

Setup de proyectos

Esta es una guía para hacer el setup inicial de cualquiera de los proyectos que veremos en el curso y lo único que va a diferir es en el nombre que le asignemos.

1 - En la ubicación donde desees tu proyecto en la terminal, ejecuta lo siguiente, donde [name] lo reemplazamos por el nombre de la app correspondiente.

```
npx create-react-app [name]
```

Ejemplo: `npx create-react-app store-app`

2 - Ya que se termine de ejecutar el comando, navega dentro del proyecto con:

```
cd [name] // el nombre que hayas puesto
```

3 - Ahora vamos a instalar las dependencias de desarrollo para linteo de código:

```
npm i eslint-config-airbnb eslint-config-prettier  
eslint-plugin-jsx-a11y eslint-plugin-prettier prettier  
eslint-config-prettier typescript --save-dev
```

4 - Creas un archivo **.eslintrc** en raíz del proyecto con el siguiente contenido:

```
{  
  "extends": [  
    "react-app",  
    "airbnb",  
    "plugin:jsx-a11y/recommended",  
    "prettier",  
    "prettier/react",  
    "eslint-config-prettier"  
  ],  
  "plugins": ["jsx-a11y", "prettier"],  
  "rules": {  
    "react/jsx-filename-extension": [1, {"extensions": [".js",  
".jsx"]}]}  
  }  
}
```

Aquí puedes configurar las reglas como te hagan sentido. Yo usaré esta configuración para los proyectos.

4.1 - Si usas visual studio code, puedes instalar las extensiones de lint para que el editor aplique las reglas anteriores de manera automática.

5 - Ahora crearemos un archivo .prettierrc y dentro colocamos la siguiente configuración:

```
{
  "arrowParens": "avoid",
  "bracketSpacing": false,
  "htmlWhitespaceSensitivity": "css",
  "insertPragma": false,
  "jsxBracketSameLine": false,
  "jsxSingleQuote": false,
  "printWidth": 80,
  "proseWrap": "always",
  "quoteProps": "as-needed",
  "requirePragma": false,
  "semi": false,
  "singleQuote": true,
  "tabWidth": 2,
  "trailingComma": "all",
  "useTabs": false
}
```

5.1 - Si usas visual studio code, puedes instalar la extensión de prettier también.

Agrega esta configuración en el archivo settings.json del editor:

```
"editor.defaultFormatter": "esbenp.prettier-vscode",
"editor.formatOnSave": true,
... rest of the file
```

Con esto vamos a poder aplicar prettier al guardar un archivo y nos olvidamos de estar corriendo el comando manualmente.

6 - Agregamos los scripts de lint y prettier en el package.json

```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject",  
  "lint": "eslint --ignore-path .gitignore .",  
  "format": "prettier --ignore-path .gitignore --write  
\"**/*.+(js)\"",  
},
```

El `--ignore-path .gitignore` hace que eslint ignore los mismos archivos que contenga `.gitignore`.

7 - Ejecutamos el comando `npm run lint` en terminal para que valide nuestros archivos. Es posible que te salgan mensajes de que ciertas partes del código no cumplen con las reglas que hemos configurado de lint.

8 - Ejecutamos `npm run format` para que prettier aplique los cambios correspondientes a nuestro código.