

---

**DATOS PERSONALES:****ALUMNOS:**

Eliana Harriet: 107205

Tomas Emanuel: 108026

**AYUDANTE:**Jorge Collinet

---

Para el funcionamiento de [AlgoGram](#) se usaron distintos TDAs y structs creados específicamente para este TP.

Funcionamiento de los comandos y estructuras utilizadas para ello:

**Login:**

Utiliza los atributos *logged* y *usuarios* del struct *algogram*, *logged* es un struct *usuario* y *usuarios* es el TDA hash. El primero es usado para saber si ya hay o no un usuario logueado y el segundo es para saber si el usuario en el que se quiere loguear se encuentra en el archivo <usuarios.txt>.

**Logout:**

Utiliza únicamente el atributo *logged* mencionado anteriormente a fin de saber si hay o no un usuario logueado en ese momento.

**Publicar:**

Utiliza *logged* en primer lugar para saber si hay un usuario logueado o no, *posts*, el cual es un hash (atributo de *algogram*), donde se guardan todos los posts publicados de la forma clave: id, valor: struct *post*, y así poder acceder por su ID. Finalmente, usa el hash *usuarios* (mencionado para el comando login) para poder agregar el post al feed de cada usuario (excepto el propio del usuario que postea).

El feed de cada usuario está hecho en base a un heap, en el cual las prioridades son las distancias del usuario que postea al usuario del feed. Dicho heap se guarda en el struct *usuario* que se crea por cada usuario de la red social.

Se utiliza el struct *post* para guardar todos los datos del post (ID, likes, usuario que lo postea, texto).

**Ver próximo post:**

Al igual que los comandos anteriores, utiliza *logged* para saber si hay o no un usuario logueado. Además, utiliza el heap de posts de dicho usuario (mencionado anteriormente en el comando publicar) para saber si hay más posts para ver y en caso de haber saber cuál es el siguiente que corresponde mostrar en el feed. También se utilizan atributos del struct *post* a fin de saber quién fue que lo posteó, el ID y la cantidad de likes.

**Dar like:**

Se utiliza el atributo *logged* del struct *algogram* para saber si hay algún usuario logueado. Si es que lo hay, se utiliza el hash *posts* (atributo de *algogram* mencionado en publicar) para encontrar el post en base al ID pasado. En el post hallado, se actualiza la cantidad de likes y para guardar que fue este usuario se usa el TDA ABB para poder tener en orden alfabético a la hora de mostrar todos los usuarios que le dieron like a dicho post.

**Mostrar likes:**

A partir de un ID administrado por el usuario, se lo halla en el TDA hash *posts* para saber si en verdad existe dicho post. La cantidad de likes del post se averigua a través del atributo *cant\_Likes* del struct *posts*, y los usuarios que le dieron like se obtienen alfabéticamente al hacer un recorrido in-order del ABB mencionado anteriormente.