



# Tarea Enero – Marzo 2023

## Twitter

### 1 Introducción

La compra de Elon Musk de Twitter ha generado un clamor para el desarrollo de una alternativa. Se le pide que desarrolle el back-end de una de estas alternativas con un sencillo programa de consola para probarlo. A este fin, debe asegurarse de modularizar bien el código, manteniendo separado el código que puede ser usado en el producto final, del que es sólo para el programa de prueba.

### 2 Requerimientos del programa

Debe crear un programa en ANSI C que presente una consola virtual. La consola debe iniciar presentando el prompt

DON'T MISS WHAT'S HAPPENING! LOGIN, SIGNUP OR LEAVE

Si el usuario escribe "login", se debe presentar el prompt

USERNAME:

Seguido del prompt

PASSWORD:

Tras una entrada válida, se debe mostrar el *timeline* del usuario, con los *tweets* de los usuarios a los que sigue en orden cronológico. Al final, se debe presentar el prompt

WHAT'S HAPPENING?

El cual debe aceptar los siguientes comandos

+ indicando que el texto a continuación es un tweet de este usuario,  
@ indicando que el texto a continuación es un nombre de usuario cuyo perfil se desea ver  
logout el cual debe regresar al usuario al prompt inicial

Si el usuario está en el perfil de un usuario, se acepta un comando adicional: "follow" o seguir

Si el usuario escribe "signup", se debe igualmente presentar los prompts "username" y "password", pero en vez de validarlos, se deben almacenar y luego volver al prompt inicial

Finalmente, si el usuario escribe "leave", se debe cerrar el programa

## 2.1 Almacenamiento de Datos

El programa debe almacenar los usuarios en una tabla de hash que permita el fácil acceso a cada uno. Cada usuario debe ser un struct que contenga su nombre de usuario (username), el hash de su contraseña (password), un apuntador a su lista de tweets – la cual debe estar en orden cronológico – y un apuntador a su lista de “siguiendos”

La función de hash aplicada a la contraseña debe ser la misma usada para encontrar el *bucket* indicado en la tabla de hash. La función de hash más sencilla consiste en tomar el módulo. Bajo ninguna circunstancia debe almacenarse la contraseña en sí.

La lista de tweets debe ser una lista enlazada. La lista de “siguiendos” puede ser cualquier estructura, aunque se recomienda una lista enlazada.

## 2.2 Funcionamiento del Programa

Cuando se agrega un tweet con el comando “+”, se debe almacenar en la lista de tweets del usuario. Se debe solicitar el espacio del tweet con malloc y obtener la fecha y hora con time(). El tweet sólo debe aceptar los primeros 280 caracteres, cualquier carácter adicional debe ser consumido y descartado.

Al ver el perfil de un usuario con “@”, se deben imprimir todos sus *tweets*, en orden cronológico, del más antiguo al más reciente. Para ello, se debe acceder al struct de ese usuario en la tabla de hash y recorrer su lista de tweets.

El comando “seguir” agrega el usuario siendo visualizado a la lista de “siguiendos” del usuario que ha hecho login. Aunque se puede almacenar solamente el nombre de usuario, se recomienda guardar un apuntador al struct del usuario

Agregar un usuario requiere la generación del nodo en la tabla de hash y el cómputo de su valor de hash y el hash de su contraseña. Debe asegurarse de que el acceso a esta tabla funcione correctamente.

El timeline se construye recorriendo la lista de “siguiendos” del más antiguo al más reciente. Aunque se puede iterar entre los siguiendos buscando siempre el que tenga el siguiente tweet sin imprimir, se recomienda construir una lista enlazada y dejar que la función de inserción ordene los tweets a medida que se van agregando los tweets de los siguiendos a lo que se va a imprimir, ya que de lo contrario se deben almacenar tantos apuntadores a elementos de lista como haya siguiendos.

Finalmente, para iniciar sesión se debe buscar en la tabla de hash el nodo correspondiente al nombre de usuario. Luego se debe calcular el hash de la contraseña introducida para comparar hash contra hash. Si coinciden, se muestra el *timeline*.

## 3 Requerimientos de la entrega

Debe entregar un archivo comprimido que contenga

- Todos sus archivos .c con sus respectivos .h
- Un Makefile para compilar su programa

El programa debe compilar sin errores al llamar al comando make. No incluya archivos .o ni archivos ejecutables en su entrega.

Debe documentar su código, incluyendo una breve definición al inicio de cada archivo, función y macro que justifique qué es e indique para qué es cada parámetro; y para cualquier variable cuyo nombre no sea intuitivo.

El proyecto debe ser subido al Moodle de la materia en la asignación marcada como “📁 Tarea” en la sección de “Semana 2” Sólo deberá efectuar una entrega por grupo.

## 4 Evaluación

La tarea tiene una ponderación de 10 puntos. Se asignarán

- 5 puntos por código
  - 1 punto por su tabla de hash
  - 1 punto por su lista enlazada
  - 1 punto por su manejo de los tweets
  - 1 punto por su manejo de los usuarios
  - 1 punto por su programa principal
- 5 puntos por ejecución
  - 1 punto por *signup* y *login*
  - 1 punto por el *timeline*
  - 1 punto por agregar correctamente los tweets
  - 1 punto por mostrar correctamente el perfil de los usuarios
  - 1 punto por permitir seguir a otros usuarios

El programa debe correr sin errores y seguir todas las prácticas indicadas en el laboratorio para la programación de C en Unix.

### 4.1 Puntos Extra

Se asignarán puntos extra por

- Permitir agregar una descripción al perfil cuando se crea el usuario
- Mostrar la lista de seguidores al llegar a un perfil
- Aceptar el comando “mentions” el cual busca, en todos los tweets, aquellos en donde aparezca el nombre de usuario del usuario actual
- Permitir, en el makefile, la opción “make backend”, el cual debe compilar solo los archivos no relacionados con la pantalla que presenta el programa principal