

Esta clase va a ser

- grabada

Certificados oficialmente por



PedidosYa

CODERHOUSE

Clase 11. DESARROLLO WEB

GitHub

Certificados oficialmente por



PedidosYa

CODERHOUSE

Objetivos de la clase

- Aprender qué es un repositorio en Github.
- Crear un repositorio para nuestro proyecto.
- Subir el proyecto al repositorio usando los comandos de Git.

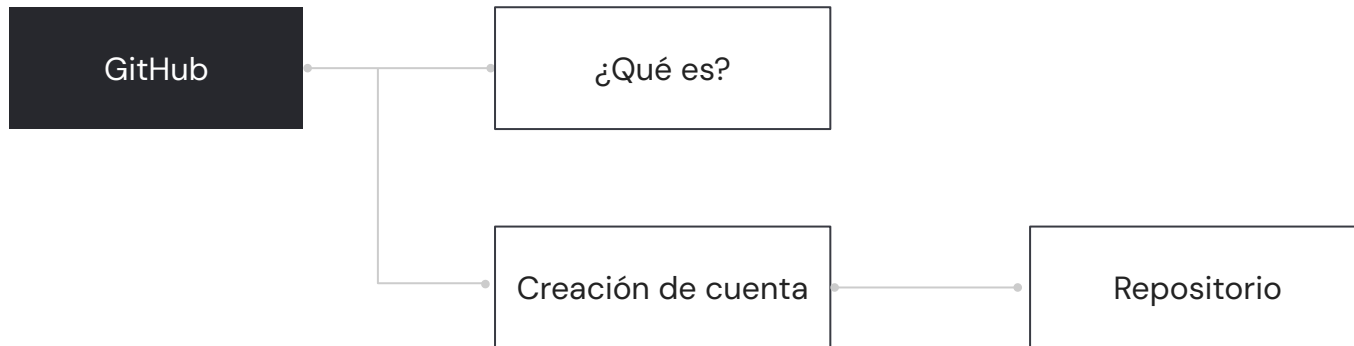
Glosario

Git: es un sistema de control de versiones gratuito y de código abierto, diseñado para manejar desde pequeños a grandes proyectos de manera rápida y eficaz. Se entiende como control de versiones a todas las herramientas que nos permiten hacer modificaciones en nuestro proyecto. Este sistema registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo.

Repositorio: es un espacio centralizado donde se almacena, organiza, mantiene y difunde información.

- **Git Init:** este comando se usa para crear un nuevo repositorio en Git.
- **Git Add:** se utiliza para agregar el o los archivos al Staging Area.
- **Git Commit:** una vez que nuestros archivos están en el Staging Area debemos pasarlos a nuestro repositorio local y para eso debemos usar el git commit, que es el comando que nos va a permitir comprometer nuestros archivos

MAPA DE CONCEPTOS



Temario

10

GIT

- ✓ Git
- ✓ Comandos básicos de la terminal
- ✓ Repositorios

11

GitHub

- ✓ [GitHub](#)
- ✓ [Creación de cuenta](#)
- ✓ [Creación de repositorio](#)

11

Frameworks+Bootstrap

- ✓ Librerías y Frameworks
- ✓ Bootstrap
- ✓ Estructura de Bootstrap

GitHub

¿Dudas de la clase
pasada?

¿Qué es GitHub?

Por ahora todo lo que venía ocurriendo en Git, sucedía de manera local, no necesitábamos nada de internet para guardar nuestros commits y nuestro repositorio.

Ahora, si queremos compartir nuestro trabajo con otras personas (colegas de proyecto, clientes, etc), ¡para eso utilizamos Github!

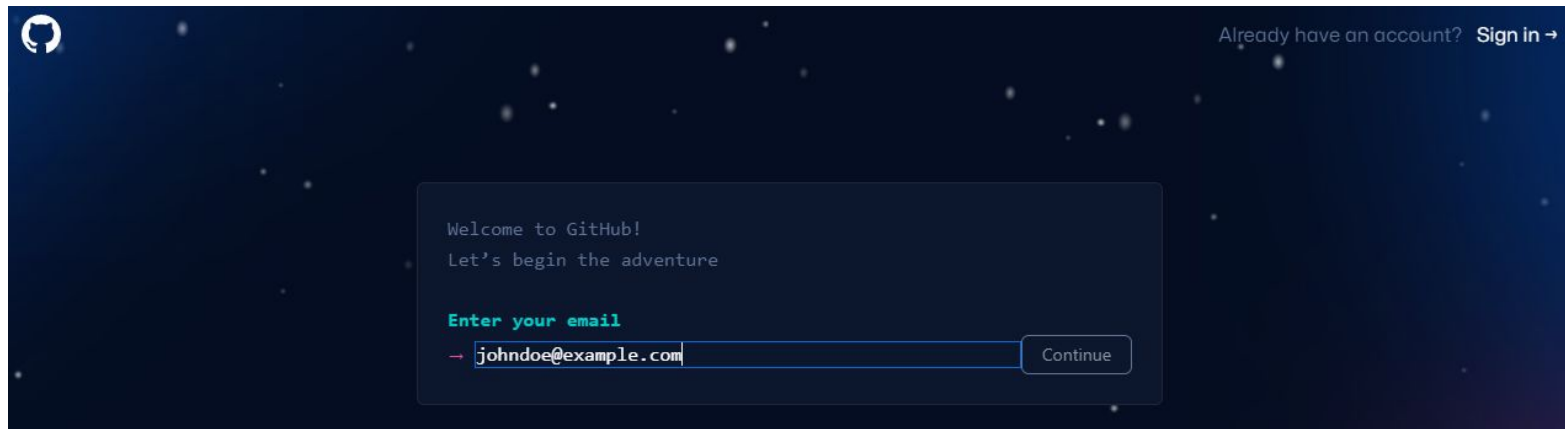
Github es una especie de red social de programadores. Con este sitio podemos subir nuestros proyectos y lograr que otras personas colaboren en ellos.



Creación de cuenta

Creando cuenta en GitHub

Ve a <https://github.com/> (está en inglés). Haz clic en "sign up".



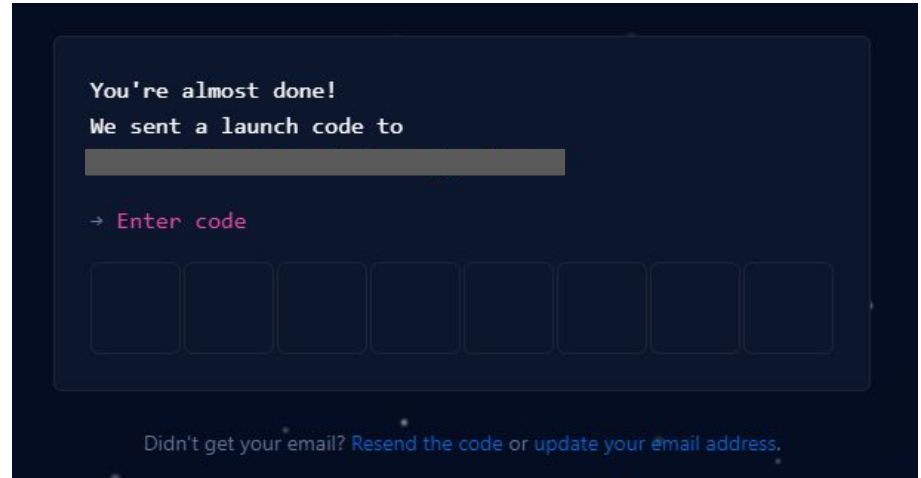
Completa tus datos

Deberás ir completando una serie de datos:

- **Email:** debe ser el mismo que usaste para Git.
- **Password:** una contraseña que puedas recordar luego.
- **Username:** será el nombre con el que te encuentren los demás usuarios. Elige uno que sea fácil de recordar, te identifique, y que puedas compartir con tus colegas y clientes.
- *Would you like to receive product updates and announcements via email?* Puedes elegir si quieres recibir (o no) información sobre actualizaciones y publicidad de GitHub por email.

Verificación

- Antes de poder completar el registro, deberás realizar una verificación para demostrar que no eres un robot.
- Una vez realizada, haz click en **create account**. Deberías ver una pantalla como la de la imagen.
- Ingresa el código de 8 dígitos que fue enviado a tu email, y habrás completado la creación de tu cuenta.

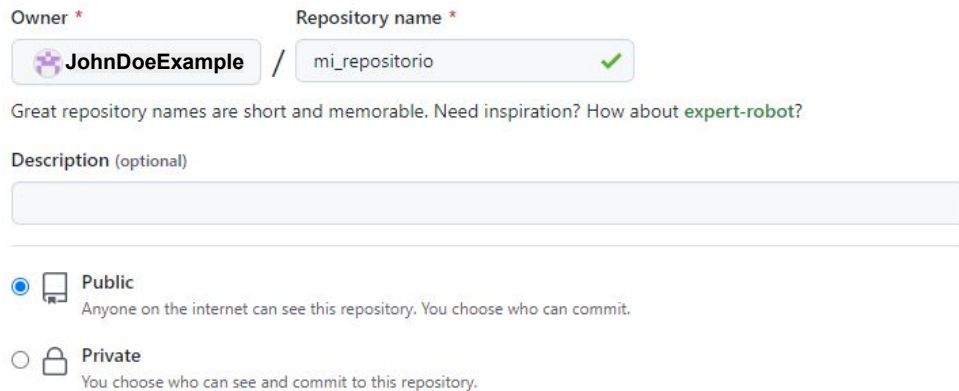


¿Preguntas?

#CoderTip: Ingresa al [siguiente link](#) y revisa el material interactivo que preparamos sobre **Preguntas Frecuentes**, estamos seguros de que allí encontrarás algunas respuestas.

Creación de repositorio

Crear un repositorio remoto





The screenshot shows the GitHub interface for creating a new repository. At the top, there are two input fields: 'Owner' with a red asterisk and 'Repository name' with a red asterisk. The 'Owner' field contains a user icon and the text 'JohnDoeExample'. The 'Repository name' field contains 'mi_repositorio' and a green checkmark. Below these fields is a hint: 'Great repository names are short and memorable. Need inspiration? How about [expert-robot?](#)'. Underneath is a 'Description (optional)' label followed by a large, empty text input box. At the bottom, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option is accompanied by a globe icon and the text 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is accompanied by a lock icon and the text 'You choose who can see and commit to this repository.'

Una vez dentro de tu cuenta, podrás ver en la parte superior una serie de pestañas. La opción “Repositories” te permite ver tus repositorios creados o, en este caso, **crear tu primer repositorio**.

Por ejemplo, podría ser llamado **“mi_repositorio”**, para que pruebes con los archivos que trabajaste en el desafío de GIT.

Configuración del repo

- ☐  **Public**
Anyone can see this repository. You choose who can commit.
- ☒  **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

- ☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▼

Add a license: **None** ▼



Create repository

Elegimos “**público**” o “**privado**”. Si bien con privado limitamos el acceso a cualquier persona, no nos permitirá mostrar nuestro código como página web, por lo que elegimos “**público**”. Luego hacemos clic en “**create repository**”.



Break

¡10 minutos y volvemos!

Subida al repositorio

Git + GitHub

Llegó el momento de alojar la información del repositorio local de tu computadora, en el nuevo repositorio remoto que acabamos de crear. Para lograrlo, una vez que completaste la creación, GitHub te proporcionará una serie de comandos.

```
john@MyShopSolutions: ~$ git remote add origin  
https://github.com/JohnDoeExample/mi_repositorio.git
```

```
/*Este comando renombra la rama master a main, es el nombre  
que usan los repositorios modernos*/
```

```
john@MyShopSolutions: ~$ git branch -M main
```

```
/*Este comando envía todos los cambios que hayan sido  
commiteados, al repositorio remoto*/
```

```
john@MyShopSolutions: ~$ git push -u origin main
```

Paso a paso

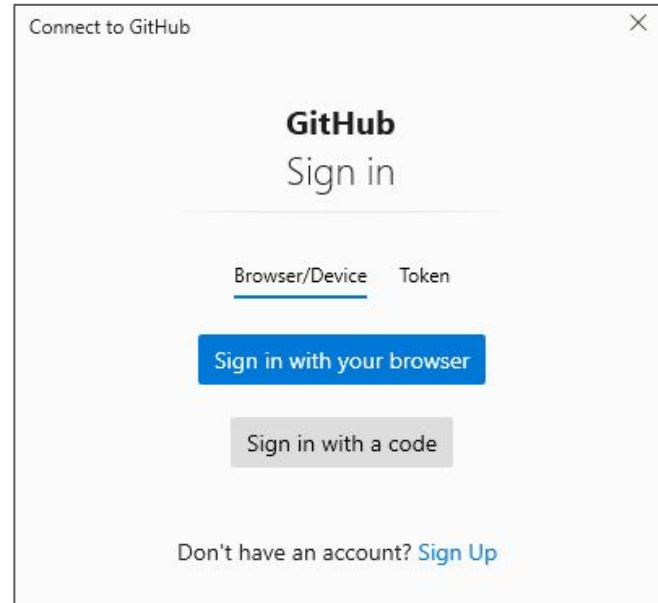
Para subir nuestro proyecto al repositorio vamos a nuestra terminal, y nos ubicamos en el proyecto elegido. Copiaremos las siguientes líneas para realizar el “push” de los archivos a nuestro servidor en GitHub.

```
/* Paso 1: Debes ubicarte en tu repositorio local*/  
john@MyShopSolutions :~$ cd Documents/Proyectos_Coder/mi_repositorio  
  
/* Paso 2: Indica cuál será tu nuevo repositorio remoto. Debes reemplazar  
miuser por tu nombre de usuario elegido en github */  
john@MyShopSolutions:~/Documents/Proyectos_Coder/mi_repositorio$ git remote add  
origin https://github.com/miuser/mi_repositorio.git  
  
/*Paso 3: Este comando renombra la rama master a main, es el nombre que usan  
los repositorios modernos*/  
john@MyShopSolutions: ~$ git branch -M main
```

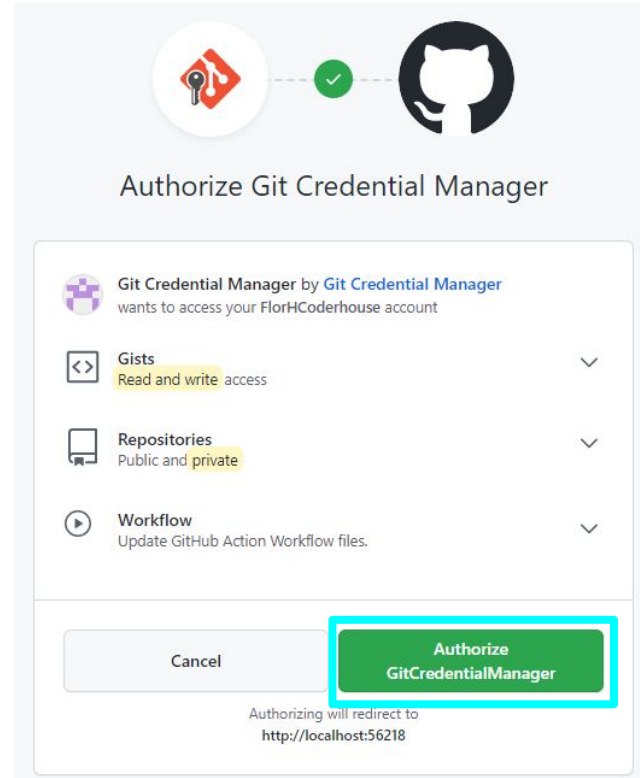
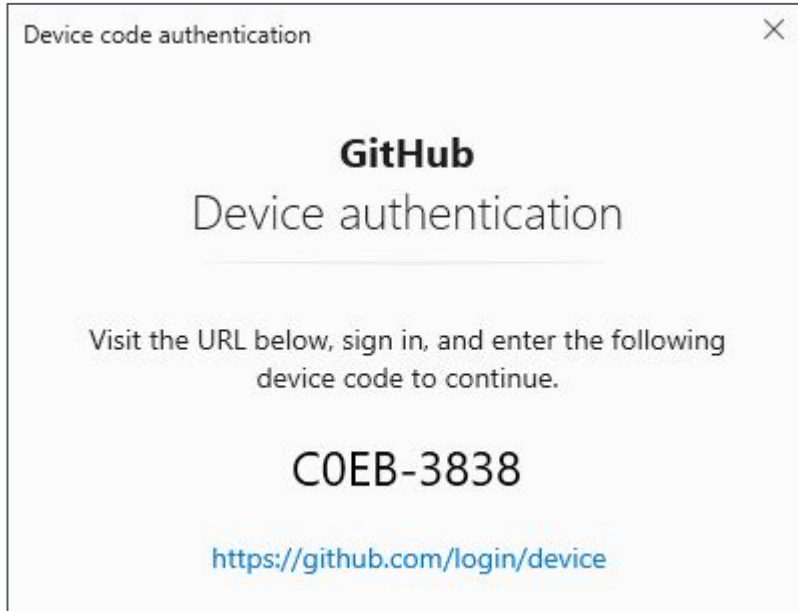
Paso a paso

```
/* Paso 4: Este comando envía todos  
los cambios que hayan sido  
commiteados, al repositorio remoto*/  
john@MyShopSolutions:~/Documents/Pro  
yectos_Coder/mi_repositorio$ git  
push -u origin main
```

/*En este punto, se abre una ventana que nos permite loguearnos y autenticarnos desde el navegador, o a través de un código. Ambas opciones terminan en un botón de autorizar a Git Credential Manager*/



Paso a paso



Paso a paso

```
/* Paso 4: Este comando envía todos los cambios que hayan sido commiteados, al repositorio remoto*/
```

```
john@MyShopSolutions:~/Documents/Proyectos_Coder/mi_repositorio$ git push -u origin main
```

```
/*Una vez que completamos la autenticación, veremos el siguiente resultado*/
```

```
Enumerating objects: 3, done.
```

```
Counting objects: 100% (3/3), done.
```

```
Delta compression using up to 8 threads
```

```
Compressing objects: 100% (2/2), done.
```

```
Writing objects: 100% (3/3), 407 bytes | 407.00 KiB/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

```
To https://github.com/miuser/mi_repositorio.git
```

```
* [new branch]    main -> main
```

```
Branch 'main' set up to track 'origin/main'.
```


Archivos en GitHub

Ahora, si vuelves a la página de GitHub y revisas tu repositorio remoto, podrás ver el archivo `index.html` que acabas de pushear.

main ▾


1 branch


0 tags

Go to file

Add file ▾

<> Code ▾

 **florenciahendel** Primer archivo del repositorio 96e7e59 2 hours ago ⌚ 1 commit

 index.html

Primer archivo del repositorio

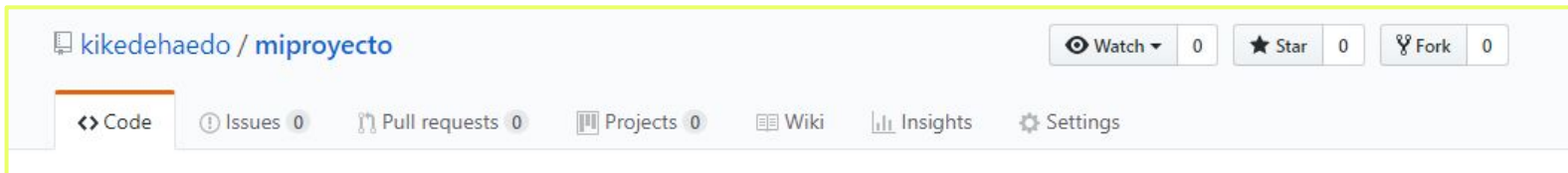
2 hours ago

Help people interested in this repository understand your project by adding a README.

Add a README

Más propiedades en GitHub

Como plataforma colaborativa, GitHub **ofrece a sus usuarios una gran cantidad de funcionalidades para la gestión de proyectos**, todas apoyadas por la comunidad. Por esta razón, a lo mejor dentro de un año tenga agregadas nuevas características que le permitan a los usuarios un mejor desenvolvimiento en el desarrollo de código.



GitHub pages

GitHub Pages

Una de las funcionalidades de GitHub, es la posibilidad de publicar nuestros sitios web y poder verlos online. Veamos el paso a paso

1

Ve a los "Settings" de nuestro repositorio.

2

Activa tu GitHub page.

3

Selecciona qué rama quieres usar.

4

Guarda los cambios y GitHub cumplirá la función básica de cualquier otro Hosting.

Importante: el proyecto sólo debe ser de archivos estáticos, ningún archivo que requiera de BackEnd especial.

Options

[Collaborators](#)[Branches](#)[Webhooks](#)[Notifications](#)[Integrations & services](#)[Deploy keys](#)[Secrets](#)[Actions](#)

Moderation

[Interaction limits](#)

Settings

Repository name

prueba-git

Rename

☐ Template repository

Template repositories let users generate new repositories with the same directory structure and files. Indicate if tutorial-hash/prueba-git can be used as a template for creating other repositories.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾

Select source

master branch

Use the master branch for GitHub Pages.

master branch /docs folder

Use only the /docs folder for GitHub Pages.

GitHub Pages

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https://isaine.github.io/mi_repositorio/.

Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▾

Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

Se auto recarga la página, y a continuación haz scroll nuevamente hasta “GitHub Pages”.

¡Encontrarás la dirección web para poder acceder a tu sitio!



Hands on lab

En esta instancia de la clase **aplicaremos nuestros conocimientos sobre GitHub** en un ejercicio práctico

¿De qué manera?

El profesor demostrará cómo hacerlo y tú lo puedes ir replicando en tu computadora. Si surgen dudas las puedes compartir para resolverlas en conjunto de la mano de los tutores.

Tiempo estimado: **15 minutos**

Repositorio en GitHub

¿Cómo lo hacemos?

- ✓ A partir de la rama main, crea una nueva rama en tu proyecto principal
- ✓ Agrega un componente de Bootstrap en esta nueva rama.
- ✓ Mergea con la rama principal.
- ✓ Pushea todos los cambios al repositorio remoto



De ahora en más, continuarás el trabajo en el repositorio del proyecto directamente en GitHub, y utilizarás el Public URL para la presentación del mismo.

¡Atención!

La próxima clase comenzaremos a trabajar con SASS.
Para eso, necesitas tener instalado Node JS en tu computadora.
A continuación veremos el proceso de descarga e instalación.



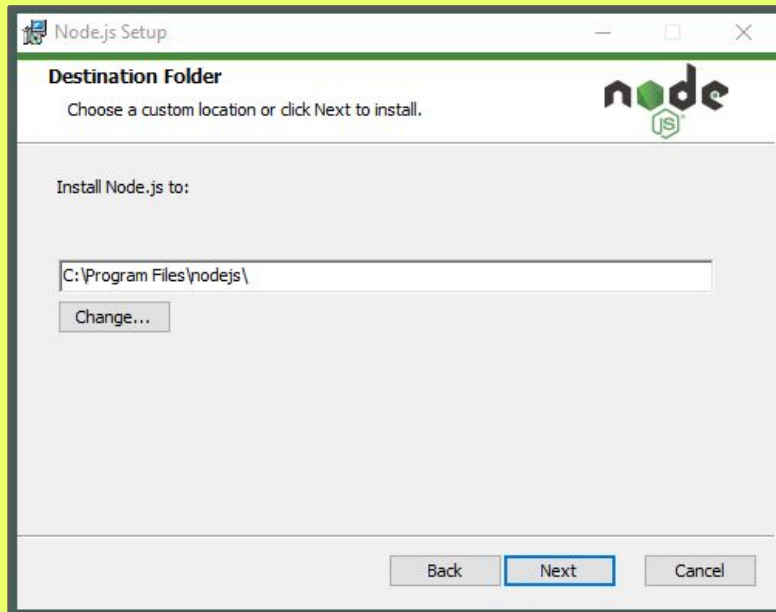
Instalación de Node

El primer paso es dirigirse a la [página oficial de Node](#). A continuación vamos a descargar el archivo instalador.
En este momento, la versión más nueva y estable de Node es la **LTS 18.12**, pero siempre debemos prestar atención a descargar la que nos sugiera la web oficial.



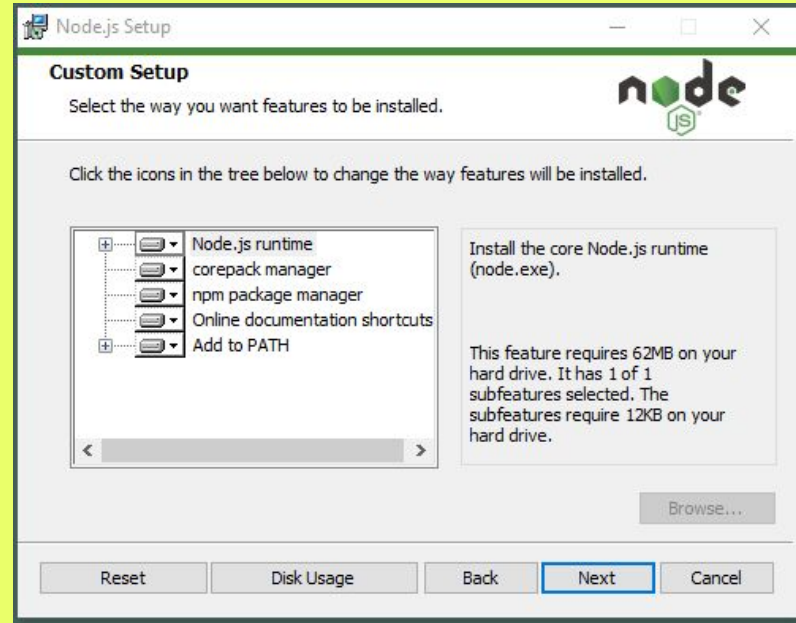
Instalación de Node

Ahora debes ejecutar el archivo descargado, y una vez que aceptas el contrato, elegir la carpeta donde ubicar los archivos de Node. Si tienes dudas, te recomendamos dejar la opción que viene predeterminada.



Instalación de Node

En la siguiente pantalla, dejaremos todas las opciones predeterminadas.



Instalación de Node

En esta pantalla, chequeamos la casilla para que, al completarse la instalación de Node, el asistente comience a añadir de manera automática las herramientas adicionales que se requieren para el trabajo.

Some npm modules need to be compiled from C/C++ when installing. If you want to be able to install such modules, some tools (Python and Visual Studio Build Tools) need to be installed.

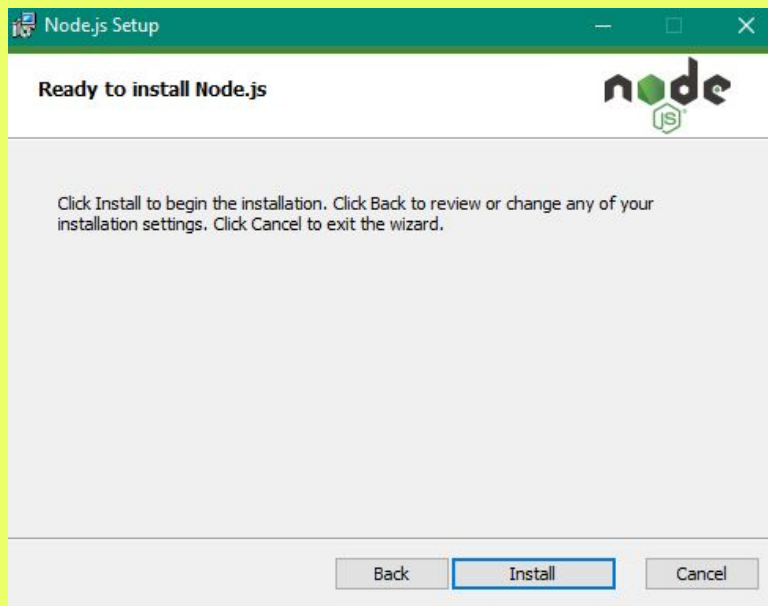


Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.

Alternatively, follow the instructions at <https://github.com/nodejs/node-gyp#on-windows> to install the dependencies yourself.

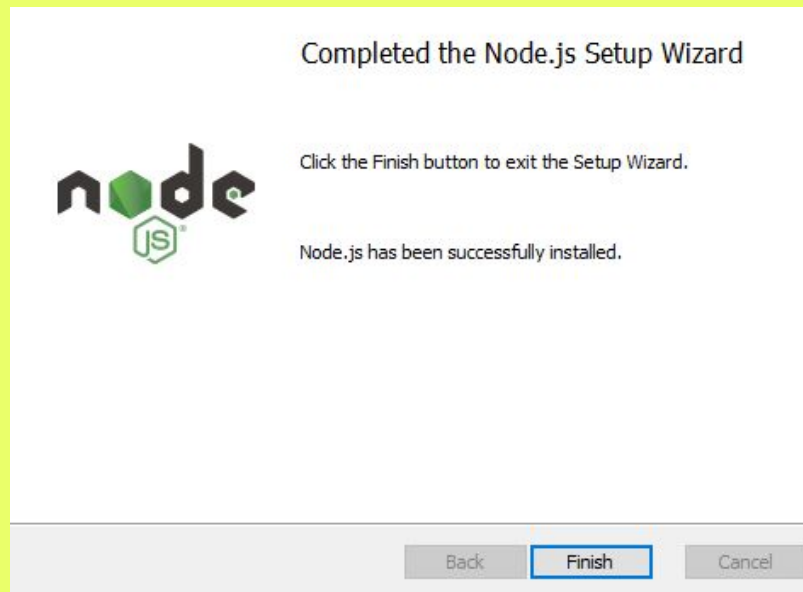
Instalación de Node

Al hacer click en "Install" el asistente da inicio a la instalación de Node.



Instalación de Node

Presionamos "finish" para que termine la instalación de Node y se agreguen las herramientas adicionales.



Instalación de Node

En la consola que se despliega, presionamos cualquier tecla para continuar, y repetimos hasta que se cierra la consola negra, y se abre una azul.

```
CA: Administrador: Install Additional Tools for Node.js
=====
Tools for Node.js Native Modules Installation Script
=====

This script will install Python and the Visual Studio Build Tools, necessary
to compile Node.js native modules. Note that Chocolatey and required Windows
updates will also be installed.

This will require about 3 GiB of free disk space, plus any space necessary to
install Windows updates. This will take a while to run.

Please close all open programs for the duration of the installation. If the
installation fails, please ensure Windows is fully updated, reboot your
computer and try to run this again. This script can be found in the
Start menu under Node.js.

You can close this window to stop now. Detailed instructions to install these
tools manually are available at https://github.com/nodejs/node-gyp#on-windows

Presione una tecla para continuar . . .
```


Instalación de Node

Una vez finalizada la instalación de todas las herramientas adicionales, cerramos la consola azul, pulsando la tecla "enter".

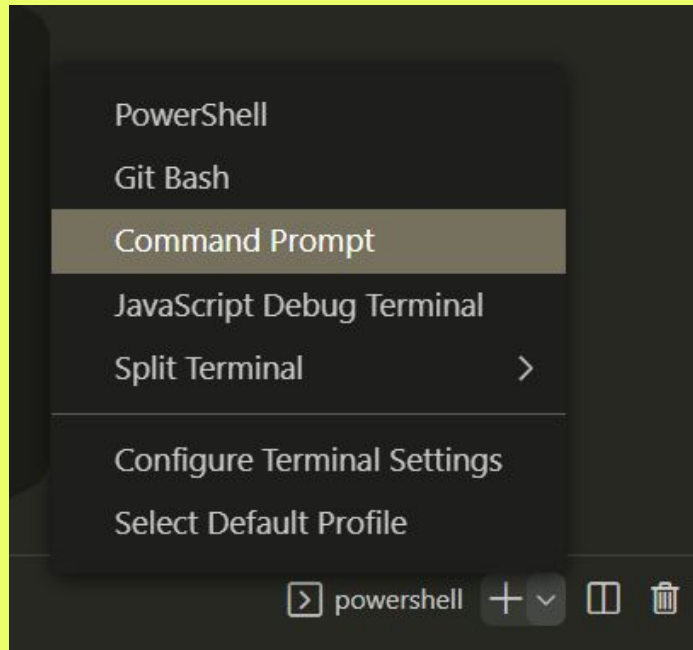
```
python v3.11.0 [Approved]
python package files upgrade completed. Performing other installation steps.
The upgrade of python was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\python'
visualstudio2019-workload-vctools v1.0.1 is the latest version available based on your source(s).

Chocolatey upgraded 2/3 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
Type ENTER to exit: _
```

Atención: Este proceso puede llevar media hora o más, y se detiene en algunos momentos. ¡Paciencia!

Instalación de Node

Una vez que se cierra la consola, podemos ir a nuestro editor de código para comprobar que todos los elementos se instalaron correctamente. Para eso, abrimos la terminal, y elegimos **command prompt**. Esta será la consola que usemos para implementar correctamente SASS.



Instalación de Node

Ejecutamos los comandos **node -v** y **npm -v** para obtener información sobre las versiones.

```
john@MyShopSolutions: ~$ node -v  
v18.12.1  
john@MyShopSolutions: ~$ npm -v  
8.19.2
```

Y ahora sí, todo está preparado para poder usar SASS.

¿Preguntas?



Encuesta

Por encuestas de Zoom

Cuéntanos qué temas te resultaron más complejos de entender. **Puedes elegir más de uno.** Vamos a retomar aquellos temas que resultaron de mayor dificultad en el próximo AfterClass.



Segunda pre-entrega de tu Proyecto final



Segunda pre-entrega

Formato: Link al repositorio en GitHub donde está hosteado el proyecto. Debe tener el nombre "PreEntrega2+Apellido"

Sugerencias: Activar comentarios en el archivo

Consigna

- ✓ Hacer entrega del proyecto (dos páginas a elección) adaptado a la vista mobile y la vista desktop. Para generar un diseño responsive utilizar el Grid system de Bootstrap.
- ✓ La entrega se deberá cargar a un repositorio de GitHub público.

Objetivos

- ✓ Realizar una **estructura avanzada de la web**.
- ✓ Efectuar el **estilo avanzado de la web**.
- ✓ Generar la **carga en un repositorio en GitHub**.



Segunda pre-entrega

Estructura avanzada de la web

Formato: Archivo HTML

Consigna

- ✓ Realizar una estructura del HTML prolija, limpia, fácil de leer. Agregar elementos HTML según la necesidad de armar contenedores o elementos web determinados, basándose en el framework elegido y la documentación del mismo.

Se debe entregar

- ✓ **Maquetado de la web:** Maquetar dos páginas de tu sitio web. Para esto se puede usar bootstrap o Grids + Flexbox. Las estructuras deberán maquetar a la web en base al framework elegido, haciendo usos de clases utilitarias para armar grillas, elementos web y estilos propios del framework, además del HTML de contenido.
- ✓ **Páginas:** todas las páginas deben tener el contenido estructurado y el estilo linkeado. Agregar las diferentes librerías de JavaScript y CSS pertinentes al framework.



Segunda pre-entrega

Estilo avanzado de la web

Formato: Archivo CSS

Consigna

- ✓ Crear archivos de CSS para darle estilo a la web. Agregar todas las propiedades CSS necesarias. Hacer uso de selectores de CSS para poder darle estilo propio a los elementos que ya vienen con su propia identidad del framework.

Se debe entregar

- ✓ **Estilo del Framework:** no todos los elementos del framework van a tener una estética que condice con el sitio en el que son implementados, por lo que se usará CSS para darles un estilo acorde.
- ✓ **Estructura de la web:** uso de etiquetas no solo para armar contenido, sino para armar los elementos que van a conformar el layout de la web, los contenedores, etc.



Segunda pre-entrega

Repositorio en GitHub

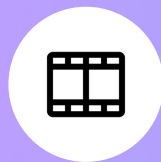
Formato: Link al repositorio en GitHub donde está hosteado el proyecto.

Consigna

- ✓ Utilizar Git de forma correcta para versionar el proyecto. Hacer uso de GitHub para brindar acceso al proyecto versionado.

Se debe entregar

- ✓ Todos los archivos necesarios para visualizar correctamente la web.
- ✓ Mostrar los commit usados para actualizar/versionar el código.



¿Quieres saber más?
**Te dejamos material
ampliado de la clase**



MATERIAL AMPLIADO

Recursos multimedia

Título

✓ [Git &](#) GitHub | TE LO EXPLICO CON GATITOS

✓ [Páginas de](#) GitHub | GitHub Pages

Resumen de la clase hoy

- ✓ Repositorio en Github .
- ✓ Creación de un repositorio para el proyecto.
- ✓ Cómo subir el proyecto al repositorio usando los comandos de Git.

Opina y valora
esta clase

Muchas gracias.

#DemocratizandoLaEducación