

Proceso de debugging

Después de poner un breakpoint en la primera línea se activa el modo debugging:

The screenshot shows a Python debugger interface. On the left, the 'VARIABLES' panel displays local variables: a=5, b=10, c=15, d=20. The code editor on the right has a breakpoint set on the first line of code, which is highlighted in yellow. The code is as follows:

```
# Paso 1: Operación matemática
a, b, c, d = 5, 10, 15, 20
suma = a + b + c + d # suma = 50

# Paso 2: División por cero con manejo de excepción
try:
    resultado = suma / 0
except ZeroDivisionError as e:
    print(f"⚠️ Error de división por cero: {e}")
else:
    print(f"✅ Resultado: {resultado}")
finally:
    print("👉 Fin del bloque de debugging.")
```

Segunda línea:

The screenshot shows the same Python debugger interface. Now, there is a red dot on the line 'a, b, c, d = 5, 10, 15, 20', indicating it is the current line being debugged. The code editor shows the same code as before, with the first line having a yellow background.

Tercera línea inicia el código de la división en cero:

The screenshot shows the Jupyter Notebook interface with the following code in the cell:

```
# Paso 1: Operación matemática
❶ a, b, c, d = 5, 10, 15, 20
suma = a + b + c + d # suma = 50 a = 5, b = 10, c = 15, d = 20

# Paso 2: División por cero con manejo de excepción
try:
    resultado = suma / ❷ 0 suma = 50
except ZeroDivisionError as e:
    print(f"⚠️ Error de división por cero: {e}")
else:
    print(f"✅ Resultado: {resultado}")
finally:
    print("🏁 Fin del bloque de debugging.")
```

The line `❷ 0` is highlighted in yellow, indicating it is the source of the error.

Cuarta línea:

The screenshot shows the Jupyter Notebook interface with the following code in the cell:

```
# Paso 1: Operación matemática
❶ a, b, c, d = 5, 10, 15, 20
suma = a + b + c + d # suma = 50 a = 5, b = 10, c = 15, d = 20

# Paso 2: División por cero con manejo de excepción
try:
    resultado = suma / ❷ 0 suma = 50
except ZeroDivisionError as e:
    print(f"⚠️ Error de división por cero: {e} e = ZeroDivisionError('division by zero')")
else:
    print(f"✅ Resultado: {resultado}")
finally:
    print("🏁 Fin del bloque de debugging.")
```

The line `❷ 0` is highlighted in yellow, indicating it is the source of the error.

Quinta línea:

The screenshot shows the Jupyter Notebook interface with the following code in the cell:

```
# Paso 1: Operación matemática
❶ a, b, c, d = 5, 10, 15, 20
suma = a + b + c + d # suma = 50 a = 5, b = 10, c = 15, d = 20

# Paso 2: División por cero con manejo de excepción
try:
    resultado = suma / ❷ 0 suma = 50
except ZeroDivisionError as e:
    print(f"⚠️ Error de división por cero: {e}")
else:
    print(f"✅ Resultado: {resultado}")
finally:
    print("🏁 Fin del bloque de debugging.")
```

The line `❷ 0` is highlighted in yellow, indicating it is the source of the error. At the bottom of the screen, an error message is displayed:

... ⚠️ Error de división por cero: division by zero

Se muestra el resultado final:

The screenshot shows a Jupyter Notebook interface with the following details:

- Top Bar:** RUN AND DEBUG, Python D..., Welcome, Tarea M27-AD – Jose Alberto Hernandez Moran.ipynb, Debugging.ipynb.
- VARIABLES Panel:** Shows variables `a`, `b`, `c`, `d` with values 5, 10, 15, 20 respectively.
- Code Cell:** Contains Python code for summing variables and performing division with exception handling.

```
# Paso 1: Operación matemática
a, b, c, d = 5, 10, 15, 20
suma = a + b + c + d # suma = 50

# Paso 2: División por cero con manejo de excepción
try:
    resultado = suma / 0
except ZeroDivisionError as e:
    print(f"⚠️ Error de división por cero: {e}")
else:
    print(f"✅ Resultado: {resultado}")
finally:
    print("🏁 Fin del bloque de debugging.")
```
- Output Cell:** Displays the result of the division attempt, showing an error message and the end of the debugging block.

```
... ⚠️ Error de división por cero: division by zero
🏁 Fin del bloque de debugging.
```