

¿Qué es un controlador de versiones?

Es una combinación de tecnologías y practicas para seguir y controlar los cambios realizados en los ficheros del proyecto, en particular en el código fuente, en la documentación y en las páginas web.

La razón por la cual el control de versiones es universal es porque ayuda virtualmente en todos los aspectos al dirigir un proyecto: comunicación entre los desarrolladores, manejo de los lanzamientos, administración de fallos, estabilidad entre el código y los esfuerzos de desarrollo experimental y atribución y autorización en los cambios de los desarrolladores.

¿Qué es Git?

Es el sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel de Linux.

Git, que presenta una arquitectura distribuida, es un ejemplo de DVCS (sistema de control de versiones distribuido, por sus siglas en inglés). En Git, la copia de trabajo del código de cada desarrollador es también un repositorio que puede albergar el historial completo de todos los cambios.

Además de contar con una arquitectura distribuida, Git se ha diseñado teniendo en cuenta el rendimiento, la seguridad y la flexibilidad.

¿Qué es GitHub?

GitHub es un sitio web y un servicio en la nube que ofrece un servicio de hosting de repositorios almacenados en la nube, ayuda a los desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código.

Esencialmente, hace que sea más fácil para individuos y equipos usar Git como la versión de control y colaboración.

Mi GitHub:

<https://github.com/SoySkiper>

```

4
5  arrAstetic = ['*', '*', '*']
6  arrNoAstetic = ['*']
7  end = ' '
8  astetic = '*'
9  salto = '\n'
10
11  ## Ejercicio 1:
12  def ejercicio1 (arrAs):
13      ## Solución en cadena
14      for indice in arrAs:
15          print(' '.join(arrAs))
16
17      print(salto)
18
19      ## Ejercicio (otra manera que se me ocurrió):
20      for contador in range(3):
21          print(arrNoAstetic*3)
22

```

Ejercicio 1:

```

* * *
* * *
* * *

```

```

['*', '*', '*']
['*', '*', '*']
['*', '*', '*']

```

```

23
24  ## Ejercicio 2:
25  def ejercicio2 (a):
26      for contador in range(3):
27          print(' '.join(a))
28          escalera = a.append(astetic)
29
30      del a[0:3]
31      print(salto)
32
33      ## Solución en lista
34      for contador in range(3):
35          print(a)
36          escalera = a.append(astetic)
37
38

```

Ejercicio 2:

```

*
* *
* * *

```

```

['*']
['*', '*']
['*', '*', '*']

```

```

38
39  ## Ejercicio 3:
40  def ejercicio3 (arrA):
41      for contador in range(3):
42          print(arrA)
43          longitud = len(arrA) - 1
44          del arrA[longitud]
45          longitud -= 1
46
47      arrA.extend(['*', '*', '*'])
48      print(salto)
49
50      for contador in range(3):
51          print(' '.join(arrA))
52          longitud = len(arrA) - 1
53          del arrA[longitud]
54          longitud -= 1
55

```

Ejercicio 3:

```

['*', '*', '*']
['*', '*']
['*']

```

```

* * *
* *
*

```

```

56
57  ## Ejercicio 4:
58  def ejercicio4 (listAstetic):
59      for contador in range(3):
60          print(listAstetic)
61          longitud = len(listAstetic) - 1
62          del listAstetic[longitud]
63          listAstetic.insert(contador, end)
64          longitud -= 1
65
66      del listAstetic[0:3]
67      listAstetic.extend(['*', '*', '*'])
68      print(salto)
69
70      for contador in range(3):
71          print(' '.join(listAstetic))
72          longitud = len(listAstetic) - 1
73          del listAstetic[longitud]
74          listAstetic.insert(contador, end)
75          longitud -= 1
76

```

Ejercicio 4:

```

['*', '*', '*']
[' ', '*', '*']
[' ', ' ', '*']

```

```

* * *
 * *
  *

```

Llamadas de funciones

```
77
78 print('Ejercicio 1:' + salto)
79 ejercicio1(arrAstetic)
80 print(salto + 'Ejercicio 2:' + salto)
81 ejercicio2(arrNoAstetic)
82 print(salto + 'Ejercicio 3:' + salto)
83 ejercicio3(arrAstetic)
84 print(salto + 'Ejercicio 4:' + salto)
85 ## Reinicio variable
86 arrAstetic.extend(['*', '*', '*'])
87 ejercicio4(arrAstetic)
```