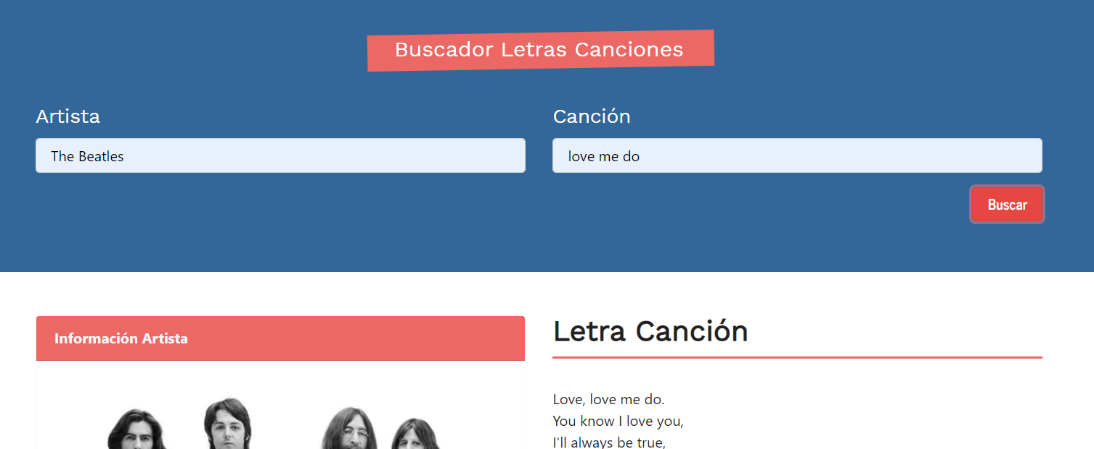
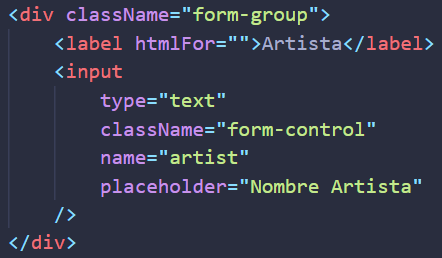
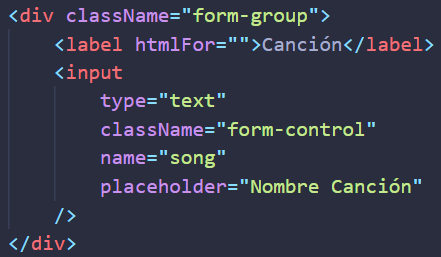
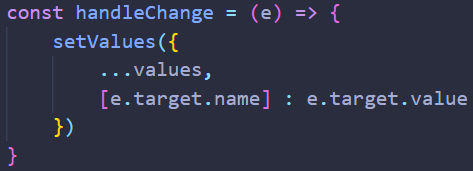
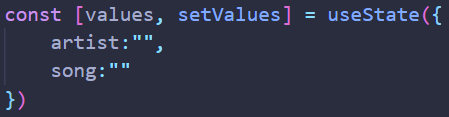
**Proyecto 10**

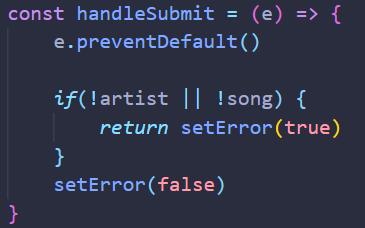
**Resultado final:**

**1**. Incluimos el **código** de **github** en el **index.css** y en el **<head>** del **index.html**

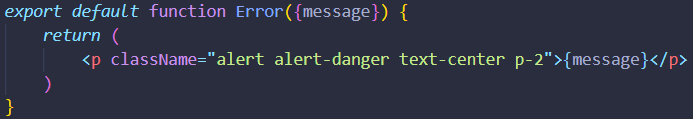
**2**. Instalamos axios: **npm i axios**

**3**. **Creamos** el componente **Form.js** que va constar de **dos <input>** de tipo texto.

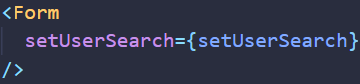
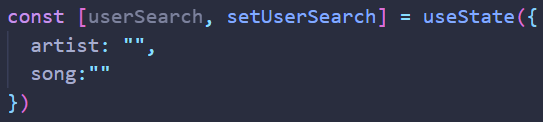
**4**. **Guardando los datos en el estado**: dentro de **Form.js** creamos un **nuevo estado** para **almacenar** los **datos** de los **<input>**. Bajo el evento **onChange** de cada input ejecutamos la función **handleChange** que se va a encargar de **guardar** los **valores** en el **estado**.

**5**. **Validando los campos**: **creamos** un **estado** para los **errores** y desestructuramos los valores de los <input>. Luego le pasamos la función **handleSubmit** al <**form**> bajo el evento **OnSubmit**.



**6**. **Mostrando los errores**: creamos el componente **Error.js** y lo **renderizamos** dentro de **Form.js en caso de que haya un error**.

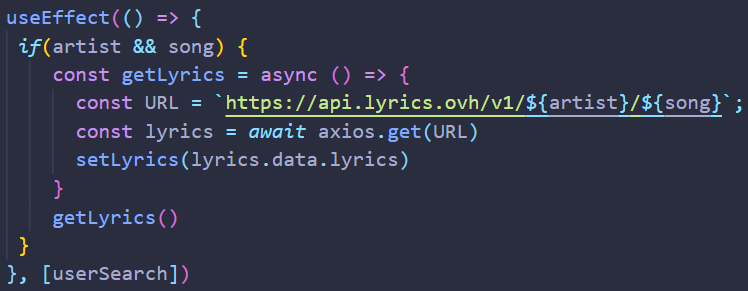


**7**. **Guardando datos validados**: dentro de **App.js** **creamos** un **estado** para **almacenar** los **datos** **validados** y se lo **pasamos** a **Form.js**. Allí luego de validar los campos guardamos los valores.

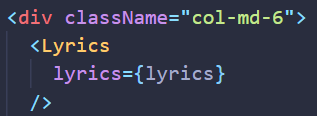
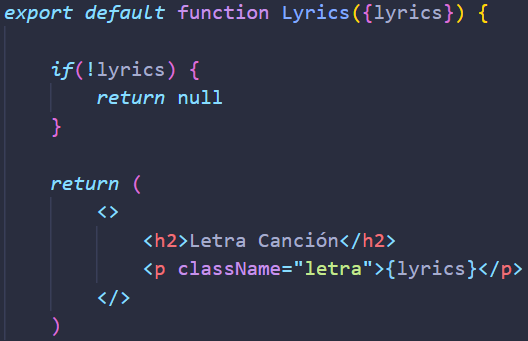


**8**. **Pidiendo la letra a la API**: dentro de **App.js** creamos un **nuevo estado** para **guardar** la **letra**, desestructuramos artista y canción y **luego** hacemos un **useEffect**. Requerimos **axios** y en caso de que el usuario haya hecho una búsqueda hacemos el **pedido a la API**. Luego **guardamos** la **información** en el **estado** que habíamos creado.





**9**. **Renderizando las letras**: **creamos** el componente **Lyrics.js** que va a **recibir** como props un **string** con la **letra de la canción**. Si el string **llega vacío** (es decir, si no hicimos el pedido aun) **evitamos** que el componente se **renderize**. Finalmente lo **importamos** dentro de **App.js** y le pasamos la letra.



**10**. **Pidiendo información del artista a la API**: primero debemos **crear** un nuevo **estado** para **almacenar** la **información**. Lo que debemos hacer es **consultar** las **dos APIs al mismo tiempo** y no una después de la otra. Por este motivo vamos a utilizar **Promise.all().** Primero debemos definir **entre corchetes las variables** que van a guardar lo que recibamos de la api, las **igualamos a Promise.all()** y le pasamos un **array** con todos **los llamados**. Finalmente **guardamos** **en los estados la información**.



**11**. **Imprimiendo información del artista**: **creamos** el componente **ArtistInfo.js** que va a **recibir** como props un **objeto** con la **información del artista**. **Evitamos** que el componente se **renderize** si el objeto **está vacío** y desestructuramos la información que nos parece relevante. Luego la incluimos en el JSX y finalmente la importamos en App.js.

