Ejecución de un Proceso en Pila

- Ejemplo de cómo se ejecuta un proceso en su propia pila.
- Responderemos algunas preguntas.

Ejecución de un Proceso en Pila

```
1
2  class SumaRecur{
3
4
5      public int SumaR(int a){
6          if (a==0) return 0;
7          return (a + SumaR(a-1));
8      }
9
10
11
12     public static void main (String arg[]){
13         try{
14             System.out.println("El valor del sumatorio es " + new SumaRecur().SumaR(3));
15             //Thread.sleep(5000);
16         }
17         catch(Exception e){
18             e.printStackTrace();
19         }
20     }
21
22
23 }
```

Ejecución de un Proceso en Pila

- Imaginar que ejecutamos el mismo programa pero con dos terminales diferentes.
  - ¿Son cada uno dos procesos totalmente independientes?
  - ¿Cada uno tiene su propia pila de ejecución?
  - ¿Cómo se ejecutan en sus pilas?
  - ¿Comparten alguna variable u objeto?
  - ¿Son ejecuciones independientes del misma instancia o son ejecuciones de instancias diferentes?

SumaRecur.java (~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1/)

```java

class SumaRecur{


    public int SumaR(int a){
        if (a==0) return 0;
        return (a + SumaR(a-1));
    }



    public static void main (String arg[]){
        try{
            System.out.println("El valor del sumatorio es " + new SumaRecur().SumaR(3));
            Thread.sleep(5000);
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }

```

System

tema1 — -bash — 158×14

```
iMacdesantiago:tema1 santi$
iMacdesantiago:tema1 santi$
iMacdesantiago:tema1 santi$
iMacdesantiago:tema1 santi$
iMacdesantiago:tema1 santi$
iMacdesantiago:tema1 santi$ java SumaRecur
El valor del sumatorio es 6
iMacdesantiago:tema1 santi$
```

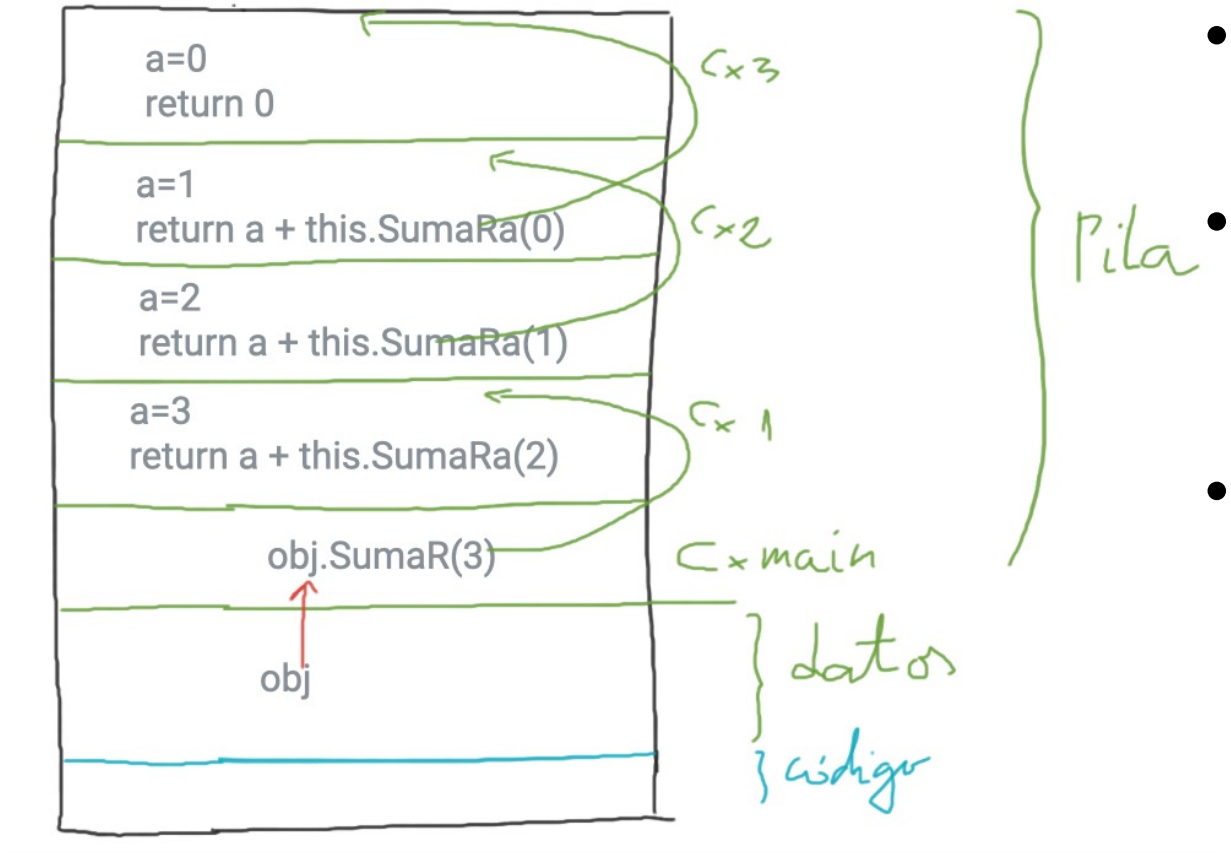
```
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1>
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1> javac SumaRecur.java
Process javac exited with code 0
~/Documents/trabajo_instituto/instituto_21_22/2DAM/Psp/tema1> java SumaRecur
```

Console

Ejecución de un Proceso en Pila

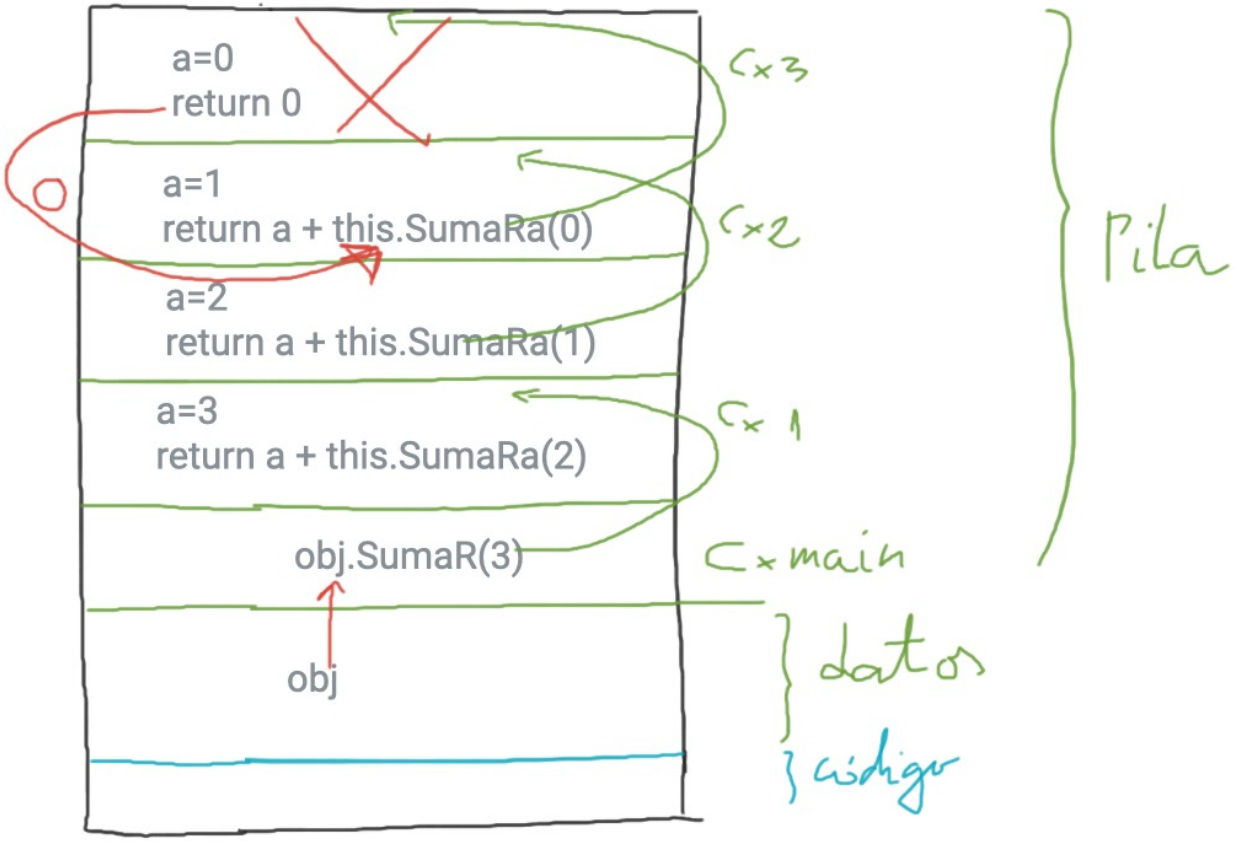
- Aunque sea el mismo programa, son dos ejecuciones de instancias completamente diferentes.
- El sistema operativo los ejecuta como si fueran programas totalmente diferentes, aunque el código .class sea el mismo.
- El S.O. asigna memoria totalmente independiente a cada uno de ellos y los ejecuta independientemente.
- De una forma vulgar, mostraremos la ejecución en su pila de ejecución.

Ejecución de un Proceso en Pila



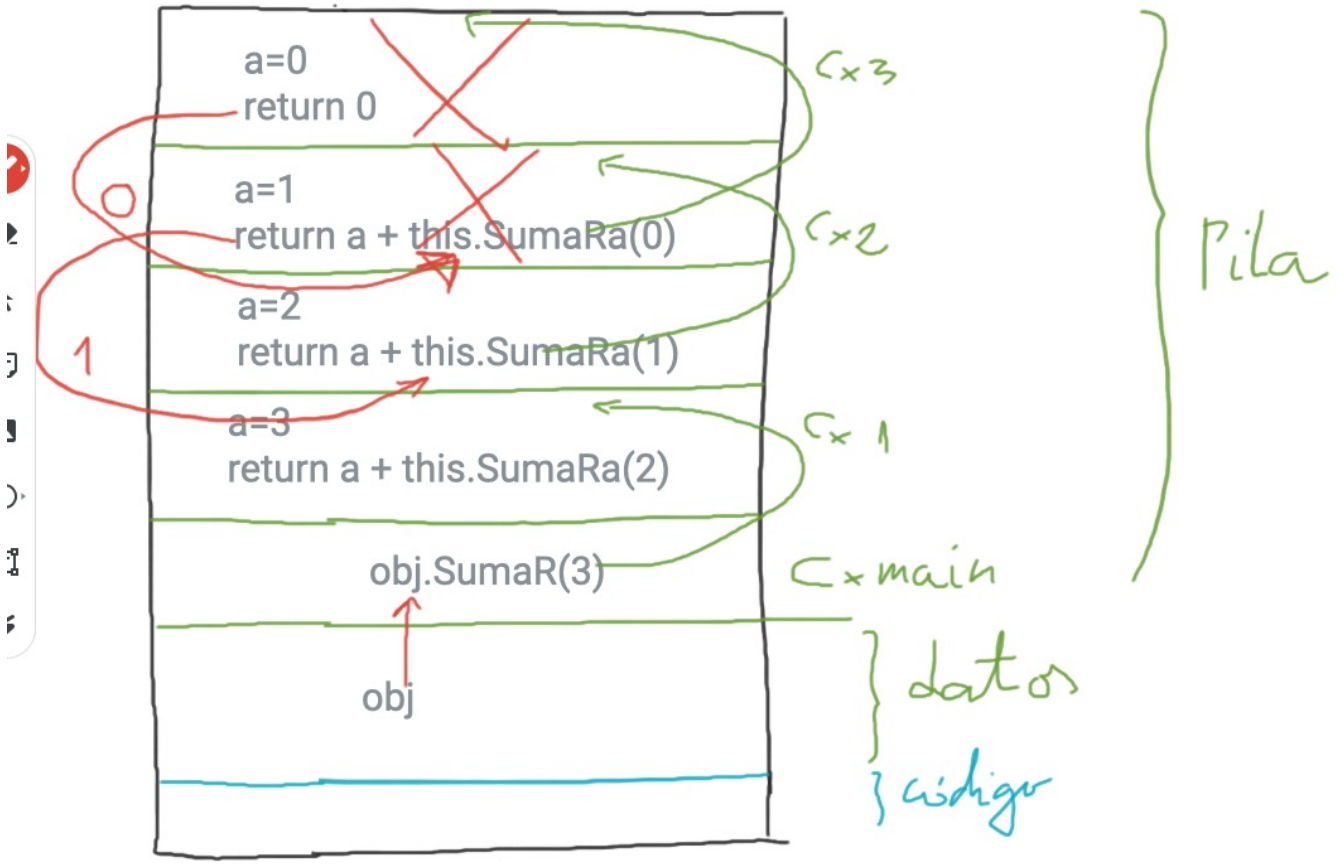
- El código se ejecuta en su pila de ejecución.
- Las llamadas recursivas van apilando contextos (Cx main, Cx1, Cx2, Cx3)
- Cuando se para la recursión, empiezan los retornos y a desapilar contextos.

Ejecución de un Proceso en Pila



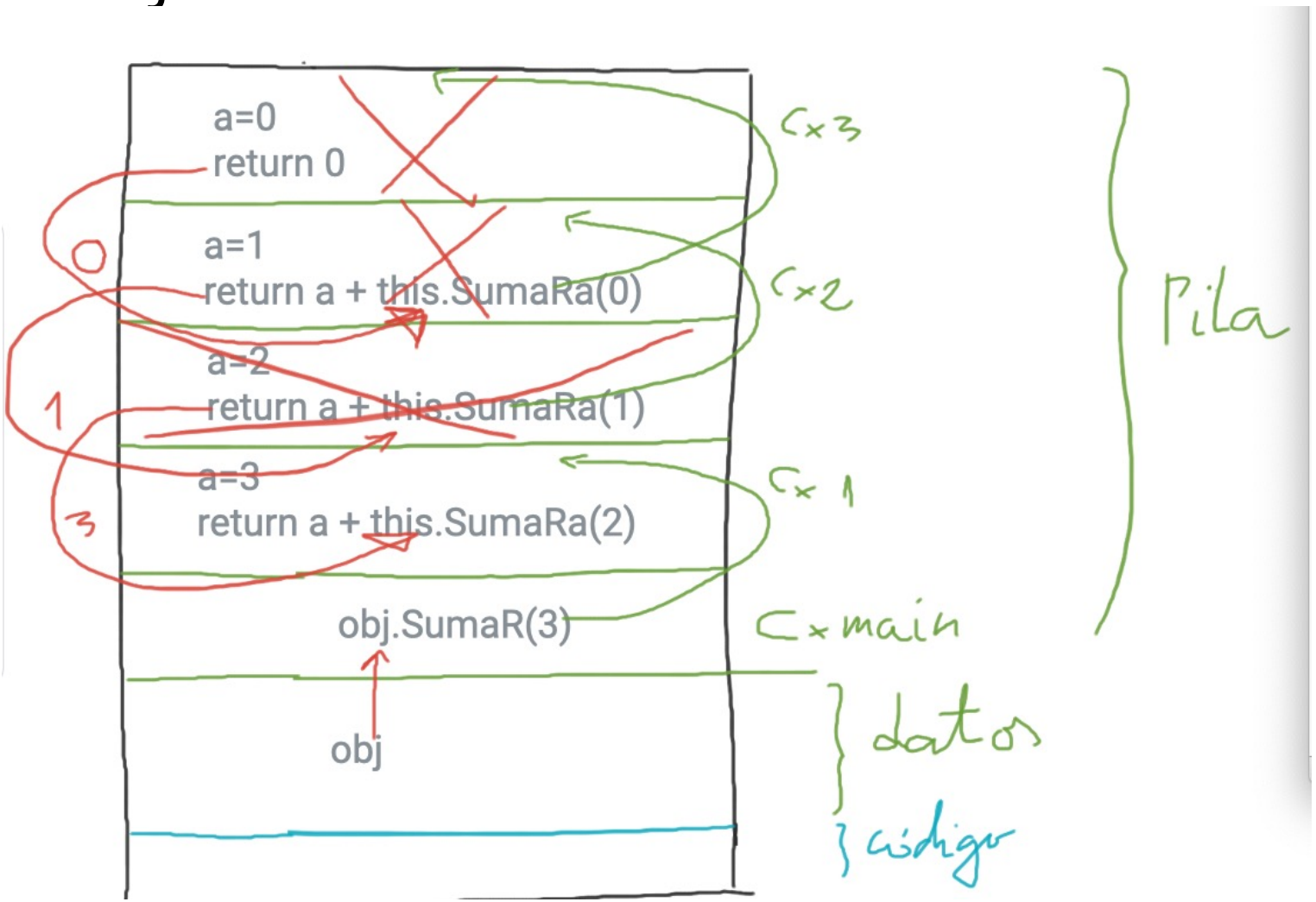
- Retorna un 0 y se elimina el contexto de la cabeza de la pila Cx3

Ejecución de un Proceso en Pila



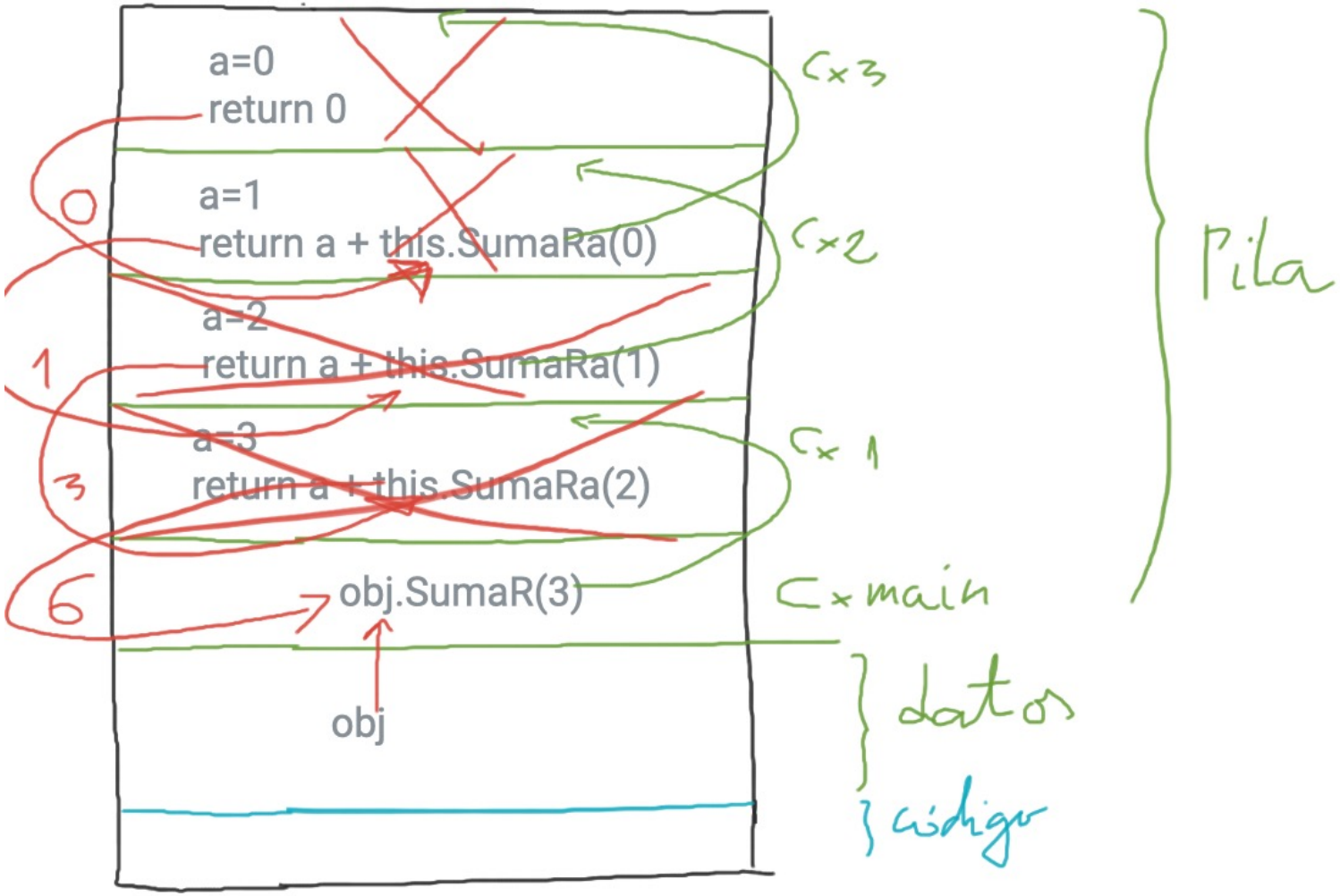
- Retorna un 1 y se elimina el contexto de la cabeza de la pila Cx2

Ejecución de un Proceso en Pila



- Retorna un 3 y se elimina el contexto de la cabeza de la pila (Cx1)

Ejecución de un Proceso en Pila



Retorna un 6 y Finaliza el contexto de la primera llamada a la función. Vuelve al main e imprime en pantalla.

Ejecución de un Proceso en Pila

- ¿Por qué este ejemplo?
  - Cada proceso, tiene su propia pila, su copia de código, sus variables, su estado de ejecución y demás información.
  - El sistema operativo decide en un momento dado qué proceso ejecutar y si tuviera que elegir a un proceso antes de que el actual finalize, éste debe guardar el estado actual del proceso.
  - Las pilas de ejecución son totalmente independientes entre cada uno de los procesos.