**Creación de proyecto**

npx create-react-app proyecto

cd proyecto  
npm install react-router-dom

npm install axios

npm install –g json-server

**Iniciar proyecto**

npm start  
json-server –watch db.json –port 3001

**Pasos para trabajar en equipo con Git y GitHub**

**1. Clonar el repositorio (solo la primera vez)**

Si aún no tienen el proyecto clonarlo de la siguiente forma:

git clone https://github.com/Joansinho/PROYECTO\_FORMACION.git

Se descarga los recursos del proyecto, más no las dependencias. **SE DEBE INSTALAR EL PROYECTO NUEVAMENTE DESDE EL EQUIPO  
  
MOVER LAS CARPETAS A LA CARPETA DONDE ESTE NODE\_MODULES (si es la primera vez creando el proyecto)**

**2. Crear una nueva rama para trabajar**

Antes de comenzar a hacer cambios en el proyecto, cada uno debe crear una nueva rama. Esto permite trabajar en paralelo sin interferir con el código principal. Es recomendable nombrar la rama según la funcionalidad en la que se esté trabajando.

Para crear y moverte a una nueva rama:

git checkout -b nombre-de-la-rama

Por ejemplo, si estás trabajando en una nueva funcionalidad de login, puedes crear una rama llamada `feature-login`:

git checkout -b feature-login

Ahora estarás trabajando en esta rama, y cualquier cambio que hagas se mantendrá separado de la rama principal.

**3. Hacer cambios y commits regularmente**

A medida que hagas cambios en el código, es importante guardarlos con commits regulares. Sigue estos pasos:

- Añadir los cambios que has hecho al "staging area":

git add .

- Guardar los cambios con un commit:

git commit -m "Descripción del cambio realizado"

Asegúrate de hacer descripciones claras para que ambos entiendan los cambios realizados.

**4. Subir la rama al repositorio remoto**

Una vez que hayas hecho varios commits y quieras guardar tu trabajo en GitHub, puedes subir tu rama:

- Subir tu rama al repositorio remoto:

git push origin nombre-de-la-rama

Si estás trabajando en `feature-login`, sería:

git push origin feature-login

**5. Hacer un Pull Request (PR)**

Después de subir tu rama, debes crear un **Pull Request** (PR) en GitHub para fusionar tus cambios en la rama principal (`main` o `master`).

- Ve al repositorio en GitHub y selecciona tu rama.

- Haz clic en "Compare & pull request".

- Revisa los cambios y crea el PR.

- Ambos pueden revisar el PR antes de aprobar la fusión (merge).

**6. Fusionar la rama con la principal**

Para fusionar tu trabajo con la rama principal, uno de ustedes debe hacer lo siguiente (el ultimo en trabajar en el proyecto o **SIEMPRE** que se termine y este testeada la funcionalidad en la que se trabajó):

- Cambiarse a la rama principal (`main` o `master`):

git checkout main

- Traer los últimos cambios del repositorio remoto:

git pull origin main

- Luego, fusionar la rama que contiene los cambios:

git merge nombre-de-la-rama

Si tu rama se llama `feature-login`:

git merge feature-login

Si no hay conflictos, la fusión se realizará automáticamente. Si hay conflictos, Git te pedirá que los resuelvas manualmente.

**8. Actualizar el código antes de empezar a trabajar (importante)**

Antes de hacer más cambios, asegúrense de traer siempre los últimos cambios de la rama principal para evitar problemas:

- Cambiarse a la rama principal:

git checkout main

- Traer los últimos cambios:

git pull origin main

Ahora pueden crear una nueva rama y seguir trabajando en nuevas funcionalidades.