TU PRIMERA APP WEB

1. **PRIMEROS PASOS**
   1. **CREAR APLICACIÓN REACT**
2. **AUTENTICACIÓN**
3. **ESTADO (I)**
4. **SDK**
5. **NAVEGACIÓN**
6. **ESTADO**
7. **CÓDIGO COMPLETO**
8. **PRIMEROS PASOS**

**(React)** es una biblioteca JS de código abierto para crear interfaces de usuario (mantenido por Facebook y comunidad de software libre).

**(Renderizar)** proceso de generar una imagen fotorrealista a partir de un modelo 2D/3D o programas informáticos. Se suele utilizar Blender.

**(Licencia MIT)** licencia de software del Instituto Tecnológico de Massachusetts.

Pila tecnológica:

-Rúter de alcance: **@reach/router**

-Neurosity (SDK): **@neurosity/sdk**

-Aplicación React

Crear aplicación React

Los siguientes enlaces con archivos **no los he usado por ahora. No sé dónde meterlos…**

<https://github.com/facebook/create-react-app>

<https://github.com/streamich/react-use>

<https://github.com/facebook/create-react-app#creating-an-app>

Primeros pasos

**(CRA: Create React App)** sirve para arrancar la aplicación y ejecutarla localmente.

-No hace falta volver a hacerlo. Instalar **create-react-app:** npx create-react-app my-app

-Abrir <http://localhost:3000/>

-**npm start (desde la carpeta my-app)**

-**index.css:** agrega los estilos para la app

Autenticación

-Formulario inicio de sesión

-3 efectos secundarios que sincronizan el estado de autenticación (onLogin, loading y error)

**//src/components/LoginForm.js:** componente formulario inicio de sesión

**//src/pages/Login.js:** página de inicio de sesión (usa el componente)

Estado

Para el mantenimiento de la aplicación usamos:

-useState: enlace de React

-Reach Router (instalación: npm install @reach/router react-use)

-Enlace de almacenamiento local que ofrece react-use

Por tanto, mantenemos el estado global en el componente de la aplicación y pasamos los accesorios necesarios a los componentes secundarios.

**App.js:** líneas 1:23.

SDK

-Integramos la App con Neurosity->instalando el SDK e importándolo en formato **App.js.**

Npm install @neurosity/sdk

**src/pages/Logout.js:** página de cierre de sesión

NAVEGACIÓN

-Conocer el estado del dispositivo es importante

-Opciones: en línea, no, cargándose, suspendido.

**// src/components/Status.js**

**// src/components/Nav.js**

ESTADO: LÓGICA EN FUNCIÓN DE NUESTRO ESTADO COGNITIVO

<https://docs.neurosity.co/docs/api/calm/>

Al suscribirnos a neurosity.calm() obtenemos una puntuación/segundo.

**// src/pages/Calm.js:** nos suscribimos a la API de Calm

-Se agrega también en **App.js**

Utilizamos:

-React (¿?). Se puede clonar el Repositorio de NotionJS (para que funcione con las 2).

-Autenticación NotionJS (utilizada en primera NodeApp).

-Uso de Reaction (¿¿)

**-Empezando-**

**Create React App (CRA):** ejecutamos la aplicación localmente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Este código crea una hoja en VSCode:

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Con el código: **npm start (Dentro de la carpeta de mind-controller-ocean)**

Se ejecuta la aplicación, se compila en la terminal y te da la dirección del host: <http://192.168.1.38:3000/>

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**./src/index.css:** Se agregan todos los estilos (desde GitHub)

**-Autenticación-**

Notion (computadora cerebral) cuenta con ello por temas de privacidad.

**src/components/LoginForm.js (CREADO):** formulario inicio de sesión y 3 efectos secundarios para sincronizar la autenticación.

**src/pages/Login.js (CREADO):** página de inicio de sesión.

**-Estado de la aplicación-**

Para mantener la aplicación usamos **El Reach Router** + un enlace de almacenamiento local que ofrece react-use (aplicación?)

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**src/App.js:** Se tiene a deviceID en almacenamiento local para tener una mayor experiencia de usuario sin tener que ingresarlo varias veces. Necesitamos acceder a él antes y después de que el usuario haya iniciado sesión.

**-NotionJS-**

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente