

El objetivo de la prueba es replicar el diseño de figma aportado añadiendo las funcionalidades que se describen en este documento. Tecnologías que se deben utilizar:

- **Frontend:** React con typescript, ya sea con Next (preferiblemente) o Vite.
- **Backend:** Node.js con typescript
- **Blockchain:** Librería ethers.js

Figma

<https://www.figma.com/design/EiMzp2NwOeMXkpnH8Tq1wl/Metlabs-Tech?node-id=1-12288&t=xRvqkFE8s8HSPMwX-1>

Resumen de la prueba

El diseño cuenta con dos páginas, una primera de *Login/registro* y otra de *perfil de usuario*.

En la página de *Login/registro* se pretende implementar un registro de usuario mediante email y contraseña así como a través de una cuenta de Google.

- En el backend, no se requiere estrictamente utilizar una base de datos para gestionar los usuarios. Se puede utilizar un archivo json como base de datos para simplificar el tiempo de montar todo el entorno de trabajo.

En la página de *perfil de usuario* se tienen que poder hacer las siguientes tareas:

- Conectar una wallet de usuario. Para esta conexión se debe utilizar la librería [web3-onboard](#).
- Realizar las acciones de [depositar y retirar](#). Estas acciones son llamadas blockchain que se realizan en el contrato que se añade más abajo. Además, se debe guardar en la base de datos (puede ser un archivo json que simula una base de datos postgres) el hash de esas transacciones y la wallet de usuario que las ha realizado.
- Visualizar el [balance de un usuario en el contrato](#), en base a las cantidades que haya depositado y retirado.
- Los datos del usuario en el perfil, son datos hardcoded, pero tienen que ser cargados desde el backend.

El resto de detalles de la página de perfil de usuario solo tienen que estar maquetados, sin necesidad de que tengan ninguna funcionalidad.

Información sobre el smart contract

- **Red:** Sepolia
- **Smart contract Address:** 0x25d7bC103596F11749605aA6ff935e96022ce63B
- **ABI:** En archivo adjunto

Aspectos que se valoran

- Maquetación del frontend acorde al figma.
- Buenas prácticas en el código y organización del proyecto.
- Diseño correcto del frontend y backend en términos de eficiencia y seguridad.
- Conocimiento sobre blockchain.
- Código realista, es decir, código que se pueda hacer a una velocidad adecuada en el día a día.

Entrega de la prueba técnica

- Entregar un repositorio público para el frontend y otro para el backend
- Opcional: Entregar la aplicación desplegada