

# mcpp\_taller\_2\_Juan\_Munoz

August 15, 2019

## 1 Taller 2

Métodos Computacionales para Políticas Públicas - UROSARIO

**Entrega: viernes 16-ago-2019 11:59 PM**

**Juan Sebastián Muñoz Vargas** jsebastianmvargas@gmail.com

### 1.1 Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp\_taller2\_santiago\_matallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
  1. Descárguelo en PDF.
  2. Suba los dos archivos (.pdf y .ipynb) a su repositorio en GitHub antes de la fecha y hora límites.

(El valor de cada ejercicio está en corchetes [ ] después del número de ejercicio.)

---

### 1.2 1. [1]

[Pensar como un computador] Considere el siguiente código: `if x > 2: if y > 2: z = x + y print("z es", z) else: print("x es", x)` ¿Cuál es el resultado si

- a)  $x = 2, y = 5$ ?
- b)  $x = 3, y = 1$ ?
- c)  $x = 1, y = 1$ ?

d)  $x = 4, y = 3$ ?

a).  $x$  es 2

Dado que  $X$  no es mayor que 2, por lo que el primer condicional. Siendo así, se da el else.

b). No arroja nada, dado que entra al primer if, pero no cumple el segundo y no hay un else en este caso.

c).  $x$  es 1

Dado que  $X$  no es mayor que 2, por lo que el primer condicional. Siendo así, se da el else.

d).  $z$  es 7 En este caso  $x$  es mayor que dos, por lo que cumple el primero condicional.  $Y$  es mayor que 2, también, por lo que se ejecuta la proposición del segundo condicional.

---

### 1.3 2. [1]

[Pensar como un computador] ¿Cuál es el resultado del siguiente código y cuántas veces se recorre el loop?  $i = 0$  while  $i < 10$ :  $i = i + 1$  if  $i$  print( $i$ ) 2 4 6 8 10 El loop se corre 11 veces (desde el 0 hasta el 10) pero solo imprime los resultados para los 5 números pares.

---

### 1.4 3. [1]

[Pensar como un computador] ¿Cuál es el resultado del siguiente código y cuántas veces se recorre el loop?  $i = 0$  while  $i > 10$ :  $i = i + 1$  if  $i$  print( $i$ ) Solo se ejecuta satisfactoriamente la primera línea, dado que  $i$  es menor a diez (porque es 0) y, por lo tanto, nunca se empieza a ejecutar el 'while' y así, nunca arroja nada.

---

### 1.5 4. [2]

Escriba un programa que pida al usuario ingresar un número entero, y que imprima "par" si el número es par e "impar" si el número es impar. Agregue a su programa un código que genere una advertencia en caso de que el usuario ingrese algo diferente a un número entero: "Error. El usuario debe ingresar un número entero." (Investigue por su cuenta cómo lograr dicha validación y la generación del mensaje.)

```
[3]: def IngreseN (numero):  
    while True:  
        try:  
            userInput = int(input(numero))  
        except ValueError:  
            print("Error. El usuario debe ingresar un número entero.")  
            continue  
        else:  
            return userInput  
            break
```

*#Código del programa:*

```

num = IngreseN ("Bienvenido. Por Favor ingresa un número y te diré si es par o
↳impar.")

if ((num % 2) == 0):
    print(num, "es par.")
else:
    print(num, "es impar.")

```

Bienvenido. Por Favor ingresa un número y te diré si es par o impar.65411186214{  
 Error. El usuario debe ingresar un número entero.  
 Bienvenido. Por Favor ingresa un número y te diré si es par o impar.46156  
 46156 es par.

## 1.6 5. [2]

Escriba un for loop que imprima todos los múltiplos de 3 desde 40 hasta 0 en orden decreciente. Esto es, 39, 36, 33,..., 3, 0.

```

[4]: for n in range(39,-1, -3):
      print (n)

```

39  
 36  
 33  
 30  
 27  
 24  
 21  
 18  
 15  
 12  
 9  
 6  
 3  
 0

## 1.7 6. [2]

Escriba un loop que imprima todos los números entre 6 y 30 que no son divisibles por 2, 3 o 5.

```

[21]: for n in range(6, 30):
      if (n%2 == 0):
          pass
      elif (n%3 == 0):
          pass

```

```
elif (n%5 == 0):  
    pass  
else:  
    print (n)
```

7  
11  
13  
17  
19  
23  
29

---

## 1.8 7. [4]

Escriba un programa llamado “Adivine mi número”. El computador generará aleatoriamente un entero entre 1 y 100. El usuario digita un número y el computador responde “Menor” si el número aleatorio es menor que el escogido por el usuario, “Mayor” si el número aleatorio es mayor, y “¡Correcto!” si el usuario adivina el número. El jugador puede continuar ingresando números hasta que adivine correctamente.

**Ejemplo:** - El número aleatorio es 79. - El computador muestra el texto “Adivine el número entre 1 y 100:” y espera a que el usuario lo digite. - El usuario digita el número que está abajo en *itálicas*. - El computador devuelve uno de tres textos, según el caso: “Mayor”, “Menor”, o “¡Correcto!”.

Adivine el número entre 1 y 100: *40* Mayor  
Adivine el número entre 1 y 100: *70* Mayor  
Adivine el número entre 1 y 100: *80* Menor  
Adivine el número entre 1 y 100: *77* Mayor  
Adivine el número entre 1 y 100: *79* ¡Correcto!

### ¿Cómo generar números aleatorios en Python?

- Al comienzo de su programa escriba: `import random`
- Para generar un número aleatorio entre 1 y 100 escriba: `random.randint(1, 100)`

### Pistas:

- Piense en qué estructuras de control le sirven para resolver el problema.
- ¿Cómo determina si el número es mayor, menor o correcto?
- ¿Cómo le da turnos adicionales al usuario para adivinar, dependiendo de si en el turno anterior adivinó o no?

```
[2]: ***Adivine mi número**  
  
#Definición de la función para que solo acepte números enteros  
def IngresaN (numero):  
    while True:
```

```

try:
    userInput = int(input(numero))
except ValueError:
    print("Error. El usuario debe ingresar un número entero.")
    continue
else:
    return userInput
    break

#Bienvenida al jugador explicando el juego
print ("Bienvenido/a al juego 'Adivine mi número'")
print ("En este juego, el computador generará aleatoriamente un número entero_
→entre 1 y 100 y su misión es adivinar este número.")
print ("Para lograrlo, le solicitamos que escriba un número en la casilla y el_
→computador le dirá 'Menor' si el número aleatorio es menor que el escogido_
→por usted, 'Mayor' si el número aleatorio es mayor, y le avisara si adivina_
→el número.")

#Inicio del código
import random
al = random.randint(1, 100)
num = 101
while num != al:
    num = IngresaN ("Ingresa tu número:")
    if num < al:
        print ("Mayor")
    elif num > al:
        print ("Menor")
    else:
        print ("¡Correcto!")
        break

print ("¡Felicitaciones, has adivinado!")

```

Bienvenido/a al juego 'Adivine mi número'

En este juego, el computador generará aleatoriamente un número entero entre 1 y 100 y su misión es adivinar este número.

Para lograrlo, le solicitamos que escriba un número en la casilla y el computador le dirá 'Menor' si el número aleatorio es menor que el escogido por usted, 'Mayor' si el número aleatorio es mayor, y le avisara si adivina el número.

Ingresa tu número:84

Menor

Ingresa tu número:57

Menor

Ingresa tu número:22

Menor  
Ingresa tu número:10  
Mayor  
Ingresa tu número:20  
Menor  
Ingresa tu número:15  
Menor  
Ingresa tu número:13  
Menor  
Ingresa tu número:11  
Mayor  
Ingresa tu número:12  
¿Correcto!  
¿Felicitaciones, has adivinado!

---