

CC3001 Algoritmos y Estructuras de Datos**Profesores:** Nelson Baloian, Patricio Poblete, e Iván Sipirán**Auxiliares:** Valentína Alarcón Yañez, Antonia G. Calvo, Cristián Llull, y Raimundo Lorca Correa**Auxiliar 0**

14 de marzo de 2025

Bienvenidos a Algoritmos y Estructuras de Datos. Durante el ramo utilizaremos Google Colab para entregar los ejercicios y tareas, por lo que vamos a comenzar con un pequeño repaso de Python 3, Google Colab, y Numpy.

P1. Intro a Google Colab

Durante el semestre haremos uso extensivo de Google Colab, una plataforma que provee un entorno virtual de Python (y algunas otras cosas). Además, durante parte del semestre, utilizaremos la librería **AED-utilities**, por lo que le facilitamos un tutorial que explica el uso básico de la librería.

- (a) Para comenzar, visite el tutorial que está en <https://colab.research.google.com/github/ivansipiran/aed-utilities/blob/main/tutorial/tutorial.ipynb>
- (b) Copie este notebook a su cuenta personal de Colab.
- (c) Ejecute la primera celda del tutorial.
- (d) Comparta su copia del notebook por un enlace público.
- (e) Descargue el cuaderno resultante.
- (f) **Propuesto:** Haga el tutorial! *No dude en consultar a sus Auxiliares!*

P2. Numpy

El objetivo de esta pregunta es familiarizarnos con la estructura de datos Numpy la cual utilizaremos en remplazo a las listas de Python. A diferencia de estas últimas, Numpy permite controlar el uso de memoria y los recorridos a través de esta de manera eficiente.

Por lo tanto, esta pregunta consiste de problemas cortos utilizando esta estructura.

- (a) Crear un arreglo de enteros tamaño 10 que solo contenga ceros.
- (b) Itere por el arreglo escribiendo los números del 0 al 9.
- (c) Multiplique cada uno de los elementos del arreglo por 3.
- (d) Cree una matriz de 4x4 que solo contenga unos e imprima la matriz.
- (e) Reemplace los unos en la matriz por los números del 4 al 19 e imprima la matriz.
- (f) **Propuesto:** En el arreglo, remplace los números pares por 0 y los impares por 1.

P3. Invariante

Dado un string, diseñe un algoritmo iterativo que determine si este es palíndromo. Un string es palíndromo si se lee igual al revés y al derecho.

Ejemplo:

```
esPalindromo("Oshawott")  
  
False  
  
esPalindromo("Eevee")  
  
True  
  
esPalindromo("Ho Oh")  
  
True  
  
esPalindromo("Alomomolaa")  
  
False
```

P4. Clases, Objetos, y Pokémon

El objetivo de esta pregunta es recordar cómo implementar y usar clases en Python. Para esto, modelaremos una batalla pokémon por medio de la creación de una clase Pokémon y Entrenador con los siguientes atributos y métodos.

(a) Cree las siguientes clases y funciones.

(i) Clase Pokémon:

- Atributo `nombre`: `str`
- Atributo `hp`: `int`
- Atributo `dueño`: `Entrenador`
- Método `atacar(otroPokemon: Pokemon): None`
- Método `estaVivo(): bool`

(ii) Clase Entrenador:

- Atributo `nombre`: `str`
- Atributo `equipo`: `???`
- Método `capturar(pokemon: Pokemon): None`

(b) Cree dos Pokémon y dos Entrenadores. Luego, haga que capturen uno de los `Pokemon` cada uno.

(c) **Propuesto:** Cree la función `batallar(entrenador1, entrenador2)` y haga que ambos entrenadores batallen.

Contexto: En Pokémon, un Entrenador tiene un equipo compuesto por un número finito y pequeño de Pokémon. Para iniciar una batalla, un entrenador desafía a otro, y ambos despliegan al primer pokémon “vivo” (“despierto”) de su respectivo equipo. Las batallas Pokémon son combates por turnos donde dos entrenadores usan sus Pokémon para reducir los HP del oponente a cero, cuando este es reemplazado por el siguiente pokémon del equipo de su entrenador. Este proceso se repite hasta que un entrenador se queda sin pokémon despiertos, en cuyo caso gana su contrincante. Supondremos que un pokémon hace una cantidad de daño proporcional a su vida actual.