

## Etapa 4: Resolución del *frontend*

Para esta etapa ya deberías tener resuelto:

- El modelo de negocios del sistema de *fintech* desarrollado en JavaScript.
- Las pruebas unitarias para validar el modelo de negocios anterior.

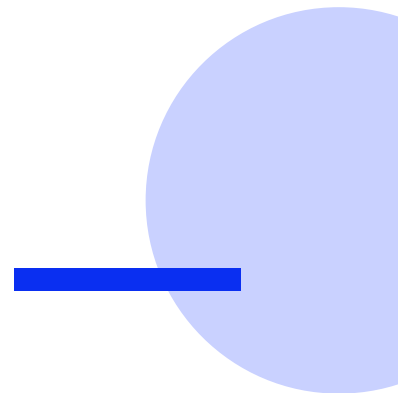
En esta etapa desarrollarás:

- La **página principal**.
- Los **componentes** de la aplicación.
- **Conectar los componentes** con la lógica de negocios.



## 1. Componentes de la aplicación a desarrollar:

- Enrutador.
- *Login*.
- Cuenta de usuario.
- Transferencia.
- Para **simular el ingreso** del dinero.
- Para registrar un **usuario nuevo** (opcional).  
O bien, crear algunos usuarios de prueba para *testear*.



## Prompts a utilizar

### Clase Router

Genera una clase **Router** en JavaScript con un constructor que reciba **parent** (el elemento HTML contenedor), **model** (datos del modelo), y un método **navigateTo** que instancie un **Componente** (la clase del componente) con **parent**, **model**, y **this** (la instancia del Router), llamando al método render del Componente para mostrarlo.



### Componente Login

Genera una clase **LoginComponent** en JavaScript con un constructor que reciba **parent** (el elemento HTML contenedor), **model** (el objeto que maneja los datos y la lógica de autenticación), y **router** (el objeto **Router**). Esta clase debe tener los métodos **handleRegistrarse** para navegar a un **RegistroComponent**, **handleIniciarSesion** para manejar el inicio de sesión con validación de username y password, y render para construir y mostrar el formulario de inicio de sesión con eventos de clic para los botones de registro e inicio de sesión. Usar **innerHTML**.

## Componente Cuenta

Genera una clase **CuentaComponent** en JavaScript con un constructor que reciba **parent** (elemento HTML contenedor), **model** (datos de la cuenta actual), y **router** (objeto Router). Incluye métodos **handleSalir**, **handleIngresarDinero**, **handleVerMovimientos**, y **handleTransferir** para manejar la navegación a los componentes correspondientes, y un método **render** que muestre la interfaz de la cuenta con botones estilizados para estas acciones y asigne eventos de clic a cada botón. Utilizar **innerHTML** en el **render**.

## Componente Transferencia

Genera una clase **TransferenciaComponent**, en JavaScript, con un constructor que reciba **parent** (elemento HTML contenedor), **model** (datos de la cuenta actual), y **router** (objeto Router). Incluye los métodos **handleVolver** y **handleTransferir** para manejar la navegación de vuelta al componente de cuenta y realizar la transferencia de dinero, respectivamente. También, un método **render** que muestre la interfaz para transferir dinero, incluyendo **campos para correo del destinatario y monto**, botones estilizados para confirmar la transferencia y volver, y la asignación de eventos de clic a cada botón.

## Componente *Simular ingreso*

Genera una clase **IngresoDineroComponent**, en JavaScript, con un constructor que reciba **parent** (elemento HTML contenedor), **model** (datos de la cuenta actual), y **router** (objeto Router). Incluye métodos **handleVolver** para regresar al componente de cuenta y **handleIngresarDinero** para realizar el ingreso de dinero, y un método **render** que muestre la interfaz para ingresar dinero, incluyendo **campos para el monto y motivo**, botones estilizados para confirmar el ingreso y volver, y la asignación de eventos de clic a cada botón.

## Componente *Registro*

Genera una clase **RegistroComponent** en JavaScript con un constructor que reciba **parent** (elemento HTML contenedor), **model** (datos de la aplicación), y **router** (objeto Router). Incluye métodos **handleRegistro** para registrar un nuevo usuario y **handleVolverAlLogin** para regresar al componente de login, y un método **render** que muestre la interfaz de registro con campos para nombre, correo, contraseña, y confirmación de contraseña, botones estilizados para registrarse y volver, y la asignación de eventos de clic a cada botón.

## 2. Comparte tu resolución:

Recuerda **compartir tu proyecto**, una vez que lo hayas terminado, con **Github Pages**.

Puedes mostrarlo como parte de tu portfolio, compartirlo con tus conocidos y toda la comunidad.



A continuación, en la línea de tiempo de Alumni, encontrarás su resolución, para que puedas verificar cómo te fue.

