

# CONCEPTOS GENERALES



## **Introducción a la Programación (11071)**

Departamento de Ciencias Básicas  
Universidad Nacional de Luján



# MATERIAL RELACIONADO

## TEORIA



**Teórica 1** - Introducción y elementos básicos.



**Apunte**

**Cap.**

**1**

-

**Conceptos**

**básicos.**

## PRÁCTICA



**TP I** - Conceptos Teóricos Básicos.



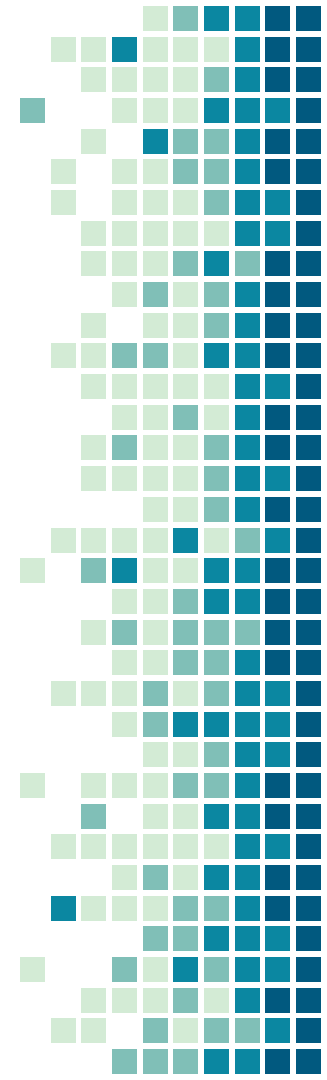
**TP II** - Introducción a Python.



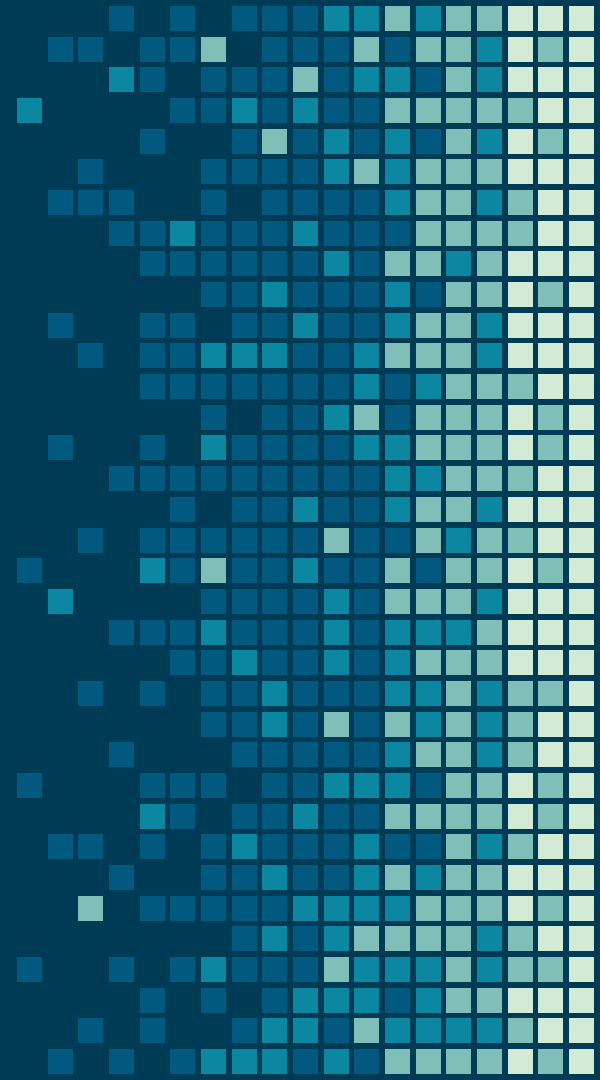
**Notebook 0** - Empezando a programar.

## ENTREGABLES

*No posee.*



# REPASO DE CONCEPTOS BÁSICOS



# CONCEPTOS BÁSICOS

## SOFTWARE

Conjunto de instrucciones y datos que definen el comportamiento de un programa de computadora.

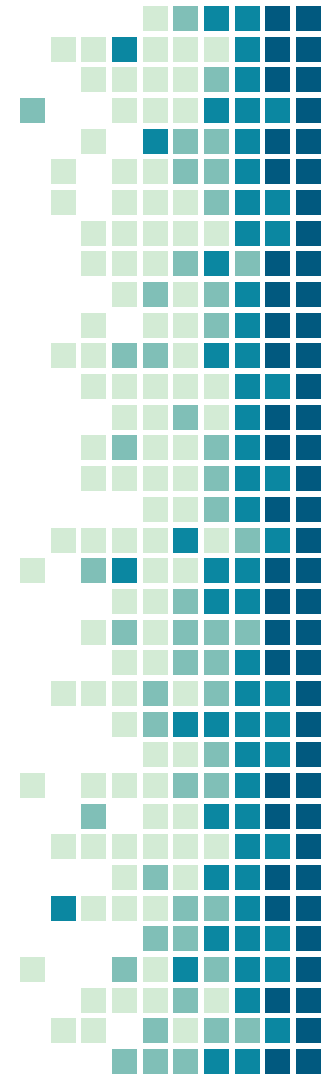
## PROGRAMACIÓN

La actividad de crear software.

## CÓDIGO

## FUENTE

Código escrito en un lenguaje de programación que, una vez compilado o interpretado por la computadora, genera un programa ejecutable.



# CONCEPTOS BÁSICOS (CONT.)

## INSTRUCCIÓN

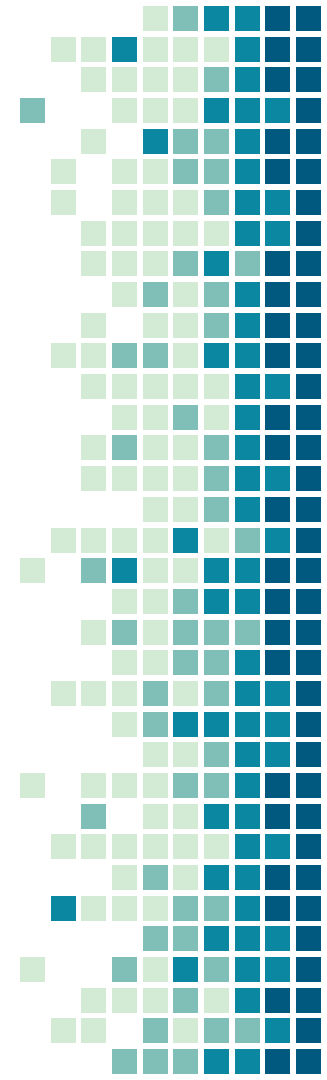
Acción concreta que la máquina sabe ejecutar.

## ALGORITMO

“Receta” o conjunto de pasos a seguir para resolver una tarea en particular.

## LENGUAJE DE PROGRAMACIÓN

“Idioma” en el que se escriben los algoritmos para que la máquina pueda interpretarlos o compilarlos. Cada lenguaje tiene su propia **sintaxis**.



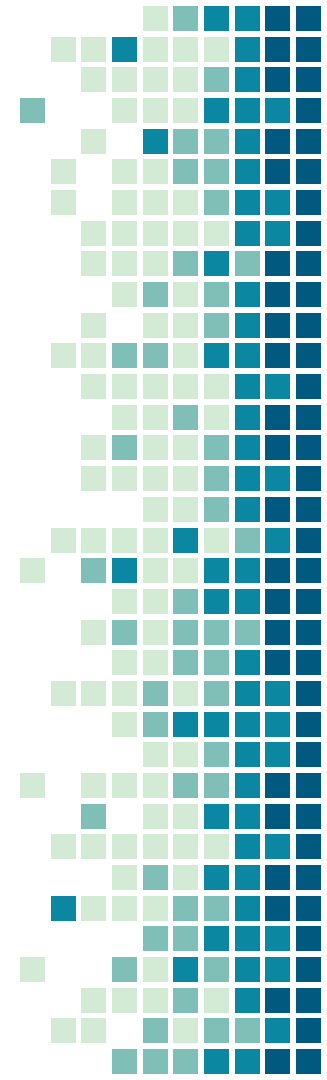
# CONCEPTOS BÁSICOS (CONT.)

## PARADIGMA DE PROGRAMACIÓN

Filosofía o ideales que definen cómo debe codificarse correctamente el software.

## PROGRAMACIÓN ESTRUCTURADA/PROCEDURAL

Paradigma de programación que sostiene que un programa es un conjunto de algoritmos que se ejecutan para resolver un problema o brindar un servicio.



# CONCEPTOS BÁSICOS (CONT.)

## PARADIGMA DE PROGRAMACIÓN

Filosofía o ideales que definen cómo debe codificarse correctamente el software.

## PROGRAMACIÓN

## ESTRUCTURADA/PROCEDURAL

Paradigma de programación que sostiene que un programa es un conjunto de algoritmos que se ejecutan para resolver un problema o brindar un servicio.



*¡El que vamos a utilizar en la asignatura!*

# TIPOS DE LENGUAJES

Una característica importante de los lenguajes de programación es que pueden ser:

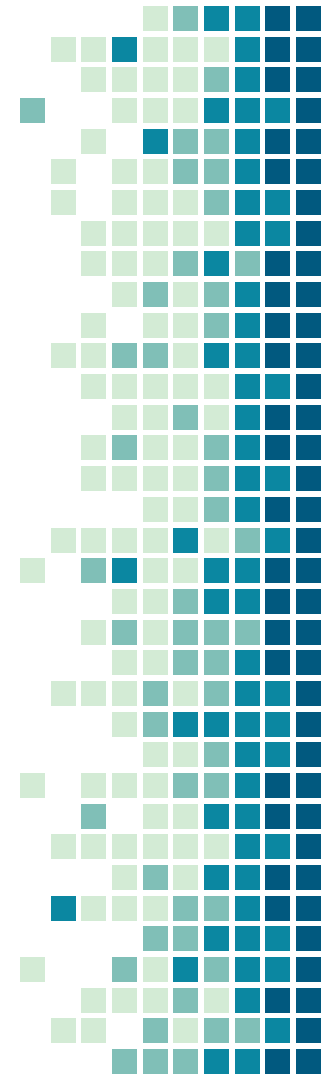
## COMPILADOS

El código fuente es transformado a un lenguaje comprensible por la máquina para dar como resultado un programa ejecutable.

... o bien ...

## INTERPRETADOS

El código fuente no se transforma, sino que es interpretado en tiempo real (por un *intérprete*) para poder ejecutarse.





# COMPILADORES



El programador escribe el *código fuente* del programa utilizando un lenguaje de programación en particular.

El código fuente es analizado por un programa *compilador*, que se encarga de traducirlo a instrucciones legibles por la máquina.

Si el compilador no detecta errores, se genera una versión ejecutable (o compilada) del programa.

# INTÉRPRETES



El programador escribe el *código fuente* del programa utilizando un lenguaje de programación en particular.



El código fuente *no* se compila, sino que es interpretado en tiempo real por un *intérprete*.



El código fuente no se transforma para ser ejecutado, sino que es ejecutado por intermediación del intérprete.



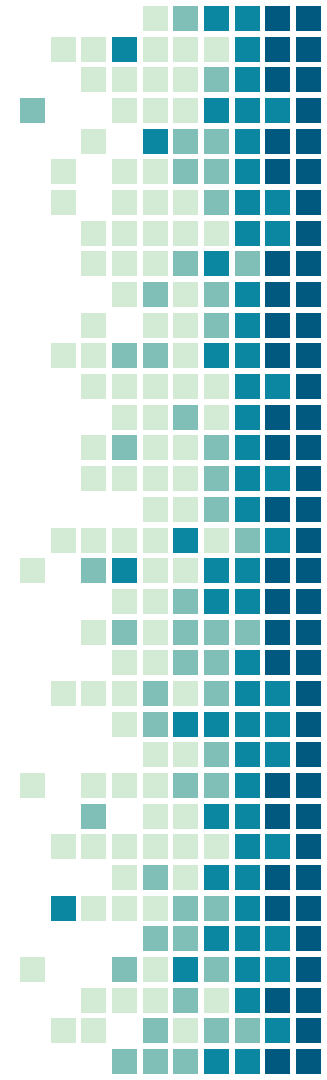
# EJEMPLOS DE LENGUAJES

## COMPILADOS

- Ada.
- Basic.
- Pascal.
- C, C++, C#.
- Java.

## INTERPRETADOS

- JavaScript.
- PHP.
- Smalltalk.
- Python.
- Lisp.



# EJEMPLOS DE LENGUAJES

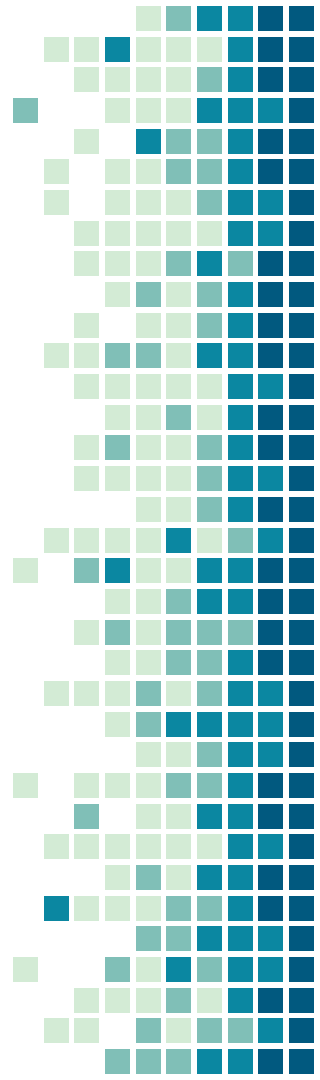
## COMPILADOS

- Ada.
- Basic.
- Pascal.
- C, C++, C#.
- Java.

## INTERPRETADOS

- JavaScript.
- PHP.
- Smalltalk.
- **Python.**
- Lisp.

*¡El que vamos a utilizar en la asignatura!*



# PYTHON

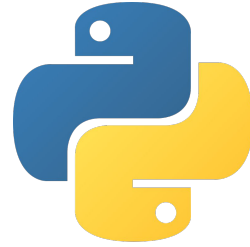


## CARACTERÍSTICAS GENERALES

- Creado por Guido van Rossum. La primera versión data de diciembre de 1989.
- Versión actual: 3.9 (a febrero del 2021).
- Multipropósito.
- Multiparadigma.
- Tipado *fuerte* y *dinámico* (más adelante veremos qué significa).

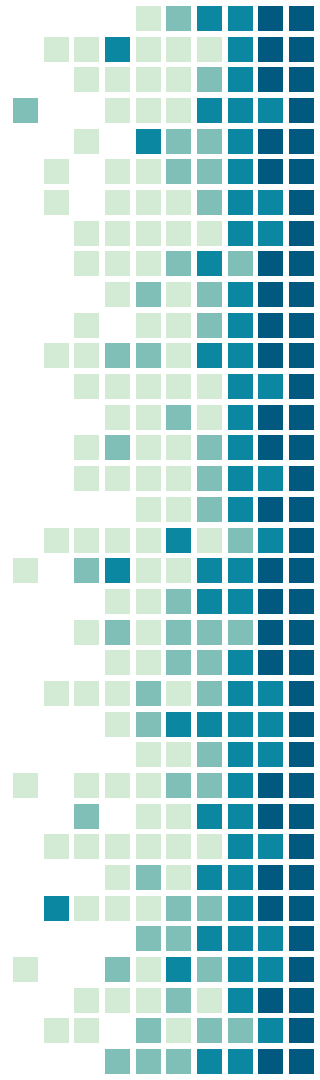


# PYTHON

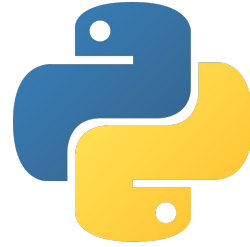


## LENGUAJE HÍBRIDO

- Desde su concepción más básica, se lo puede considerar un **lenguaje interpretado**.
- En realidad, Python compila el código, pero no a una versión ejecutable, sino a un código intermedio llamado *bytecode*.
- El *bytecode* no es ejecutable por la máquina, sino que Python utiliza una máquina virtual para ejecutarlo.

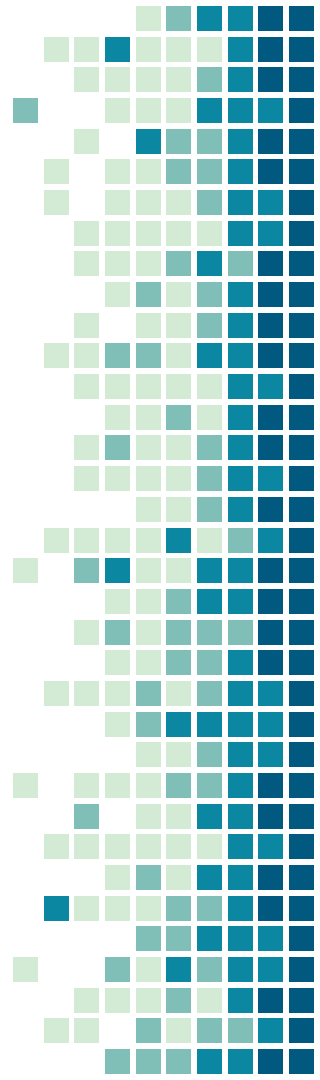


# PYTHON



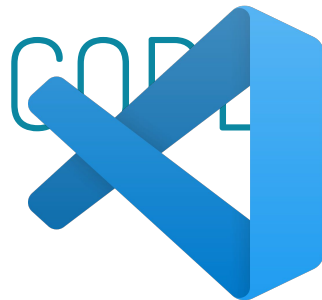
## INFORMACIÓN

- Descargable desde su sitio oficial: [www.python.org](http://www.python.org).
- Incluye el *intérprete*, una *terminal de comandos*, y una *IDE* (Integrated Development Environment) simple.
- Se puede utilizar cualquier editor de texto/código para codificar.



# VISUAL STUDIO

<https://code.visualstudio.com/>



## ¿QUÉ ES?

- Es un **IDE (Entorno de Desarrollo Integrado)**.
- Un entorno de desarrollo integrado tiene todas las herramientas y funcionalidad necesaria para poder codificar en determinado lenguaje de programación.
- Es independiente del lenguaje de programación en sí (en este caso, de Python).

## ¿ES OBLIGATORIO USARLO?

- No. Como programadores, somos libres de codificar con la herramienta que prefiramos (¡incluso con el Bloc de Notas!).



# GIT

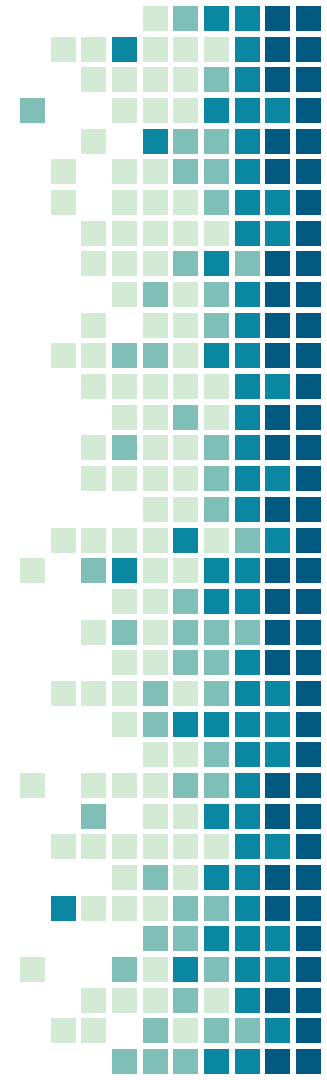


## ¿QUÉ ES?

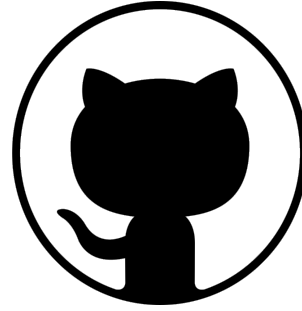
- Es un **sistema de versionado de código (VCS)**.

## ¿POR QUÉ LO NECESITAMOS?

- Acompaña la evolución del proyecto, permitiendo “recordar” cada una de sus versiones.
- Facilita el trabajo el grupo (varias personas manipulando el mismo código simultáneamente).



# GITHUB

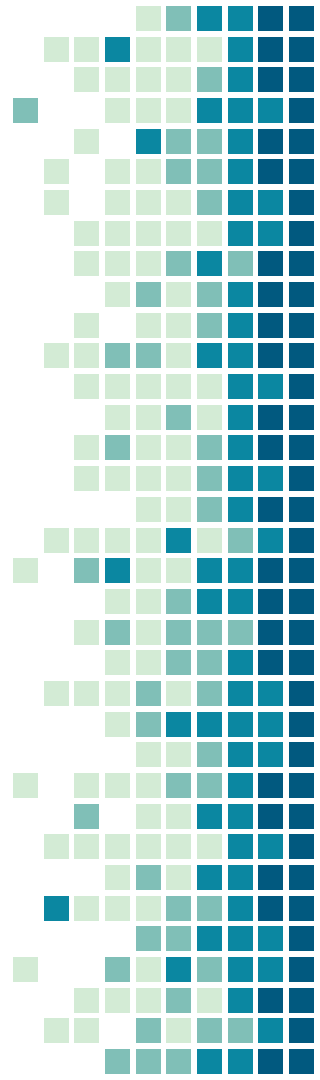


## ¿QUÉ ES?

- Es una plataforma online para hostear repositorios de Git.

## ¿POR QUÉ LO NECESITAMOS?

- Permite tener repositorios Git “en la nube” de manera gratuita.
- Las entregas de la asignatura se realizarán mediante la plataforma GitHub (preferentemente).



# RESUMEN DE TECNOLOGÍAS

## PYTHON

Lenguaje de programación que utilizaremos en la asignatura para codificar algoritmos en máquina.

## VISUAL

## STUDIO

## CODE

IDE que recomendamos para escribir código.

## GIT

Herramienta de control de versiones que utilizaremos para el seguimiento de nuestro código.

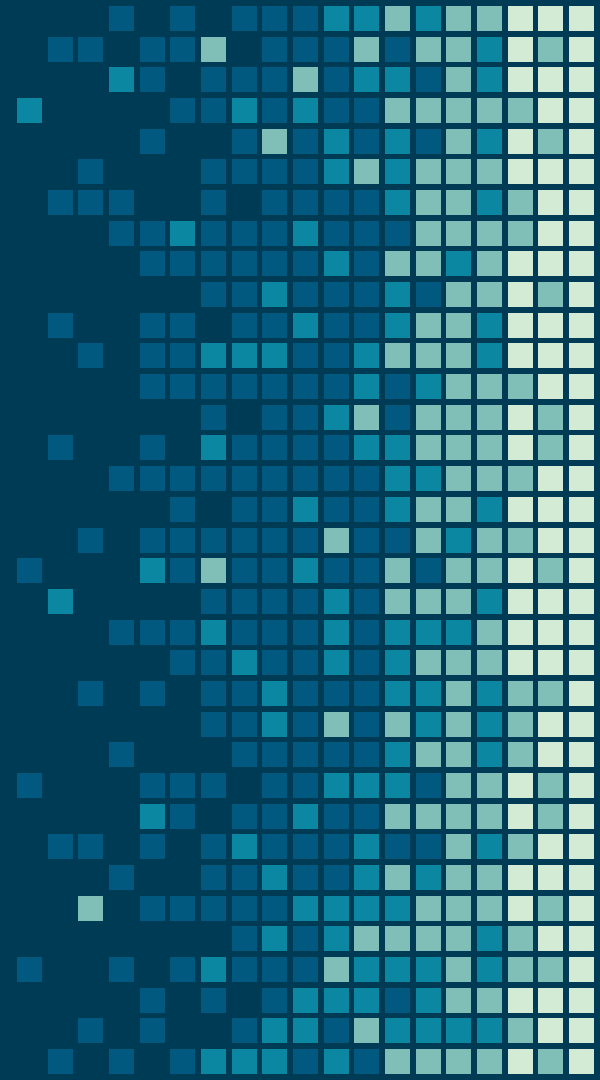
## GITHUB

Implementación de Git en la nube, para tener repositorios en la Web.



# EJERCICIOS GUIADOS

*Para resolución de TP I y TP II.*



# EJERCICIO MODELO (TP I)

## OBJETIVO DEL EJERCICIO

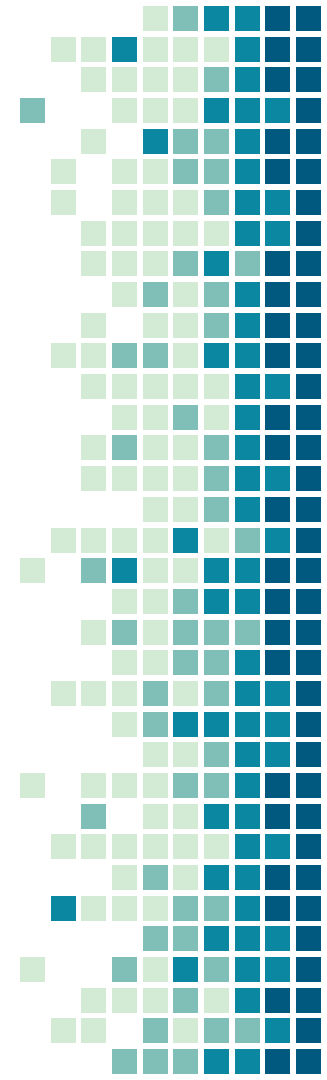
Comprender y ejercitar el concepto de **algoritmo**.

## ENUNCIADO

Utilizando lenguaje natural, confeccione un algoritmo para reemplazar la batería de un teléfono celular (elija un modelo concreto, a modo de ejemplo utilizaremos iPhone 6).

## TIPS

Recordar que un algoritmo es una secuencia de pasos a seguir para resolver una tarea o brindar un servicio. Escribir una lista ordenada de pasos para lograr lo que pide el enunciado.

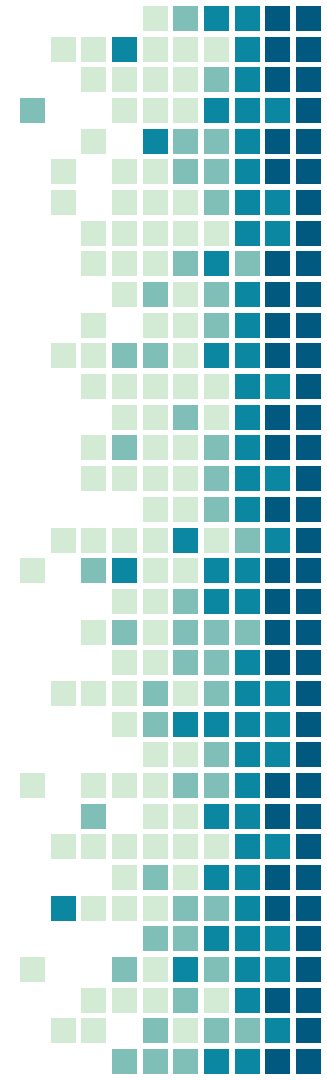


# EJERCICIO MODELO (TP I)

**SOLUCIÓN**  
INICIO

DEL

ALGORITMO



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina su





# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.
4. Desconectar el Flex de la pantalla.



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.
4. Desconectar el Flex de la pantalla.
5. Quitar la batería vieja.



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.
4. Desconectar el Flex de la pantalla.
5. Quitar la batería vieja.
6. Colocar la batería nueva posicionando la cinta de doble cara.



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.
4. Desconectar el Flex de la pantalla.
5. Quitar la batería vieja.
6. Colocar la batería nueva posicionando la cinta de doble cara.
7. Colocar el Flex de la batería nueva.



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.
4. Desconectar el Flex de la pantalla.
5. Quitar la batería vieja.
6. Colocar la batería nueva posicionando la cinta de doble cara.
7. Colocar el Flex de la batería nueva.
8. Colocar la chapa protectora del Flex con sus tornillos correspondientes.



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.
4. Desconectar el Flex de la pantalla.
5. Quitar la batería vieja.
6. Colocar la batería nueva posicionando la cinta de doble cara.
7. Colocar el Flex de la batería nueva.
8. Colocar la chapa protectora del Flex con sus tornillos correspondientes.
9. Cerrar el teléfono colocando los dos tornillos inferiores.



# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.
4. Desconectar el Flex de la pantalla.
5. Quitar la batería vieja.
6. Colocar la batería nueva posicionando la cinta de doble cara.
7. Colocar el Flex de la batería nueva.
8. Colocar la chapa protectora del Flex con sus tornillos correspondientes.
9. Cerrar el teléfono colocando los dos tornillos inferiores.

### FIN DEL ALGORITMO





# EJERCICIO MODELO (TP I)

## SOLUCIÓN

### INICIO DEL ALGORITMO

1. Quitar los dos tornillos inferiores.
2. Quitar la pantalla haciendo palanca sobre la esquina superior izquierda.
3. Quitar los cinco tornillos de la chapa protectora del Flex.
4. Desconectar el Flex de la pantalla.
5. Quitar la batería vieja.
6. Colocar la batería nueva posicionando la cinta de doble cara.
7. Colocar el Flex de la batería nueva.
8. Colocar la chapa protectora del Flex con sus tornillos correspondientes.
9. Cerrar el teléfono colocando los dos tornillos inferiores.

### FIN DEL ALGORITMO

## NOTAR QUE

- El algoritmo tiene un inicio y fin bien definidos.
- Cada paso debe realizarse en el orden especificado, y debe ser una acción clara y concisa.
- Es recomendable utilizar verbos en infinitivo (*quitar, desconectar, colocar, cerrar, etcétera*).



# EJERCICIO MODELO (TPII)

## OBJETIVO DEL EJERCICIO

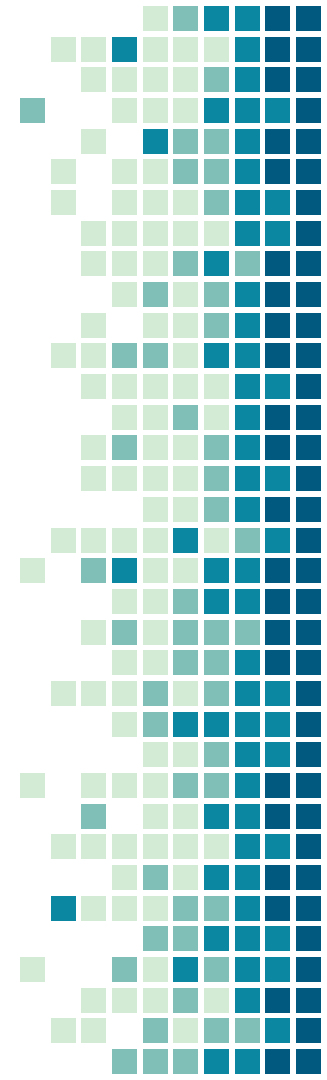
Escribir un algoritmo básico en Python para comenzar a familiarizarnos con el lenguaje.

## ENUNCIADO

Escriba un programa en Python que permita almacenar dos valores enteros (arbitrarios, por ejemplo, 47 y 68), así como el resultado de su suma. Luego, el programa debe mostrar en pantalla el siguiente mensaje:

*El resultado de sumar  $X + Y$  dio como resultado  $Z$ .*

Procure reemplazar  $X$ ,  $Y$ , y  $Z$  por sus valores correspondientes.



# EJERCICIO MODELO (TPII)

SOLUCIÓN

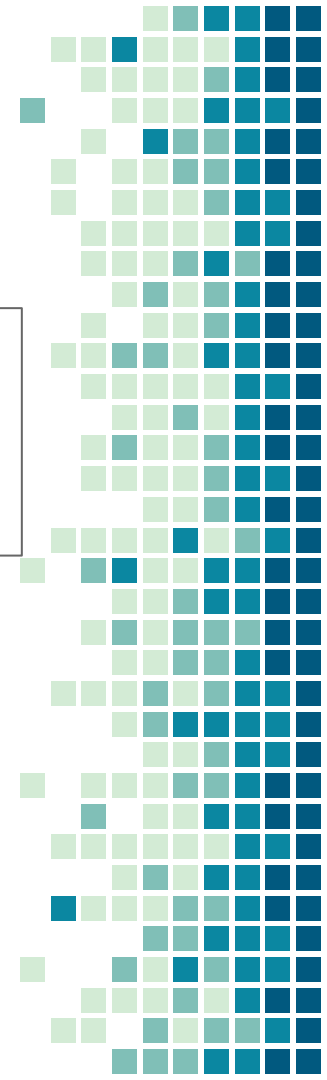


# EJERCICIO MODELO (TPII)

## SOLUCIÓN

### PROBLEMA

¿Cómo o dónde “almacenamos”  
un número entero?



# EJERCICIO MODELO (TPII)

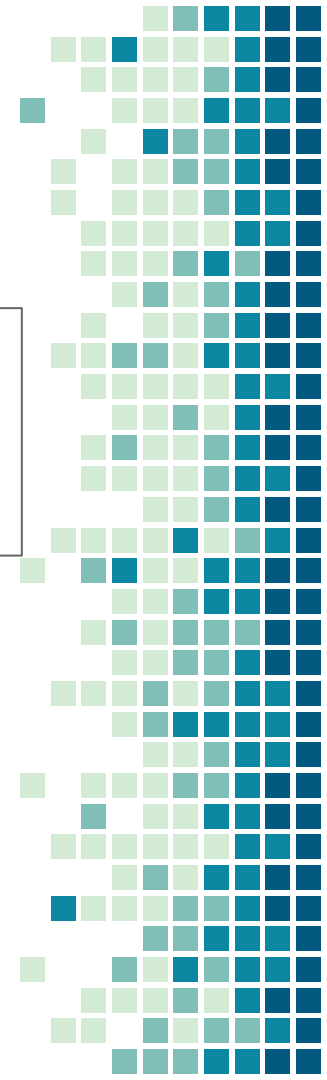
## SOLUCIÓN

### PROBLEMA

¿Cómo o dónde “almacenamos”  
un número entero?

### SOLUCIÓN

En una variable.



# EJERCICIO MODELO (TPII)

## SOLUCIÓN

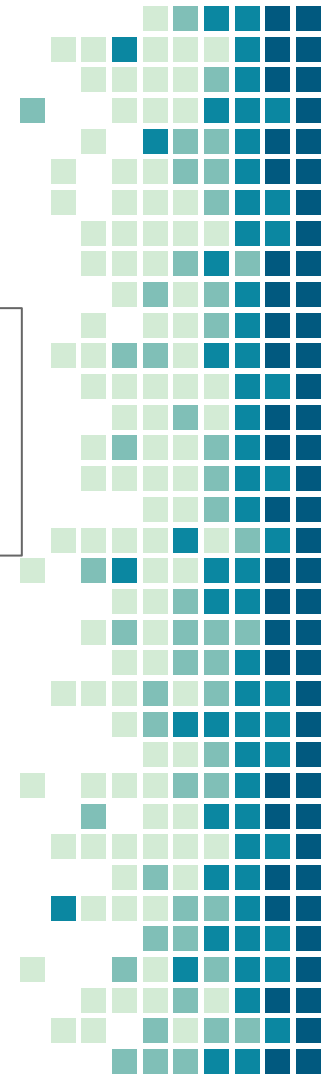
```
# variables para almacenar los datos requeridos  
numero1 = 47  
numero2 = 68
```

### PROBLEMA

¿Cómo o dónde “almacenamos”  
un número entero?

### SOLUCIÓN

En una variable.



# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68
```

Identificador de la variable

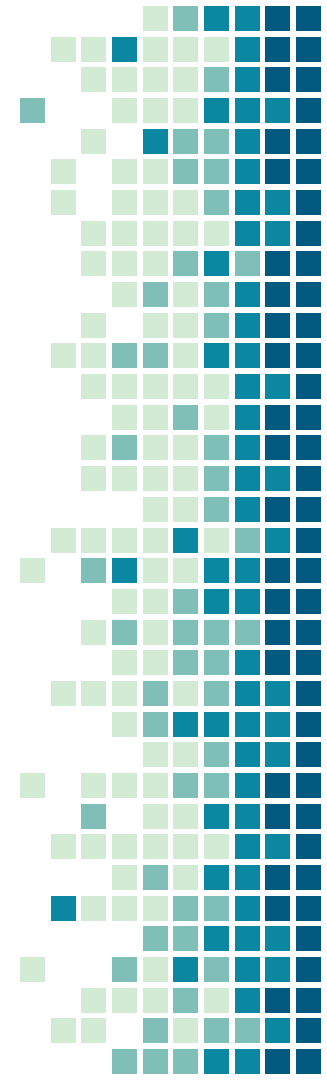


# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos  
numero1 = 47  
numero2 = 68
```

Operación de asignación



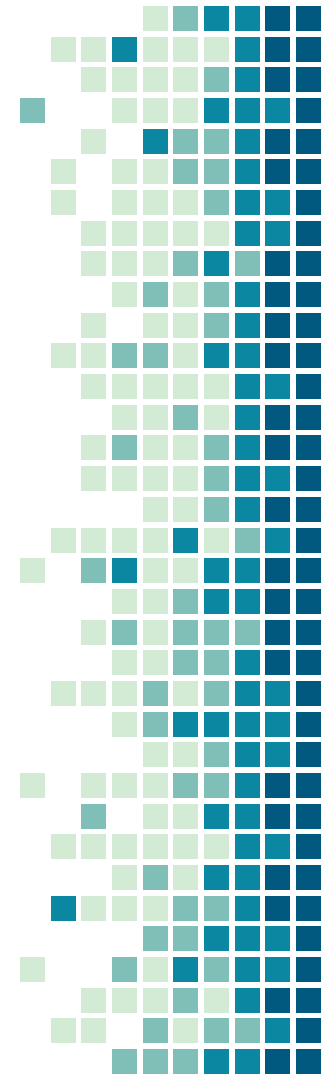


# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos  
numero1 = 47  
numero2 = 68
```

Valor asignado a la variable



# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos  
numero1 = 47  
numero2 = 68
```



# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = ?
```

## PROBLEMA

La suma no es un valor arbitrario, sino que es el resultado de una operación aritmética entre los valores de otras dos variables.

# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2
```

### PROBLEMA

La suma no es un valor arbitrario, sino que es el resultado de una operación aritmética entre los valores de otras dos variables.

### SOLUCIÓN

Utilizamos la operación de suma, y asignamos su resultado a una nueva variable.

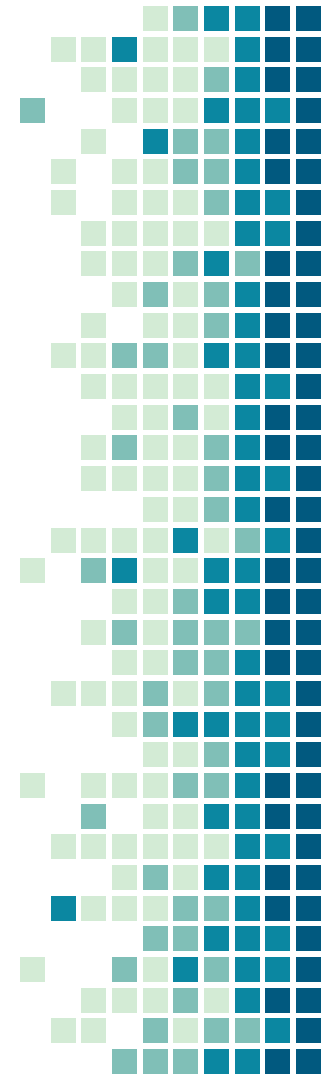
# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos  
numero1 = 47  
numero2 = 68
```

```
# variable para almacenar el resultado de la suma  
suma = numero1 + numero2
```

Identificador de la variable



# EJERCICIO MODELO (TPII)

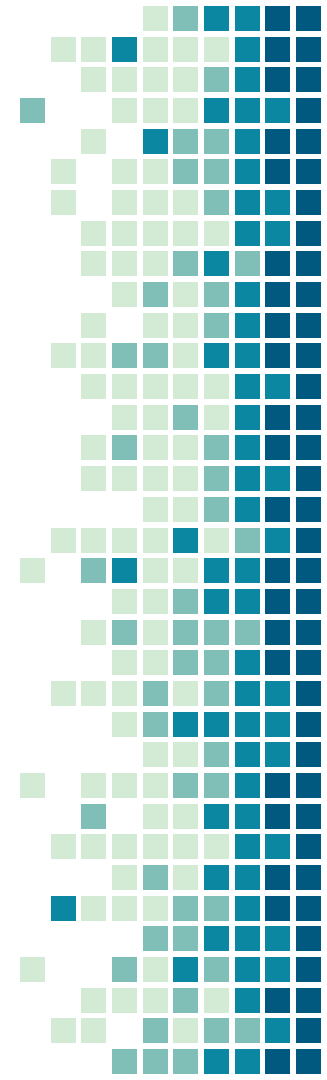
## SOLUCIÓN

```
# variables para almacenar los datos requeridos  
numero1 = 47  
numero2 = 68
```

```
# variable para almacenar el resultado de la suma  
suma = numero1 + numero2
```



Operador de asignación



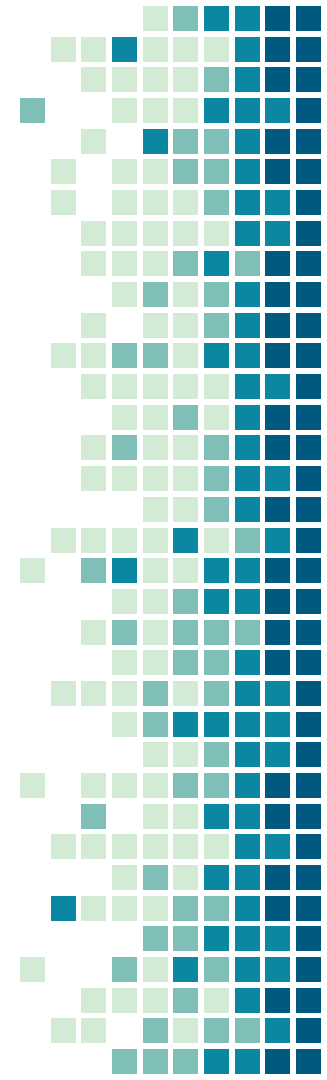
# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos  
numero1 = 47  
numero2 = 68
```

```
# variable para almacenar el resultado de la suma  
suma = numero1 + numero2
```

Expresión que dará como resultado el valor a asignar.



# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2
```





# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

¿qué ve el usuario?
```

## PROBLEMA

El programa realiza todas estas instrucciones, pero no muestra nada en pantalla.

# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('Hola mundo!')
```

### PROBLEMA

El programa realiza todas estas instrucciones, pero no muestra nada en pantalla.

### SOLUCIÓN

Utilizar la función *print* para mostrar algo en pantalla.

# EJERCICIO MODELO (TPII)

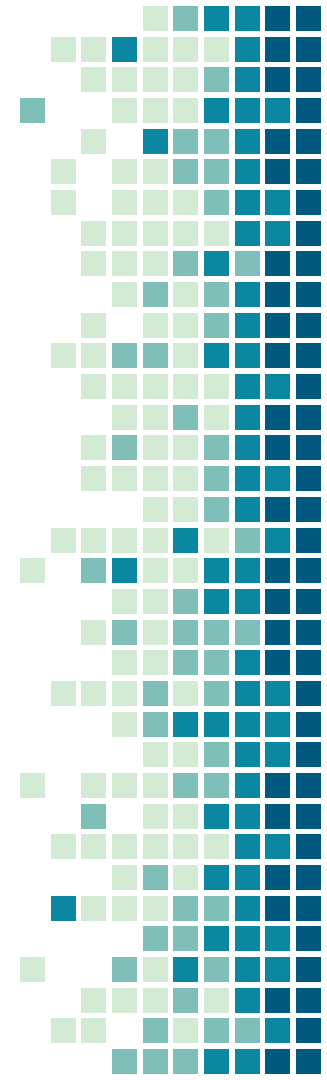
## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('Hola mundo!')
```

Función provista por el lenguaje para mostrar mensajes en pantalla.



# EJERCICIO MODELO (TPII)

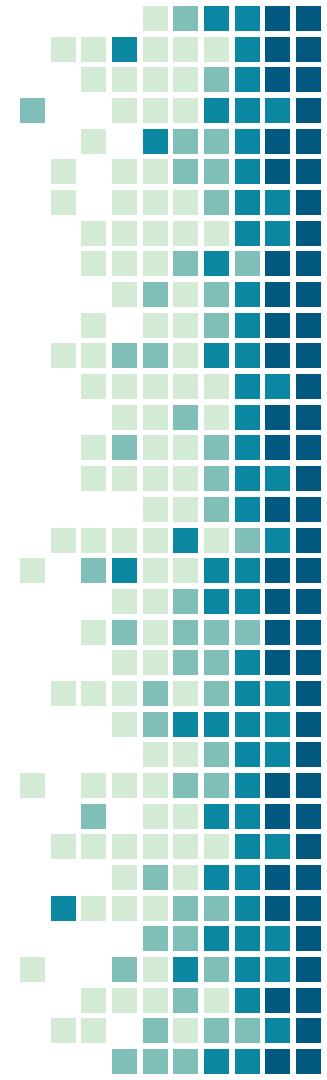
## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('Hola mundo!')
```

Los paréntesis son parte de la sintaxis, delimitan los parámetros que recibe la función *print*.



# EJERCICIO MODELO (TPII)

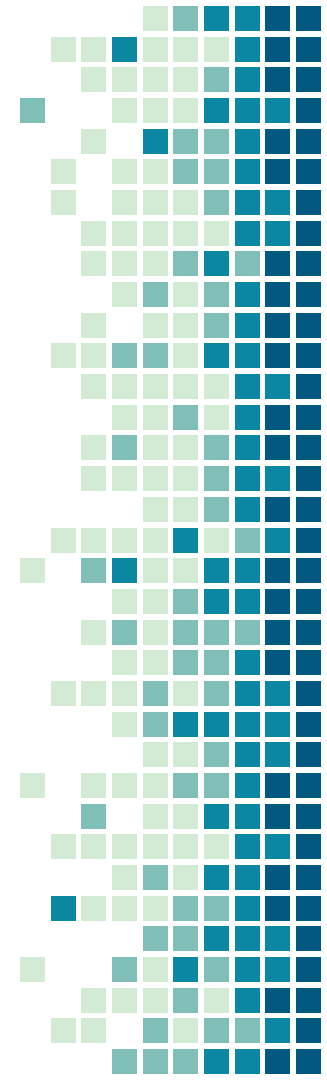
## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('Hola mundo!')
```

En este caso, el parámetro es el literal *Hola mundo!*.



# EJERCICIO MODELO (TPII)

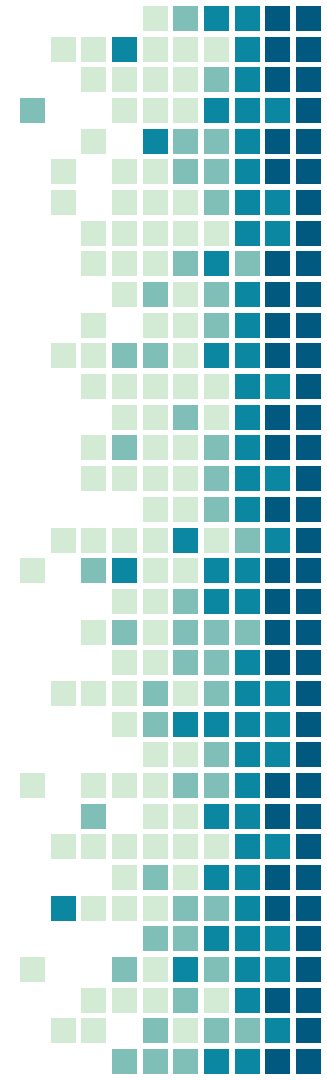
## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('Hola mundo!')
```

Los literales se escriben entre comillas (simples o dobles).



# EJERCICIO MODELO (TPII)

## SOLUCIÓN

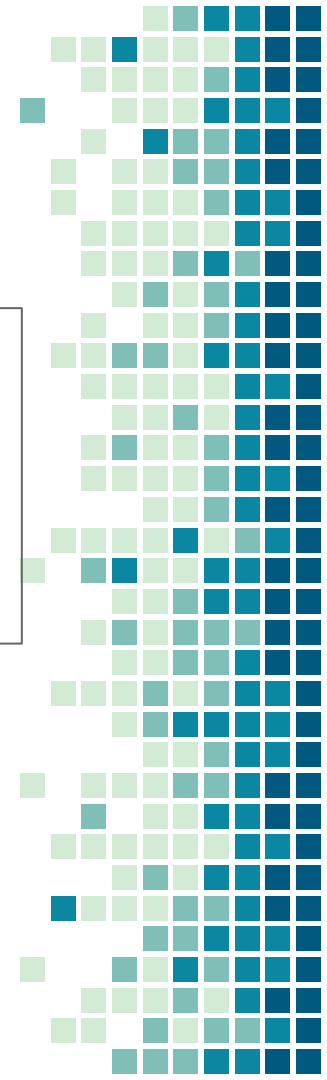
```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print(?)
```

### PROBLEMA

No queremos mostrar un mensaje literal, sino que necesitamos responder al pedido del enunciado.



# EJERCICIO MODELO (TPII)

## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('El resultado de sumar', numero1, '+', numero2, 'dio como resultado', suma)
```

### PROBLEMA

No queremos mostrar un mensaje literal, sino que necesitamos responder al pedido del enunciado.

### SOLUCIÓN

Formatear la salida.



# EJERCICIO MODELO (TPII)

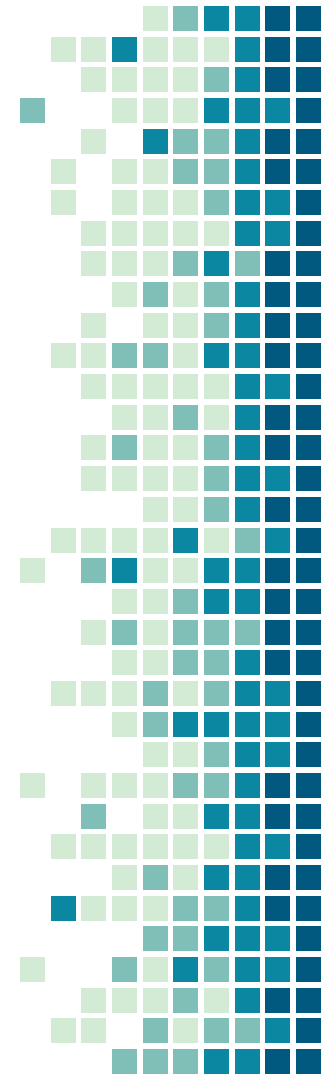
## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('El resultado de sumar', numero1, '+', numero2, 'dio como resultado', suma)
```

El parámetro de la función *print* ahora es una expresión compuesta.



# EJERCICIO MODELO (TPII)

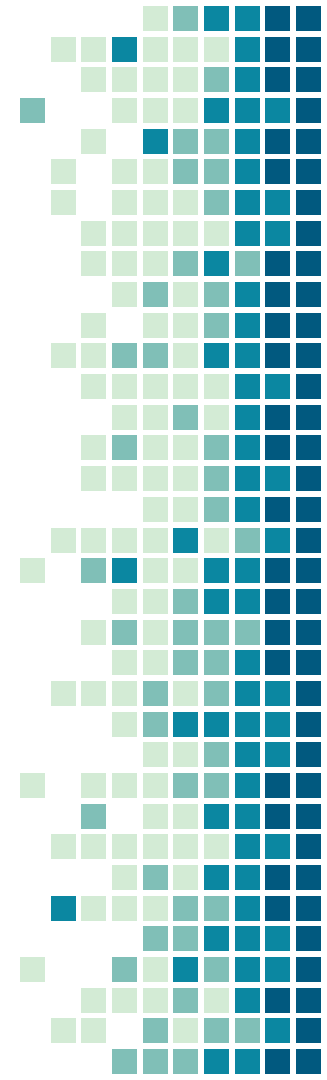
## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('El resultado de sumar', numero1, '+', numero2, 'dio como resultado', suma)
```

La coma (,) es la operación de concatenación.



# EJERCICIO MODELO (TPII)

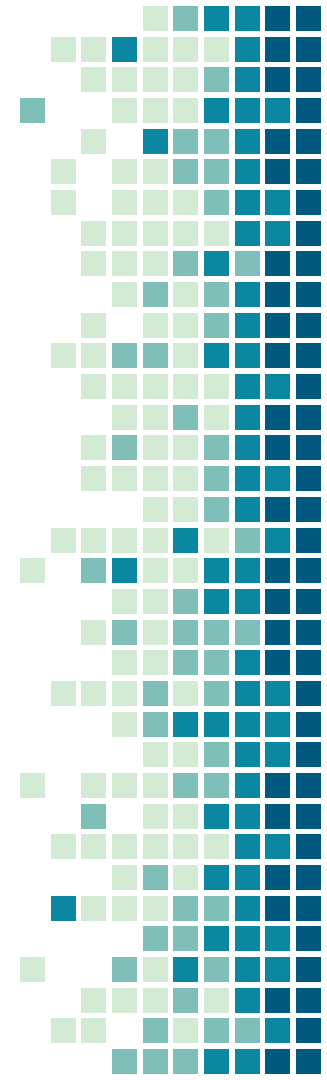
## SOLUCIÓN

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('El resultado de sumar', numero1, '+', numero2, 'dio como resultado', suma)
```

Notar que es posible concatenar variables, además de literales.



# EJERCICIO MODELO (TPII)

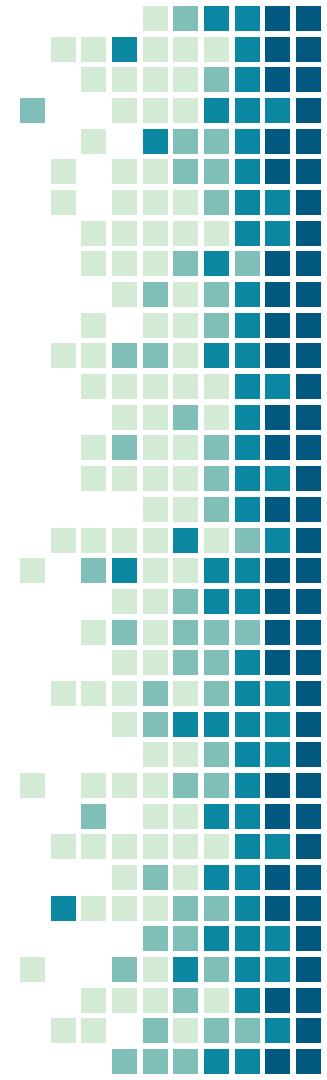
## SOLUCIÓN FINAL

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('El resultado de sumar', numero1, '+', numero2, 'dio como resultado', suma)
```

¿QUÉ VERÁ UN USUARIO AL EJECUTAR EL PROGRAMA?



# EJERCICIO MODELO (TPII)

## SOLUCIÓN FINAL

```
# variables para almacenar los datos requeridos
numero1 = 47
numero2 = 68

# variable para almacenar el resultado de la suma
suma = numero1 + numero2

# mensaje en pantalla
print('El resultado de sumar', numero1, '+', numero2, 'dio como resultado', suma)
```

## ¿QUÉ VERÁ UN USUARIO AL EJECUTAR EL PROGRAMA?

```
PS C:\Users\juani> & C:/Users/juani/AppData/Local/Programs/Python/Python38-32/python.exe "c:/Users/juani/Desktop/EJ_MODELO_EP I.py"
El resultado de sumar 47 + 68 dio como resultado 115
PS C:\Users\juani> █
```

