

Proyecto Sprint 1 Módulo 2: Entrenamiento Front-End

Objetivo: Construir una interfaz para mantenerse en contactos con familiares y amigos.

El aplicativo de cumplir con los siguientes requerimientos técnicos:

1. Implementar CSS para aplicar estilos y diseño responsive. El uso de [SASS como preprocesador CSS](#) duplica el porcentaje del ítem sobre la calificación del proyecto.
2. Implementar JSON-server que contenga la siguiente información:
 - a. Usuarios
 - b. Mensajes

Nota: Cada uno de estos ítems debe tener su propio endpoint. Asimismo, el JSON-server debe ser desplegado en un hosting como [railway.app](#).

3. Cada objeto usuario debe contener mínimo la siguiente información:
 - a. Id
 - b. Nombre
 - c. Número de celular
 - d. Password
 - e. Url de la imagen de perfil
 - f. Una propiedad “flag” que se modifique cuando el usuario está en línea o desconectado
 - g. Info o frase que identifique al usuario
 - h. Propiedad con la fecha y hora de la última vez que estuvo en línea (opcional).
4. Cada objeto mensajes debe tener las siguientes propiedades:
 - a. idUser1: id del usuario que inició la conversación.
 - b. idUser2: id del usuario que recibió el mensaje
 - c. un array de conversaciones donde cada elemento sea un objeto con la siguiente estructura:
 - i. una propiedad llamada sendBy que contenga el id del usuario que envió el mensaje
 - ii. una propiedad llamada date con la fecha en que fue enviado el mensaje
 - iii. una propiedad llamada hour con la hora en que fue enviado el mensaje
 - iv. una propiedad llamada message
 - v. una propiedad “flag” que se modifique cuando el usuario receptor haya visto el mensaje

5. El aplicativo debe consistir en tres páginas:

a. La primera página es un Sign in o inicio de sesión que debe:

- i. Contener un form con dos inputs: el primero con el atributo type="number" para iniciar sesión con el número celular y el segundo input con type="password"
- ii. Al dar click en el botón Sign in se debe validar los siguiente:
 1. Si hay campos vacíos debe mostrar una ventana emergente o alert que indique cual es el dato faltante.
 2. Si el número celular es incorrecto mostrar una ventana emergente o alert que muestre el mensaje "el número ingresado no existe".
 3. Si la contraseña es incorrecta mostrar una ventana emergente o alert que muestre el mensaje "la contraseña ingresada es incorrecta".
 4. Si el número celular y contraseña ingresada son correctas debe:
 - a. Mostrar una ventana emergente o alert que muestre el mensaje "Bienvenido nombre del usuario", y
 - b. Redireccionar al usuario a la página home.

b. La página home debe tener este diseño:

i. Desktop:

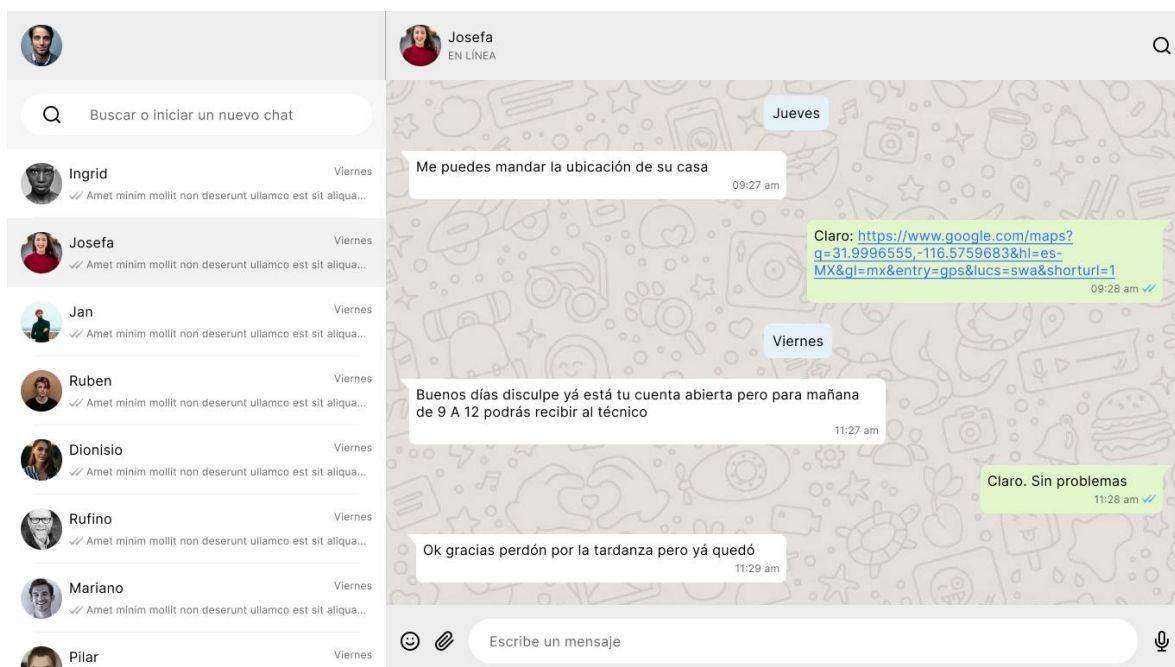


Imagen 1. <https://raw.githubusercontent.com/uxcrisopher/imagenes/main/Readmes/MensajesApp/Desktop.png>

ii. Mobile:

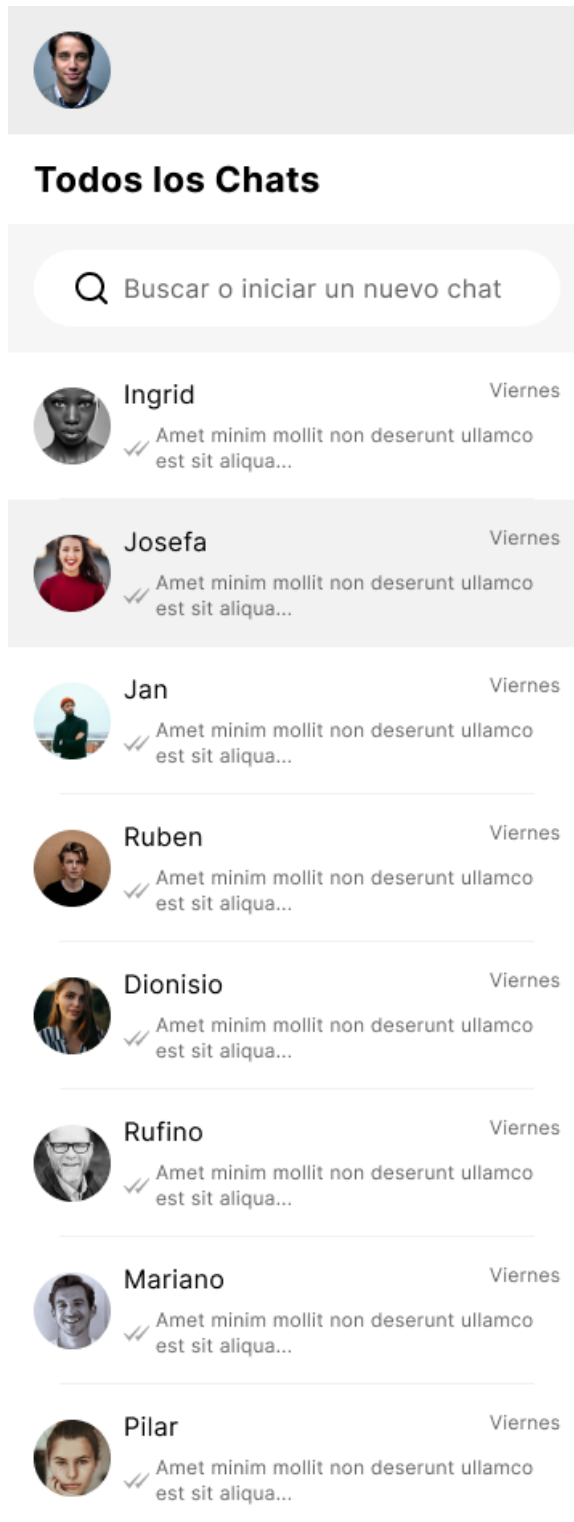


Imagen 2. <https://raw.githubusercontent.com/uxcristopher/imagenes/main/Readmes/MensajesApp/Mobile.png>

- c. Asimismo, en el home el usuario debe tener la oportunidad de:
 - i. Observar el listado de todas las conversaciones que haya tenido con otros usuarios
 - ii. Una vez que inicie sesión, por defecto se debe visualizar en la parte derecha el historial de la última conversación enviada o recibida.
 - iii. Realizar búsqueda por nombre de usuario o mensajes enviados o recibidos.
 - iv. Al dar click sobre una conversación ver desplegado, en la parte derecha de la página, todo el historial de mensajes enviados y recibidos con ese contacto.
 - v. Al dar click sobre la imagen de perfil debe reemplazar el contenido de la sección izquierda de la página como se muestra en la imagen 3:
 - 1. Al dar click en el botón editar debe activarse el input para editar el nombre de perfil
 - 2. Este contenedor con la información de perfil también debe incluir un input con atributo type="url" que permita editar la imagen de perfil.
 - 3. Al dar click en el botón back debe devolverlo a la vista principal de la página home



Imagen 3.

- i. En la conversación desplegada en la parte derecha de la página el usuario debe tener la posibilidad de enviar un nuevo mensaje mediante el evento 'submit' del form ubicado en la parte inferior de la página. Este form debe contener un input o textarea y el botón de enviar mensaje.
- ii. Cada mensaje enviado por el usuario logueado debe cambiar de color azul al ícono doblecheck cuando el mensaje haya sido visto y tener habilitado un botón que permita realizar las acciones de editar o eliminar mensaje.
- iii. El usuario al dar click sobre el botón de actions o acciones indicadas en el ítem anterior se debe desplegar una caja con las opciones mencionadas:
 - a. Al dar click sobre la opción editar se debe activar o desplegar un input con el mensaje listo para ser modificado.
 - b. Al dar click sobre la opción eliminar debe aparecer un confirm preguntando al usuario si está seguro de eliminar el mensaje, si el usuario da click en aceptar se procede a realizar la petición DELETE.
- iv. Al dar click sobre el botón buscar de la conversación seleccionada (parte derecha), debe aparecer una nueva

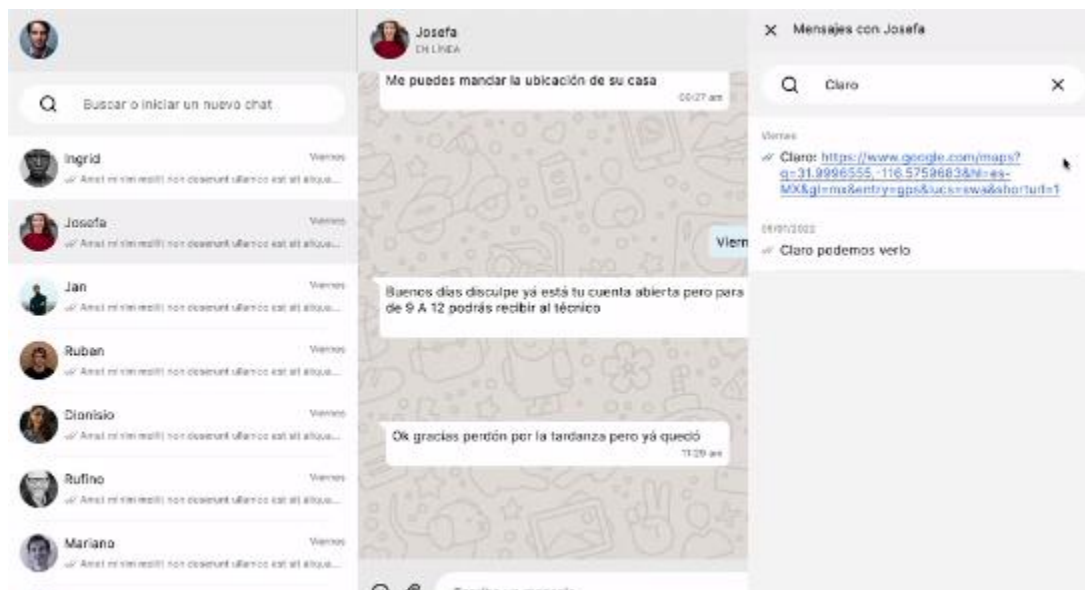


Imagen 4

sección en la parte derecha de la página como se muestra en la imagen 4 y ser posible la búsqueda por palabra incluida en un mensaje enviado.

- d. En la página Sign in debe tener un link llamado “Sign up for free” que al dar click sobre él debe permitir al usuario:
 - i. Redireccionarlo a la página Sign up
 - ii. La página Sign up debe contener:
 - 1. El título Create an account
 - 2. Un form con mínimo 4 inputs, una etiqueta <textarea> como campo de entrada para ingresar la frase y un botón type=”submit” llamado Sign up.
 - 3. Los inputs debenser:
 - a. Un input Type=”text”para el nombre
 - b. Un input type=”number” para el número de celular
 - c. Un input type=”password”
 - d. Un input type =”url” para ingresar la url de la imagen del usuario.
 - e. Al dar click en el botón Sign up debe:
 - i. Verificar que el número de celular no exista en la lista de usuarios.
 - 1. Si existe, debe mostrar un mensaje o alert diciendo que el número de celular ingresado ya está registrado.
 - 2. Si no existe, se procede a crear el nuevo usuario mediante la petición POST y por último mostrar una ventana modal o alert que indique que el nuevo usuario fue creado exitosamente.
 - 3. Los usuarios que sean creados en el Signup deben ser capaces de loguearse o iniciar sesión en el sign in y continuar el flujo especificado anteriormente.

6. En el aplicativo se debe evidenciar el uso de:
- Condicionales
 - Peticiones HTTPs con Axios
 - Eventos del DOM
 - Uso del objeto sessionStorage o localStorage
 - Métodos de arrays
 - Uso de WebPack y babel
 - Uso de Flexbox y Media queries
 - Principios de código limpio
 - Implementación de JSON server (API)
 - Implementación de [Sweetalert](#) para mostrar las alertas.
 - Despliegue de la página en GitHub pages

Nota: Todas las páginas deben conservar los mismos estilos, para el manejo de fechas y horas se sugiere uso de la librería [luxon](#). Cualquier funcionalidad adicional implementada, será tomada en cuenta como punto extra sobre la calificación del taller, después de haber completado los requerimientos mínimos.