

# Taller 1: repaso de punteros y operaciones con vectores

---

1. Qué es un operador de dereferenciación y qué es un operador de dirección de memoria. ¿Cómo se relacionan?
  1. Un operador de dereferenciación, como su nombre lo indica, es un operador que "quita" el puntero, y retorna el valor que está almacenado en dicha posición de memoria.
  2. Los operadores de dirección de memoria se refiere, en cambio, a operadores que nos regresan una dirección de memoria en la que está almacenada un valor.
  3. Estos se relacionan en tanto nos permiten realizar operaciones directamente en la memoria, es decir, nos permiten realizar el manejo manual del como guardamos datos dentro de nuestro código.
2. Si se usan estructuras y punteros ¿Cuál es la diferencia entre usar el operador `.` Y el operador `->` ?
  1. El operador punto nos permite acceder al objeto hijo de un objeto. La `->`, hace lo mismo pero trabajando con punteros. Es decir que realiza la dereferenciación antes, es decir, es equivalente a hacer `(*ptr).campo`.
3. Resultados de la implementación
  1. Tras realizar la implementación y ejecución del código, se tuvo como resultado los tiempos de ejecución. Estos pueden verse a continuación.

```
./a.out -n 1000 -i malloc
Execution time: 3855

./a.out -n 1000 -i new
Execution time: 3881

./a.out -n 1000 -i vec
Execution time: 7818

./a.out -n 1000 -i vec
Execution time: 15002
```

De esto, tenemos que la implementación con `malloc` fue la implementación más rápida, probablemente a su manejo directo a la memoria; mientras que `vecvec`, o la implementación con vectores en 2d, fue la más lenta y por mucho.