# Universidad de Costa Rica

# Ciencias de Computación e Informática

# CI-1201 Programación II

# Profesor Adolfo Di Mare

Datos de prueba para la [Matriz Chirrisquitica](http://www.di-mare.com/adolfo/p/Matrix.htm)

# Gabriel Castañaza B31520 ([gcmelmejor@gmail.com](mailto:gcmelmejor@gmail.com))

# David Ceciliano B31676 ([dacs200595@gmail.com](mailto:dacs200595@gmail.com) )

Memoria Dinámica

Uno de los grandes hitos de la computación, fue la introducción de la memoria dinámica, que ha permitido un mejor manejo de recursos, y otorgándole a los programadores la posibilidad de elaborar rutinas de mucha más fluidez y movimiento, con un uso de la memoria más eficientemente. La memoria dinámica se basa, simplificando el término a grandes rasgos, en variables cuyo identificador se conoce, ya que son creadas de manera más procedimental, no declarándose explícitamente en el código, sino durante el tiempo de ejecución. Un programa de uso estático de la memoria, “pide” al sistema operativo cierta cantidad de memoria, que sabe que requerirá. Esto, al ser fijo, podría provocar desborde, pero al implementar el uso de memoria dinámica, la memoria disponible para correr el programa, está en constante movimiento, ajustándose a las necesidades de la tarea. Para lograrlo, el compilador y el programador deben de jugar con punteros, para asignar los espacios de la memoria adecuadamente, y evitar la sobreescritura de datos, haciendo de este sistema de punteros, fundamental.

Además, las fluctuaciones de memoria en un programa que utilice memoria dinámica, no sería posible, o no tendría un fin claro, si los datos ya no utilizados, no se les asignará como espacios vacíos, para recuperar la memoria sin utilizar. Sería un descontrolado uso de la memoria hasta llegar al desborde. Para esto, el compilador utiliza la función virtual del destructor, el cual libera toda la memoria utilizada por un objeto, al momento de ser desechado. Por esta razón, es vital un diseño inteligente de destructores, para asegurar que la memoria sea liberada correctamente, y quitar los comunes “huecos” en la memoria, los cuales son espacios inutilizados a los cuales no se tiene acceso, ya que no han sido liberados. Y aquí, es donde el [Deja Huecos](#https://github.com/soygcm/Programacion2/blob/master/Tarea5/NOcompila/dejaHuecos.h) entra, con su objetivo de facilitar el entendimiento de estos huecos, valorar la importancia de un destructor bien diseñado, y comprender el funcionamiento de la memoria dinámica.