

PROYECTO 2

Introducción a la Programación y Computación 1

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

SECCIÓN: -

CATEDRÁTICO: -

TUTOR ACADÉMICO: -

USocial



PROYECTO 2

Objetivos

Generales

- El estudiante aplique los conocimientos adquiridos en el curso de Introducción a la Programación y Computación 1 en el desarrollo de aplicaciones web.

Específicos

- Familiarizar al estudiante con el concepto de la programación web.
- Lograr que el estudiante comprenda el concepto de backend y frontend empleando una arquitectura cliente – servidor.
- Adentrar al estudiante a la programación web por medio de APIs
- Que el estudiante se relacione con el uso de NodeJS
- El estudiante se familiarice con librerías y frameworks para el desarrollo de aplicaciones frontend

PROYECTO 2

Descripción General

Definición:

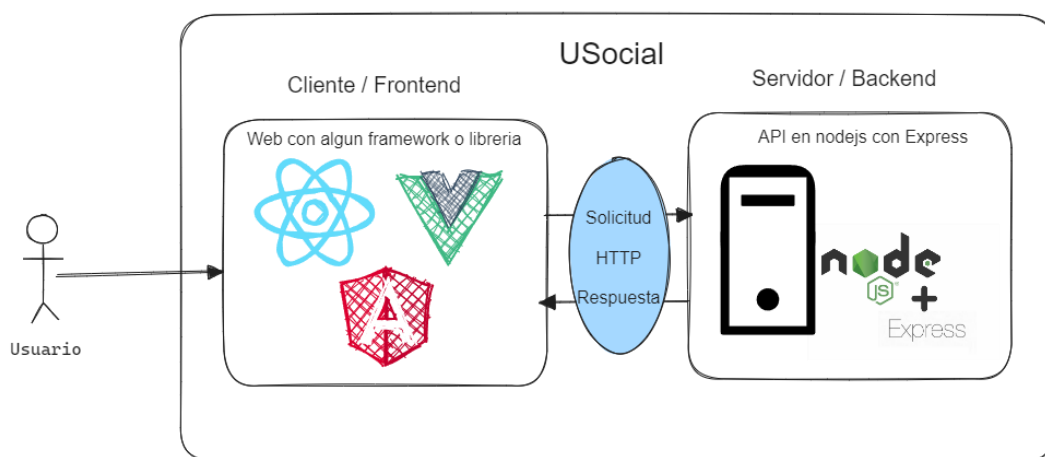
Se desea crear un espacio de tipo blog y foro llamado *USocial* donde todos los estudiantes o catedráticos de la Universidad de San Carlos de Guatemala puedan publicar sus opiniones, reportes y/o anuncios de incidentes que pasen en la universidad, compartiendo imágenes, links o únicamente texto. Estas publicaciones pueden tener interacciones con comentarios, armar hilos de conversación y votar por las publicaciones para que sean “*Trending Topic*” o sean de mayor importancia para que los usuarios miren las publicaciones que tengan más relevancia.

La aplicación contará con las siguientes funcionalidades:

- Ingreso de datos.
- Modificación de datos.
- Visualización de datos.
- Eliminación de datos.

El sistema estará conformado por dos partes: frontend y backend:

- La parte de backend debe ser una API desarrollada en NodeJS haciendo uso de la librería de Express js.
- La parte de frontend debe ser una aplicación web, haciendo el uso de la librería React o con algún framework como Angular o Vue además de las herramientas que considere necesarias.



PROYECTO 2

Flujo de la aplicación

***Se utilizará el framework o librería que el auxiliar enseñe para frontend**

1. El usuario entra a la página web ingresando la ruta local de la aplicación frontend.
2. El usuario realiza acciones como ver, agregar, modificar y eliminar datos.
3. Por cada acción se realiza una solicitud HTTP a la API que a su vez retornará los datos pertinentes en su respuesta HTTP.
4. La API en NodeJS es la que almacenará todos los datos.

Página Principal

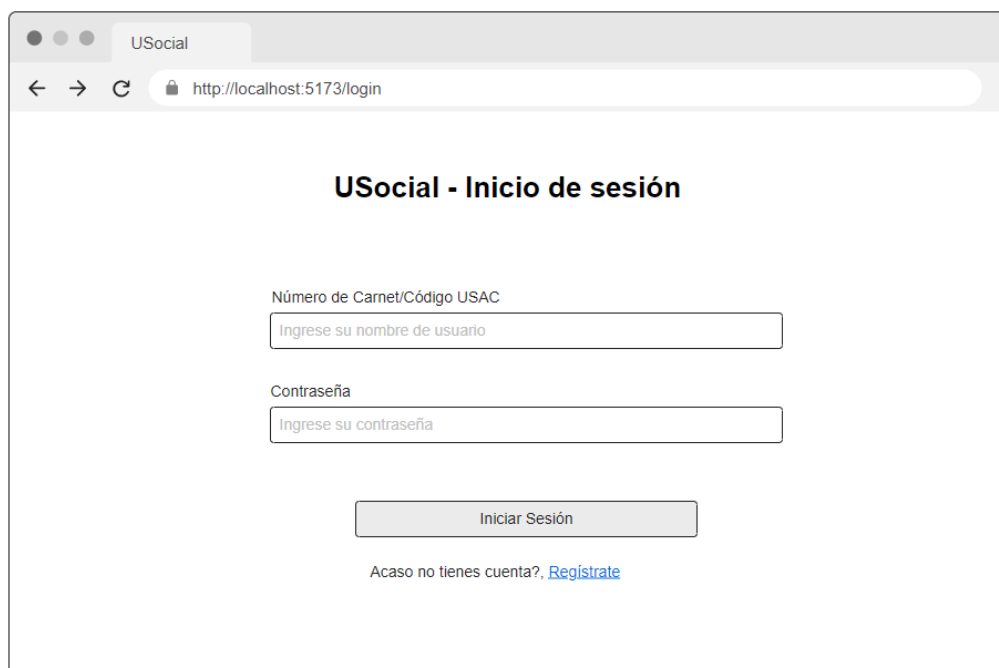
Deberá diseñar una página de inicio de la aplicación, de esta página se partirá para el inicio de sesión o registro de usuario. Deberá contener la siguiente información de manera creativa:

- Nombre de la aplicación
- Logo
- Descripción de a lo que se dedica su página web
- Acerca de
- Contacto de soporte
- Información del estudiante que desarrolló la página

Autenticación al sistema

Inicio de Sesión

Para ingresar al sistema el usuario debe iniciar sesión dentro de la aplicación, ingresando su número de carnet y su contraseña. Si no tiene cuenta, puede registrar su usuario. El usuario debe estar previamente registrado para iniciar sesión. En dado caso haya un error de autenticación, se debe mostrar un mensaje. Y desde esta página se debe tener un enlace que si no tiene cuenta tiene la opción de registrarse.



The screenshot shows a web browser window with the title 'USocial'. The address bar displays 'http://localhost:5173/login'. The main content area has the heading 'USocial - Inicio de sesión'. Below the heading, there are two input fields: the first is labeled 'Número de Carnet/Código USAC' with a placeholder 'Ingrese su nombre de usuario', and the second is labeled 'Contraseña' with a placeholder 'Ingrese su contraseña'. Below these fields is a button labeled 'Iniciar Sesión'. At the bottom, there is a link that says 'Acaso no tienes cuenta?, [Regístrate](#)'.

PROYECTO 2

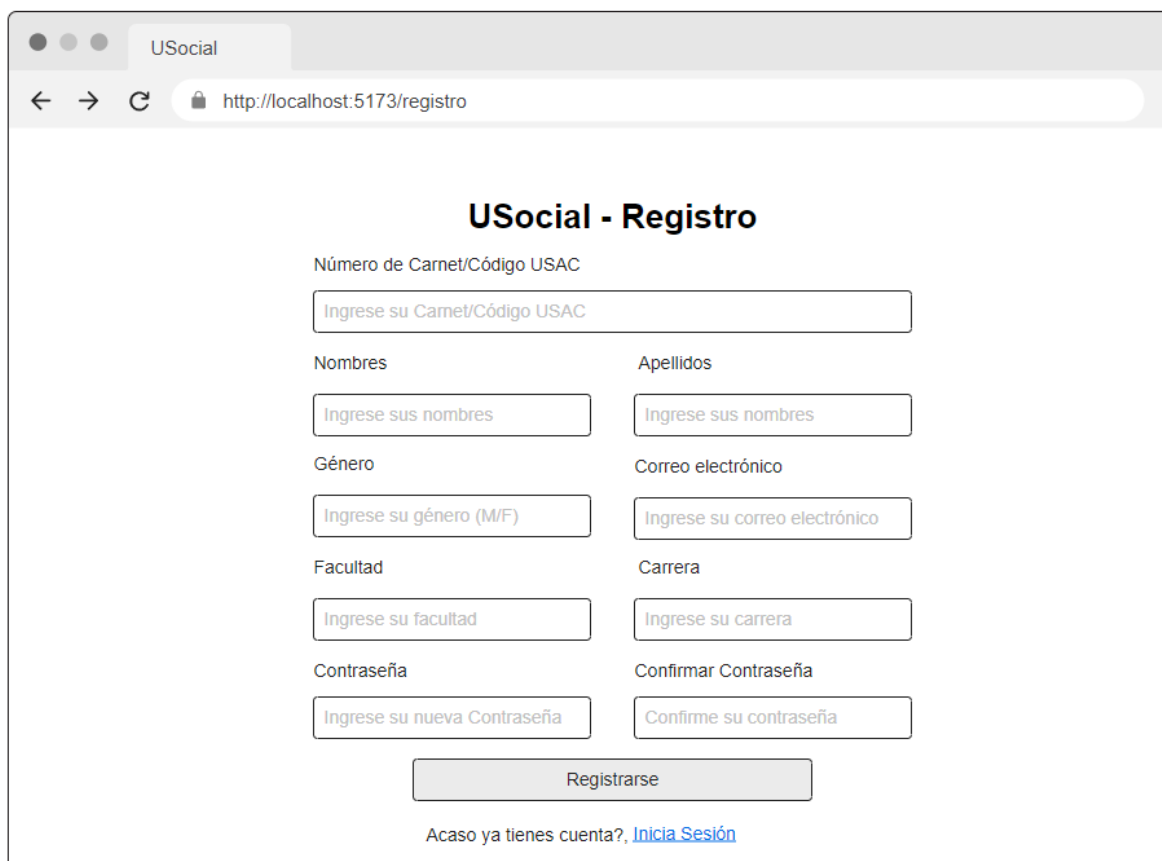
Registro

Los usuarios pueden registrarse usando el código único proporcionado por la universidad, tiene que tomar en cuenta las siguientes consideraciones:

- El código o carnet debe ser único en toda la aplicación.
- La contraseña debe tener al menos 8 caracteres, entre estos caracteres deberá contar con al menos 1 mayúscula, 1 minúscula y 1 carácter especial.
- Debe tener algún botón para redirigir al inicio de sesión en dado caso el usuario tenga una cuenta.
- Cualquier error de registro, se tiene que mostrar un aviso desde el cliente.

Los datos que se le deben de solicitar a un nuevo usuario al registrarse son:

- Código USAC/ Carnet
- Nombres
- Apellidos
- Género
- Facultad
- Carrera
- Correo electrónico
- Contraseña



The screenshot shows a web browser window with the title 'USocial' and the URL 'http://localhost:5173/registro'. The page content is titled 'USocial - Registro'. It contains a registration form with the following fields and labels:

- Número de Carnet/Código USAC: Ingrese su Carnet/Código USAC
- Nombres: Ingrese sus nombres
- Apellidos: Ingrese sus nombres
- Género: Ingrese su género (M/F)
- Correo electrónico: Ingrese su correo electrónico
- Facultad: Ingrese su facultad
- Carrera: Ingrese su carrera
- Contraseña: Ingrese su nueva Contraseña
- Confirmar Contraseña: Confirme su contraseña

Below the form fields is a 'Registrarse' button. At the bottom, there is a link: 'Acaso ya tienes cuenta?, [Inicia Sesión](#)'.

(Ejemplo)

El sistema estará estructurado con 2 módulos diferentes: Administrador y Usuario. **Se debe redirigir al módulo correspondiente según el tipo de usuario que inicie sesión.**

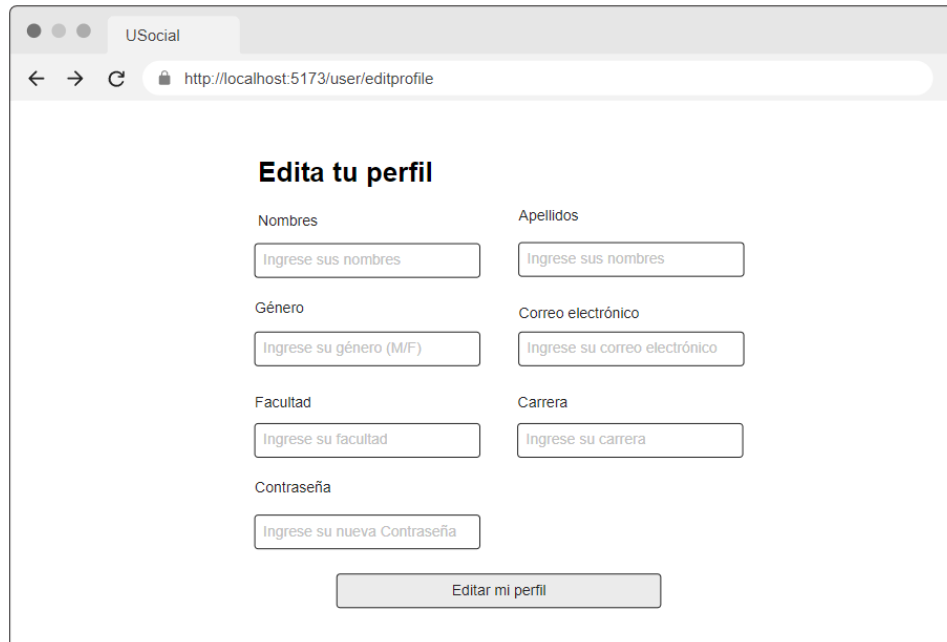
PROYECTO 2

Módulo de Usuarios

Editar Perfil

El usuario será capaz de modificar todos sus datos.

Se deberá colocar un botón o apartado que redirija a esta página de modificación y se mostrará siempre y cuando se haya iniciado sesión.



The screenshot shows a web browser window titled 'USocial' with the address bar displaying 'http://localhost:5173/user/editprofile'. The main content area is titled 'Edita tu perfil' and contains several input fields for user information. The fields are arranged in two columns: 'Nombres' and 'Apellidos' at the top, followed by 'Género' and 'Correo electrónico', then 'Facultad' and 'Carrera', and finally 'Contraseña' at the bottom. Each field has a placeholder text indicating what to enter. At the bottom center, there is a button labeled 'Editar mi perfil'.

El carnet/código USAC no puede ser modificado por el usuario ya que es su id único para la aplicación.

Publicar Post

El usuario puede en cualquier momento realizar un post dentro de USocial. El usuario puede adjuntar lo siguiente en su publicación:

- Descripción del post (Obligatorio)
- Categoría (Anuncio Importante/Divertido/Académico/Variedad) (Obligatorio)
- Imagen (Opcional, para esto puede enviarlo en base64)
- Opción de publicarlo anónimo o no.

Y el servidor es capaz de almacenar lo siguiente del post:

- Descripción del post (Obligatorio)
- Código del usuario quien lo publicó (Obligatorio)
- Categoría (Obligatorio)
- Imagen (Opcional)
- Anónimo (true/false) (Obligatorio)
- Fecha y hora de la publicación (Obligatorio)

PROYECTO 2

The screenshot shows a web browser window with the address bar displaying 'http://localhost:5173/user/createpost'. The page title is 'Nuevo Post'. Below the title, there is a text input field with the placeholder 'Describe lo que quieras escribir en tu post:' and 'Type here'. Below the text field, there is a 'Categoría' dropdown menu with 'Divertido' selected. Below the dropdown, there is a toggle switch for '¿Quieres publicarlo de forma anonima?' with 'Si' (Yes) selected. Below the toggle, there is a button 'Adjuntar Imagen' and a 'Vista Previa' (Preview) button. At the bottom, there are two buttons: 'Publicar' (Publish) and 'Cancelar' (Cancel).

Vista de Home (Todos los Posts)

Los posts los pueden ver todos los usuarios teniendo el orden, del mas reciente al ultimo posteado por los usuarios en la plataforma. Los usuarios pueden darle “Me gusta” y comentar en los posts. Además hay que mostrar en cada post:

- Nombre y apellidos
- Carrera y entre paréntesis la facultad a la que pertenecen
- Un tag de la categoría a la que pertenece cada post
- El texto del post
- La imagen (si es que tuviera)
- Fecha y hora de publicación
- Cantidad de likes y de comentarios

Para los posts anónimos lo que cambia es lo siguiente:

- Nombre y apellidos se sustituyen por “Usuario anónimo”
- Carrera y Facultad se sustituye por “Universidad de San Carlos de Guatemala”

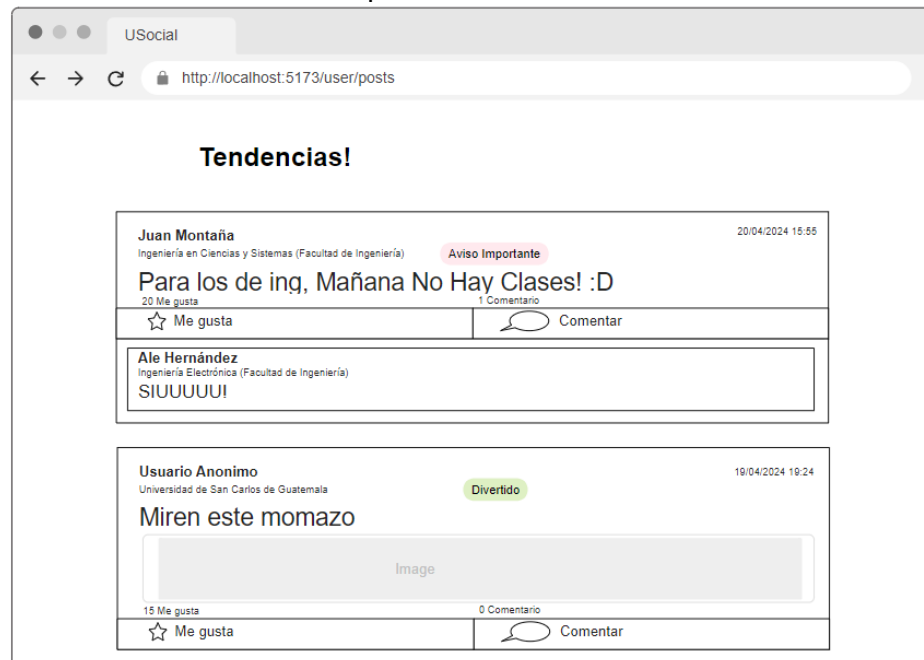
Además que no se puede comentar de forma anónima.

The screenshot shows a web browser window with the address bar displaying 'http://localhost:5173/user/posts'. The page title is 'Mira lo que los demás están compartiendo!'. Below the title, there are three post cards. The first card is by 'Juan Montaña' (Ingeniería en Ciencias y Sistemas (Facultad de Ingeniería)) with a 'Aviso Importante' tag and the text 'Para los de ing, Mañana No Hay Clases! :D'. It has a 'Me gusta' button and a 'Comentar' button. The second card is by 'Ale Hernández' (Ingeniería Electrónica (Facultad de Ingeniería)) with the text 'SIUUUUU!'. The third card is by 'Usuario Anonimo' (Universidad de San Carlos de Guatemala) with a 'Divertido' tag and the text 'Miren este momazo'. It has a placeholder for an image and 'Me gusta' and 'Comentar' buttons.

PROYECTO 2

Vista Tendencias

En este apartado se encontrará el top 10 de publicaciones con la suma más grande de entre “Me gusta” y cantidad de comentarios. Tienen que estar ordenados de forma descendente.



Comentarios

Se pueden hacer de 1 a muchos comentarios en 1 o muchas publicaciones, para este caso únicamente puede ingresar texto y con ello para mostrarlo únicamente muestra el nombre, apellido la carrera, entre paréntesis la facultad y el comentario de texto. Este comentario debe verse visible junto con la publicación relacionada.

PROYECTO 2

Módulo Administrador

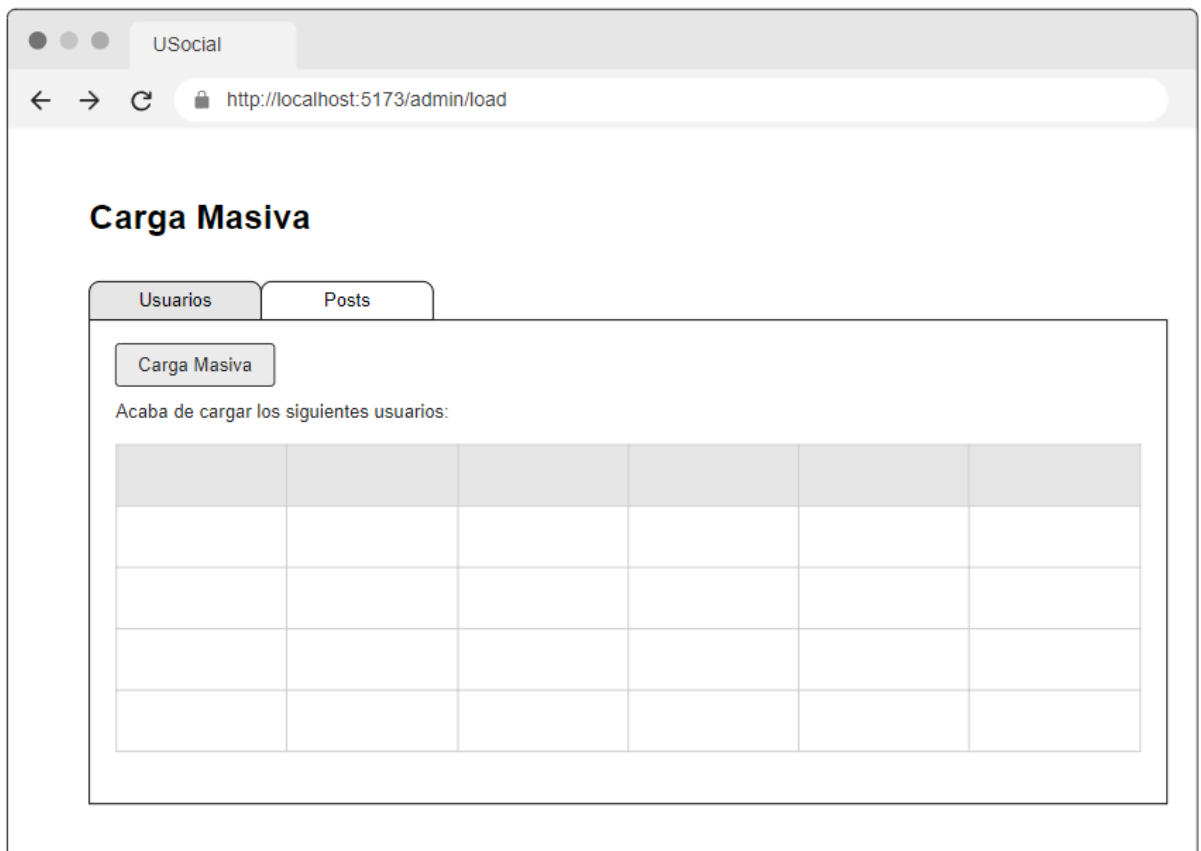
El administrador contará con las siguientes credenciales:

- **Nombre:** Nombre de su auxiliar
- **Apellido:** Apellido de su auxiliar
- **Código USAC:** 12024
- **Facultad:** Ingeniería
- **Carrera:** Ingeniería en Ciencias y Sistemas
- **Correo electrónico:** ipc11s2024@email.com
- **Contraseña:** @dminIPC1

Dentro de sus funcionalidades contamos con:

Carga Masiva

Se elegirá la opción de carga que desee realizar.



El administrador puede cargar un archivo de tipo JSON dependiendo de lo que desea cargar esta será la estructura:

```
[
  {
    "codigo": 2024000001,
    "nombres": "Alberto Juan",
    "apellidos": "Pérez López",
    "genero": "M",
    "facultad": "Arquitectura",
    "carrera": "Diseño Gráfico",
    "correo": "albertlop@email.com",
    "contrasenia": "Albert123-"
  },
  ...
]
```

PROYECTO 2

Donde al cargar se mostrará en esa tabla temporal los datos que cargó exceptuando la contraseña.

En al caso de las publicaciones tendrá el siguiente formato:

```
[
  {
    "codigo": 202400001,
    "descripcion": "Hoy es mi primer día, recomendaciones?",
    "categoria": "Variedad",
    "anonimo": false
  },
  ...
]
```

Y la fecha y hora de publicación serán la fecha y hora de carga del archivo JSON. Ninguno va a contener comentarios ni likes. También se cargarán a la tabla temporal mostrando el código, la descripción, la categoría y el booleano de si es anónimo o no.

Visualización de datos

El administrador es capaz de poder visualizar y eliminar los usuarios y las publicaciones en USocial. Las formas de visualizarlo son desde la página web o exportándolos como csv. Además es posible visualizar de forma individual cada usuario o publicación con toda la información.



Código/Carnet	Nombres	etc	Acciones
202300001	Juan Alberto		<button>Ver</button> <button>Eliminar</button>
202200001	Luis Pedro		<button>Ver</button> <button>Eliminar</button>
202100001	Marcos Alexander		<button>Ver</button> <button>Eliminar</button>
202000001	Byron Ramiro		<button>Ver</button> <button>Eliminar</button>



Código/Carnet	Descripción	etc	Acciones
202300001	Hoy es día de U		<button>Ver</button> <button>Eliminar</button>
202200001	Vamos a cañas		<button>Ver</button> <button>Eliminar</button>
202100001	Ya hay nota del parcial		<button>Ver</button> <button>Eliminar</button>
202000001	Sale en vacas		<button>Ver</button> <button>Eliminar</button>

PROYECTO 2

Reportes

- Top 5 de post con más likes. (Gráfica de pie)
- Cantidad de posts por categorías (Gráfica de pie)
- Top 10 usuarios con más publicaciones creadas (Gráfica de barras)

Cada reporte los podrá visualizar desde la página web, para las gráficas se le recomienda el uso de la librería de Chart.js.

PROYECTO 2

Requerimientos

Documentación:

- Manual técnico de la API en formato PDF (Backend)
 - Descripción general de la solución
 - Diagrama de clases
 - Tabla de endpoints donde se describa el método, el endpoint, la entrada (no aplica para los GET y DELETE) y salida del endpoint y una pequeña descripción.
- Manual de usuario en formato PDF (Frontend)
 - Descripción general del programa
 - Requerimientos mínimos del sistema
 - Descripción de los pasos para el uso de las opciones más importantes

Restricciones:

- La aplicación debe ser desarrollada en NodeJS y el framework a elección para el frontend.
- No se permite utilizar código copiado o bajado de Internet.
- Para ser tomada la entrega debe agregar al auxiliar a su repositorio.
- El editor de texto a utilizar queda a discreción del estudiante (se recomienda el uso de Visual Studio Code)
- Copias obtendrán una nota de 0 y reportará a la Escuela de Ciencias y Sistemas.

Consideraciones:

- Todas las imágenes mostradas en el enunciado tienen el fin de ser una guía, el estudiante debe utilizar su creatividad para el desarrollo de cada una de las páginas.
- Todas las funcionalidades deben realizarse en la parte del backend (api en NodeJS), no se debe realizar ninguna funcionalidad desde el frontend, desde la página solamente se permite realizar consultas a la API en NodeJS.

PROYECTO 2

HABILIDADES A EVALUAR

- Habilidad para analizar y sintetizar información
- Habilidad para resolver problemas.
- Interacción con objetos en entornos de API.
- Manejo de propiedades en componentes gráficos web.
- Uso de peticiones HTTP.
- Conocimiento de los conceptos de backend y frontend para el desarrollo web.

Entrega

- FECHA DE ENTREGA: 25/04/2024 antes de las 23:59 (No se aceptarán entregas a partir de esa hora) **NO HABRA PRORROGA.**
- Adjuntar todos los entregables solicitados dentro de un repositorio en GitHub con el nombre de IPC1_Proyecto2_carnet, por ejemplo:
IPC1_Proyecto2_202300000. Y debe contener lo siguiente:
 - Código fuente de la API en NodeJS
 - Código fuente del frontend desarrollado con el framework o librería.
 - Documentación en archivos PDF
- Subir link del repositorio en la tarea asignada en UEDi.
- Agregar de colaborador al repositorio al auxiliar correspondiente a su sección:
 - Sección A: rodrialeipc
 - Sección B: RMorales202010033
 - Sección C: DavidMaldo02
 - Sección D: Ennox2
 - Sección E: rodriporon
 - Sección F: Ayeser-Cristian
 - Sección G: fedestm