

Prototipo – Tarea Corta

Bases de Datos

Estudiantes:

Brayan Alpízar

Christian Navarro

Jorge Gutiérrez

Dario Garro

Giancarlo Vega Marín

Documentación técnica

## **Métodos implementados, estructuras de datos y algoritmos**

### **1. Admin.js**

- **Función:** Interfaz para la gestión de usuarios y cuentas.
- **Estructuras de datos:**
  - **useState** para manejar estados como usuarios, cuentas, **selectedUser**, etc.
  - **useEffect** para cargar datos al montar el componente.
- **Métodos:**
  - **handleDeleteUser:** Elimina un usuario seleccionado.
  - **handleDeleteAccount:** Elimina una cuenta seleccionada.
  - **handleModifyUser:** Modifica los datos de un usuario.
- **Algoritmos:**
  - Filtrado y mapeo de listas para mostrar y actualizar usuarios y cuentas.

### **2. Inicio.js**

- **Función:** Pantalla de inicio después del login.

- **Estructuras de datos:**
  - **useState** para manejar el estado de la cuenta actual.
  - **useEffect** para cargar la cuenta desde **localStorage**.
- **Métodos:**
  - **handleLogout:** Cierra la sesión y redirige al login.
- **Algoritmos:**
  - **Lectura y escritura en localStorage.**[tecnetium.com](https://tecnetium.com)

### 3. Login.js

- **Función:** Manejo de autenticación y registro de usuarios.
- **Estructuras de datos:**
  - **useState** para **username**, **password**, **newUser**, etc.
  - **useEffect** para cargar usuarios desde **localStorage**.
- **Métodos:**
  - **handleLogin:** Autentica al usuario mediante una solicitud POST al backend.
  - **handleRegister:** Registra un nuevo usuario mediante una solicitud POST al backend.
  - **handleChange:** Actualiza el estado **newUser** al cambiar los campos del formulario.
  - **toggleForm:** Alterna entre los formularios de login y registro.
- **Algoritmos:**
  - **Validación de campos de entrada.**
  - **Manejo de respuestas del backend para autenticación y registro.**

### 4. Movimientos.js

- **Función:** Visualización de movimientos financieros de la cuenta.
- **Estructuras de datos:**
  - `useState` para movimientos y `cuentaActual`.
  - `useEffect` para cargar movimientos desde el backend.
- **Métodos:**
  - `fetchMovimientos`: Obtiene los movimientos de la cuenta actual desde el backend.
- **Algoritmos:**
  - Filtrado de movimientos por número de cuenta.

## 5. Prestamos.js

- **Función:** Gestión de préstamos de los usuarios.
- **Estructuras de datos:**
  - `useState` para `prestamos`, `nuevoPrestamo`, etc.
  - `useEffect` para cargar préstamos desde el backend.
- **Métodos:**
  - `handleSolicitarPrestamo`: Envía una solicitud de préstamo al backend.
  - `handlePagarPrestamo`: Realiza el pago de un préstamo existente.
- **Algoritmos:**
  - Cálculo de montos y validación de datos de préstamos.

## 6. Tarjetas.js

- **Función:** Gestión de tarjetas asociadas a la cuenta.
- **Estructuras de datos:**
  - `useState` para `tarjetas`, `nuevaTarjeta`, etc.

- **useEffect** para cargar tarjetas desde el backend.
- **Métodos:**
  - **handleAgregarTarjeta:** Agrega una nueva tarjeta mediante una solicitud POST al backend.
  - **handleEliminarTarjeta:** Elimina una tarjeta existente.
- **Algoritmos:**
  - **Validación de datos de tarjetas y actualización del estado local.**

## 7. Transferencia.js

- **Función:** Realización de transferencias entre cuentas.
- **Estructuras de datos:**
  - **useState** para transferencia, cuentaOrigen, etc.
  - **useEffect** para cargar cuentas desde el backend.
- **Métodos:**
  - **handleTransferir:** Envía una solicitud de transferencia al backend.
- **Algoritmos:**
  - **Validación de cuentas y montos antes de realizar la transferencia.**

## Problemas conocidos

- La conexión entre el servidor y la App Móvil causó muchos problemas. El login y registro funcionaban, pero no lograban obtener los otros valores en los JSON que permitía realizar las funciones de transferencia, pagos, préstamos, etc.
- El emulador de teléfono consumió muchos recursos, y de paso, no funcionaba en Android Studio.

- Las URLs no funcionaban directamente con el servidor.

## **Problemas encontrados**

- Uno de los principales problemas que encontramos en este proyecto fue un inconveniente a largo plazo relacionado con los modelos. Esto se debió a que los modelos sufrían cambios, y en varios casos, dichos cambios no se reflejaban adecuadamente en los archivos .json correspondientes. Como resultado, muchos datos enviados no coincidían con la estructura esperada, lo que provocó fallos en funciones que antes funcionaban correctamente. Al no considerar la actualización de los archivos .json, algunas funciones dejaron de ejecutarse y ciertos elementos no se mostraban como se esperaba.
- Otro problema fue la creación de estilos globales de css, los cuales evitaban diferenciar las diferentes UI creadas en este proyecto. Se solucionó creando módulos de css para que de esa manera los estilos no interrumpieran a otros.
- La comunicación entre el servidor y el frontend por medio de los JSON, la solución fue definir un standard para comunicarse con los nombres, un status de si la operación fue correcta y un mensaje.
- El recibimiento de los JSON provenientes del frontend se solucionó utilizando clases modelo para convertirlas a objetos y luego operar con ellas
- Al principio, el servidor y frontend no lograban ser cargados en el IIS. Se requirió instalar componentes y programas adicionales, añadir permisos manualmente, y limpiar procesos que estaban acaparando los puertos.

## **Bitácora**

### **Actividades Planeadas**

Creación de repositorio: 3 de abril

Responsable: Giancarlo Vega

Creación de comunicación JSON: 7 de abril

Responsable: Jorge Gutierrez

Creación de servidor Backend: 8 de abril

Responsable: Brayan Alpizar

Creación y conexión de App: 9 de abril

Responsable: Dario Garro y Giancarlo Vega

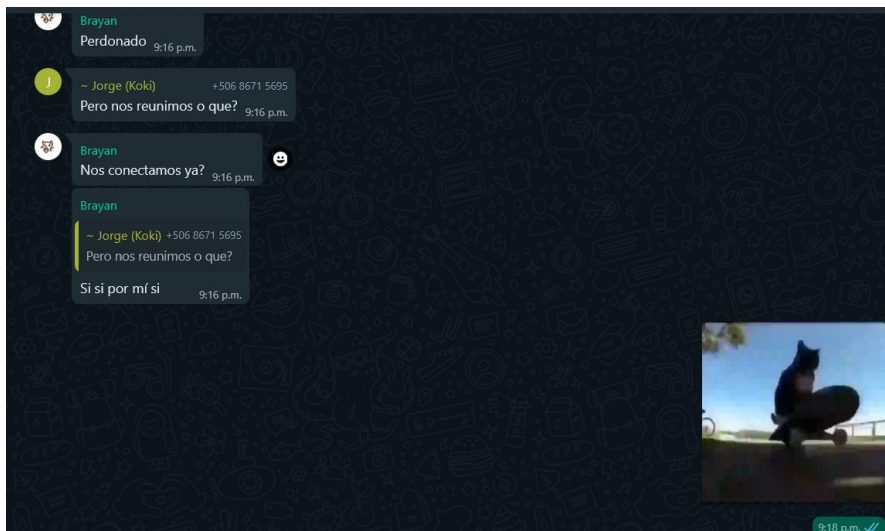
Realización de bitácoras: 10 de abril


Responsable: Giancarlo Vega Marín

## Sesiones

3 de abril

Se convoca a una reunión rápida en Discord, de 30 minutos a 1 hora, para discutir sobre el proyecto, como nos dividimos el trabajo, y tener un estimado de cuando entregarlo. Se divide a cada uno los roles, y se crea el repositorio





**jungianca6**
4/3/25, 9:29 PM

[https://github.com/jungianca6/Proyectos\\_Bases](https://github.com/jungianca6/Proyectos_Bases)

GitHub

**GitHub - jungianca6/Proyectos\_Bases**

Contribute to jungianca6/Proyectos\_Bases development by creating an account on GitHub.

jungianca6/

**Proyectos\_Bases**

Repositorio para los proyectos, y otras asignaciones, del curso Bases de Datos

1

Contributor

0


Issues

0


Stars

0


Forks




Internet Information Services (IIS)




**SpaceBA**
4/3/25, 9:36 PM




**CasoTecBank (1).pdf**  
 149.15 KB

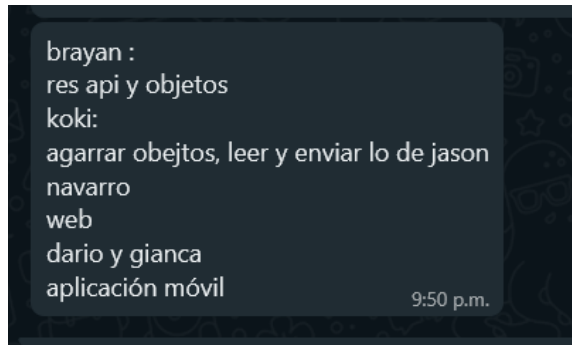


**Bases**


Brayan, Christian, Dario, +506 8671 5695, Tú

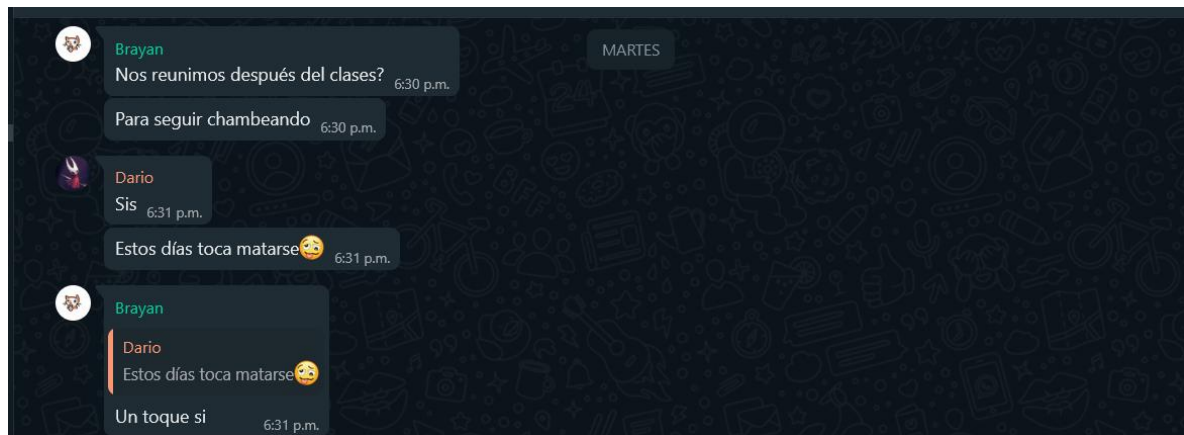
rest api  
función de lectura y escritura de jasons  
conexión del rest api y la web  
conexión de del rest api y la aplicación móvil  
aplicación móvil  
modificación de los paramtros desde la werb (peticiones post y delet)  
el ejecutable IIS  
back end( c# )  
gestion de roles  
gestion de clientes  
gestion de cuentas  
gestion de tarjetas  
gestion de ascesores de crédito  
generación de reportes  
gestion de prestamos  
1impelmentacion del calendario de pagos  
gestion de mora(generación del reporte)  
lectura y escritura de datos de disco  
incriptacion  
back end parte cliente  
registro y log in  
movimientos de cuentas  
transferencias  
trajetas  
listado de compras realizadas  
prestamos  
incriptacion  
aplicación móvil  
mismas aplicaciones móvil y web cliente



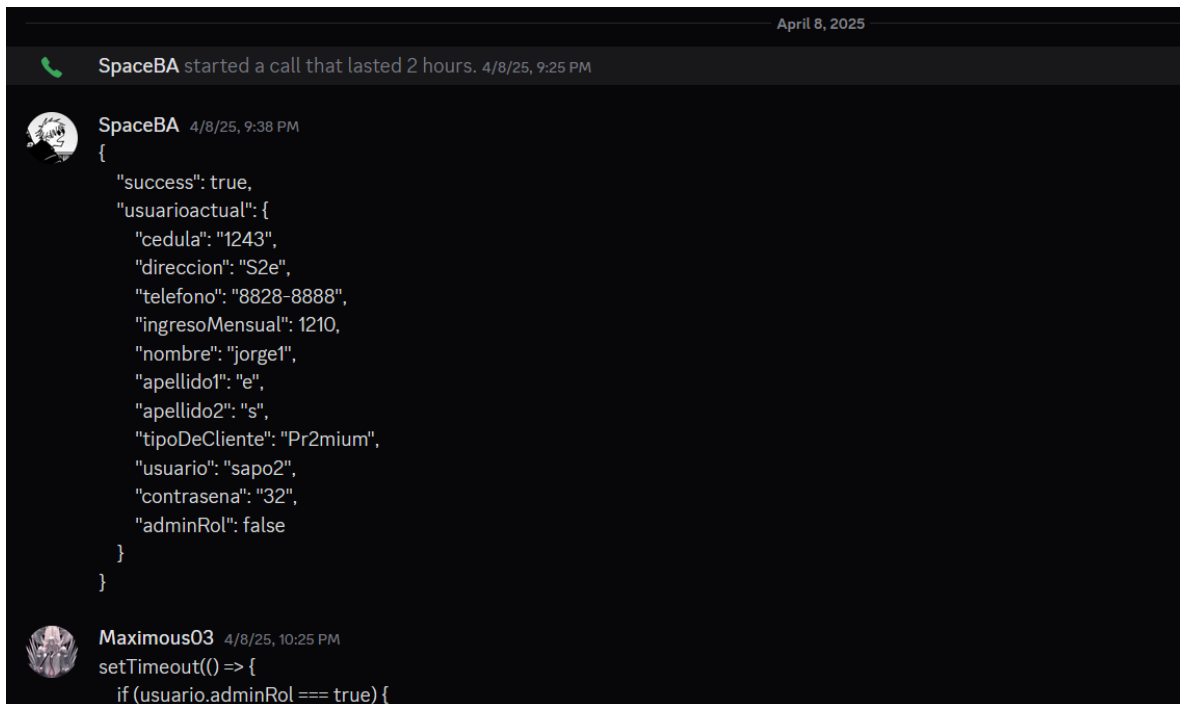


8 de abril de 2025

Se convoca a una reunión larga, de mínimo 2 horas, donde comenzamos a trabajar en el servidor y en la Web. Se inician también con los JSON que servirán para las conexiones



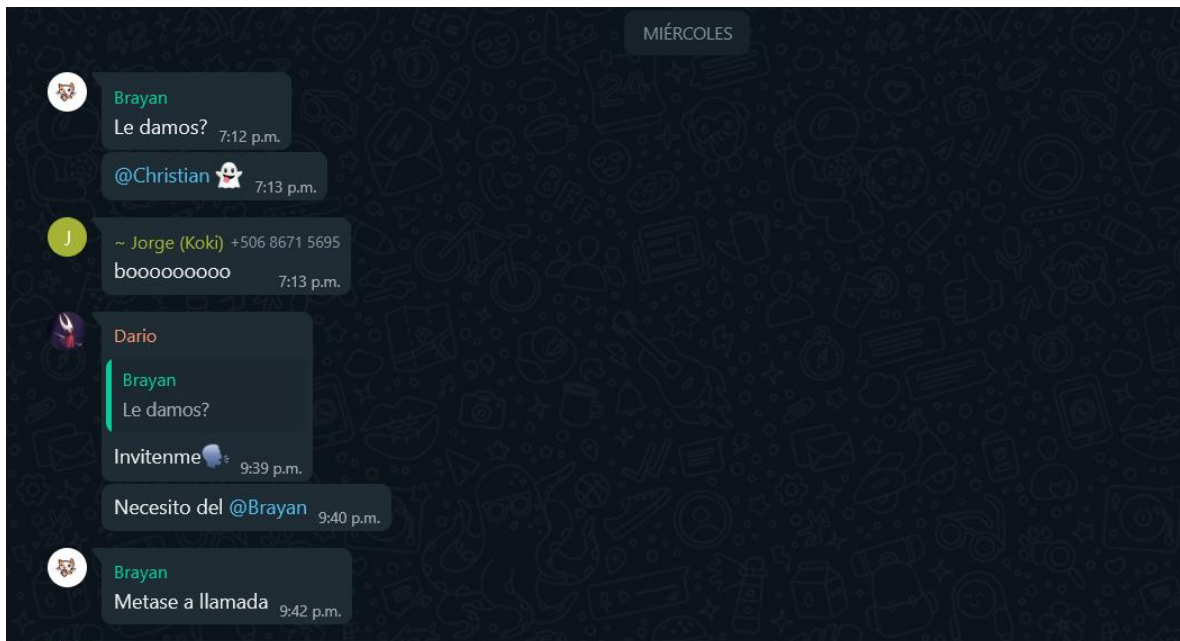


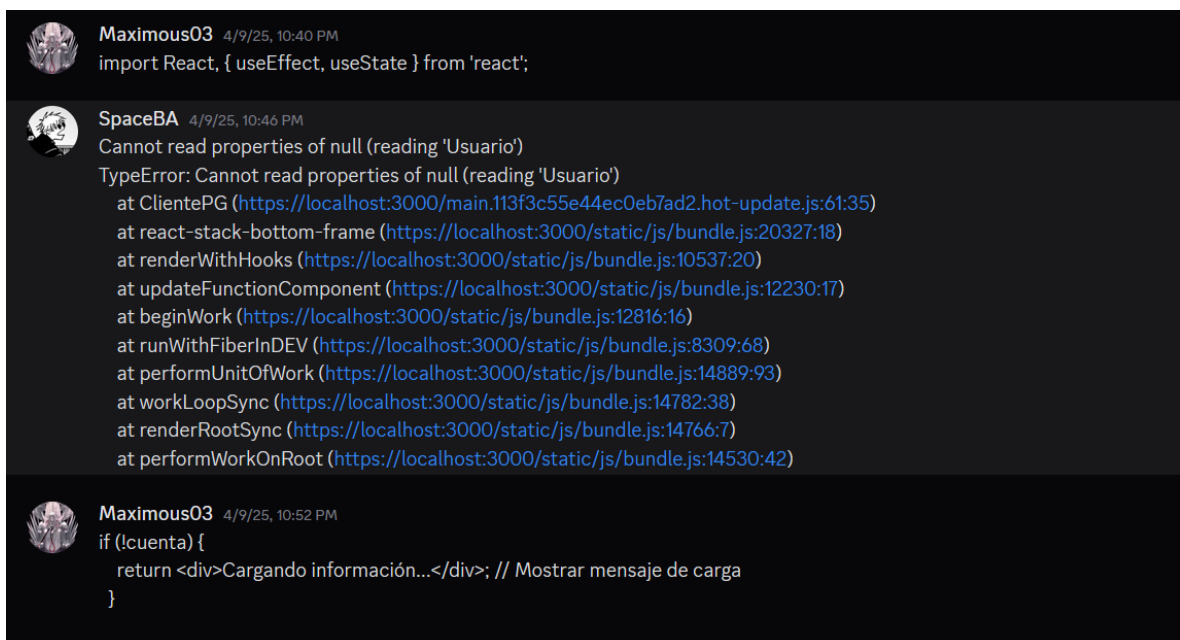
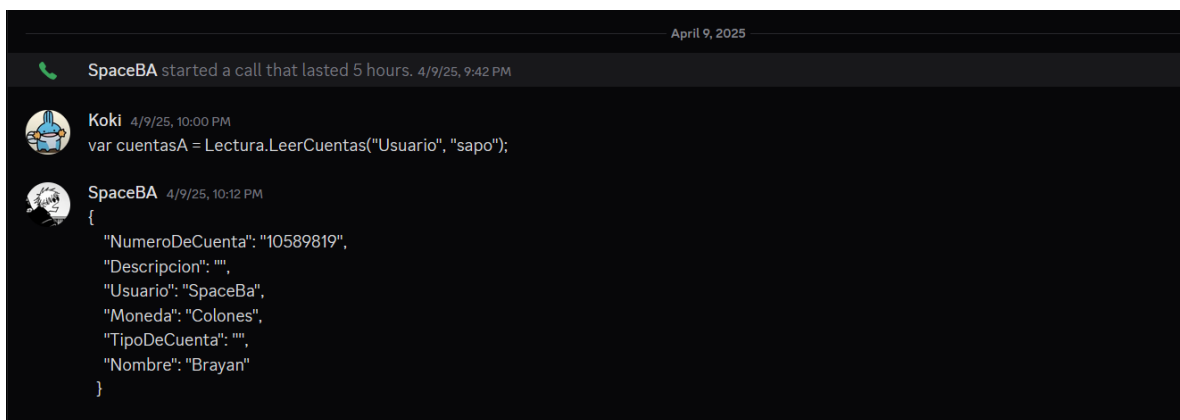
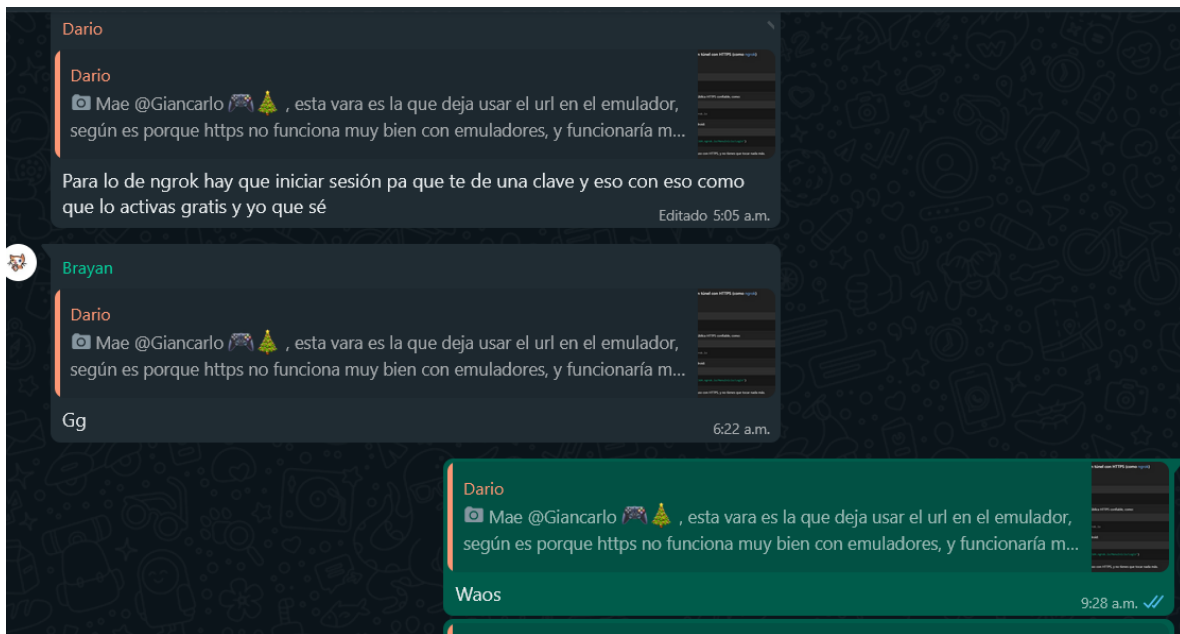


9 de abril

Se continúa trabajando por aparte los roles de cada uno. El compañero Dario logra resolver un problema en el que la URL no conectaba la Aplicación al servidor Backend.

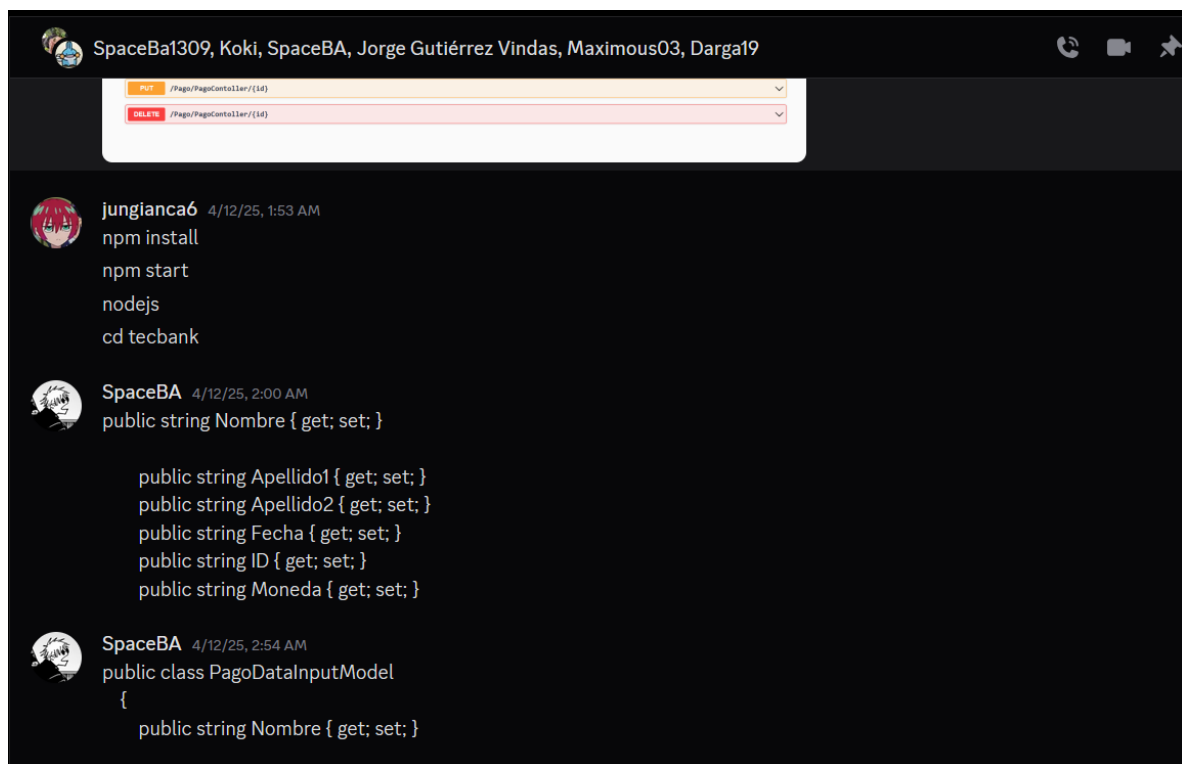
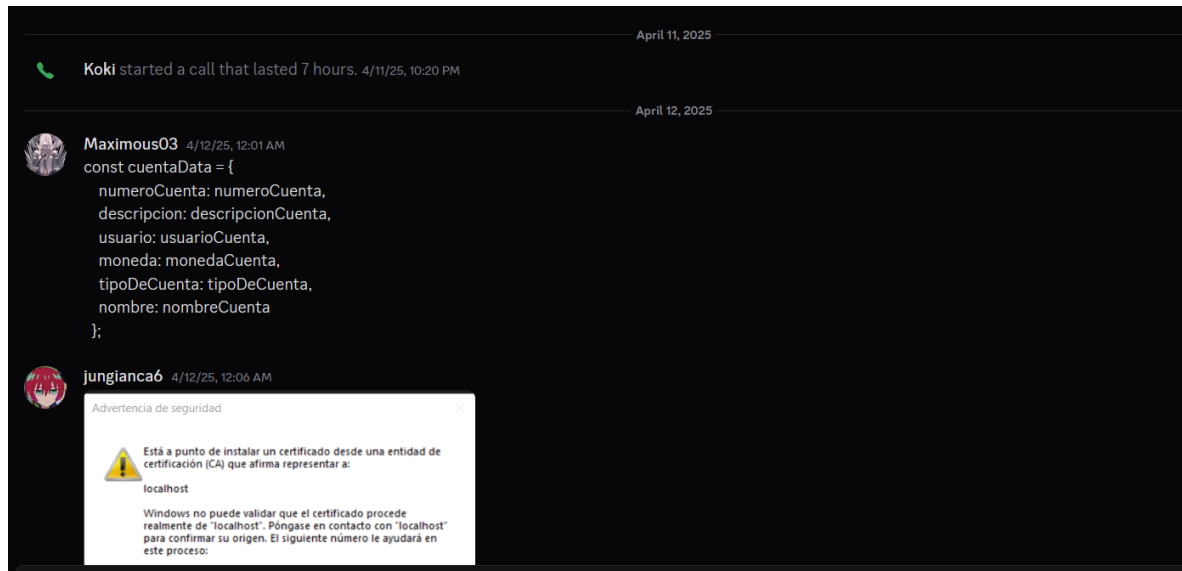
Se continúa modelando y codificando el Frontend y Backend. En medio de la llamada, se realizaban pruebas para ir probando la conexión entre ellos.





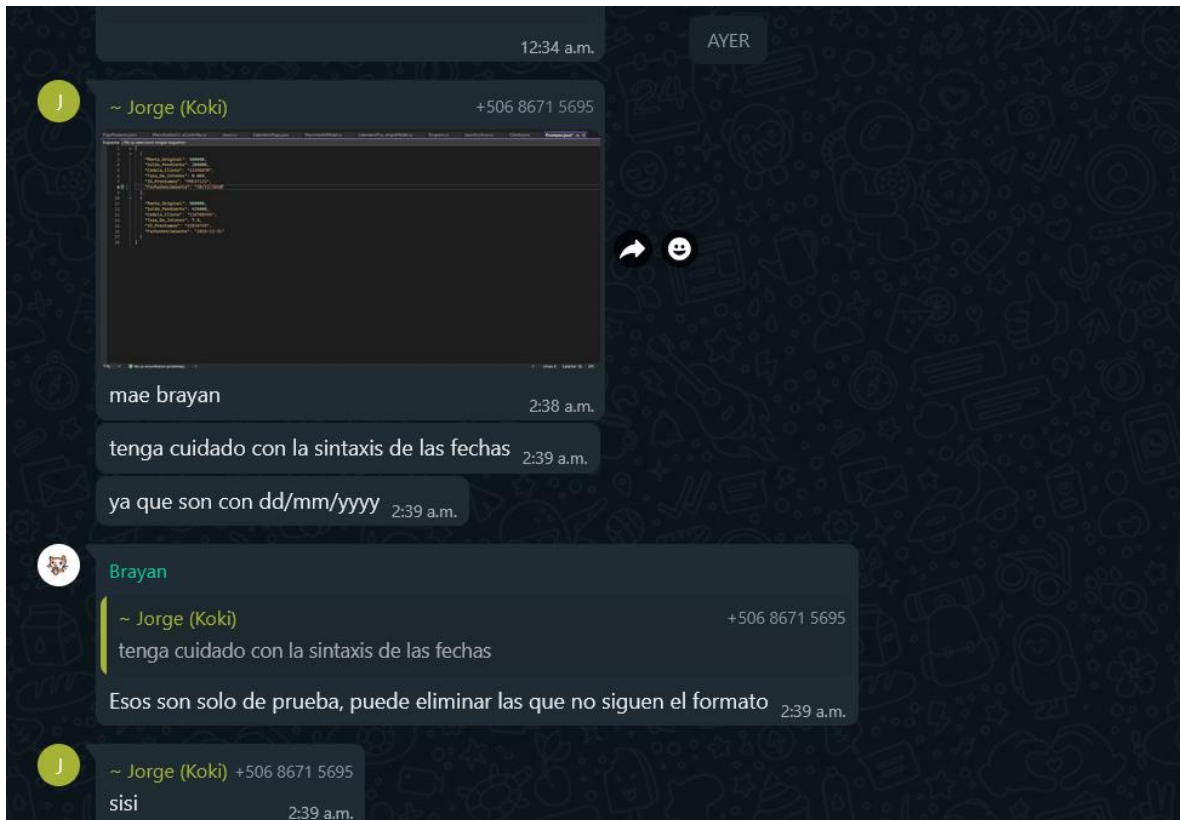
11 y 12 de abril

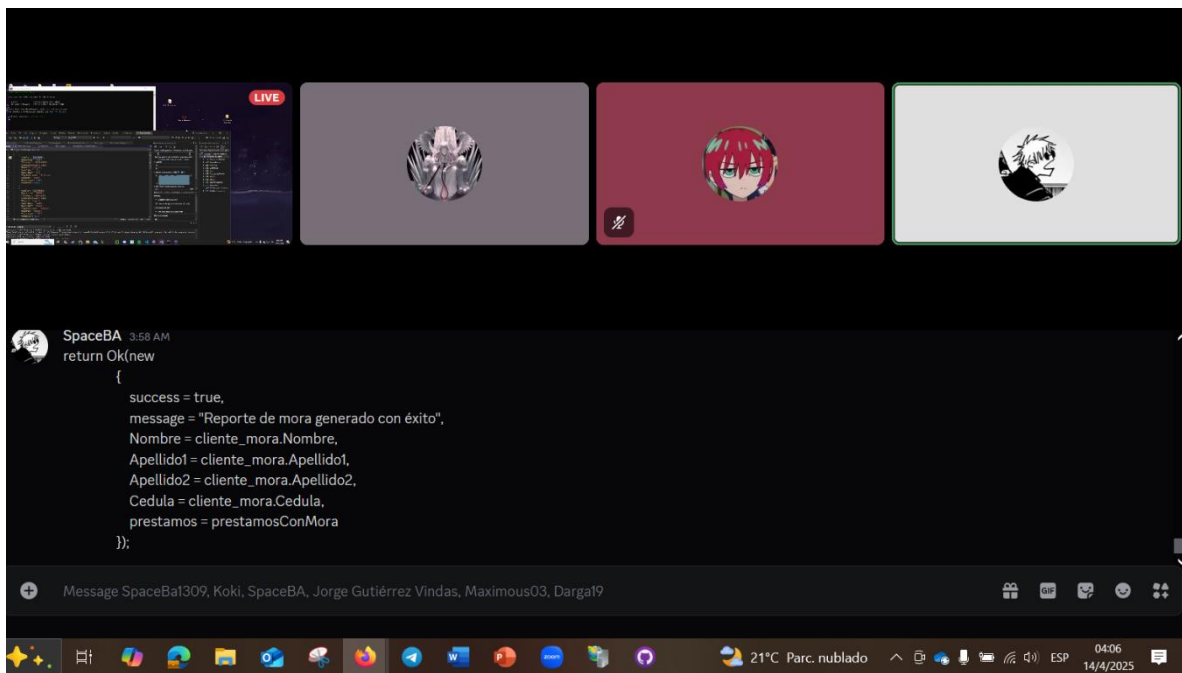
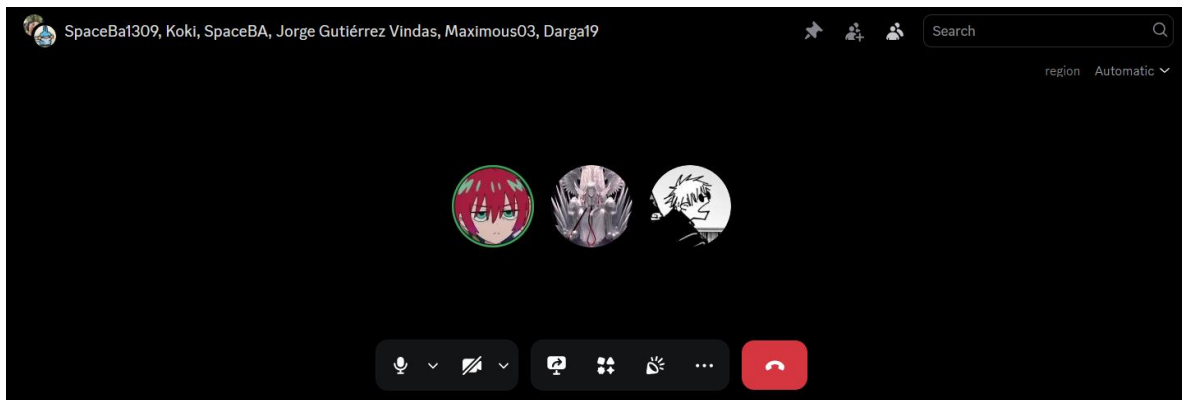
En una llamada que casi duró 7 horas, se continuó puliendo los toques finales. Ya existía una versión ejecutable para probar en la computadora de la defensa (la de Giancarlo), y en la reunión se aseguró de que ambas corrieran bien y se conectaran. Se continuó trabajando con los JSON, Endpoints y las conexiones.



13 de abril de 2025

El día antes de la entrega, se hace otra reunión para los toques finales. El compañero Christian añade las funciones faltantes en la FrontEnd del Administrador, los compañeros Brayan y Jorge terminan el Backend en conjunto con el trabajo de Christian, y el compañero Giancarlo añade el Frontend y el Backend al IIS.

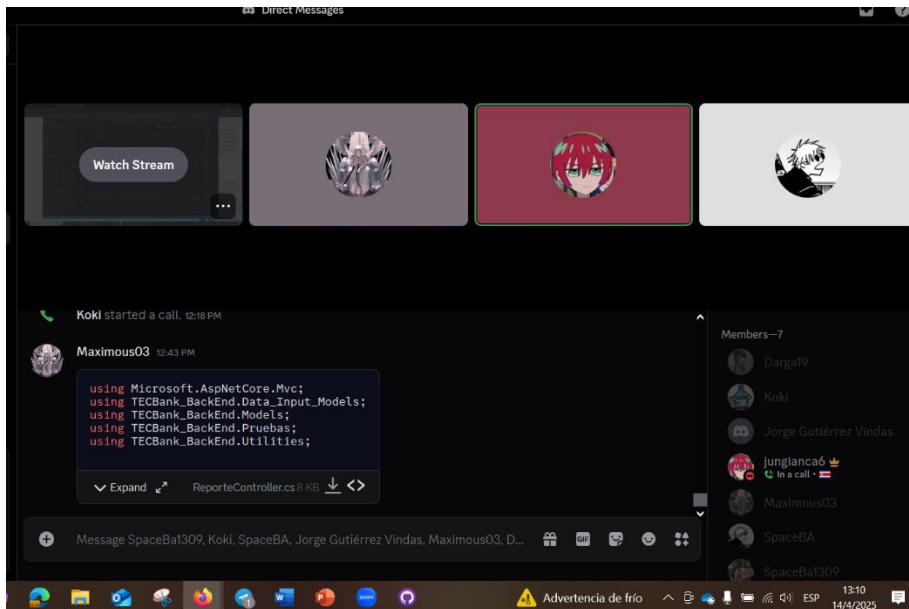
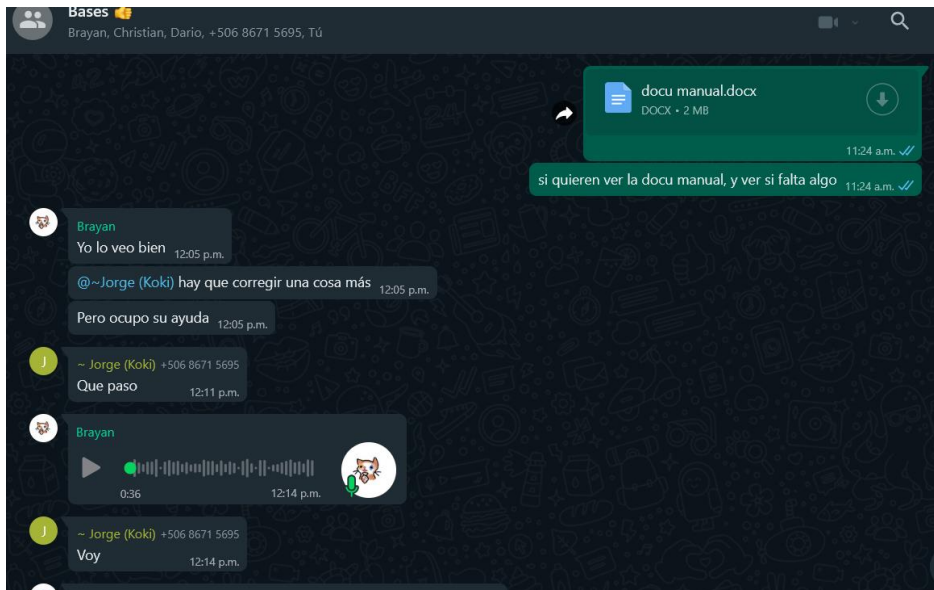




14 de abril de 2025

Último día para reunirse. El compañero Giancarlo termina las bitácoras, los compañeros Jorge y Brayan terminan con detalles finales del Backend, y Christian finaliza el Frontend. Se sube todo al repositorio, y se realiza una última reunión corta para asegurarse de que todo esté funcional.

Lastimosamente, la App Móvil queda incompleta.



## Conclusiones

- Se desarrolla la aplicación que permita gestionar la descripción del caso TecBank
- SE crea un API en Backend programado en C# que realiza la conexión con el FrontEnd y la App
- Se crea la App Móvil, pero incompleto y muchas fallas y funciones faltantes
- Se crea el prototipo de la pagina web de forma funcional, con vista de cliente y administrador.

- Se utiliza React y Bootstrap para la elaboración, lo cual facilitó bastante el proceso de la creación del FrontEnd
- Se logra conectar exitosamente el FrontEnd y Backend al local IIS, permitiendo una mayor facilidad al iniciar dichos sitios.

## Recomendaciones

- Utilizar Retrofit y no OKHttp para la conexión de la App con el servidor
- Utilizar un teléfono y no un emulador para el desarrollo en la IDE de la App
- Si es posible, investigar que tanto facilita Flutter el desarrollo móvil
- Utilizar mejor el tiempo, sobre todo porque problemas siempre pueden ocurrir