**Memoria**

Se divide en dos partes (que vamos a utilizar):

Stack/Pila

Heap/Montón

Las variables pueden almacenarse en cualquiera de las dos.

Diagrama

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Cuando ejecuto el main() se crean espacios en la memoria STACK para almacenar las variables que se definen.

Para las funciones que se llamen, se crean un espacio que luego se elimina cuando se termina de ejecutar la función. Este espacio, se crean subáreas que van a almacenar todas las variables temporales que use la función. Finalmente, el resultado de la función se va a almacenar en el STACK.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



El ciclo de vida de una variable depende de cómo se ejecute el programa.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Le indicas al sistema operativo que necesitas un espacio de memoria y de qué tamaño.



punteros



**Puntero**

Es una dirección de memoria. Todos los espacios de memoria que se asignan a las variables tienen una dirección de memoria que nos ayuda a identificar donde he almacenado la variable.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Separa 2 espacios de memoria del tamaño de INT.

Podemos **eliminar los espacios de memoria**.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Cuando creamos un espacio de memoria, tenemos que eliminarlo manualmente ya que, aunque no lo usemos va a seguir ocupando espacio y el espacio en un sistema operativo es limitado.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



La asignación o liberación de memoria en el HEAP es **manual**.

Las variables almacenadas en el **heap** son más lentas y ocupan un mayor costo direccional.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

