



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencias de la Computación

Clase 04: Input y Control de Flujo

Rodrigo Toro Icarte (rntoro@uc.cl)

IIC1103 Introducción a la Programación - Sección 5

16 de Marzo, 2015

¿Qué aprendimos la clase pasada?

Tipos de datos básicos:

- 1 Números
 - int (3)
 - float (3.0)
 - complex (3 + 0j)
- 2 Texto
 - str (“Texto con comillas dobles” o ‘simples’)
- 3 Booleano
 - bool (True, False)

¿Qué aprendimos la clase pasada?

Ejemplos:

```
1 5 #ok
2 3.54 #ok
3 2+3j #ok
4 "Hola" #ok
5 True #ok
6 true #fail
```

¿Qué aprendimos la clase pasada?

Números: Operaciones

Operador	Descripción	Aridad	Precedencia
**	Exponente	Binario	1
+	Identidad	Unario	2
-	Negación	Unario	2
*	Multiplicación	Binario	3
/	División	Binario	3
//	División entera	Binario	3
%	Módulo	Binario	3
+	Suma	Binario	4
-	Resta	Binario	4

Obs: Para ahorrarse problemas, usen paréntesis.

¿Qué aprendimos la clase pasada?

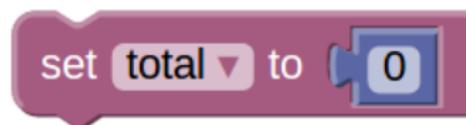
Ejemplos:

```
1 4+2          # Suma          4+2 => 6
2 4-2          # Resta          4-2 => 2
3 -7           # Negación       -7 => -7
4 3*4          # Multiplicación 3*4 => 12
5 2**3         # Exponente     2**3 => 8
6 3.5/2        # División       3.5/2 => 1.75
7 3.5//2       # División entera 3.5//2 => 1.0
8 7%2          # Módulo         7%2 => 1
```

Observación: recuerden **, // y %.

¿Qué aprendimos la clase pasada?

Variables



Sintaxis

nombre_variable = valor

```
1 a = 3-2+9      # Ahora "a" tiene valor 10
2 b = 4          # Ahora "b" tiene valor 4
3 pi = 3.1415    # Ahora "pi" tiene valor 3.1415
```

¿Qué aprendimos la clase pasada?

Ejemplo:

```
1 # Antes
2 res = 5
3 res = res*2
4 res = res**2
5 res = res%6
6 print(res)
```

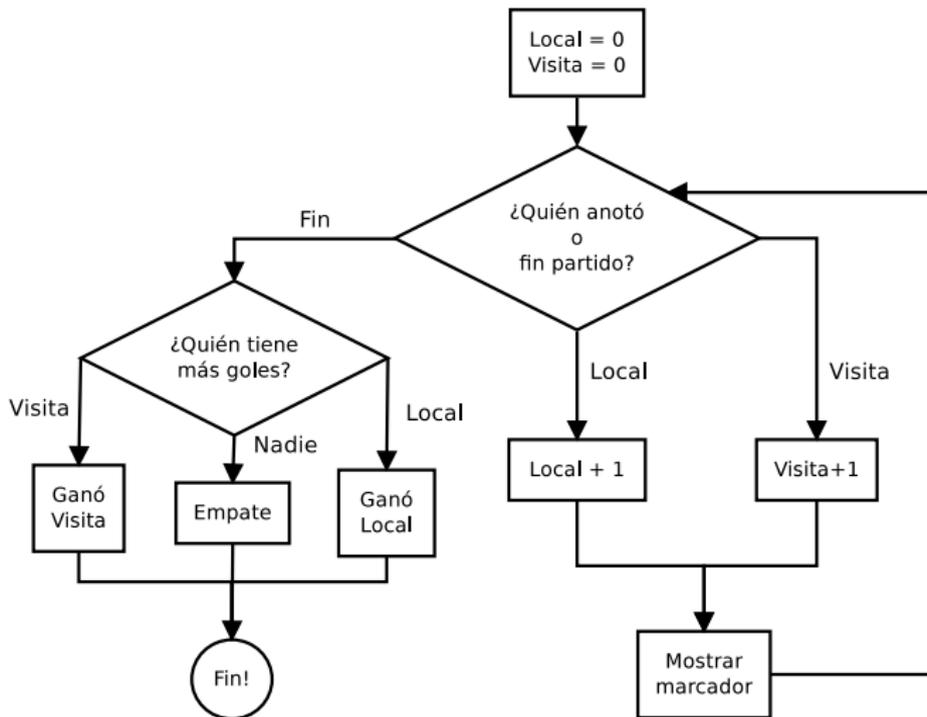
```
1 # Después
2 res = 5
3 res *= 2
4 res **= 2
5 res %= 6
6 print(res)
```

Objetivo: Marcador

“Queremos programar un marcador de fútbol. El usuario debe indicar quién anotó un gol (local o visita) o si terminó el partido. Con cada anotación debe actualizarse el marcador. Al finalizar el partido, se debe indicar qué equipo ganó el encuentro.”

Demo y diagrama.

Objetivo: Marcador



Objetivo: Marcador

Solución basada en código de **Ambar Zenteno**, disponible **Aquí**.

Objetivo: Marcador

Solución basada en código de **Ambar Zenteno**, disponible [Aquí](#).

Variables:

```
print " Bienvenidos al partido más esperado del mes!!! "
```

```
print " Hoy veremos jugar a Alemania v/s Brasil en la final de la copa del mundo!!! "
```

```
print " Esperamos que disfruten del partido!!! Que comience el juego!! "
```

```
set Brasil to 0
```

```
set Alemania to 0
```

```
set input to prompt for number with message " Ingrese quién metió el gol: 1: Alemania ; 2: Brasil ; 0: Fin del partido "
```

Objetivo: Marcador

Loop:

```
repeat while
  input ≠ 0
do
  if input = 1
  do
    change Alemania by 1
    print "Gooooool de Alemania!!!"
  if input = 2
  do
    change Brasil by 1
    print "Gooooool de Brasil!!!"
  print
  create text with
    "Alemania: "
    Alemania
    " "
    "Brasil: "
    Brasil
  set input to
    prompt for number with message "Ingrese quién metió el gol: Alemania: 1 ; Brasil: 2 ; Fin del partido:0"
```

The image shows a Scratch script designed to simulate a goal-scoring game. It uses a 'repeat while' loop that continues as long as the 'input' variable is not equal to 0. Inside the loop, there are two 'if' statements. The first 'if' statement checks if the input is 1; if true, it increases the 'Alemania' score by 1 and prints 'Gooooool de Alemania!!!'. The second 'if' statement checks if the input is 2; if true, it increases the 'Brasil' score by 1 and prints 'Gooooool de Brasil!!!'. After the 'if' statements, there is a 'print' block that concatenates the current scores of 'Alemania' and 'Brasil' into a single line of text. Finally, a 'set input to' block prompts the user to enter a number, with a message that says 'Ingrese quién metió el gol: Alemania: 1 ; Brasil: 2 ; Fin del partido:0'. The 'repeat while' loop will repeat these steps until the user enters 0.

Objetivo: Marcador

Ganador:

```
print " A terminado el partido, veamos quién es el ganador.... "
```

```
if (Alemania > Brasil)
```

```
do print " El ganador es Alemaniaaaaaa!!!! "
```

```
else if (Alemania < Brasil)
```

```
do print " El ganador es Brasi!!!!!! "
```

```
else print " Esto es un empateeee!!! "
```

```
print " Nadie esperaba este resultado!! Gracias por sintonizar el partido con nosotros y que tenga un buen día! "
```

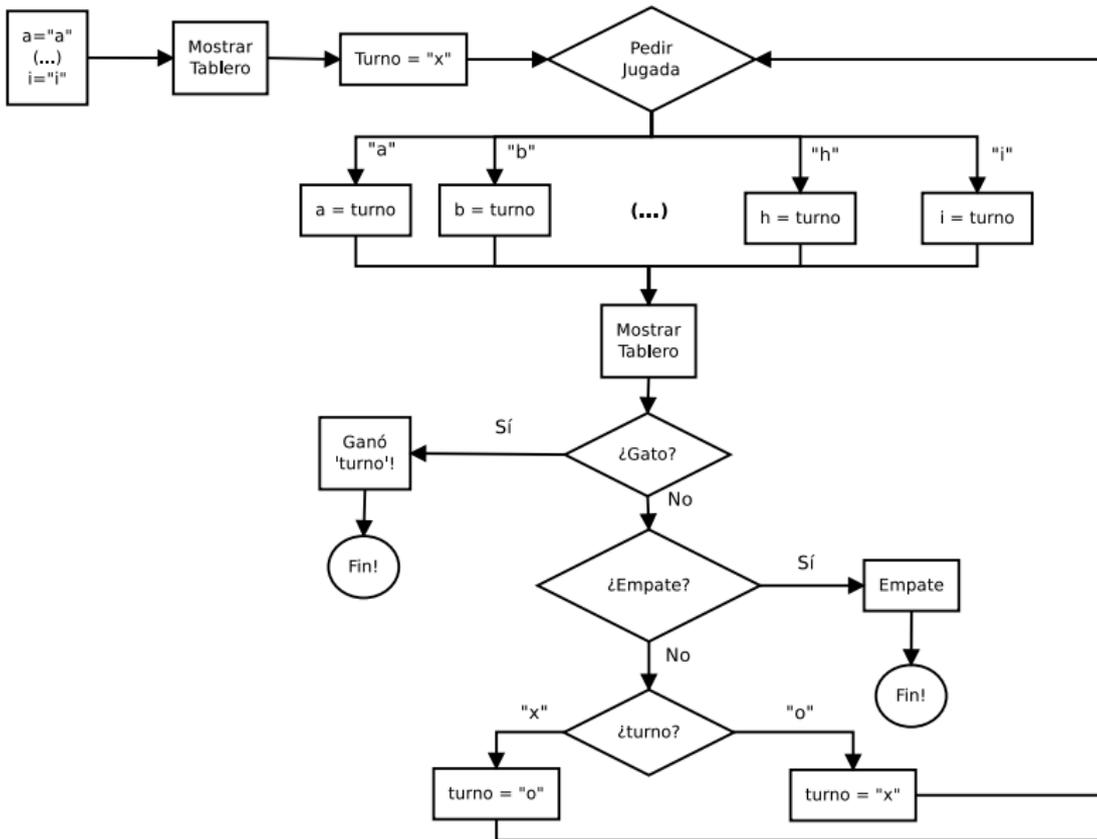
Objetivo: Gato

Ustedes: *“Programe el gato”*.

x		o		a	b	c
	x	o	→	d	e	f
		x		g	h	i

Demo.

Objetivo: Gato



Objetivo: Marcador

¿Qué sabemos hasta ahora?

Objetivo: Marcador

¿Qué sabemos hasta ahora?

Variables y operaciones simples sobre ellas.

```
1 local = 0           # goles del local
2 visita = 0         # goles de la visita
3
4 local += 1         # gol local
5 visita += 1       # gol visita
```

Objetivo: Marcador

¿Qué falta?

Objetivo: Marcador

¿Qué falta?

- 1 Mostrar el marcador.
- 2 Preguntar quién hizo el gol.
- 3 Aumentar marcador de quien hizo el gol.
- 4 Repetir hasta que finalice el partido.

Print

Print permite imprimir código en consola.

Sintaxis

```
print(variable_1, variable_2, variable_3, ...)
```

Print

Print permite imprimir código en consola.

Sintaxis

```
print(variable_1, variable_2, variable_3, ...)
```

```
1 local = 0           # goles del local
2 visita = 0         # goles de la visita
3 local += 1         # gol local
4 visita += 1        # gol visita
5
6 print("Local", local, "-", visita, "Visita")
```

Print

Print permite imprimir código en consola.

Sintaxis

```
print(variable_1, variable_2, variable_3, ...)
```

```
1 local = 0           # goles del local
2 visita = 0         # goles de la visita
3 local += 1        # gol local
4 visita += 1       # gol visita
5
6 print("Local", local, "-", visita, "Visita")
```

¿Qué muestra print?

Print

Print permite imprimir código en consola.

Sintaxis

```
print(variable_1, variable_2, variable_3, ...)
```

```
1 local = 0           # goles del local
2 visita = 0         # goles de la visita
3 local += 1         # gol local
4 visita += 1        # gol visita
5
6 print("Local", local, "-", visita, "Visita")
7 # >>> "Local 1 - 1 Visita"
```

Objetivo: Marcador

- 1 Mostrar el marcador. ✓
- 2 Preguntar quién hizo el gol.
- 3 Aumentar marcador de quien hizo el gol.
- 4 Repetir hasta que finalice el partido.

Input

Input retorna un str con un valor ingresado por el usuario.

Sintaxis

```
input(mensaje_para_usuario)
```

Input

Input retorna un str con un valor ingresado por el usuario.

Sintaxis

```
input(mensaje_para_usuario)
```

```
1 # pido input al usuario y lo guardo en a
2 a = input("Ingrese un str: ")
3 print(a)
```

Input

Input retorna un str con un valor ingresado por el usuario.

Sintaxis

```
input(mensaje_para_usuario)
```

```
1 # pido input al usuario y lo guardo en a
2 a = input("Ingrese un str: ")
3 print(a)
```

¿Cómo pido un int, float o bool al usuario?

Input

Input retorna un str con un valor ingresado por el usuario.

Sintaxis

```
input(mensaje_para_usuario)
```

```
1 # pido input al usuario y lo guardo en a
2 a = input("Ingrese un str: ")
3 print(a)
```

¿Cómo pido un int, float o bool al usuario?

no se puede... pero podemos *castear*.

Input

```
1 # pido input al usuario y lo guardo en a
2 a = input("Ingrese un str: ")
3 # pido un int al usuario
4 b = int(input("Ingrese un int: "))
5 # pido un float al usuario
6 c = float(input("Ingrese un float: "))
7 # pido un bool al usuario
8 d = bool(input("Ingrese un bool: "))
9 # imprimo valores obtenidos
10 print(a,b,c,d)
```

Input

Para el marcador...

Input

Para el marcador...

```
1 local = 0          # goles del local
2 visita = 0        # goles de la visita
3
4 # Guardo el equipo que anotó en variable "gol"
5 # [1: local, 2: visita, otro: fin partido]
6 gol = int(input("¿Qué equipo anotó el gol?"))
7
8 local += 1        # gol local
9 visita += 1       # gol visita
10
11 print("Local", local, "-", visita, "Visita")
12 # >>> "Local 1 - 1 Visita"
```

Objetivo: Marcador

- 1 Mostrar el marcador. ✓
- 2 Preguntar quién hizo el gol. ✓
- 3 Aumentar marcador de quien hizo el gol.
- 4 Repetir hasta que finalice el partido.

Control de flujo: if, elif y else

Condicionales: Permiten ejecutar (o no) trozos de código *si* se cumple una *condición*.

Control de flujo: if, elif y else

Condicionales: Permiten ejecutar (o no) trozos de código *si* se cumple una *condición*.

Condición: Variable de tipo bool.

Operatoria booleana

Operadores de comparación que retornan un booleano.

Operatoria booleana

Operadores de comparación que retornan un booleano.

Operador	Descripción
$a == b$	Retorna True ssi <i>a es igual a b</i>
$a != b$	Retorna True ssi <i>a es distinto de b</i>
$a < b$	Retorna True ssi <i>a es menor que b</i>
$a <= b$	Retorna True ssi <i>a es menor o igual que b</i>
$a > b$	Retorna True ssi <i>a es mayor que b</i>
$a >= b$	Retorna True ssi <i>a es mayor o igual que b</i>

Operatoria booleana

Ejemplos con números:

```
1 8 == 8           # >>> True
2 8 == 9           # >>> False
3 8 != 9           # >>> True
4 8 <= 9           # >>> True
5 8 > 8            # >>> False
```

Operatoria booleana

Ejemplos con texto:

```
1 a = "hola"
2 a == "hola"      # >>> True
3 a == "oli"       # >>> False
4 a != "oli"       # >>> True
5 a < "oli"        # >>> True
6 a > "a"          # >>> True
7 "A" < "a"        # >>> True
8 "ñ" > "z"        # >>> True
9 "á" > "z"        # >>> True
```

Operatoria booleana

Ejemplos con texto:

```
1 a = "hola"
2 a == "hola"      # >>> True
3 a == "oli"       # >>> False
4 a != "oli"       # >>> True
5 a < "oli"        # >>> True
6 a > "a"          # >>> True
7 "A" < "a"        # >>> True
8 "ñ" > "z"        # >>> True
9 "á" > "z"        # >>> True
```

Obs: las comparaciones en texto son según orden alfabético.

Obs 2: si hay 'ñ' o tildes no funciona.

Operatoria booleana

Operadores lógicos sobre booleanos.

Operatoria booleana

Operadores lógicos sobre booleanos.

Operador	Descripción
<i>not</i> a	Retorna True ssi a es False
a <i>or</i> b	Retorna True ssi a o b son True
a <i>and</i> b	Retorna True ssi a y b son True

Operatoria booleana

Ejemplos:

```
1 not True # >>> False
2 True and True # >>> True
3 True and False # >>> False
4 True or False # >>> True
5 False or True # >>> True
6 not (False or False) # >>> True
```

Operatoria booleana

Ejemplo práctico:

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 condicion = (0 <= a) and (a <= 9)
3 print(condicion)
```

Operatoria booleana

Ejemplo práctico:

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 condicion = (0 <= a) and (a <= 9)
3 print(condicion)
```

¿Por qué podemos mezclar **comparaciones** con **operadores booleanos**?

Control de flujo: if, elif y else

if: Ejecuta un trozo de código *si* se cumple una condición.

Sintaxis

```
if(condición):  
    bloque_de_código_if  
bloque_de_código_fuera_del_if
```

Control de flujo: if, elif y else

if: Ejecuta un trozo de código *si* se cumple una condición.

Sintaxis

```
if(condición):  
    bloque_de_código_if  
bloque_de_código_fuera_del_if
```

```
1 a = int(input("Ingrese número del 0 al 9: "))  
2 condicion = (0 <= a) and (a <= 9)  
3 if(not(condicion)):  
4     print("Error! número mal ingresado")  
5 print("Gracias!")
```

Control de flujo: if, elif y else

if: Ejecuta un trozo de código *si* se cumple una condición.

Sintaxis

```
if(condición):  
    bloque_de_código_if  
bloque_de_código_fuera_del_if
```

```
1 a = int(input("Ingrese número del 0 al 9: "))  
2 if(not((0 <= a) and (a <= 9))):  
3     print("Error! número mal ingresado")  
4 print("Gracias!")
```

Obs: También se puede poner la condición directo en el *if*.

Control de flujo: if, elif y else

else: Ejecuta un trozo de código *si* no se cumple la condición de un if anterior.

Sintaxis

```
if(condición):  
    bloque_de_código_if  
else:  
    bloque_de_código_else  
bloque_de_código_fuera_del_if_else
```

Control de flujo: if, elif y else

Ejemplo:

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 condicion = (0 <= a) and (a <= 9)
3 if(condicion):
4     print("Gracias!")
5 else:
6     print("Error! número mal ingresado")
```

Control de flujo: if, elif y else

elif: ejecuta un trozo de código *si* se cumple una condición y no se ha cumplido ningún if o elif anterior.

Control de flujo: if, elif y else

elif: ejecuta un trozo de código *si* se cumple una condición y no se ha cumplido ningún if o elif anterior.

Sintaxis

```
if(condición_if):
```

```
    bloque_de_código_if
```

```
elif(condición_elif):
```

```
    bloque_de_código_elif
```

```
else:
```

```
    bloque_de_código_else
```

```
bloque_de_código_fuera_del_if_else
```

Control de flujo: if, elif y else

elif: ejecuta un trozo de código *si* se cumple una condición y no se ha cumplido ningún if o elif anterior.

Sintaxis

```
if(condición_if):  
    bloque_de_código_if  
elif(condición_elif):  
    bloque_de_código_elif  
else:  
    bloque_de_código_else  
bloque_de_código_fuera_del_if_else
```

Obs: debe existir un if

Obs 2: pueden haber varios elif

Obs 3: podría no haber else

Control de flujo: if, elif y else

Ejemplo:

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 if(0 > a):
3     print("Error, número menor que cero")
4 elif(a > 9):
5     print("Error, número mayor que 9")
6 else:
7     print("Gracias!")
```

Control de flujo: if, elif y else

Ejemplo:

```
1 a = int(input("Ingrese número del 0 al 9: "))
2 if(0 > a):
3     print("Error, número menor que cero")
4 elif(a > 9):
5     print("Error, número mayor que 9")
6 else:
7     print("Gracias!")
```

Obs: cada if puede contener varias instrucciones.

Control de flujo: if, elif y else

¿Cuál es la diferencia entre estos dos códigos?

```
1 a = int(input("a: "))
2 if(0 < a):
3     print(1)
4 if(5 < a):
5     print(2)
6 if(10 < a):
7     print(3)
8 if(15 < a):
9     print(4)
10 else:
11     print(5)
```

```
1 a = int(input("a: "))
2 if(0 < a):
3     print(1)
4 elif(5 < a):
5     print(2)
6 elif(10 < a):
7     print(3)
8 elif(15 < a):
9     print(4)
10 else:
11     print(5)
```

Control de flujo: if, elif y else

Marcador de fútbol:

Control de flujo: if, elif y else

Marcador de fútbol:

```
1 local = 0          # goles del local
2 visita = 0        # goles de la visita
3 # [1: local, 2: visita, otro: fin partido]
4 gol = int(input("¿Qué equipo anotó el gol?"))
5
6 if(gol == 1):     # gol local
7     local += 1
8 elif(gol == 2):  # gol visita
9     visita += 1
10 else:
11     print("Fin del partido!")
12
13 print("Local", local, "-", visita, "Visita")
```

Objetivo: Marcador

- 1 Mostrar el marcador. ✓
- 2 Preguntar quién hizo el gol. ✓
- 3 Aumentar marcador de quien hizo el gol. ✓
- 4 Repetir hasta que finalice el partido.

Loops: while

while: Permite ejecutar varias veces la misma sección de código.

Loops: while

while: Permite ejecutar varias veces la misma sección de código.

Sintaxis

```
while(condición):  
    bloque_de_código_while  
bloque_de_código_fuera_del_while
```

```
1 i = 0  
2 while(i < 10):           # loop que se ejecuta 10 veces  
3     i += 1  
4     print(i)  
5 while(True):           # loop infinito  
6     i += 1  
7     print(i)
```

Loops: while (comandos útiles)

break: Sale automáticamente del loop.

continue: Retorna al inicio del loop.

Loops: while (comandos útiles)

break: Sale automáticamente del loop.

continue: Retorna al inicio del loop.

```
1 i = 0
2 while(True):      # loop infinito(?)
3     i += 1
4     if(i == 5):
5         continue  # dejo de ejecutar para caso i == 5
6     if(i == 11):
7         break     # salgo del loop
8     print(i)
9 print("Fin loop")
```

Loops: while

Marcador versión final (ojo con la sintaxis comprimida):

Loops: while

Marcador versión final (ojo con la sintaxis comprimida):

```
1 # Defino variables
2 local = 0; visita = 0; fin_juego = False
3 while(not fin_juego):
4     # pido input
5     gol = int(input("¿Qué equipo anotó el gol?"))
6     if gol == 1: local += 1      # Gol local
7     elif gol == 2: visita += 1  # Gol visita
8     else: fin_juego = True      # Fin partido
9
10    # Muestro marcador
11    print("Local", local, "-", visita, "Visita")
12
13    # Digo quién ganó
14    if local > visita: print("Ganó el local")
15    elif local < visita: print("Ganó la visita")
16    else: print("Empate")
```

Lo aprendido!

- 1 Mostrar el marcador (print). ✓
- 2 Preguntar quién hizo el gol (input). ✓
- 3 Aumentar marcador de quien hizo el gol (if-elif-else). ✓
- 4 Repetir hasta que finalice el partido (while). ✓

Ejercicios

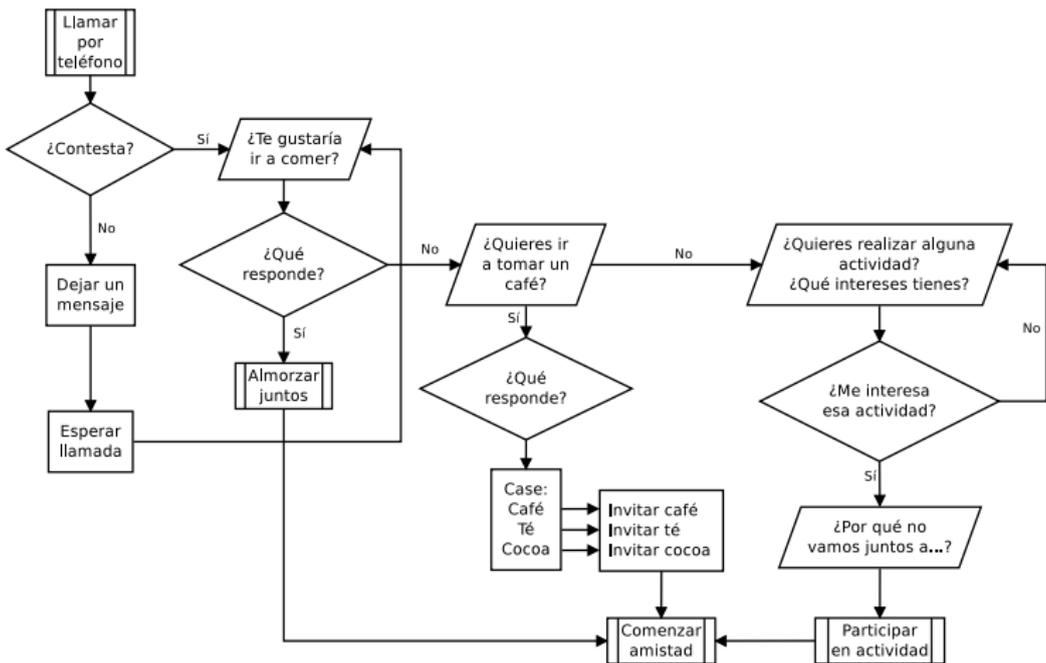
- 1) Muestre los números del 1 al n . (5 líneas de código)
- 2) Muestre los números pares entre 1 y n . (6 líneas de código)
- 3) Cree un programa que pida un número e imprima la suma de sus dígitos. (6 líneas de código)

Ejercicios

- 4) Cree un programa que pida un número y muestre True ssi el número es primo. (8 líneas de código)
- 5) Cree un programa que muestre las tablas de multiplicar del 1 al 10. (8 líneas de código)
- 6) Modifique código del marcador para que sea posible quitarle goles a un equipo. (agregar 4 líneas de código)

Ejercicios

7) Programe el algoritmo para hacer amigos del doctor Sheldon Cooper.



Ejercicios

- 8)** Programe un marcador de 1 game de tenis. Se debe preguntar al usuario quién ganó el punto, y el marcador actualizarse considerando 0, 15, 30, 40 y v (ventaja). Cuando un jugador gane el game, el programa debe finalizar felicitando al ganador.
- 9)** Programe el gato (los colores son opcional).

Ejercicio 1: Números

1) Muestre los números del 1 al n . (5 líneas de código)

Ejercicio 1: Números

1) Muestre los números del 1 al n . (5 líneas de código)

```
1 n = int(input("Ingrese número entero: "))
2 i = 1
3 while(i <= n):
4     print(i)
5     i+=1
```

Ejercicio 1: Números

1) Muestre los números del 1 al n . (5 líneas de código)

```
1 n = int(input("Ingrese número entero: "))
2 i = 1
3 while(i <= n):
4     print(i)
5     i+=1
```

Objetivo: Que logren ejecutar su primer programa en Python.

Ejercicio 2: Números pares

2) Muestre los números pares entre 1 y n . (6 líneas de código)

Ejercicio 2: Números pares

2) Muestre los números pares entre 1 y n . (6 líneas de código)

```
1 n = int(input("Ingrese número entero: "))
2 i = 1
3 while(i <= n):
4     if(i % 2 == 0):
5         print(i)
6     i+=1
```

Ejercicio 2: Números pares

2) Muestre los números pares entre 1 y n . (6 líneas de código)

```
1 n = int(input("Ingrese número entero: "))
2 i = 1
3 while(i <= n):
4     if(i % 2 == 0):
5         print(i)
6     i+=1
```

Objetivo: Aprender que el módulo 2 permite diferenciar entre números pares e impares.

Ejercicio 3: Sumar dígitos

3) Cree un programa que pida un número e imprima la suma de sus dígitos. (6 líneas de código)

Ejercicio 3: Sumar dígitos

3) Cree un programa que pida un número e imprima la suma de sus dígitos. (6 líneas de código)

Idea:

- $n\%10$ retorna la unidad de un número.
- $n//10$ retorna el n sin su unidad.
- Iteremos por cada dígito y los sumamos en una variable externa

Ejercicio 3: Sumar dígitos

Primer intento, tratemos de mostrar los dígitos del número:

Ejercicio 3: Sumar dígitos

Primer intento, tratemos de mostrar los dígitos del número:

```
1 num = int(input("Ingrese número entero: "))
2 while(num > 0):
3     print(num%10)
4     num//10
```

Ejercicio 3: Sumar dígitos

Primer intento, tratemos de mostrar los dígitos del número:

```
1 num = int(input("Ingrese número entero: "))
2 while(num > 0):
3     print(num%10)
4     num//10
```

Esto no funciona!

Ejercicio 3: Sumar dígitos

Segundo intento:

Ejercicio 3: Sumar dígitos

Segundo intento:

```
1 num = int(input("Ingrese número entero: "))
2 while(num > 0):
3     print(num%10)
4     num //= 10
```

Ejercicio 3: Sumar dígitos

Segundo intento:

```
1 num = int(input("Ingrese número entero: "))
2 while(num > 0):
3     print(num%10)
4     num //= 10
```

Funciona!

Ejercicio 3: Sumar dígitos

Ahora sumemos los dígitos en una variable externa.

Ejercicio 3: Sumar dígitos

Ahora sumemos los dígitos en una variable externa.

```
1 num = int(input("Ingrese número entero: "))
2 suma = 0
3 while(num > 0):
4     suma += num%10
5     num //= 10
6 print(suma)
```

Ejercicio 3: Sumar dígitos

Ahora sumemos los dígitos en una variable externa.

```
1 num = int(input("Ingrese número entero: "))
2 suma = 0
3 while(num > 0):
4     suma += num%10
5     num //= 10
6 print(suma)
```

Objetivo: Frustrarlos.

Ejercicio 3: Sumar dígitos

Ahora sumemos los dígitos en una variable externa.

```
1 num = int(input("Ingrese número entero: "))
2 suma = 0
3 while(num > 0):
4     suma += num%10
5     num //= 10
6 print(suma)
```

Objetivo: Frustrarlos.

- Aprender diferencia entre $n//10$ y $n=//10$.
- Aprender a iterar sobre dígitos de un `int`.

Ejercicio 4: Números primos

4) Cree un programa que pida un número y muestre True ssi el número es primo. (8 líneas de código)

Ejercicio 4: Números primos

4) Cree un programa que pida un número y muestre True ssi el número es primo. (8 líneas de código)

Idea:

- Pedimos número al usuario y lo guardamos en n .
- Vemos si algún número entre 2 y $n-1$ divide en forma exacta a n .
 - Si logré una división exacta entonces n es compuesto.
 - En otro caso es primo.

Ejercicio 4: Números primos

Primer intento:

```
1 n = int(input("n: "))
2 i = 2
3 while(i < n):
4     if(n%i == 0):
5         print(False)
6     i += 1
7 print(True)
```

Ejercicio 4: Números primos

Primer intento:

```
1 n = int(input("n: "))
2 i = 2
3 while(i < n):
4     if(n%i == 0):
5         print(False)
6     i += 1
7 print(True)
```

¿Funciona?

Ejercicio 4: Números primos

Segundo intento: Guardemos el resultado en una variable externa.

Ejercicio 4: Números primos

Segundo intento: Guardemos el resultado en una variable externa.

```
1 n = int(input("n: "))
2 i = 2
3 es_primo = True
4 while(i < n):
5     if(n%i == 0):
6         es_primo = False
7     i += 1
8 print(es_primo)
```

Ejercicio 4: Números primos

Segundo intento: Guardemos el resultado en una variable externa.

```
1 n = int(input("n: "))
2 i = 2
3 es_primo = True
4 while(i < n):
5     if(n%i == 0):
6         es_primo = False
7     i += 1
8 print(es_primo)
```

¿Qué pasa con el 1?

Ejercicio 4: Números primos

```
1 n = int(input("n: "))
2 i = 2
3 es_primo = True
4 while(i < n):
5     if(n%i == 0):
6         es_primo = False
7     i += 1
8 print(es_primo and n > 1)
```

Ejercicio 4: Números primos

```
1 n = int(input("n: "))
2 i = 2
3 es_primo = True
4 while(i < n):
5     if(n%i == 0):
6         es_primo = False
7     i += 1
8 print(es_primo and n > 1)
```

Objetivo: Aprender a resolver problemas mediante probar todas las combinaciones posibles.

Ejercicio 5: Tablas de multiplicar

5) Cree un programa que muestre las tablas de multiplicar del 1 al 10. (8 líneas de código)

Ejercicio 5: Tablas de multiplicar

5) Cree un programa que muestre las tablas de multiplicar del 1 al 10. (8 líneas de código)

Idea:

- Defino una while que modifique una variable i de 1 a 10.
- Dentro del while anterior defino otro que modifique una variable j de 1 a 10.
- Muestro $i*j$,

Ejercicio 5: Tablas de multiplicar

```
1 i = 1
2 while(i < 11):
3     print("-----")
4     j = 1
5     while(j < 11):
6         print(i, "x", j, "=", i*j)
7         j+=1
8     i+=1
```

Ejercicio 5: Tablas de multiplicar

```
1 i = 1
2 while(i < 11):
3     print("-----")
4     j = 1
5     while(j < 11):
6         print(i, "x", j, "=", i*j)
7         j+=1
8     i+=1
```

Objetivo: Que aprendan a trabajar con loops anidados.

Ejercicio midterm 2014-1

“Realice un programa que reciba un número e indique si es la hipotenusa de un trio-pitagórico”.

Obs: a es la hipotenusa de un trío pitagórico si existe b y c enteros tales que:

- $a > b > c > 0$
- $a^2 = b^2 + c^2$

Ejemplo: 5-4-3 y 13-12-5 son tríos pitagóricos.

Ejercicio 6: Marcador de fútbol

6) Modifique código del marcador para que sea posible quitarle goles a un equipo. (agregar 4 líneas de código)

Ejercicio 6: Marcador de fútbol

6) Modifique código del marcador para que sea posible quitarle goles a un equipo. (agregar 4 líneas de código)

Objetivos:

- Entender el código del marcador de fútbol.
- Ser capaz de modificar código realizado por alguien más.

Ejercicio 6: Marcador de fútbol

Antes:

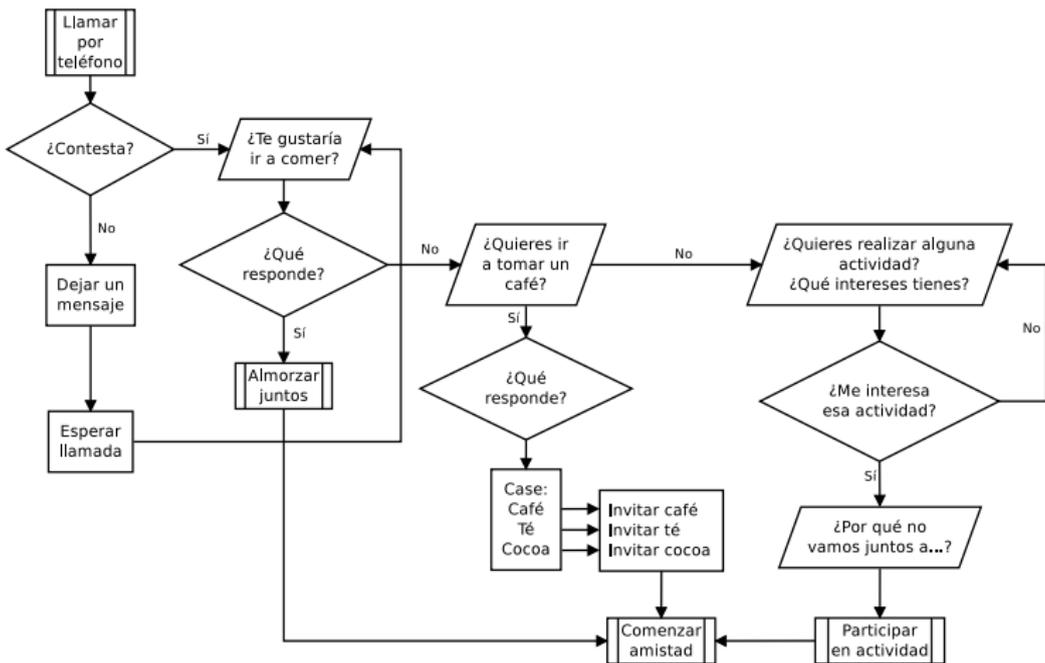
```
1 # Defino variables
2 local = 0; visita = 0; fin_juego = False
3 while(not fin_juego):
4     # pido input
5     gol = int(input("¿Qué equipo anotó el gol?"))
6     if gol == 1: local += 1          # Gol local
7     elif gol == 2: visita += 1     # Gol visita
8     else: fin_juego = True         # Fin partido
9
10    # Muestro marcador
11    print("Local", local, "-", visita, "Visita")
12
13 # Digo quién ganó
14 if local > visita: print("Ganó el local")
15 elif local < visita: print("Ganó la visita")
16 else: print("Empate")
```

Ejercicio 6: Marcador de fútbol

```
1 # Defino variables
2 local = 0; visita = 0; fin_juego = False
3 while(not fin_juego):
4     # pido input
5     gol = int(input("¿Qué equipo anotó el gol?"))
6     if gol == 1: local += 1          # Gol local
7     elif gol == -1: local -= 1     # Quito gol local
8     elif gol == 2: visita += 1    # Gol visita
9     elif gol == -2: visita -= 1   # Quito gol visita
10    else: fin_juego = True         # Fin partido
11
12    # Muestro marcador
13    print("Local", local, "-", visita, "Visita")
14
15    # Digo quién ganó
16    if local > visita: print("Ganó el local")
17    elif local < visita: print("Ganó la visita")
18    else: print("Empate")
```

Ejercicio 7: Hacedor de amigos :)

7) Programe el algoritmo para hacer amigos del doctor Sheldon Cooper.



Ejercicio 7: Hacedor de amigos :)

```
1 print("Llamar por teléfono")
2 b = input("¿Contesta? (s/n)")
3 if(b == "n"):
4     print("Dejar mensaje")
5     print("Esperar llamada...")
6     print("Llamada recibida!")
7
8 b = input("¿Te gustaría ir a comer? (s/n)")
9 if(b == "s"):
10    print("Ir a almorzar juntos")
11 else:
12    b = int(input("¿Quieres ir a tomar un café, té o
    cocoa? (1- café, 2- té, 3- cocoa, otro- no)"))
```

Ejercicio 7: Hacedor de amigos :)

```
13  if(b > 0 and b < 4):
14      if(b == 1):
15          print("Invitar café")
16      if(b == 2):
17          print("Invitar té")
18      if(b == 3):
19          print("Invitar cocoa")
20  else:
21      while(True):
22          p = input("¿Quieres realizar alguna actividad? ¿
Qué intereses tienes? ")
23          r = input("¿Sheldon está de acuerdo? (s/n)")
24          if(r=="s"):
25              print("ir a",p)
26              break
27          else:
28              print("Emm...")
29  print("Comenzar amistad")
```

Ejercicio 7: Hacedor de amigos :)

Objetivos:

- Trabajar con entradas y salidas complejas desde la consola.
- Acostumbrarse a mezclar `if` con `while`.
- Trabajar con varios niveles de indentación.
- Lograr llegar desde distintos caminos al mismo final.

Ejercicio 8: Game de tenis

8) Programe un marcador de 1 game de tenis. Se debe preguntar al usuario quién ganó el punto, y el marcador actualizarse considerando 0, 15, 30, 40 y v (ventaja). Cuando un jugador gane el game, el programa debe finalizar felicitando al ganador.

Ejercicio 8: Game de tenis

Estrategia: Manejemos los puntos de cada jugador como 0, 1, 2, 3 y 4.

Ejercicio 8: Game de tenis

Estrategia: Manejemos los puntos de cada jugador como 0, 1, 2, 3 y 4.

```
1 # defino variables con los puntos (0,1,2,3 o 4)
2 puntos_1 = 0; puntos_2 = 0
3 while(True):
4     # Pregunto quién hizo el punto
5     p = int(input("¿Qué hizo el punto? (1 o 2)"))
6     if p == 1: # Agrego punto a 1
7         puntos_1 += 1
8     if p == 2: # Agrego punto a 2
9         puntos_2 += 1
```

Ejercicio 8: Game de tenis

Estrategia: Manejemos los puntos de cada jugador como 0, 1, 2, 3 y 4.

```
1 # defino variables con los puntos (0,1,2,3 o 4)
2 puntos_1 = 0; puntos_2 = 0
3 while(True):
4     # Pregunto quién hizo el punto
5     p = int(input("¿Qué hizo el punto? (1 o 2)"))
6     if p == 1: # Agrego punto a 1
7         puntos_1 += 1
8     if p == 2: # Agrego punto a 2
9         puntos_2 += 1
```

¿Cómo muestro el marcador?

Ejercicio 8: Game de tenis

Para el jugador 1:

```
11 # Muestro puntos Jugador 1
12 p1_string = "0"
13 if puntos_1 == 1: p1_string = "15"
14 elif puntos_1 == 2: p1_string = "30"
15 elif puntos_1 == 3: p1_string = "40"
16 elif puntos_1 == 4: p1_string = "v"
17 print("Jugador 1:",p1_string)
```

Ejercicio 8: Game de tenis

Para el jugador 1:

```
11 # Muestro puntos Jugador 1
12 p1_string = "0"
13 if puntos_1 == 1: p1_string = "15"
14 elif puntos_1 == 2: p1_string = "30"
15 elif puntos_1 == 3: p1_string = "40"
16 elif puntos_1 == 4: p1_string = "v"
17 print("Jugador 1:",p1_string)
```

¿Cómo muestro el jugador 2?

Ejercicio 8: Game de tenis

```
11 # Muestro puntos Jugador 1
12 p1_string = "0"
13 if puntos_1 == 1: p1_string = "15"
14 elif puntos_1 == 2: p1_string = "30"
15 elif puntos_1 == 3: p1_string = "40"
16 elif puntos_1 == 4: p1_string = "v"
17 print("Jugador 1:",p1_string)
18 # Muestro puntos Jugador 2
19 p2_string = "0"
20 if puntos_2 == 1: p2_string = "15"
21 elif puntos_2 == 2: p2_string = "30"
22 elif puntos_2 == 3: p2_string = "40"
23 elif puntos_2 == 4: p2_string = "v"
24 print("Jugador 2:",p2_string)
```

Ejercicio 8: Game de tenis

```
11 # Muestro puntos Jugador 1
12 p1_string = "0"
13 if puntos_1 == 1: p1_string = "15"
14 elif puntos_1 == 2: p1_string = "30"
15 elif puntos_1 == 3: p1_string = "40"
16 elif puntos_1 == 4: p1_string = "v"
17 print("Jugador 1:",p1_string)
18 # Muestro puntos Jugador 2
19 p2_string = "0"
20 if puntos_2 == 1: p2_string = "15"
21 elif puntos_2 == 2: p2_string = "30"
22 elif puntos_2 == 3: p2_string = "40"
23 elif puntos_2 == 4: p2_string = "v"
24 print("Jugador 2:",p2_string)
```

Spoiler: Nunca es bueno copiar y pegar código... para ello sirven las *funciones*.

Ejercicio 8: Game de tenis

Gana quien supere los 4 '*puntos*' y tenga diferencia de 2.
Modifiquemos el `if` para el player 1.

Antes:

```
6  if p == 1: # Agrego punto a 1
7      puntos_1 += 1
```

Ejercicio 8: Game de tenis

Gana quien supere los 4 '*puntos*' y tenga diferencia de 2.
Modifiquemos el if para el player 1.

Antes:

```
6  if p == 1: # Agrego punto a 1
7      puntos_1 += 1
```

Después:

```
6  if p == 1: # Agrego punto a 1
7      puntos_1 += 1
8      # Cuando supero el 40 y hay diferencia de 2
9      if(puntos_1 > 3 and (puntos_1 - puntos_2) > 1):
10         print("Game jugador 1")
11         break
```

Ejercicio 8: Game de tenis

Gana quien supere los 4 *'puntos'* y tenga diferencia de 2.
Modifiquemos el `if` para el player 1.

Antes:

```
6   if p == 1: # Agrego punto a 1
7       puntos_1 += 1
```

Después:

```
6   if p == 1: # Agrego punto a 1
7       puntos_1 += 1
8       # Cuando supero el 40 y hay diferencia de 2
9       if(puntos_1 > 3 and (puntos_1 - puntos_2) > 1):
10          print("Game jugador 1")
11          break
```

¿Cómo agrego esto para el player 2?

Ejercicio 8: Game de tenis

```
1 # defino variables con los puntos (0,1,2,3 o 4)
2 puntos_1 = 0; puntos_2 = 0
3 while(True):
4     # Pregunto quién hizo el punto
5     p = int(input("¿Qué hizo el punto? (1 o 2)"))
6     if p == 1: # Agrego punto a 1
7         puntos_1 += 1
8         # Cuando supero el 40 y hay diferencia de 2
9         if(puntos_1 > 3 and (puntos_1 - puntos_2) > 1):
10             print("Game jugador 1")
11             break
12     if p == 2: # Agrego punto a 2
13         puntos_2 += 1
14         # Cuando supero el 40 y hay diferencia de 2
15         if(puntos_2 > 3 and (puntos_2 - puntos_1) > 1):
16             print("Game jugador 2")
17             break
```

Ejercicio 8: Game de tenis

```
1 # defino variables con los puntos (0,1,2,3 o 4)
2 puntos_1 = 0; puntos_2 = 0
3 while(True):
4     # Pregunto quién hizo el punto
5     p = int(input("¿Qué hizo el punto? (1 o 2)"))
6     if p == 1: # Agrego punto a 1
7         puntos_1 += 1
8         # Cuando supero el 40 y hay diferencia de 2
9         if(puntos_1 > 3 and (puntos_1 - puntos_2) > 1):
10             print("Game jugador 1")
11             break
12     if p == 2: # Agrego punto a 2
13         puntos_2 += 1
14         # Cuando supero el 40 y hay diferencia de 2
15         if(puntos_2 > 3 and (puntos_2 - puntos_1) > 1):
16             print("Game jugador 2")
17             break
```

... solo falta manejar el paso de v-40 a 40-40

Ejercicio 8: Game de tenis

```
1 puntos_1 = 0; puntos_2 = 0
2 while(True):
3     p = int(input("¿Qué hizo el punto? (1 o 2)"))
4     if p == 1: # Agrego punto a 1
5         puntos_1 += 1
6         if(puntos_1 > 3 and (puntos_1 - puntos_2) > 1):
7             print("Game jugador 1")
8             break
9     if p == 2: # Agrego punto a 2
10        puntos_2 += 1
11        if(puntos_2 > 3 and (puntos_2 - puntos_1) > 1):
12            print("Game jugador 2")
13            break
14
15 # Si ambos tienen 4 vuelven a 40
16 if(puntos_1 == 4 and puntos_2 == 4):
17     puntos_1 = 3; puntos_2 = 3
18
19 # (...)
```

Ejercicio 8: Game de tenis

Objetivos:

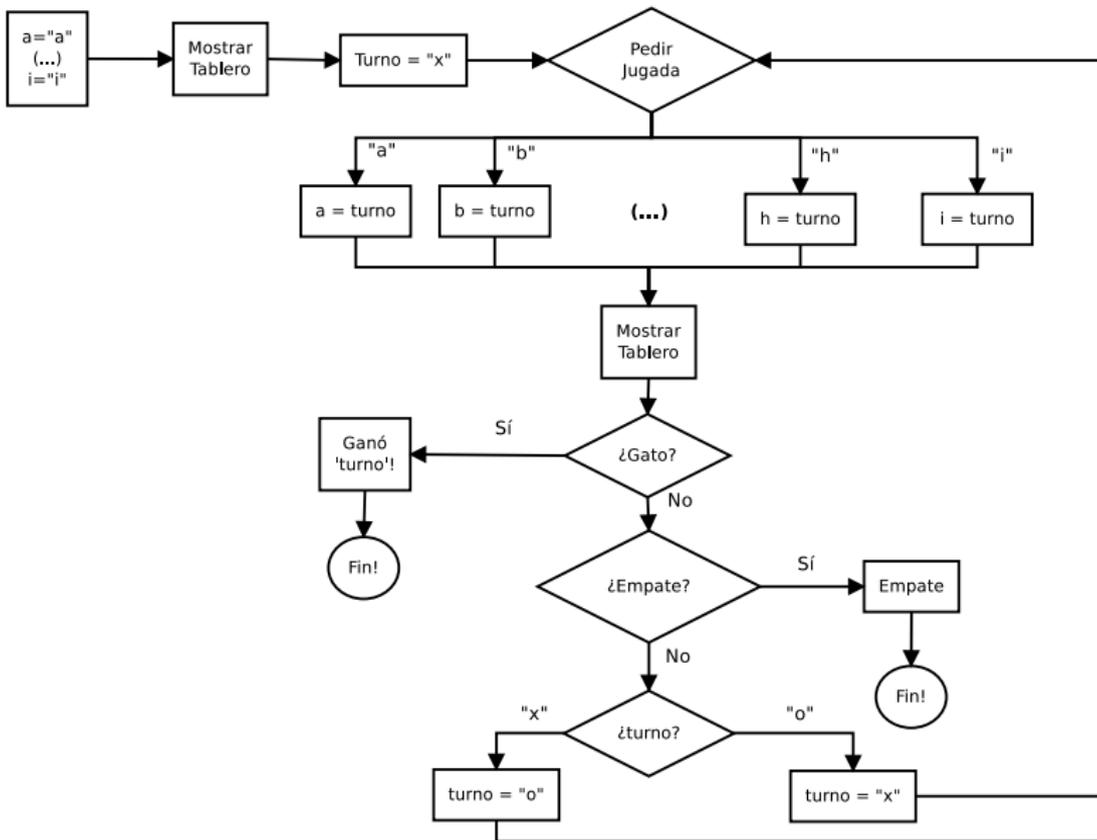
- Enfrentarse a un problema *real* con múltiples formas de ser programado.
- Ser capaz de evaluar cada camino y elegir *el mejor*.
- Identificar las limitaciones de las herramientas vistas hasta el momento.

Ejercicio 9: El gato

9) Programe el gato (los colores son opcional).

x		o		a	b	c
	x	o	→	d	e	f
		x		g	h	i

Ejercicio 9: El gato



Ejercicio 9: El gato

Primero definamos las variables.

Ejercicio 9: El gato

Primero definamos las variables.

```
1 # Defino variables del tablero
2 a = "a"; b = "b"; c = "c";
3 d = "d"; e = "e"; f = "f";
4 g = "g"; h = "h"; i = "i"
5 turno = "x"
```

Ejercicio 9: El gato

Reaccionemos ante la jugada.

Ejercicio 9: El gato

Reaccionemos ante la jugada.

```
17 # Reacciono ante la movida,  
18 # chequeo que casilla no esté ocupada  
19 # y le asigno el valor del turno (x/o)  
20 if(pos == "a" and a == "a"): a = turno  
21 if(pos == "b" and b == "b"): b = turno  
22 if(pos == "c" and c == "c"): c = turno  
23 if(pos == "d" and d == "d"): d = turno  
24 if(pos == "e" and e == "e"): e = turno  
25 if(pos == "f" and f == "f"): f = turno  
26 if(pos == "g" and g == "g"): g = turno  
27 if(pos == "h" and h == "h"): h = turno  
28 if(pos == "i" and i == "i"): i = turno
```

Ejercicio 9: El gato

Verifiquemos si, luego de su jugada, **turno** ganó.

Ejercicio 9: El gato

Verifiquemos si, luego de su jugada, turno ganó.

```
30 # Veo si turno ganó
31 l_vertical = ((a == b and b == c) or
32              (d == e and e == f) or
33              (g == h and h == i))
34 l_horizontal = ((a == d and d == g) or
35                (b == e and e == h) or
36                (c == f and f == i))
37 diagonales = ((a == e and e == i) or
38               (c == e and e == g))
39 if(l_vertical or l_horizontal or diagonales):
40     print("Gato! ganó", turno)
41     break
```

Ejercicio 9: El gato

Si no ganó, podría haber un empate...

Ejercicio 9: El gato

Si no ganó, podría haber un empate...

```
43 # Si no quedan casillas disponibles es un empate!  
44 ocup_f1 = (a != "a") and (b != "b") and (c != "c")  
45 ocup_f2 = (d != "d") and (e != "e") and (f != "f")  
46 ocup_f3 = (g != "g") and (g != "g") and (i != "i")  
47 if(ocup_f1 and ocup_f2 and ocup_f3):  
48     print("Empate!")  
49     break
```

Ejercicio 9: El gato

Si el juego continua, realizamos el cambio de turno.

Ejercicio 9: El gato

Si el juego continua, realizamos el cambio de turno.

```
51 # Cambio de turno
52 if(turno == "x"): turno = "o"
53 else: turno = "x"
```

Ejercicio 9: El gato

Si el juego continua, realizamos el cambio de turno.

```
51 # Cambio de turno
52 if(turno == "x"): turno = "o"
53 else: turno = "x"
```

Objetivo gato: Programar algo entretenido :)

Gato: Colores consola

Colores en consola python: **[Link](#)**

Más tipos de colores: **[Link](#)**

Tarea 1: SUDOKU

	0	1	2	3	4	5	6	7	8
0	5	3			7				
1	6			1	9	5			
2		9	8					6	
3	8				6				3
4	4			8		3			1
5	7				2				6
6		6					2	8	
7				4	1	9			5
8					8			7	9

Programar un jugador de sudoku en consola.

Tarea 1: SUDOKU

Tu programa debe permitir:

- Cargar un tablero (fácil, medio o difícil).
- Mostrar el tablero actual.
- Dar al usuario tres opciones:
 - Ingresar una jugada.
 - Salir.

Tarea 1: SUDOKU

Les daremos la librería `sudoku.py`.

```
1 import sudoku # permite cargar la librería
2
3 # Carga un nuevo tablero de dificultad 'n'
4 # n -> 1-fácil 2-medio 3-difícil
5 sudoku.juego.cambiarTablero(n)
6
7 # Retorna el valor de la casilla en la posición (i,j)
8 # i -> fila (de 0 a 8)
9 # j -> columna (de 0 a 8)
10 sudoku.juego.obtener(i,j)
11
12 # Cambia el valor de la casilla (i,j) por v
13 # i -> fila (de 0 a 8)
14 # j -> columna (de 0 a 8)
15 # v -> nuevo valor de la casilla (de 1 a 9)
16 sudoku.juego.definir(i,j,v)
```

Tarea 1: SUDOKU

Ver ejemplo.

Tarea 1: SUDOKU

Ver ejemplo.

Metas de la semana:

- Tener los 9 ejercicios propuestos resueltos por ti.
- Tarea 1:
 - Cargar un tablero de alguna dificultad.
 - Mostrar el tablero en consola.

Tarea 1: SUDOKU

Ver ejemplo.

Metas de la semana:

- Tener los 9 ejercicios propuestos resueltos por ti.
- Tarea 1:
 - Cargar un tablero de alguna dificultad.
 - Mostrar el tablero en consola.

Hint:

- Para mostrar el tablero necesitarán: `print(..., end="")`.