

Proyecto 1 - Datos y visualización

Nota sobre nombres de funciones y variables:

Los nombres de las funciones y parámetros deben ser los indicados en las consignas, pues se probarán con un script automatizado. Si desean funcionalidades o parámetros crear una segunda versión de la función con nombre distinto.

Todos los nombres de variables (incluyendo filas y columnas) **deben** ser escritos **en minúscula**. Por ejemplo: los días deberían llamarse ["lunes", "martes", (...)] en lugar de ["Lunes", "Martes", (...)].

Ejercicio 1.1

Crear el siguiente DataFrame llamado "**df_semanal_sin_cargar**". Esta tabla será la base para cada semana.

Tener en cuenta las siguientes consideraciones:

- Los nombres de las columnas serán los días de la semana.
- Los nombres de las filas serán los nombres de los ejercicios.
- Los datos del dataset serán inicializados *todos* en cero.

df_semanal_sin_cargar:

	lunes	martes	miercoles	jueves	viernes	sabado	domingo
flexiones	0	0	0	0	0	0	0
abdominales	0	0	0	0	0	0	0
sentadillas	0	0	0	0	0	0	0

Tip:

Pueden crear la plantilla de ceros en **Numpy**, de dimensiones (3,7).

Luego convertirlo en un DataFrame de **Pandas**.

Luego editar los nombres de filas y columnas con **Pandas**.

Ejercicio 1.2

Crear la función:

```
def crear_plantilla_de_ejercicios_semanal(lista_de_ejercicios)
```

Esta función **retorna** un DataFrame con los ejercicios indicados en la lista. Es solo el ejercicio anterior dentro de una función y con la opción de pasarle los ejercicios que tendrá dentro de una lista.

Tip: La cantidad de filas que deberá tener el arreglo de ceros base será igual a la cantidad de ejercicios que haya en la lista indicada.

Ejercicio 2.1

Crear la función:

```
def agregar_ejercicio(ejercicio_nuevo, df_base)
```

Esta función **retorna** una copia del DataFrame “**df_base**” que con una fila adicional contiene el ejercicio nuevo. Esta nueva fila también debe ser cargada con ceros.

Ejemplo, al usar:

```
df_nuevo = agregar_ejercicio("plancha", df_semanal_sin_cargar)
```

Nos retorna el siguiente DataFrame en **df_nuevo**:

	lunes	martes	miercoles	jueves	viernes	sabado	domingo
flexiones	0	0	0	0	0	0	0
abdominales	0	0	0	0	0	0	0
sentadillas	0	0	0	0	0	0	0
plancha	0	0	0	0	0	0	0

Tip:

*Pueden crear primero un DataFrame de **Pandas** con el nombre del ejercicio a agregar y los ceros para cada día (igual que hicieron en el ejercicio anterior pero para un solo ejercicio en lugar de 3).*

Luego unirlo (concatenar) con el DataFrame que contiene todos los ejercicios.

Ejercicio 2.2

Editar la función anterior para que solo agregue la nueva fila **si no existe una fila con el nombre del ejercicio**.

Es decir, que no pueda haber ejercicios duplicados.

Ejercicio 3

Crear la función:

```
def eliminar_ejercicio(ejercicio_a_eliminar, df_base)
```

Esta función **retorna** una copia del DataFrame “**df_base**” sin el ejercicio indicado.

Si el ejercicio no existe, informar.



Ejercicio 4

Crear la función:



```
def completar_dia(dia, arreglo_de_repeticiones)
```

Esta función retorna una copia del DataFrame original con las repeticiones cargadas en el día indicado cargado.

El arreglo de repeticiones debe tener el mismo tamaño que la cantidad de ejercicios del DataFrame, es decir: si tenemos tres ejercicios: flexiones, abdominales, sentadillas; entonces “**arreglo_de_repeticiones**” deberá tener tres elementos.

Si el arreglo no tiene la misma cantidad de elementos que la cantidad de ejercicios, indica un error y no carga nada.

Tip:

Para probar la carga de días puede generar un vector de enteros aleatorio en lugar de cargar manualmente.

Ejercicio 5

Crear la función:

```
def graficar_ejercicio(ejercicio, df_base)
```

Esta función graficará la cantidad de repeticiones del ejercicio indicado a lo largo de la semana. Elija un gráfico que demuestre claramente esta información y justifique su elección.

Ejercicio 6

Crear la función:

```
def graficar_proporcion_ejercicios(ejercicio, df_base)
```

Esta función sumará la cantidad de repeticiones de cada ejercicio a lo largo de la semana. Luego, mostrará en un único gráfico la proporción de cada ejercicio respecto al total. Elija un gráfico que demuestre claramente esta información y justifique su elección.

Ejercicio 7

Crear una librería llamada “app_ejercicios” que contenga todas estas funciones. La librería generada debe ser descargada y enviada adjunta.



Ejercicio 8

Crear un menú que permita utilizar todas estas funciones. Para esto crear también las variables del entorno necesarias.