

# Actividad 2 - Figuritas

Exactas Programa

Verano 2020

## Las figuritas del mundial

El objetivo de esta actividad será realizar un programa en **Python** que responda la pregunta:

*¿Cuántas figuritas hay que comprar para completar el álbum del Mundial?*

**Deben subir la tarea a:** <http://bit.do/entregas-v2020> .

### Datos

- Álbum con 670 figuritas.
- Cada figurita se imprime en cantidades iguales y se distribuye aleatoriamente.
- Cada paquete trae cinco figuritas.

Vamos a utilizar este disparador para presentar conceptos claves en cualquier lenguaje de programación.

### Otras herramientas útiles de Python

Para que estén disponibles más funciones de **Python**, tenemos que utilizar el comando `import`. En particular, en esta actividad vamos a usar dos módulos:

- El módulo `random` ya lo usamos en la actividad anterior, el comando es `import random`.
- Vamos a utilizar un nuevo módulo que se usa mucho para operar numéricamente: `numpy`. El comando es: `import numpy as np`<sup>1</sup>.

### Ejercicios iniciales

Experimente los siguientes comandos *en la consola* (debe tener importados los módulos):

1. `random.random()`
2. `random.randint(1, 10)`
3. `lista = np.arange(10)`
4. `print(lista)`
5. `np.mean(lista)`

### El modelo del álbum de figuritas

Vamos a representar un álbum de  $n$  figuritas utilizando una lista de  $n$  posiciones numeradas de 0 a  $n - 1$ . Cada posición representa el estado de una figurita con dos valores: 0 para indicar que aún no la conseguimos y 1 para indicar que sí.

Por ejemplo, si tuviéramos un álbum de seis figuritas vacío lo vamos a representar como `[0, 0, 0, 0, 0, 0]`. Cuando consigamos la figurita 3 tendremos que indicarlo poniendo un 1 en el tercer lugar de la lista, es decir `album[2]=1` y el álbum nos va a quedar `[0, 0, 1, 0, 0, 0]`.

---

<sup>1</sup>Si están usando su propia máquina con Linux, deben tener el paquete `numpy` instalado

## Primera simplificación

Suponga que las figuritas se compran **individualmente** (de a una, no en un paquete con varias). En este caso, **la dinámica** del llenado es la siguiente:

- A) Iniciamos con un álbum vacío y sin haber comprado ninguna figurita.
- B) Compramos figuritas (de a una) hasta llenar el álbum; es decir, se repite la acción (*el paso*) de comprar figuritas *mientras* el álbum este incompleto.
- C) Al terminar nos interesa saber cuántas figuritas tuvimos que comprar para llenar el álbum.

## Ejercicios

Vamos ahora a implementar computacionalmente este modelo:

1. Implemente la función `crear_album(figus_total)` que devuelve un álbum *vacío* con `figus_total` espacios para pegar figuritas.
2. Implemente la función `hay_alguno(l,e)` que recibe una lista `l` y un elemento `e` y devuelve `True` si la lista contiene el elemento `e`. En el caso en que no esté, debe devolver `False`.
3. Alguna de las funciones que probamos antes sirve para devolver un número aleatorio dentro de un rango (¿Cuál era?). Implemente la función `comprar_una_figu(figus_total)` que recibe el número total de figuritas que tiene el álbum (dado por el parámetro `figus_total`) y devuelve un número entero aleatorio que representa la figurita que nos tocó.
4. Implemente la función `cuantas_figus(figus_total)` que genere un álbum de tamaño `figus_total`, simule su llenado y devuelva la cantidad de figuritas que se debieron adquirir para completarlo.
5. Utilizando la función `cuantas_figus`, repita el experimento 5 veces, y calcule el promedio de los resultados.
6. Utilizando la función `cuantas_figus`, implemente una función `experimentar(figus_total, n_rep)` que simule `n_rep` veces el llenado de un álbum y devuelva los resultados en una lista.
7. Realice `n_rep=1000` repeticiones utilizando la función anterior con `figus_total=6` para estimar cuántas figuritas hay que comprar, en promedio, para completar el álbum. **Ayuda:** El comando `np.mean(l)` devuelve el promedio de la lista `l`.
8. Realice `n_rep=100` repeticiones utilizando la función anterior con `figus_total=670` para estimar cuántas figuritas hay que comprar, en promedio, para completar el álbum (de 670 figuritas).

## Ahora con paquetes

Para pensar y discutir con su compañero o alguna de sus otras personalidades:

- Describa en palabras cómo impacta en lo realizado tener paquetes con figuritas.
- ¿Cómo puede representar un paquete?

## Ejercicios

1. Simule la generación de un paquete con cinco figuritas, sabiendo que el álbum es de 670. Notemos que, como en la vida real, puede haber figuritas repetidas en un paquete.
2. Implemente una función `generar_paquete(figus_total, figus_paquete)` que, dado el tamaño del álbum (`figus_total`) y la cantidad de figuritas por paquete (`figus_paquete`), genere un paquete de figuritas al azar.

3. Implemente una función `cuantos_paquetes(figus_total, figus_paquete)` que genere un álbum del tamaño dado, simule su llenado, y devuelva cuántos paquetes se debieron adquirir para completarlo.
4. Implemente una función `experimentar_con_paquetes(figus_total, figus_paquete, n_rep)` que repita el experimento del llenado de un álbum con paquetes y devuelva los resultados en una lista.
5. Utilice la función `experimentar_con_paquetes` para repetir 100 veces el experimento y estimar cuántos paquetes hay que comprar, en promedio, para completar un álbum de 670 figuritas con paquetes de 5 figuritas.

## Optativos

1. *Optativo 1*: Utilizando lo implementado en el ítem 4, estime la probabilidad de completar el álbum con 850 paquetes o menos.
2. *Optativo 2*: Utilizando lo implementado en el ítem 4, estime cuántos paquetes debe comprar para tener una chance del 90 % de completar el álbum.
3. *Optativo 3*: Repita suponiendo ahora que no hay figuritas repetidas en un paquete.