

Depuración de Programas

Introducción a la Programación

Departamento de Ciencias Básicas, UNLu



Cómo NO se programa

- 1 Escribo el programa en la computadora.
- 2 Continúo haciendo el programa hasta que el compilador me dice que no tengo más errores.
- 3 Listo, terminé!!

Cómo Sí se programa

- 1 Escribo el programa en la computadora.
- 2 Continúo haciendo el programa hasta que el compilador me dice que no tengo más errores.
- 3 Lo testeamos!!

Recordar: *Las computadoras no son capaces de leer el pensamiento del programador!! Simplemente ejecutan las instrucciones del programa.*

Tipos de errores

- *Sintácticos*: no cumple con las reglas sintácticas (los detecta el compilador o intérprete, y si hay error de sintaxis no permite la ejecución)
- *De tiempo de ejecución*: el programa no tiene errores sintácticos, pero cuando se ejecuta se corta la ejecución (por ejemplo, división por cero). Se detecta en tiempo de ejecución.
- *Semánticos*: El programa hace algo distinto a lo que se buscaba que hiciera. Los detectamos ejecutando varias veces el programa (con una batería de tests).

Testing básico

Hay muchas técnicas y tipos de testing.

Un testing básico consiste en hacer una batería de casos de tests simples, ejecutar el programa y comprobar que el programa se comporta en la forma que queremos.

Ejemplo: Hacemos el programa elevar al cuadrado, que recibe un número y devuelve su cuadrado. La batería de tests es un conjunto de casos sencillos que trata de cubrir las posibles entradas. Una batería de casos de tests para este problema podría ser: el cuadrado de -10 , de cero, de uno, y de $+10$.

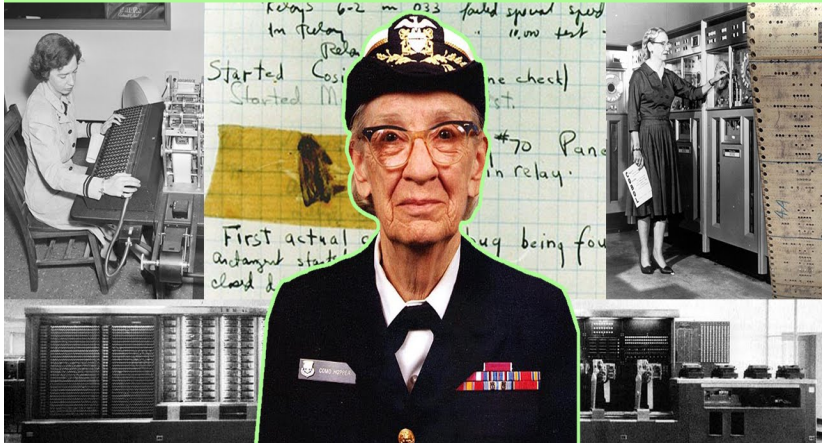
Debugging

Si el programa no devuelve el resultado esperado cuando lo probamos con los tests, entonces debemos buscar el error del programa y corregirlo.

A un error en el programa, lo denominamos **bug** (que significa "bicho" en inglés). Al proceso de buscar los errores del programa y corregirlos, lo llamamos **debugging** (que significa en inglés "eliminar los bichos")

Primer bug de la historia de la computación

ADMIRAL "AMAZING GRACE" HOPPER



Testing y Debugging



Una computadora es una máquina que almacena y manipula información, bajo el control de un programa. Entonces, si el caso de test no me da la salida esperada para ese caso de test, para encontrar el bug tengo que fijarme que está haciendo el programa con los datos del caso de test instrucción por instrucción.

Estado y Computación

A lo largo de la ejecución de un programa las variables pueden cambiar el valor con el que están asociadas. En un momento dado uno puede detenerse a observar a qué valor se refiere cada una de las variables del programa. Esa "foto" que indica en un momento dado **a qué valor hace referencia cada una de las variables** se denomina **estado**. También hablaremos del **estado de una variable** para indicar a qué valor está asociada esa variable.

A medida que las variables cambian de valores a los que se refieren, el programa va cambiando de estado. **La sucesión de todos los estados por los que pasa el programa en una ejecución dada se denomina computación.**

Testing y Debugging



Si el caso de test no me da la salida esperada, tengo que fijarme que está haciendo el programa con los datos del caso de test instrucción por instrucción.

Fase inicial de testing: sólo me fijo en el estado inicial y en el estado final del caso de test.

Fase de debugging: para el caso de test que me falla, voy a observar toda la computación (el estado inicial, la sucesión de los estados intermedios, y el estado final).

Depuración de Programas

Una manera de seguir la evolución del estado es **insertar instrucciones de impresión en sitios críticos del programa**. Esto nos será de utilidad para detectar errores y también para comprender cómo funcionan determinadas instrucciones.

Depuración de Programas: print

```
def imprimir_cuadrados():  
    print("Se calcularán cuadrados de números")  
  
    n1 = int(input("Ingrese un número entero: "))  
    print("el valor de n1 es:", n1)  
    n2 = int(input("Ingrese otro número entero: "))  
    print("el valor de n2 es:", n2)  
  
    for x in range(n1, n2):  
        print("el valor de x es:", x)  
        print(x * x)  
  
    print("Es todo por ahora")  
  
imprimir_cuadrados()
```

Depuración de Programas: print

```
$ python cuadrados.py
Se calcularán cuadrados de números
Ingrese un número entero: 5
el valor de n1 es: 5
Ingrese otro número entero: 8
el valor de n2 es: 8
el valor de x es: 5
25
el valor de x es: 6
36
el valor de x es: 7
49
Es todo por ahora
```

Depuración de Programas: debugger de algún IDE

Escribir muchos "prints" puede ser muy trabajoso. Además, después en la versión final del programa, hay que borrar o comentar los "prints" que pusimos para debuggear.

Un IDE (integrated development environment) es una aplicación de software que nos provee de herramientas útiles para facilitar la tarea del desarrollo de software. Un IDE normalmente consiste de al menos un editor de código fuente, herramientas automáticas para construir código, y un **debugger**.

El IDE que estamos usando en la materia es el Visual Studio Code.

Depuración de Programas: debugger en VS Code

The screenshot shows the Visual Studio Code interface with a Python program named `cuadrados.py` being debugged. The program calculates the squares of numbers from `n1` to `n2`.

Variables Panel:

- Locals:**
 - `n1`: 5
 - `n2`: 8
 - `x`: 5
- __builtins__:** {'Arithm...', '__cached__': None, '__doc__': None, '__file__': 'c:\\Users\\rmatu\\...', '__loader__': None, '__name__': '__main__', '__package__': '', '__spec__': None}

Watch Panel:

- (Empty)

Call Stack Panel:

- PAUSED ON STEP
- <module> cuadrados.py

Breakpoints Panel:

- ☒ Raised Exceptions
- ☒ Uncaught Exceptions
- ☒ cuadrados.py 1

Code Editor:

```
1 print("Se calcularán cuadrados de números")
2 n1 = int(input("Ingrese un número entero: "))
3 n2 = int(input("Ingrese otro número entero: "))
4
5 for x in range(n1, n2):
6     print(x * x)
7
8 print("Es todo por ahora")
```

Output/Terminal/Debug Console:

```
PS C:\Users\rmatu\veamos> cd 'c:\Users\rmatu\veamos'; ${env:PYTHONIOENCODING}='UTF-8'; ${env:PYTHONUNBUFFERED}='1'; & 'C:\Users\rmatu\AppData\Local\Programs\Python\Python38-32\python.exe' 'c:\Users\rmatu\.vscode\extensions\ms-python.python-2020.3.69010\pythonFiles\ptvsd_launcher.py' '--default' '--client' 'localhost' '--port' '50121' 'c:\Users\rmatu\veamos\cuadrados.py'
Se calcularán cuadrados de números
Ingrese un número entero: 5
Ingrese otro número entero: 8
```

Status Bar:

- master*
- Python 3.8.1 32-bit
- Ln 6, Col 1 Spaces: 4 UTF-8 CRLF Python

Depuración de Programas: ejecutando el debugger del VS Code

