3xValueError.</pre> | |

Genere una documentación en un archivo markdown con una guía de cómo provocar las excepciones correspondientes.

Parte 2) Manejo de Excepciones

Modifique el código con sentencias try/except/else/finally de tal forma que cada uno de las exceptions de la Parte 1 sean manejadas, para: 1) Asegurar la robustez del programa, y 2) Entregar mensajes claros al usuario sobre lo ocurrido, cómo ha sido resuelto o cómo debe reintentar el usuario en el ingreso de los datos.

Instrucciones generales

Debe trabajarse en forma grupal interviniendo el código de manera distribuida entre los integrantes del grupo. Deben asignarse roles de desarrolladores, revisores, testers, esquema de ramas de preferencia del grupo, y privilegios de intervención / push / merge de ramas. El código final debe estar en la rama master luego de resolución de eventuales conflictos y de haber sido probado.

ANEXO: Jerarquía de Excepciones Python

```
BaseException
+-- SystemExit
+-- KeyboardInterrupt
+-- GeneratorExit
+-- Exception
    +-- StopIteration
    +-- StopAsyncIteration
    +-- ArithmeticError
        |   +-- FloatingPointError
        |   +-- OverflowError
        |   +-- ZeroDivisionError
    +-- AssertionError
    +-- AttributeError
    +-- BufferError
    +-- EOFError
    +-- ImportError
        |   +-- ModuleNotFoundError
    +-- LookupError
        |   +-- IndexError
        |   +-- KeyError
    +-- MemoryError
    +-- NameError
        |   +-- UnboundLocalError
    +-- OSError
        |   +-- BlockingIOError
        |   +-- ChildProcessError
        |   +-- ConnectionError
        |       |   +-- BrokenPipeError
        |       |   +-- ConnectionAbortedError
        |       |   +-- ConnectionRefusedError
```

```
| |      +-- ConnectionResetError
| +-- FileExistsError
| +-- FileNotFoundError
| +-- InterruptedError
| +-- IsADirectoryError
| +-- NotADirectoryError
| +-- PermissionError
| +-- ProcessLookupError
| +-- TimeoutError
+-- ReferenceError
+-- RuntimeError
|   +-- NotImplementedError
|   +-- RecursionError
+-- SyntaxError
|   +-- IndentationError
|       +-- TabError
+-- SystemError
+-- TypeError
+-- ValueError
|   +-- UnicodeError
|       +-- UnicodeDecodeError
|       +-- UnicodeEncodeError
|       +-- UnicodeTranslateError
+-- Warning
    +-- DeprecationWarning
    +-- PendingDeprecationWarning
    +-- RuntimeWarning
    +-- SyntaxWarning
    +-- UserWarning
    +-- FutureWarning
    +-- ImportWarning
    +-- UnicodeWarning
    +-- BytesWarning
    +-- ResourceWarning
```

Recursos Bibliográficos :

[1] Módulo 3 - Contenido 5: "Codificar un programa Python utilizando generación y captura de errores y excepciones para el control del flujo durante excepciones"