

## Problema 1

---

### Objetivos

---

- Implementar el TAD lista doblemente enlazada respetando la especificación lógica proporcionada
- Realizar una gráfica de N (cantidad de elementos) vs tiempo de ejecución para los siguientes métodos: len, copiar e invertir.
- Explicar los resultados y deducir los órdenes de complejidad a partir de las gráficas.

---

### Desarrollo

---

De los métodos a analizar en este programa, y de acuerdo a las gráficas podemos concluir:

#### Método len:

El método `__len__` es el mas optimo con una complejidad de  $O(1)$ , ya que solo devuelve el valor de la variable interna `self.tamano` que se actualiza cada vez que se agrega o elimina un elemento.

#### Método copiar:

Este método utiliza un ciclo comenzando por la cabeza de la lista hasta llegar al final (se ejecuta n veces). Tiene una complejidad  $O(n)$ .

#### Método invertir:

Este método también recorre la lista completa, el bucle se ejecuta n veces para una lista de n elementos, lo que significa que el tiempo de ejecución es proporcional al tamaño de la lista. Complejidad  $O(n)$