

MANUAL TÉCNICO
SISTEMA ORGANIZACIONAL
DE DATOS “SOD”



INTEGRANTES:

MARTINEZ VIDALES CESAR ALBERTO
PINEDA CIFUENTES JORGE STEVEN
PENAGOS POSADA SERGIO ANDRÉS

SERVICIO NACIONAL DE APRENDIZAJE SENA
ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

2067469

2021

Contenido

| | |
|--|----|
| PRESENTACIÓN | 4 |
| OBJETIVOS | 4 |
| OBJETIVOS ESPECÍFICOS..... | 4 |
| REQUERIMIENTOS TÉCNICOS | 5 |
| REQUERIMIENTOS MÍNIMOS DE HARDWARE | 5 |
| REQUERIMIENTOS MÍNIMOS DE SOFTWARE | 5 |
| HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO | 5 |
| VISUAL STUDIO CODE..... | 5 |
| LARAVEL..... | 5 |
| MYSQL | 6 |
| NODEJS | 6 |
| COMPOSER | 6 |
| XAMPP | 6 |
| GIT | 7 |
| LARAGON (OPCIONAL) | 7 |
| INSTALACIÓN DE SOFTWARE..... | 7 |
| INSTALACIÓN DE LARAGON (OPCIONAL) | 8 |
| PASO 1: DESCARGAR LARAGON | 8 |
| PASO 2: INSTALACIÓN DE LARAGON..... | 8 |
| PASO 3: CONFIGURACIÓN DE LARAGON | 11 |
| PASO 4: INSTALACIÓN DE “phpMyAdmin” | 14 |
| PASO 5: INICIAR LARAGON | 18 |
| INSTALACIÓN DE XAMPP..... | 19 |
| PASO 1: DESCARGAR XAMPP | 19 |
| PASO 2: INSTALACIÓN XAMPP | 21 |
| PASO 3: INICIAR XAMPP | 23 |
| INSTALACIÓN DE COMPOSER | 24 |
| PASO 1: DESCARGAR COMPOSER: | 24 |
| PASO 2: INSTALAR COMPOSER: | 25 |
| PASO 3: CONFIRMAR INSTALACIÓN | 27 |
| INSTALACIÓN DE GIT | 29 |
| PASO 1: DESCARGAR: | 29 |

| | |
|---|----|
| PASO 2: INSTALAR | 29 |
| PASO 3: CONFIRMAR INSTALACIÓN..... | 39 |
| INSTALACIÓN DE VISUAL STUDIO CODE..... | 40 |
| PASO 1: DESCARGAR | 40 |
| PASO 2: INSTALACIÓN | 41 |
| IMPLEMENTACIÓN DEL PROYECTO | 43 |
| CLONACIÓN DE PROYECTOS LARAVEL DESDE GITHUB | 43 |
| COMO HACERLO | 43 |

PRESENTACIÓN

El siguiente manual les guiará a los usuarios que harán uso del sistema, en donde se les dará a conocer los requerimientos y estructuras para la construcción del sistema, en el desarrollo del aplicativo web mediante una base de datos, el cual se muestra las herramientas necesarias para la construcción y la funcionalidad del sistema.

Es importante tener en cuenta que en este manual se darán los pasos a seguir y especificaciones mínimas de hardware y software para la correcta instalación del aplicativo

OBJETIVOS

Informar y especificar al usuario la estructuración, instalación y configuración del sistema de información con el fin de que puedan hacer uso de él

OBJETIVOS ESPECÍFICOS

- Presentar la estructuración, diseño y definición del aplicativo.
- Explicar específicamente el procedimiento y el paso a paso de la instalación del aplicativo.
- Especificar detalladamente los requisitos del Hardware y Software para la instalación del aplicativo.
- Definir las herramientas utilizadas para el sistema de información.

REQUERIMIENTOS TÉCNICOS

REQUERIMIENTOS MÍNIMOS DE HARDWARE

- Procesador Core i3
- 4GB RAM
- Conexión a internet
- Disco duro de 250GB

REQUERIMIENTOS MÍNIMOS DE SOFTWARE

- Sistema operativo Windows 7/8/8.1/10

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

VISUAL STUDIO CODE

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

LARAVEL

Es un framework PHP. Es uno de los frameworks más utilizados y de mayor comunidad en el mundo de Internet.

Como framework resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web.

Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además, permite y facilita el trabajo en equipo y promueve las mejores prácticas.

MYSQL

Es un sistema de gestión de base de datos (SGBD) de código abierto. El SGBD MySQL pertenece actualmente a Oracle. Funciona con un modelo cliente-servidor. Eso quiere decir que los ordenadores que instalan y ejecutan el software de gestión de base de datos se denominan clientes. Cada vez que necesitan acceder a los datos, los clientes se conectan al servidor del sistema de gestión de base de datos y le solicitan la información que necesitan. El servidor se la brinda siempre y cuando tenga los derechos de acceso.

NODEJS

Node.js, es un entorno en tiempo de ejecución multiplataforma para la capa del servidor (en el lado del servidor) basado en JavaScript.

Node.js es un entorno controlado por eventos diseñado para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tienes que preocuparte con el bloqueo de procesos, pues no hay bloqueos

COMPOSER

Composer es un manejador de paquetes para PHP que proporciona un estándar para administrar, descargar e instalar dependencias y librerías.

XAMPP

Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

GIT

Es un sistema de control de versiones distribuido de código abierto desarrollado por Linus Torvalds, el creador de Linux. El control de versiones distribuido permite a los desarrolladores descargar un software, realizar cambios y subir la versión que han modificado.

LARAGON (OPCIONAL)

Es una herramienta para equipos técnicos que permite crear diferentes entornos de desarrollo, facilitando el trabajo con las aplicaciones. También instalando todos los componentes necesarios para empezar a trabajar minimizando las configuraciones necesarias

¿Qué componentes nos ofrece laragon?

- Apache
- Cmdr (Consola)
- COMPOSER
- Git
- HeidiSQL
- MySQL
- NodeJS
- PHP
- Notepad++
- Posibilidad de incorporar phpMyAdmin

INSTALACIÓN DE SOFTWARE

A continuación, se explicará la instalación de XAMPP y LARAGON, usted como cliente es libre de instalar cualquiera de los dos para poder ejecutar el sistema de información

¿Cuál es la diferencia entre XAMPP y LARAGON?

En XAMPP deberá instalar todos los programas necesarios para poder ejecutar el sistema de información, pero es mas sencillo y amigable para las personas que están iniciando en el mundo del desarrollo, por otra parte, LARAGON instala automáticamente todo lo

necesario para poder ejecutar el sistema de información, pero requiere un poco más de conocimiento para realizar las configuraciones.

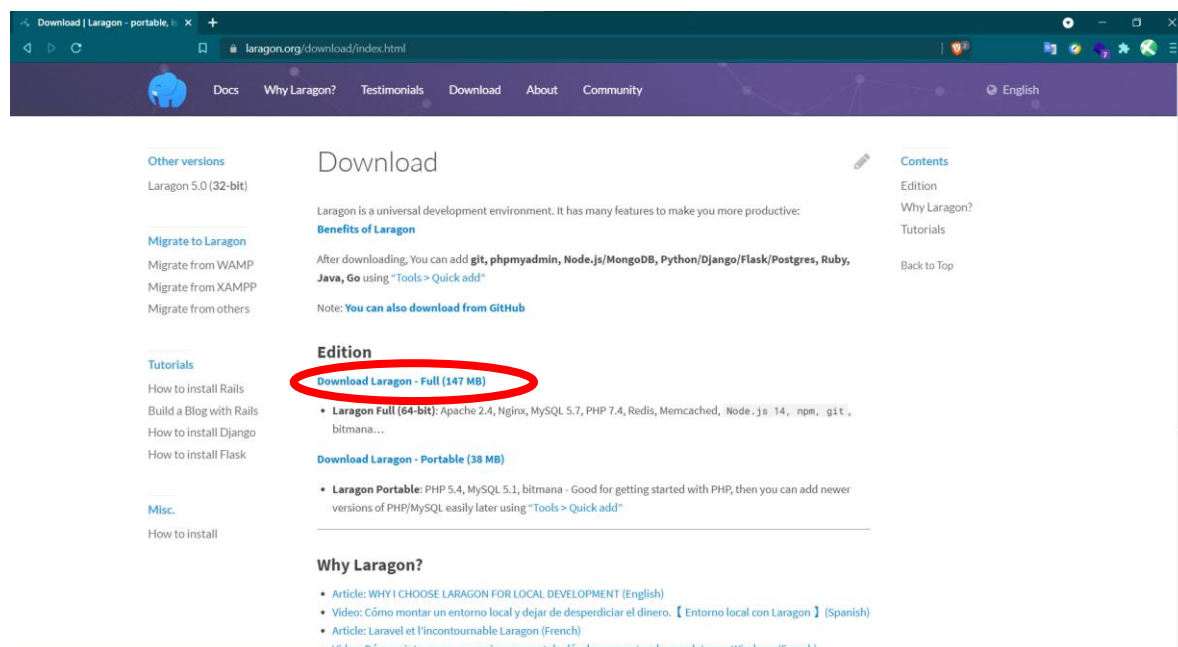
“Recordamos que es indispensable instalar alguno de los dos Administradores de bases de datos, si desea puede instalar los dos o solo uno”

INSTALACIÓN DE LARAGON (OPCIONAL)

“Si no desea instalar LARAGON como administrador de bases de datos, omita esta explicación y continúe en “INSTALACIÓN DE XAMPP”

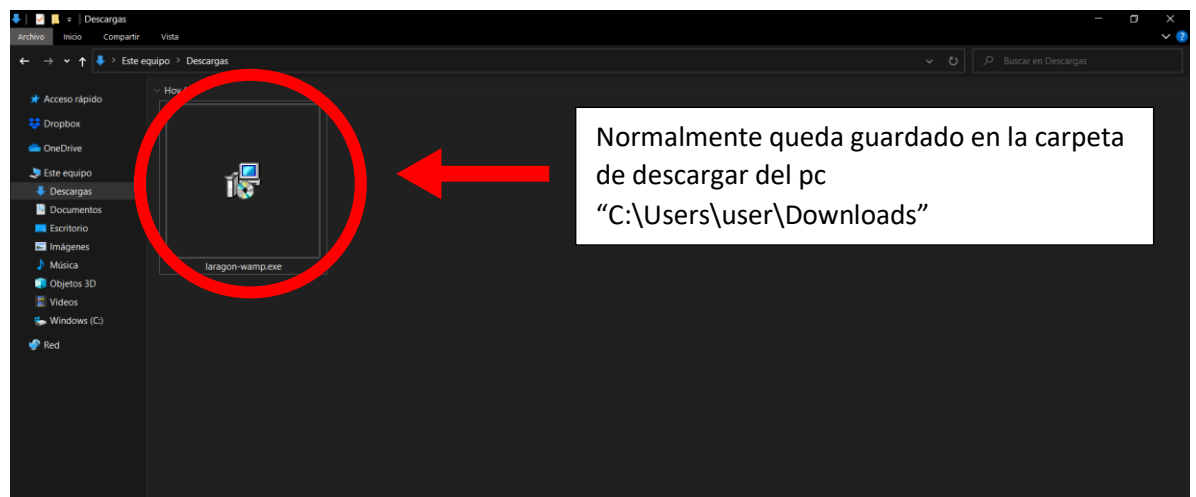
PASO 1: DESCARGAR LARAGON

Para descargar LARAGON nos dirigimos a la siguiente ruta <https://laragon.org/download/index.html> nos aparecerá una ventana con la siguiente información y procedemos a la descarga(al dar clic en **“DOWNLOAD LARAGON - FULL (147 MB)”**) se iniciara automáticamente la descarga:

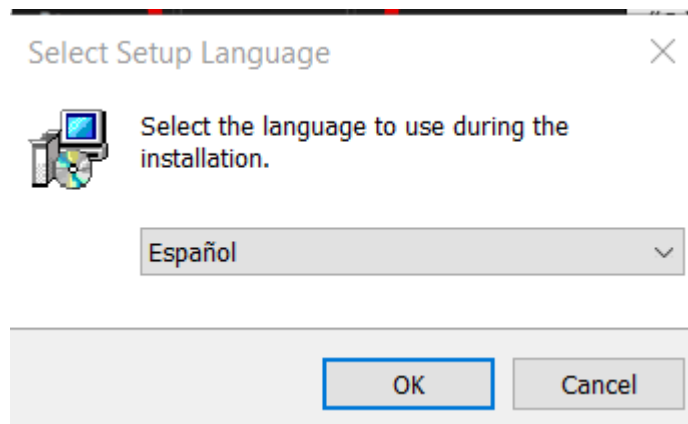


PASO 2: INSTALACIÓN DE LARAGON

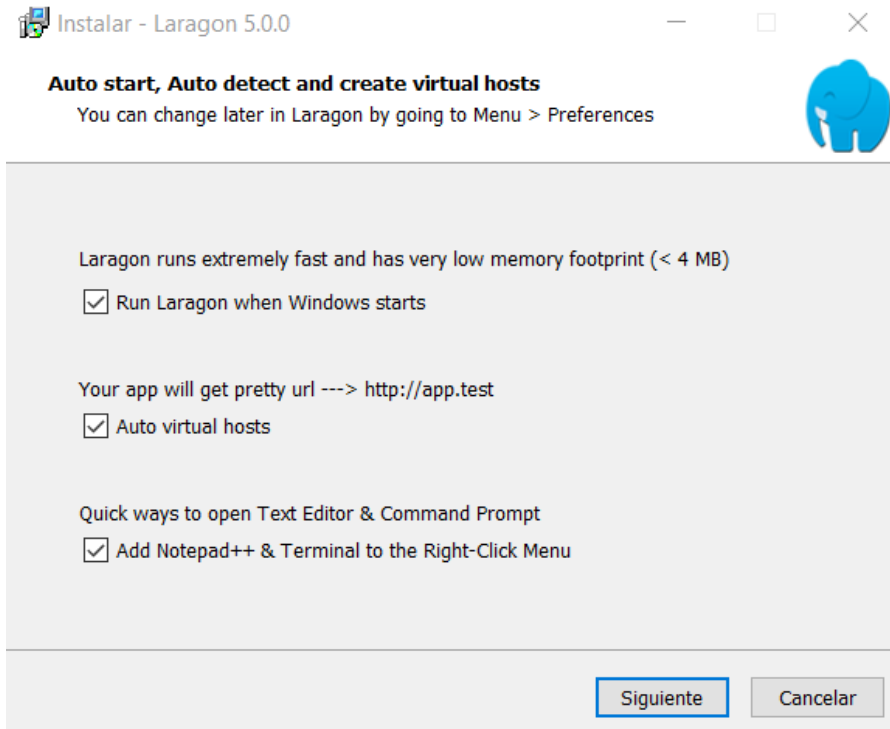
Procedemos a ejecutar el archivo “laragon-wamp.exe” que nos dejo en la carpeta de descargas y aceptamos permisos de administrador.



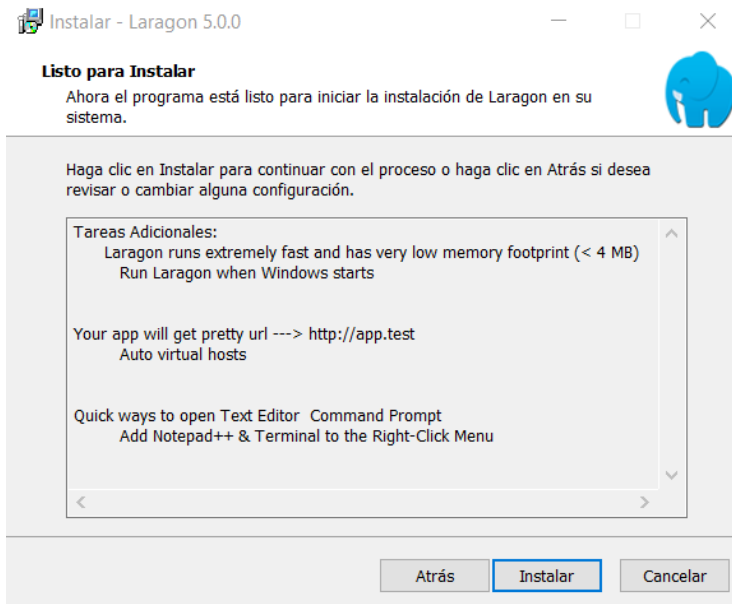
Seleccionamos el idioma que deseamos, en nuestro caso sera en español y damos en OK.



Dejamos todo tal cual esta y solo damos en Siguiente.

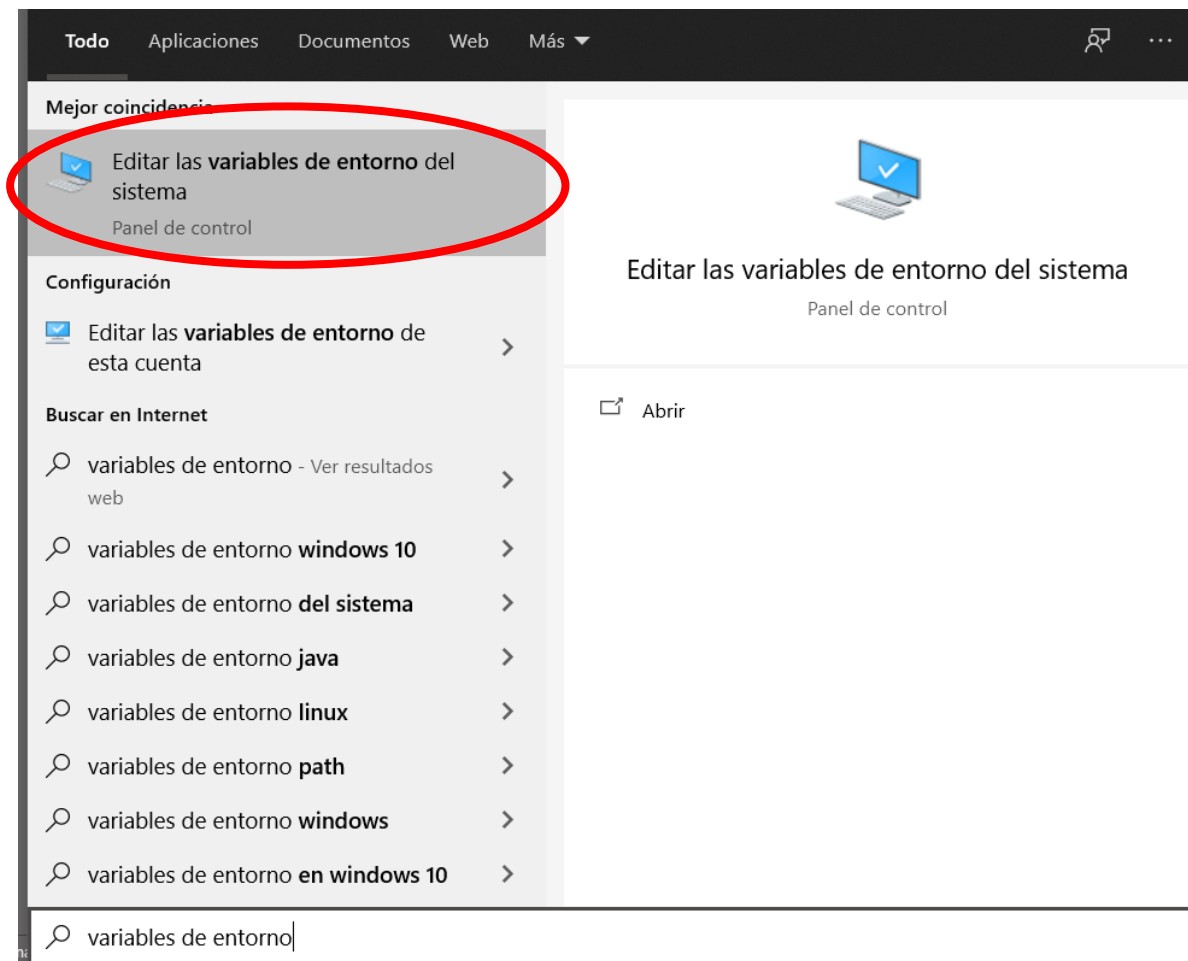


Por último, damos en instalar

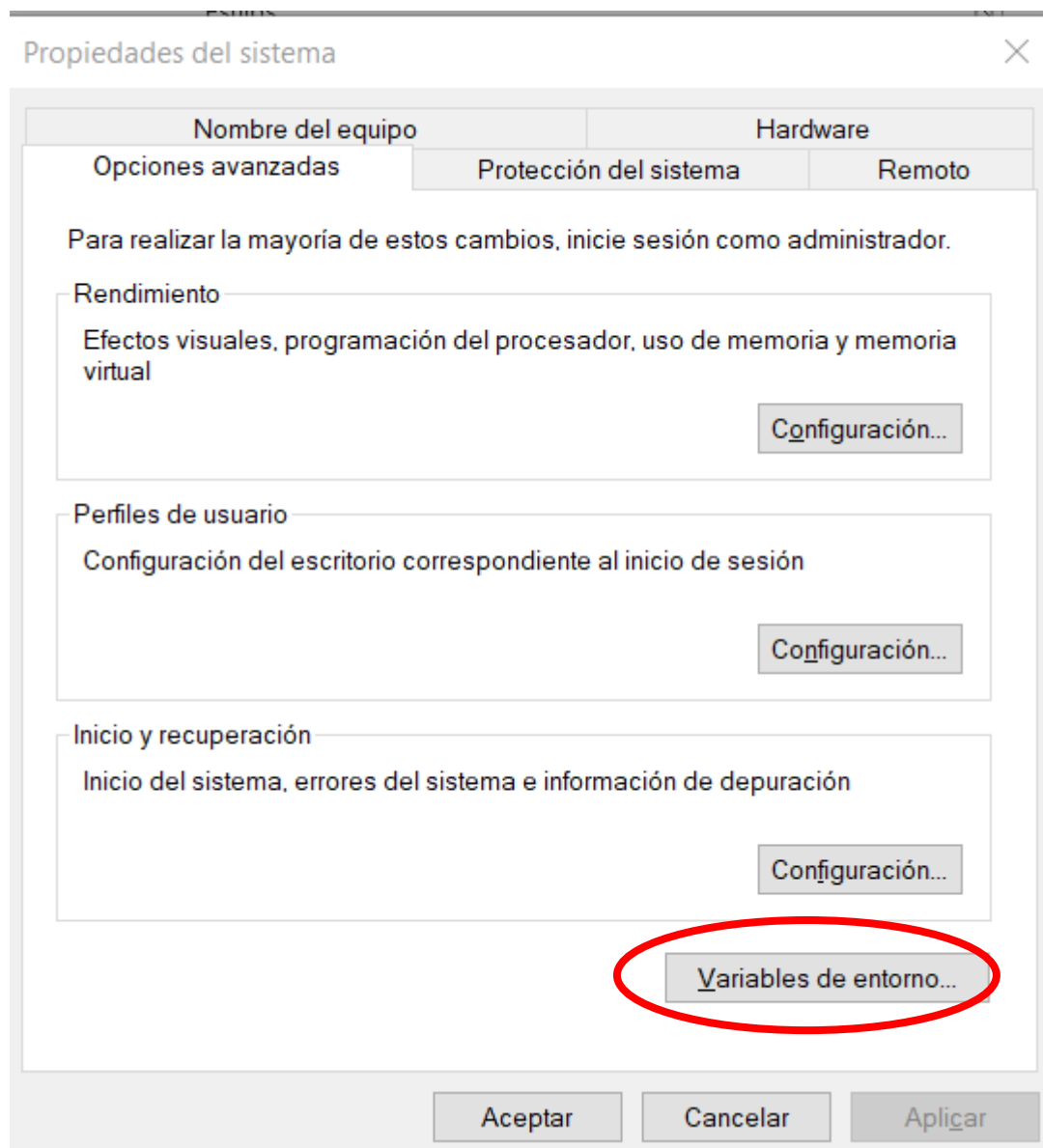


PASO 3: CONFIGURACIÓN DE LARAGON

Accedemos a la sección variables de entorno



Seleccionamos “Variables de entorno...”



Seleccionamos la variable "Path" y procedemos a editarla

Variables de usuario para steve

| Variable | Valor |
|----------|---|
| Path | C:\Users\steve\AppData\Local\Programs\Microsoft VS Code\... |

2. Seleccionar variable "Path"

Nuevo...

Editar...

Eliminar

1. Editar variable "Path"

Variables del sistema

| |
|--|
| |
|--|

Nueva...

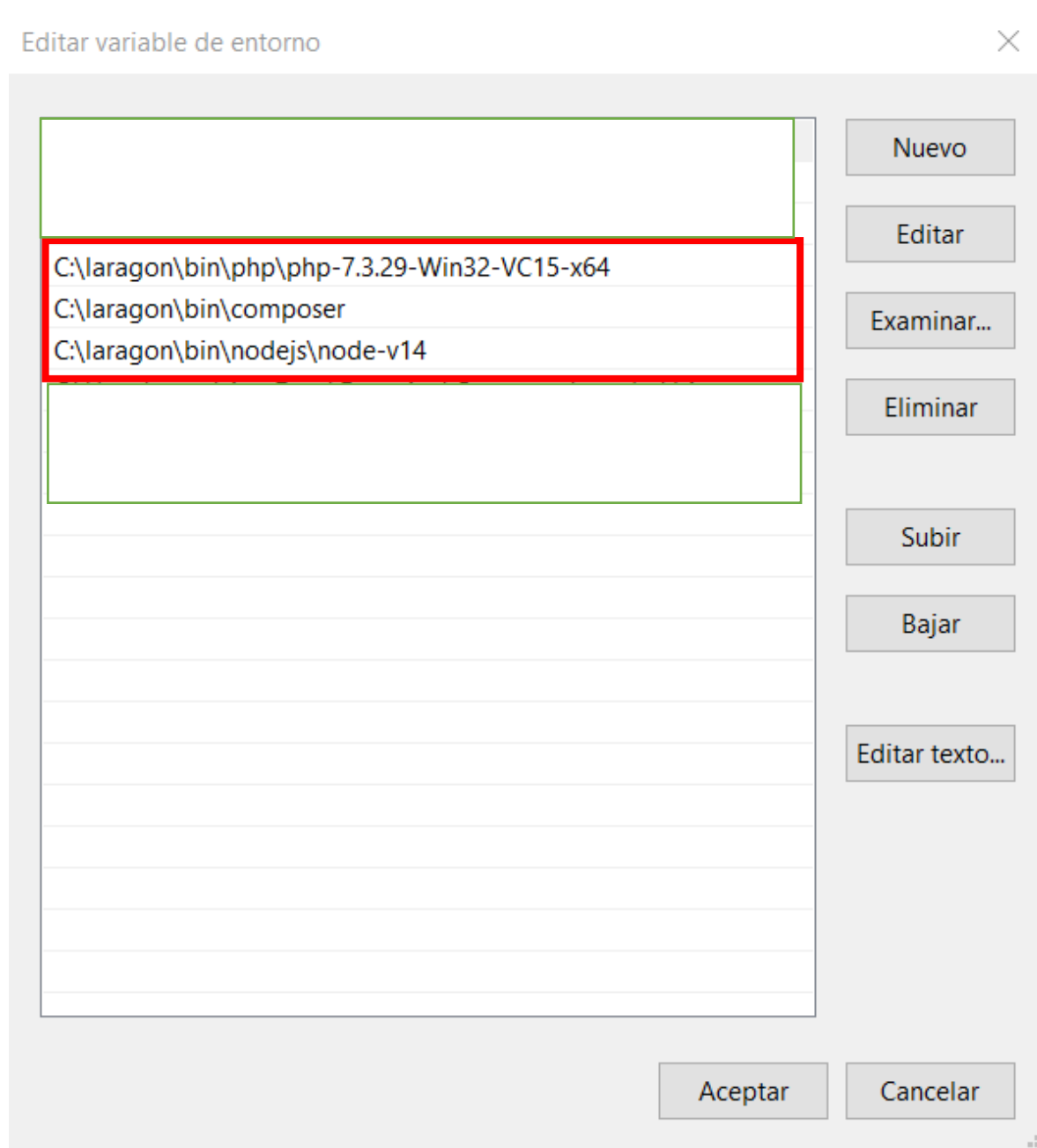
Editar...

Eliminar

Aceptar

Cancelar

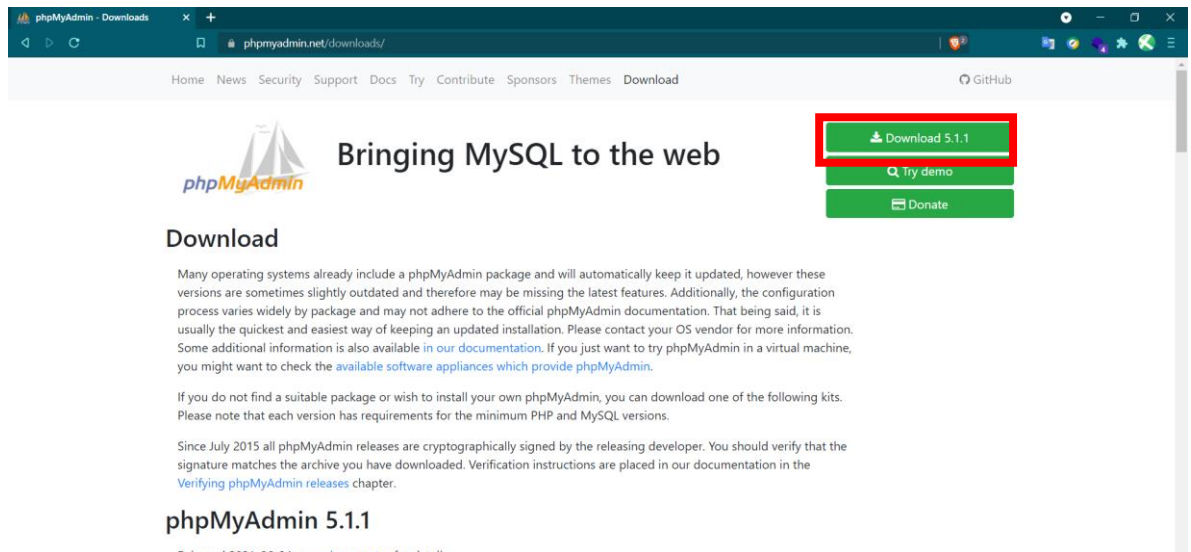
Por último, procedemos a insertar las siguientes rutas y damos en Aceptar



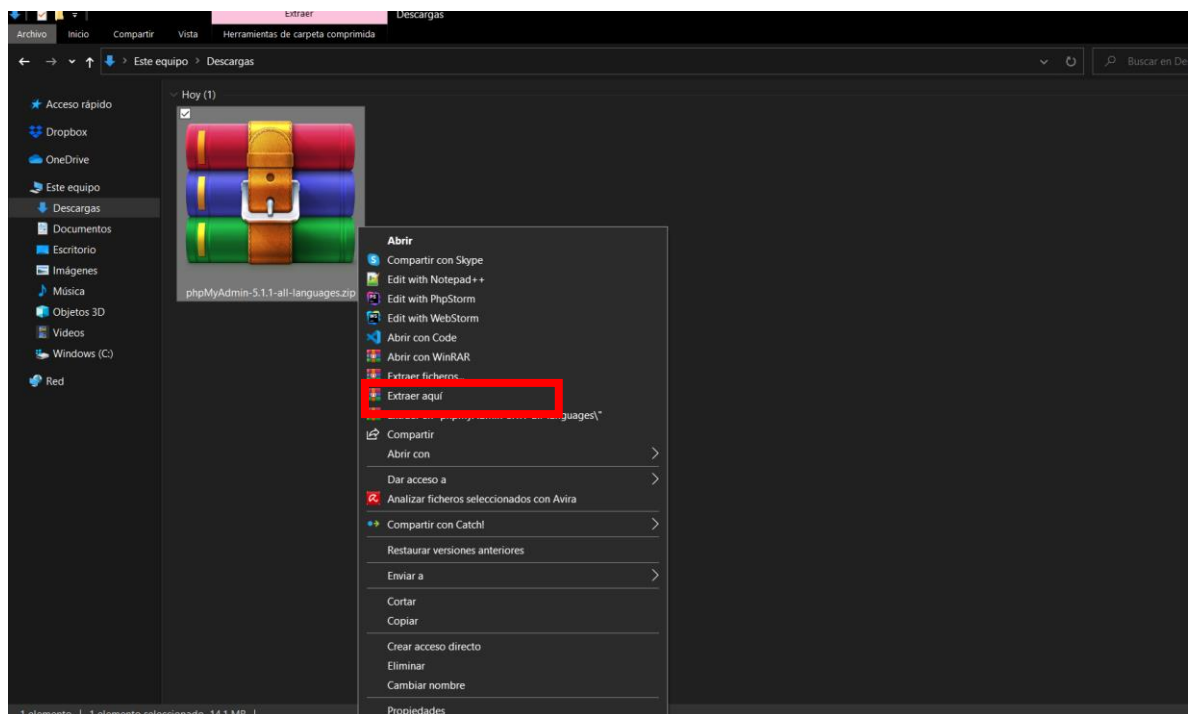
“La versión varía dependiendo del tiempo en que se encuentre, como se observa en la imagen se está trabajando con una versión de PHP 7.3 y NodeJs versión 14”

PASO 4: INSTALACIÓN DE “phpMyAdmin”

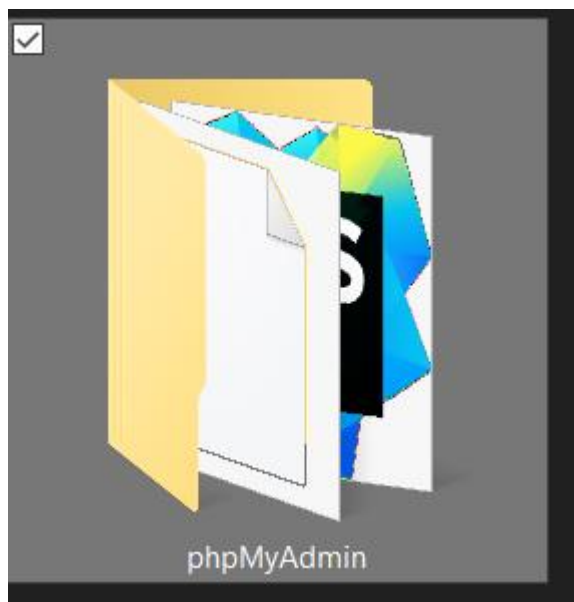
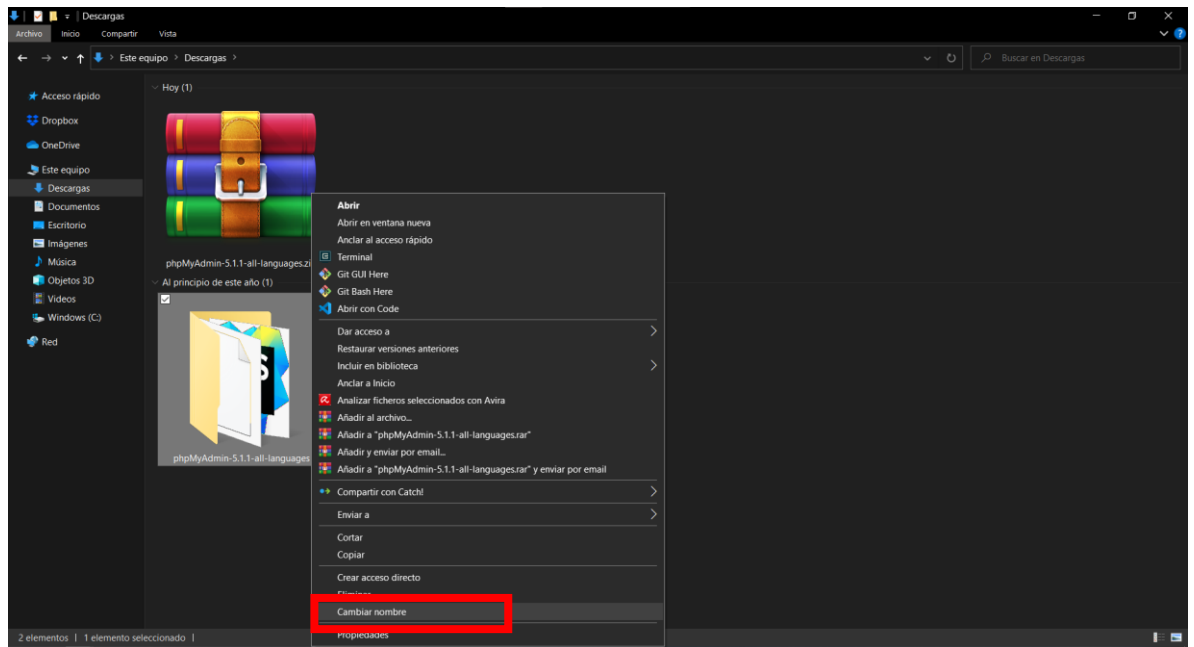
Nos dirigimos a la siguiente ruta <https://www.phpmyadmin.net/downloads/> y descargamos (actualmente está en la versión 5.1.1)



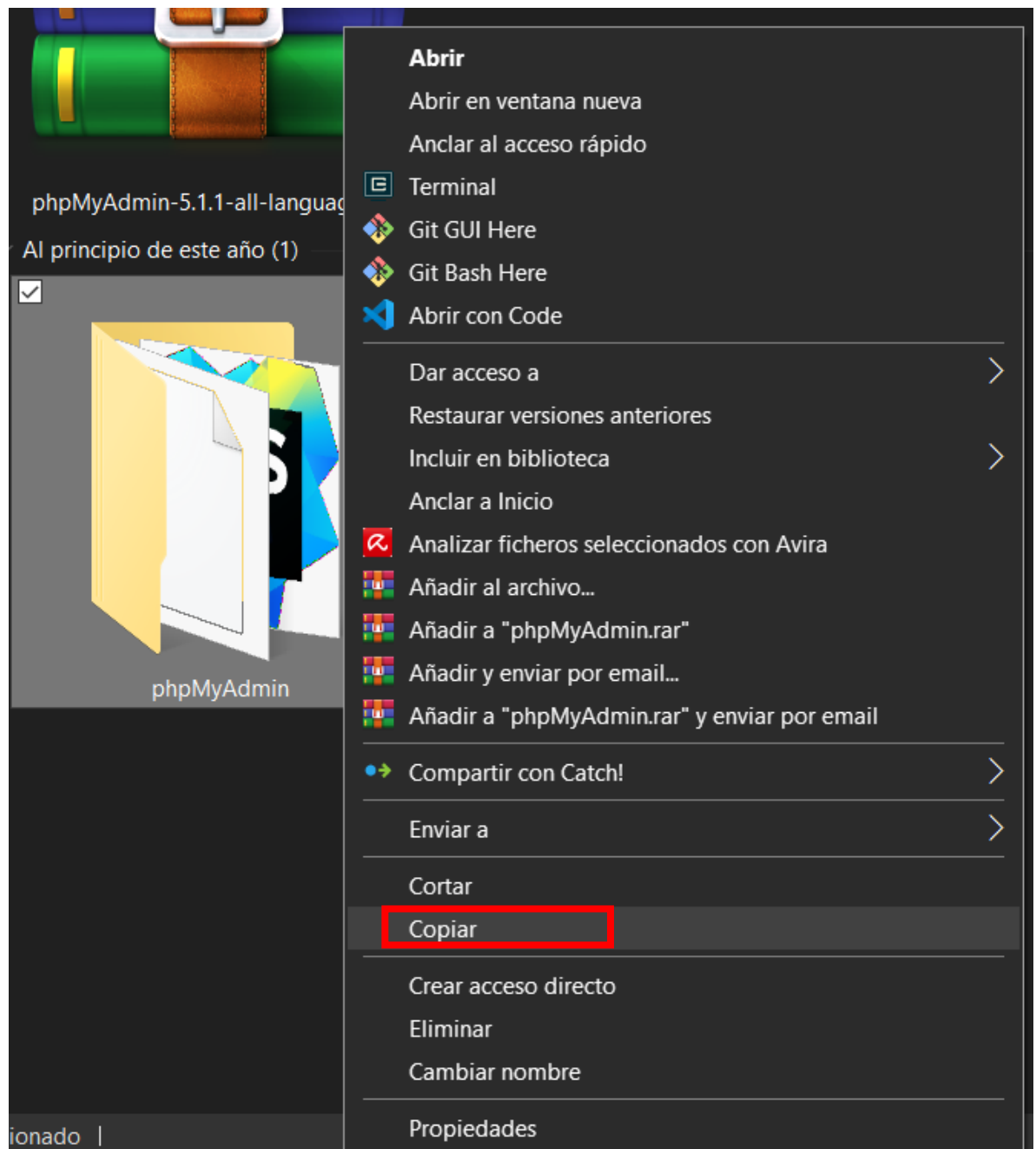
Extraemos el archivo que nos deja en la carpeta de descargas



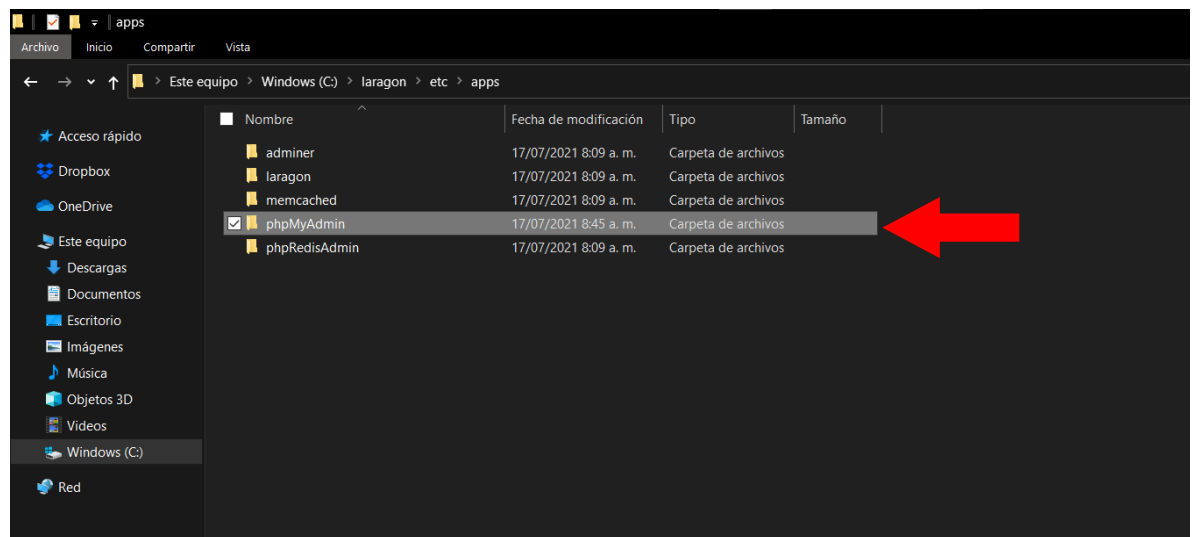
Le cambiamos el nombre a la carpeta que nos creó al momento de la extracción



“phpMyAdmin” deberá ser el nombre que le dejaremos a la carpeta, procedemos a copiarla

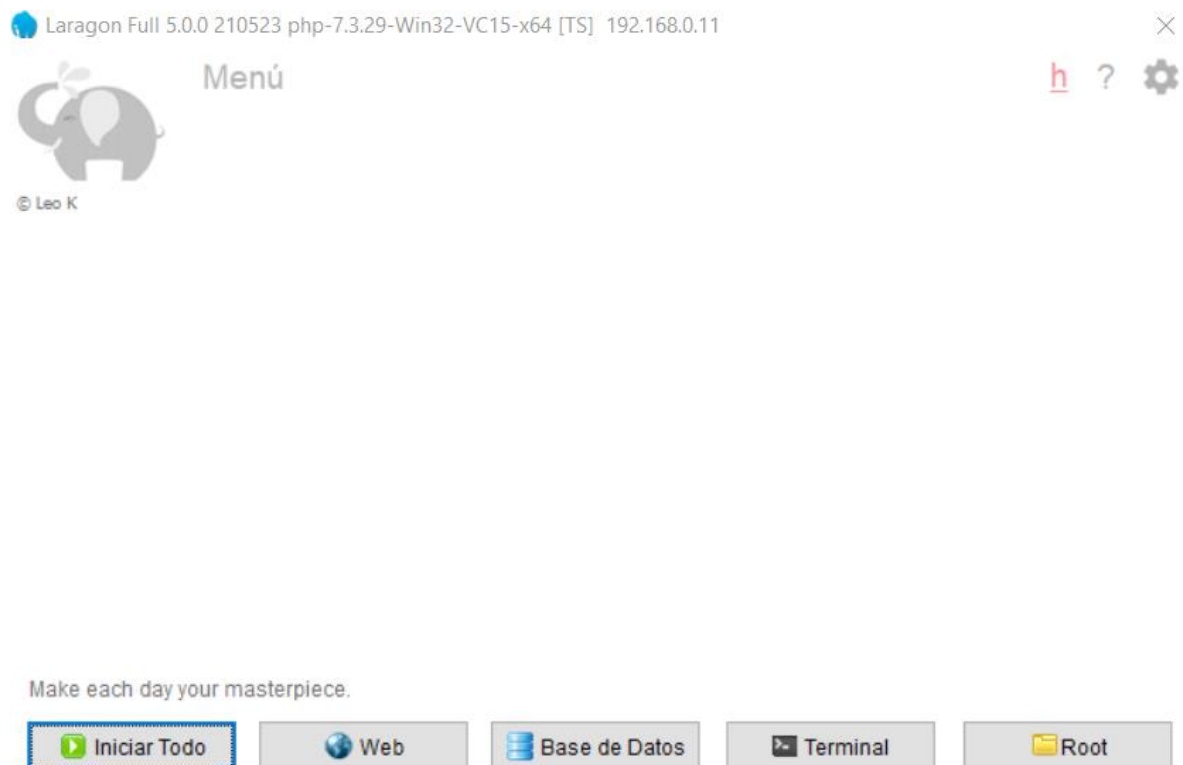


Pegamos la carpeta en la siguiente ruta "C:\laragon\etc\apps"



PASO 5: INICIAR LARAGON

Accedemos a LARAGON y damos en “Iniciar Todo”



Si todo quedo correctamente deberá aparecer de la siguiente manera



“En caso de que alguno de esos dos textos salga en rojo es por qué ocurrió un error, para solucionarlo verifique que no esté usando el mismo puerto en otro entorno de desarrollo o revise las versiones de php y MySQL.”

INSTALACIÓN DE XAMPP

PASO 1: DESCARGAR XAMPP

Para descargar XAMPP es necesario que des clic en la siguiente URL <https://www.apachefriends.org/index.html>, aquí se puede encontrar la versión más reciente y para todos los sistemas operativos:

XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?


XAMPP is the most popular PHP development environment


XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.




Download

Click here for other versions

 XAMPP for Windows
8.0.10 (PHP 8.0.10)

 XAMPP for Linux
8.0.10 (PHP 8.0.10)

 XAMPP for OS X
8.0.10 (PHP 8.0.10)

Al pulsar sobre la opción deseada, nos saldrá la siguiente página:

[Apache Friends](#) [Download](#) [Add-ons](#) [Hosting](#) [Community](#) [About](#) [Search](#) [EN](#)

Awesome!

Your download will start automatically. If it doesn't, [click here](#).

Reading

Be sure to read the install instructions and FAQs:

- [Linux FAQs](#)
- [Windows FAQs](#)
- [OS X FAQs](#)
- [OS X XAMPP-VM FAQs](#)

You can find additional help on our [forums](#) or [Stack Overflow](#).

Tell Your Friends about XAMPP

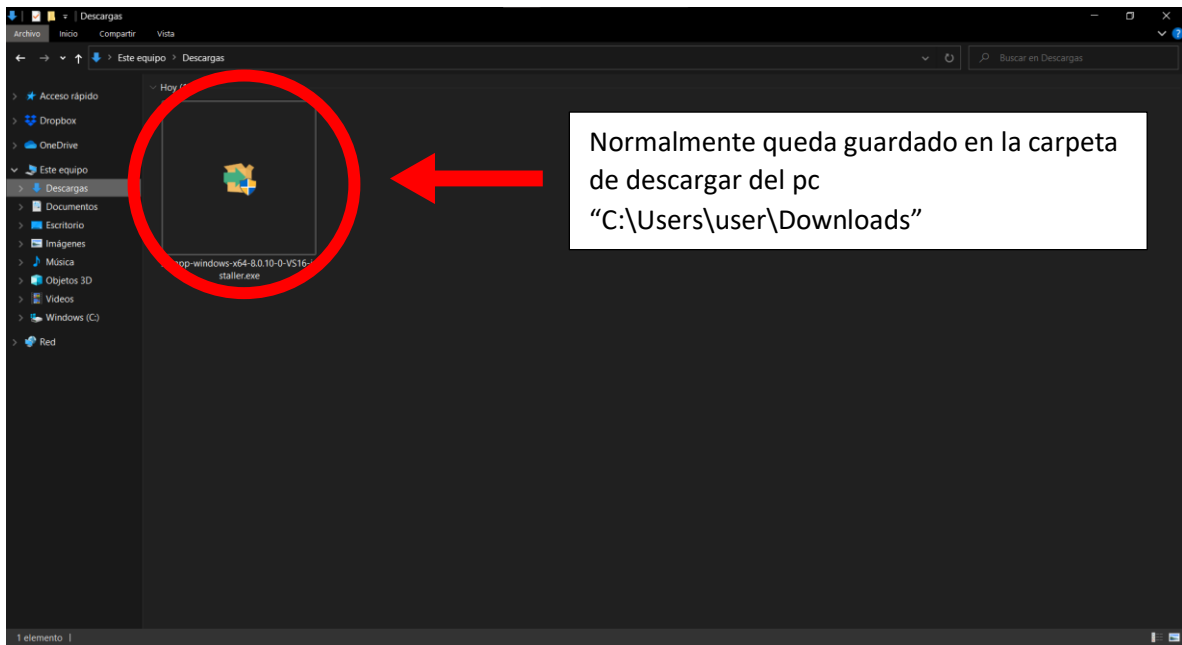
I just got #XAMPP from @ApacheFriends
<https://www.apachefriends.org> #opensource.

[Tweet!](#)



Inmediatamente comenzará la descarga del ejecutable de XAMPP; si esto no ocurriese, pulsar sobre “[click here](#)” de la página anterior.

Una vez descargado el ejecutable, buscarlo en la carpeta de descargas y dar doble clic sobre él.



Como puede observarse, se ha instalado la versión más reciente para Windows, la versión 1.8.10.

PASO 2: INSTALACIÓN XAMPP

Al hacer doble clic sobre el ejecutable descargado, comenzará la instalación en sí. En primer lugar, saldrá la siguiente pantalla de instalación:

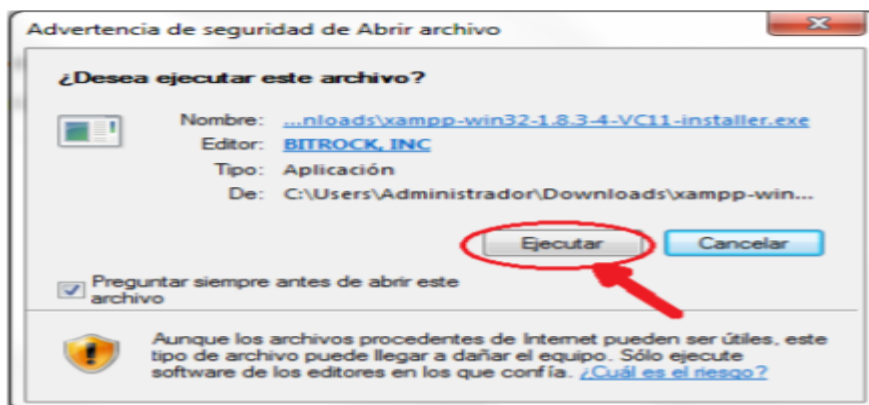


Ilustración 4: Inicio de la instalación.

Al pulsar sobre ejecutar, comienza la instalación, apareciendo:

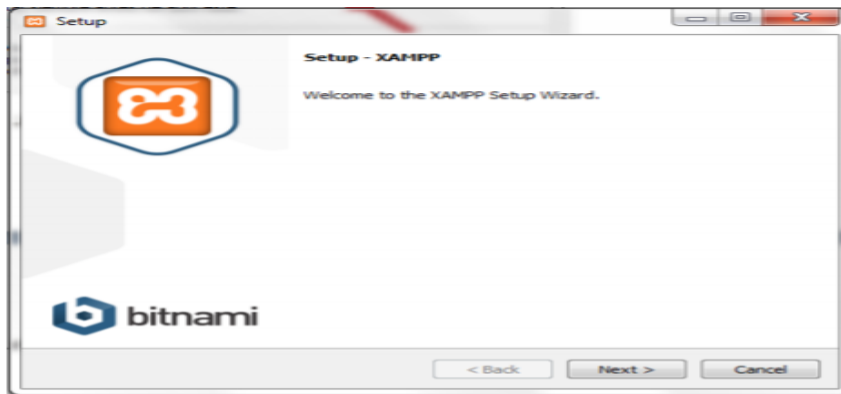


Ilustración 5: Primera pantalla de instalación

Debemos pulsar sobre “Next” para continuar, para poder elegir en la siguiente pantalla de instalación qué opciones deseamos instalar en el equipo:

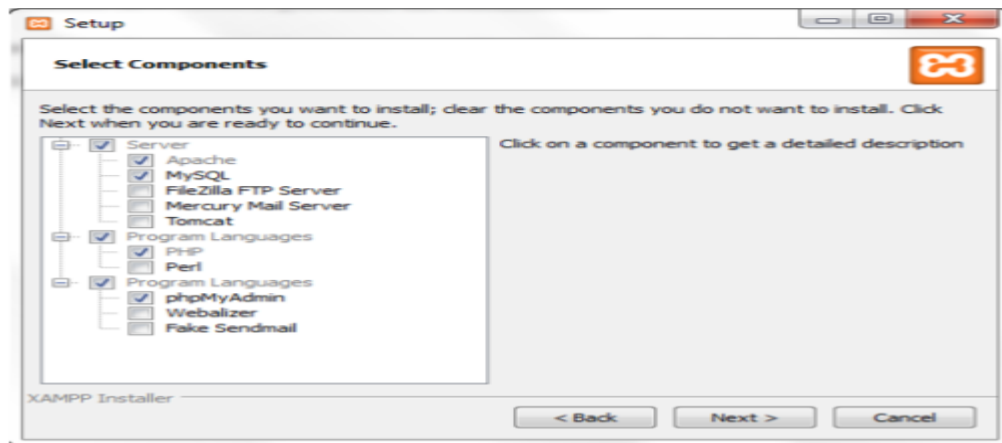


Ilustración 6: Elección opciones a instalar.

Tras esto, debemos seleccionar la carpeta donde deseamos sea instalado XAMPP, tal como muestra la siguiente ilustración:

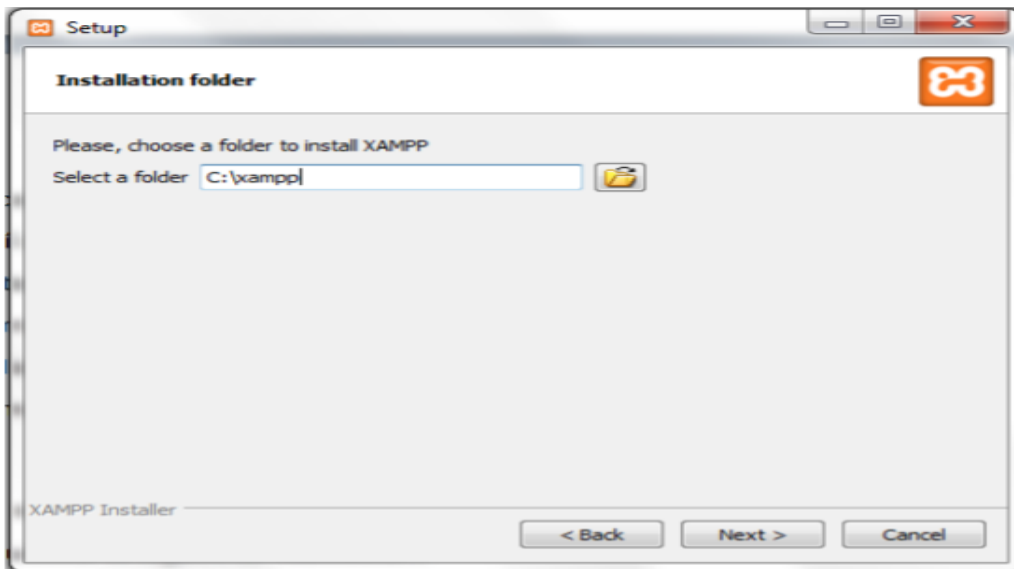


Ilustración 7: Carpeta donde instalar XAMPP.

Al pulsar sobre “Next” comienza la instalación

PASO 3: INICIAR XAMPP

Una vez ha concluido la instalación de XAMPP podemos iniciar la ejecución del programa pulsando sobre el icono que se encuentra en el escritorio, o bien buscándolo en el directorio donde el usuario deseó su instalación, en nuestro caso C:/XAMPP. La pantalla que aparecerá será:



Como observamos, solo aparecen las opciones instaladas, en nuestro caso, el servidor Apache y el servidor MySQL, los cuales pueden ser iniciados pulsando “Start” y administrados pulsando sobre “Admin”, una vez estos los servicios son arrancados.

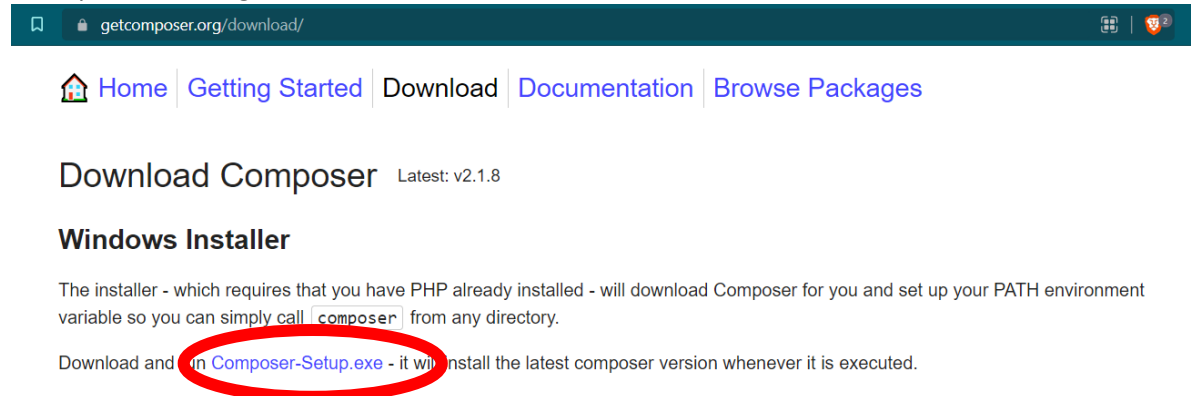
INSTALACIÓN DE COMPOSER

PASO 1: DESCARGAR COMPOSER:

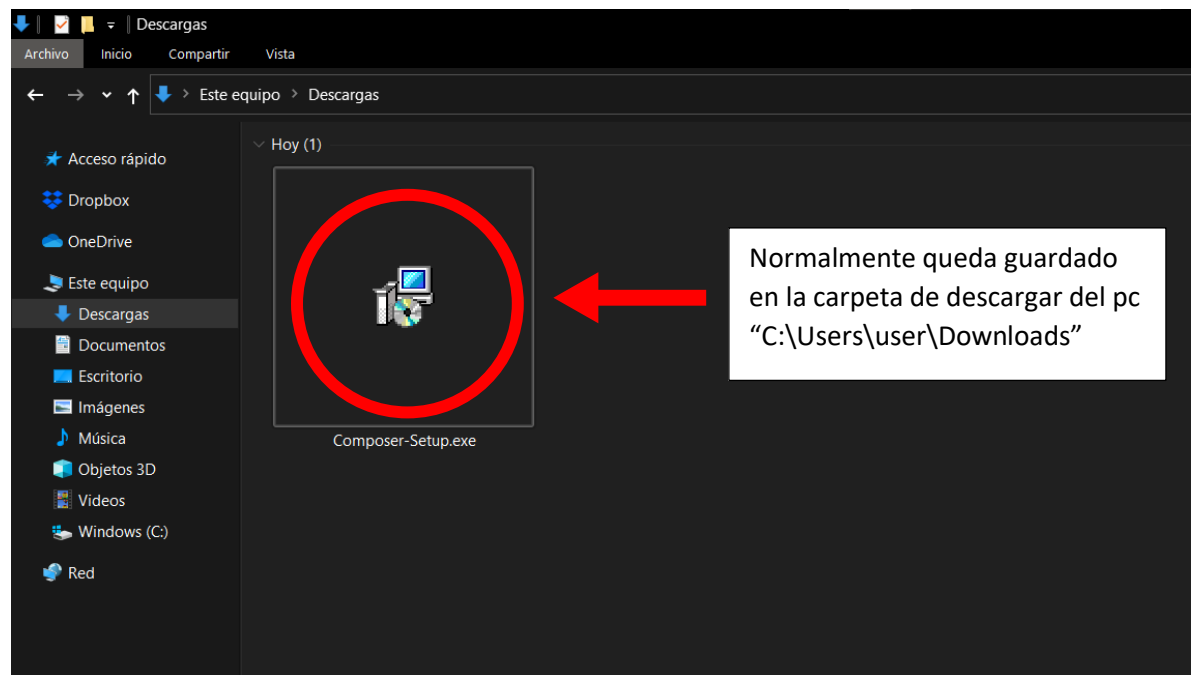
Procedemos a ir a nuestro navegador y dirigimos a la siguiente ruta

<https://getcomposer.org/download/>

Nos aparecerá lo siguiente:

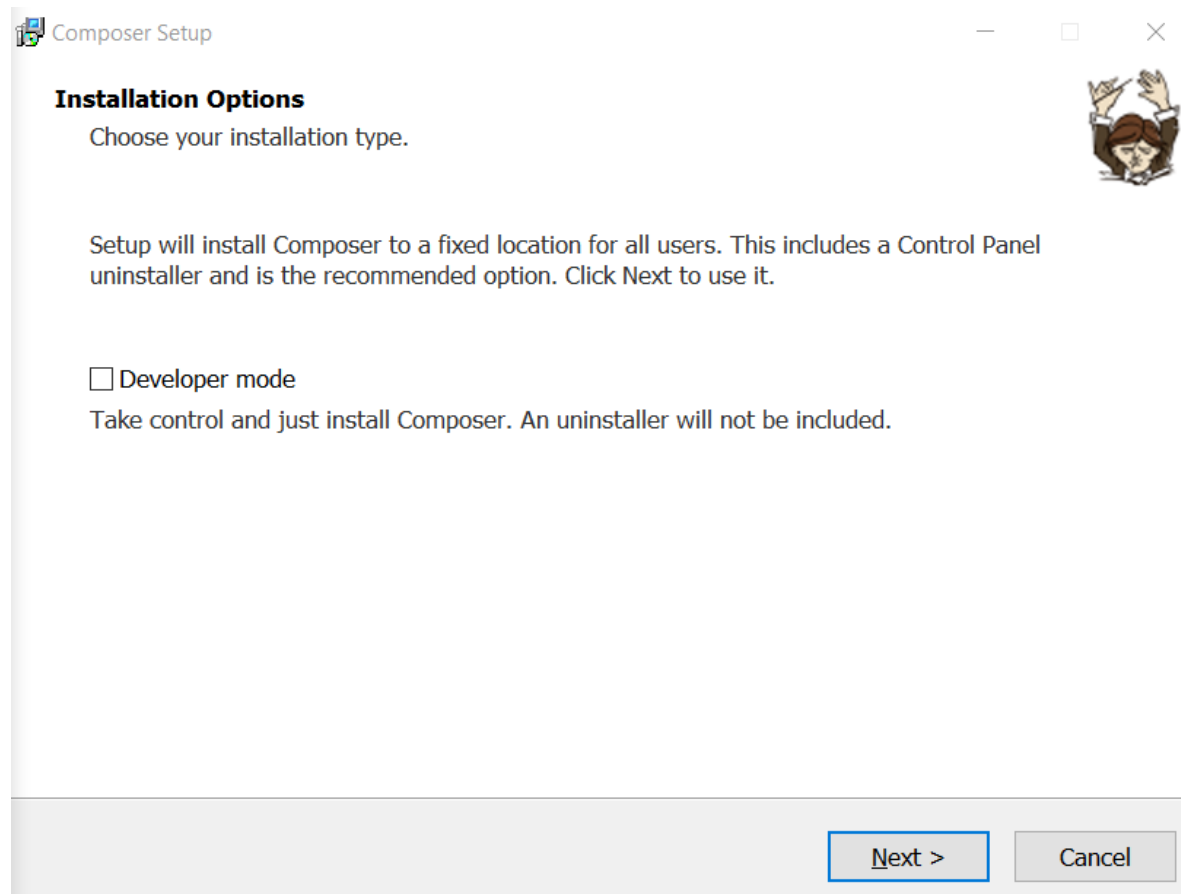


Damos clic en “Composer-Setup.exe”, lo cual nos descargara automáticamente el ejecutable para instalar en nuestro sistema.

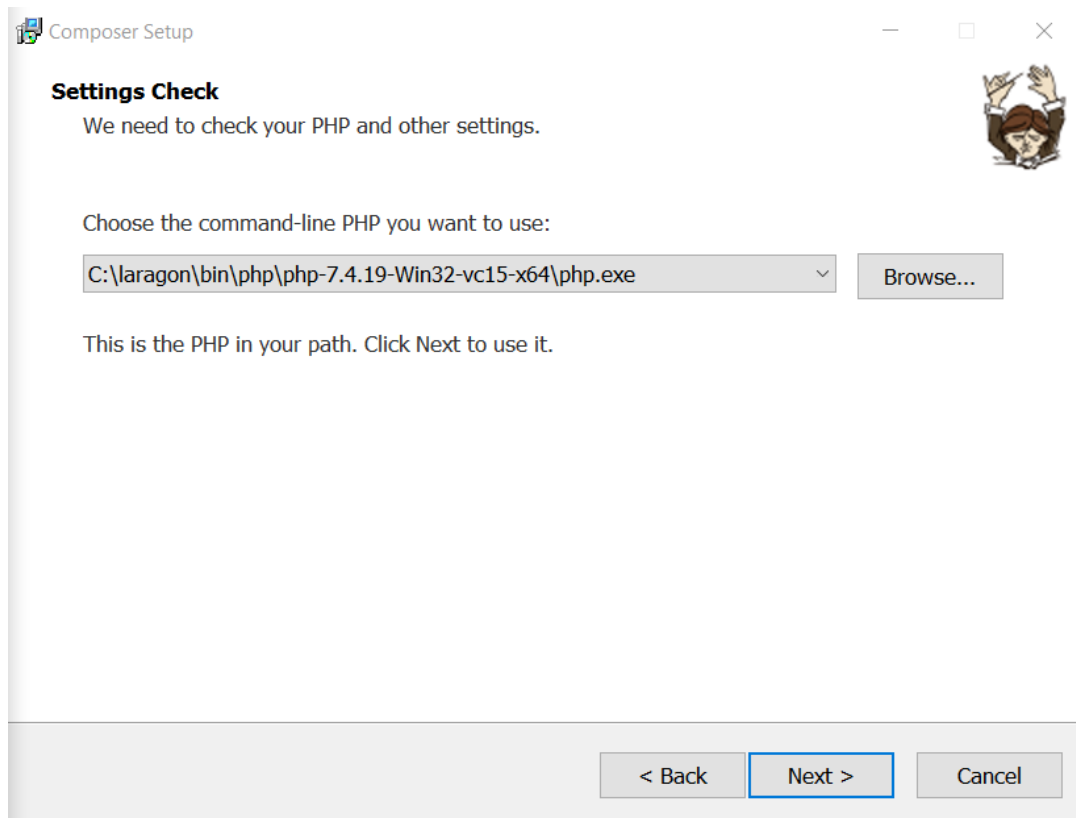


PASO 2: INSTALAR COMPOSER:

Ejecutamos el archivo descargado aceptando permisos de administración (En caso de que salga una ventana emergente preguntando que, si desea instalar en todos los usuarios, selecciona la opción “Instalar en todos los usuarios), y procede a aparecer la siguiente ventana:

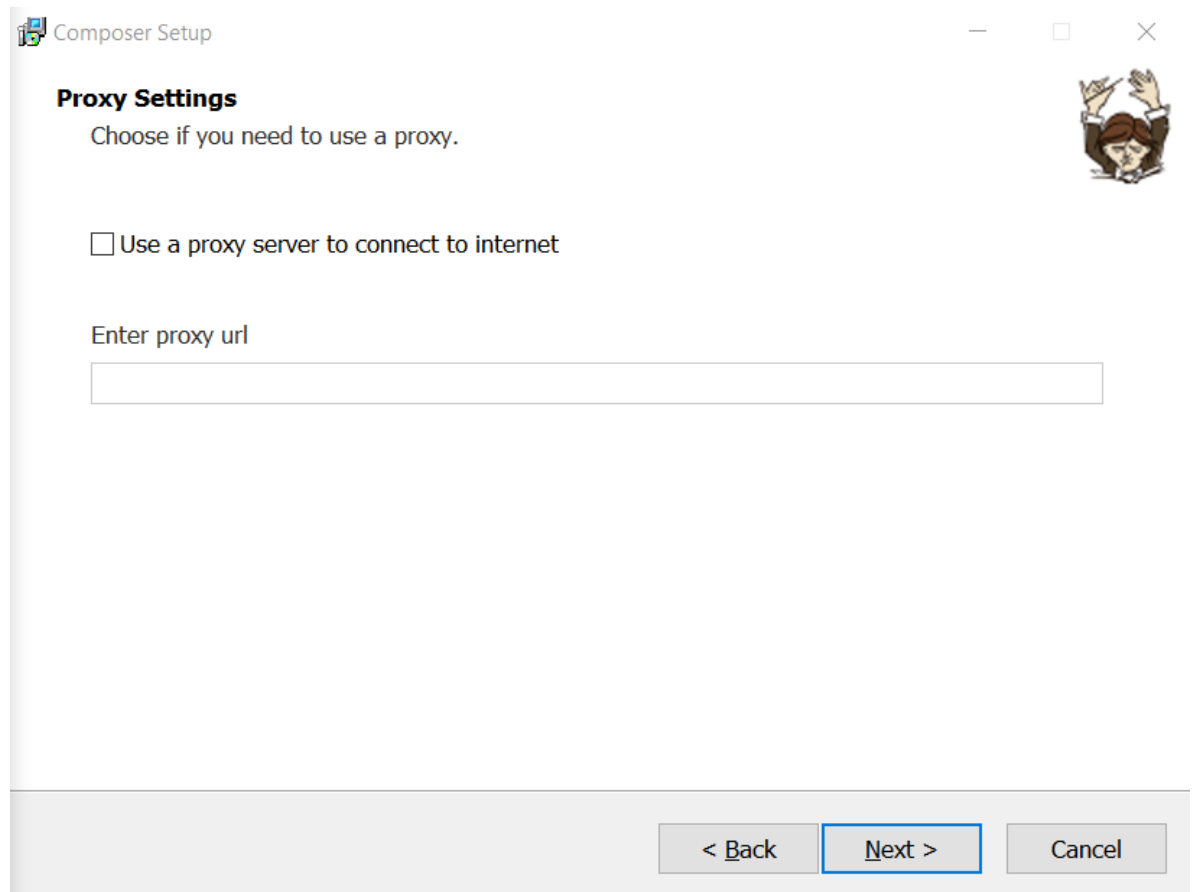


Damos clic en “Next” sin modificar nada, luego nos saldrá la siguiente ventana:



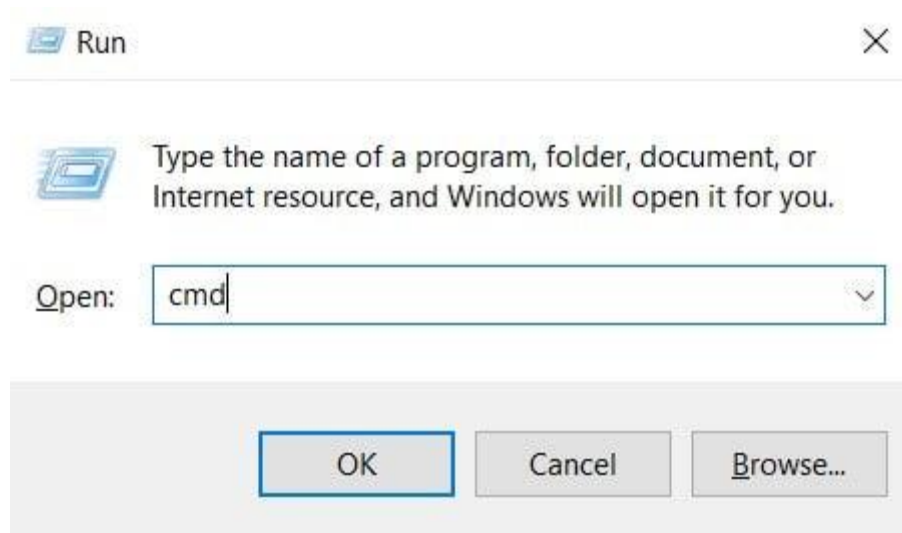
Aparecerá otra ventana y te pedirá que ubiques la línea de comando PHP. Por defecto, está en C:/xampp/php/php.exe. Después de especificar la ubicación, haz clic en Next. En caso de que no aparezca esa ruta por defecto, solo es buscarla o copiarla y pegarla

Se te solicitará la configuración de proxy. Deja la casilla sin marcar y omite esta parte presionando Next. Luego, en la última ventana, haz clic en Instalar.



PASO 3: CONFIRMAR INSTALACIÓN

Después de completar la instalación, abre el símbolo del sistema. Presiona CTRL + R, escribe «cmd» y haz clic en OK.



Ingresa el siguiente comando y luego oprima enter: composer

```

C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.19042.1237]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\steve>composer

```

Si todo salió bien deberá aparecer lo siguiente:

```

C:\> Símbolo del sistema

Microsoft Windows [Versión 10.0.19042.1237]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\steve>composer

Composer version 2.0.12 2021-04-01 10:14:59

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache               Prevent use of the cache
  -v|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

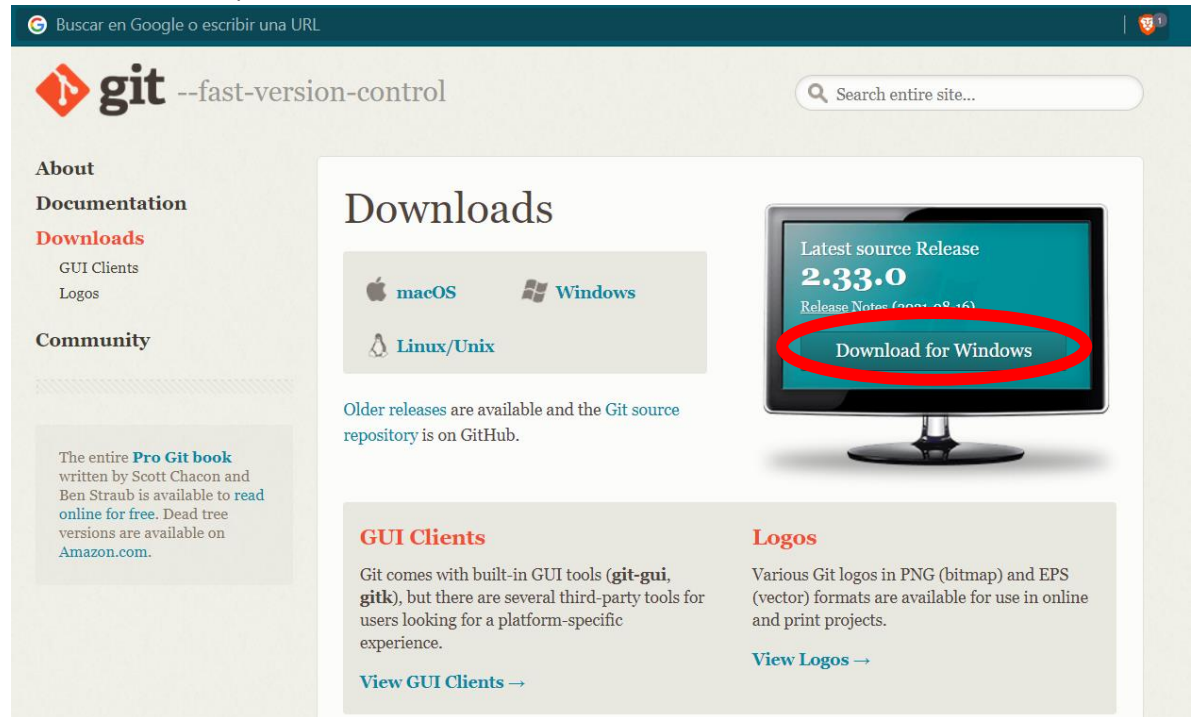
Available commands:
  about                Shows the short information about Composer.
  archive              Creates an archive of this composer package.
  browse               Opens the package's repository URL or homepage in your browser.
  cc                   Clears composer's internal package cache.
  check-platform-reqs Check that platform requirements are satisfied.
  clear-cache           Clears composer's internal package cache.
  clearcache            Clears composer's internal package cache.
  config               Sets config options.
  create-project        Creates new project from a package into given directory.
  depends              Shows which packages cause the given package to be installed.
  diagnose              Diagnoses the system to identify common errors.
  dump-autoload         Dumps the autoloader.
  dumpautoload          Dumps the autoloader.
  exec                 Executes a vendored binary/script.
  fund                 Discover how to help fund the maintenance of your dependencies.
  global               Allows running commands in the global composer dir ($COMPOSER_HOME).
  help                 Displays help for a command
  home                 Opens the package's repository URL or homepage in your browser.
  i                    Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json.
  info                 Shows information about packages.
  init                 Creates a basic composer.json file in current directory.

```

INSTALACIÓN DE GIT

PASO 1: DESCARGAR:

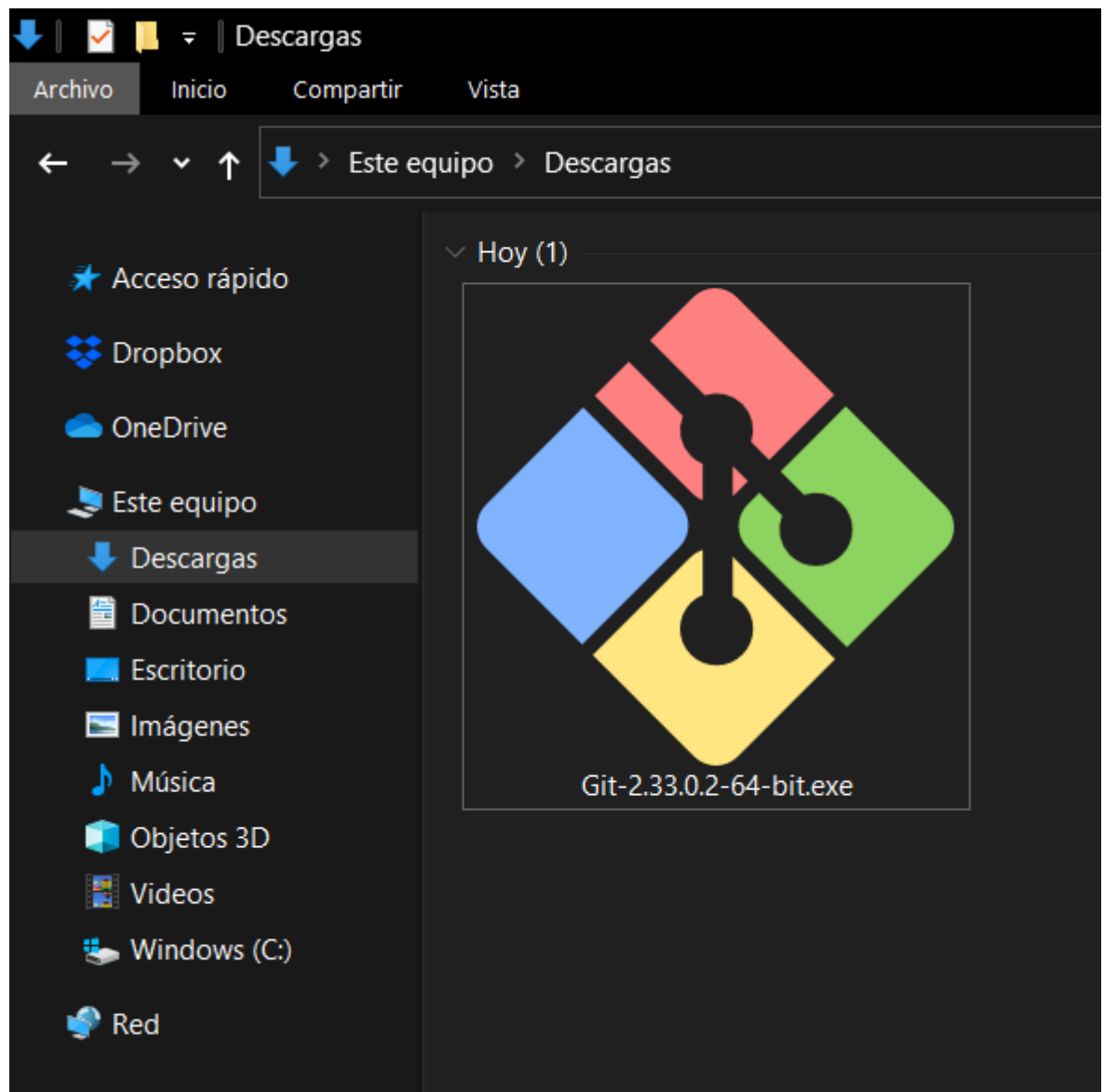
Procedemos a la descarga desde el siguiente enlace <https://git-scm.com/downloads> seleccionamos la opción “Download for Windows”



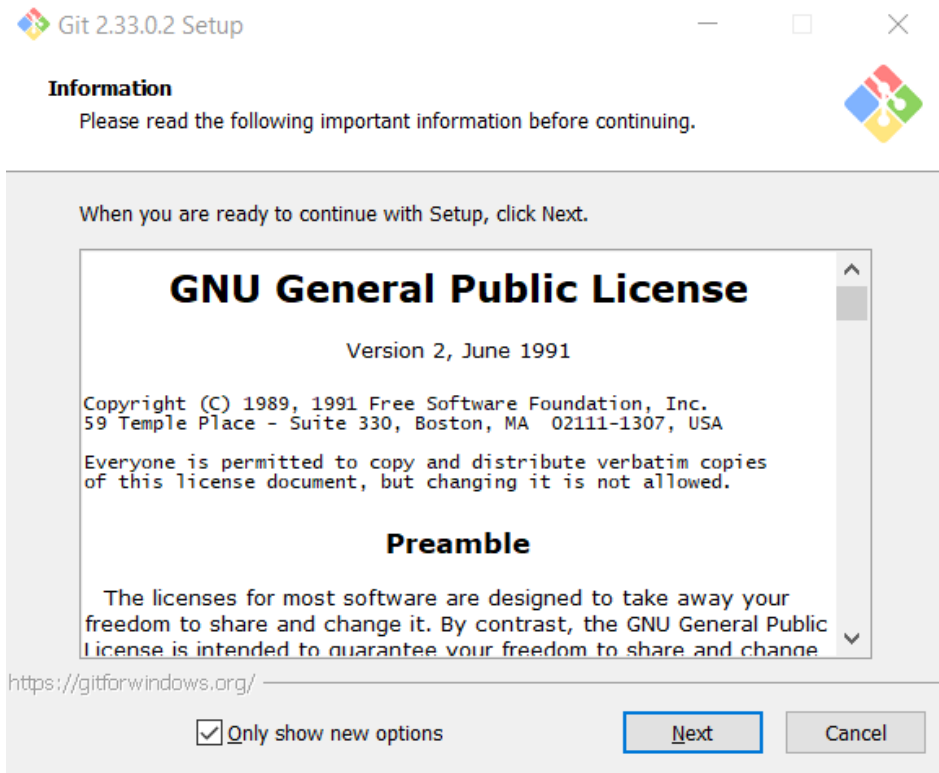
Iniciando una descarga automática del ejecutable para instalar GIT.

PASO 2: INSTALAR

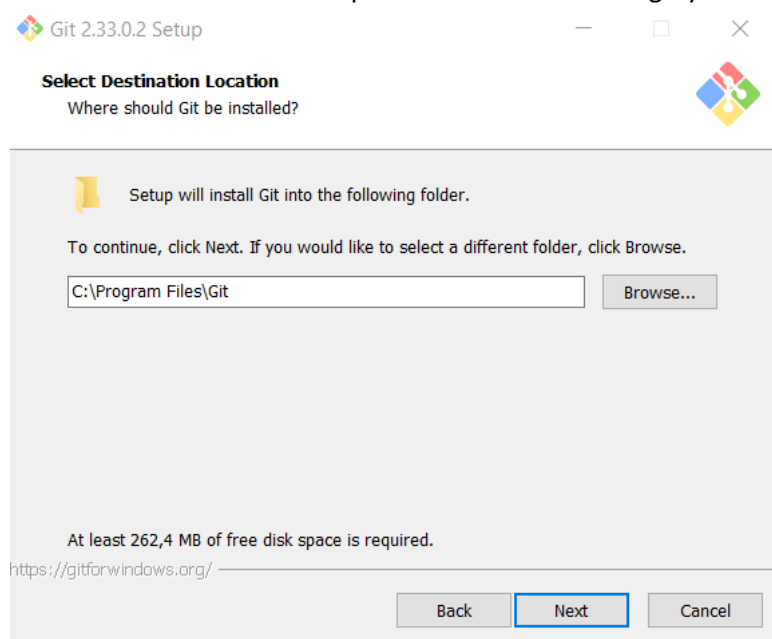
Damos clic en el ejecutable (archivo) que se nos descargo en nuestro caso se llama “Git-2.33.0.2-64-bit.exe”



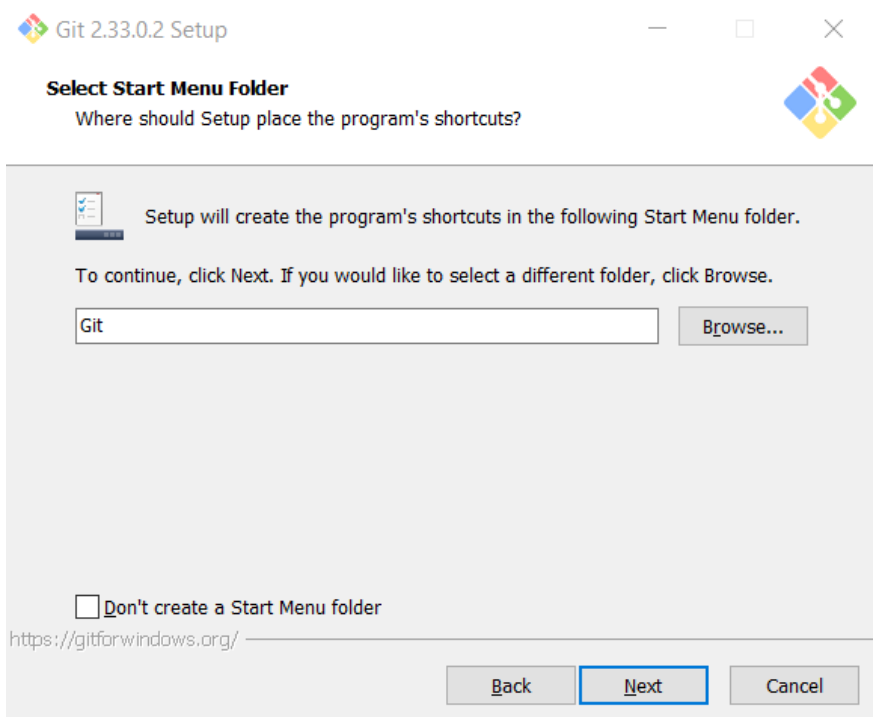
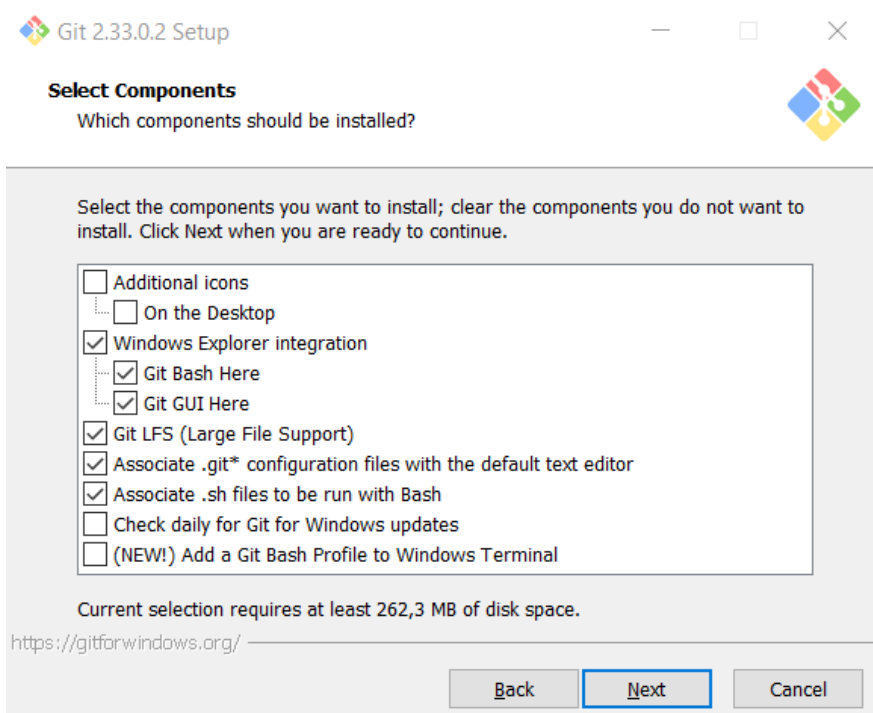
Aceptamos permisos de administrador y procedemos con la instalación dando next

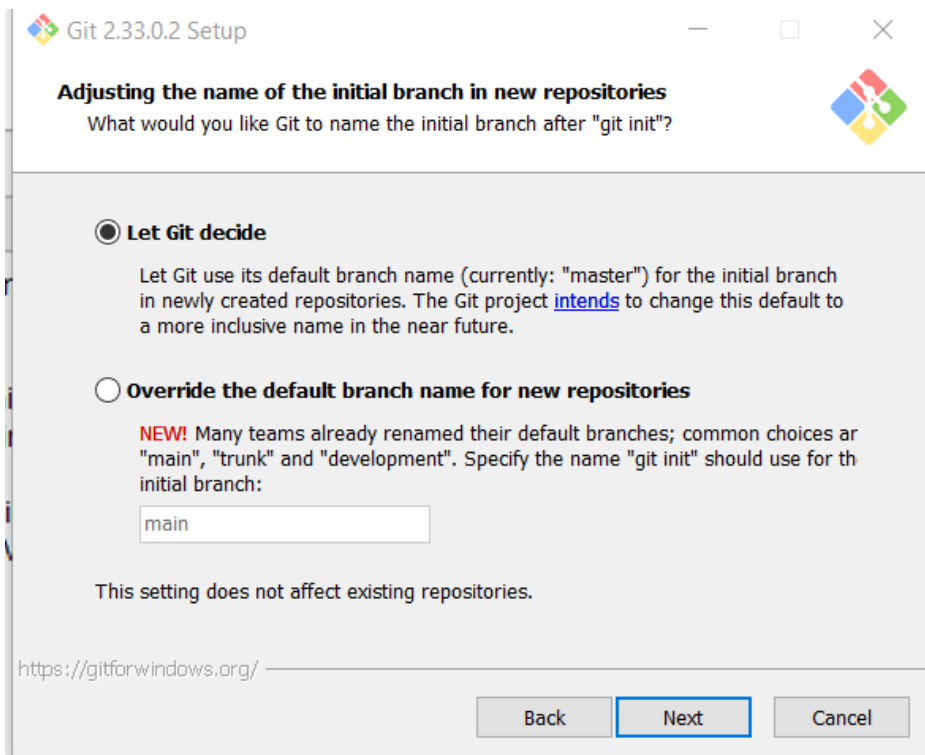
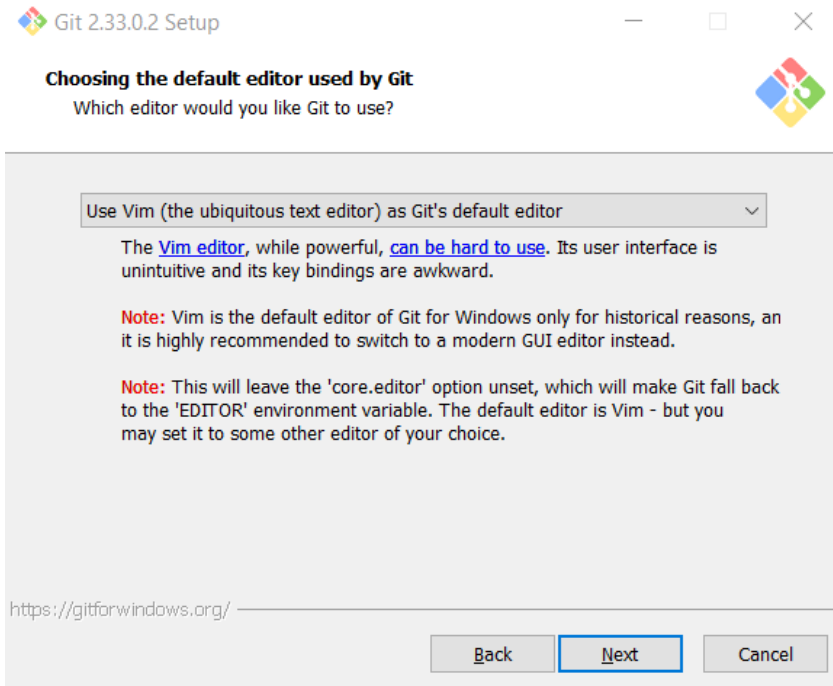


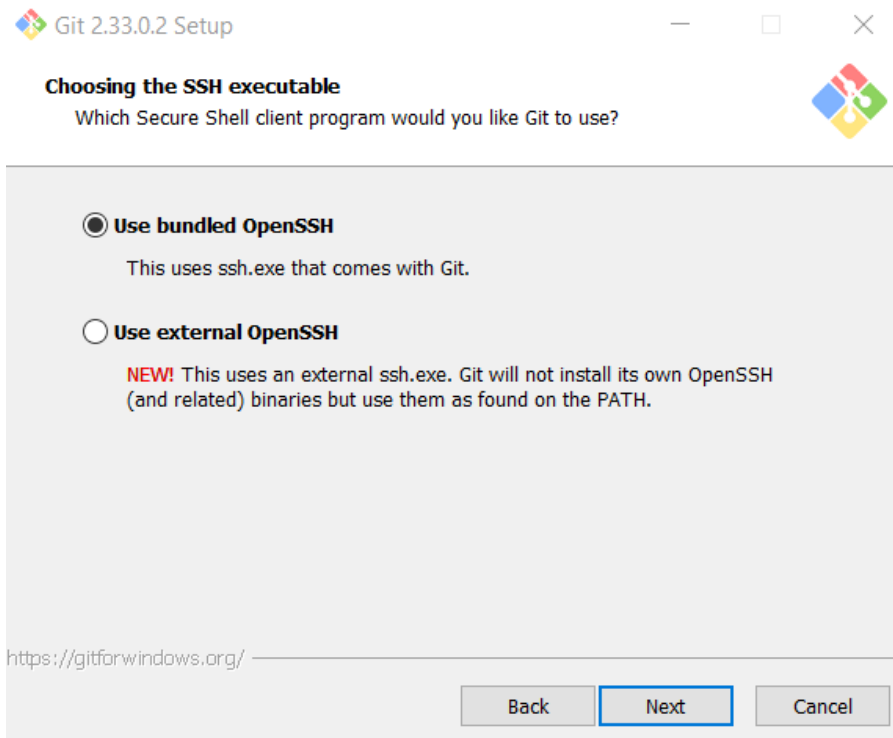
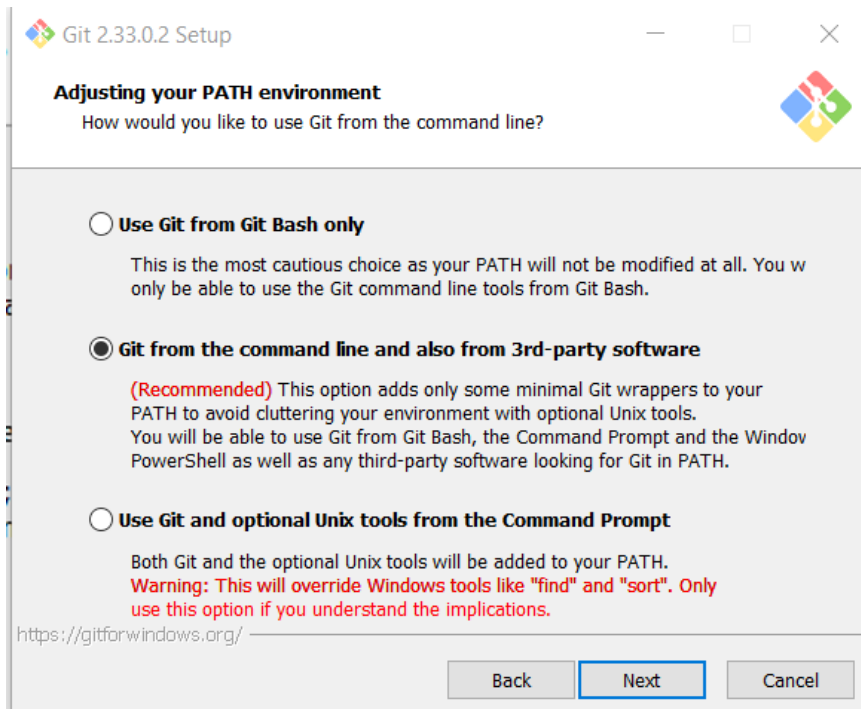
Nos muestra en donde va a quedar instalado nuestro git y damos en Next

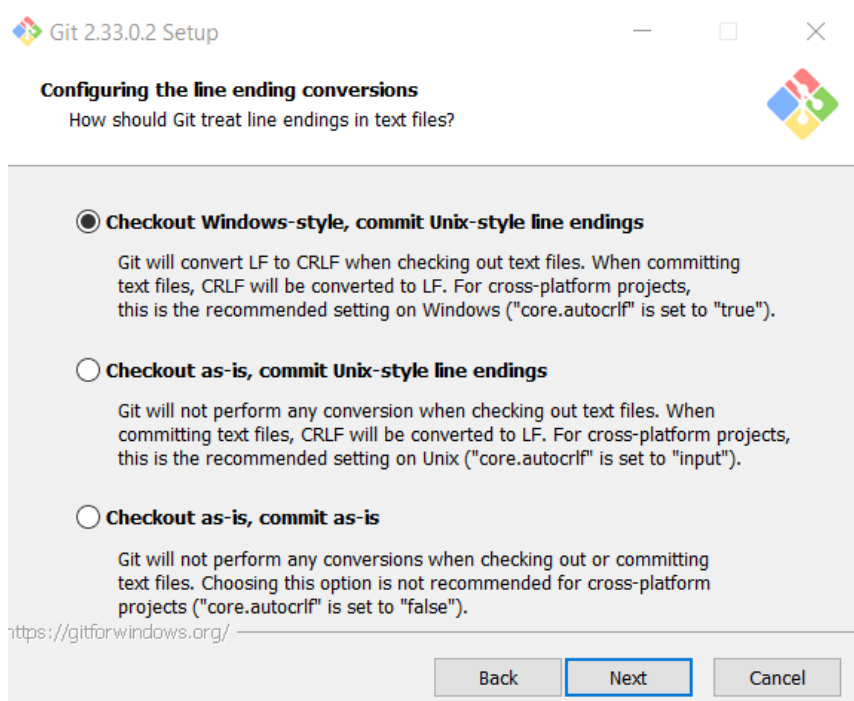
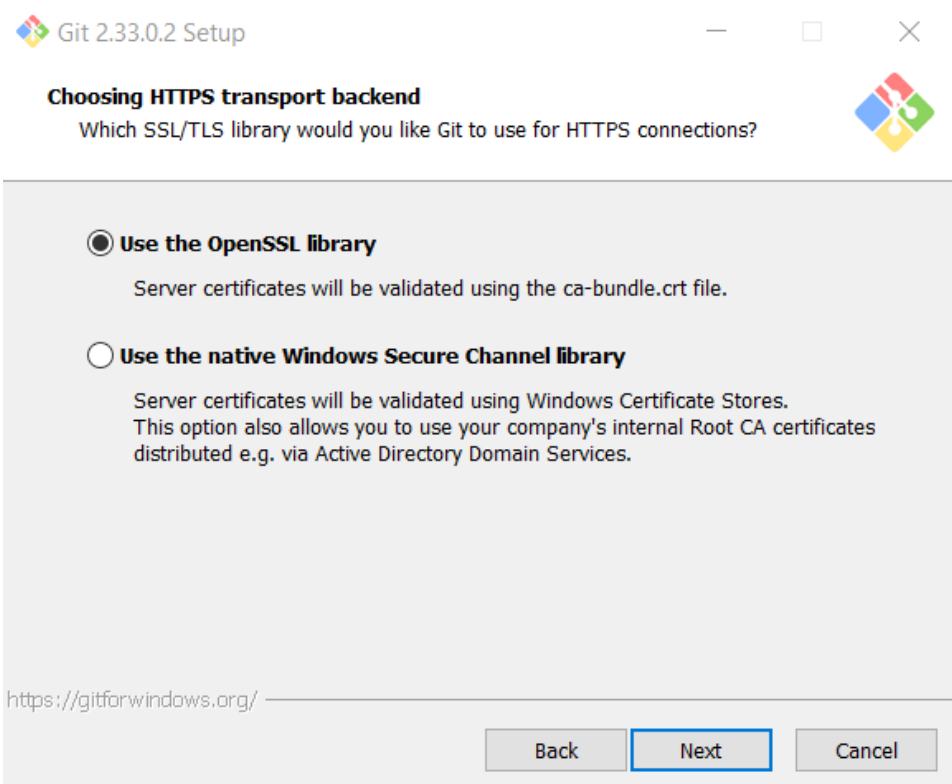


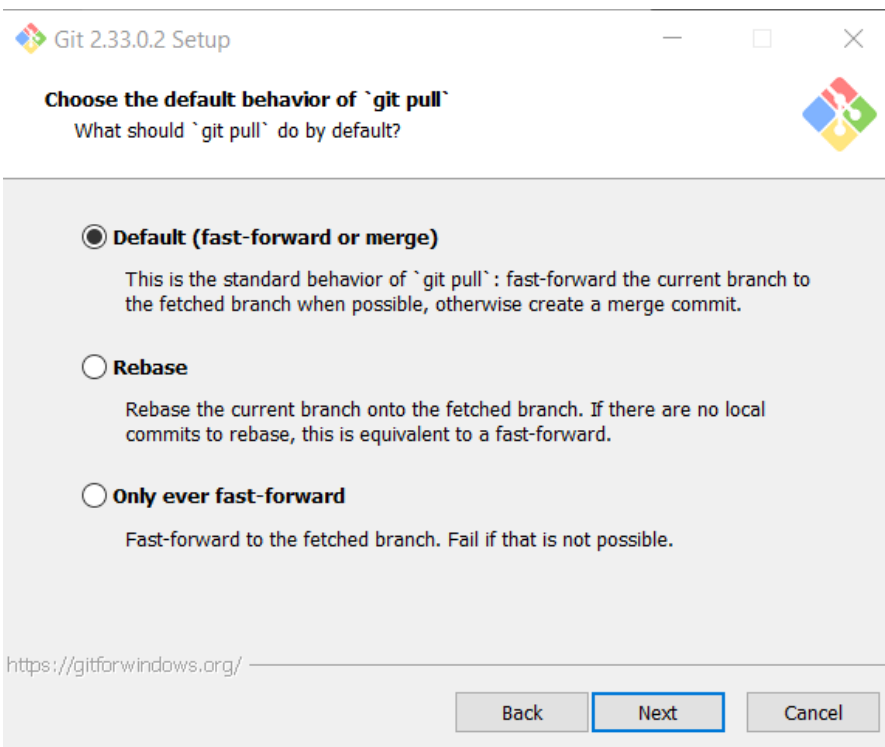
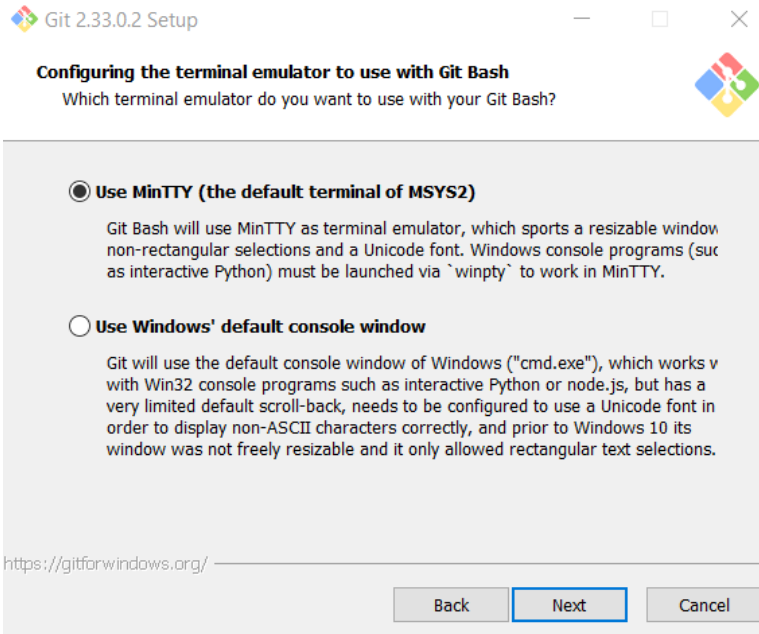
Dejamos todo tal cual aparece sin desmarcar o marcar alguna casilla hasta que nos aparezca la opción de install

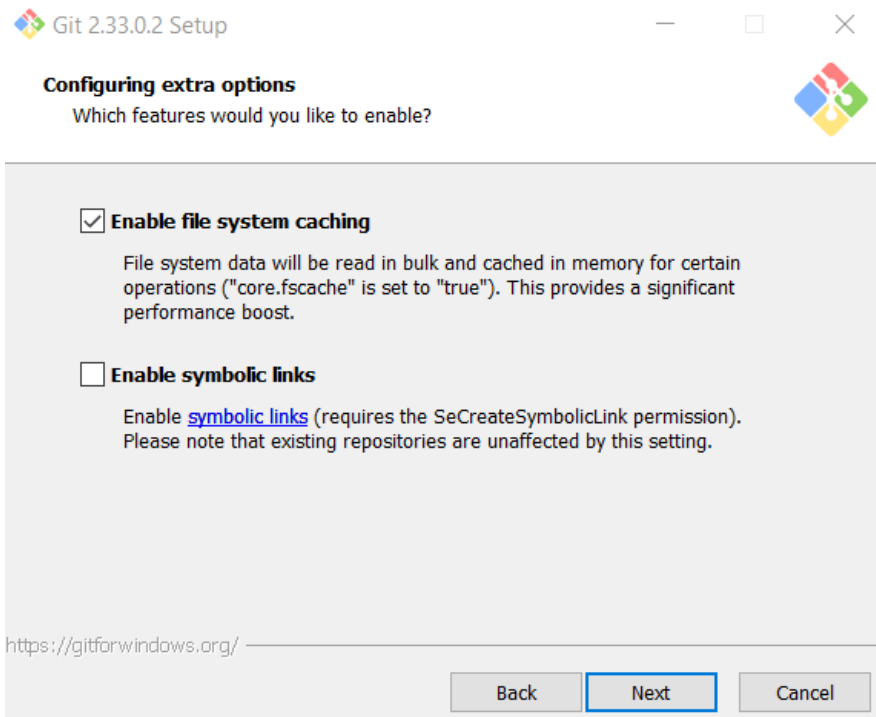
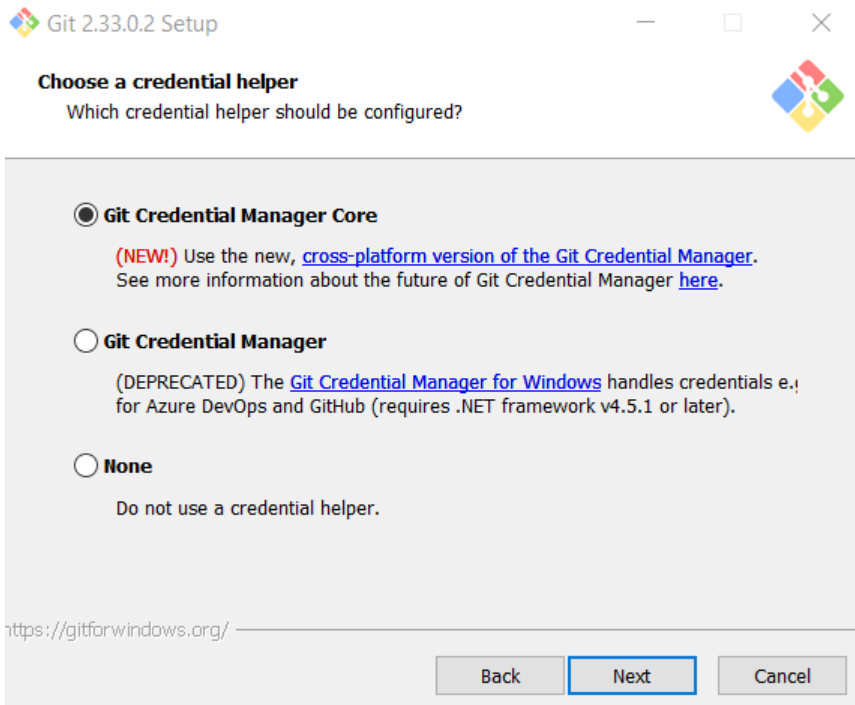


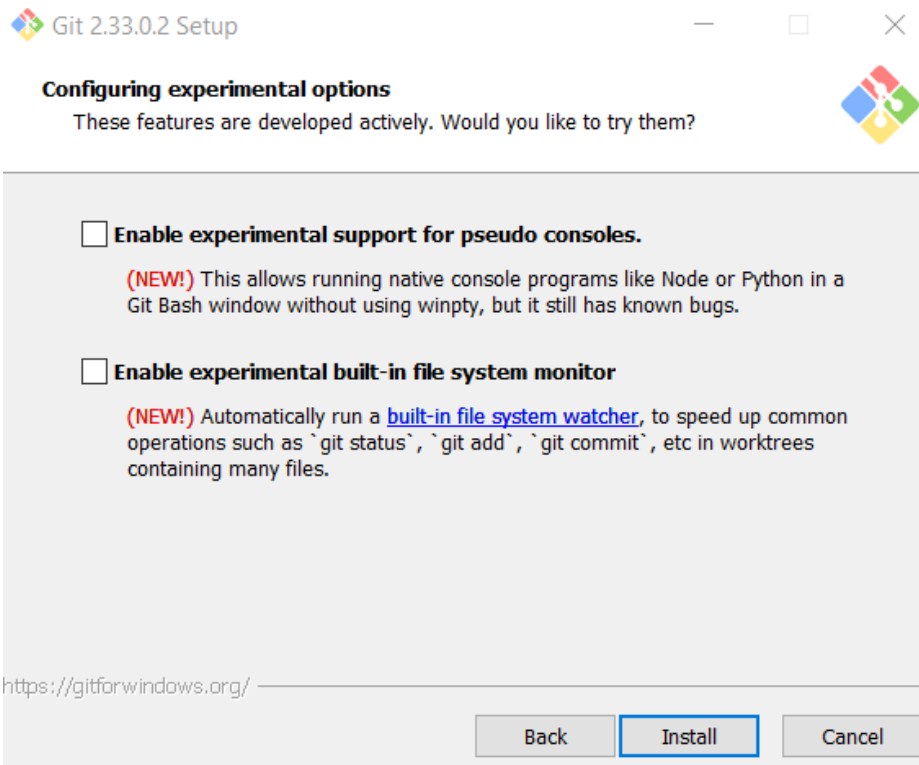




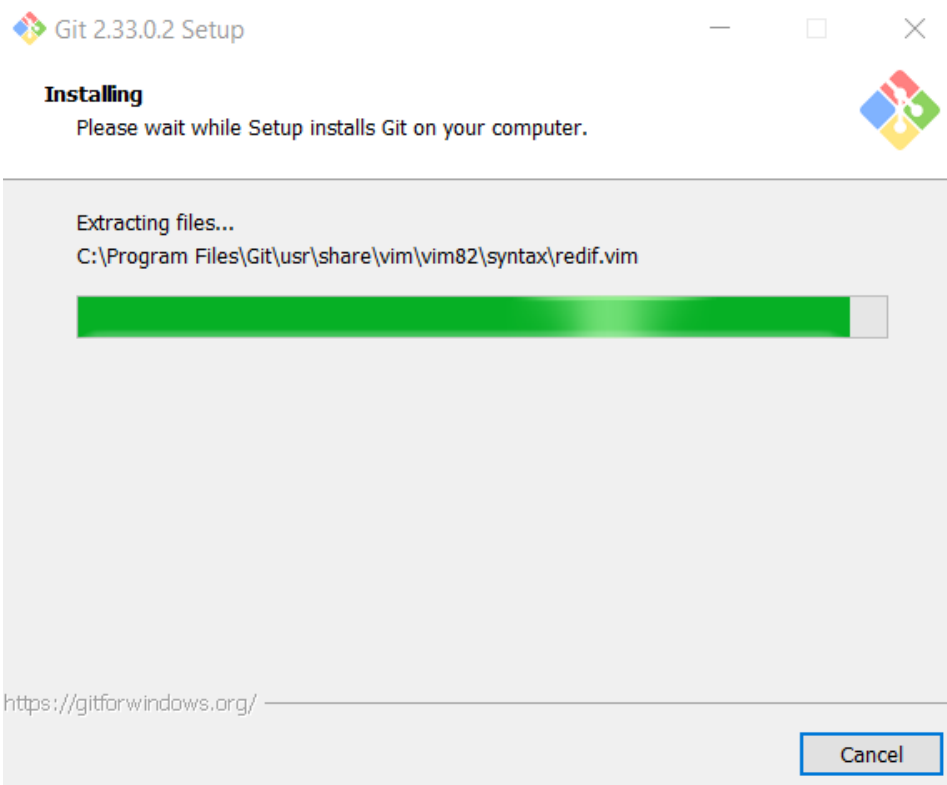






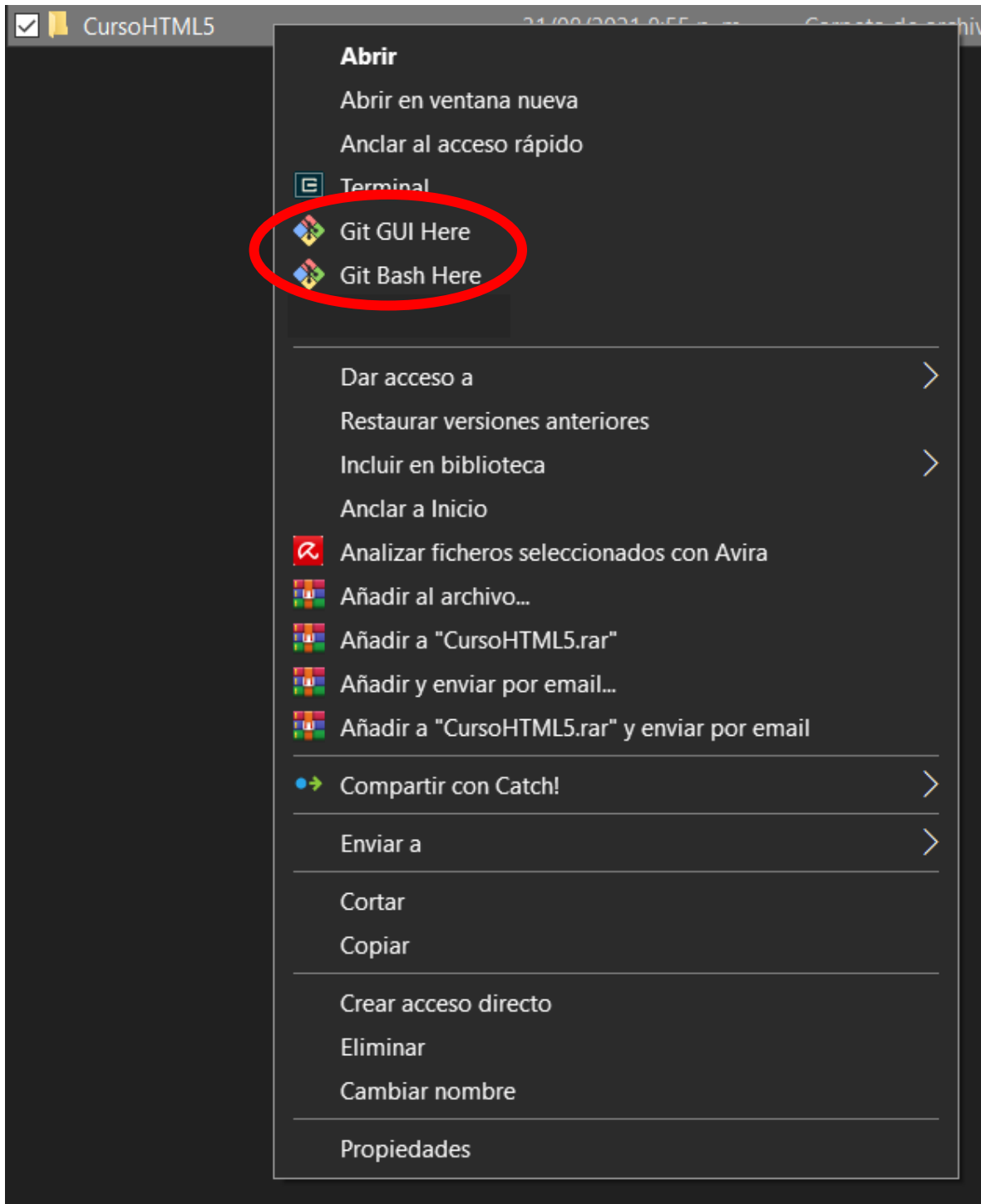


Por último, damos clic en Install, esperamos que se instale



PASO 3: CONFIRMAR INSTALACIÓN

Al finalizar la instalación damos clic en Finish y podremos confirmar que se instalo correctamente al dar clic derecho en alguna carpeta en documentos y aparecerá las opciones de GIT.



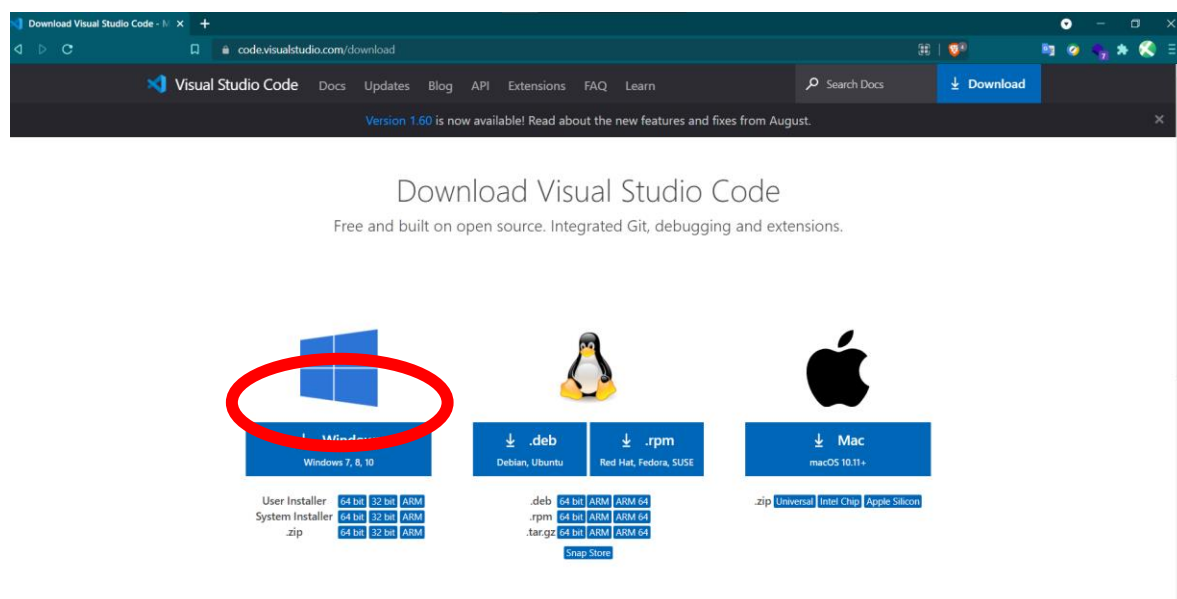
Para acceder a la consola de Git, de clic en “Git Bash Here”



INSTALACIÓN DE VISUAL STUDIO CODE

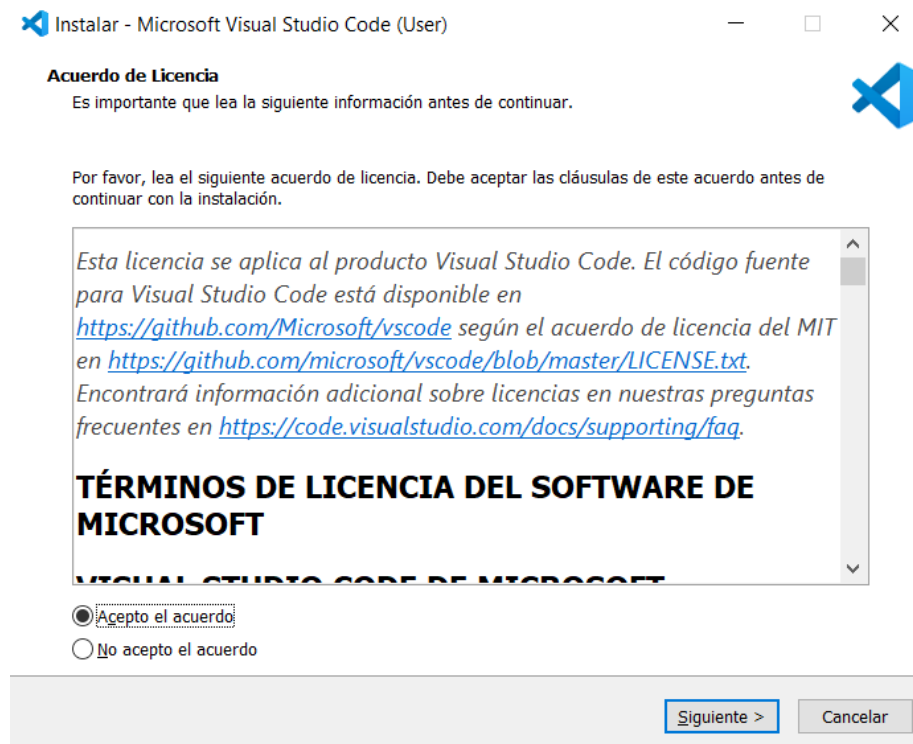
PASO 1: DESCARGAR

Nos dirigimos al siguiente enlace <https://code.visualstudio.com/download> y procedemos con la descarga

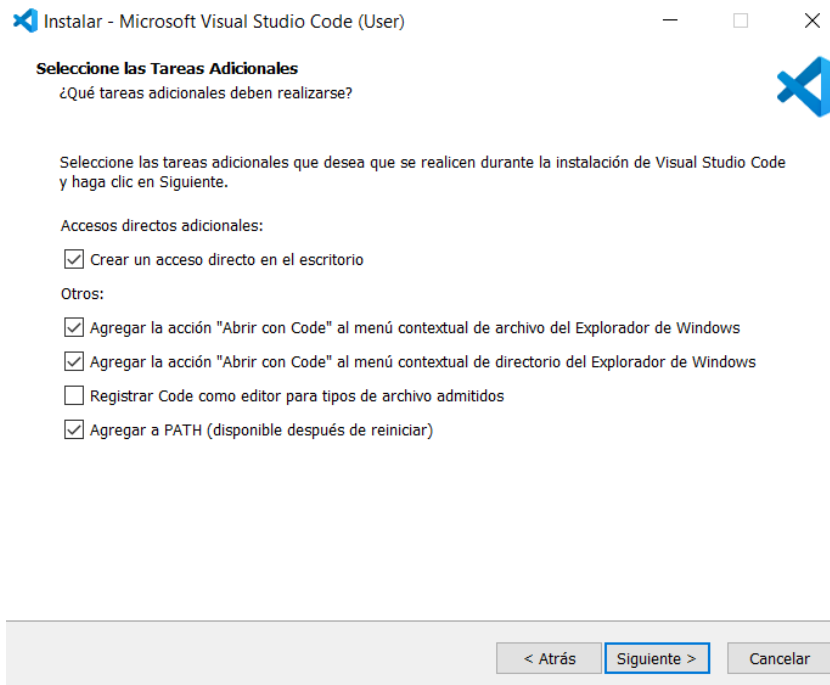


PASO 2: INSTALACIÓN

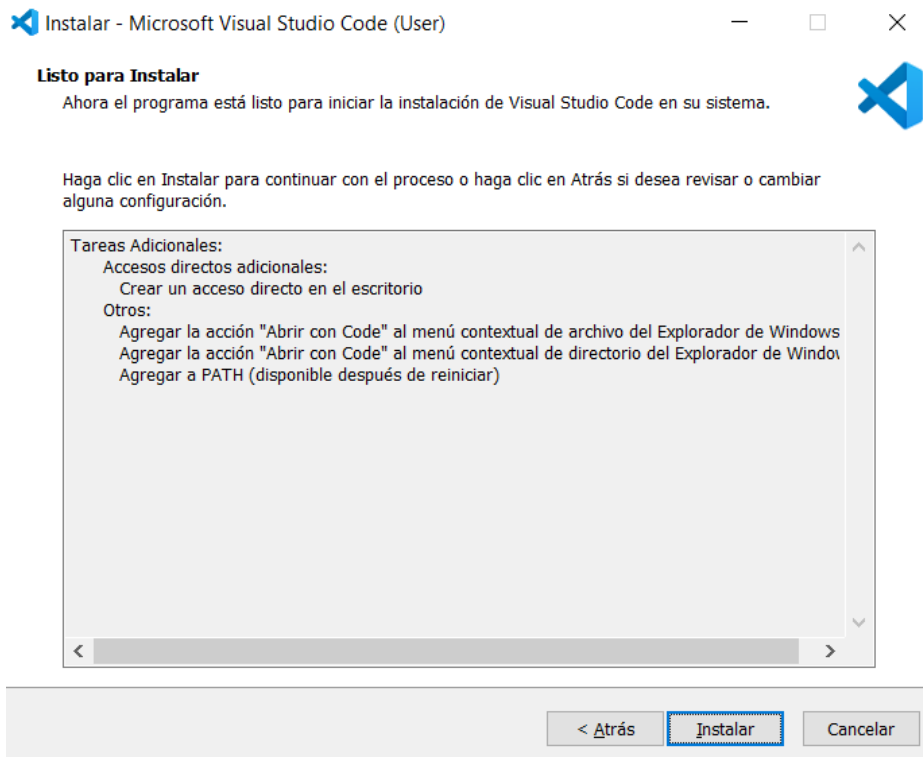
Procedemos a ejecutar el archivo “VSCodeUserSetup-x64-1.60.2.exe”, aceptamos los términos y condiciones y damos en siguiente.



Marcamos las casillas mostradas en la imagen y damos siguiente



Por último, instalar



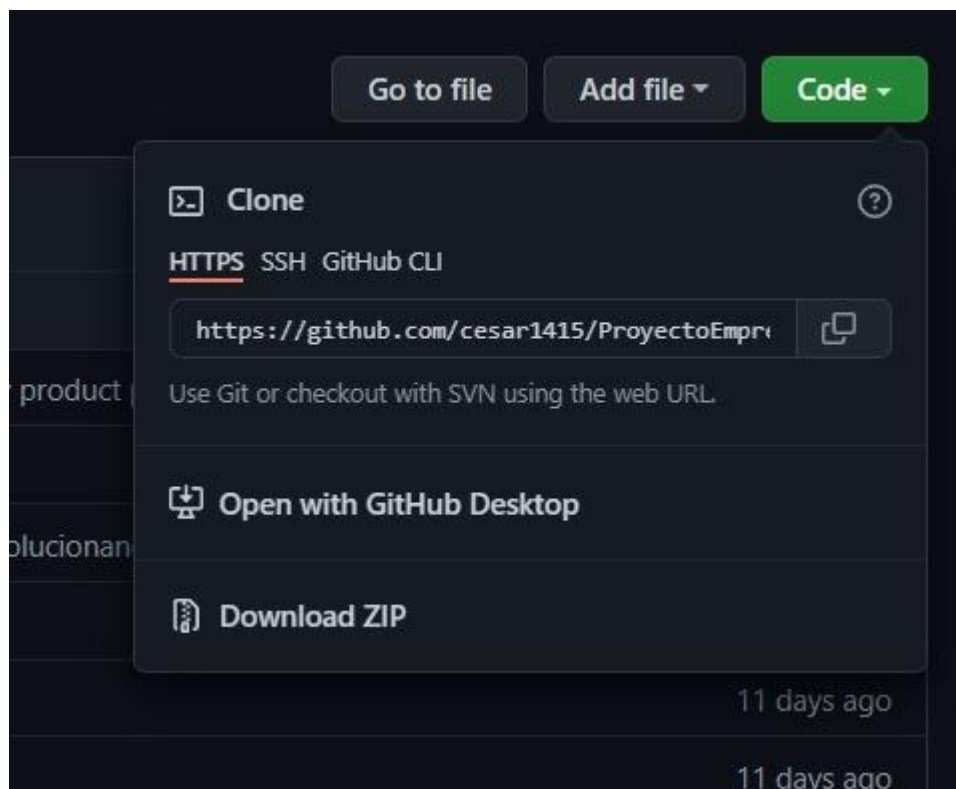
IMPLEMENTACIÓN DEL PROYECTO

CLONACIÓN DE PROYECTOS LARAVEL DESDE GITHUB

COMO HACERLO

1. Copiar la dirección HTTPS del repositorio

Copia la dirección HTTPS del repositorio alojado en la plataforma de código la encuentras en la opción de clonar o descargar luce algo similar a esta <https://github.com/cesar1415/ProyectoEmpresa.git>.



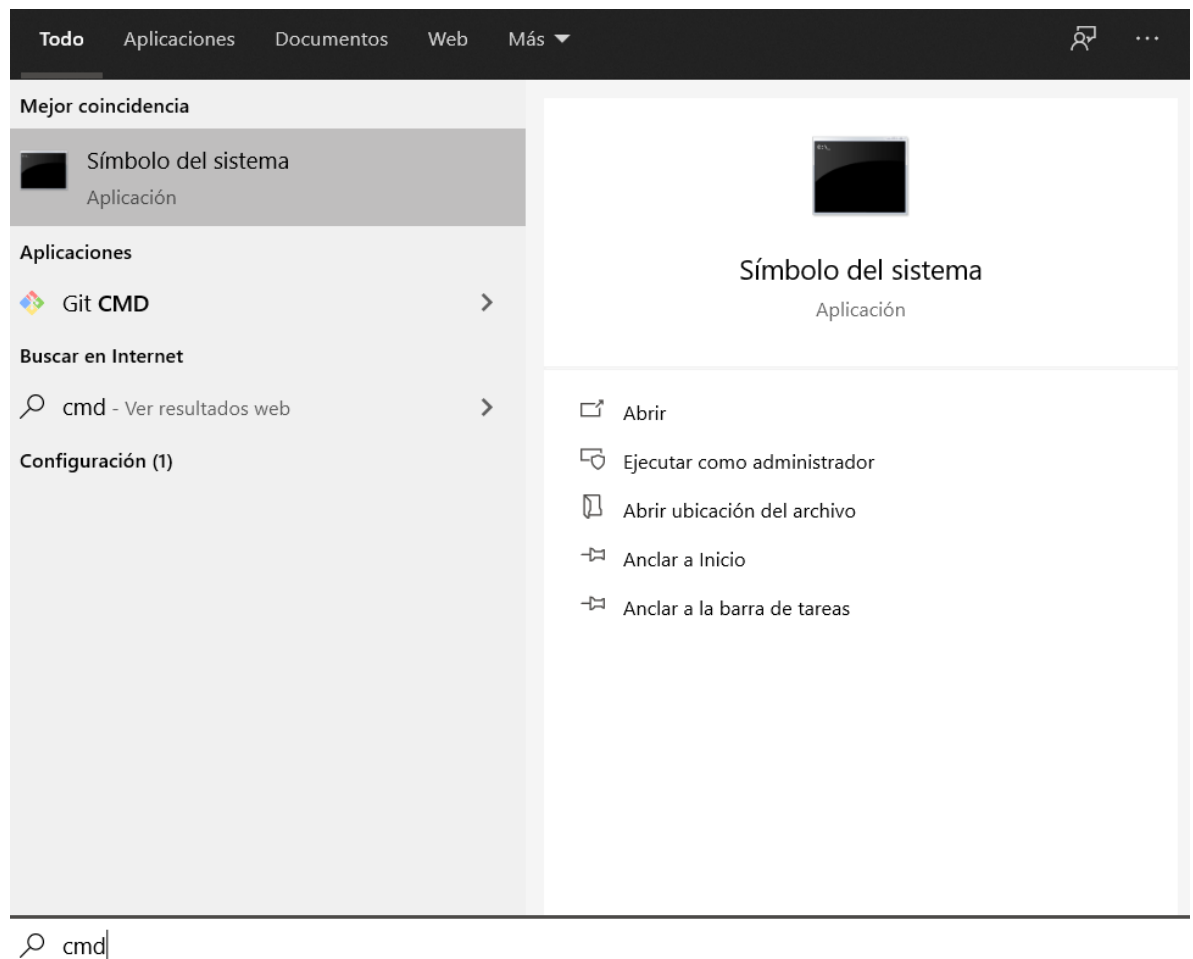
Después de eso abre la consola de comandos de Git (Windows) o la Terminal en sistemas basados en Unix (Mac o Linux) y posíciónate en la carpeta que clonarás tu repositorio con el comando `cd`. Por ejemplo:

Con este comando te ubicas en la carpeta donde deseas clonar el repositorio

`cd "nombre carpeta"`

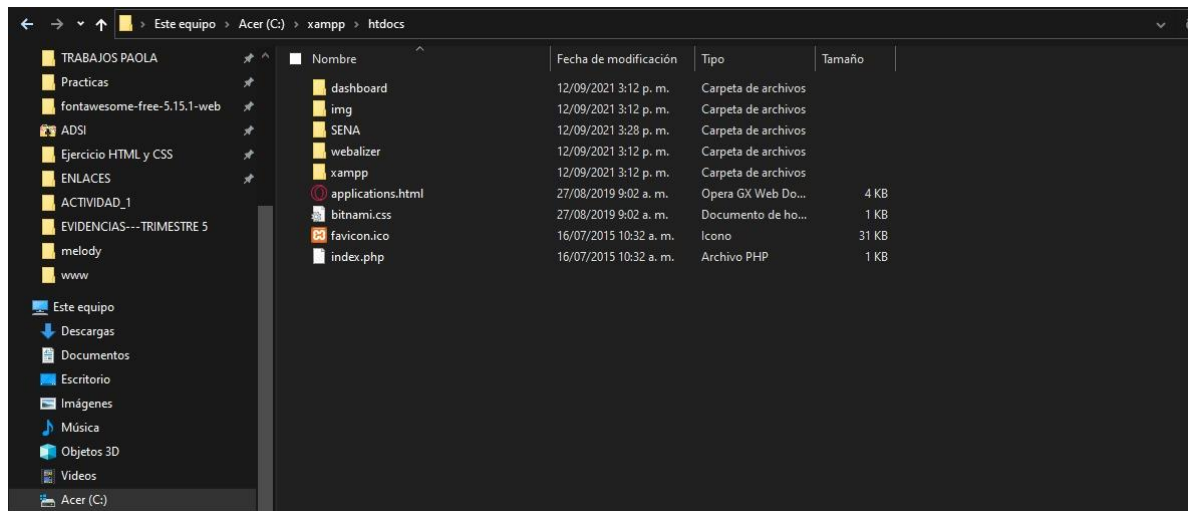
Explicación visual

1.

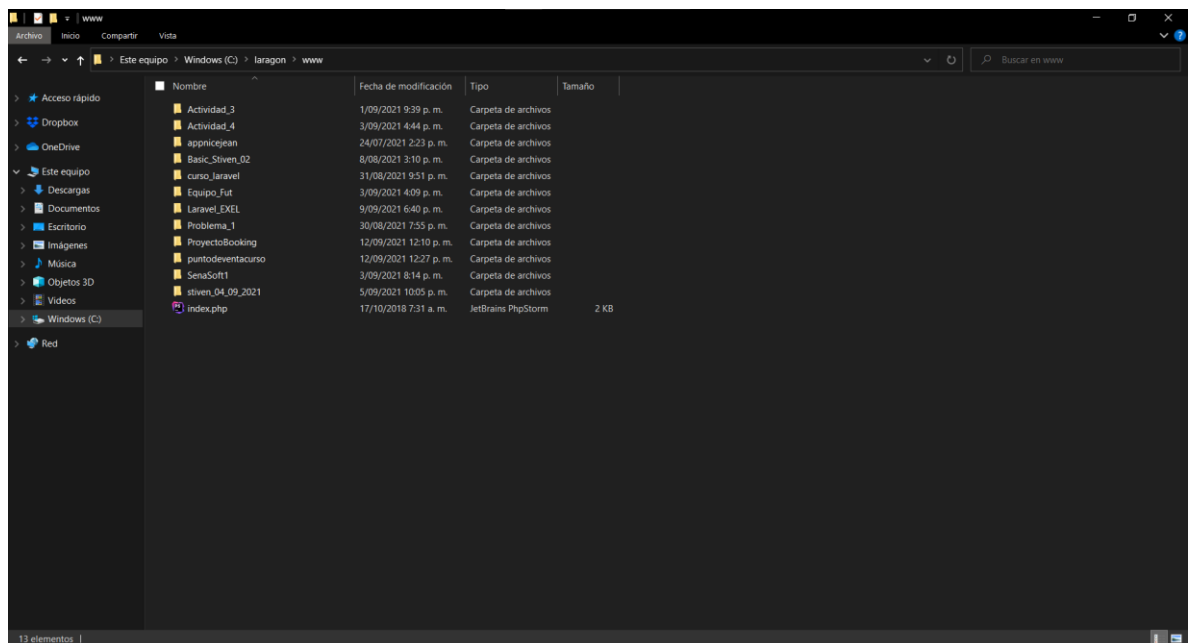


2. buscamos la ruta en la cual deseamos clonar el repositorio (si instalo XAMPP la ruta donde deberá crear la carpeta del repositorio es “C:\xampp\htdocs”, pero si instalo LARAGON la ruta donde deberá clonar el repositorio es “C:\laragon\www”)

Así aparecerá en XAMPP

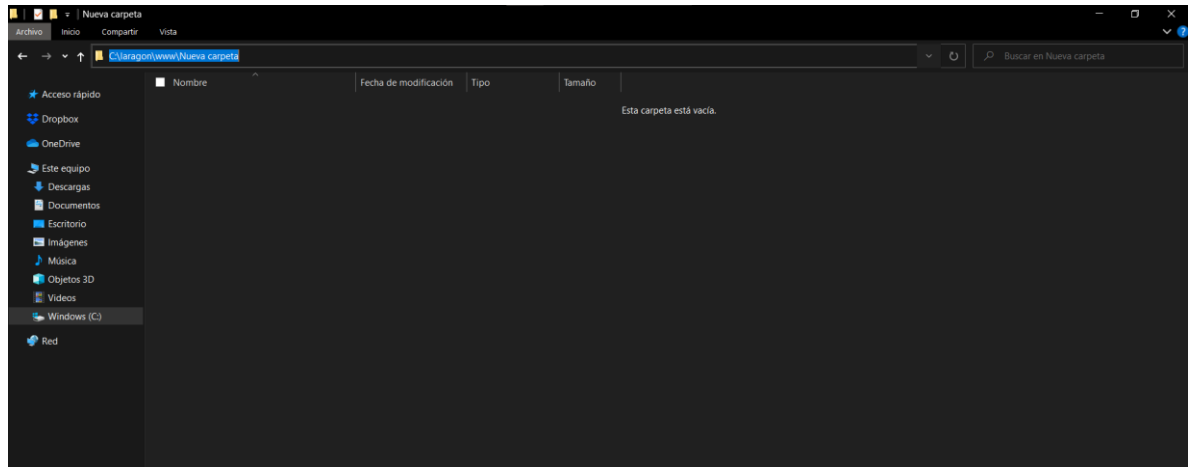


Así aparecerá si uso LARAGON

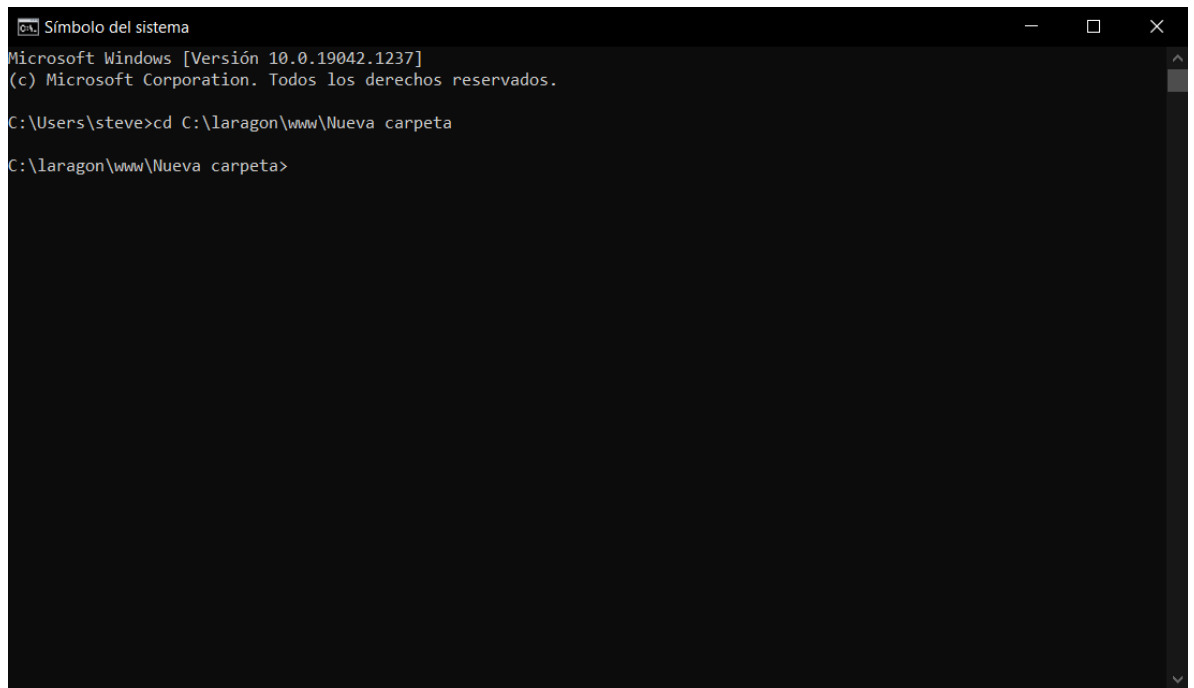


3. Creamos una nueva carpeta, con el nombre que queramos en este caso la llamaremos “Nueva carpeta”

4. Luego de tener la ruta, nos dirigimos a la terminal (cmd) ejecutado en el paso 1 en nuestro caso la ruta es “C:\laragon\www\Nueva carpeta”

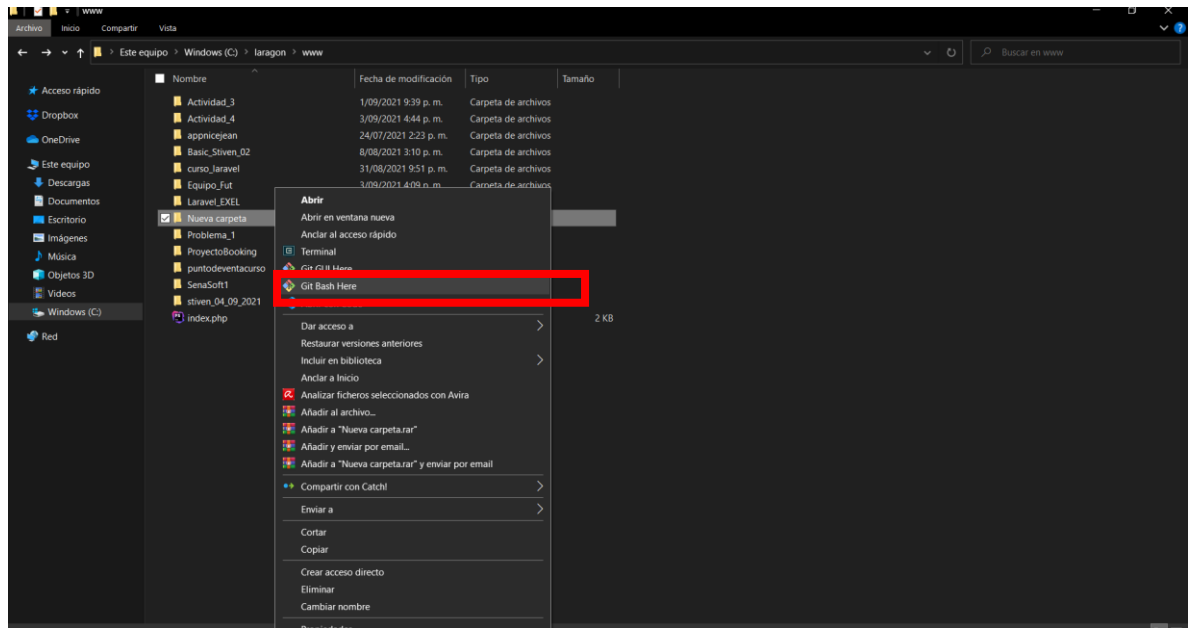


Ingresamos el comando “cd C:\laragon\www\Nueva carpeta”

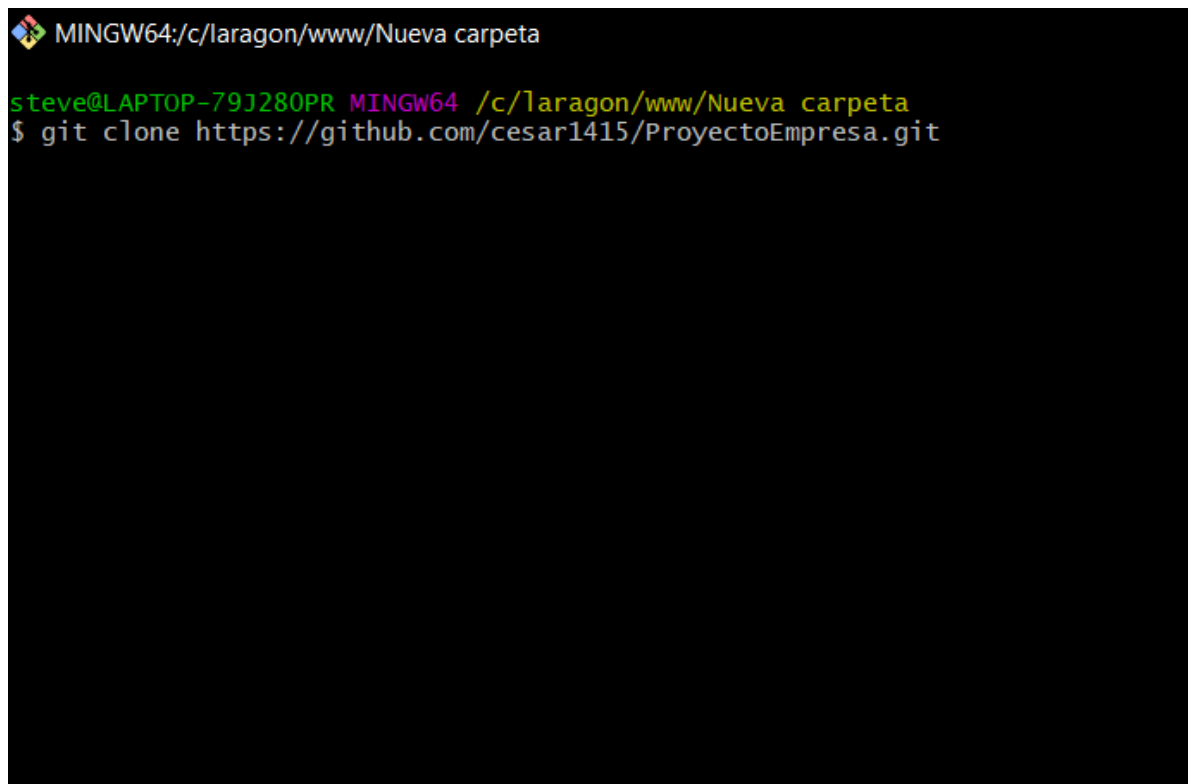


2. Clonar repositorio

Ejecuta el siguiente comando (en git bash):



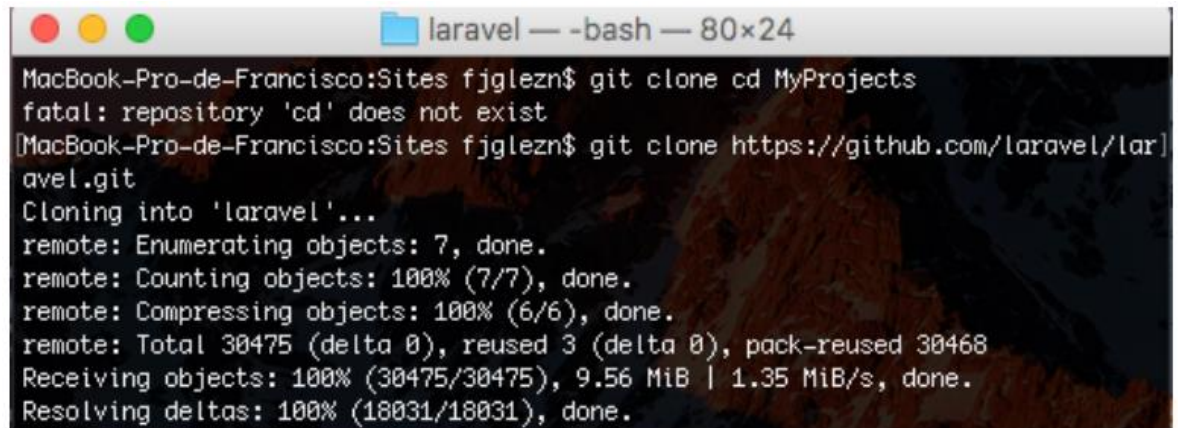
git clone <https://github.com/cesar1415/ProyectoEmpresa.git>



Por ejemplo:

git clone <https://github.com/laravel/laravel.git>

Si todo fue bien, tendrás un resultado similar a la imagen de abajo.

A terminal window titled 'laravel — -bash — 80x24' on a macOS system. The user is at the prompt 'MacBook-Pro-de-Francisco:Sites fjglezn\$'. They first run 'git clone cd MyProjects', which fails with the message 'fatal: repository 'cd' does not exist'. They then run 'git clone https://github.com/laravel/laravel.git', which succeeds. The output shows the cloning process: 'Cloning into 'laravel'...', 'remote: Enumerating objects: 7, done.', 'remote: Counting objects: 100% (7/7), done.', 'remote: Compressing objects: 100% (6/6), done.', 'remote: Total 30475 (delta 0), reused 3 (delta 0), pack-reused 30468', 'Receiving objects: 100% (30475/30475), 9.56 MiB | 1.35 MiB/s, done.', and 'Resolving deltas: 100% (18031/18031), done.'

```
MacBook-Pro-de-Francisco:Sites fjglezn$ git clone cd MyProjects
fatal: repository 'cd' does not exist
MacBook-Pro-de-Francisco:Sites fjglezn$ git clone https://github.com/laravel/laravel.git
Cloning into 'laravel'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 30475 (delta 0), reused 3 (delta 0), pack-reused 30468
Receiving objects: 100% (30475/30475), 9.56 MiB | 1.35 MiB/s, done.
Resolving deltas: 100% (18031/18031), done.
```

3. Instalar paquetes o dependencias

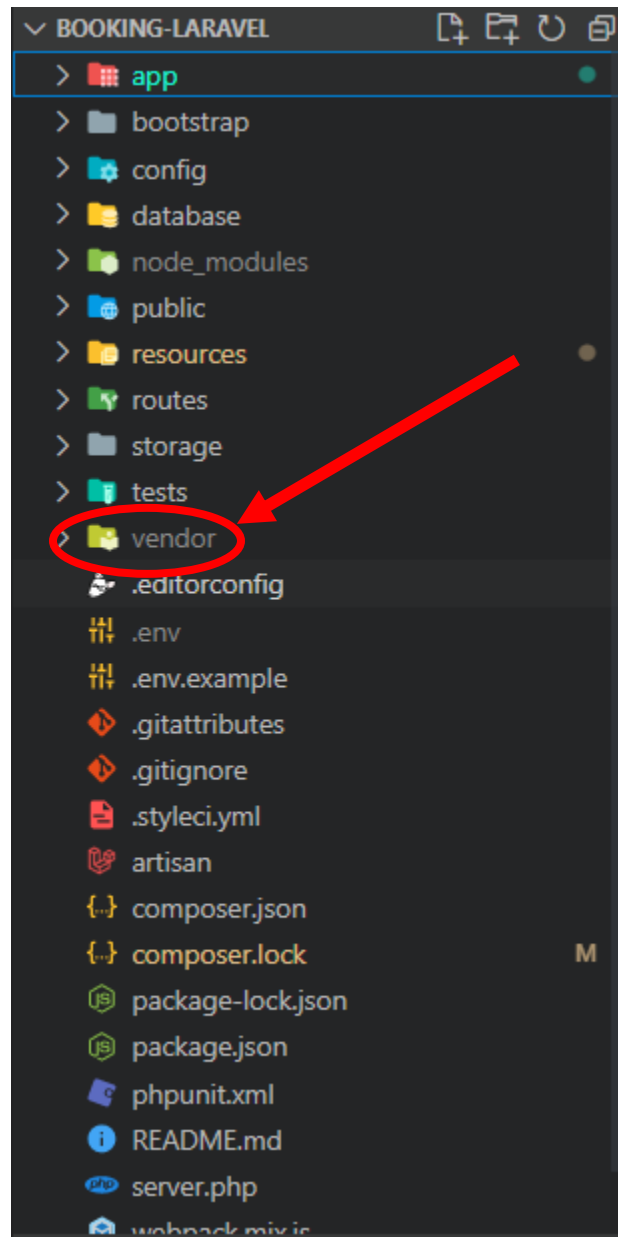
- Ahora entra en la carpeta del proyecto que acabas de clonar desde la Terminal o consola de Git con el comando `cd` y el nombre del proyecto justo como se muestra a continuación
- `cd` nombre del proyecto clonado
- Ya adentro del proyecto empezaremos hacer los siguientes ajustes para que este pueda correr en nuestro equipo
- Instalar dependencias
- Crear una base de datos
- Crear el archivo `.env`
- Generar una clave
- Migrar y sembrar la base de datos
- Instalar dependencias
- Instalaremos con Composer, el manejador de dependencias para PHP, las dependencias definidos en el archivo `composer.json`. Para ello abriremos una terminal en la carpeta del proyecto y ejecutaremos:

```
composer install
```

Esto te descargará e instalará las dependencias utilizadas por el proyecto. Si todo sale bien te saldrá algo similar a la imagen que todo esta correcto


```
laravel — -bash — 80x24
to use PSR-7 bridging features (^1.1).)
laravel/framework suggests installing wilddbit/swiftmailer-postmark (Required to
use Postmark mail driver (^3.0).)
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to
work.)
psy/psysh suggests installing hoa/console (A pure PHP readline implementation. Y
ou'll want this if your PHP install doesn't already support readline or libedit.
)
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.6.0)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0)
phpunit/phpunit suggests installing ext-xdebug (*)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
MacBook-Pro-de-Francisco:laravel fjglezn$
```

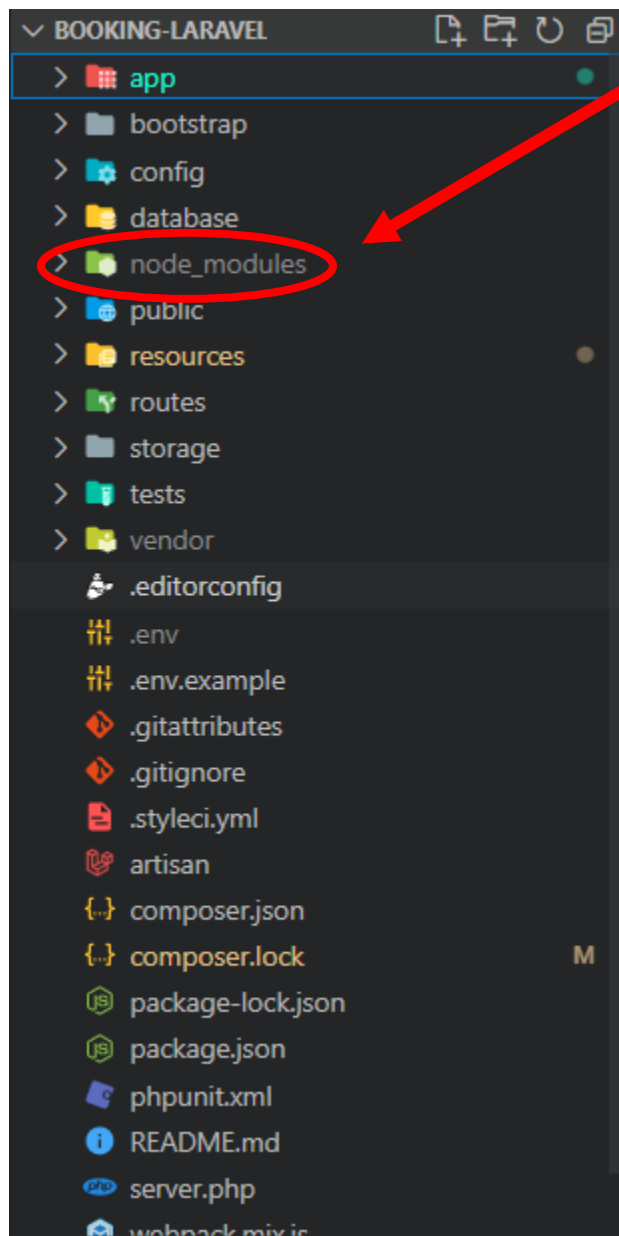
Vemos cómo se ha creado la carpeta **vendor**.



También debemos instalar las dependencias de NPM definidas en el archivo package.json con:

```
npm install
```

Y en esta ocasión vemos cómo se crea la carpeta node_modules.



Crear el archivo .env

El siguiente paso es copiar el contenido del archivo .env.example en un nuevo archivo con el nombre .env. El archivo .env.example lo encuentras en la raíz del proyecto de Laravel. Pero espera, te tengo una solución más rápida y sencilla utilizando la Terminal. Dentro de la carpeta de tu proyecto ejecuta el siguiente comando:

```
cp .env.example .env
```

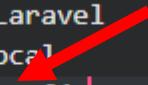
ejemplo archivo .env

```
.....  
DB_CONNECTION=mysql  
DB_HOST=localhost  
DB_PORT=3306  
DB_DATABASE=laravel_project  
DB_USERNAME=root  
DB_PASSWORD=secret  
....
```

Generar una clave

La **clave de aplicación** es una cadena aleatoria almacenada en la clave APP_KEY dentro del archivo **.env**. Notarás que también falta.

```
❗ .env  
1 APP_NAME=Laravel  
2 APP_ENV=local  
3 APP_KEY=base64:  
4 APP_DEBUG=true  
5 APP_URL=http://localhost  
6
```

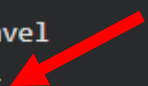


Para crear la nueva clave e insertarla automáticamente en el **.env**, ejecutaremos:

```
php artisan key:generate
```

Luego de ejecutar este comando en la terminal dentro de tu proyecto puedes ver el archivo **.env** que se ha creado

```
❗ .env  
1 APP_NAME=Laravel  
2 APP_ENV=local  
3 APP_KEY=base64:JKbmwqPp7SziPEnqxBJr13Eut25ii1/JNNNLRZSPKeQ=  
4 APP_DEBUG=true  
5 APP_URL=http://localhost  
6
```



Crear una base de datos que soporte Laravel

Para crear la base de datos es importante que te dirijas al archivo .env de tu proyecto y te ubiques en la siguiente línea de código

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=
DB_USERNAME=root
DB_PASSWORD=
```

Una vez estés ubicado puedes ponerle el nombre de la base de datos que tu desees

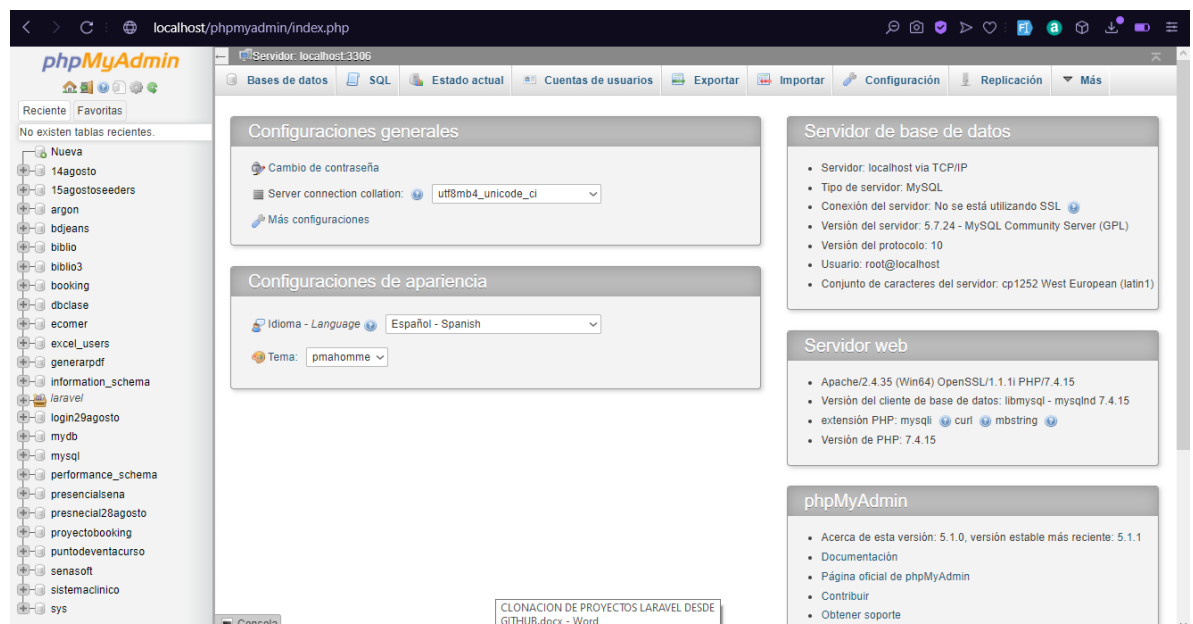
```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=ProyectoBooking
DB_USERNAME=root
DB_PASSWORD=
```

Hecho este paso es necesario que te vayas al administrador de base de datos

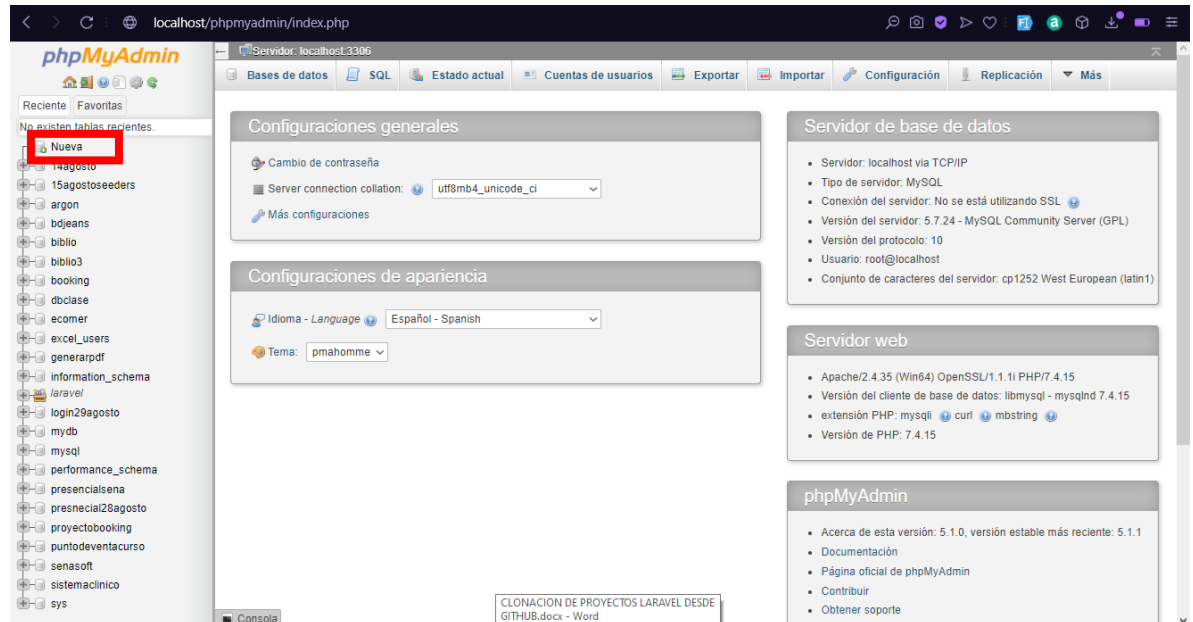
PhpMyAdmin ubicándote en tu navegador y escribiendo la ruta

<http://localhost/phpmyadmin/index.php>

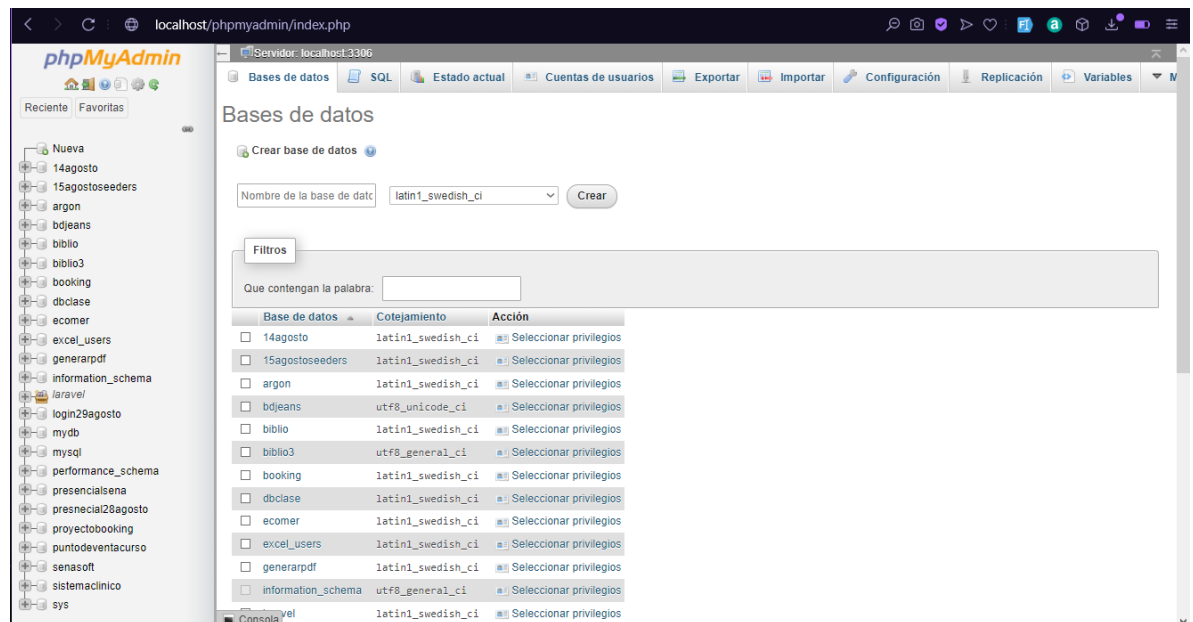
Cuando ingreses esta ruta se te abrirá una ventana como la que se muestra a continuación



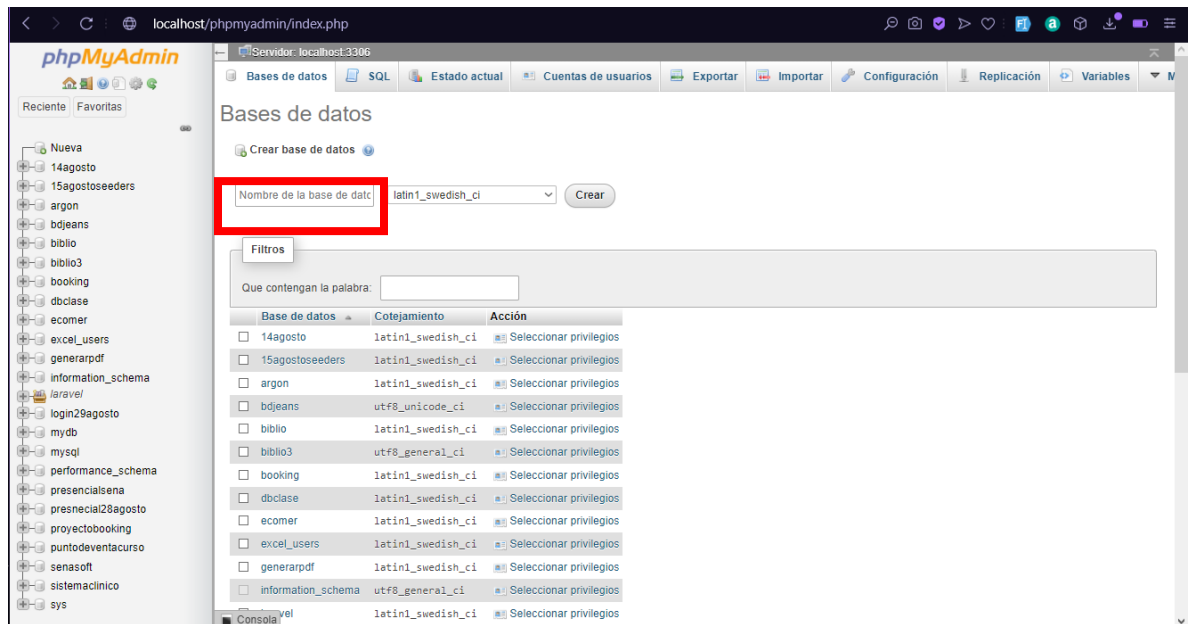
Luego crea una nueva base de datos dando clic en la opción “Nueva”



Al dar clic en esa opción se te abrirá una nueva ventana



Te ubicas en el apartado nombre de base de datos



Ahí escribes el mismo nombre de la base de datos que pusiste en tu archivo .env y das clic en la opción crear

Migraciones a bases de datos

Hecho estos pasos puedes digitar el siguiente comando en tu terminal para ejecutar las migraciones

```
php artisan migrate
```

Si tuviéramos que incluir o crear nuevas migraciones utilizaríamos:

```
php artisan migrate:refresh  
php artisan migrate:fresh //borra y crea todas las tablas de nuevo
```

Con esto ya tendría que correr sin problemas la aplicación de Laravel que hemos clonado. Para correr nuestro proyecto clonado digita **php artisan serve**

```
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.6.0)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0)
phpunit/phpunit suggests installing ext-xdebug (*)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
[MacBook-Pro-de-Francisco:laravel fjglezn$ cp .env.example .env
[MacBook-Pro-de-Francisco:laravel fjglezn$ php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

Es importante que copies la url que se genera en tu terminal cuando digitas el comando `php artisan serve`

Esta url es necesario copiarla en tu navegador para que veas el proyecto iniciado

Laravel

[DOCS](#) [LARACASTS](#) [NEWS](#) [BLOG](#) [NOVA](#) [FORGE](#) [GITHUB](#)

Prueba exitosa, el proyecto de Laravel se está ejecutando correctamente