

# Actividad: Paso a Paso App Encuestas en React v2

## Objetivo

Los estudiantes desarrollarán una aplicación de encuestas en React, permitiéndoles poner en práctica y consolidar sus conocimientos sobre los fundamentos de React, JSX, componentes, estado, enrutamiento, formularios y estilos.

## Consigna

Desarrollar una aplicación que permita a los usuarios crear encuestas, ver una lista de las encuestas creadas y responder a ellas. La aplicación debe incluir:

- Una página de inicio con la lista de encuestas disponibles.
- Una página para crear nuevas encuestas.
- Una página para ver una encuesta en particular.

## Desarrollo Paso a Paso

1. Creamos un proyecto React utilizando create-react-app, para esto abrimos un terminal e ingresamos el siguiente comando:

```
npx create-react-app actividad_encuestas_v2
```

2. Ingresamos a la carpeta creada y luego abrimos VSCode en esa carpeta:

```
cd actividad_encuestas_v2  
code .
```

3. Agregar React Router para el Enrutamiento. Ejecutar el siguiente comando para instalar React Router en el proyecto:

```
npm install react-router-dom
```

4. Agregar Dependencias para Formularios. Para habilitar el uso de formularios en React, los alumnos pueden usar la dependencia react-hook-form. Ejecutar el siguiente comando para instalarlo:

```
npm install react-hook-form
```

5. Abrimos el terminal de VSCode y ejecutamos el siguiente comando para iniciar la aplicación y verificamos que todo funcione:

```
npm start
```

## Estructura de Carpetas

1. Organiza tu proyecto creando una estructura de carpetas similar a la siguiente:

```
actividad_encuestas_v2/  
├─ public/  
│   ├── index.html  
│   ├── manifest.json  
│   └─ favicon.ico  
├─ src/  
│   ├── components/  
│   │   ├── CrearEncuesta.jsx  
│   │   ├── Encuesta.jsx  
│   │   ├── Inicio.jsx  
│   │   ├── Menu.jsx  
│   │   └─ NotFound.jsx  
│   ├── data/  
│   │   └─ encuestas.json  
│   └─ styles/
```

```
| | └─ App.css
| └─ App.js
| └─ index.js
└─ package.json
└─ README.md
└─ .gitignore
```

## Datos de Encuestas

1. Dentro de la carpeta **src/data**, crea o utiliza un archivo JSON llamado **encuestas.json** para almacenar los datos de las encuestas.

Este archivo JSON contiene dos encuestas con títulos, descripciones y preguntas ficticias. Cada pregunta tiene un identificador único (id) y un conjunto de opciones de respuesta.

```
[
  {
    "id": 1,
    "titulo": "Encuesta de Satisfacción",
    "descripcion": "Por favor, tómese un momento para responder a nuestras preguntas de satisfacción.",
    "preguntas": [
      {
        "id": 1,
        "pregunta": "¿Cómo calificaría nuestro servicio?",
        "opciones": [
          { "id": 1, "texto": "Excelente" },
          { "id": 2, "texto": "Bueno" },
          { "id": 3, "texto": "Regular" },
          { "id": 4, "texto": "Malo" }
        ]
      }
    ]
  },
  {
    "id": 2,
    "pregunta": "¿Qué podemos mejorar?",
    "opciones": [
      { "id": 5, "texto": "Atención al cliente" },
      { "id": 6, "texto": "Tiempo de entrega" },
    ]
  }
]
```

```

        { "id": 7, "texto": "Variedad de productos" },
        { "id": 8, "texto": "Calidad de productos" }
    ]
}
]
},
{
    "id": 2,
    "titulo": "Encuesta de Productos",
    "descripcion": "Ayúdenos a mejorar nuestros productos
completando esta encuesta.",
    "preguntas": [
        {
            "id": 3,
            "pregunta": "¿Cuál es su producto favorito?",
            "opciones": [
                { "id": 9, "texto": "Producto A" },
                { "id": 10, "texto": "Producto B" },
                { "id": 11, "texto": "Producto C" },
                { "id": 12, "texto": "Producto D" }
            ]
        },
        {
            "id": 4,
            "pregunta": "¿Cómo podemos mejorar nuestro servicio al
cliente?",
            "opciones": [
                { "id": 13, "texto": "Mayor disponibilidad de atención"
},
                { "id": 14, "texto": "Más opciones de contacto" },
                { "id": 15, "texto": "Respuestas más rápidas" },
                { "id": 16, "texto": "Entrenamiento al personal" }
            ]
        }
    ]
}
]

```

## Creación de Componentes

### 1. Inicio.jsx (Página de Inicio con la lista de encuestas disponibles):

```
import React from "react";
import { Link } from "react-router-dom";

const Inicio = ({ listaEncuestas }) => {
  console.log(listaEncuestas);
  return (
    <div>
      <h1>Lista de Encuestas Disponibles</h1>
      {listaEncuestas.map((encuesta) => (
        <div className="encuesta-item-container">
          <div className="encuesta-item">
            <h2>{encuesta.titulo}</h2>
            <p>{encuesta.descripcion}</p>
            <Link to={`/encuesta/${encuesta.id}`}>Ver
encuesta</Link>
            <br />
          </div>
        </div>
      ))}
    </div>
  );
};

export default Inicio;
```

### 2. CrearEncuesta.jsx (Página para Crear Encuestas):

```
import React from 'react';
import { useForm } from 'react-hook-form';
import { useNavigate } from 'react-router-dom';

const CrearEncuesta = ({ agregarEncuesta }) => {
```

```
const { register, handleSubmit, formState: {errors} } =
useForm();
const navigate = useNavigate();
const onSubmit = (data) => {
  agregarEncuesta(data)
  navigate('/');
};

return (
  <div>
    <h1>Crear Nueva Encuesta</h1>
    <form onSubmit={handleSubmit(onSubmit)}>
      <label>Título:</label>
      <input
        type="text"
        id="titulo"
        name="titulo"
        {... register("titulo", {
          required: 'Este campo es obligatorio',
          maxLength: { value: 50, message: 'El título debe
tener menos de 50 caracteres' }
        })}
      />
      {errors.titulo && <p>{errors.titulo.message}</p>}

      <label>Descripción:</label>
      <textarea
        id="descripcion"
        name="descripcion"
        {... register("descripcion", {
          maxLength: { value: 200, message: 'La descripción
debe tener menos de 200 caracteres' }
        })}
      />
      {errors.descripcion &&
      <p>{errors.descripcion.message}</p>}
```

```
    /* Agrega aquí campos adicionales y sus validaciones  
    según sea necesario */
```

```
        <button type="submit">Guardar Encuesta</button>  
    </form>  
</div>  
);  
};  
  
export default CrearEncuesta;
```

### 3. Encuesta.jsx (Página para Ver Encuestas):

```
import { useParams, Link } from "react-router-dom";  
  
const Encuesta = ({ listaEncuestas, responderEncuesta }) => {  
    const { id } = useParams(); // Obtiene el ID de la encuesta  
    desde la URL  
    console.log(listaEncuestas);  
    const encuesta = listaEncuestas.find((enc) => enc.id ===  
    parseInt(id));  
    console.log("Preguntas: " + encuesta.preguntas);  
    return (  
        <div>  
            <div className="encuesta-item-container">  
                <div className="encuesta-item">  
                    <h2>{encuesta.titulo}</h2>  
                    <p>{encuesta.descripcion}</p>  
                    <br />  
                </div>  
            </div>  
            <div className="encuesta-item-container">  
                <div className="encuesta-item">  
                    <h2>Preguntas</h2>  
                    <p>  
                        {!encuesta.preguntas && <p>Sin preguntas  
definidas.</p>}
```

```

    {encuesta.preguntas &&
      encuesta.preguntas.map((pregunta) => (
        <div key={pregunta.id}>
          <p>{pregunta.pregunta}</p>
          <ol>
            {pregunta.opciones.map((opcion) => (
              <div key={opcion.id}>
                <label>
                  <li>{opcion.texto}</li>
                </label>
              </div>
            ))}
          </ol>
        </div>
      ))}
    </p>
    <br />
  </div>
</div>

  <Link to="/">Volver</Link>
</div>
);
};

export default Encuesta;

```

## Enrutamiento

1. Crear el componente de Menú de Navegación:

Crea un nuevo componente llamado **Menu.jsx** en la carpeta de componentes de tu proyecto.

```

import React from 'react';
import { Link } from 'react-router-dom';

const Menu = () => {

```



```
return (
  <nav>
    <ul>
      <li>
        <Link to="/">Inicio</Link>
      </li>
      <li>
        <Link to="/encuesta/crear">Crear Encuesta</Link>
      </li>
    </ul>
  </nav>
);
};

export default Menu;
```

2. Crear el componente de NotFound para Navegación de páginas no definidas:

Creas un nuevo componente llamado NotFound.jsx en la carpeta de componentes de tu proyecto.

```
const NotFound = () => (
  <div>
    <h2>Página no encontrada</h2>
    <p>Lo sentimos, la página que estás buscando no existe.</p>
  </div>
);

export default NotFound;
```

3. Agregar las rutas necesarias en el componente principal App.js:

```
import React, { useState } from "react";
import "../styles/App.css";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Inicio from "../components/Inicio";
import CrearEncuesta from "../components/CrearEncuesta";
import Encuesta from "../components/Encuesta";
import Menu from "../components/Menu"; // Importa el componente
Menu
import NotFound from "../components/NotFound"; // Importa el
componente Menu
import encuestas from "../data/encuestas.json"
function App() {

  const [listaEncuestas, setListaEncuestas] =
  useState(encuestas);

  const agregarEncuesta = (nuevaEncuesta) => {
    nuevaEncuesta.id = listaEncuestas.length + 1
    setListaEncuestas([...listaEncuestas, nuevaEncuesta]);
  };

  const responderEncuesta = (id, respuestas) => {
    const encuesta = listaEncuestas.find(enc => enc.id ===
    parseInt(id));
    encuesta.respuestas = [respuestas];

  };

  return (
    <BrowserRouter>
      <Menu /> { /* Agrega el menú de navegación */ }
      <Routes>
        <Route path="/" element={<Inicio
listaEncuestas={listaEncuestas} />} /> />
```

```

    <Route path="/encuesta/crear" element={<CrearEncuesta
agregarEncuesta={agregarEncuesta} />} />
    <Route path="/encuesta/:id" element={<Encuesta
listaEncuestas={listaEncuestas}
responderEncuesta={responderEncuesta} />} />

    <Route path="*" element={<NotFound />} />
  </Routes>
</BrowserRouter>
);
}

export default App;

```

## Estilos

1. Utiliza el archivo app.css en la carpeta styles para estilizar tus componentes:

```

/* Estilos para el menú de navegación */
nav {
  background-color: #007bff;
  color: white;
  padding: 10px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
}

ul {
  list-style: none;
  display: flex;
  justify-content: center;
  padding: 0;
}

li {
  margin: 0 20px;
}

```

```
a {
  text-decoration: none;
  color: 007bff;
  font-weight: bold;
  transition: color 0.3s;
}

a:hover {
  color: #0056b3;
}

/* Estilos para los formularios */
form {
  background-color: #fff;
  padding: 20px;
  border: 1px solid #ddd;
  border-radius: 5px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
  max-width: 600px;
  margin: 0 auto;
}

label {
  display: block;
  margin-bottom: 10px;
  font-weight: bold;
}

input[type="text"],
textarea {
  width: 95%;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  margin-bottom: 15px;
  transition: border-color 0.3s;
}
```

```
input[type="text"]:focus,
textarea:focus {
  border-color: #007bff;
}

button {
  background-color: #007bff;
  color: white;
  border: none;
  padding: 10px 20px;
  cursor: pointer;
  font-weight: bold;
  transition: background-color 0.3s;
}

button:hover {
  background-color: #0056b3;
}

/* Estilos para el listado de encuestas */

.encuesta-item-container {
  background-color: #f9f9f9;
  border: 1px solid #ddd;
  border-radius: 10px; /* Agregamos bordes redondeados */
  padding: 20px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
  display: block;
  transition: transform 0.3s; /* Agregamos una transición suave */
  margin: 10px 0;
}

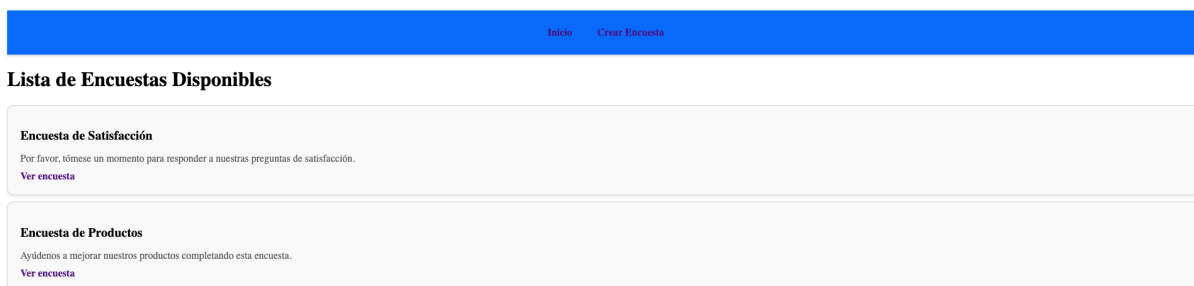
.encuesta-item-container:hover {
  transform: translateY(-5px); /* Efecto de elevación al pasar el mouse */
}
```

```
.encuesta-item h2 {
  font-size: 20px;
  margin-bottom: 15px;
}

.encuesta-item p {
  color: #666;
  margin-bottom: 10px;
}
```

## Probamos de la Aplicación

En este paso, vamos a probar todas las características que hemos agregado a nuestra aplicación de encuestas. Asegúrate de haber seguido todos los pasos anteriores para configurar la aplicación correctamente. Comprobar que la aplicación funcione como se espera y que todas las características estén en funcionamiento.



Realizar las siguientes pruebas en tu aplicación:

Página de Inicio:

- Abre la página de inicio y verifica que se muestre una lista de encuestas disponibles.
- Haz clic en una encuesta para ir a la página de vista de encuesta.

Creación de Encuestas:

- Ve a la página de creación de encuestas haciendo clic en "Crear Nueva Encuesta" en la página de inicio.
- Rellena el formulario de creación de encuestas con datos válidos y guarda una encuesta.
- Asegúrate de que la encuesta se guarde correctamente y se muestre en la lista de encuestas disponibles en la página de inicio.

Vista de Encuesta:

- Desde la página de inicio, haz clic en una encuesta existente para verla.

#### Validaciones:

- Intenta enviar formularios incompletos o con datos incorrectos en la página de creación de encuestas y verifica que se muestren mensajes de error adecuados.
- Asegúrate de que las validaciones funcionen correctamente para garantizar que los campos requeridos y las restricciones de longitud se apliquen correctamente.

#### Estilos:

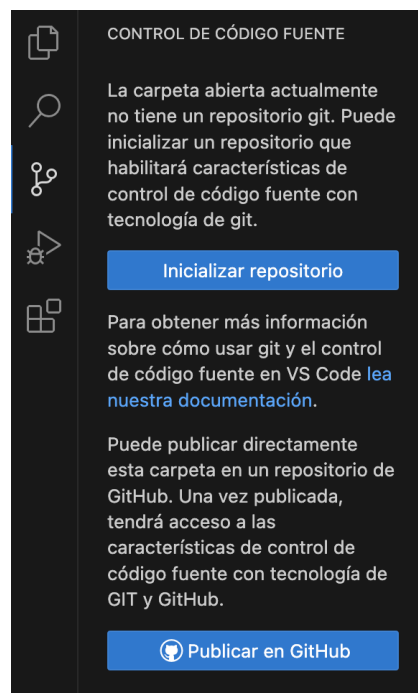
- Verifica que los estilos generales definidos en App.css se apliquen a toda la aplicación.
- Comprueba que los botones y otros elementos de la interfaz tengan el aspecto deseado.

#### Enrutamiento:

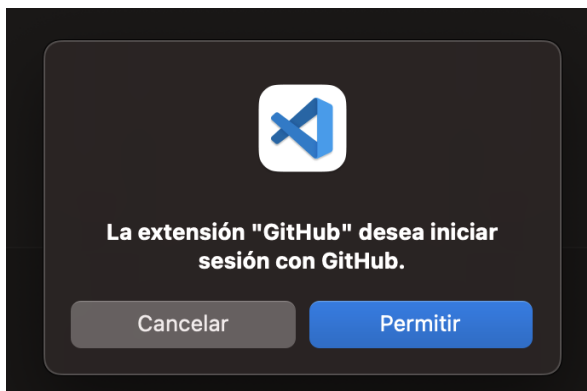
- Asegúrate de que las rutas y el enrutamiento funcionen correctamente. Navega entre las diferentes páginas de la aplicación y verifica que la URL y el contenido se actualicen correctamente.

## Entrega del Trabajo

1. Ahora volvemos a VSCode y vamos a llevar a GitHub el código fuente generado usando la herramienta de git de VSCode. Hacemos click en la opción "Publicar en GitHub":

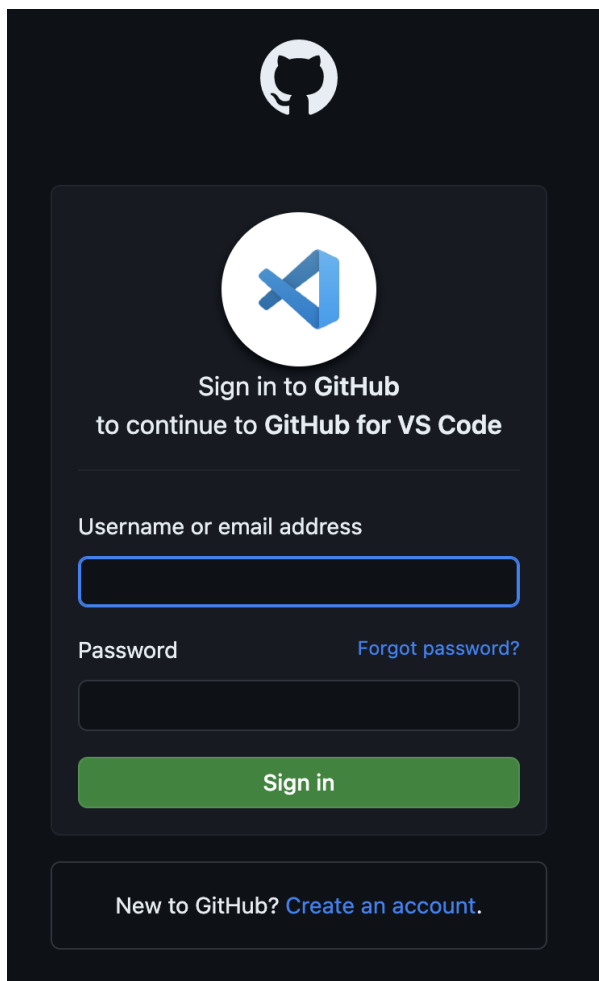


2. VSCode muestra este mensaje para que le darles permisos a iniciar sesión en GitHub a través de un navegador:



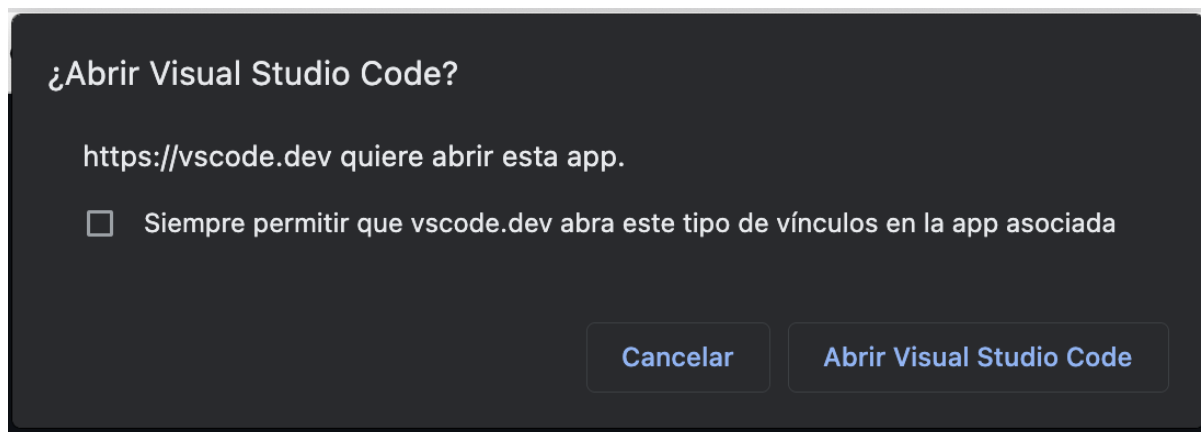


3. VSCode nos lleva al navegador web que tengamos por defecto y nos pide que ingresemos los datos de nuestra cuenta GitHub (el que no tenga cuenta se crea una nueva con su correo electrónico):



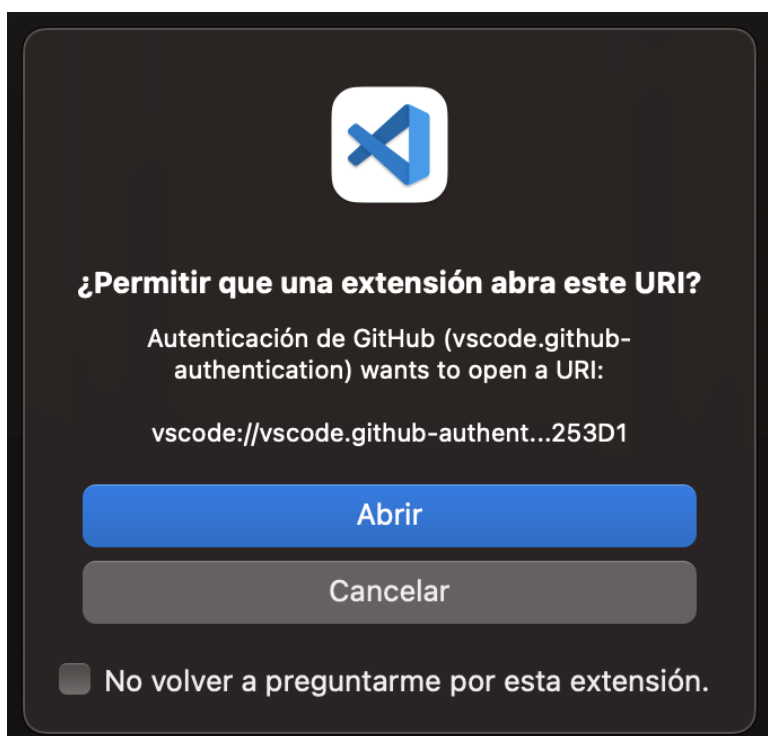
The image shows a dark-themed GitHub login interface. At the top is the GitHub Octocat logo. Below it is a white circle containing the VS Code logo. The text reads: "Sign in to GitHub to continue to GitHub for VS Code". There are two input fields: "Username or email address" and "Password". A "Forgot password?" link is next to the password field. A green "Sign in" button is below the fields. At the bottom, it says "New to GitHub? [Create an account.](#)"

4. El navegador solicita que permitamos abrir VSCode, seleccionamos la opción "Abrir Visual Studio Code"

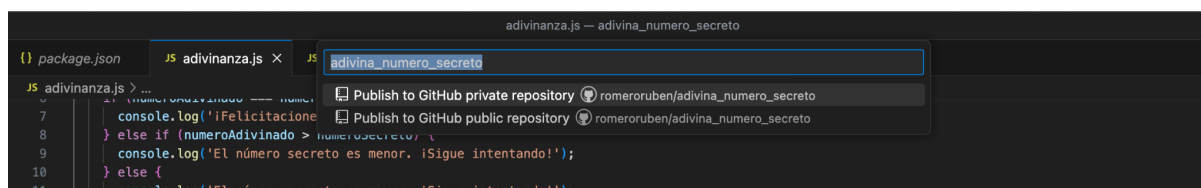


The image shows a system dialog box with a dark background. The title is "¿Abrir Visual Studio Code?". The text says: "https://vscode.dev quiere abrir esta app." Below this is a checkbox with the text "Siempre permitir que vscode.dev abra este tipo de vínculos en la app asociada". At the bottom are two buttons: "Cancelar" and "Abrir Visual Studio Code".

5. VSCode nos pide permisos para abrir una url, hacemos click en la opción “Abrir”:

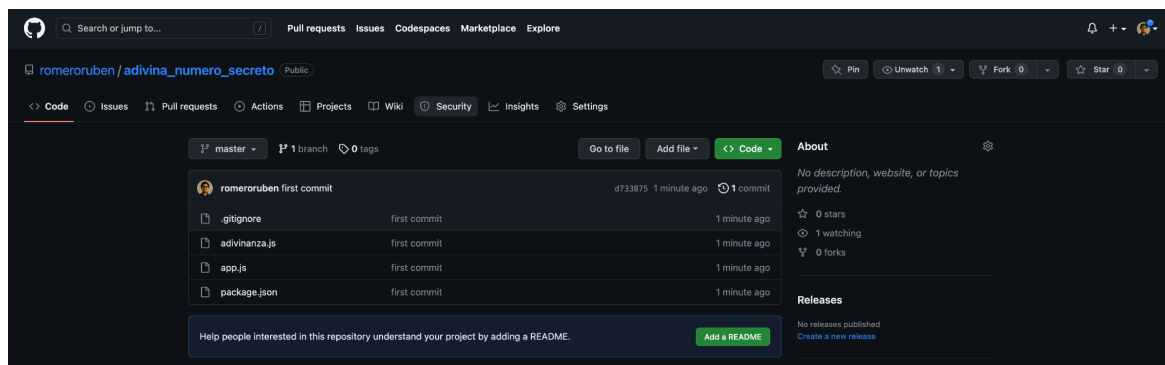


6. Ahora debemos seleccionar que tipo de repositorio queremos crear, seleccionar “Publish to GitHub **public** repository”, esto significa que vamos a publicar nuestro código en forma pública.



7. A continuación seleccionamos los archivos y carpetas que queremos “versionar” y llevar a GitHub.
8. Esperamos unos minutos hasta que en VSCode podemos ver la siguiente notificación (abajo a la derecha), hacemos click en “Abrir en GitHub”.

9. Se abre un navegador con el repositorio de GitHub disponible para que cualquiera lo pueda acceder:



10. **IMPORTANTE!** Para dar por finalizada la actividad compartir la url de GitHub en la actividad del aula virtual en <https://uve.frc.utn.edu.ar/>.