

# PRÁCTICA Nº 3

Cátedra Programación II

Agosto 2023

## 1. Listas

**EJERCICIO 1.** Escriba una función `posicionesMultiplo` que tome una lista y un número y retorne la lista formada por los elementos que están en las posiciones múltiplos de ese número. Por ejemplo: `posicionesMultiplo([1,2,3,4,5,6,7],2)` retorna `[1,3,5,7]` y `posicionesMultiplo([1,2,3,4,5,6,7],3)` da como resultado `[1,4,7]`.

**EJERCICIO 2.** Escriba una función que tome una lista de números y devuelva la suma acumulada, es decir, una nueva lista donde el primer elemento es el mismo, el segundo elemento es la suma del primero con el segundo, el tercer elemento es la suma del resultado anterior con el siguiente elemento y así sucesivamente. Por ejemplo, la suma acumulada de `[1,2,3]` es `[1, 3, 6]`.

**EJERCICIO 3.** Escriba una función llamada `elimina` que tome una lista y elimine el primer y último elemento de la lista. La función debe devolver una nueva lista con los elementos que no fueron eliminados.

**EJERCICIO 4.** Escriba una función `ordenada` que tome una lista como parámetro y devuelva `True` si la lista está ordenada en orden ascendente y `False` en caso contrario. Por ejemplo, `ordenada([1, 2, 3])` retorna `True` y `ordenada(['b', 'a'])` retorna `False`.

**EJERCICIO 5.** Escriba una función llamada `duplicado` que tome una lista y devuelva `True` si tiene algún elemento duplicado. La función no debe modificar la lista.

**EJERCICIO 6.** Escriba una función llamada `eliminaDuplicados` que tome una lista y devuelva una nueva lista con los elementos únicos de la lista original. No tienen porque estar en el mismo orden. Ayuda: puede utilizar la función `sort`.

**EJERCICIO 7.** Escriba una función que tome una lista y retorne la cantidad de elementos distintos que tiene. Se recomienda usar la función del ejercicio anterior.

**EJERCICIO 8.** Para comprobar si una palabra está en una lista se puede utilizar el operador `in`, pero sería una búsqueda lenta ya que busca a través de las palabras según el orden en el cual aparecen en la lista. Si la lista almacena las palabras en orden alfabético, podemos acelerar las cosas con una búsqueda dicotómica (también conocida como búsqueda binaria). Esta es similar a lo que hacemos cuando buscamos una palabra en el diccionario. Comenzamos por el centro y comprobamos si la palabra que buscamos está antes o después del centro. Si está antes, buscamos sólo en la primera mitad, si está después, buscamos en la segunda mitad de la lista. Con esto podemos reducir el tiempo de búsqueda.

Implemente la función `busquedaDicotomica` que toma una lista de palabras ordenadas alfabé-

ticamente y una palabra a buscar y retorna si la palabra está en la lista o no.

## 2. Cadenas

**EJERCICIO 9.** Escriba un programa que tenga una función que tome una cadena y muestre cada caracter que la forma del último al inicial.

**EJERCICIO 10.** Escriba un programa que contenga a la función `contar(1, x)` que cuente cuántas veces aparece un carácter `1` dado en una cadena `x`.

**EJERCICIO 11.** Escriba un programa que cuente cuántas veces aparecen cada una de las vocales en una cadena. No importa si la vocal aparece en mayúscula o en minúscula.

**EJERCICIO 12.** Escriba un programa que contenga una función que reciba como parámetro una cadena de palabras separadas por espacios y devuelva como resultado cuántas palabras de más de cinco letras tiene la cadena dada.

## 3. Tuplas

### EJERCICIO 13. Cartas como tuplas

- a) Proponga una representación con tuplas para las cartas de la baraja francesa.
- b) Escriba una función llamada `poker` que reciba cinco cartas de la baraja francesa e informe (devuelva el valor lógico correspondiente) si esas cartas forman o no un póker (es decir que hay 4 cartas con el mismo número).

### EJERCICIO 14. El tiempo como tuplas

- a) Proponga una representación con tuplas para el tiempo.
- b) Escriba una función `sumaTiempo` que reciba dos tiempos dados y devuelva su suma.

**EJERCICIO 15.** Escriba una función `diaSiguienteE` que, dada una fecha expresada como la tupla *(Día, Mes, Año)* (donde *Día*, *Mes* y *Año* son números enteros), calcule el día siguiente al dado en el mismo formato.

**EJERCICIO 16.** Escriba una función `diaSiguienteT` que, dada una fecha expresada como la tupla *(Día, Mes, Año)* (donde *Día* y *Año* son números enteros, y *Mes* es el texto "Ene", "Feb", ..., "Dic", según corresponda), calcule el día siguiente al dado en el mismo formato.

### EJERCICIO 17. Dominó

- a) Escriba una función que tome dos fichas de dominó e indique si encajan o no. Las fichas son recibidas en dos tuplas, por ejemplo: *(3, 4)* y *(5, 4)*.
- b) Escriba una función que tome dos fichas de dominó e indique si encajan o no. Las fichas son recibidas en una cadena, por ejemplo: `"[3-4 2-5]"`. Nota: utilizar la función `split` de cadenas.