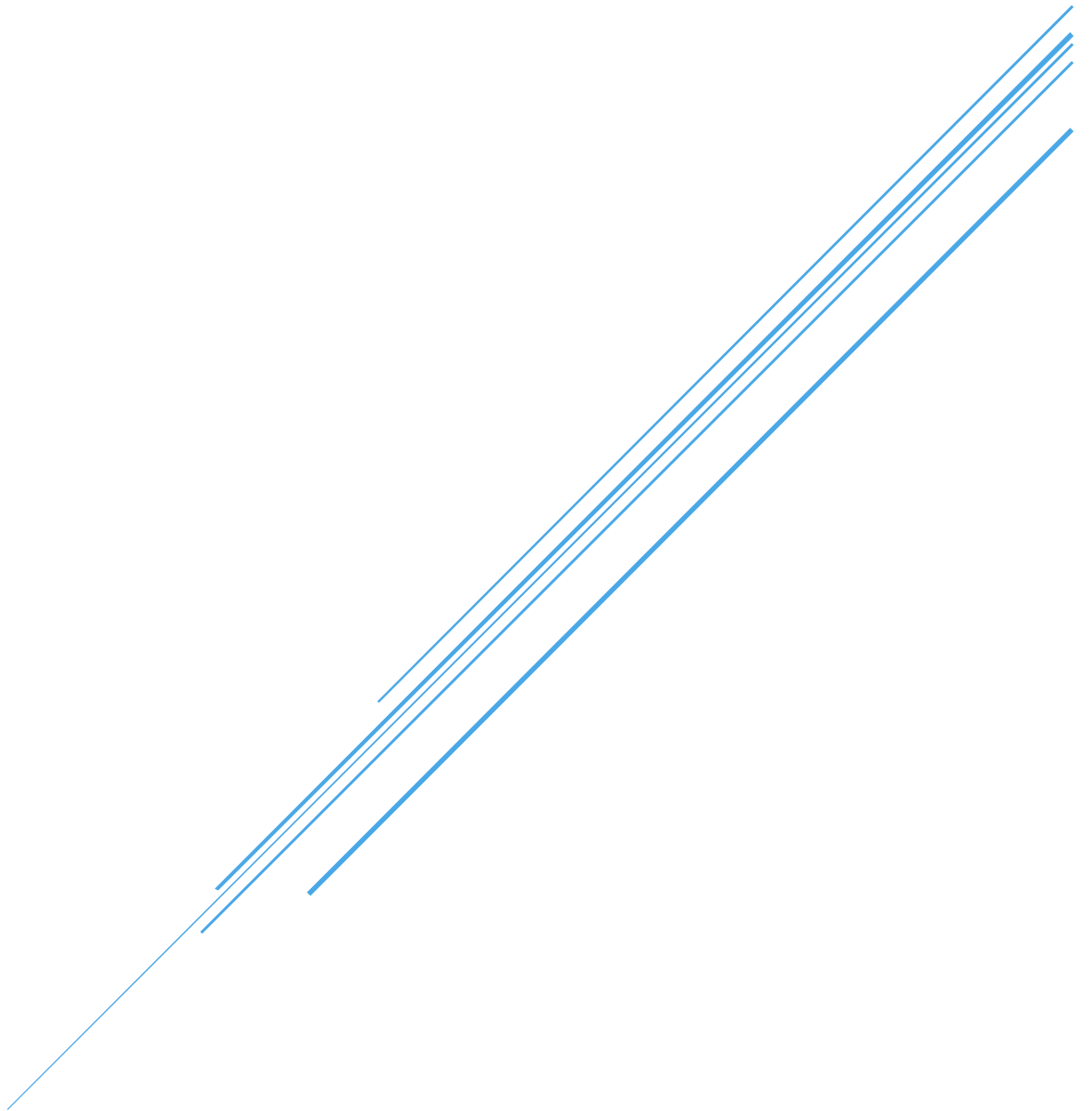


# SEMANA 4

Bootcamp Desarrollo Web



Daniel Bonilla González

**BUSCAR CUATRO CASOS EN LOS QUE SE PUEDA GENERAR ERROR EN PYTHON Y CACHAR EN CADA UNO DE ESTOS CASOS EL ERROR.**

**División por cero (ZeroDivisionError):** Intentar dividir un número por cero resulta en un error. Este error se maneja con un bloque try-except que captura ZeroDivisionError.

```
try:
    resultado = 10 / 0
except ZeroDivisionError as e:
    print(f"Error: {e}. No se puede dividir por cero.")
```

**Acceso a un índice fuera del rango de una lista (IndexError):** Intentar acceder a un índice que no existe en una lista genera un IndexError. Este error se maneja con un bloque try-except que captura IndexError.

```
mi_lista = [1, 2, 3]

try:
    elemento = mi_lista[5]
except IndexError as e:
    print(f"Error: {e}. El índice está fuera del rango de la lista.")
```

**Acceso a una clave inexistente en un diccionario (KeyError):** Intentar acceder a una clave que no existe en un diccionario produce un `KeyError`. Este error se maneja con un bloque `try-except` que captura `KeyError`.

```
mi_diccionario = {"clave1": "valor1", "clave2": "valor2"}

try:
    valor = mi_diccionario["clave3"]
except KeyError as e:
    print(f"Error: {e}. La clave no existe en el diccionario.")
```

**Llamada a una función con el número incorrecto de argumentos (TypeError):** Llamar a una función con un número incorrecto de argumentos produce un `TypeError`. Este error se maneja con un bloque `try-except` que captura `TypeError`.

```
def suma(a, b):
    return a + b

try:
    resultado = suma(1)
except TypeError as e:
    print(f"Error: {e}. La función requiere dos argumentos.")
```

## BUSCAR UN TEXTO EN EL QUE PUEDAS APLICAR 5 FUNCIONES DE LISTAS

```
texto = "Python es un lenguaje de programación popular y versátil. Es conocido por su simplicidad y legibilidad."

# Convertimos el texto a una lista de palabras
palabras = texto.split()

# 1. append(): Agregar una palabra al final de la lista
palabras.append("Ampliamente")
print("Después de append:", palabras)

# 2. insert(): Insertar una palabra en una posición específica
palabras.insert(1, "es")
print("Después de insert:", palabras)

# 3. remove(): Eliminar la primera aparición de una palabra específica
palabras.remove("Python")
print("Después de remove:", palabras)

# 4. sort(): Ordenar la lista de palabras alfabéticamente
palabras.sort()
print("Después de sort:", palabras)

# 5. count(): Contar cuántas veces aparece una palabra específica
contador_es = palabras.count("es")
print("La palabra 'es' aparece", contador_es, "veces.")
```