



# UNAH

UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE  
INGENIERÍA EN SISTEMAS

**Desarrollo Web**  
Examen Práctico Unidad I

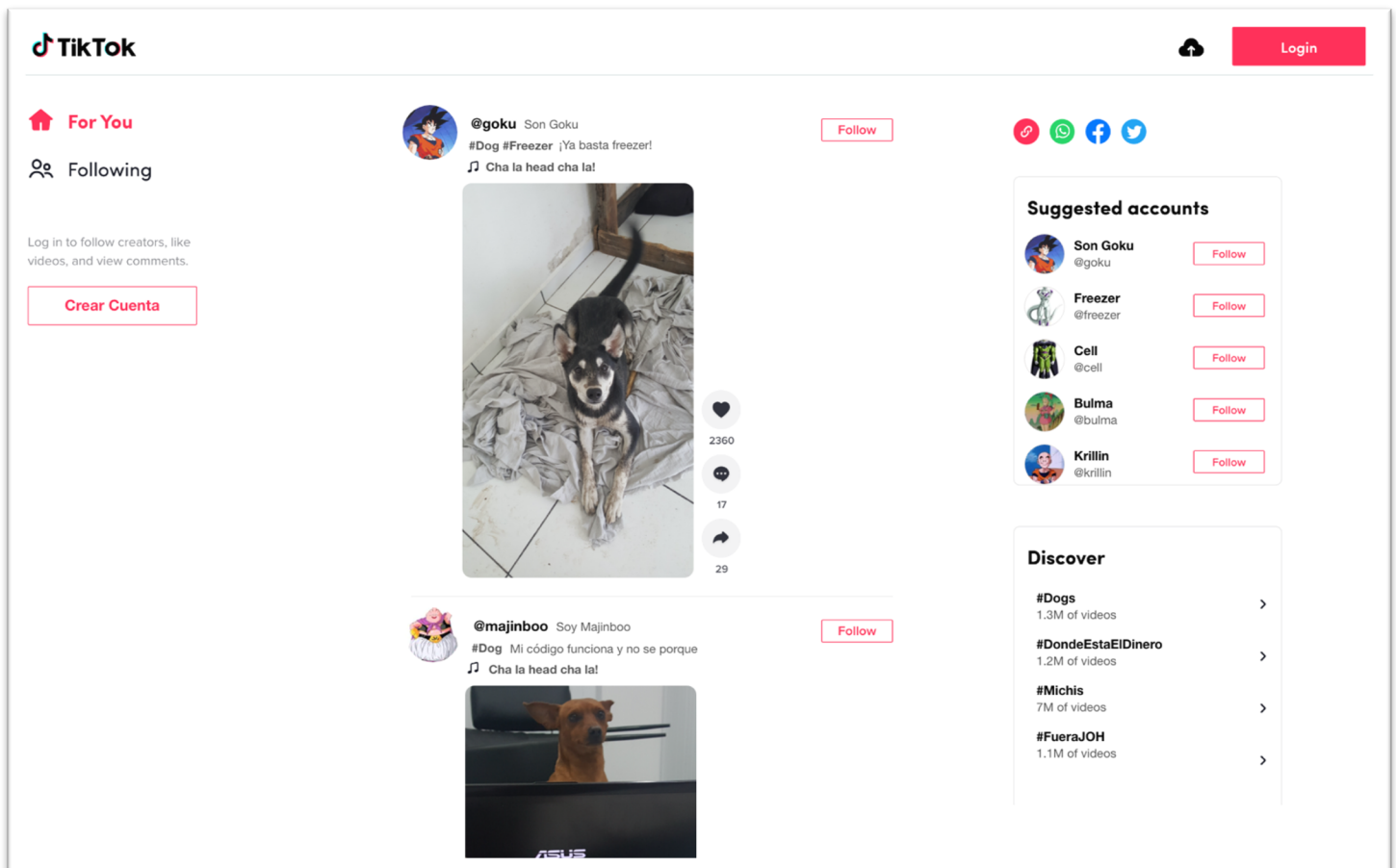
**Catedrático: Erick Marín**

La presente evaluación es de carácter individual, evite cualquier tipo de plagio o será penalizado.

El objetivo de esta evaluación es aplicar conceptos de frontend con Angular y backend con NodeJS/MongoDB tomando como referencia la aplicación TikTok, la cual se utilizó como tema de evaluación de la primera unidad.

### Se le provee

Frontend HTML, CSS, Bootstrap y JS: Se provee una funcionalidad básica de la evaluación de la unidad anterior. **Se toma como base, el proyecto solución de un estudiante de la clase en la unidad anterior.** El proyecto es completamente funcional con LocalStorage, el objetivo principal es migrar toda la funcionalidad hacia el backend en NodeJS y MongoDB. No debe quedar ninguna referencia o funcionalidad con LocalStorage.



En base a los recursos proveídos desarrollar lo siguiente:

- **Crear proyecto utilizando Angular y NgBootstrap:** Como mínimo crear tres componentes además del componente app, uno para la lista de tiktoks, uno para la lista de usuarios y otro para la lista de hashtags. Se le provee un frontend html/css, este deberá migrarlo a Angular.
- **Crear base de datos en MongoDB:** Crear una base de datos con el nombre **tiktok** y las colecciones; **usuarios**, **tiktoks** y **hashtags**. La estructura de los documentos a guardar se describe en el apartado posterior.



- **Colección usuarios**

```
{
  _id: "1",
  usuario: "goku",
  password: "asd.456",
  nombre: "Son Goku",
  seguidores: [
    { _id: "5", usuario: "krillin" },
    { _id: "6", usuario: "bulma" },
    { _id: "7", usuario: "vegeta" },
  ],
  siguiendo: [
    { _id: "8", usuario: "dende" },
    { _id: "9", usuario: "gohan" },
    { _id: "10", usuario: "goten" },
  ],
};
```

Esta colección servirá para identificar el usuario que publicará tiktoks, para ello el usuario deberá hacer login usando el formulario diseñado. Para gestionar la sesión del usuario logueado puede utilizar sesiones con NodeJS o alguna alternativa que haya utilizado en su proyecto de clase.

Agregar la funcionalidad al formulario de crear usuarios.

- **Colección de tiktoks**

```
{
  _id: "1",
  titulo: "¡Ya basta freezer!",
  fecha: "12/12/2012",
  video: "videos/1.mp4",
  tituloCancion: "Cha la head cha la!",
  likes: 10,
  shares: 15,
  comentarios: [
    { _id: "1", usuario: "krillin", comentario: "Gokuuuuuuuuu!!!!!" },
    { _id: "2", usuario: "vegeta", comentario: "Insecto!!!!!" },
  ],
  hashtags: [
    { _id: "1", hashtag: "Dogs" },
    { _id: "2", hashtag: "Freezer" },
    { _id: "3", hashtag: "DragonBall" },
  ],
};
```

Dar la funcionalidad al formulario de agregar/subir video o tiktok, al momento de crear un elemento nuevo los likes y shares deben ser 0, los comentarios y hastags deberán ser arreglos vacíos.

Estos se llenarán dependiendo de las acciones del usuario. (Nota: No debe realizar la funcionalidad real de subida de archivos al servidor, bastaría con seleccionarlo de una lista de elementos previamente alojados en el backend)

Se debe renderizar la sección principal de tiktoks tal y como lo mostrado en la primera imagen. Dicha lista de publicaciones tendrá comportamientos diferentes;

- Si un usuario no ha hecho login se mostrarán todos los tiktoks
- Si un usuario hizo login se mostrarán los tiktoks de los usuarios a los que sigue



- Si el usuario da click a un hashtag de la sección lateral inferior derecha entonces se mostrarán todos los tiktoks con dicho hashtag

Al momento de dar click en el botón de comentarios deberá mostrarse una ventana modal donde muestre todos los comentarios del tiktok y un textarea para guardar un nuevo comentario usando el usuario autenticado.

- **Colección hashtags**

```
{
  _id: '1',
  hashtag: 'Dogs',
  videos: 1000000000
}
```

Al momento de realizar una nueva publicación se debe identificar en los hashtags si existe o no, en caso de no existir se debe agregar un nuevo hashtag con la cantidad de videos en 0. Se debe renderizar la lista de hashtags.

Cada vez que el usuario de click en una de las opciones se deben filtrar la lista de tiktoks que cumplan con el hashtag.

**Otras funcionalidades:**

- **Seguir:** Dar funcionalidad a la opción de Follow para seguir a un usuario.

**Observaciones:**

- Cree solo los servicios web del API Rest a utilizar, desarrolle los que sean necesarios para la solución.
- Estructure el proyecto en carpetas independientes para frontend y backend.
- Agregue unos cuantos documentos de prueba a la base de datos. (pocos).
- Genere los ids de las colecciones utilizando la función de MongoDB.
- No modifique el nombre de la base de datos, las colecciones y los nombres de los atributos de los objetos JSON ya que puede ocasionar malentendidos al momento de revisar su evaluación.
- Puede utilizar sus apuntes, google y cualquier recurso, escriba su código y evite copiar y pegar (eso no es programar).
- Al momento de subir el proyecto solución, no incluya la carpeta node\_modules. Incluya un pequeño script básico para la base de datos.
- Puede hacer adecuaciones pequeñas a los JSONs indicados en caso de encontrar alguna falencia que impida finalizar el examen.

**Recomendaciones:**

- Para cada acción verifique la estructura del json a utilizar y sus diferentes atributos.
- Enfóquese en su evaluación, aproveche el tiempo y no confunda a sus compañeros.
- Evite copiar código compartido en chats grupales, se penalizará exámenes donde se compruebe copia o plagio (ejem. Códigos similares con solo cambios de nombres de variables, funciones, orden del código, entre otros).

*"El primero y peor de todos los fraudes es engañarse a si mismo." . \_PHILIP JAMES BAILEY*  
¡Éxito!