



Depuración

Informática I - 2547100

Departamento de Ingeniería Electrónica
y de Telecomunicaciones

Facultad de Ingeniería

2015-2

Properties of bugs

evidente vs. oculto

manifestación **obvia**: la ejecución del programa se interrumpe o se demora exageradamente

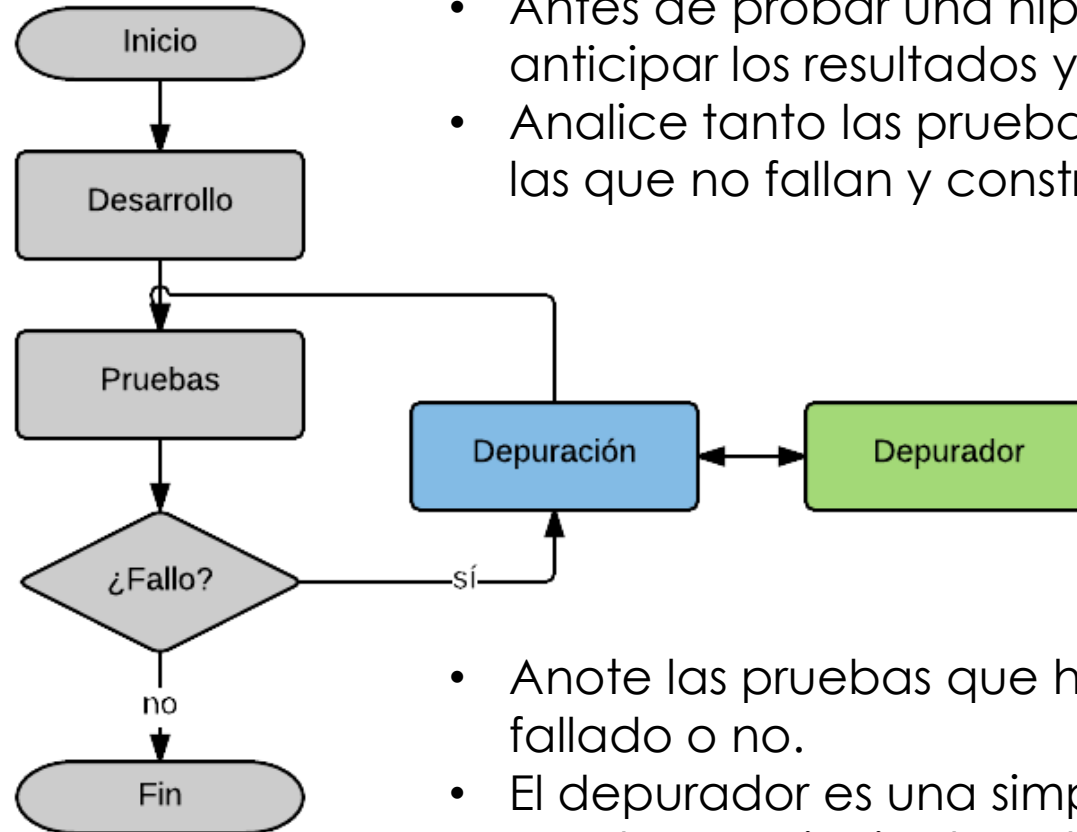
manifestación **no obvia**: ejecución normal del programa pero resultados incorrectos ante algunas entradas

persistente vs. intermitente

cuando el fallo se observa **siempre** que el programa se ejecuta con las mismas entradas

cuando el fallo se observa solo algunas veces, a pesar de ejecutarlo (aparentemente) con las mismas entradas

Debugging methodology



- Antes de probar una hipótesis, trate de anticipar los resultados y sus implicaciones.
- Analice tanto las pruebas que fallan como las que no fallan y construya hipótesis.

- Anote las pruebas que ha hecho y si han fallado o no.
- El depurador es una simple herramienta de ayuda. Lo principal es el análisis que se haga de la prueba que falló.

Debugging example

```
def isPal(x):  
    '''Asume que x es una lista  
    Retorna True si x es palíndroma o False en caso contrario'''  
    temp = x[:]  
    temp.reverse()  
    if temp == x:  
        return True  
    else:  
        return False
```

```
def silly(n):  
    '''Asume que n es un entero mayor que 0  
    Obtiene n elementos del usuario e intenta determinar si la lista es palíndroma o No de lo contrario'''  
    for a in range(n):  
        result = []  
        elem = input('Ingrese elemento: ')  
        result.append(elem)  
    if isPal(result):  
        print('Sí')  
    else:  
        print('No')
```

```
>>> silly(2)  
Ingrese elemento: a  
Ingrese elemento: b  
Sí  
  
>>> silly(2)  
Ingrese elemento: a  
Ingrese elemento: b  
['b']  
Sí  
  
>>> silly(2)  
Ingrese elemento: a  
Ingrese elemento: b  
['a', 'b']  
Sí  
  
>>> silly(2)  
Ingrese elemento: a  
Ingrese elemento: b  
['a', 'b'] ['a', 'b']  
Sí  
  
>>> silly(2)  
Ingrese elemento: a  
Ingrese elemento: b  
['b', 'a'] ['b', 'a']  
Sí  
  
>>> silly(2)  
Ingrese elemento: a  
Ingrese elemento: b  
['a', 'b'] ['b', 'a']  
No
```



Debugging hints

Pasar argumentos a una función en el orden equivocado	<code>func(b, a) ≠ func(a, b)</code>
Confundir mayúsculas con minúsculas	<code>func(a, b) ≠ Func(a, b)</code>
Olvidar dar un valor inicial a una variable	<code>cont = 0</code>
Esperar que dos valores <code>float</code> sean exactamente iguales	<code>if a==b: ≠ if abs(a-b)<0.0001:</code>
Olvidar los paréntesis al invocar una función o método	<code>str1.split ≠ str1.split()</code>
Vincular dos variables a un mismo dato en lugar de clonaras (copiarlas)	<code>a = b ≠ a = b[:]</code>

Eclipse debugger

Debug - test/main.py - Eclipse

Debug

- MainThread - pid4304_seq2
 - silly [main.py:26]
 - <module> [main.py:32]

Breakpoints

- silly [main.py:26]

No details to display for the current selection.

Outline (x)= Variables Expressions Multicore Visualizer

Name	Value
Globals	Global variables
• a	int: 1
• elem	str: b
• n	int: 2
• result	list: []

Console

```
main.py
pydev debugger: starting (pid: 4304)
Ingrese elemento: a
Ingrese elemento: b
Traceback (most recent call last):
>>>
```

Writable Insert 27: 22