

Sistemas Electrónicos Programables

Introducción a Github

¿Qué son Git y Github?

Git es un sistema de control de versiones, es decir, un *software* capaz de almacenar y gestionar los cambios hechos a archivos, generalmente código. Esto permite volver a versiones anteriores de un programa o implementar nuevas funcionalidades sin tener que preocuparse de que el código deje de funcionar permanentemente.

GitHub es una plataforma online que almacena código utilizando un sistema de control de versiones, permitiendo tener un respaldo de este. Además, varias personas pueden trabajar en un mismo código sin preocuparse de sobrescribir lo programado por otro.

En resumen, Git es un software que almacena y controla las distintas versiones de un código, mientras que GitHub es un servicio online que almacena repositorios de Git. Git corre en tu computador, mientras que GitHub es una **página web**.

A continuación aprenderás lo esencial sobre la línea de comandos, Git y GitHub. Además, se presentarán los pasos para crear los repositorios donde subirás tus laboratorios.

Tutorial de la línea de comandos

Una manera de trabajar con GitHub es la línea de comandos, una aplicación capaz de realizar tareas (manipular archivos, por ejemplo) con comandos de texto. Es necesario que tengas los conocimientos básicos sobre esta. El **tutorial de Codecademy** es una muy buena herramienta para aprender lo esencial. Es recomendable realizar como mínimo las dos lecciones marcadas en verde.

| Overview | <u>Syllabus</u> |
|---|-----------------|
| 1 Navigating the File System ^ | |
| 4/1 Navigation Interactive Lesson | |
| 2/3 Bicycle World Freeform Project | |
| 2/3 Daily Buzz Freeform Project | |
| 1/1 Navigating the File System Multiple Choice Quiz | |
| 1/1 Command Line Interface Setup Article | |
| 2 Viewing and Changing the File System ^ | |
| 4/1 Manipulation Interactive Lesson | |
| 2/3 Artusi Freeform Project | |
| 1/1 Viewing and Changing the File System Multiple Choice Quiz | |
| 3 Redirecting Input and Output v | |
| 4 Configuring the Environment v | |

Figura 1: Contenidos del tutorial

Importante para usuarios de Windows: La línea de comandos presentada en el tutorial es la de UNIX, utilizada por macOS y Linux. Windows usa una distinta. Para utilizar una línea de comandos que contiene los comandos enseñados anteriormente puedes descargar **Git Bash**.

Tutorial de Git y GitHub

Ahora que saben utilizar la línea de comandos, pueden aprender sobre Git y GitHub. Leer el tutorial de [HubSpot](#) es una muy buena opción. No es necesario que se manejen con los *branches* y *pull requests*, pero son herramientas útiles al momento de trabajar junto a otras personas y con códigos complejos. Sí es necesario que dominen los comandos `add`, `commit` y `pull`, junto con saber clonar y crear repositorios.

Cómo subir los laboratorios a GitHub

Tienes acceso a varios repositorios de la organización IEE2463 en GitHub. El primero es [classroom](#). Aquí se sube información y material, como laboratorios y guías. Además, están las [issues](#), donde se sube información importante y puedes publicar tus dudas para que tus compañeros y ayudantes te apoyen.

Aparte de [classroom](#), tendrás un repositorio privado para cada laboratorio. Deberás subir el código que hiciste para que quede un respaldo de este. A continuación se presenta una explicación paso a paso del proceso que deberás realizar.

1. Si no tienes una cuenta de GitHub, debes [crear una](#). Puedes usar el correo y usuario que quieras.
2. En [classroom](#) hay una tabla que contiene los enlaces para crear el repositorio de cada laboratorio. Al ingresar aparecerá la imagen siguiente. Debes escoger tu número de alumno. Este paso solo será necesario para crear tu primer repositorio.

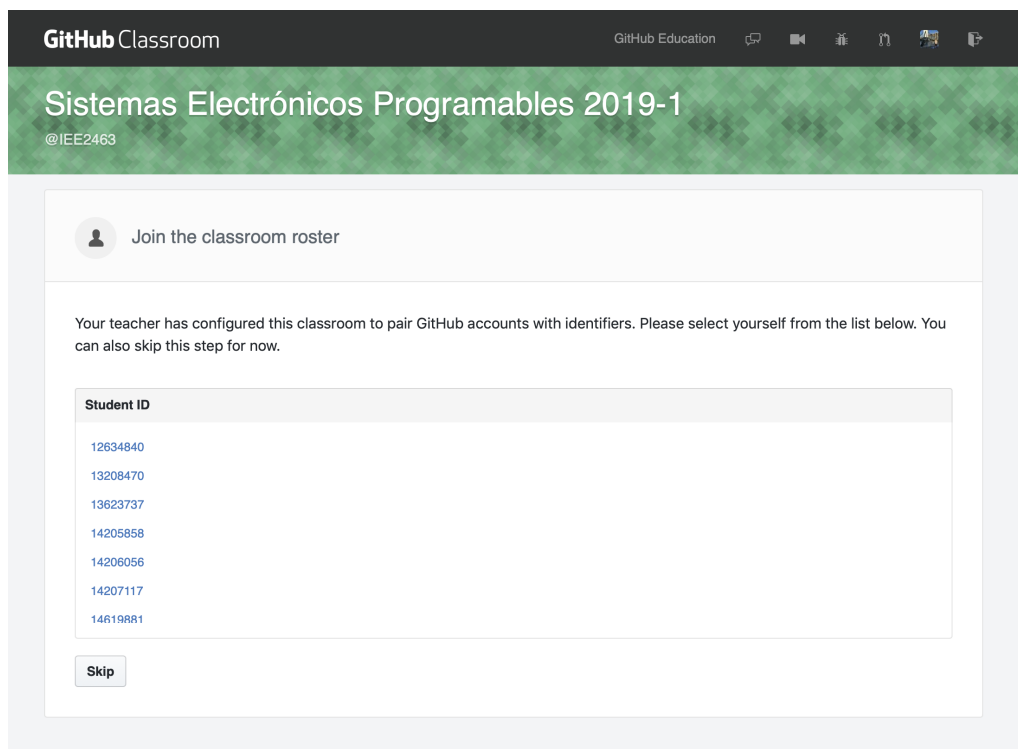


Figura 2

3. Ahora simplemente debes presionar el botón verde. Esto aparecerá cada vez que crees el repositorio para un laboratorio.

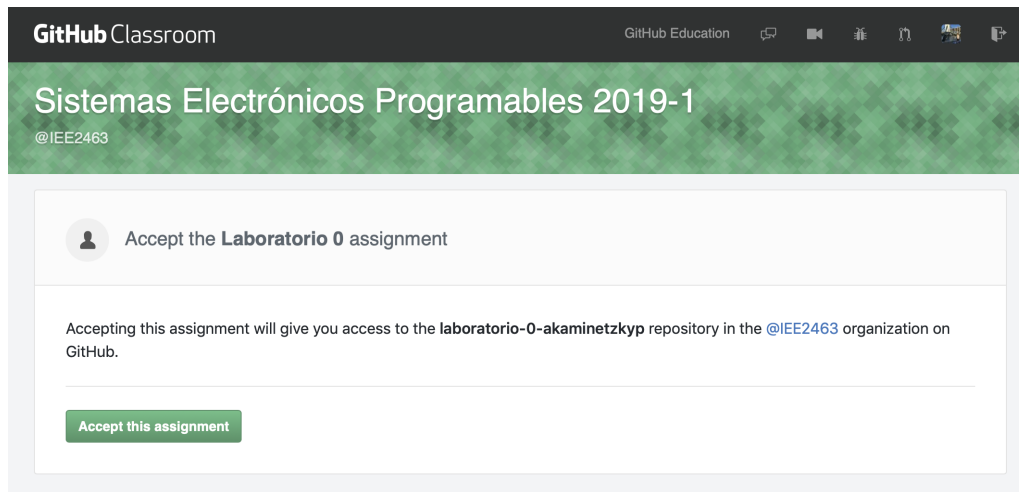


Figura 3

4. ¡Tu repositorio está listo! Ahora simplemente entra al enlace que aparece al final.

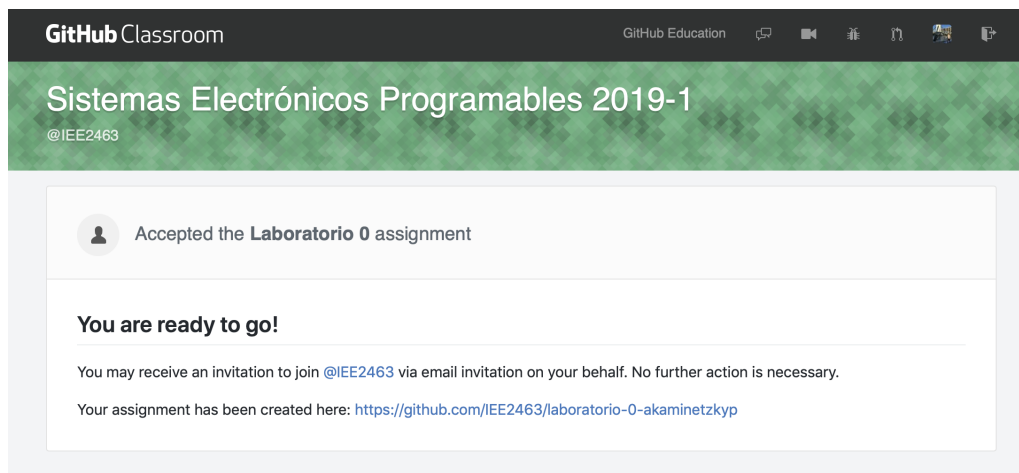


Figura 4

5. Así se ve un repositorio completamente vacío. Ahora debes enlazar este repositorio con una carpeta en tu computador. La alternativa más simple es copiar el enlace marcado con verde y clonarlo a tu computador ingresando `git clone tu_enlace` en la línea de comandos.

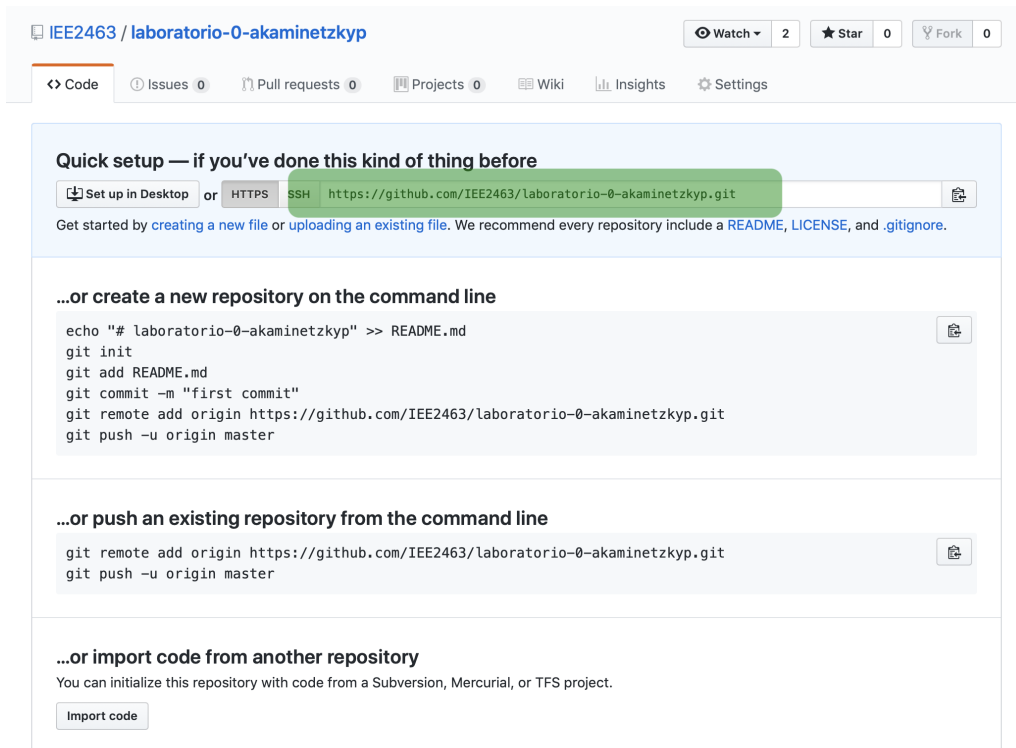


Figura 5

6. ¡Todo listo! Ahora puedes subir tus archivos a GitHub con los comandos aprendidos anteriormente (`add`, `commit` y `push`).

Tips

- Es recomendable que crees una carpeta en tu computador y que clones los repositorios dentro de esta para mantener el orden.
- Intenta realizar *commits* regularmente. Así se almacenaran versiones anteriores de tu código y podrás volver atrás en caso que algo ya no funcione o elimines algo por equivocación.
- Trata de ser descriptivo con los mensajes de *commit*. Si quisieras volver atrás, sería difícil encontrar la versión que quieres si todos tus *commit* se llaman "adgshdfaja".

Fuentes

- [Github](#)