INTRODUCCIÓN A LA PROGRAMACIÓN

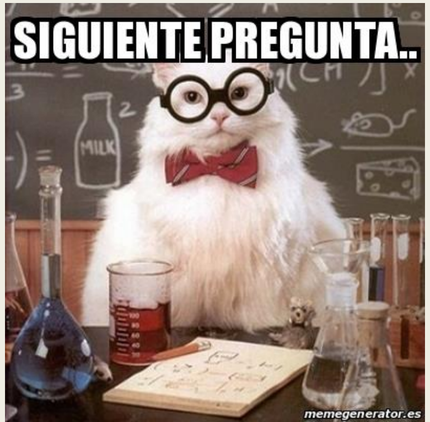
FUNCIONES RECURSIVAS

Rodrigo López A.
rilopez3@uc.cl

Funciones recursivas

- Hasta el momento hemos aprendido a definir nuestras propias funciones, dentro de estas podemos implementar algoritmos para su resolución.
- Las funciones que hemos utilizado han tenido un diseño lineal, secuencial, existe una secuencia clara de sus pasos desde inicio a fin.
- Podemos incluir ciclos de repetición en nuestras funciones, pero su ejecución continua siendo secuencial y contenida.

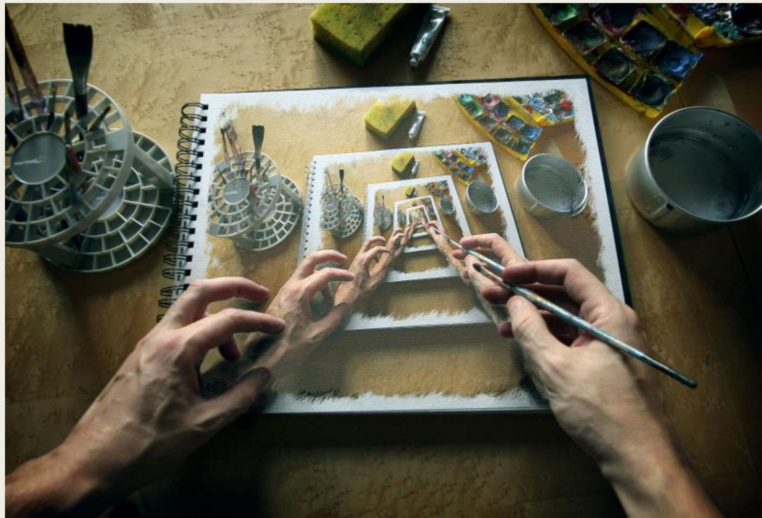
- También pudimos probar que es posible utilizar otras funciones dentro de nuestra función.
- PERO, ¿Qué pasa si una función se llamara a si misma?



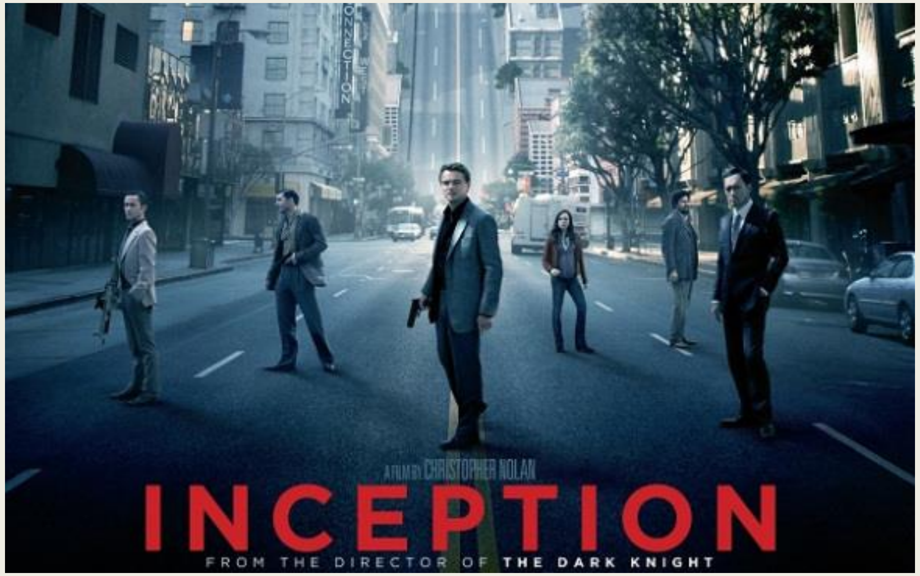
Funciones recursivas

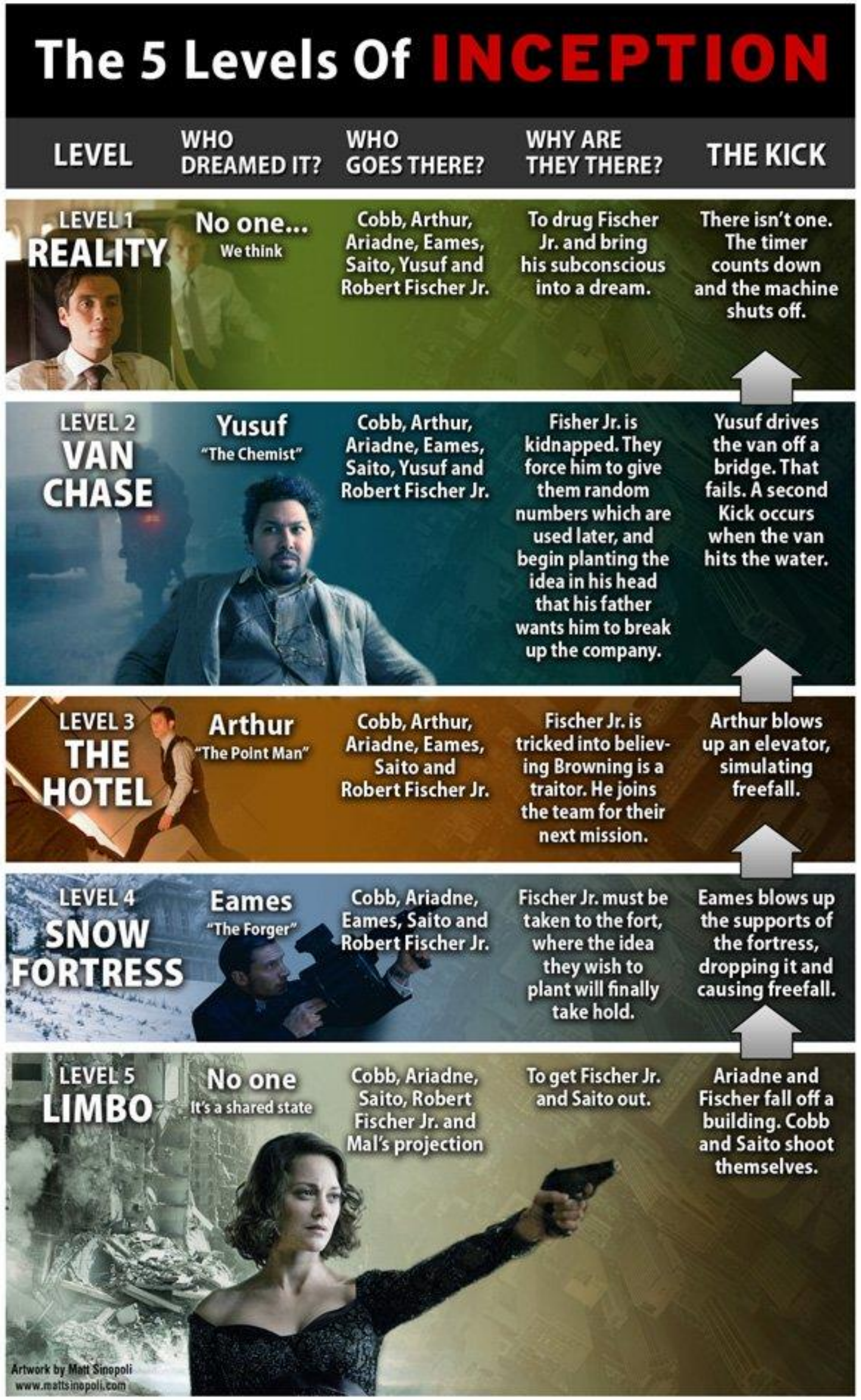
- A las funciones que tienen una estructura en la que se llaman, utilizan a si mismas, se les conoce como Funciones Recursivas.
- Que significa ser recursivo:
  - *Matemáticamente, es un proceso inductivo inverso, en el cual se define un resultado base y un resultado n (lejano), la función itera al llamarse a si misma de manera regresiva, parte en un estado n e itera hasta alcanzar el estado base.*

  - *Funcionalmente, es una función que define su resultado en términos de si misma, hasta un punto base, que tiene solución estática.*



Un sueño dentro de un sueño, dentro de un sueño…





The 5 Levels Of INCEPTION

LEVEL	WHO DREAMED IT?	WHO GOES THERE?	WHY ARE THEY THERE?	THE KICK
LEVEL 1 REALITY	No one... We think	Cobb, Arthur, Ariadne, Eames, Saito, Yusuf and Robert Fischer Jr.	To drug Fischer Jr. and bring his subconscious into a dream.	There isn't one. The timer counts down and the machine shuts off.
LEVEL 2 VAN CHASE	Yusuf "The Chemist"	Cobb, Arthur, Ariadne, Eames, Saito, Yusuf and Robert Fischer Jr.	Fisher Jr. is kidnapped. They force him to give them random numbers which are used later, and begin planting the idea in his head that his father wants him to break up the company.	Yusuf drives the van off a bridge. That fails. A second Kick occurs when the van hits the water.
LEVEL 3 THE HOTEL	Arthur "The Point Man"	Cobb, Arthur, Ariadne, Eames, Saito and Robert Fischer Jr.	Fischer Jr. is tricked into believing Browning is a traitor. He joins the team for their next mission.	Arthur blows up an elevator, simulating freefall.
LEVEL 4 SNOW FORTRESS	Eames "The Forger"	Cobb, Ariadne, Eames, Saito and Robert Fischer Jr.	Fischer Jr. must be taken to the fort, where the idea they wish to plant will finally take hold.	Eames blows up the supports of the fortress, dropping it and causing freefall.
LEVEL 5 LIMBO	No one It's a shared state	Cobb, Ariadne, Saito, Robert Fischer Jr. and Mal's projection	To get Fischer Jr. and Saito out.	Ariadne and Fischer fall off a building. Cobb and Saito shoot themselves.

Artwork by Matt Sinopoli
www.mattsinopoli.com

Ejemplo base, sucesión de Fibonacci

- La sucesión de Fibonacci, consiste en una sucesión de números en la cual, el siguiente numero de la sucesión es la suma de los dos números anteriores.
- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34…
- Por lo cual el valor de un numero en la posición N , es la suma de los números en (N - 1) y (N - 2)

- Fibonacci(0) = 0
- Fibonacci(1) = 1
- Fibonacci(1000) = ?

f(1000) = f(999) + f(998)

f(999) = f(998) + f(997)

f(998) = f(997) + f(996)

… …

f(997) = f(998) + f(997)

… …

f(998) = f(997) + f(996)

f(997) = f(996) + f(995)

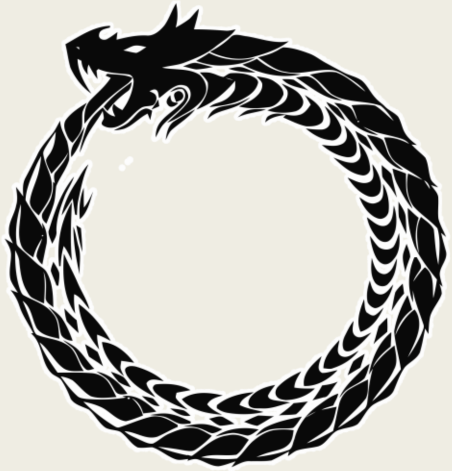
… …

f(996) = f(995) + f(994)

… …

De manera secuencial, es muy complejo…

```
[10]
1   #Sucesión de Fibonacci
2
3   def Fibonacci( n ):
4
5     if n == 0:
6       return 0
7
8     elif n == 1:
9       return 1
10
11    else:
12      return Fibonacci( n - 1 ) + Fibonacci( n - 2 )
13
14
15
```



En las líneas 5 – 9, podemos ver la definición de los casos bases

Hay 2 casos bases, por la definición:

Fibonacci(n) = Fibonacci(n - 1) + Fibonacci(n - 2)

En la línea 12, podemos ver donde esta implementada la recursión, al llamarse a si misma la función

Pero si se llama a si misma:

¿Cómo me aseguro que se detenga?

¿Puede correr de manera infinita?

¿Qué esta sucediendo?

Un sueño dentro de un sueño…

- Una de las características interesantes de un función recursiva, es que se le puede introducir cierta capacidad de “memoria”
- Además de que la función se llame a si misma, puede lograr que entienda en capa del sueño esta !!

```python
#Sucesión de Fibonacci con memoria !

def Fibonacci_mem( n , capa):
  print('Numero:',n,' Capa:',capa)
  capa = capa + 1

  if n == 0:
    print('Caso de salida 0')
    return 0

  elif n == 1:
    print('Caso de salida 1')
    return 1

  else:
    return Fibonacci_mem( n - 1, capa ) + Fibonacci_mem( n - 2, capa )

```

```
[24]  Fibonacci_mem( 4, 0 )
```

```
Numero: 4  Capa: 0
Numero: 3  Capa: 1
Numero: 2  Capa: 2
Numero: 1  Capa: 3
Caso de salida 1
Numero: 0  Capa: 3
Caso de salida 0
Numero: 1  Capa: 2
Caso de salida 1
Numero: 2  Capa: 1
Numero: 1  Capa: 2
Caso de salida 1
Numero: 0  Capa: 2
Caso de salida 0
3
```