

# TDA - HASH

[7541/9515] Algoritmos y Programación II  
Segundo cuatrimestre de 2021

Alumno:	Agustín de la Rosa
Número de padrón:	102875
Email:	adelarosa@fi.uba.ar

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Detalles de implementación</b>	<b>2</b>
2.1. Función HASH . . . . .	2
2.2. Hash insertar . . . . .	2
2.3. Hash quitar . . . . .	2
<b>3. Diagramas</b>	<b>3</b>

## 1. Introducción

Se pide implementar una Tabla de Hash abierta (direccionamiento cerrado) en C. Para ello se brindan las firmas de las funciones públicas a implementar y se deja a criterio del alumno la creación de las funciones privadas del TDA para el correcto funcionamiento de la Tabla de Hash.

## 2. Detalles de implementación

Para compilar el programa, la carpeta en el .zip se encarga de tener los archivos y vínculos necesarios para en la shell de linux usar comando makefile, este compilara tanto el header como las pruebas.

### 2.1. Función HASH

La función principal hash recibe una cadena (string) y devuelve un número, depende de la longitud de la cadena ya que se suma cada valor del char(ASCII) y devuelve la suma de este. Luego es importante tener en cuenta, que todas las funciones que la utilicen, usan el resto del cociente entre el número hash y el máximo de la tabla pasado por usuario previamente, para poder obtener una posición a la tabla.

### 2.2. Hash insertar

Una vez obtenida la posición a partir de la función Hash, en mi vector de punteros a casillas, voy acomodando la tabla. Lo importante de recalcar aquí es que implemente nodos simplemente enlazados para almacenar. Por lo tanto es una tabla de nodos enlazados.

También guardo la clave en caso de que el usuario tenga una mala utilización de la memoria.

### 2.3. Hash quitar

La función de aquí separa en dos situaciones, donde en la tabla la posición es nula, o sea es el primero en la lista, o en el segundo caso, hay al menos 2 elementos en la tabla con la misma posición. En el segundo caso utilizo de manera recursiva para revisar los nodos, hasta encontrarlo. No quise usar un recorrido post orden "porque si voy al final de la lista tendría que recorrer todos los elementos apilados en el stack. Entonces voy verificando apenas voy avanzando.

### 3. Diagramas

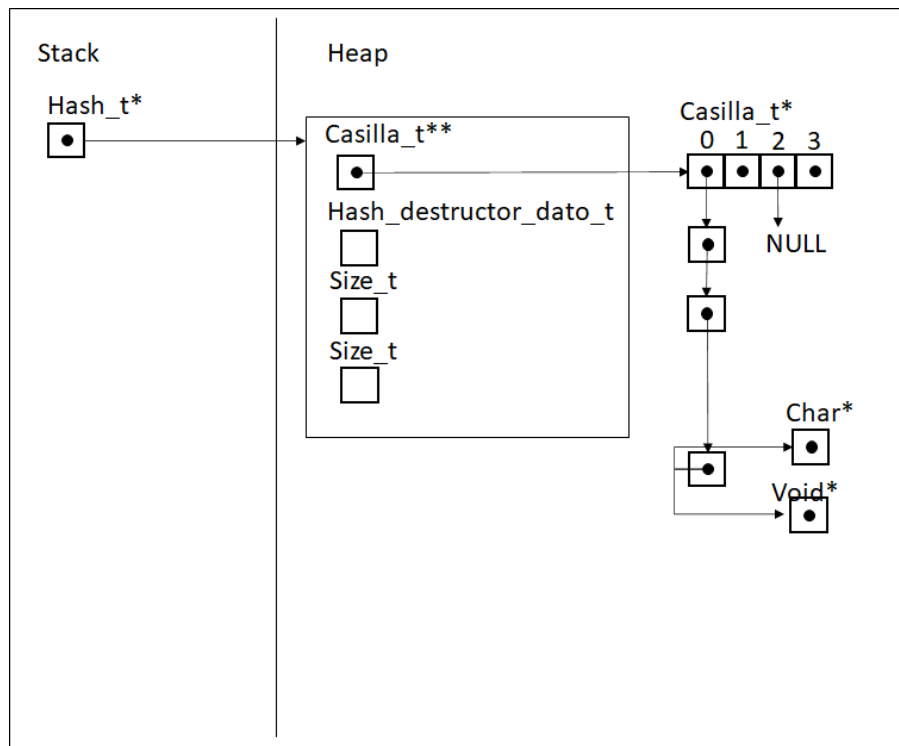


Figura 1: diagrama del hash.