Nombre: Andy Ezequiel Sanic Tiul Carnet: 202006699

Link: <https://github.com/AndyST70/Practicas-SA-A-202006699/tree/main/Tareas%20vacas/tarea%20extra>

# Caso de uso Vibe Coding : Aplicación autenticación tienda en linea

Se desarrollo una aplicación de autenticación para un sistema de tienda en línea el sistema permite lo siguiente:

* Se registran nuevos usuarios con nombre, correo y contraseña
* Inician sesión con email y contraseña

Se protegen por ultimo las contraseñas con SHA256, se izo uso de JWT para la autenticación, el frontend usa react, mui, para una interfaz moderna.

## Backend

Quiero que construyas una aplicación de autenticación básica para una tienda en línea usando las siguientes herramientas y condiciones:

* Backend: Flask en Python (sin usar Blueprints).
* Base de datos: MySQL (usa mysql-connector-python).
* Seguridad: Las contraseñas deben ser hasheadas usando hashlib.sha256.
* Frontend: React con Material-UI (MUI).
* La aplicación debe incluir: Un formulario de registro de usuario (nombre, correo, contraseña). Un formulario de inicio de sesión.
* Validación en ambos formularios (campos vacíos, correos repetidos). Manejo de sesiones simples (usando session de Flask). Interfaz clara y responsiva con Material UI.

Restricciones: No se permite usar Blueprints de Flask.

El código debe estar separado pero en un solo archivo para backend (app.py por ejemplo). Todo debe estar bien documentado con comentarios. crea el script de la base de datos y haz uso de JWT Objetivo final: Que un usuario pueda registrarse y luego iniciar sesión en una tienda. Si el login es exitoso, que se le muestre un mensaje de bienvenida con su nombre. Si falla, un mensaje de error.

## Frontend

Necesito que construyas el frontend en React con Material UI (MUI) para una aplicación de autenticación que ya tiene su backend funcionando con Flask y MySQL.

La aplicación debe tener:

* Un formulario de registro con los campos: nombre, email, password.
* Un formulario de inicio de sesión con los campos: email, password.
* Un componente de recuperación de contraseña opcional (como Recuperar.jsx).
* Requisitos: Usa useState para manejar el formulario.
* Usa FormData para enviar datos al backend.
* Usa fetch o funciones importadas desde endpoints/ para conectarse con el backend (por ejemplo: registrar(formData), login(formData)).
* Muestra mensajes de éxito o error con <Typography> usando color verde (success.main) o rojo (error).

# Herramienta de Vibe Coding : Aplicación de autenticación

Se hizo uso de Chat-Gpt para este trabajo

Salida que emitió para el backend

### Backend

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

### FrontendTexto El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

### Ajustes y mejoras para backend

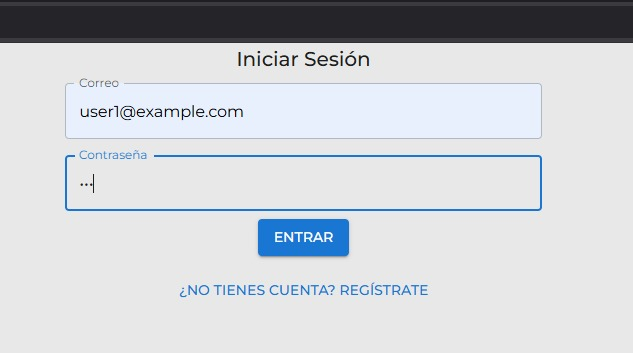
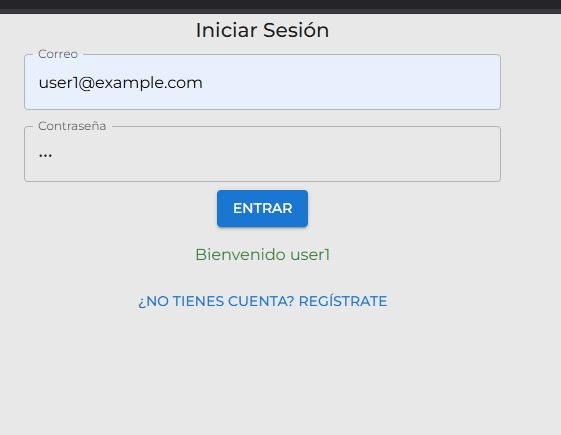
Cabe resaltar que el backend tuvo un par de complicaciones que se arreglaron manualmente, entre ellas la exposición de contraseñas críticas, y un mal manejo al cerrar y abrir las conexiones, pero se arreglo de forma manual, al igual que el manejo de CORS, pero se reestructuro el código en ese sentido y quedó funcional.

### Ajustes y mejoras PARA FRONT

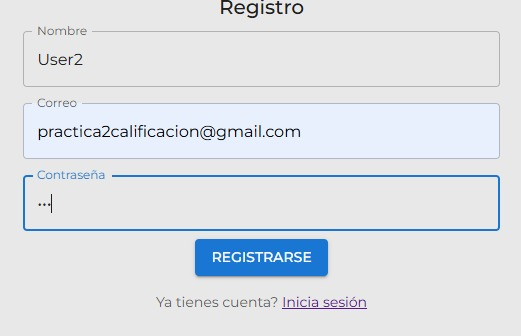
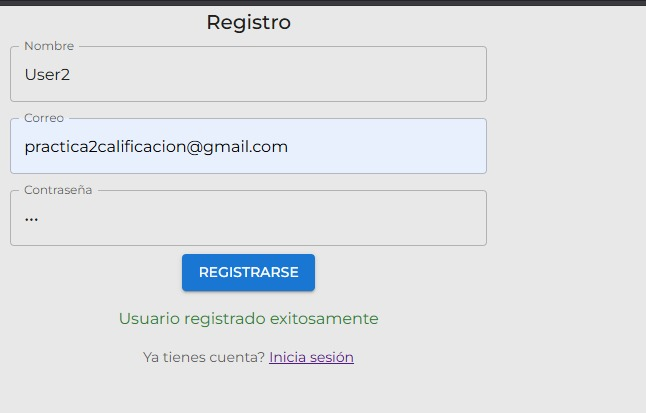
En el frontend ocurrió algo similar: el código generó algunos problemas relacionados con dimensiones, navegación entre páginas y la lógica de estructura, como hacer la transacción entre páginas, dimensiones de textos, de hecho, toco pedirle la misma documentación una ocasión pero se pudo resolver, genero una página muy bien estructurada, si se tiene un formato inicial y se agrega, se puede optimizar el tiempo de gestión y como se desarrolla nuevas páginas

# Capturas de producto final

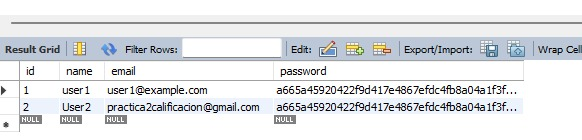
### Iniciar sesion

### Registrar usuario

### Usuario registrado



## Conclusion

El uso de Vibe Coding permitió acelerar el desarrollo de una aplicación de autenticación básica, resolviendo tareas repetitivas y permitiendo enfocarse en ajustes críticos y experiencia de usuario. Aunque fue necesario intervenir manualmente en varios puntos (como seguridad, CORS y navegación), el flujo base generado fue útil y adaptable.