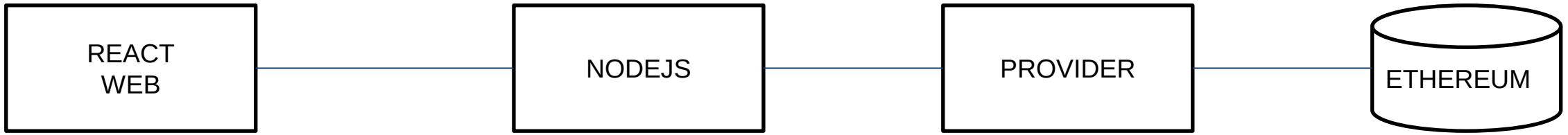
PROYECTO EXPLORER

Objetivos

1. Explorar la cadena de bloques de eth
2. Aprenderemos a usar un provider externo
3. Usaremos el api web3 para acceder a la información
4. Usaremos una aplicación react para interactuar
5. Haremos un api nodejs que interactuará con el provider.
6. Al provider le pediremos lo siguiente
    1. Un bloque con sus transacciones
    2. Una transacción
    3. El saldo de una cuenta.

Esquema del proceso



Desde un formulario realizado en react accedemos a un api en nodejs que accede a un provider de ethereum

Tareas

1. Crear carpeta de **proyecto-explorer**.
2. Estando en la carpeta creamos una carpeta donde metemos un proyecto nodejs que accederá a un provider de Ethereum.
3. Estando en la carpeta creamos con `yarn create vite front-end --template react`
4. Nos damos de alta en un proveedor de datos de Ethereum como infura. https://infura.io/
5. Modificamos la aplicación backend para hacer tres rutas
    1. Búsqueda por bloque
    2. Búsqueda por transaction
    3. Balance de una Account.
6. Modificamos el front-end para hacer un formulario que pida un bloque , tx o address, acceda al backend y muestre resultados.

Crear **proyecto-explorer**

1. Creamos el directorio del proyecto en la carpeta curso

Crear **el backend**

1. Creamos la carpeta backend dentro del directorio del proyecto
2. Hacemos yarn init –y, para crear el package.json
3. Instalamos el paquete express
4. Hacemos un programa llamado index.js para probar que el express funciona.

Crear **el frontend**

1. Dentro de la carpeta del proyecto ejecutamos
2. Ejecutamos yarn create vite frontend –template react
3. Nos pasamos al directorio frontend
4. Abrimos un code con code .
5. Abrimos un terminal dentro del code
6. Ejecutamos yarn
7. Ejecutamos yarn dev
8. Nos vamos a la dirección que nos aparece en el navegador
9. Veremos que nos aparece la aplicación react-

Alta en infura.io

1. Nos vamos a la web, nos registramos y tomamos la url del proyecto para llevarla al backend

Modificamos el backend

1. Incorporamos la librería web3
2. Hacemos las rutas
3. getBlock, getTx, getBalance
4. Las probamos con Curl

Arreglamos el frontend

1. Borramos lo que sobre de la aplicación frontend

Metemos el router

1. Borramos lo que sobre de la aplicación frontend
2. Metemos un router
3. Metermos un header con una caja de texto, donde nos pedirá el bloque, la tx or el balance

Metemos el formulario

1. Metemos el formulario para pedir el dato y navegamos a la ruta

Metemos react-query

1. Metemos react query para acceder al backend.
2. Activamos el cors en el backend
3. Controlamos errores en el backend

Hacemos los componentes

1. Hacemos los componente Balance, Tx, Bloque
2. Presentamos el resultado como un json

Presentamos datos

1. Tomamos los datos de la respuesta del servidor y lo formateamos.

Links

1. Del bloque pasamos a las transacciones
2. De la transacción pasamos al balance

Conclusión

1. Proyecto con frontend y backend usando un servicio externo.
2. Se ha procurado que no tenga mucho código para que sea abarcable.
3. Se puede extender haciendo cosas como las transacciones mas grandes en los últimos bloques.
4. Se podría haber usado un cliente local, aunque el objetivo era usar el api con un provider.
5. Se podría usar con otros providers como quicknode