

# Tabla de Hash

El trabajo que deben entregar de **forma grupal** es el tipo abstracto de datos Tabla de Hash, que se puede implementar mediante un hash abierto o cerrado, a elección.

## Primitivas del hash

```
typedef struct hash hash_t;
typedef struct hash_iter hash_iter_t;
typedef void (*hash_destruir_dato_t)(void *);

hash_t *hash_crear(hash_destruir_dato_t destruir_dato);
bool hash_guardar(hash_t *hash, const char *clave, void *dato);
void *hash_borrar(hash_t *hash, const char *clave);
void *hash_obtener(const hash_t *hash, const char *clave);
bool hash_pertenece(const hash_t *hash, const char *clave);
size_t hash_cantidad(const hash_t *hash);
void hash_destruir(hash_t *hash);
```

## Primitivas del iterador

```
hash_iter_t *hash_iter_crear(const hash_t *hash);
bool hash_iter_avanzar(hash_iter_t *iter);
const char *hash_iter_ver_actual(const hash_iter_t *iter);
bool hash_iter_al_final(const hash_iter_t *iter);
void hash_iter_destruir(hash_iter_t* iter);
```

El iterador debe permitir recorrer todos los elementos almacenados en el hash, sin importar el orden en el que son devueltos.

Se adjunta, además, un [archivo de pruebas](#) que pueden utilizar para verificar que la estructura funciona correctamente. El corrector automático, igualmente, añade más pruebas. Para esta entrega no deben entregarse pruebas propias.

Al igual que para las entregas anteriores, deberán entregar el código en papel, con el nombre y padrón de los integrantes del grupo, imprimiendo los archivos `hash.h` y `hash.c`.

Además, deben subir el código a la [página de entregas de la materia](#), con el código completo.

## Registro de tiempos

Quienes quieran mejorar sus tiempos pueden correr [este script](#), que es similar al que usa el corrector automático. Imprime por salida estándar el registro de tiempos y consumo para ciertas cantidades de inserciones:

```
12500 elementos - Tardó: 0.07s - Consumió: 3332 KiB
25000 elementos - Tardó: 0.10s - Consumió: 5208 KiB
50000 elementos - Tardó: 0.22s - Consumió: 9124 KiB
100000 elementos - Tardó: 0.31s - Consumió: 16728 KiB
200000 elementos - Tardó: 0.73s - Consumió: 32268 KiB
400000 elementos - Tardó: 1.29s - Consumió: 63700 KiB
```

Simplemente tienen que descargarlo, darse permisos de ejecución y correrlo indicándole por parámetro el nombre del ejecutable:

```
chmod +x tiempos_volumen.sh
./tiempos_volumen.sh ./main
```

No olviden revisar las [preguntas frecuentes del hash](#)

## Bibliografía recomendada

- Weiss, Mark Allen, “Data Structures and Algorithm Analysis”: *Chapter 5: Hashing*.
- Cormen, Thomas H. “Introduction to Algorithms”: *11. Hash Tables*.