

«Talento Tech»

Desarrollo Web 3

Clase 05





Clase N° 5 | Práctica y Github

Temario:

- Realizaremos un trabajo integrador que permita consolidar los conocimientos adquiridos hasta el momento.



Lista de Tareas en Github 🤖🐱

¡Te damos la bienvenida a la Clase #5 de DW3! En esta oportunidad estaremos cubriendo de manera práctica la clase para darle una conclusión a esta primera parte del curso. Los desafíos #1 y #2 reflejan que hemos hecho lo siguiente:

- **Configuración Inicial**

Crea una nueva aplicación Vite con el template de React.
Instala Bootstrap y Bootswatch como dependencias.

- **Estilos con Bootstrap y Bootswatch**

Configura la importación de estilos en la aplicación Vite para incluir Bootstrap y Bootswatch.

- **Componente de Lista de Tareas**

Crea un componente TaskList.jsx en la carpeta src/components para mostrar la lista de tareas.

Utiliza el estado (useState) para manejar la lista de tareas.

Simula la carga de datos desde una API utilizando fetch y useEffect.

- **Integración en la Aplicación Principal**

Actualiza el componente principal App.jsx para utilizar el componente TaskList.

- **Ejecución y Verificación**

Ejecutar la aplicación Vite y verificar que la lista de tareas se muestra correctamente en el navegador.

Asegurarse de que los estilos de Bootstrap y Bootswatch estén aplicados.

📌 Este **ejercicio** combina, en su totalidad, los conceptos vistos en las clases 1, 2, 3 y 4, incluyendo la creación de componentes, el manejo de estado con useState, el uso de efectos con useEffect, la integración de estilos con Bootstrap y el consumo de una API para obtener datos simulados de tareas. Se valorará la organización del código, el correcto uso de los hooks de React y la integración exitosa de estilos utilizando Bootstrap y Bootswatch.

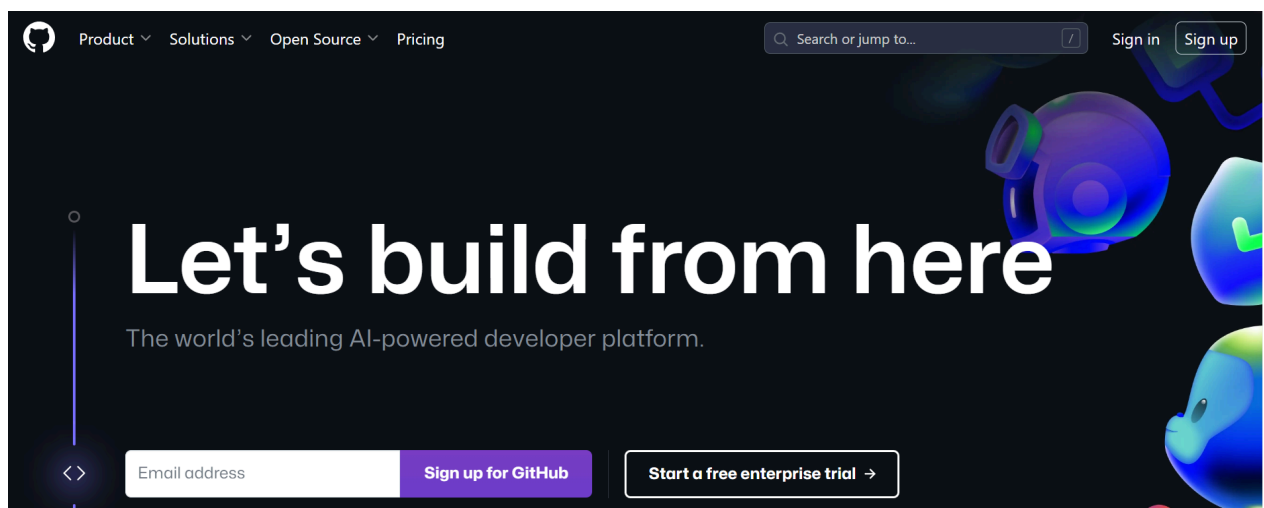
Ahora, ¿cómo seguimos? Vamos a subir nuestro proyecto en GitHub.

GitHub

GitHub es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git. Permite a los desarrolladores trabajar juntos en proyectos de software, llevar un historial de cambios, realizar un seguimiento de problemas y contribuir al código de otros. GitHub proporciona herramientas para gestionar proyectos de manera eficiente, realizar solicitudes de extracción, revisar código, y alojar repositorios de código fuente de forma pública o privada. Es ampliamente utilizado en la comunidad de desarrollo de software para facilitar la colaboración y el control de versiones en proyectos.

¿Cómo crear una cuenta en GitHub? 😊

1. Accede al Sitio Web de GitHub: <https://github.com/>
2. Inicia el Proceso de Registro
3. Haz clic en "Sign up" (Registrarse)





4. Completa el Formulario:

Welcome to GitHub!
Let's begin the adventure

Enter your email*
✓ [redacted]

Create a password*
✓ [redacted]

Enter a username*
✓ [redacted]

Email preferences
☐ Receive occasional product updates and announcements.

Continue

5. Completa el Captcha:

Verify your account

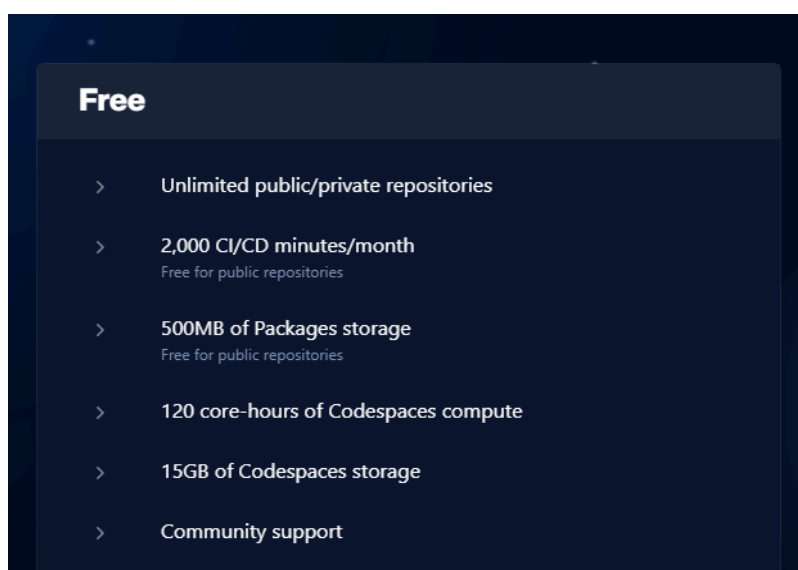
Utiliza las flechas para girar el objeto para que mire en la dirección de la mano. (1 de 1)

Emparejar este ángulo

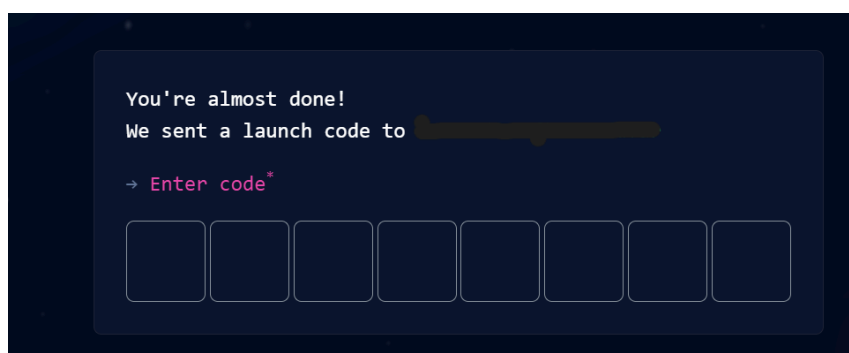
Enviar



6. Selecciona el Plan Gratuito:

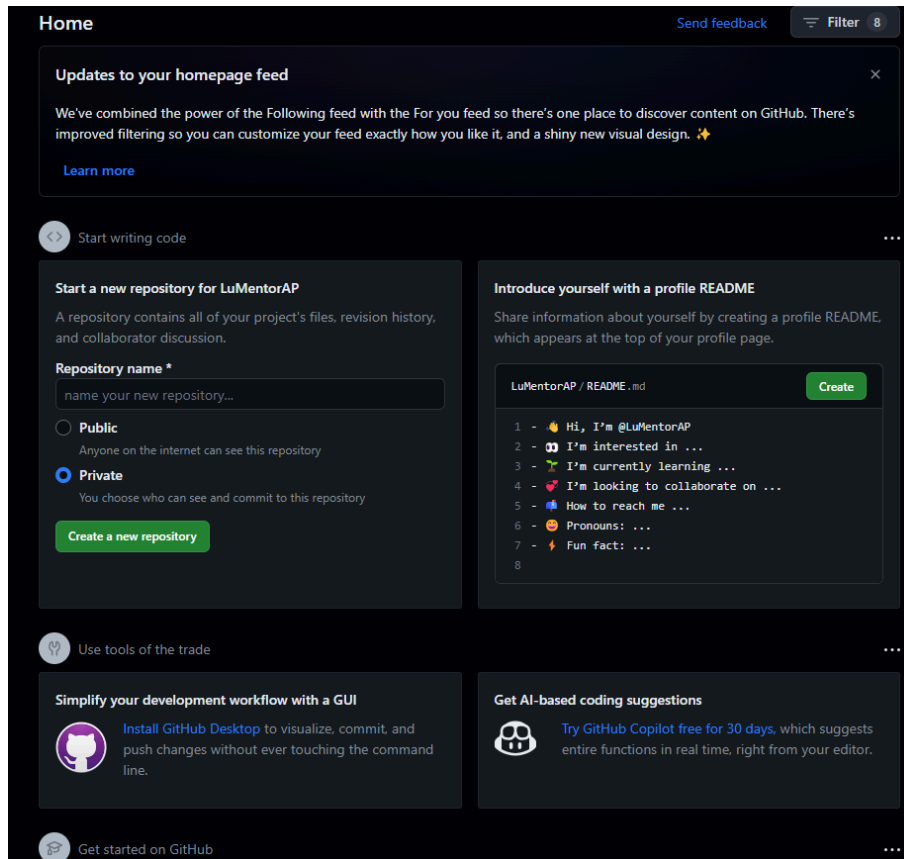


7. Verifica Tu Cuenta:



8. Abre el correo electrónico que recibas y sigue las instrucciones para verificar tu cuenta.

¡Listo! Ahora tienes una cuenta en GitHub. Personaliza tu perfil y comienza a explorar proyectos.



¿Cómo subir nuestro proyecto a Github?

Desde la terminal de VSCode vamos seguir algunos pasos básicos:

1. Inicializar un Repositorio Local:

- Abre la terminal en VSCode.
- Navega al directorio de tu proyecto usando el comando ``cd ruta/de/tu/proyecto``.
- Inicializa un nuevo repositorio Git ejecutando ``git init``.

2. Agregar Archivos al Repositorio:

- Agrega todos los archivos al repositorio con ``git add .``. Este comando preparará todos los archivos para el commit.

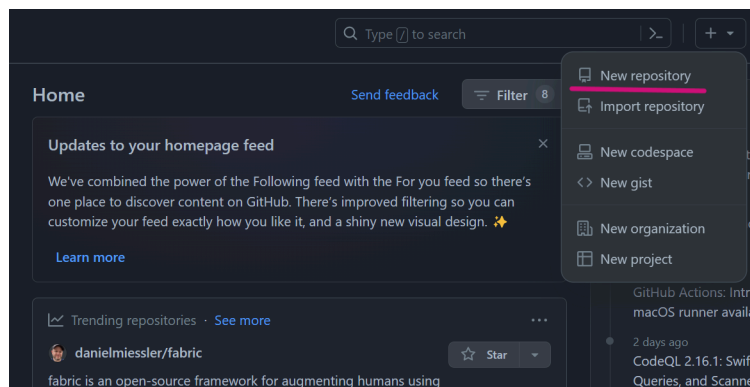
3. Realizar el Commit Inicial:



- Realiza el commit inicial con un mensaje descriptivo usando ``git commit -m "Primer commit"`.`

4. Crear un Repositorio en GitHub:

- Ve a [GitHub](<https://github.com/>) y accede a tu cuenta.
- En la esquina superior derecha, haz clic en el signo '+' y selecciona "Nuevo repositorio".



- Sigue las instrucciones para configurar el nuevo repositorio en GitHub. No inicialices con un archivo README, ya que ya tienes un repositorio local.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Repository template
No template

Start your repository with a template repository's contents.

Owner * / Repository name *
lista-tareas
lista-tareas is available

Great repository names are short and memorable. Need inspiration? How about [ubiquitous-spoon](#)?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
☒ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

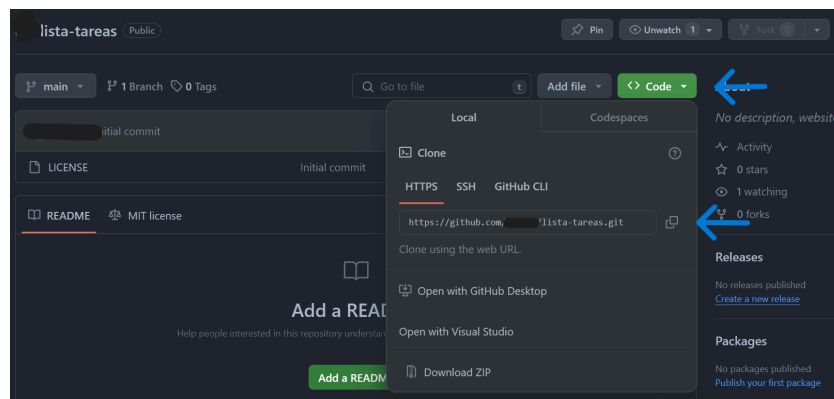
Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: MIT License
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

5. Conectar el Repositorio Local con GitHub:

- Copia el enlace proporcionado después de crear el repositorio en GitHub.



- En la terminal de VSCode, ejecuta el siguiente comando, reemplazando ``<enlace-del-repositorio>`` con el enlace que copiaste:

```
git remote add origin <enlace-del-repositorio>
```



6. Subir los Archivos a GitHub:

- Sube los archivos al repositorio en GitHub con el siguiente comando:

```
git push -u origin master
```

Si estás en una rama diferente a `master`, reemplaza `master` con el nombre de tu rama.

¡Listo! Ahora tu proyecto web está en GitHub. Puedes realizar cambios en tu código, hacer commits locales y luego subir esos cambios a GitHub usando **`git push`**.

¡Genial! Ahora ya sabemos cómo subir nuestros proyectos a Github. Principalmente, esto nos ayudará a poder aligerar el peso de memoria de nuestros PCs. Una vez que tengamos un proyecto en Github, podremos hacer lo querremos con nuestro proyecto local. Podemos continuar trabajando y seguir subiendo versión a versión hasta llegar al final, o mismo, podemos eliminarlo si sabemos que no lo estaremos usando por un buen tiempo.

¡Eso es todo por hoy! Nos vemos en la **Clase #6** 🙌





Buenos Aires
aprende
Agencia de Actividades para el Futuro

BA Buenos
Aires
Ciudad