

# 1er Recuperatorio Algoritmos I

Cátedra Lanzillotta

16/11/2023

## Ejercicio 1

La empresa organizadora del gran evento deportivo del ciclismo el “**Giro d’Italia**” nos contrata para realizar una aplicación para la acreditación, control de tiempos y estadísticas de la competencia.

La competencia está compuesta por etapas, donde cada etapa une distintas ciudades de Italia. Cada año el circuito es modificado, transformando la competencia en un evento único por su belleza y altísima complejidad deportiva.

El ganador de cada etapa (tiempo más rápido de la etapa) recibe la “maglia rosa”, un jersey color rosa que lo identifica entre todos los corredores.

Cada corredor forma parte de un equipo deportivo (*Bora Hansgrohe, Ineos Grenadiers, Jumbo – Visma, entre otros*). **El ganador del Giro será aquel que en la sumatoria de tiempos de las etapas posea el menor tiempo total.**



El programa deberá mostrar un menú con los siguientes puntos.

- 1) **Carga y eliminación** de un competidor. Para la carga de un competidor se solicitará Nombre, Apellido, equipo al que pertenece, edad.

### ATENCIÓN:

- Se deben listar los equipos disponibles y se debe validar que exista al menos un equipo deportivo cargado)
- Para la carga se debe generar un Nro de corredor único, puede ser un valor consecutivo.

- 2) **Carga** del equipo deportivo. Se debe ingresar el nombre del equipo.

- 3) **Simular etapa.** Se deberá ejecutar la función *simular\_etapa*, tantas veces como etapas se quieran simular, se deberá controlar y saber el nro de ejecución (etapa realizada). La misma recibe una lista con el nro de corredor y devuelve un diccionario {nro\_corredor: tiempo\_etapa}.

*Al final del examen se encuentra el código de la función simular\_etapa*

- 4) Lista los corredores que han llevado la “maglia rosa” durante la competencia, indicando el nro de etapa.
- 5) Mostrar el ganador y el resultado de los Top 10
- 6) Listar los corredores que hayan obtenido más de una vez una “maglia rosa” durante la competencia.

## Ejercicio 2.

Crear una función que determine si los comentarios de un sitio web tienen una connotación positiva o negativa.

Para esto se pide crear una función que reciba una cadena de texto y devuelva una cadena con la palabra “POSITIVA”, “NEGATIVA” o “NEUTRA” según corresponda.

Para determinar si una cadena de texto es positiva o negativa habrá que contar la cantidad de palabras con connotación positiva o negativa que se encuentren en el texto. Se considerará NEUTRA en caso que la cantidad de palabras positivas sea igual que las negativas o que no haya ninguna positiva y negativa.

**Palabras positivas:** (optimista, estupendo, excepcional, extraordinario, orgulloso, agradable, fantástico, alegre, feliz, paciente, amable, fiel, positivo, atento, genial, bondadoso, gran, buen, hermoso, honrado, contento, interesante, linda, maravilloso, encantador)

**Palabras negativas:** (Agrandado, Agresivo, Antipático, Avaro, Caprichoso, Corrupto, Dañino, Desconsiderado, Despiadado, Despreciable, Egocéntrico, Egoísta, Engreído, Envidioso, Falso, Farsante, Feo, Hipócrita, Horrible, Insoportable, Intolerante, Maleducado, Malo, Mentiroso, Negativo, Pedante, Resentido, Tacaño, Terco, Tirano, Vengativo, Violento)

---

## Función simular\_etapa

```
import random

def simular_etapa(participantes: list)->dict:
    """
    Pre: Recibe una lista, con los nros de corredores.
    Post: Devuelve un diccionario, donde la "key" son los nros de corredores y el "value"
    es el tiempo
    realizado en esa etapa.
    """
    tiempos={}
    for participante in participantes:
        tiempos[participante] = random.randint(10, 160)

    print("Los tiempos de todos los participantes fueron: ")
    for key,value in tiempos.items():
        print("- ",key," --> ",value, " min")

    return tiempos
```