**MANUAL TECNICO DEL SISTEMA.**

**DESARROLLO DE HABILIDADES PARA LA CREACIÓN E IMPLEMENTACIÓN DE SOFTWARE PARA LA COMERCILIZACIÓN DE PRODUCTOS USANDO PASARELA DE PAGOS Y FACTURACIÓN ELECTRONICA EN EL MUNICIPIO DE PITALITO- HUILA.**

**ADSI -2252407**

**TECNOLOGO EN ANÁLISIS EN DESARROLLO DE SISTEMAS DE INFORMACIÓN**

**CENTRO DE GESTIÓN Y DESARROLLO SOSTENIBLE SURCOLOMBIANO**

**REGIONAL – HUILA**

**PITAITO**

**2022**

Tabla de contenido

**OBJETIVOS1**

OBJETIVO GENERAL**1.1**

OBJECTIVOS ESPECIFICOS**1.2**

**INTRODUCCIÓN2**

**REQUISITOS TECNICOS MINIMOS DE HARDWARE 3**

**REQUISITOS TECNICOS MINIMOS DE SOFTWARE 4**

**REQUERIMIENTOS5**

**HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO 6**

INSTALACIÓN DE SERVIDOR WEB LOCAL**6.1**

INSTALACIÓN DE GIT **6.2**

VINCULACIÓN DE REPOSITORIO GIT GITHUB **6.3**

**ESTRUCTURA Y DIAGRAMAS7**

ESTRUCTURA**7.1**

DIAGRAMAS**7.2**

**DICCIONARIO DE DATOS8**

**MANUAL DE USUARIO SCANNER9**

**BIBLIOGRAFÍA10**

1. **OBJECTIVOS**

**1.1 Objetivo General**

Brindar la información necesaria a los administradores que llevaran a cabo el control de la plataforma acerca de los programas y herramientas utilizadas para el desarrollo y configuración del aplicativo SIREP.

**1.2 Objetivos Específicos**

* Especificar los requisitos de hardware y software necesarios para la instalación y funcionamiento del sistema.
* Ilustrar las funcionalidades técnicas del aplicativo.
* Especificar las herramientas utilizadas para el desarrollo y diseño de la aplicación.

1. **INTRODUCCIÓN**

Este manual se realiza con el fin de detallar el sistema en términos técnicos con los pasos necesarios para la instalación, desarrollo y ejecución del aplicativo.

Es importante considerar los requerimientos mínimos de software y hardware para la correcta instalación del sistema.

El aplicativo SIREP tiene la finalidad de mejorar los procesos administrativos de inventario, venta y entrega de productos de las unidades productivas de Sena-Empresa.

Se recomienda que este manual sea manipulado únicamente por la persona que se encargara de administrar, editar o configurar nuestro aplicativo con el fin de mantener la seguridad de los datos que se almacenan en la base de datos.

1. **REQUISITOS TÉCNICOS MINIMOS DE HARDWARE**

* **Equipo de cómputo con las siguientes características:**

RAM: 6GB

ALMACENAMIENTO: mínimo 150Gb.

PROCESADOR: Intel Core i3 de quinta generación o superior.

* **Información técnica del servidor implementado.**

Procesador - Intel Xeon (R) W- 2145 CPU @

Frecuencia - 3.70 GHz

RAM instalada 32.0 GB (31,7 GB Usable)

Tipo de sistema – sistema operativo de 64 bits, procesador basado en x64

Sistema operativo – Linux Ubuntu server

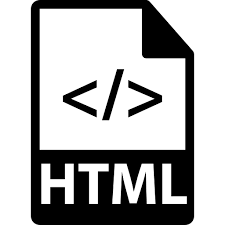
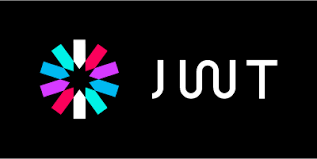
Versión – 22.04

Disco duro – mecánico 1TB

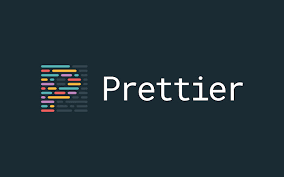
1. **REQUISITOS TÉCNICOS MINIMOS DE SOFTWARE**

* Privilegios de administrador.
* Navegadores internet: Google Chrome.
* **Node.js:** es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.



* **HTML:** Es el lenguaje de marcado que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o insertando imágenes y videos en la página.
* **CSS:** Es un lenguaje de reglas de estilo que usamos para aplicar estilo a nuestro contenido HTML, por ejemplo, establecer colores de fondo y tipos de letra, y distribuir nuestro contenido en múltiples columnas.
* **JAVASCRIPT:** Es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás. (Está bien, no todo, pero es sorprendente lo que puedes lograr con unas pocas líneas de código JavaScript).
* **Bootstrap:** Es un kit de herramientas de código abierto para desarrollos web responsive con HTML, CSS y JavaScript. Con él puedes darle forma a tu sitio web a través del uso de sus librerías CSS y JavaScript. Incluye diferentes componentes: ventanas modales, menús, cuadros, botones, formularios… Es decir, los elementos que necesitas para maquetar tu página.
* **Sweealert2:** Es una librería para mostrar notificaciones desde JavaScript al cliente. Es un remplazo bonito, personalizable y accesible para los cuadros emergentes de JavaScript. Sin dependencias.
* **JQuery**: es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.Este fue utilizado para generar tablas dinámicas con paginado, búsqueda, ordenar por columnas, etc.
* **Json Web Token:** JWT (JSON Web Token) es un estándar qué está dentro del documento RFC 7519.En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de claims o privilegios.Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro de del payload o cuerpo de un mensaje que va firmado digitalmente.
* **Github**: GitHub es un sitio "social coding". Te permite subir repositorios de código para almacenarlo en el sistema de control de versiones Git.
* **ejs:** Es un lenguaje de plantillas simple que le permite generar marcado HTML con JavaScript simple. Ninguna religiosidad sobre cómo organizar las cosas. Sin reinvención de la iteración y el flujo de control. Es simplemente JavaScript.
* **express:** Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Con miles de métodos de programa de utilidad HTTP y middleware a su disposición, la creación de una API sólida es rápida y sencilla.
* **express session:** El middleware express-session almacena los datos de sesión en el servidor; sólo guarda el ID de sesión en la propia cookie, no los datos de sesión. De forma predeterminada, utiliza el almacenamiento en memoria y no está diseñado para un entorno de producción.
* **dotenv:** nos permitirá leer estas variables desde un archivo llamado .env, para posteriormente cargar las variables de entorno en la variable process.env.
* **mysql:** Este es un controlador que permite manipular la base de datos de Mysql.
* **npm init:** Esto activará la inicialización de tu proyecto. Este comando funciona como una herramienta para crear el archivo package. json de un proyecto.

**Plugin utilizados para mejoras en el editor de código.**

* **Tabnine:** TabNine es un plugin de autocompletado, orientado a la programación, que tiene un tiempo en el mercado. Es muy útil para agilizar el trabajo de los desarrolladores. Sin embargo, han decidido subir la apuesta y lanzar Deep TabNine: un autocompletado potenciado por Deep Learning.
* **Prettier:** Lo que ofrece Prettier es tomar tu código y "re-formatearlo" en base a las configuraciones definidas manteniendo así un estilo consistente. Haciéndolo mucho más legible y sin intervención nuestra ni preocupación de como lo escribes inicialmente.
* **Snippets html:** Los snippets en Visual Studio Code1 son fragmentos de código reutilizables que puedes llamar con un atajo de teclado o con prefijos mientras escribes en el editor. VSCode te sugiere ayuda con IntelliSense al estar escribiendo.

**Herramienta para desplegar aplicaciones node.**

* **Pm2:** es un gestor de procesos de producción para las aplicaciones Node. js que tiene un equilibrador de carga incorporado. PM2 permite mantener siempre activas las aplicaciones y volver a cargarlas sin ningún tiempo de inactividad, a la vez que facilita tareas comunes de administrador del sistema.

1. **REQUERIMIENTOS**

Las unidades productivas constituyen la base de la estrategia y son el componente esencial de la estrategia SENA Empresa, es la unidad mínima de dinamización de la misma al replicar en la mayoría de sus componentes la estructura de una empresa real.

Unidades Productivas:

* Agrícola
* Agroindustria
* Gastronomía
* Pecuaria
* Escuela Nacional de la Calidad del Café
* Ambiental – Recursos Naturales
* Empresa de Servicios Públicos
* Moda – Comercio y Servicios

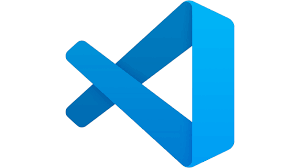
Dado a lo anterior, la estrategia Sena Empresa, solicita al tecnólogo en Análisis y Desarrollo de Sistema de Información, modelar un software que le permita registrar la producción y ventas de los productos generados por las unidades productivas.

A partir de la solicitud de Sena empresa, se generaron los requerimientos funcionales especificados en la siguiente tabla:

* **RF1** El sistema debe permitir a cada punto de venta, administrar datos básicos de los diferentes tipos de usuarios (aprendices, instructor, administrativo, invitados), tales como (identificación, nombre, teléfono, dirección).
* **RF2** El sistema debe permitir al usuario administrador de Sena Empresa, administrar los datos básicos de la empresa tales como (razón social, logo, misión, visión).
* **RF3** El sistema debe permitir al usuario administrador de Sena Empresa, administrar los datos de las unidades productivas tales como (nombre de la unidad, código, misión, visión, logo).
* **RF4** El sistema debe permitir al usuario administrador de Sena Empresa, administrar los datos de productos de las unidades Productivas, tales como (código, nombre, descripción, imagen, unidad de medida, tipo de empaque, valor del producto, estado de producto).
* **RF5** Cada unidad productiva tiene asignado un líder administrador quien es el encargo de cargar en el sistema los datos la producción tales como (fecha, cantidad, observación, estado de la producción, productos generados).
* **RF6** El sistema debe permitir al usuario administrador de Sena Empresa, administrar los datos básicos de los puntos de ventas y la asignación de productos generados por las unidades productivas.
* **RF8** El sistema debe permitir realizar reservas de producto a los usuarios de tipo (aprendices, instructor, administrativo, invitados).
* **RF9** El sistema debe permitir a cada punto de venta registrar las ventas y entrega de productos (fecha de la venta, cantidad, valor, cliente, producto, observación, estado de la venta).
* **RF10** El sistema debe permitir al usuario administrador de Sena Empresa asignar el administrador de cada unidad productiva.
* **RF11** El sistema debe permitir a cada punto de venta generar la factura de cada venta realizada.
* **RF12** El sistema no debe permitir realizar reserva si no existen productos disponibles.
* **RF13** El sistema debe validar a los usuarios para poder realizar cualquier actividad en el sistema

1. **HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO**

**StarUml:** Es una excelente aplicación de código abierto que permite trabajar con todos los diagramas relacionados. Además, puede realizar generación de códigos para Java, C++ y C#, así como documentos para todos los programas que componen el paquete de Microsoft Office, tales como: Word, Excel y PowerPoint.

**VISUAL STUDIO CODE:** Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

**XAMPP** Es una herramienta para “emular" un servidor en nuestro propio ordenador, de forma local, desarrollada por Apache Friends. Lo que hace XAMPP es instalar un servidor Apache con MySQL como gestor de base de datos, PHP como lenguaje de script del lado del servidor y también cuenta con el lenguaje de programación Perl (en este proyecto no es usado). Gracias a esto, es posible desarrollar el proyecto al completo de manera local, sin necesidad de disponer de servidores remotos. Dicho todo esto nos disponemos a instalar XAMPP en Windows. Descargamos el archivo de instalación de XAMPP en la ruta www.apachefriends.org donde podemos encontrar la versión más reciente y para todos los sistemas operativos, aunque en esta ocasión utilizaremos Windows.

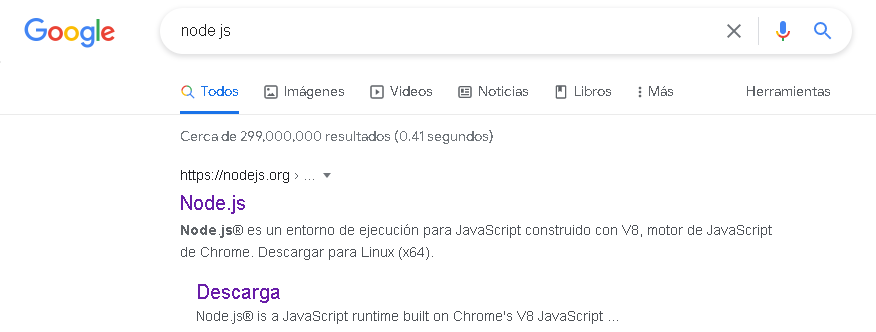
**MYSQL WORKBENCH:** MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.

**6.1 Instalación Del Servidor Web Local**

1. Se descarga e instala node js.
2. Se descarga e instala xampp para utilizar mysql como motor de base de datos y phpmyadmin como administrador.

**INSTALACION DE NODE JS**

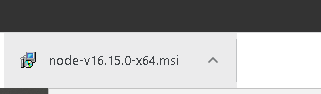
* Busca en un navegador cualquiera la palabra **NodeJS** e ingresa a su página oficial.

****

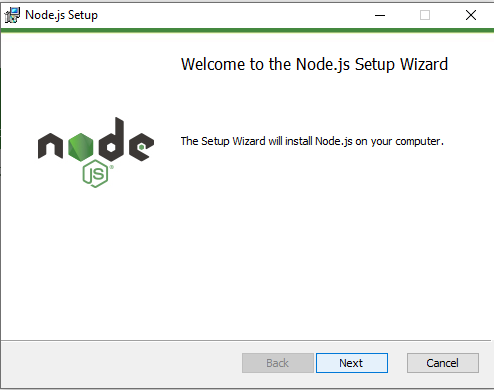
* Una vez adentro seleccionamos la descarga LTS la cual es la versión más estable de Node.



* Una vez se termine la descarga presionar para abrir el instalador.



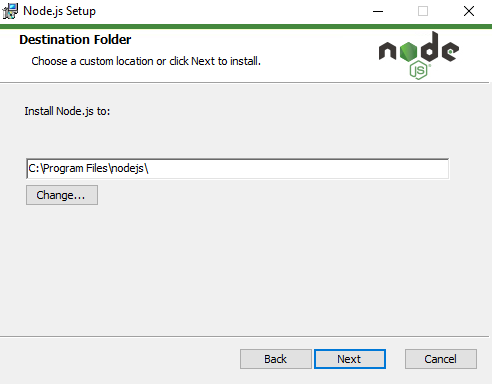
* Una vez se termine la descarga presionar para abrir el instalador y se abrirá el setup de instalación, presionamos en **Next** para continuar.



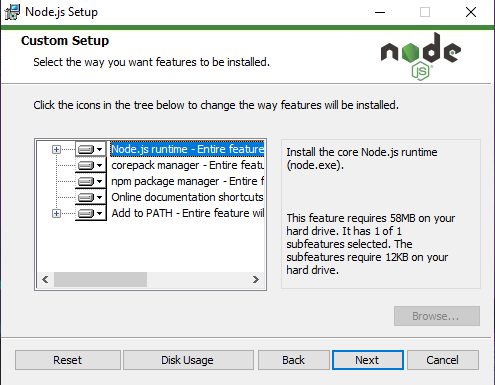
* Ahora hay que aceptar los términos y condiciones y luego dar clic en **Next.**

****

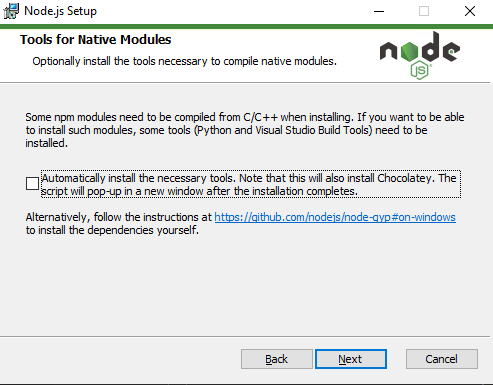
* Selecciona el destino de la carpeta de instalación. Y luego presionar en **Next**.

****

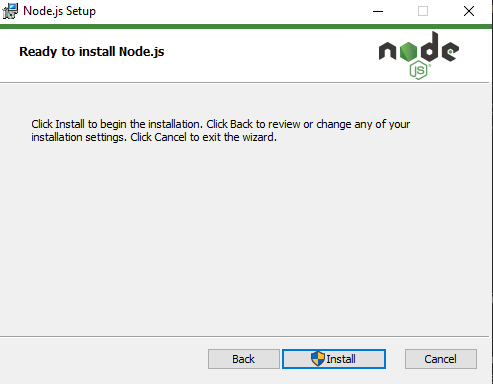
* Para continuar la instalación básica presionar en **Next.**

****

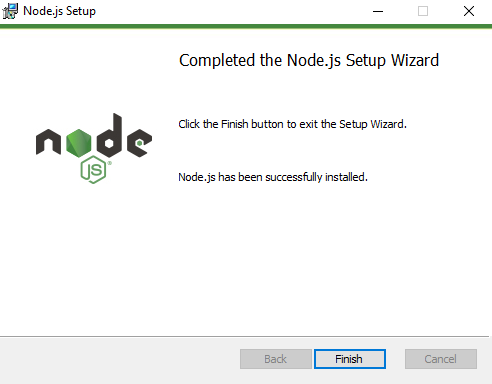
* Ahora pedirá la opción de agregar Chocolatey el cual se puede usar para descargar programas módulos y actualizaciones de estos por consola de Windows. Esta instalación es opcional.

****

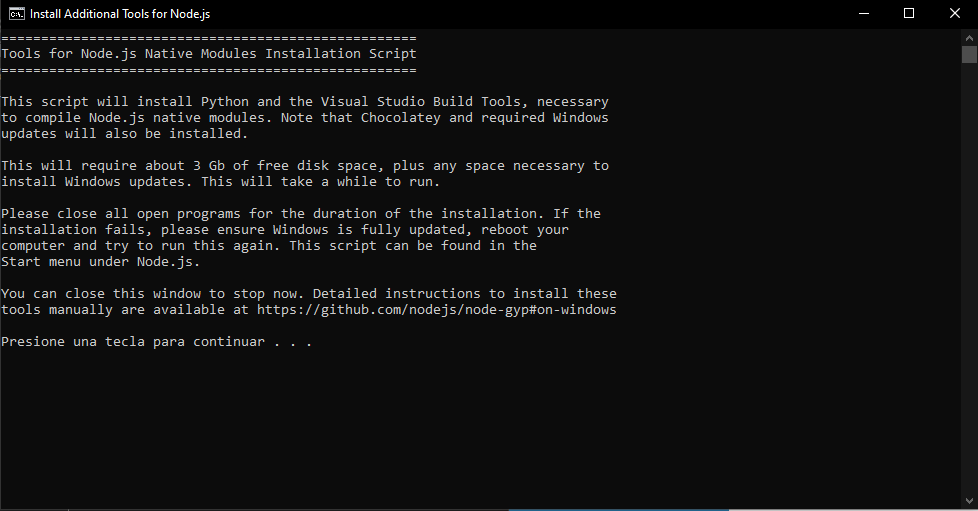
* Para finalizar la instalación presionamos **Instalar**.



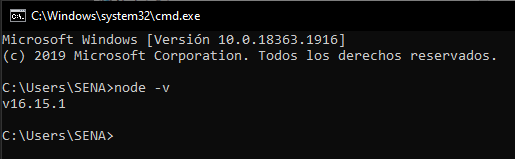
* Una vez haya finalizado la instalación aparecerá esta pantalla y presionamos en **Finish**.



* En esta ventana de consola presionaremos **Enter** hasta que se cierre.

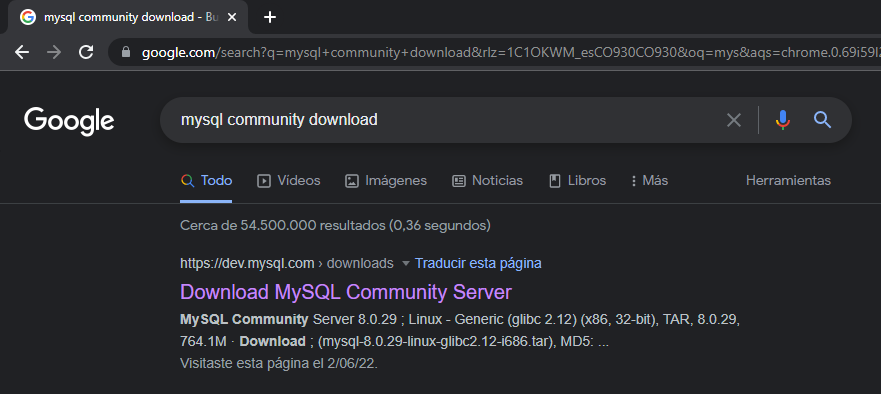


* Como ultimo paso verificamos desde la consola de que se haya instalado con el siguiente comando **node -v** y este será el resultado si se ha instalado correctamente.

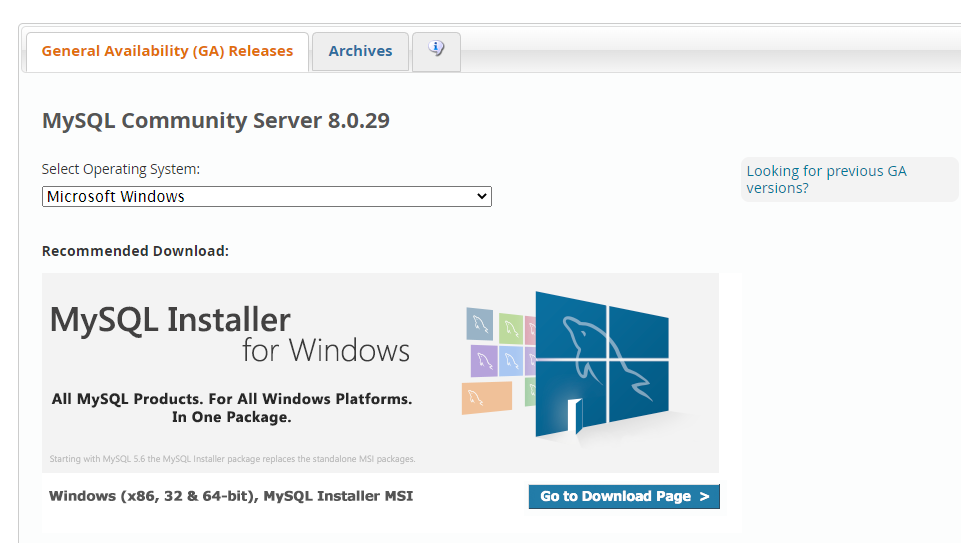


**INSTALACION DE MYSQL**

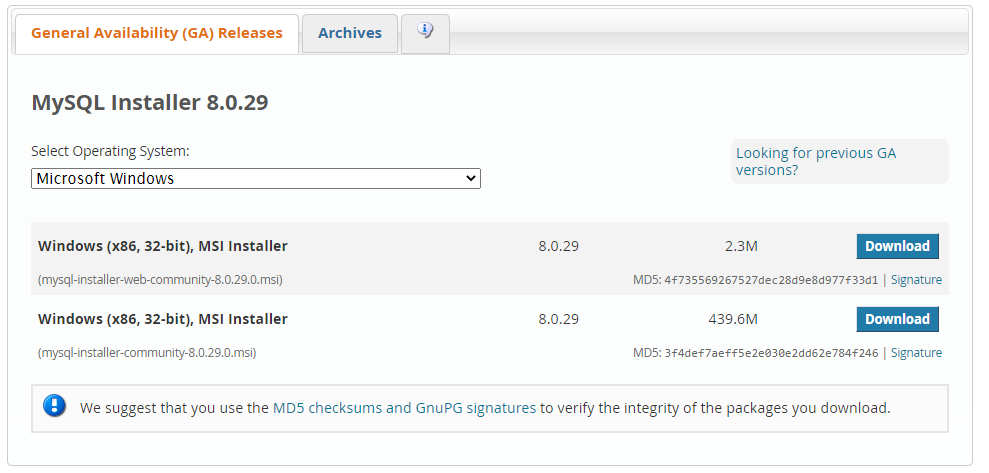
Busca en un navegador cualquiera la siguiente entrada.



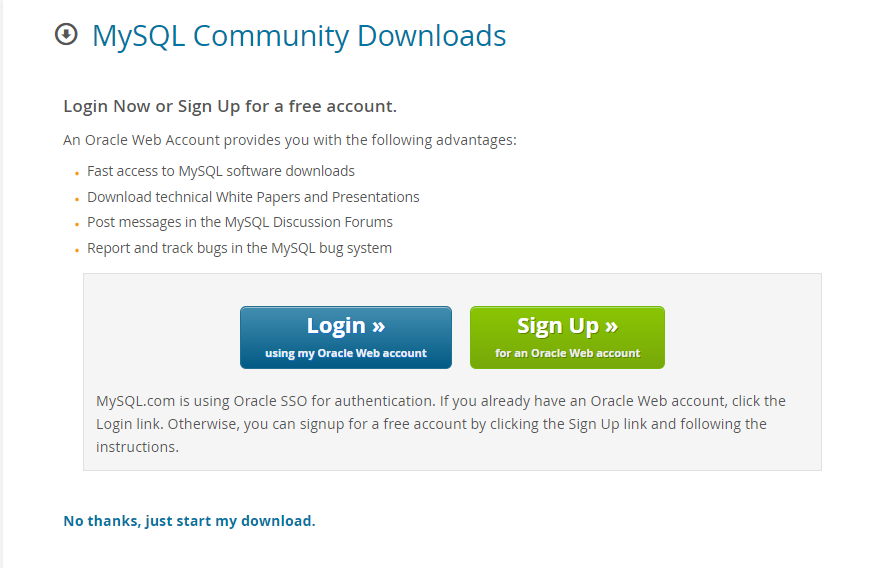
* Una esté en la pagina oficial de MySQL, dar clic en el siguiente campo.



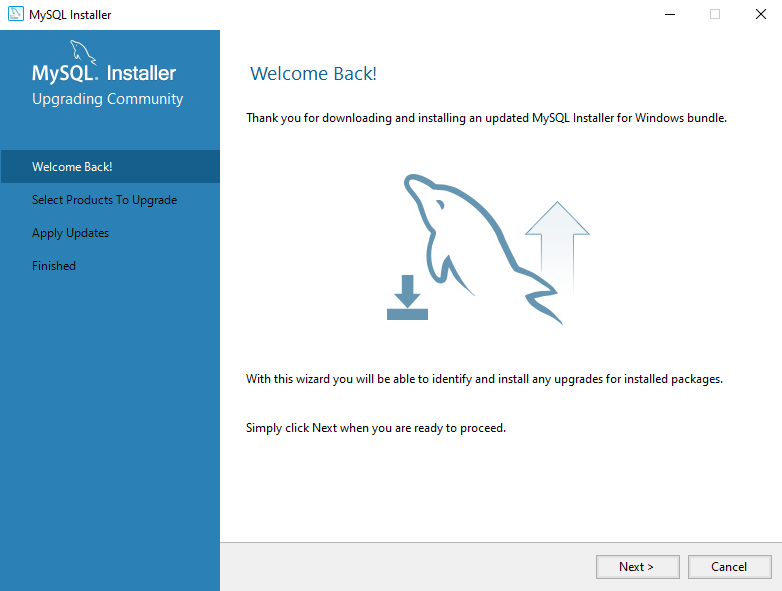
* En esta página hay dos opciones, la descarga básica (solo el instalador) o la descarga completa que incluye todos los complementos de MySQL Community que pesa un poco más. Se recomienda descargar la versión completa para ahorrar tiempo de instalación y porque ya incluye módulos necesarios.



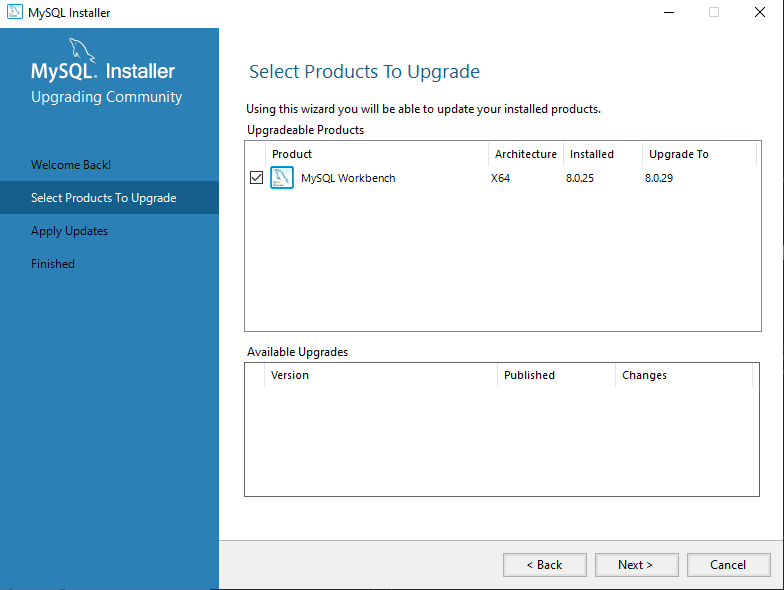
* Aquí la pagina va a pedir al usuario que inicie sesión es opcional, así que se podrá descargar sin iniciar sesión dando clic en el enlace. Y empezara la descarga.



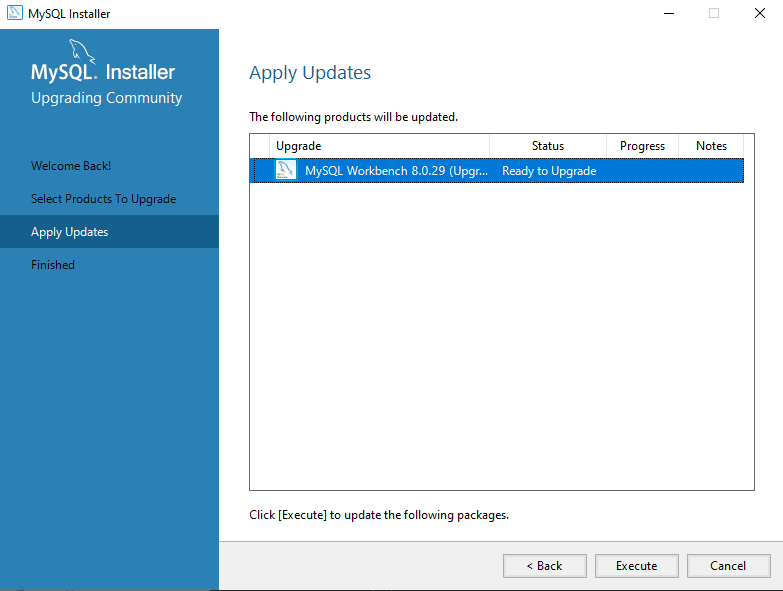
* Una vez se haya descargado el paquete instalador, lo ejecutamos y seguimos la secuencia presionando en **Next**.



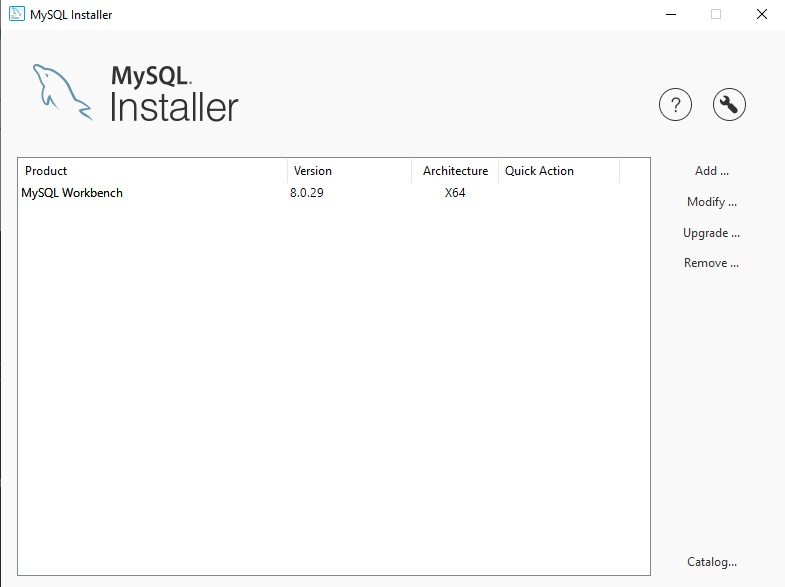
Si por alguna razón ya haya algún programa de MySQL instalado en el equipo aparecerá una ventana para actualizar la versión de ese programa es opcional actualizarlo. Presionar **Next**. Si no es así no aparecerá y seguirá.



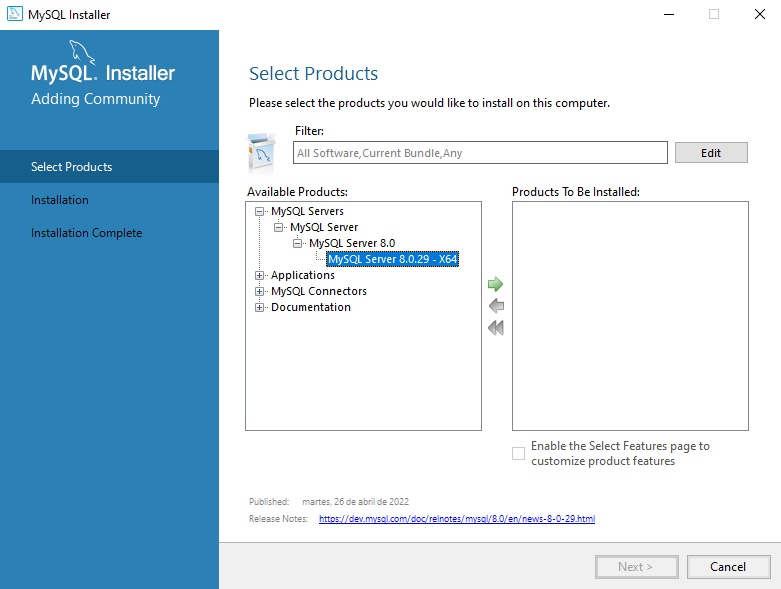
* Ahora aparecerá esta ventana en la cual hay que confirmar la actualización de lo mencionado en el paso anterior y presionar en **Execute.** Y esperar hasta que termine. Y presionar en **Finish** luego.

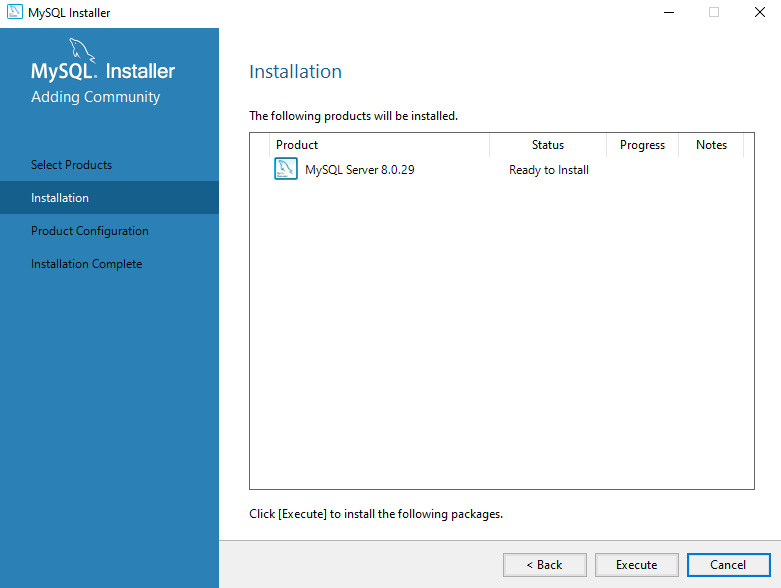


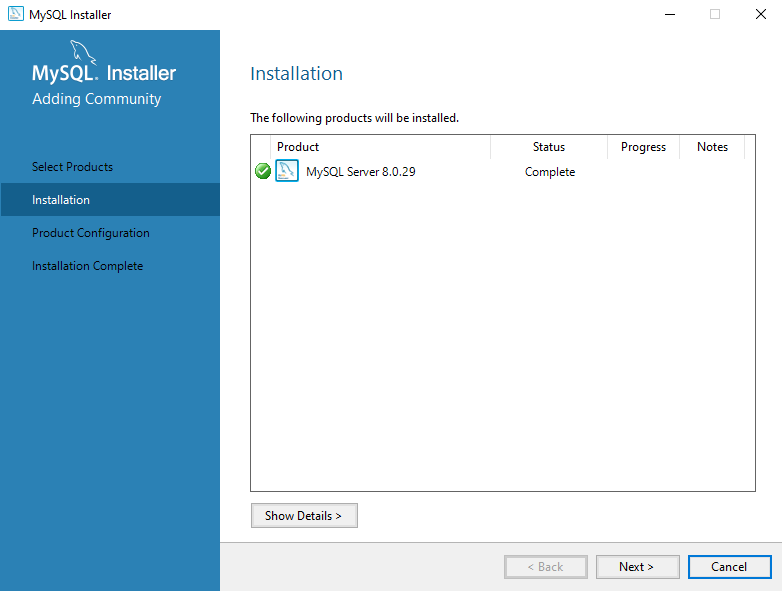
* Ahora la ventana principal del software se abrirá y da clic en **Add** para agregar complementos.



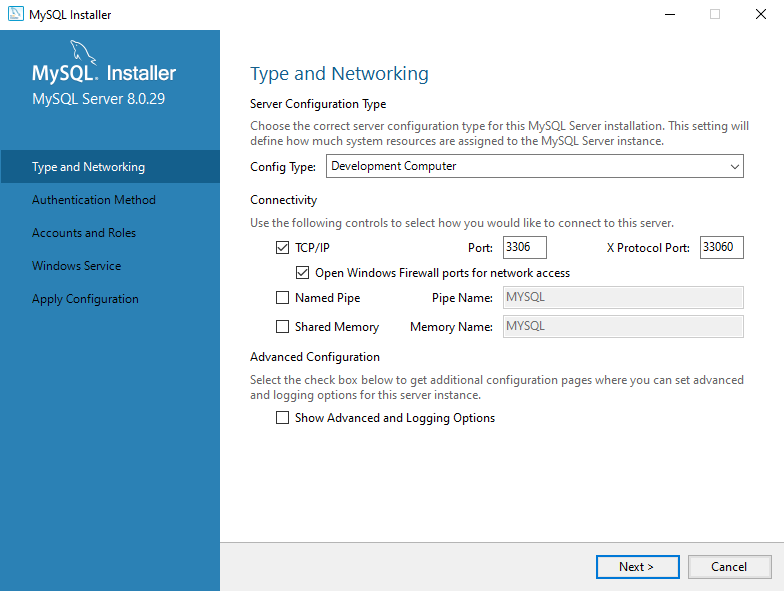
* Aparecerá la lista de complementos para instalar y busca el de **MySQL Servers** lo seleccionamos y lo movemos para instalar. Y se habilitara el botón **Next** y lo presiona.



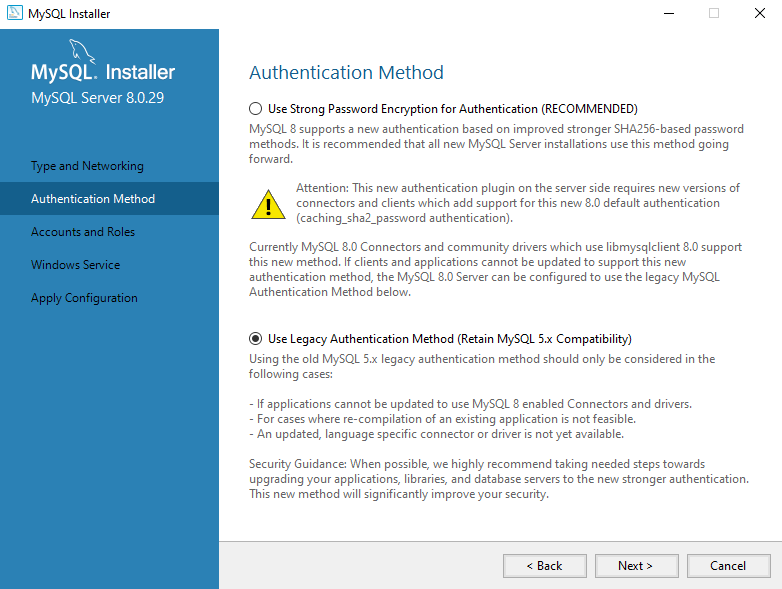
* Pedirá confirmación para instalar y presiona el botón **Execute**. Y esperar hasta que se instale. Y presionar **Next** para continuar con la configuración de MySQL.



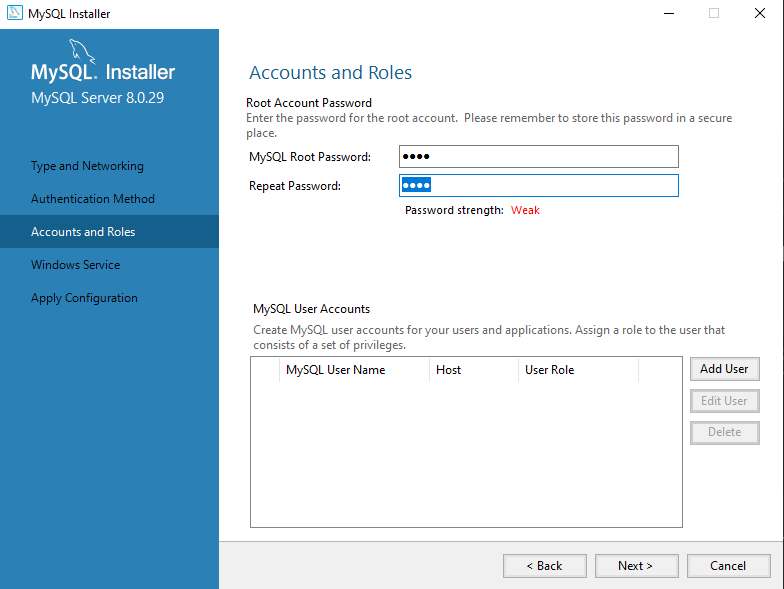
* Luego aparecerá la ventana de configuración de MySQL, dejar todo como esta y presionar en **Next**.



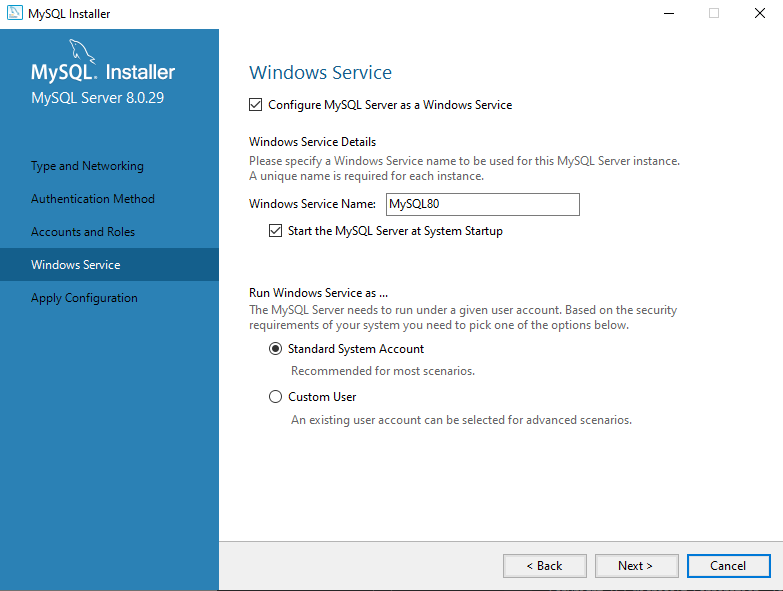
* Ahora hay que seleccionar el tipo de seguridad, por ahora seleccionar la segunda opción ya que es la más básica y fácil de utilizar. Y presionar **Next**.



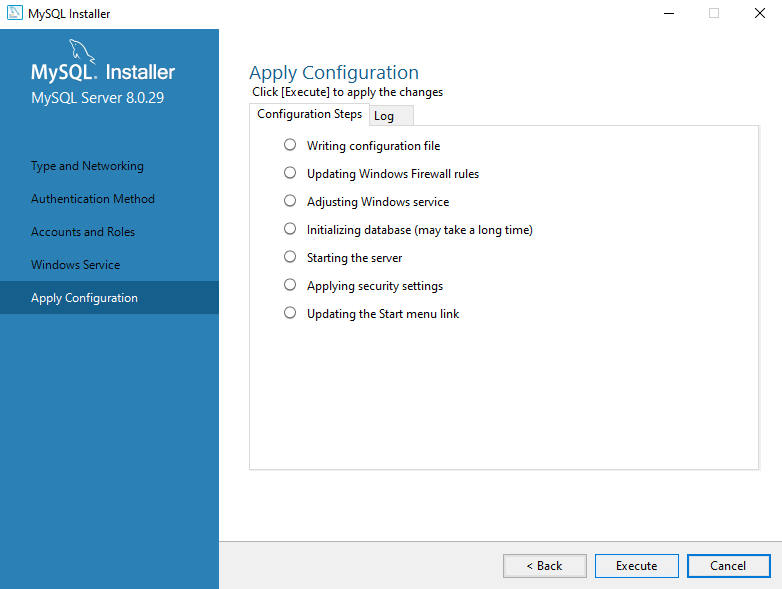
* Ahora hay que asignar la contraseña del usuario “root” se puede poner cualquiera. Y luego presionar **Next**.

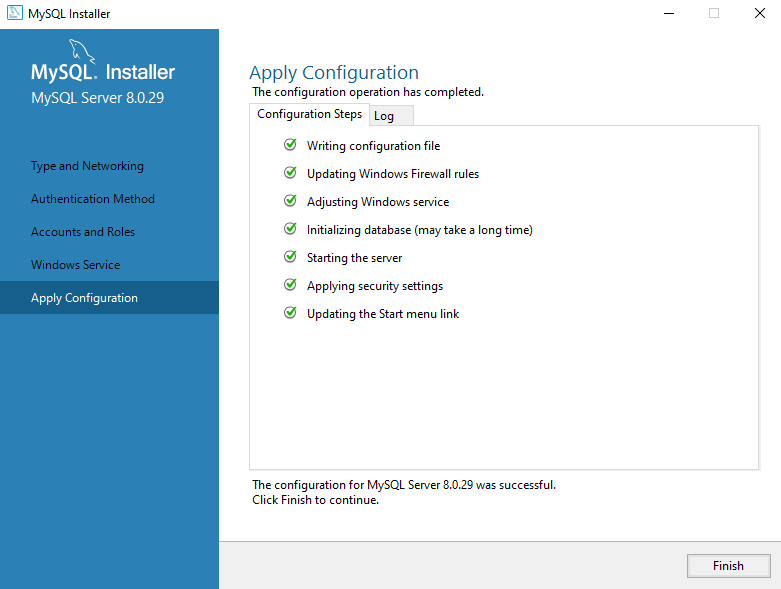


* Aparecerá esta ventana, dejar tal cual y presionar en **Next**.



* Ahora hay que confirmar que se instale y presionar el botón **Execute**. Espera hasta que se termine de instalar y configurar. Y presionar en **Finish**. Y finalizar todo.

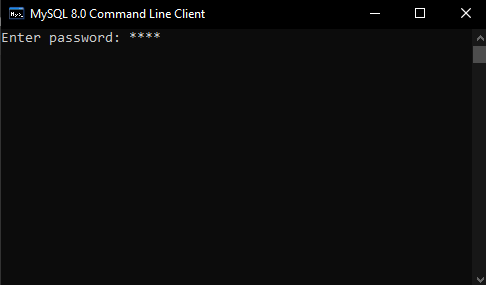




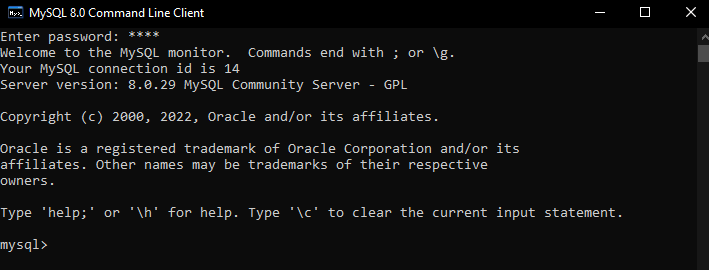
* Y para comprobar de que se haya instalado correctamente vamos al inicio y buscamos esta aplicación.



* Una vez abierta debe digitar la contraseña del usuario root.

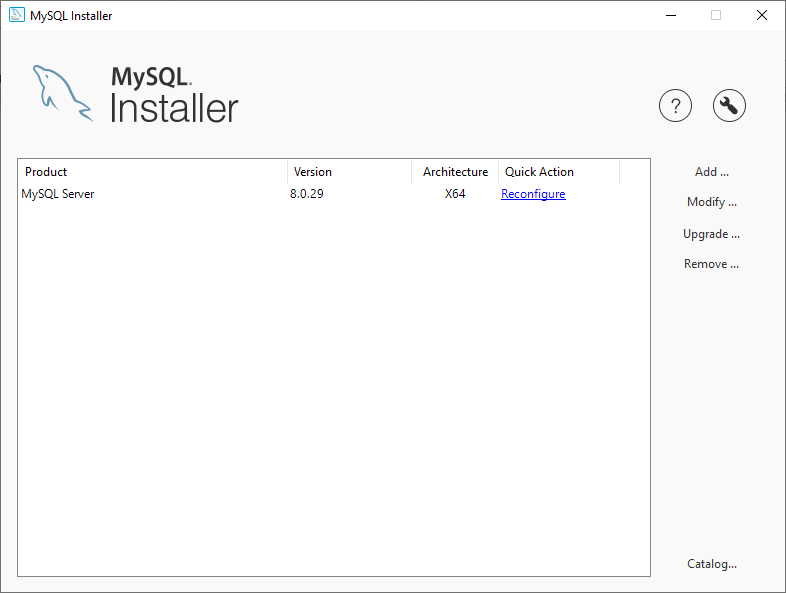


* Si todo ha salido bien debe aparecer los siguiente. Y se habrá finalizado la instalación.

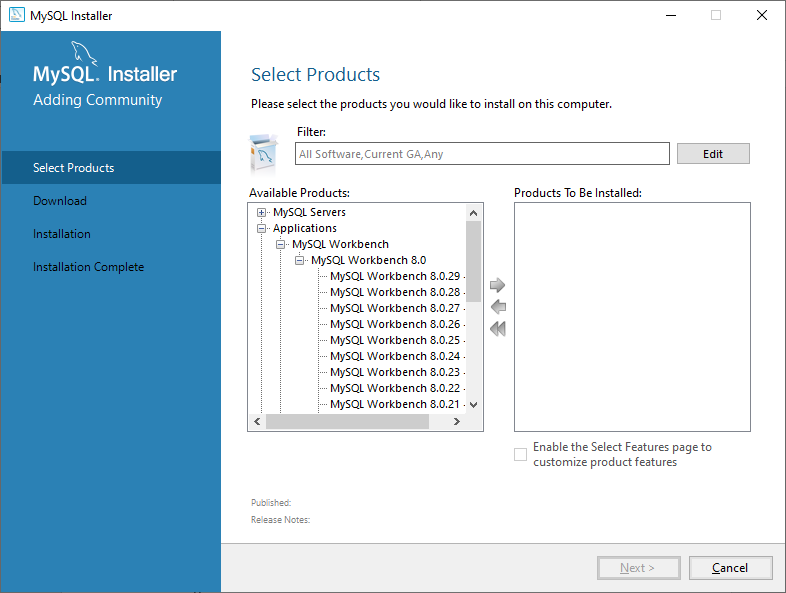


**INSTALACIÓN DE MYSQL WORKBENCH**

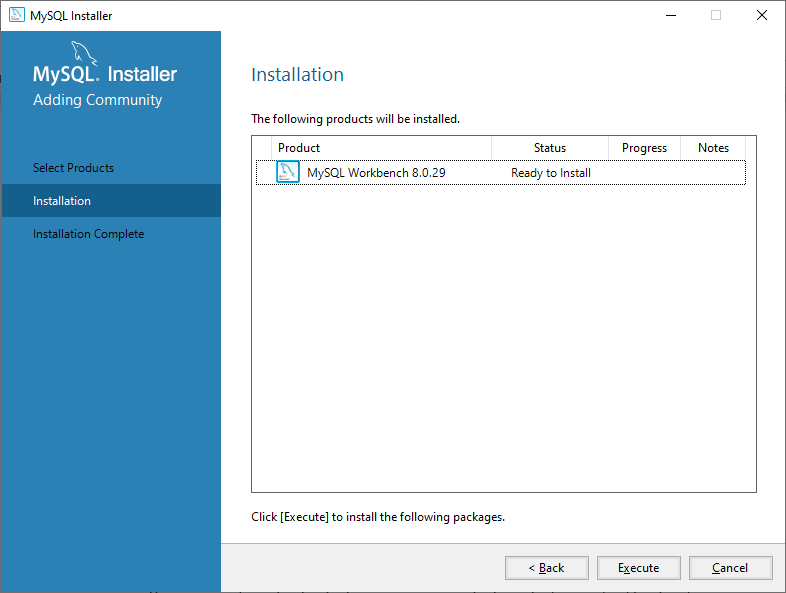
* Para instalar MySQL Workbench debe abrir la aplicación MySQL Installer Comunnity que ya debe estar instalada en el equipo y da clic en **Add**.



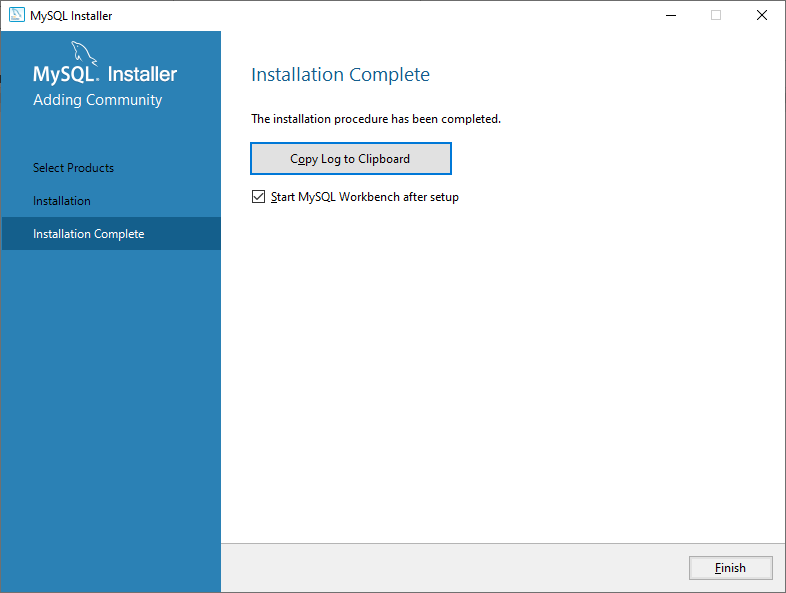
* Aparecerá una ventana con la lista de complementos y selecciona la última versión de MySQL Workbench y lo mueve para instalar. Presiona el botón **Next**.



* Ahora autoriza la instalación presionando el botón **Execute**. Esperar hasta que termine el proceso y luego presionar **Next**.



* Hecho esto ya se habrá finalizado la instalación de MySQL Workbench. Presionamos **Finish** y se habrá terminado.

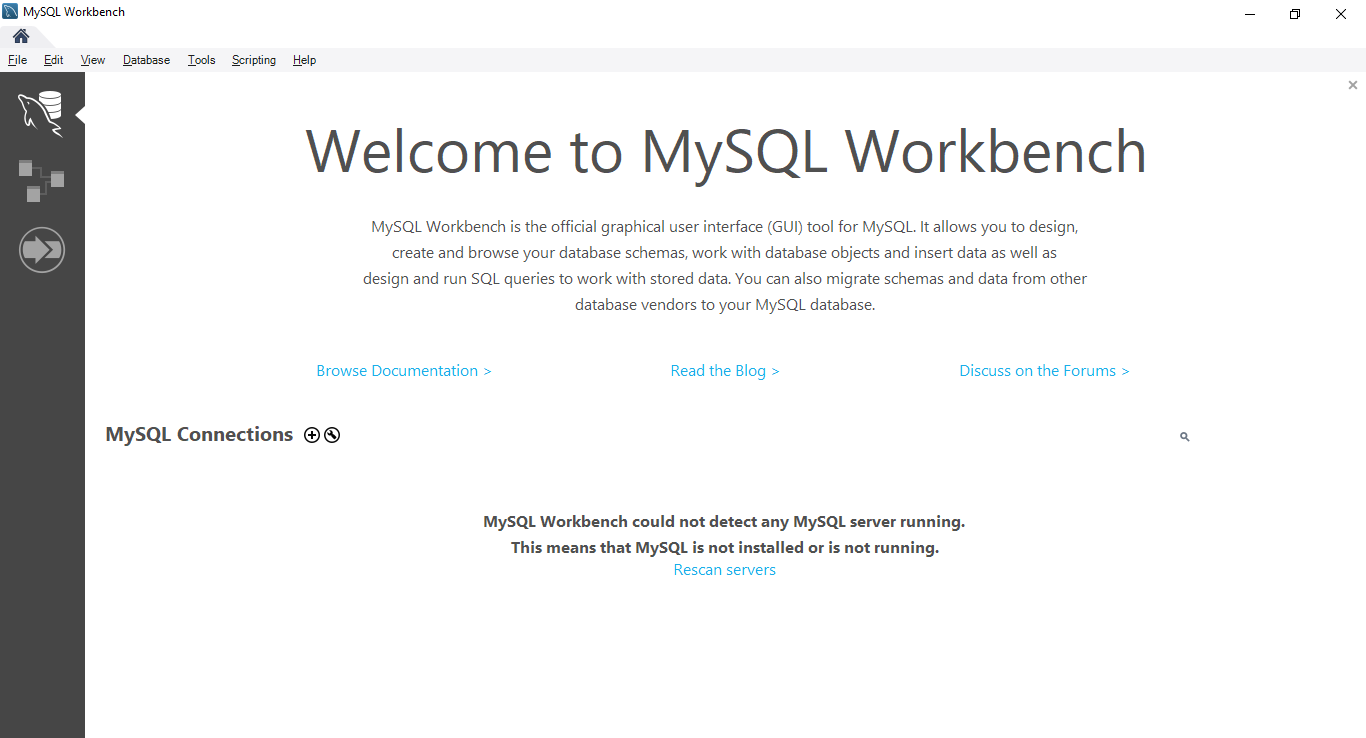


**CONFIGURACIÓN DE MYSQL WORKBENCH**

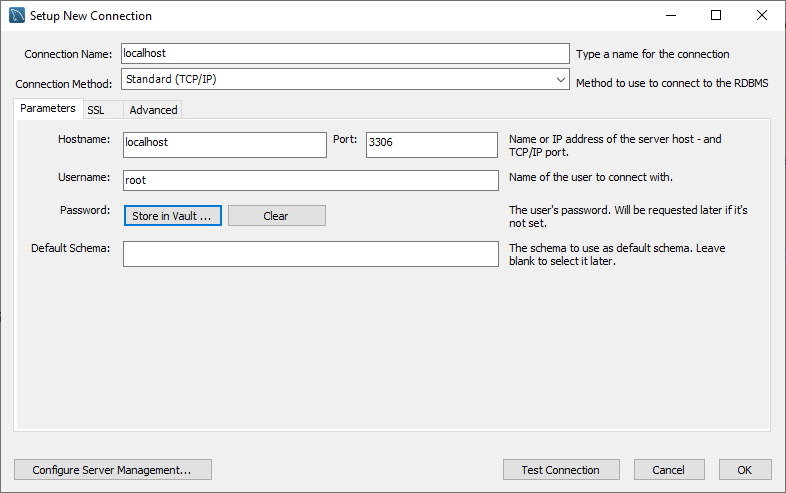
* Buscar la Aplicación MySQL Workbench en el equipo y ejecutarla.

****

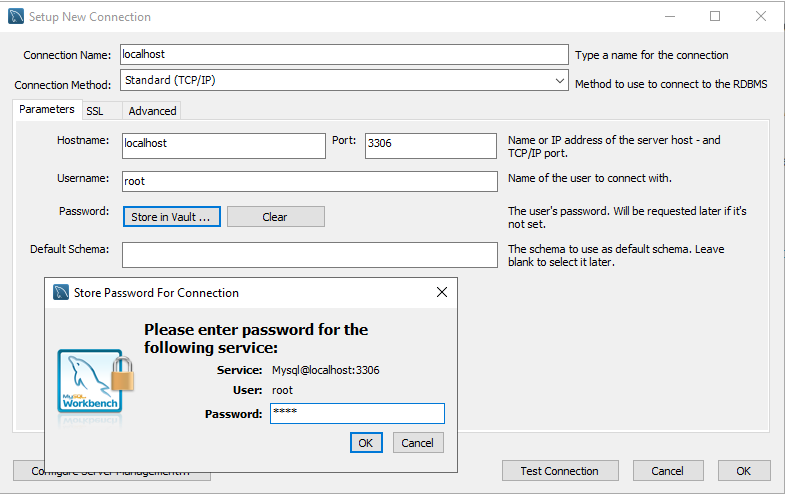
* Una vez iniciado la Aplicación vamos a crear una conexión de MySQL. Presionando el botón de **Agregar**.



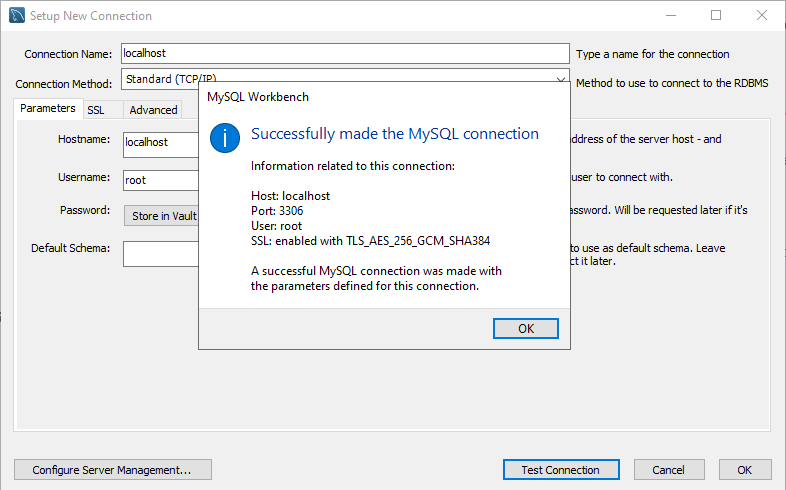
* Luego nos aparecerá esta ventana donde pondremos los datos tal cual como en la siguiente imagen.



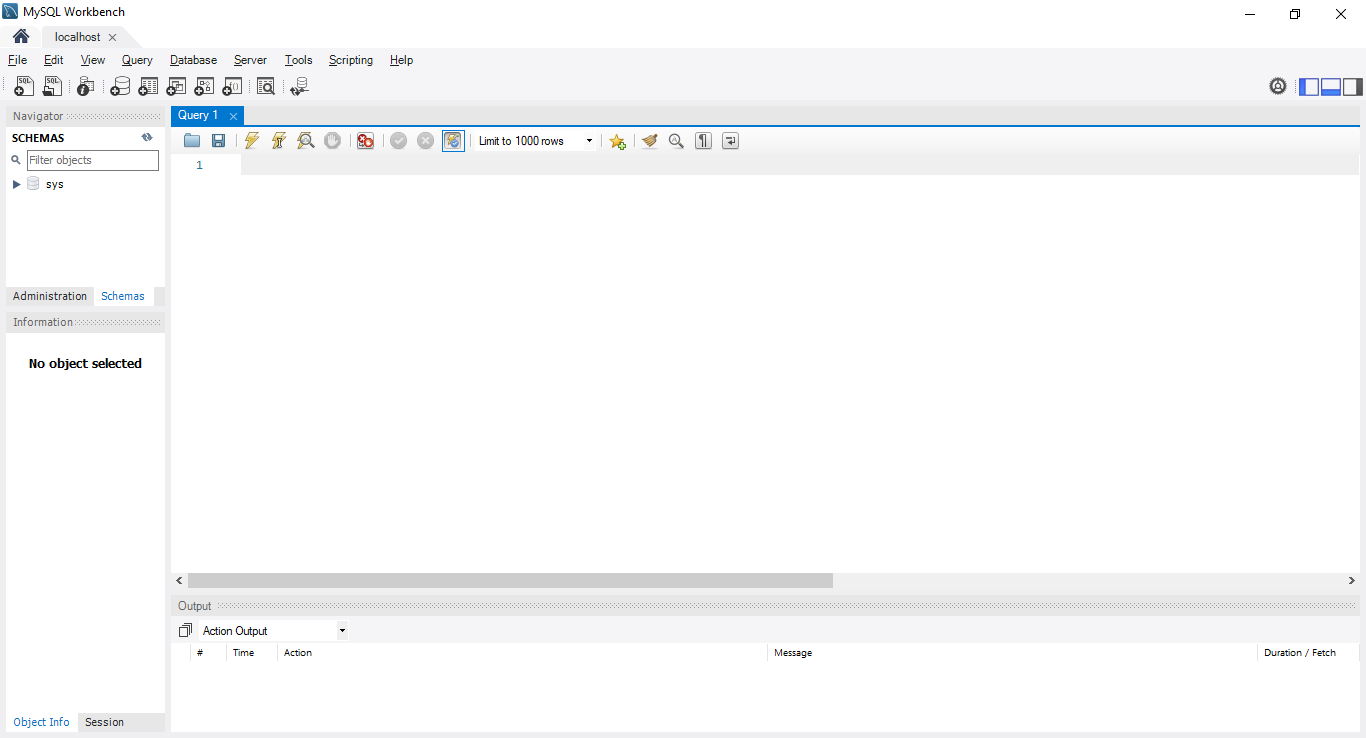
* Después presionar el botón **Store in Vault** para escribir la contraseña de MySQL. La ingresa y presionan en **OK**. Y para comprobar presionar en **Test Connection**.



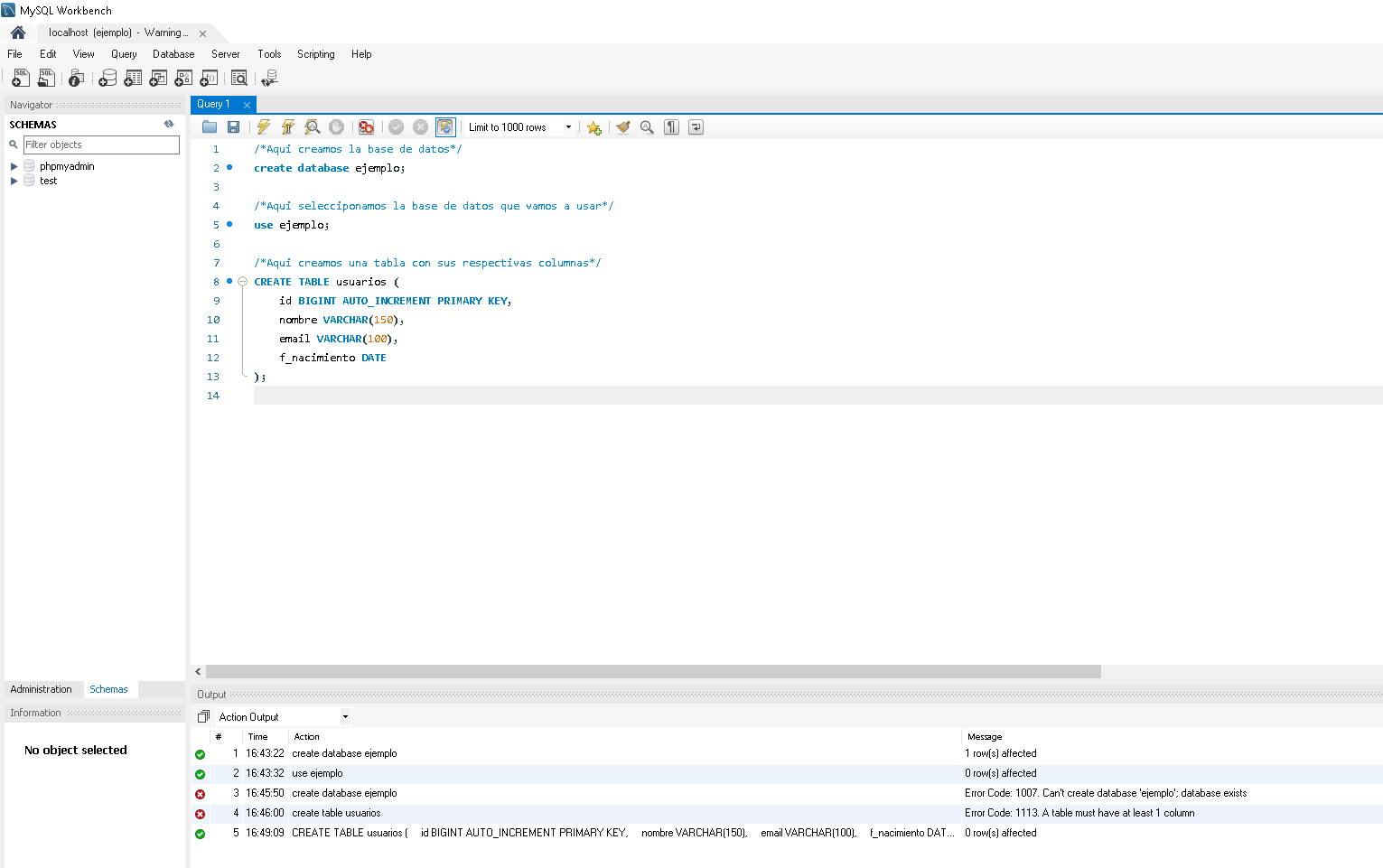
* Si todo ha salido bien aparecerá este aviso que indica que se ha conectado a MySQL. Presionamos **Ok**.

****

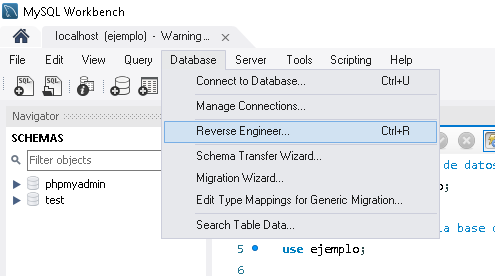
* Si todo ha salido bien aparecerá la vista principal de WorkBench y ya permitirá administrar bases de datos.



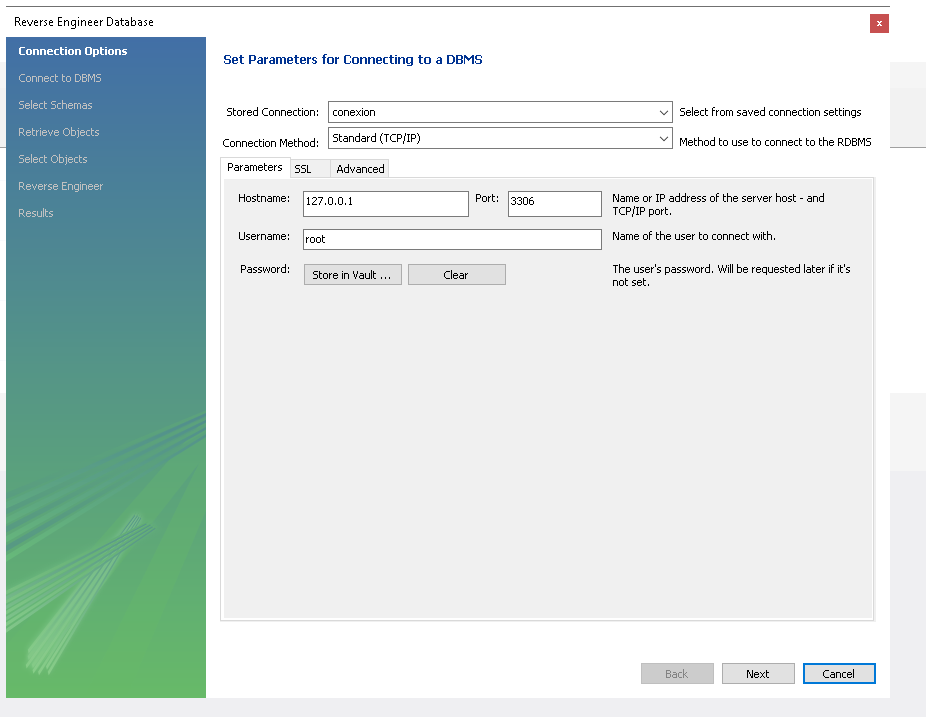
* Continuamos explicando cómo realizar una base de datos



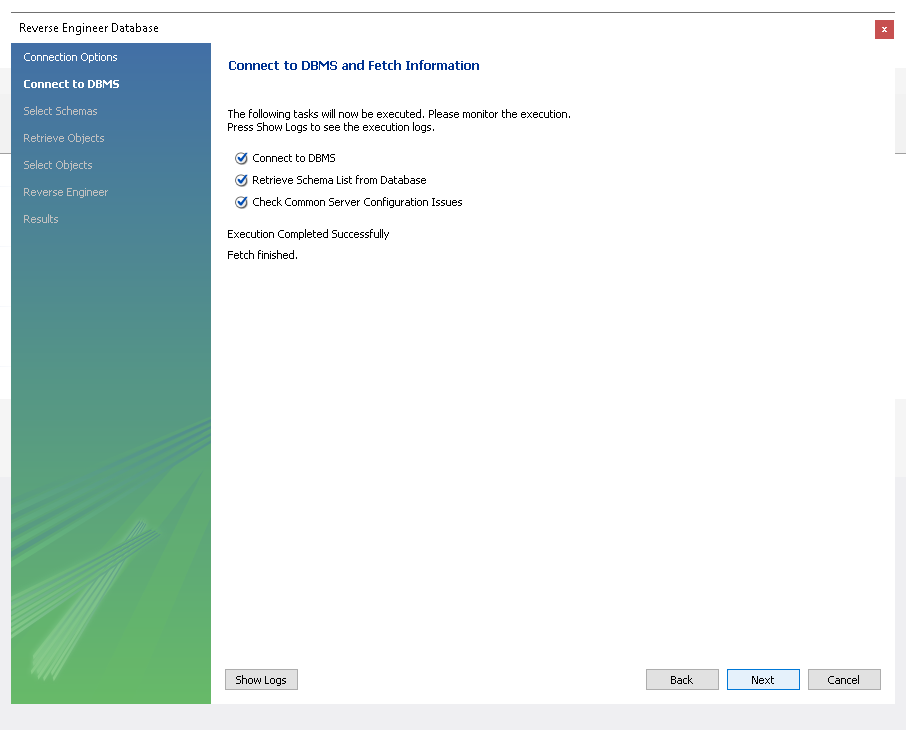
* Para visualizar la base de datos finalizada, vamos a la pestaña Database e ingeniería inversa



* Next



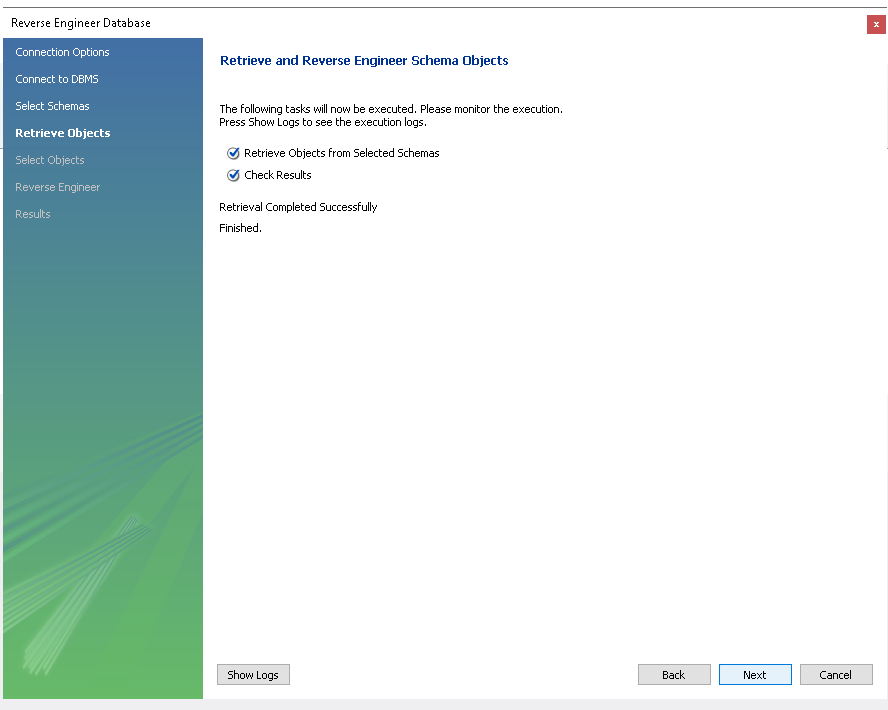
* Next



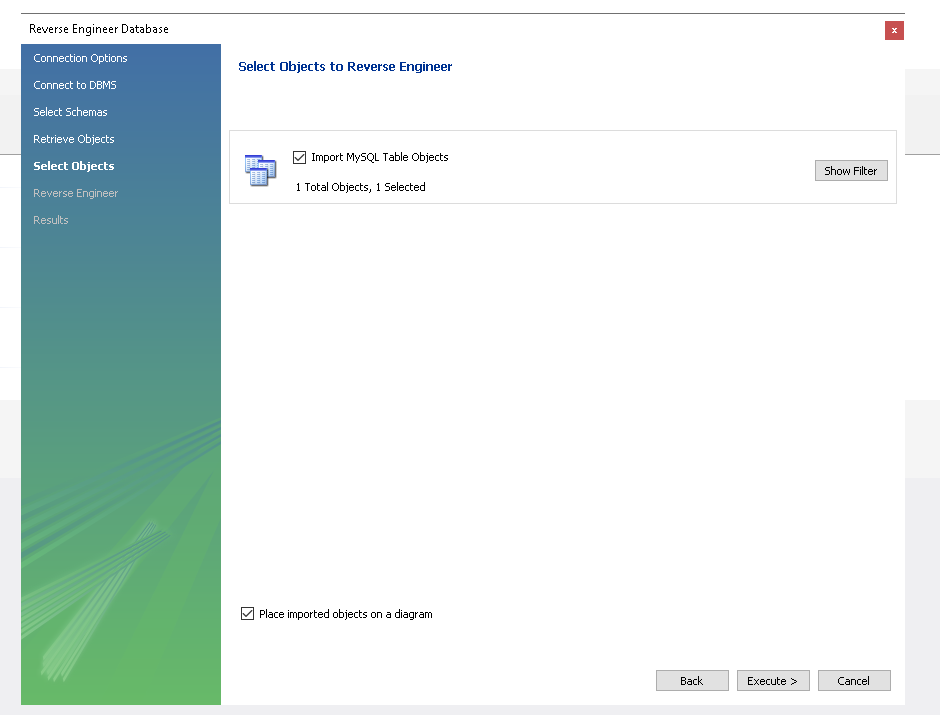
* Seleccionamos la base de datos que acabamos de crear => Next



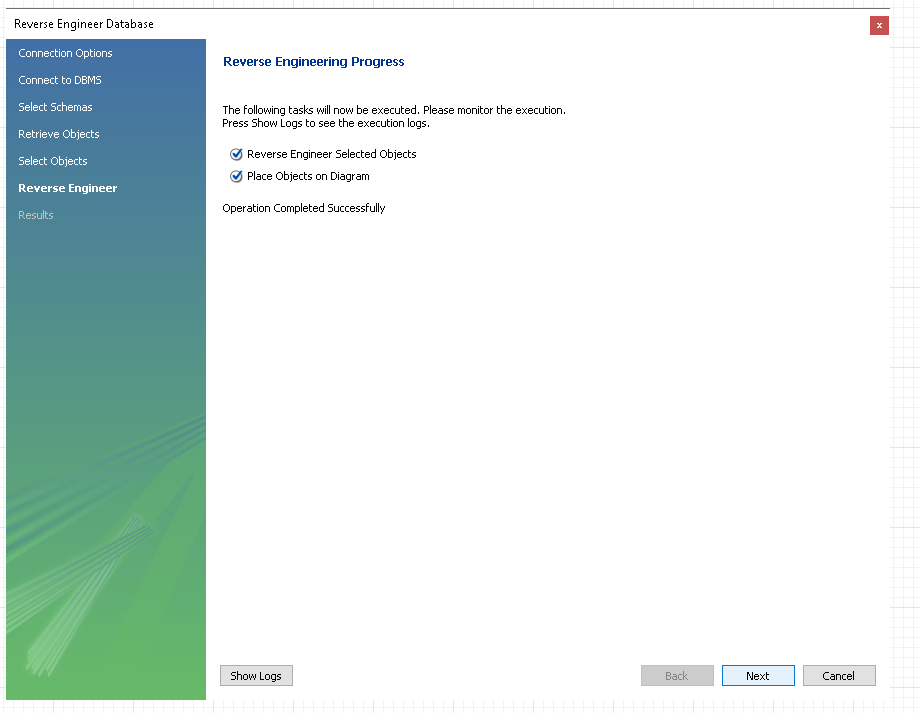
* Next



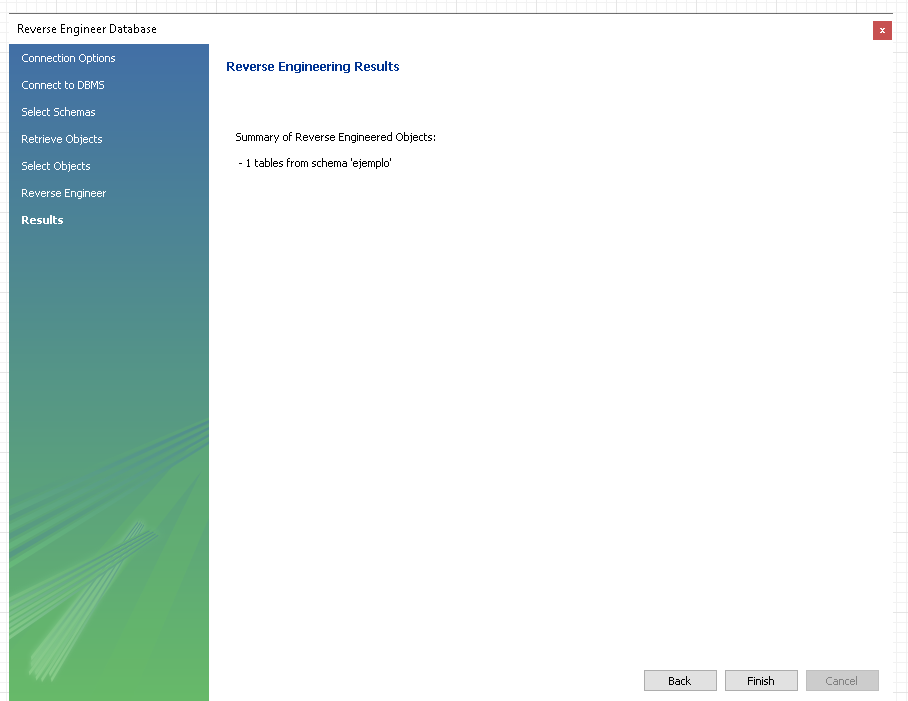
* Ejecutamos



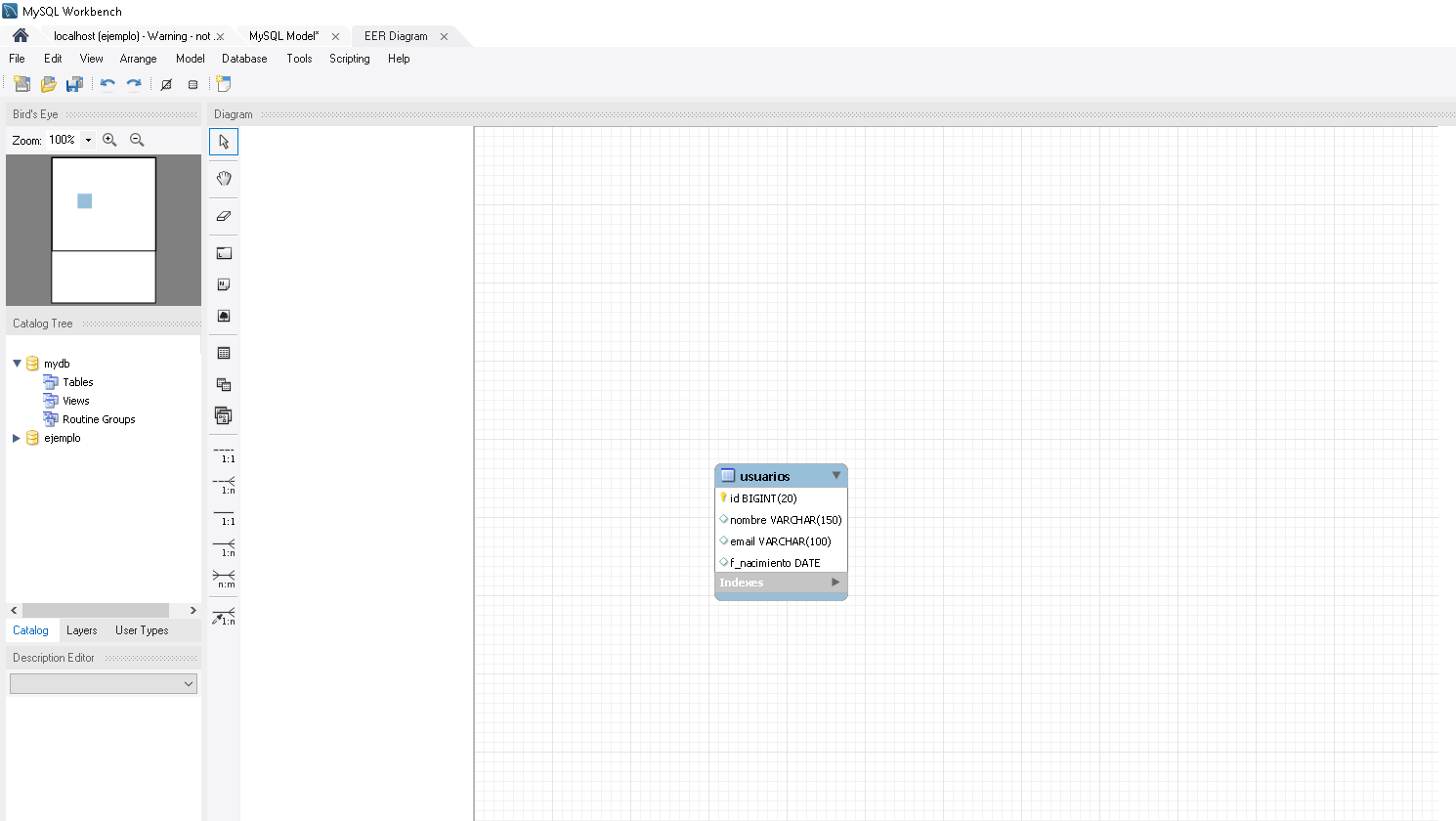
* Next



* Finish

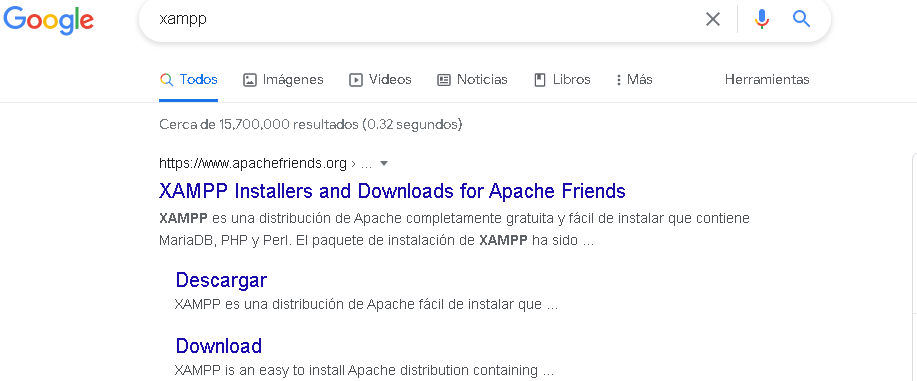


* Podemos observar la tabla creada, ejemplo

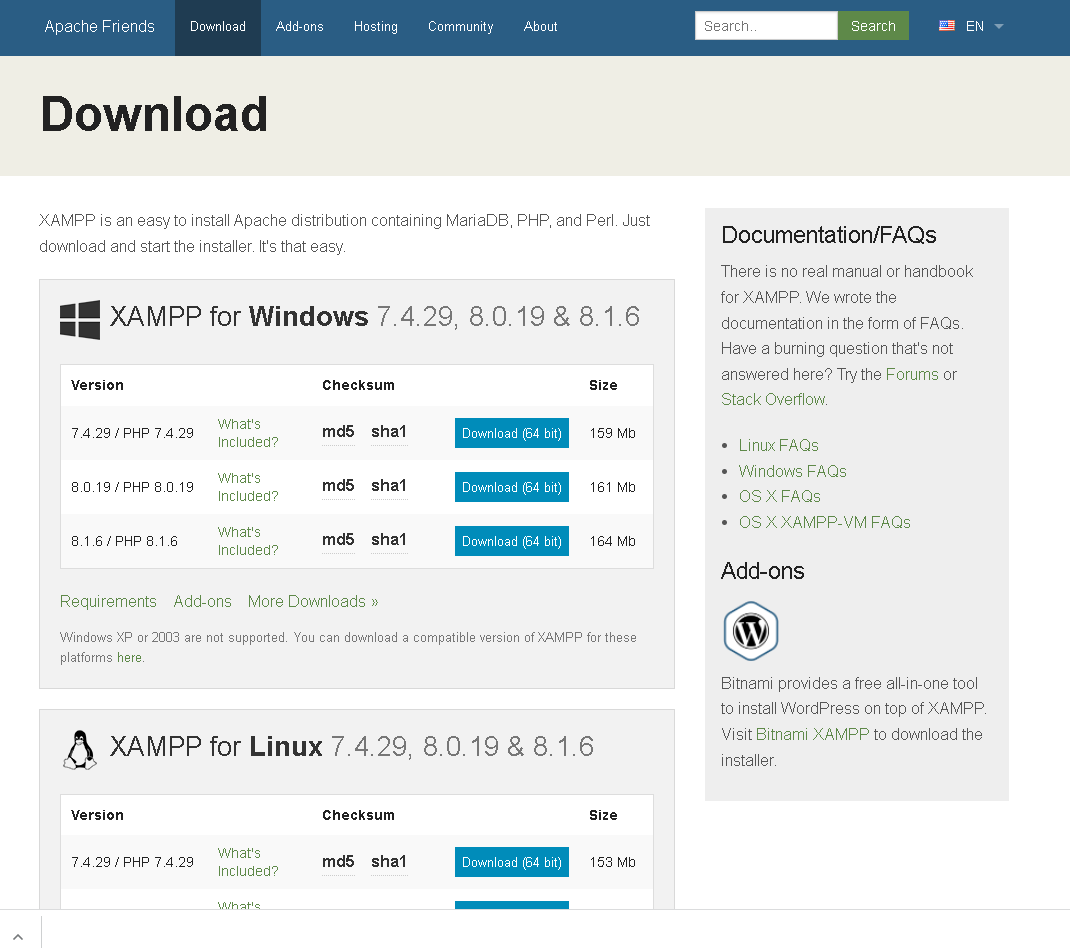


**INSTALACIÓN DE XAMPP (Alternativa a WorkBench)**

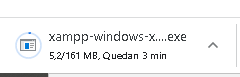
* En nuestro navegador de preferencia, buscamos Xampp => Download



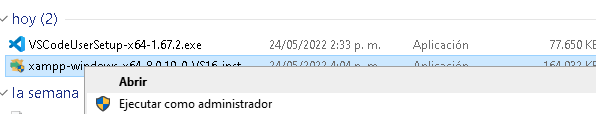
* Descargamos la versión con la que nuestro equipo es compatible, en este caso es de(64 hit) para windows



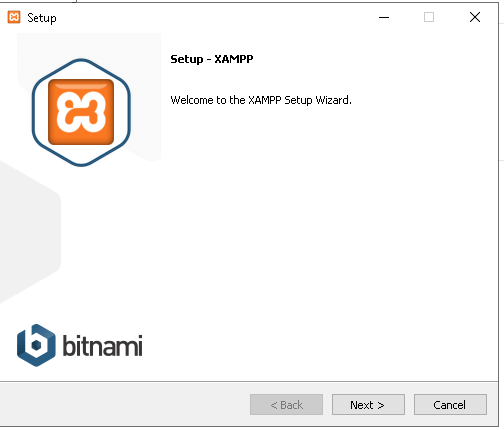
* Esperamos la descarga del archivo



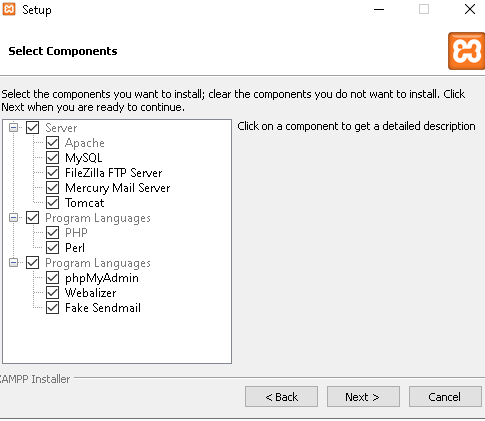
* Hacemos click y seleccionamos ejecutar como administrador.



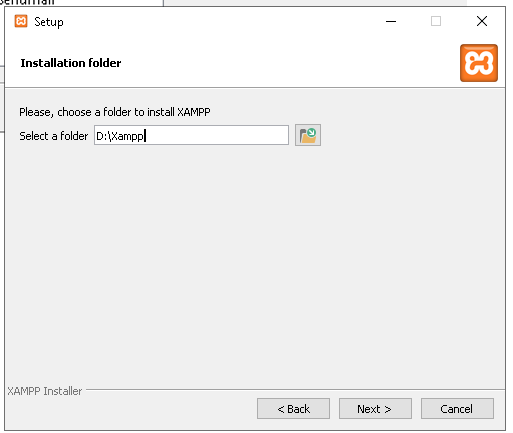
* Se abre la siguiente ventana => Next



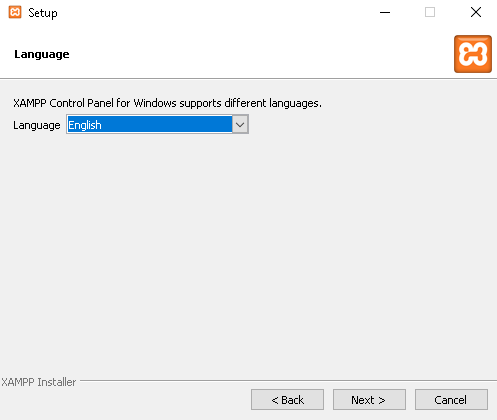
* Next



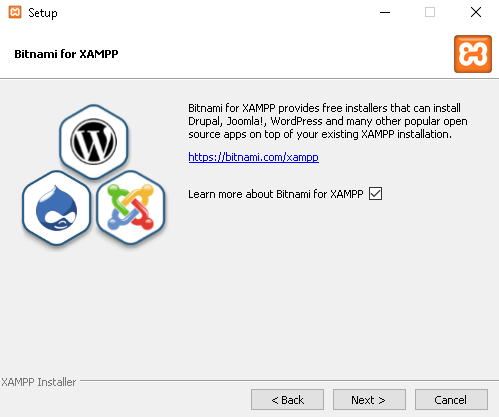
* Seleccionamos la ubicación donde queremos dejar los paquetes del programa => Next.



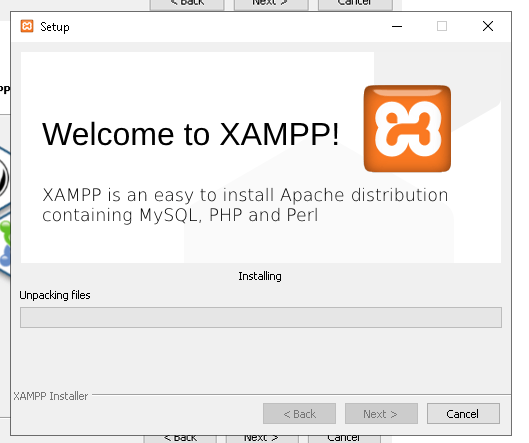
* Por defecto vine el idioma en ingles => Next



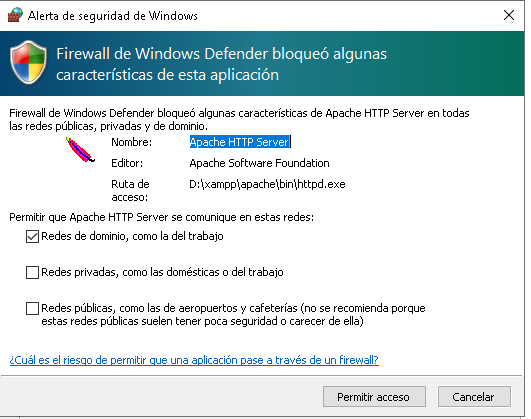
* Next



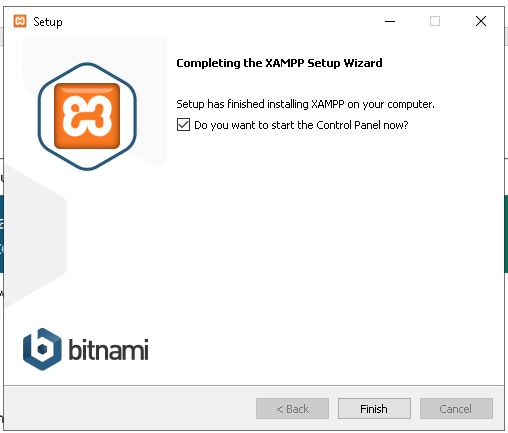
* Esperamos la instalación => Next



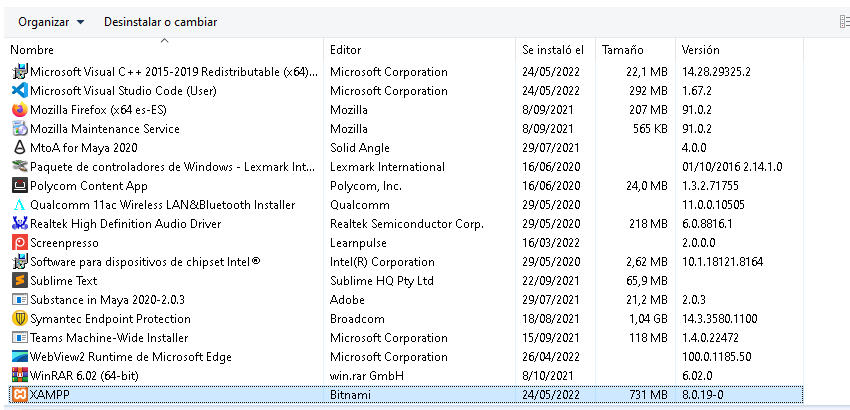
* Aceptar permiso => Permitir acceso



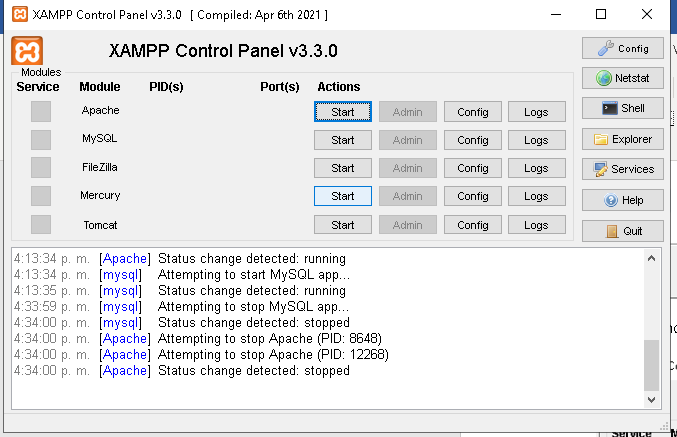
* Finish



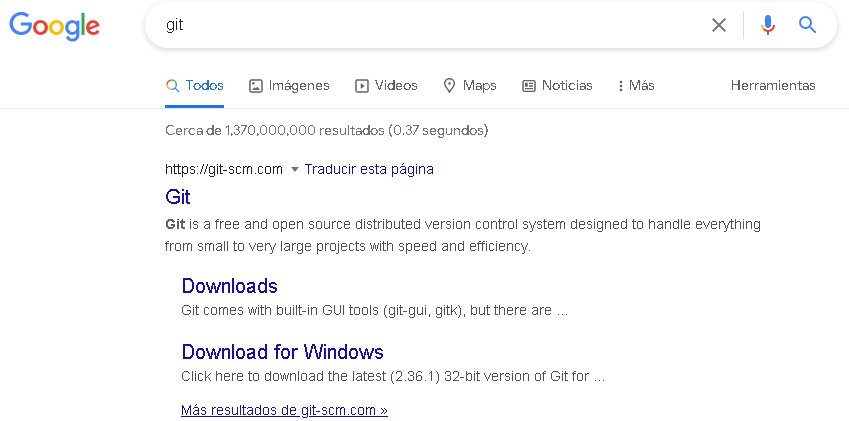
* Verificamos la versión de Xampp



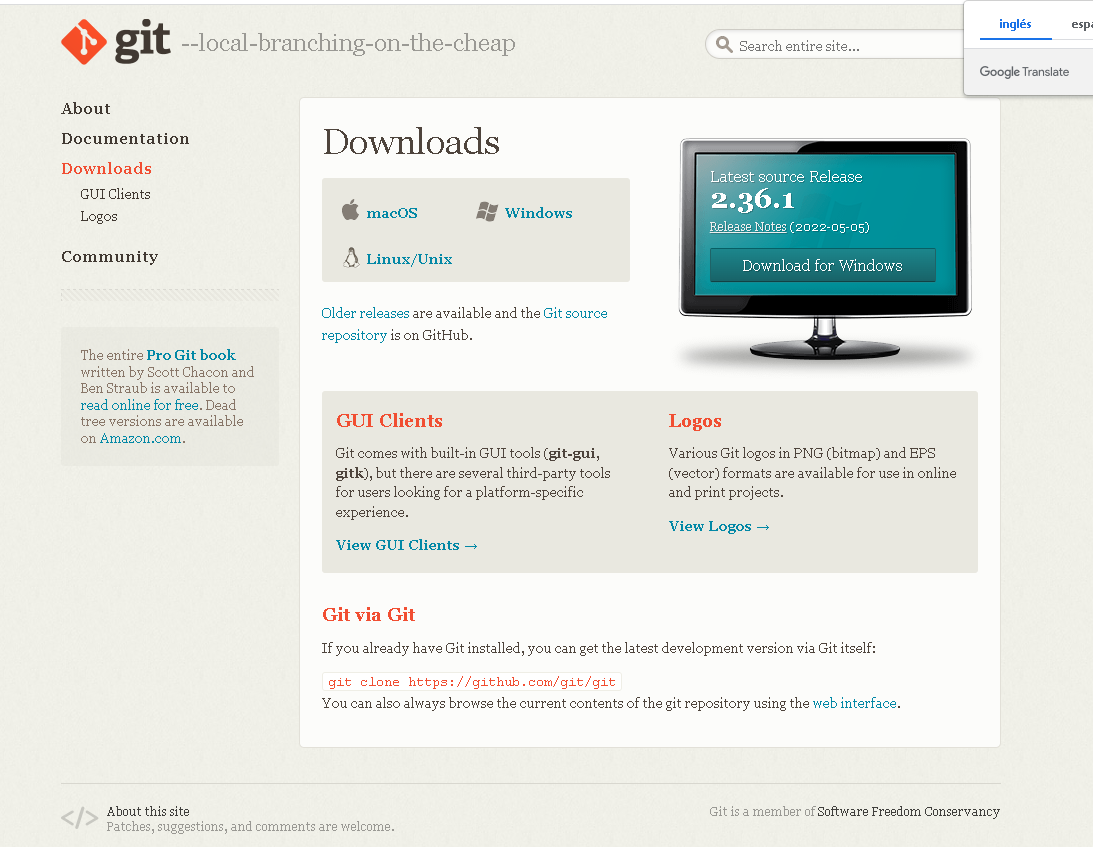
* Cuando abrimos Xampp, en este caso vamos a necesitar Apache y MySQL => Start



* 1. **INSTALACIÓN GIT**
* En nuestro navegador de preferencia buscamos git y procedemos a descargarlo



* Elegimos el sistema operativo, en este caso es Windows

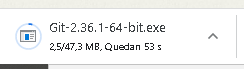


* Procedemos a elegir el tamaño que se adapte a la capacidad de nuestro pc.

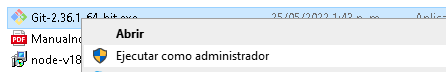
En este caso es de 64 Bit



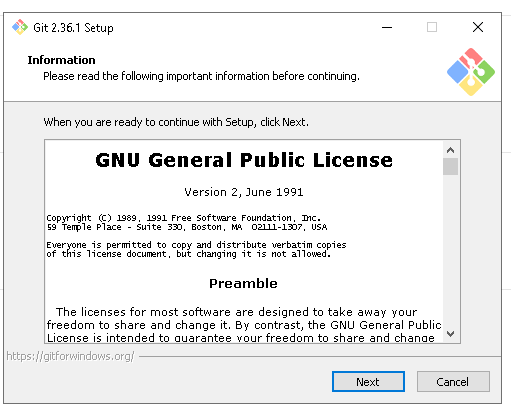
* Esperamos que se descargue



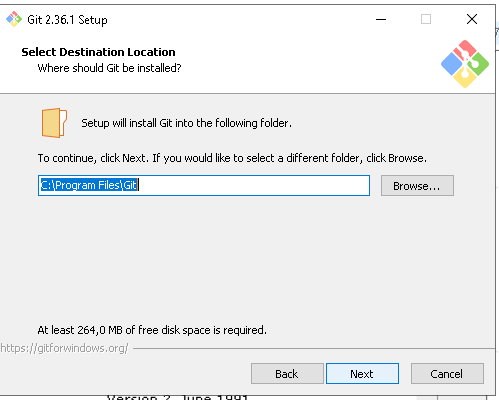
* Vamos a descargas y abrimos git, click derecho y ejecutar como administrador.



* Next



* Elegimos el lugar donde queremos guardarlo => Next



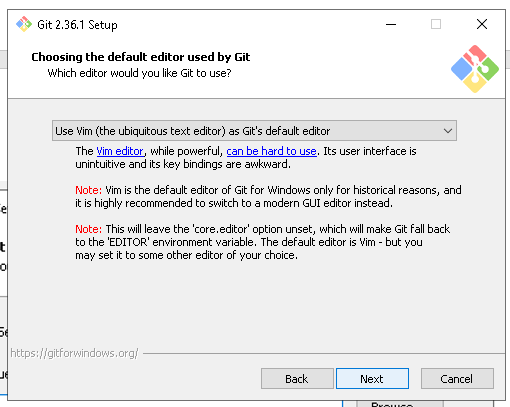
* Dejamos tal como está => Next



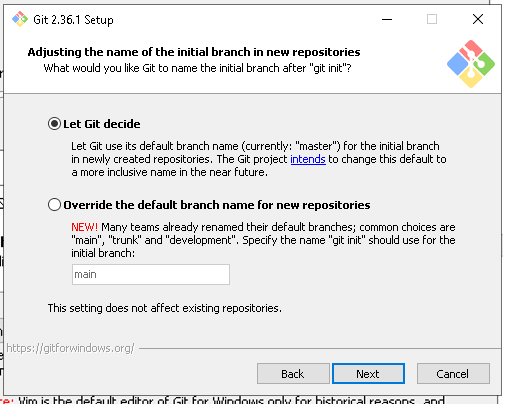
* Next



* Next



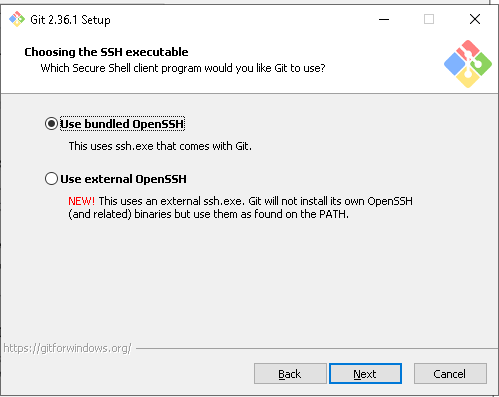
* Dejamos tal como está =>Next



* Dejamos tal como está => Next



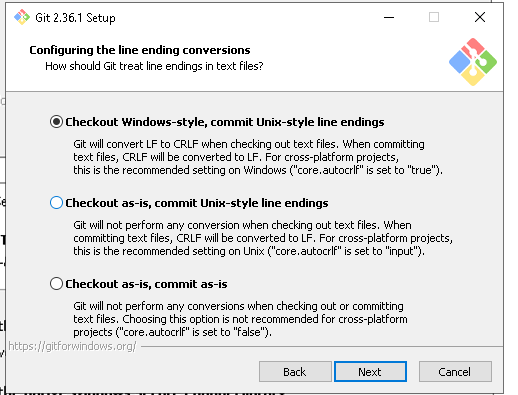
* Next



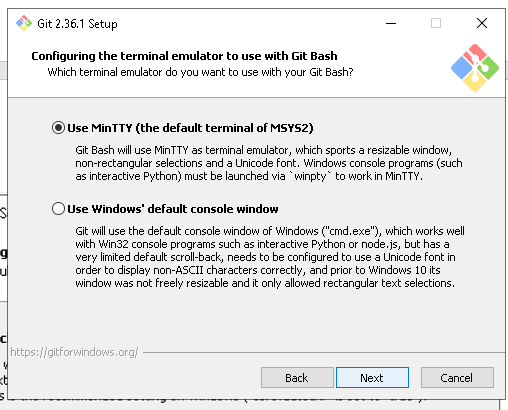
* Next



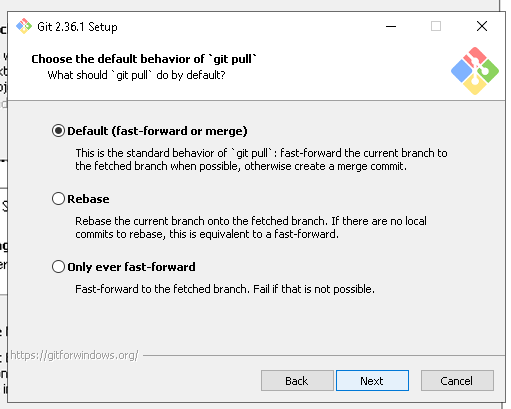
* Seleccionamos el primero => Next



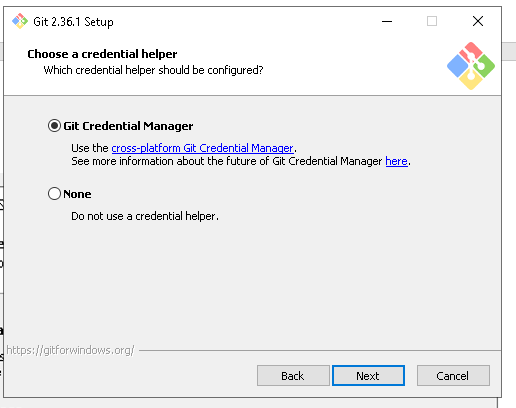
* Seleccionamos el primero => Next



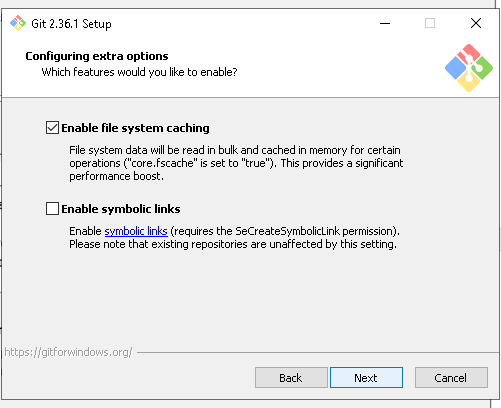
* Seleccionamos el primero => Next



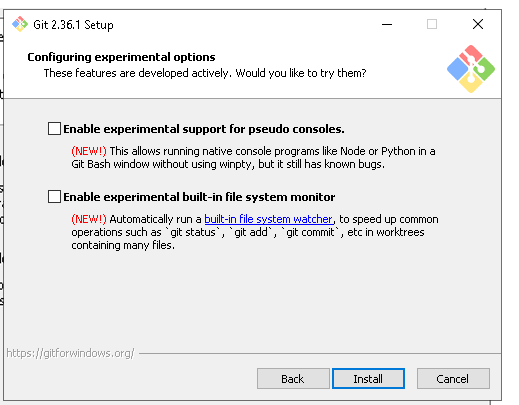
* Seleccionamos el primero => Next



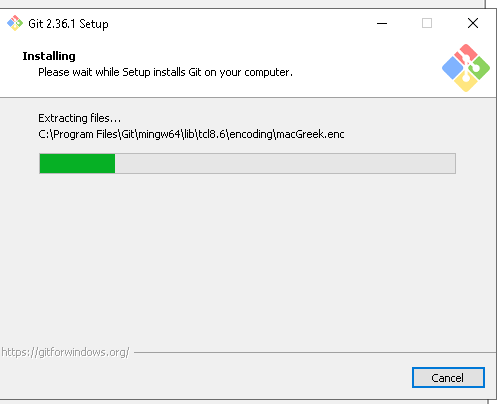
* Seleccionamos el primero => Next



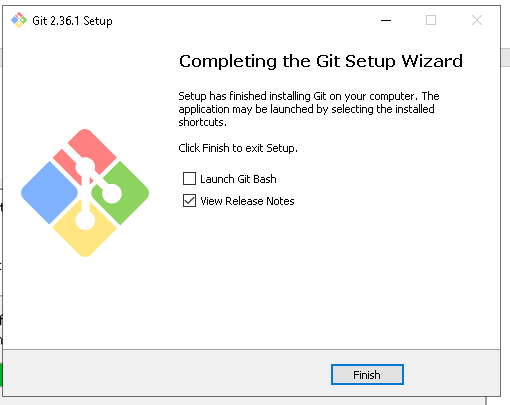
* En este caso no seleccionamos nada => Install



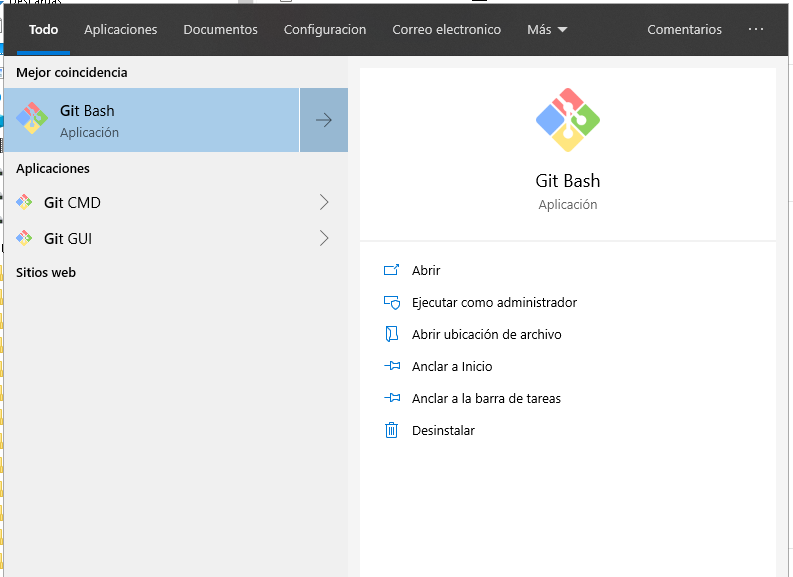
* Esperamos que se extraiga los archivos



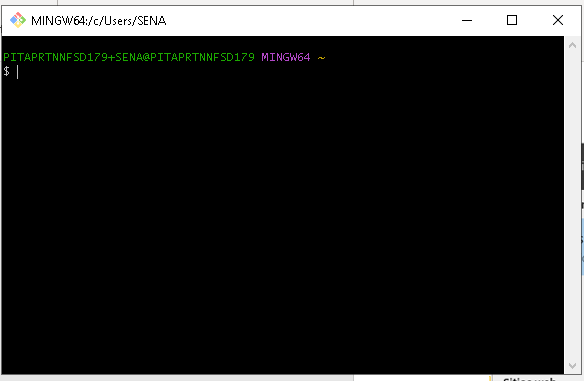
* Finish



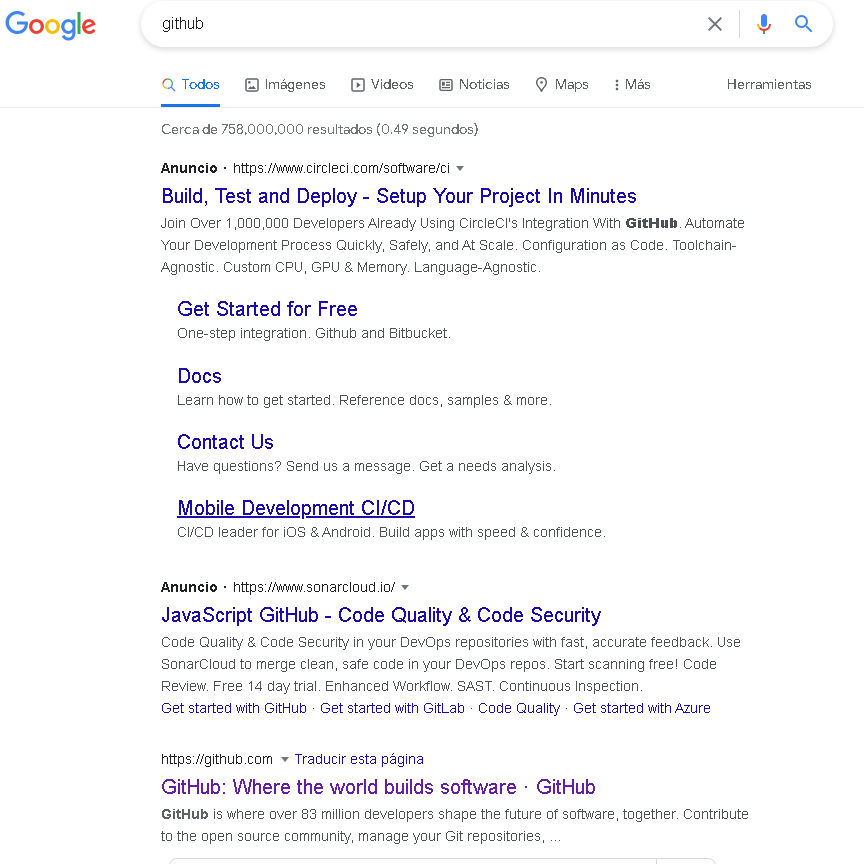
* Buscamos si quedo instalado nuestro Git Bash



* A la hora de abrirlo nos aparecerá así

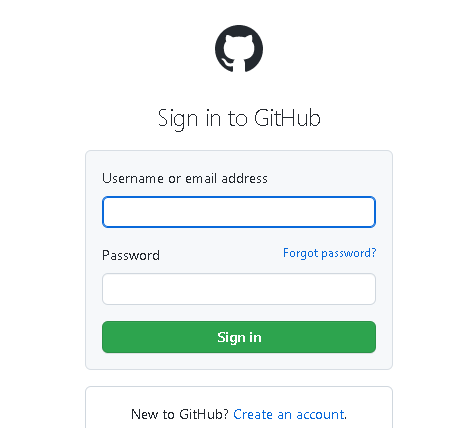


* 1. **VINCULACION DE REPOSITORIOS GIT Y GITHUB.**
* Vamos al navegador, ingresamos a Git Hub iniciamos sección

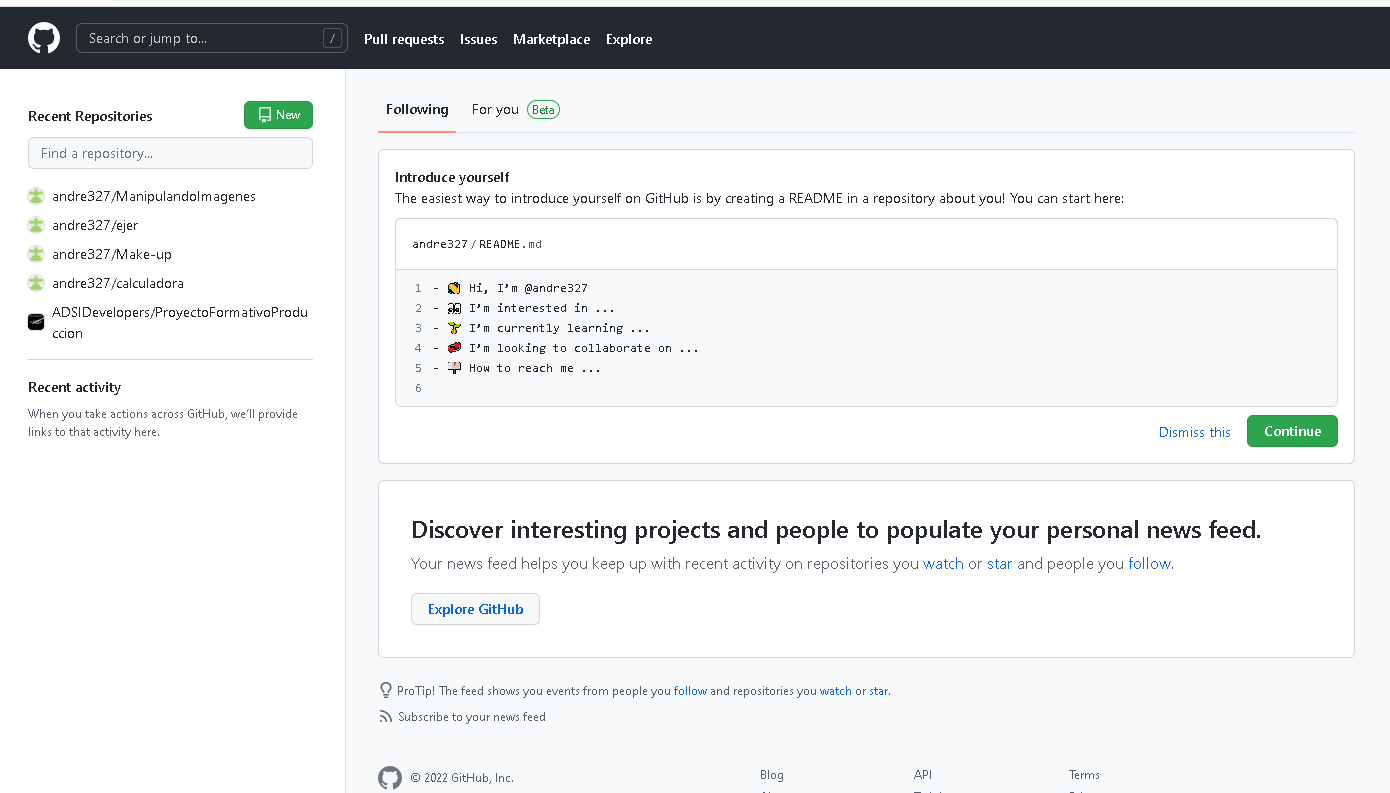


g

* Iniciamos sesión



* Una vez iniciemos sesión nos aparecerá la siguiente ventana.

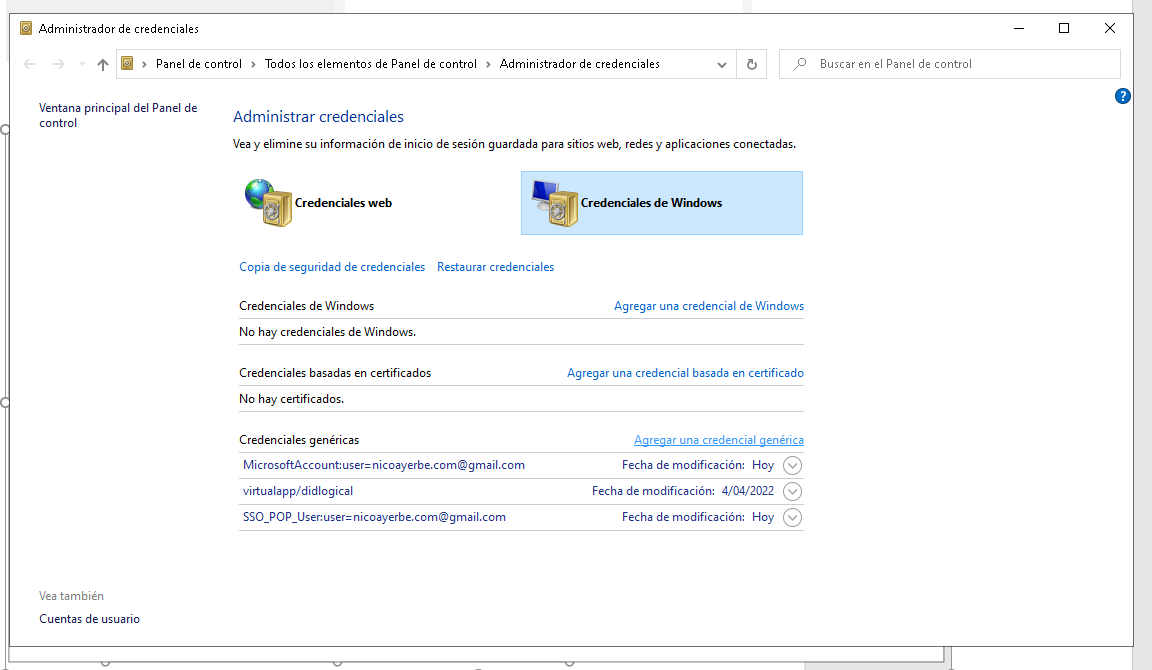


* De esta manera colocamos las credenciales de github en nuestro pc.

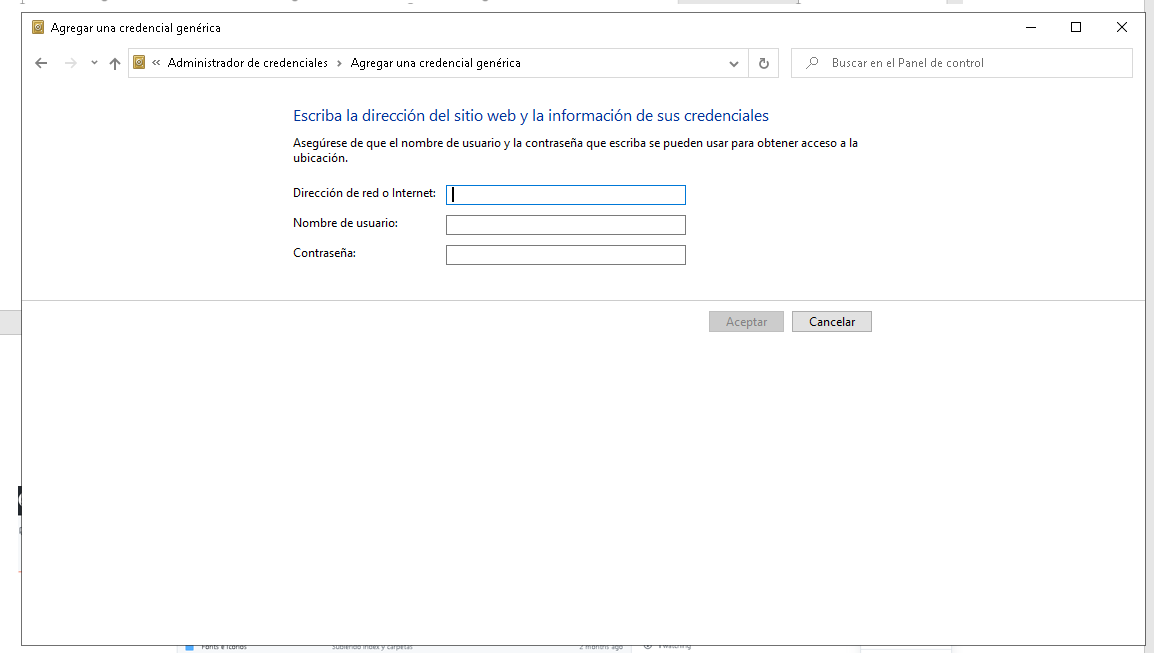
Vamos al buscador de nuestro pc y escrbimos => Administrador de credenciales => Abrir

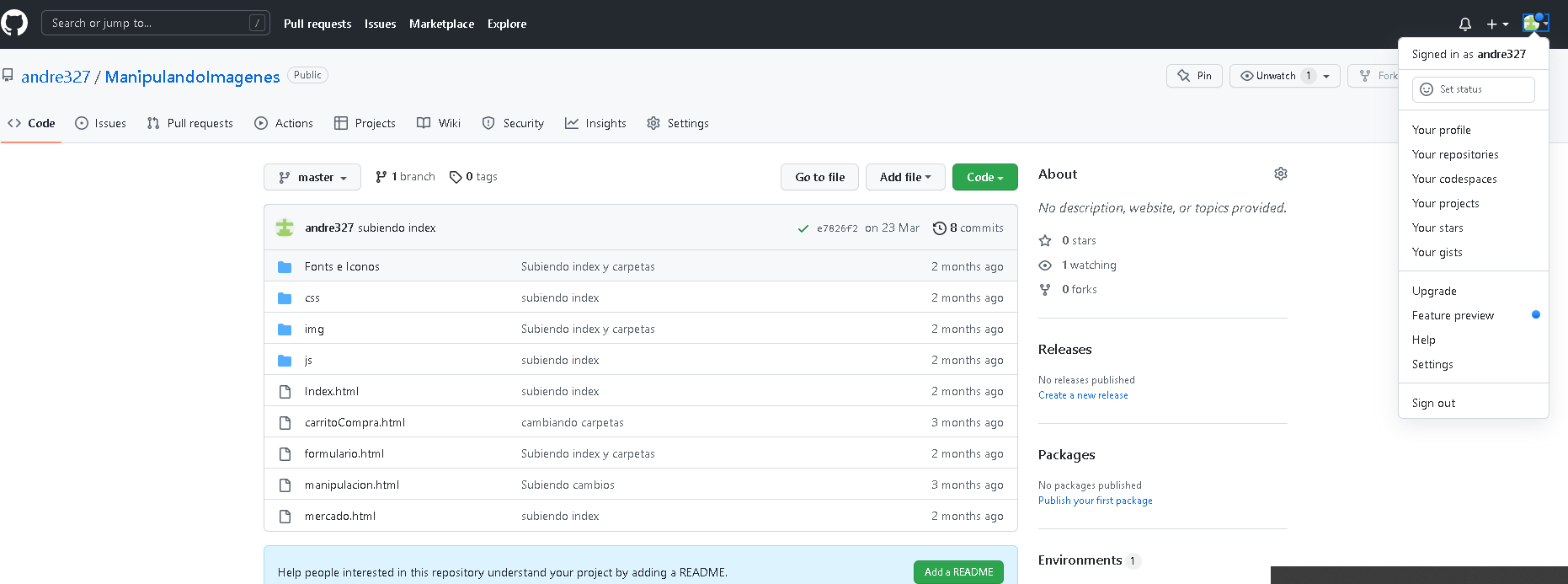


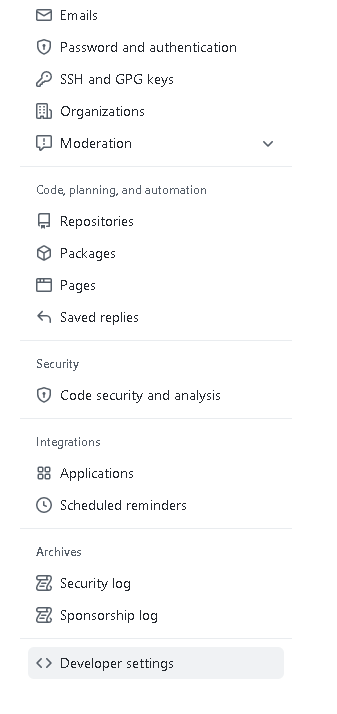
* Seleccionamos Credenciales de Windows => Agregar una credencial gen­­­­­­­­­­­­­érica.



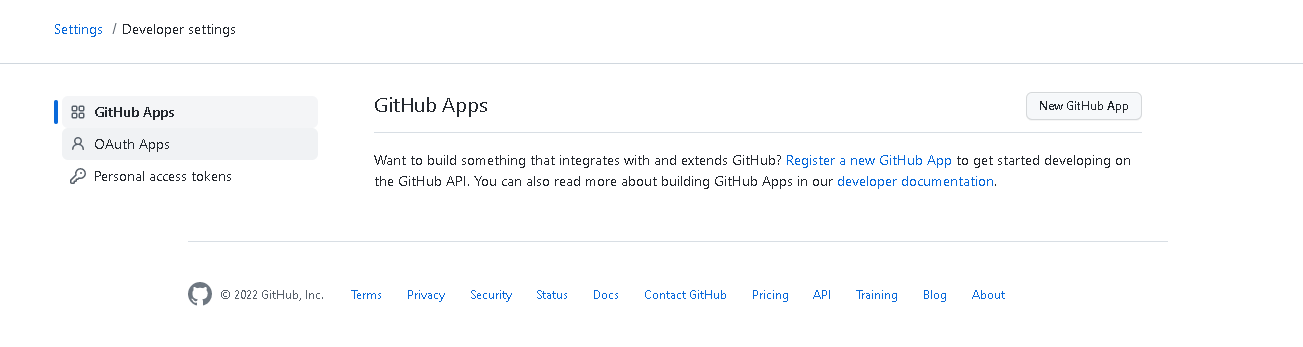
* Aparecerá la siguiente ventana en la que tendremos que colocar la dirección de red, nombre de usuario de GitHub y credenciales de GitHub.

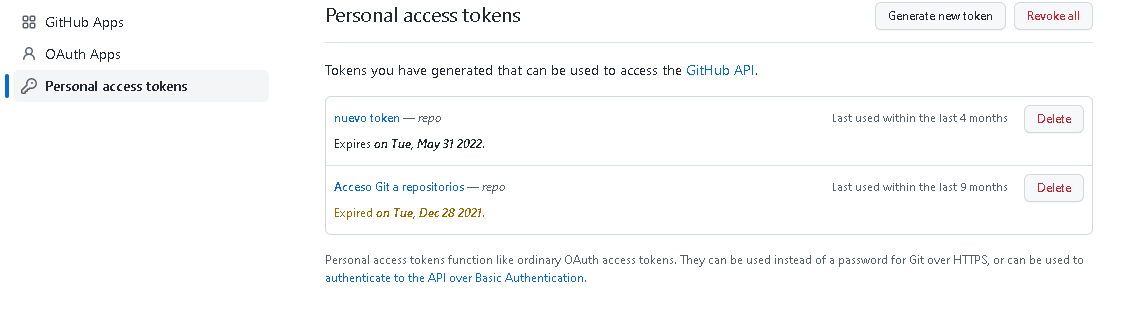


* Para sacar las credenciales de GitHub, nos dirigimos a la página y en la parte superior derecha aparece un icono, seleccionamos y al desplegarse el menú seleccionamos “Settings”
* Luego, aparecerá la siguiente lista y elegiremos developer settings que se encuentra al final.

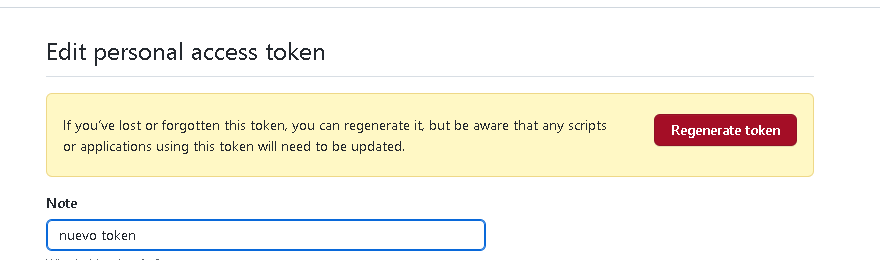


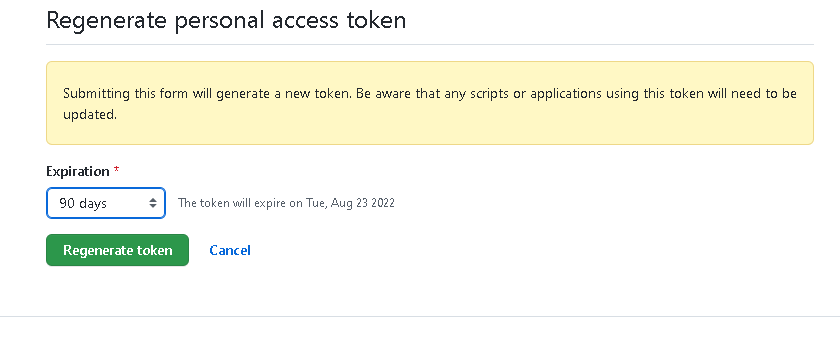
* Nos llevará a la siguiente vista y seleccionamos primero “Personal Access tokens” y luego “Nuevo token”



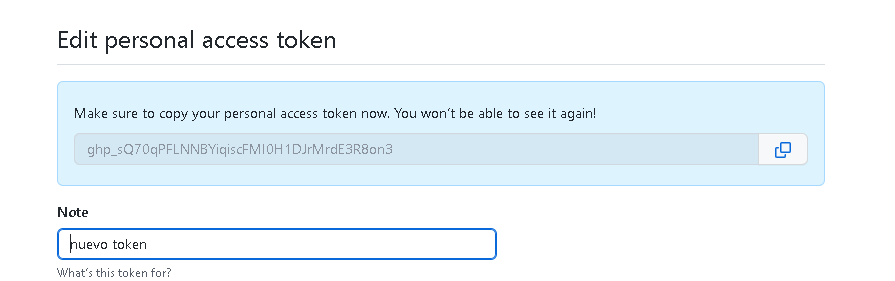


* En la siguiente vista seleccionamos “Regenerate token”, le damos un nombre, y el tiempo de expiración y seleccionamos Regenerate token

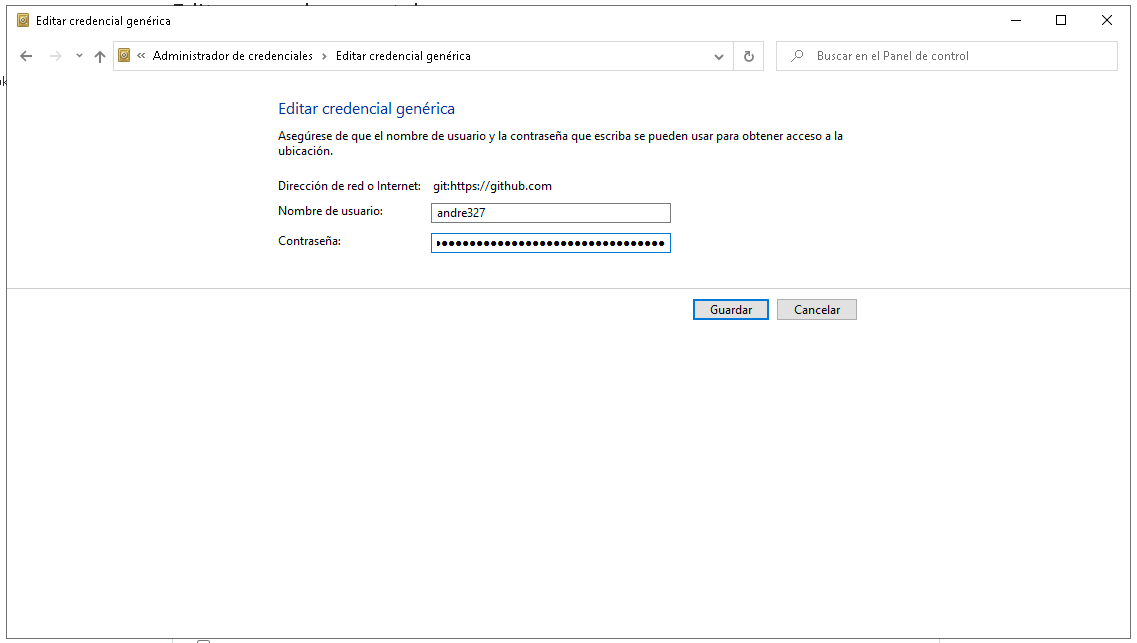




* Al generarse la credencial debemos copiarla

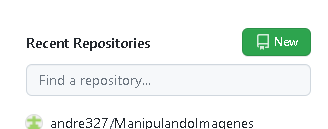


* Al ingresar la dirección, el nombre de usuario, procedemos a pegar la credencial en el campo de contraseña => Guardar



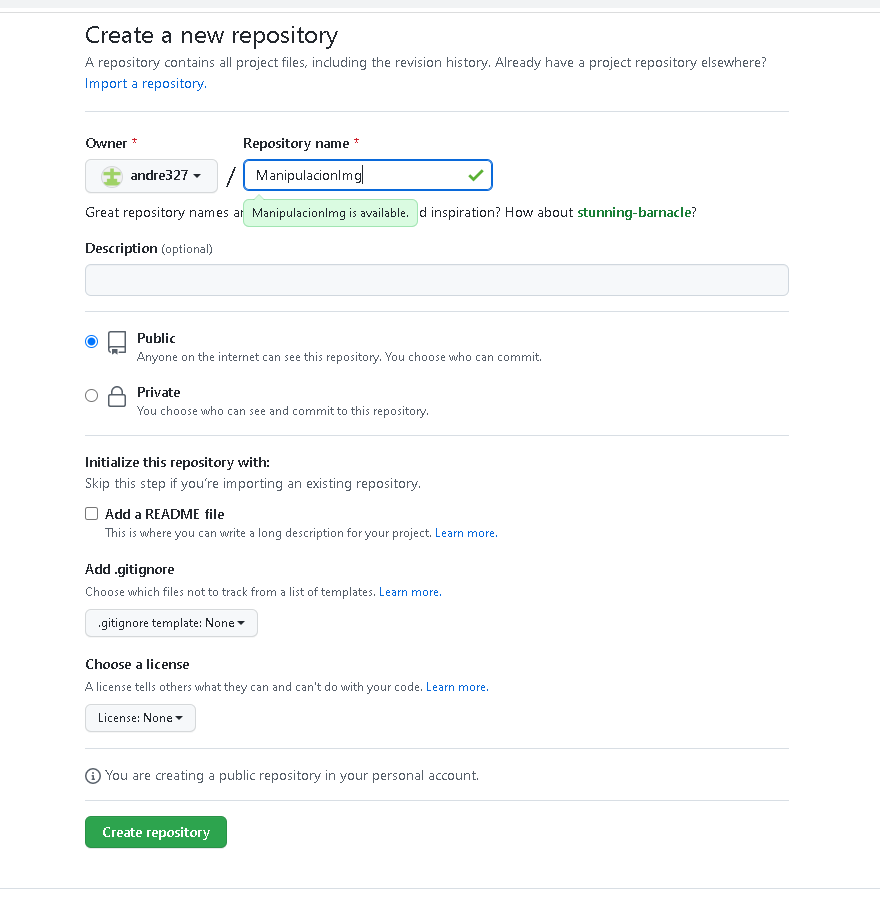
* Para crear un repositorio en GitHub.

En la parte izquierda de la página aparece New

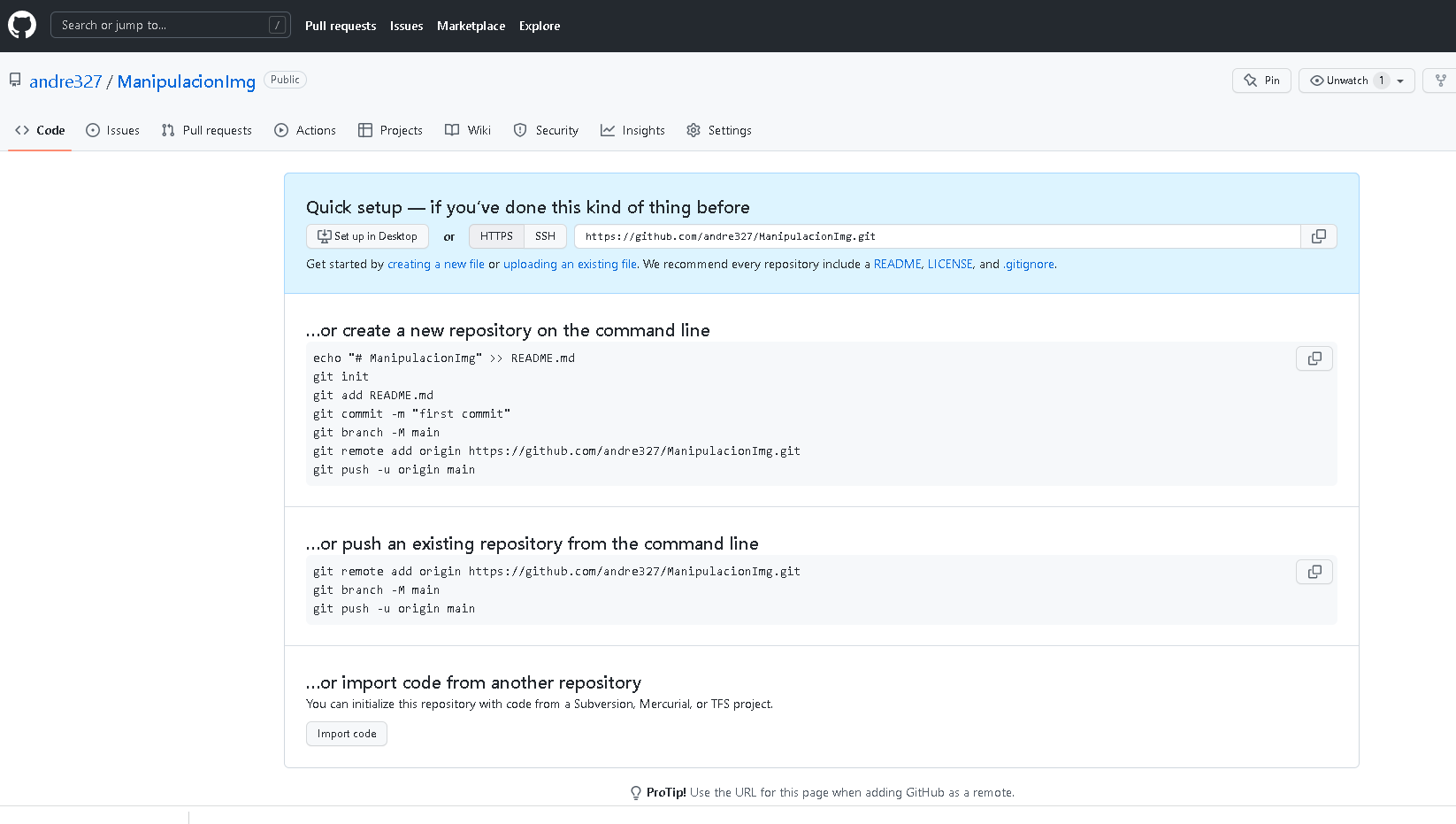


* Se mostrará lo siguiente, le damos un nombre, seleccionamos “Public” y

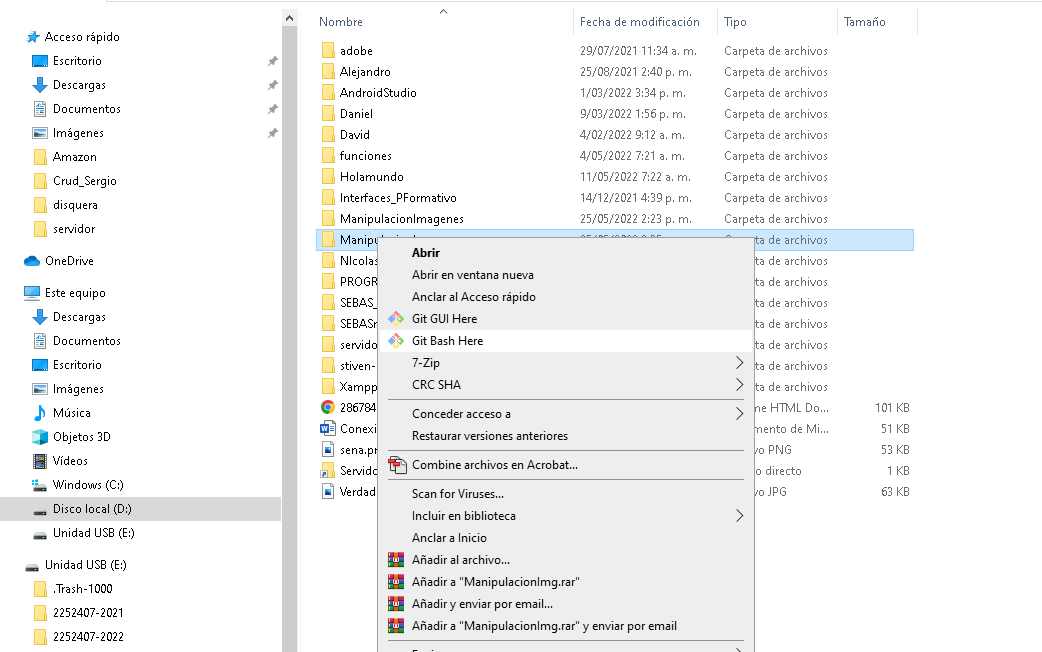
“Create repository”



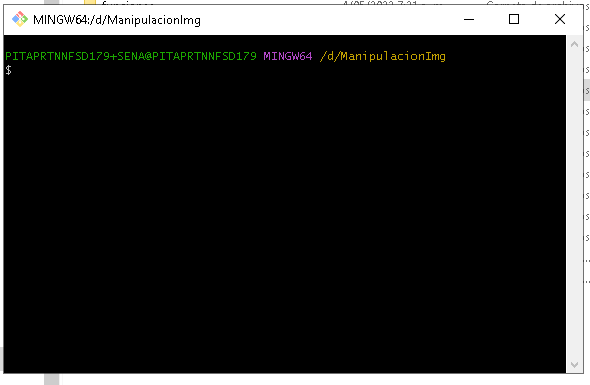
* Y aparecerá lo siguiente…



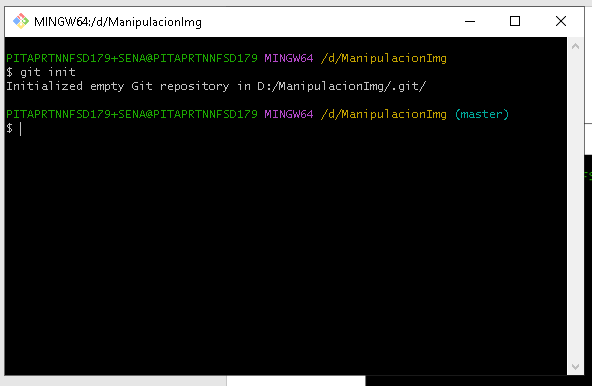
* Para poder subir nuestros archivos a GitHub y tener conocimientos básicos del mismo seguiremos los siguientes pasos.
* Nos dirigimos al equipo, la carpeta donde se encuentran los archivos que queremos subir a GitHub => click derecho y seleccionamos Git Bash Here.

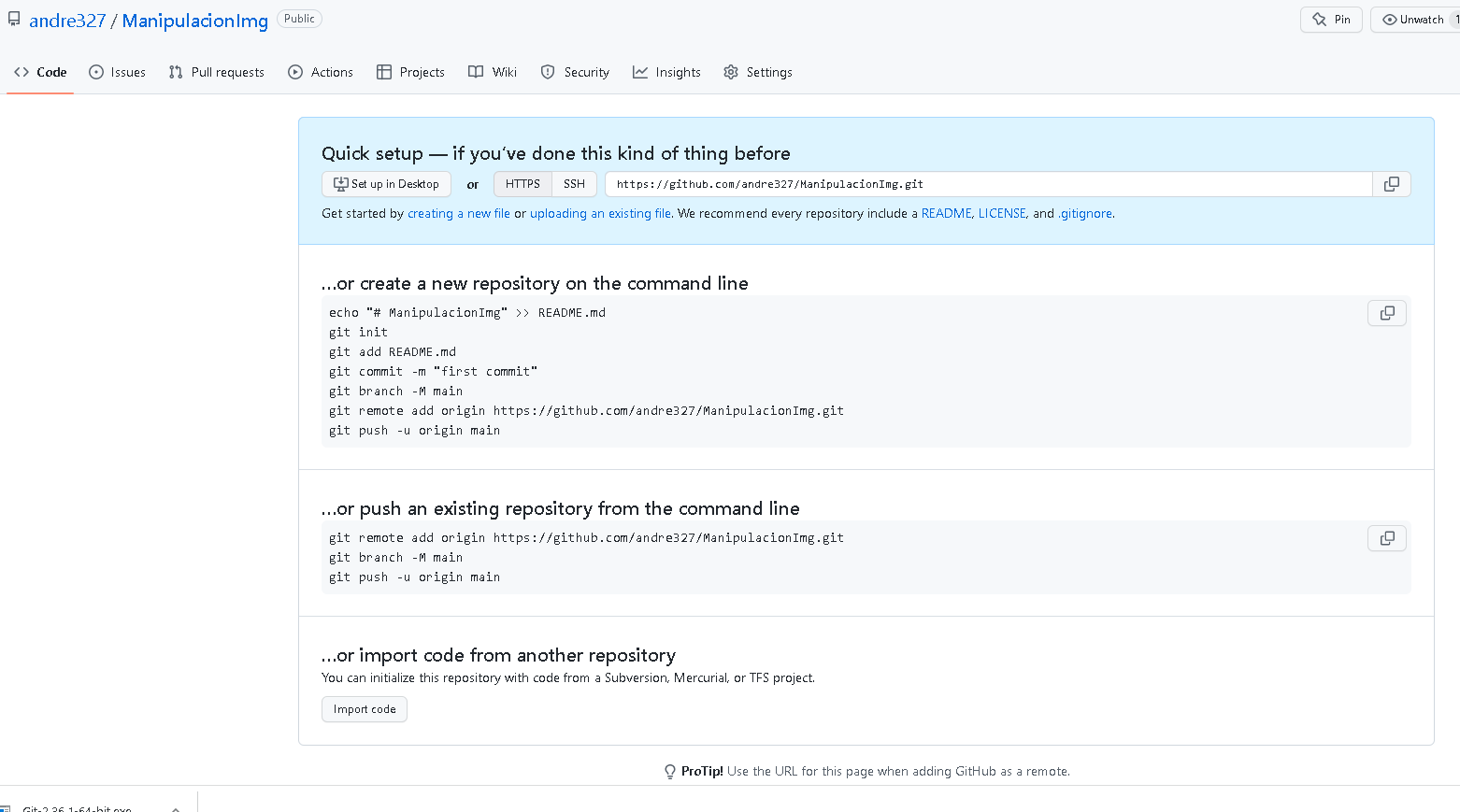


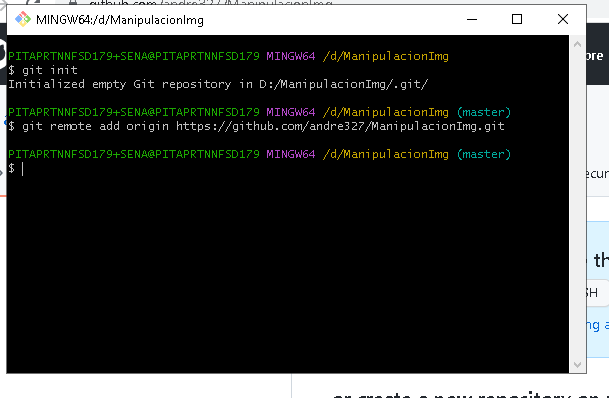
* Se abre la siguiente ventana con la ubicación de la carpeta.



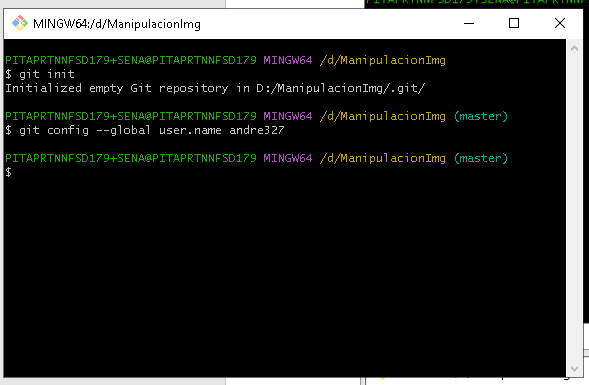
* Para inicializar el repositorio escribimos el comando git init y damos enter.



* Conectamos el repositorio remoto con el comando git remote add origin add + el link que aparece en gitHub



* Seguidamente vamos a ingresar el usuario con el que vamos a realizar cambios o subir archivos al repositorio utilizando el comando git config –global user.name + su nombre de usuario.

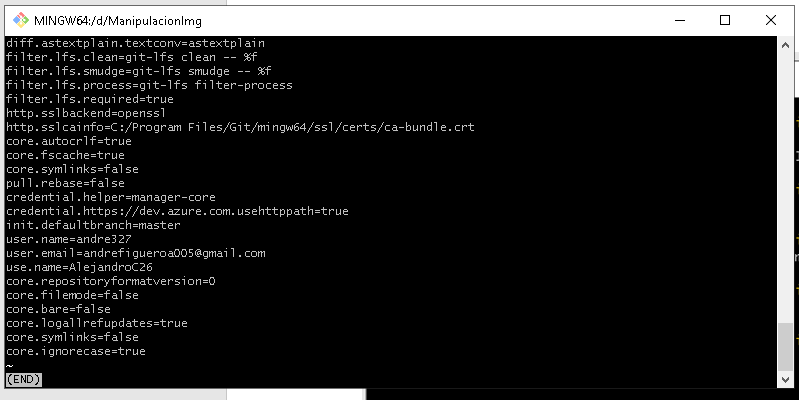


* Seguidamente vamos a ingresar el correo con el que vamos a realizar cambios o subir archivos al repositorio utilizando el comando git config –global user.email + su correo.



* Para verificar si el repositorio local está conectado con la rama remota. Al arrojar información confirma que sí, y si no arroja significa que no está conectado con ninguna rama. El comando git remote -v

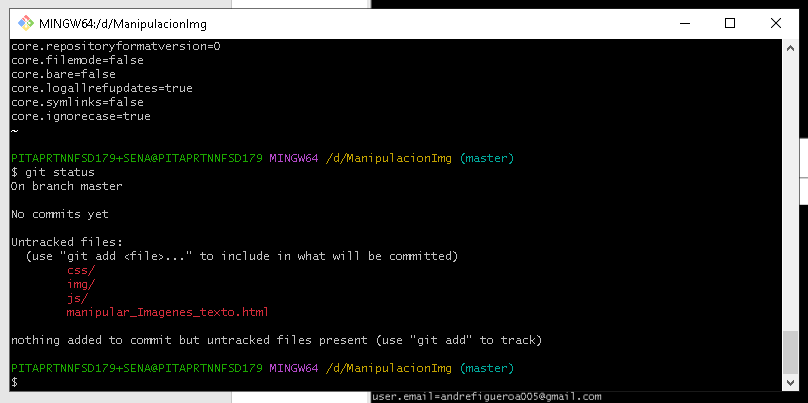




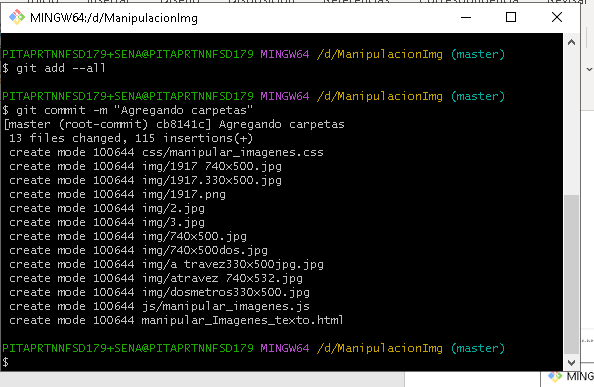
* Para guardar y poder ingresar otro comando :wq



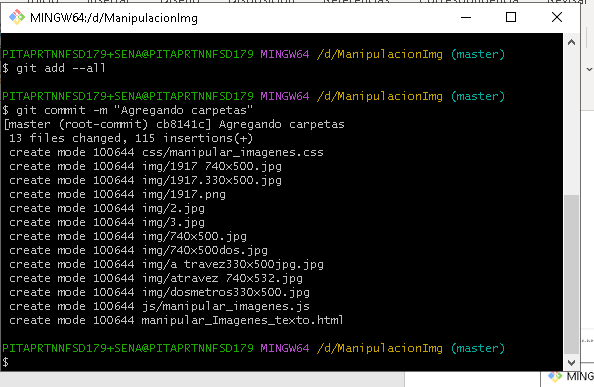
* Verificamos el estado de los archivo con el comando git status, si aparecen archivos en color rojo es porque no han sido añadidos y si aparecen archivos en color verde es porque están pendientes de confirmar.



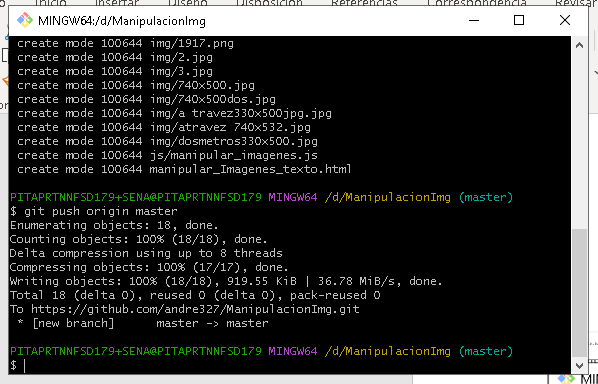
* Con el comando git add –all para añadir los archivos al Staing Area. Archivos pendiente.



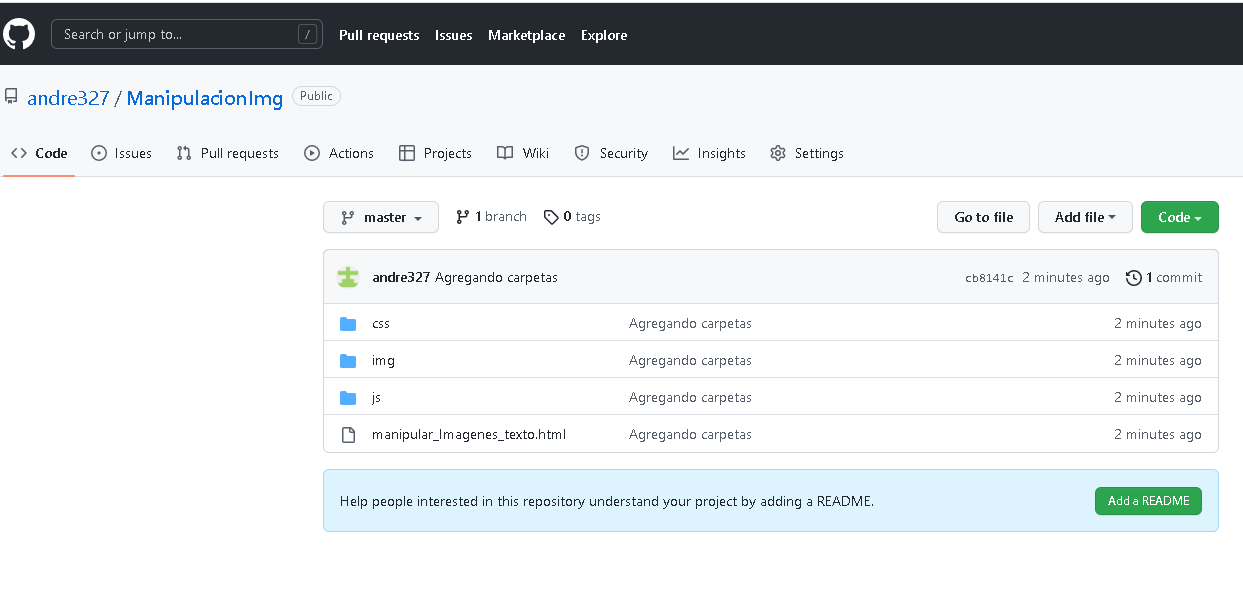
* Ingresamos un comentario para confirmar los archivos que vamos a añadir con el comando git commit –m “mensaje”



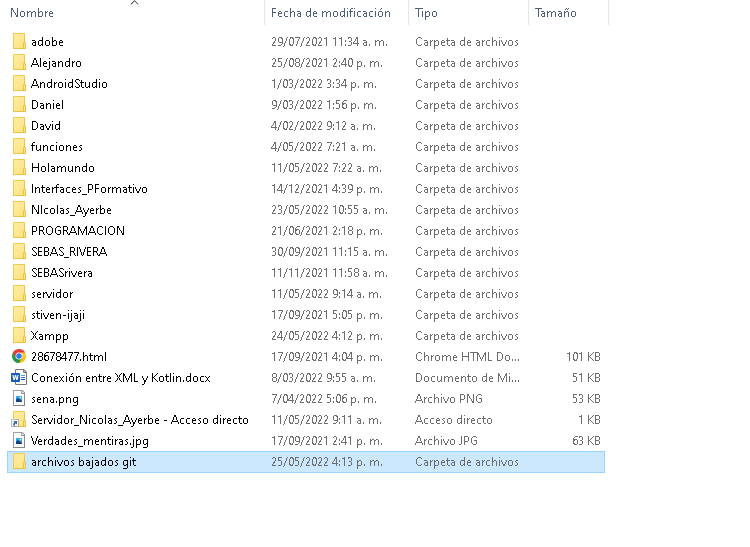
* Con el comando git push origin master para subir los cambios al repositorio remoto.



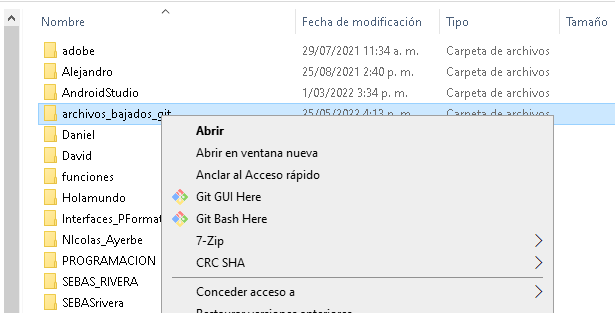
* Verificamos en el repositorio si los archivos están.



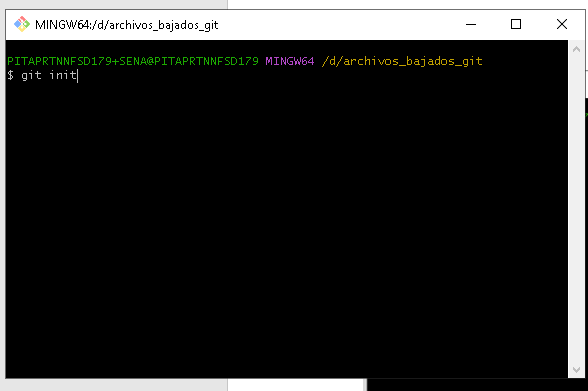
* Para bajar archivos del repositorio remoto al repositorio local haremos lo siguiente.
* Creamos una carpeta nueva.



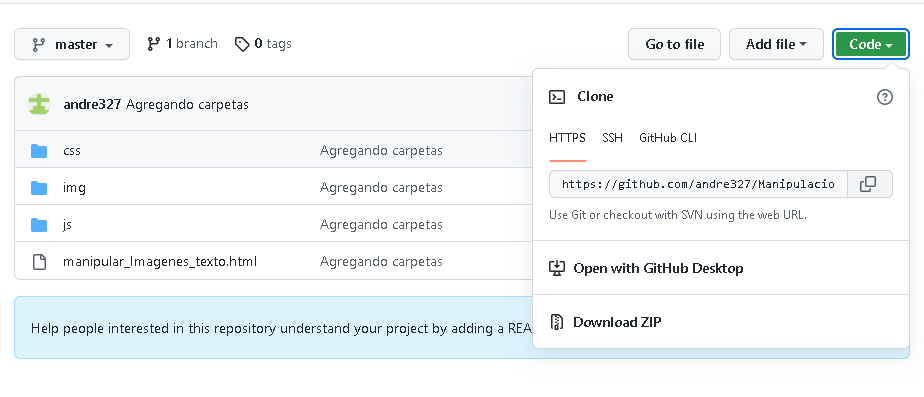
* Click derecho sobre la carpeta y seleccionamos Git Bash Here.



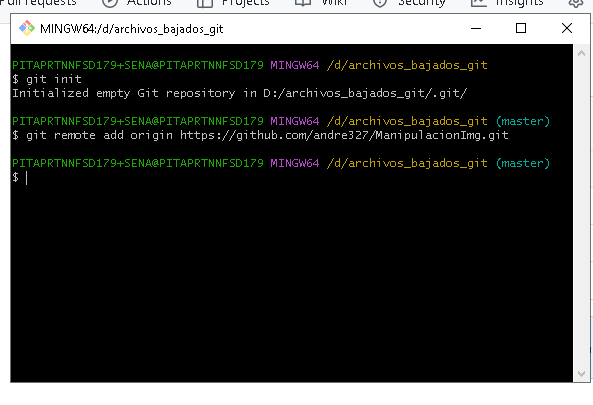
* Se abre la ventana con la ubicación de la carpeta, ingresamos el comando git init.



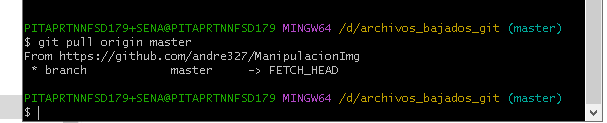
* Copiamos el link del repositorio de gitHub.



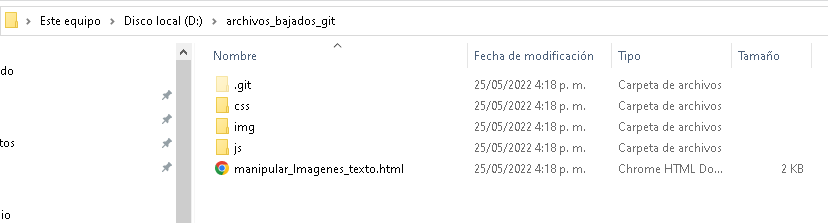
* Usamos el comando git remote add origin + pegamos el link



* Usamos el comando git pull origin master que es el que baja todos los cambios del repositorio remoto.



* Verificamos en la carpeta que previamente habíamos creado si se bajaron los archivos correspondientes.



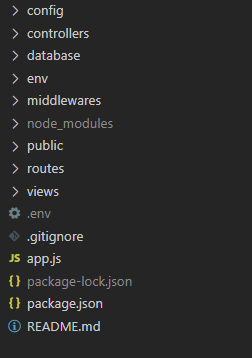
1. **ESTRUCTURA Y DIAGRAMAS**

**7.1 Estructura.**

Nuestro proyecto está desarrollado bajo el patrón de diseño modelo, vista, y controladores MVC. Las ventajas de implementar este patrón de diseño para nuestro proyecto son:

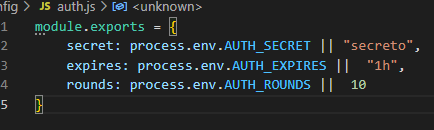
* Genera escalabilidad si es requerido.
* Permite dividir el trabajo entre un grupo, ya que sus componentes separados entre sí.
* Separación clara de dónde tiene que ir a cada tipo de lógica, facilitando el mantenimiento y la escalabilidad de nuestra aplicación.
* Sencillez para crear distintas representaciones de los mismos datos.
* Recomendable para el diseño de aplicaciones web compatibles con grandes equipos de desarrolladores y diseñadores web que necesitan gran control sobre el comportamiento de la aplicación.
* Facilita el manejo de errores.

En la siguiente imagen se ilustrar la estructuración del proyecto. Tenemos 6 capas de desarrollo, pero el patrón de diseño es modelo, vista y controlado. Las 3 capas restantes complementan las capas mencionadas anteriormente.

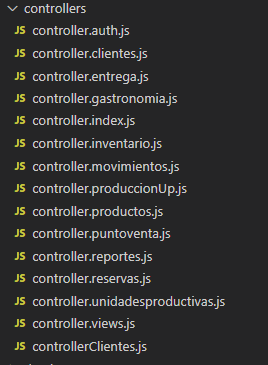


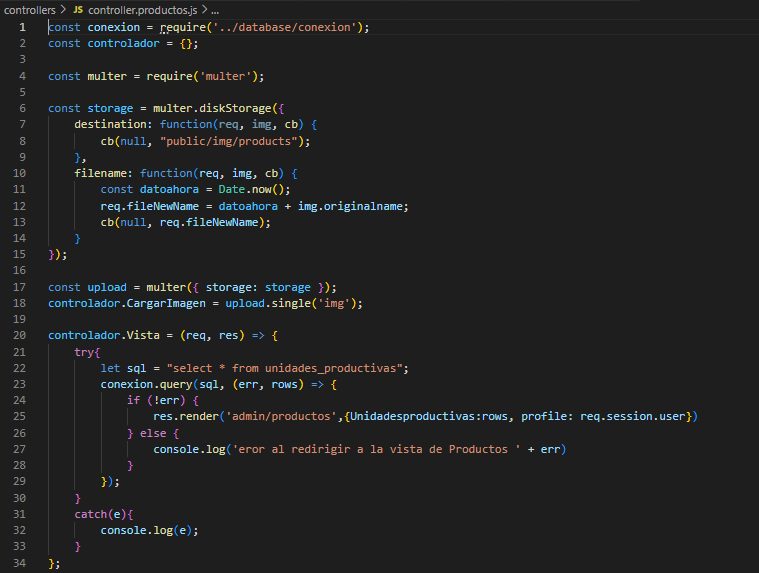
**1.Config**

En esta capa encontramos las variables de entrono que guardan las configuraciones del token y la encriptación.



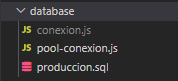
**2.Controladores**

Este componente se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas, en se encuentra el código necesario para responder a las acciones que se solicitan en la aplicación, una búsqueda de información es decir las consultas a la base de datos. Por medio de él se comunican el modelo y la vista: solicitando los datos necesarios; manipulándolos para obtener los resultados; y entregándolos a la vista para que pueda mostrarlos.

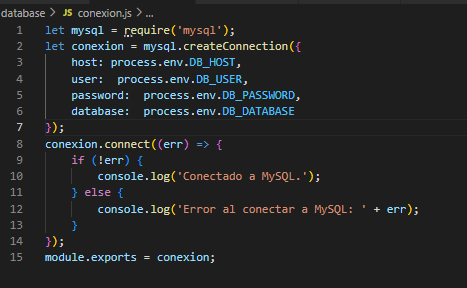


**3.database**

En esta capa encontramos la configuración para la conexión de la base de datos, la base de datos y el pool conexión que hace una conexión asincrónica.



Declaración de parámetro de conexión.



**4. .env**

Los archivos .env se pueden considerar un estándar para el almacenamiento de variables de entorno. Estos archivos tienen un formato muy sencillo y fácil de escribir y de leer.

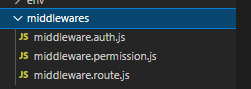
La sintaxis de los .env contiene pares clave (nombre de variable) y valor, separados por un carácter "=". Cada variable en una línea.

En esta capa encontramos las variables de entorno que nos permiten realizar la conexión a la base de datos.



**5. middlewares**

Es una función que se ejecuta en medio de una función es el mediador.

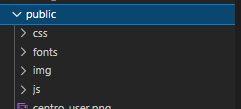


**6. node\_modules, package-lock.json, package.json**

Estas carpetas se generan automáticamente una vez hacemos la instalación de la paquetería mencionada anteriormente.

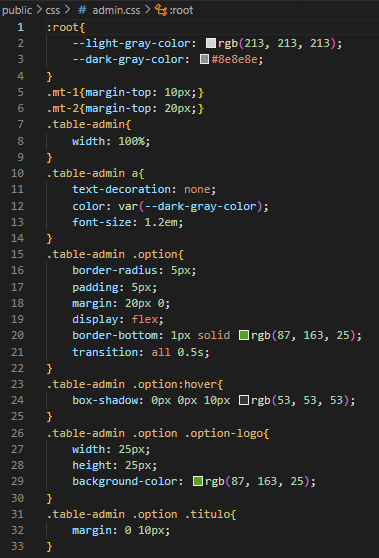
**7. Public**

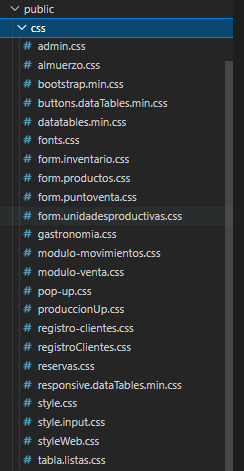
Aquí encontraremos las carpetas que alojan los archivos javascript, css, fuentes y las imágenes utilizadas.



**7.1 css**

Aquí encontraremos el código referente a los estilos de cada una de las vistas del sistema.

****



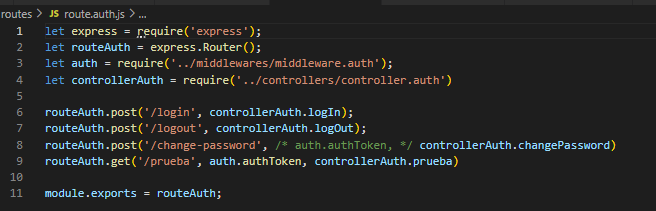
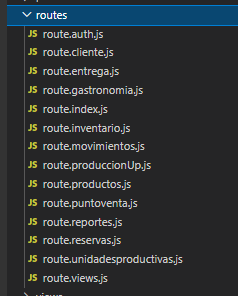
**7.2 js**

Aquí se encuentran los archivos que permiten añadir características interactivas al sitio web, (por ejemplo, eventos que ocurren cuando los botones son presionados o los datos son introducidos en los formularios, efectos de estilo dinámicos, animación, y mucho más).

****

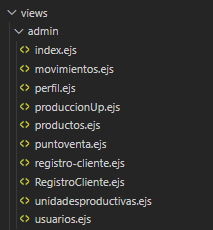
**8.Routes**

En esta capa se encuentran todos los archivos de las rutas las cuales permiten manejar los roles y configurar las vistas de nuestro proyecto de acuerdo a los mismos.



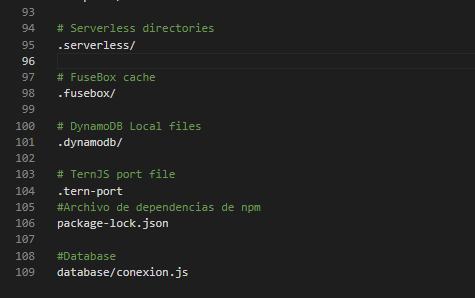
**9.views**

Este apartado contiene los archivos ejs con el marcado HTML y el contenido que se envía al explorador es decir contiene la lógica relacionada con la generación de la interfaz del usuario.



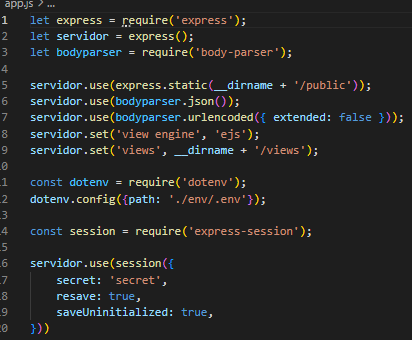
**10. Gitgnore**

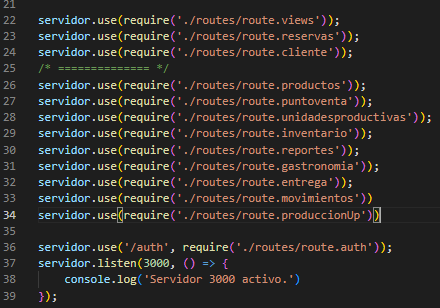
Gitignore, es un archivo de texto que le dice a Git qué archivos o carpetas ignorar en un proyecto. Un archivo local. gitignore generalmente se coloca en el directorio raíz de un proyecto. También puedes crear un archivo global.



**12. App.js**

En este archivo estructuramos creamos y configuramos el servidor, y además especificamos que se están usando en el proyecto.

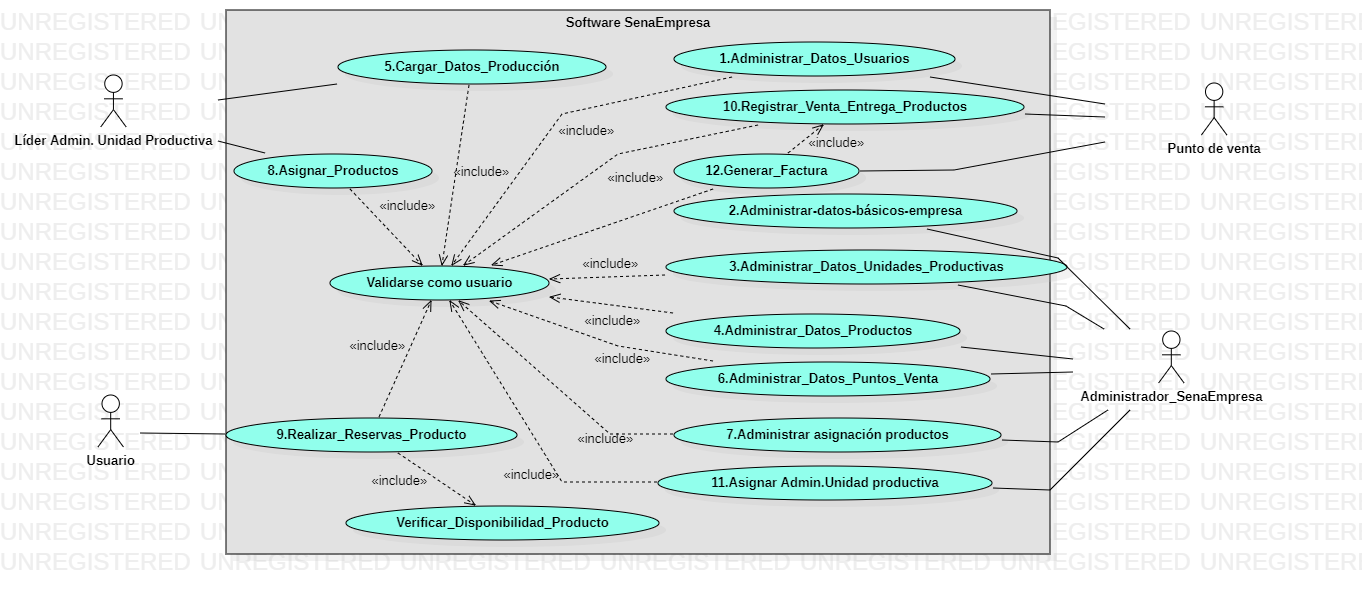




**7.2 Diagramas**

En este apartado encontramos algunos de los diagramas propios del proyecto.

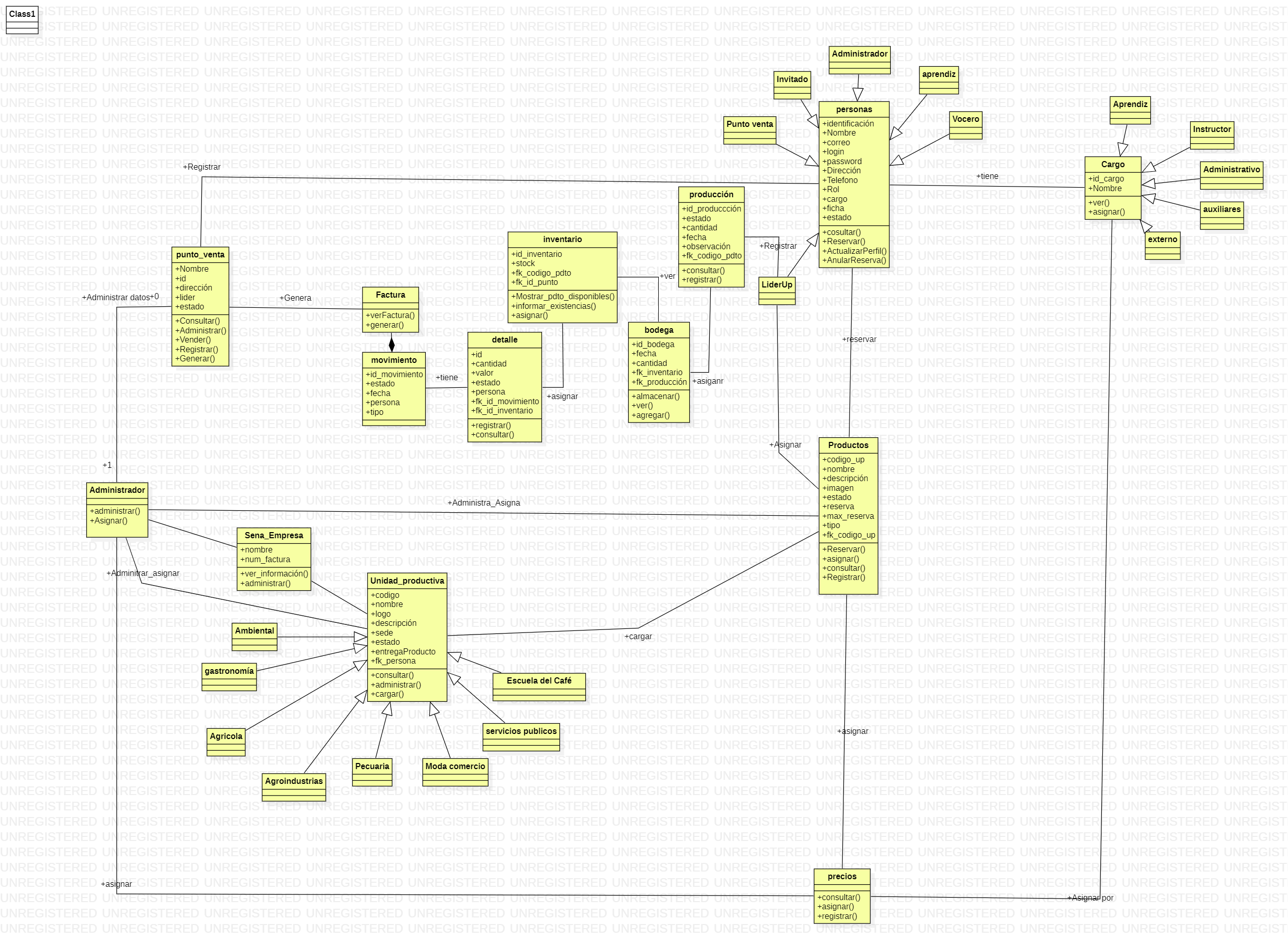
**Diagrama de caso de uso.**



**Diagrama de secuencias** del proceso de reservas.



**Diagrama de clases.**

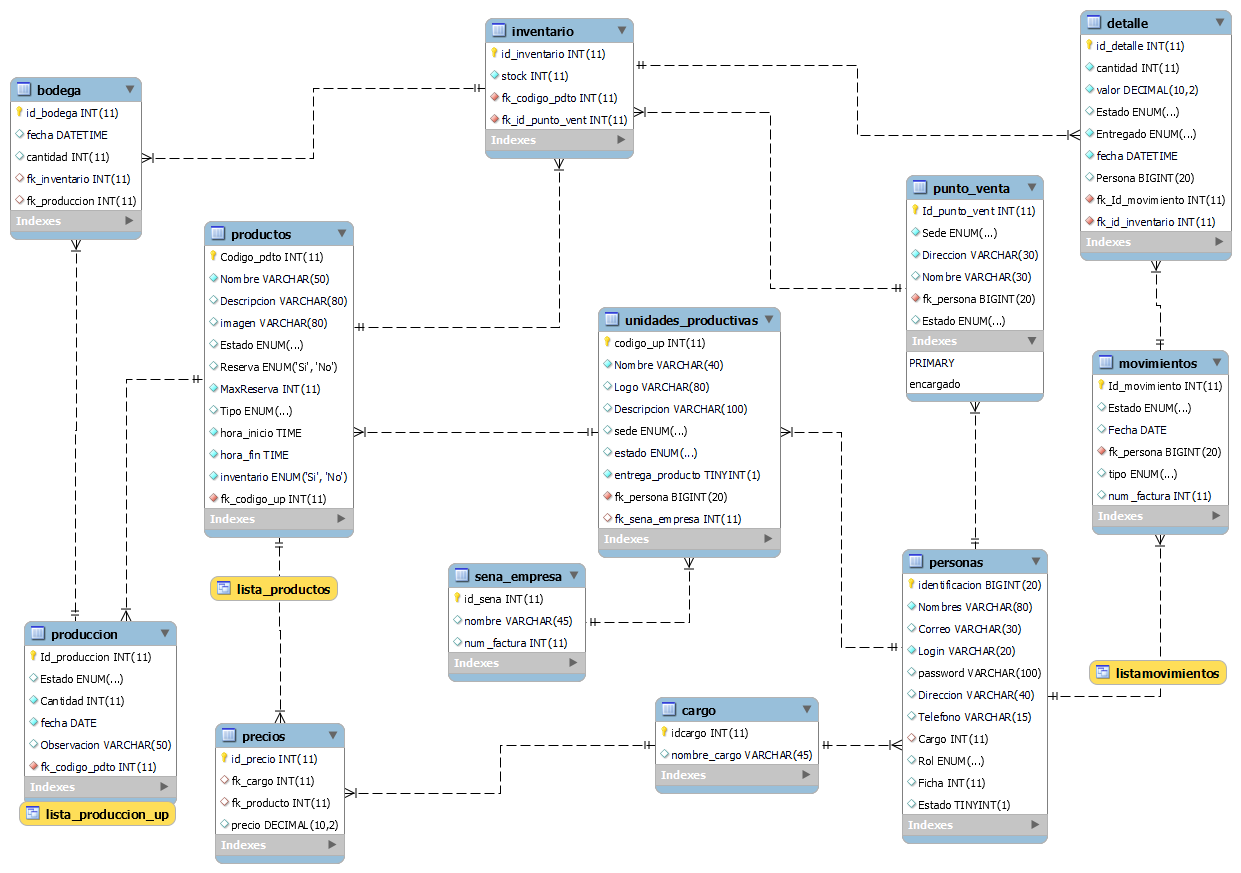
****

**Diagrama de actividades general**

**Diagramas de estado general**

**Modelo Relacional de la base de datos.**

En este diagrama se muestra las relaciones que hay entre las tablas para dar a entender mejor la base de datos, este modelo nos garantiza la integridad referencial eliminando todo lo relacionado con un registro cuando sea necesario.



1. **DICCIONARIO DE DATOS**

### Tabla bodega

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | id\_bodega | Int | Si | Identificado único de la bodega |
|  | Fecha | Datetime | No | Fecha de creación de la bodega |
|  | Cantidad | Int | No | Capacidad máxima de la bodega. |
| FK | fk\_inventario | Int | Si | Inventario de productos que maneja la bodega |
| FK | fk\_produccion | Int | Si | Producción realizada de los productos que maneja la bodega |

### Tabla Cargo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Idcargo | Int | Si | Identificador único del cargo |
|  | nombre\_cargo | Varchar(45) | No | Nombre que describe el cargo. |

TABLA DETALLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | id\_detalle | Int | Si | Identificador único del detalle |
|  | Cantidad | Int | Si | Cantidad del producto o servicio adquirido |
|  | Valor | Decimal(10,2) | Si | Precio de mercado del producto o servicio |
|  | Estado | Enum (Facturado, Anulado, Prestamo,  ) | No | Estado del producto o servicio adquirido |
|  | Entregado | Enum (Entregado, No entregado, No reclamado) | Si | Estado de entrega del producto o servicio adquirido |
|  | Persona | Bigint | No | Persona que adquiere un producto o servicio |
| FK | fk\_Id\_movimiento | Int | Si | Movimiento realizado en la adquisición de productos o servicios |
| FK | fk\_id\_inventario | Int | Si | Identificador del inventario |

### Tabla inventario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | id\_inventario | Int | Si | Identificador único de inventario |
|  | Stock | Int | Si | Cantidad de productos o servicios disponibles |
|  | fk\_codigo\_pdto | Int | Si | Productos o servicios en el inventario |
|  | fk\_id\_punto\_vent | Int | Si | Punto de venta de los productos o servicios. |

### Tabla movimientos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Id\_movimiento | Int | Si | Identificador único del movimiento |
|  | Estado | Enum (Reservado,Facturado,Anulado,Prestamo) | No | Estado en el que se encuentra el movimiento realizado por un cliente |
|  | Fecha | Date | No | Fecha en el que se realiza el movimiento |
|  | Fk\_persona | Bigint | Si | Persona que realiza el movimiento al adquirir un producto o servicio |
|  | Tipo | Enum(Grupal,Individual) | No | Tipo de movimiento que se realiza. |
|  | Num\_factura | int | No | Número consecutivo de facturación |

### Tabla Personas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Identificacion | Bigint | Si | Identificador único de la persona |
|  | Nombre | Varchar(80) | Si | Nombre de la persona |
|  | Correo | Varchar(30) | No | Correo electrónico de la persona |
|  | Login | Varchar(20) | Si | Nombre de usuario para ingreso |
|  | Password | Varchar(15) | Si | Contraseña de ingreso a la plataforma |
|  | Direccion | Varchar(40) | No | Dirección de residencia de la persona |
|  | Telefono | Varchar(15) | No | Teléfono de la persona |
|  | Cargo | Int | No | Cargo que ocupa la persona. |
|  | Rol | Enum(Invitado,Vocero,Lider UP, Punto venta,Admin,Aprendiz) | No | Rol que desempeña la persona en la plataforma |
|  | Ficha | Int | No | Ficha que pertenece la persona. |
|  | Estado | Tinyint | No | Estado actual de la persona, si esta activo o inactivo |

### Tabla Precios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Id\_precio | Int | Si | Identificador único de los precios |
|  | Fk\_cargo | Int | No | Dependiendo del rol del cliente se establece un precio |
|  | Fk\_producto | Int | No | Producto al cual se le establece el precio |
|  | Precio | Decimal(10,2) | No | Precio del producto |

### Tabla producción

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Id\_produccion | Int | Si | Identificador único de la producción |
|  | Estado | Enum(Producido, aceptado, rechazado. | No | Estado en el que se encuentra la producción |
|  | Cantidad | Int | Si | Número de unidades producidas |
|  | Fecha | Date | Si | Fecha en que se realiza la producción |
|  | Observacion | Varchar(50) | No | Detalles adicionales de la producción |
|  | fk\_codigo\_pdto | Int | Si | Producto principal de la producción |

### Tabla productos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Codigo\_pdto | Int | Si | Codigo único del producto |
|  | Nombre | Varchar(50) | Si | Nombre del producto |
|  | Descripcion | Varchar(80) | No | Descripción del producto |
|  | Imagen | Varchar(80) | No | Imagen gráfica del producto |
|  | Estado | Enum(Activo, inactivo) | No | Estado actual del producto |
|  | Reserva | Enum(Si,No) | No | Estado del producto |
|  | MaxReserva | Int | Si | Máximo numero de reservas que se pueden realizar del producto |
|  | Tipo | Enum(Venta, Servicio | No | Tipo de producto si es de venta o de servicio |
|  | Hora\_inicio | time | Si | Hora de inicio para adquirir el producto o servicio |
|  | Hora\_fin | time | Si | Hora limite para adquirir el producto o servicio |
|  | Inventario | Enum(Si,No) | Si | Para aquellos productos o servicios que son únicos y no cuentan con un inventario. |
|  | fk\_codigo\_up | Int | Si | Donde es fabricado el producto |

### Tabla punto de venta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Id\_punto\_vent | Int | Si | Identificador único del punto de venta. |
|  | Sede | Enum(Centro, Yamboro. | Si | Sede donde se encuentra ubicado el punto de venta |
|  | Direccion | Int | Si | Dirección de ubicación del punto de venta |
|  | Nombre | Varchar(30) | No | Nombre del punto de venta |
|  | Fk\_persona | Bigint | Si | Persona encargada del punto de venta |
|  | Estado | Enum(Activo, Inactivo) | No | Estado actual del punto de venta |

### Tabla sena empresa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Id\_sena | Int | Si | Identificador único de sena empresa |
|  | nombre | Varchar(45) | No | Nombre de la empresa |
|  | Num\_factura | Int | No | Número consecutivo de facturación |

### Tabla unidades productivas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Columna | Tipo de datos | Obligatorio | Descripción |
| PK | Codigo\_up | Int | Si | Código único de la unidad de protección |
|  | Nombre | Varchar(40) | Si | Nombre de la unidad productiva |
|  | Logo | Varchar(80) | No | Logo de la unidad productiva |
|  | Descripcion | Varchar(100) | No | Descripción corta de la unidad productiva. |
|  | Sede | Enum(Yamboro, Centro) | No | Sede donde se encuentra la unidad productiva |
|  | Estado | Enum(Activo, Inactivo) | No | Estado actual de la unidad productiva |
|  | Entrega\_producto | Tinyint(1) | Si | Determina el estado actual del producto o servicio, si se encuentra reservado o entregado |
|  | Fk\_persona | Bigint | Si | Persona encargada de la unidad productiva. |

1. **MANUAL DE USUARIO SCANNER.**
2. **Encendido**

****Presione el botón que se encuentra en el escáner. Este emitirá un pitido y la luz indicadora se encenderá**.**

1. **Apagado**

**Opción 1:** El escáner se apagará automáticamente cuando deje de funcionar durante un periodo de tiempo y la luz indicadora se apagará.

**Opción 2:** Escanea el siguiente código:

**SHUTDOWN**

1. **Carga**

Conecte el cable de carga a la parte inferior del escáner y cargue a través de la interfaz USB de la computadora o del adaptador del cargador. El tiempo de carga será de 2 a 3 horas.

1. **Configuración conectar por cable USB**

Conectamos el cable al equipo ya sea pc, teléfono móvil u otro.

1. Para configurar correctamente el escáner, una vez este encendido, se procede a resetear las configuraciones de este.



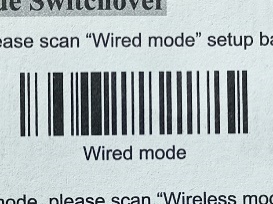
**Reset Factory default.**

1. Se escanea el siguiente código para darle un valor por defecto.

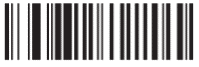


**On (default)**

1. Establecemos el modo de conexión en este caso **USB.**

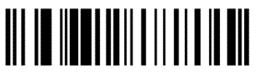
****

1. **Modo de lectura.**
2. Modo por defecto.

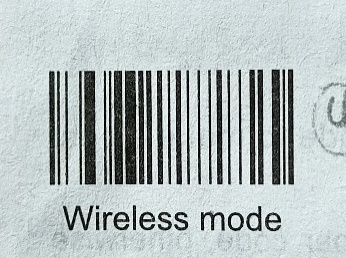


**Default mode**

1. Escaneo continuo.

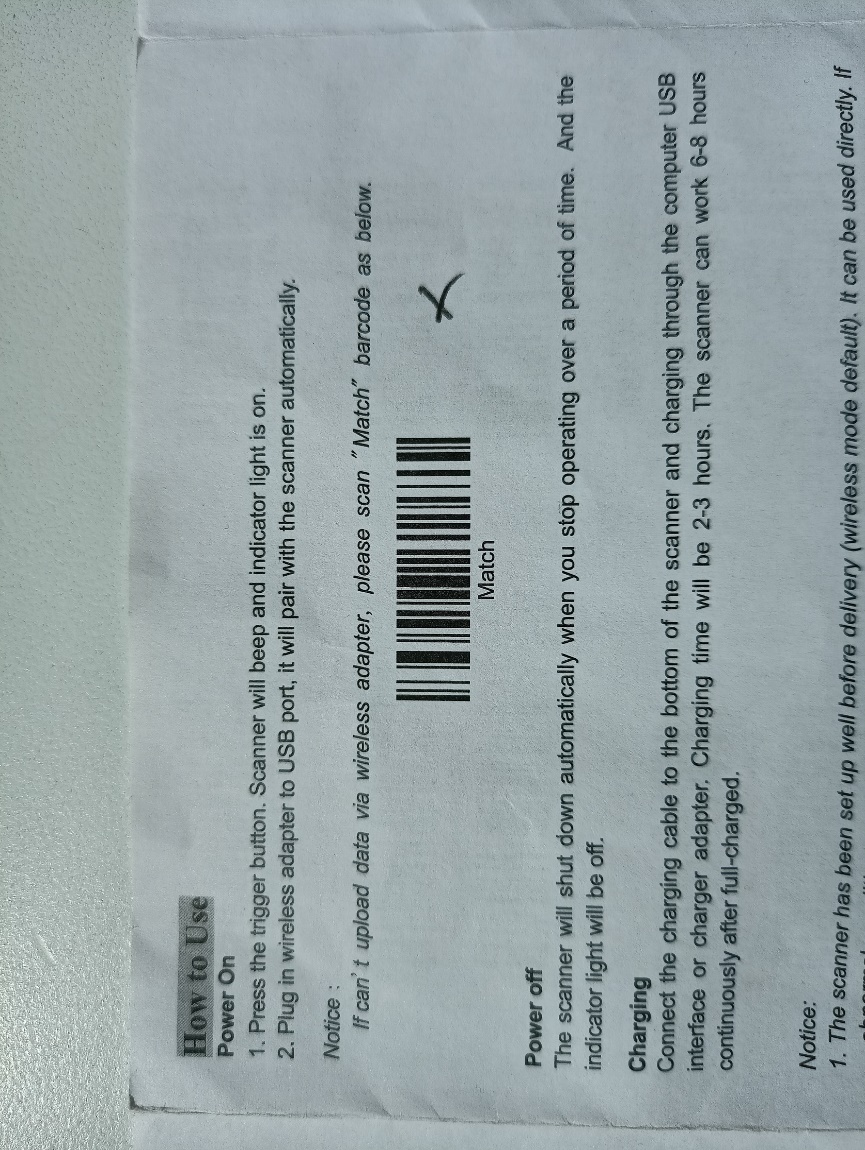
 **Continuos Scan**

1. **Configuración conectar modo inalámbrico.**
2. Volvemos al apartado **IV** y ejecutamos los pasos a y b.
3. Escaneamos el siguiente código que nos activara el modo inalámbrico.



**Wireless mode**

1. si no puede cargar a través del adaptador inalámbrico, escanee el código de barras "Match" que se muestra a continuación.

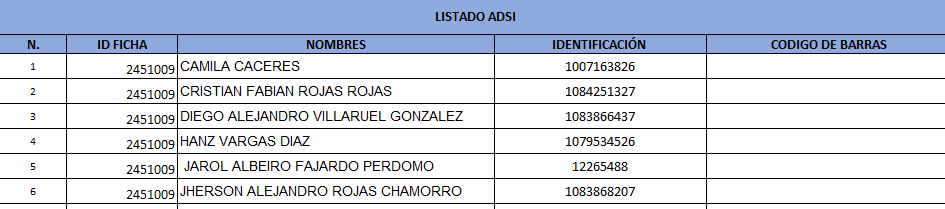


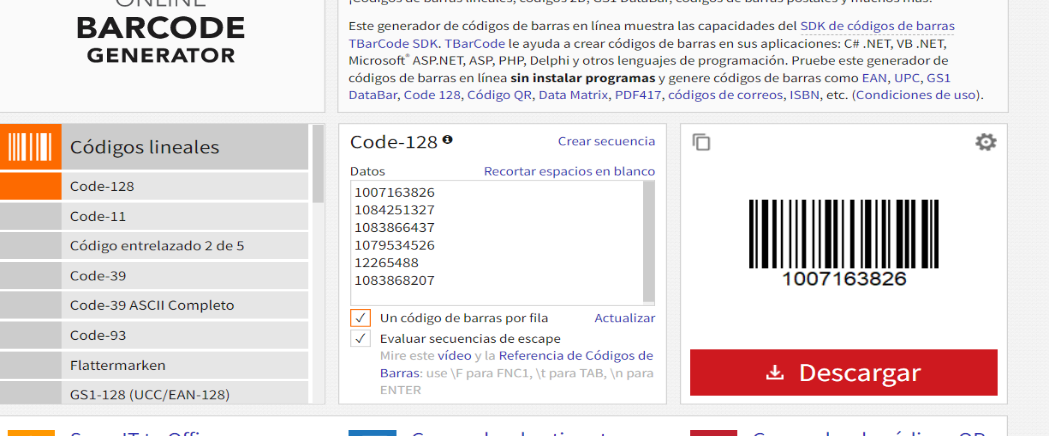
**Codigo para prueba.**

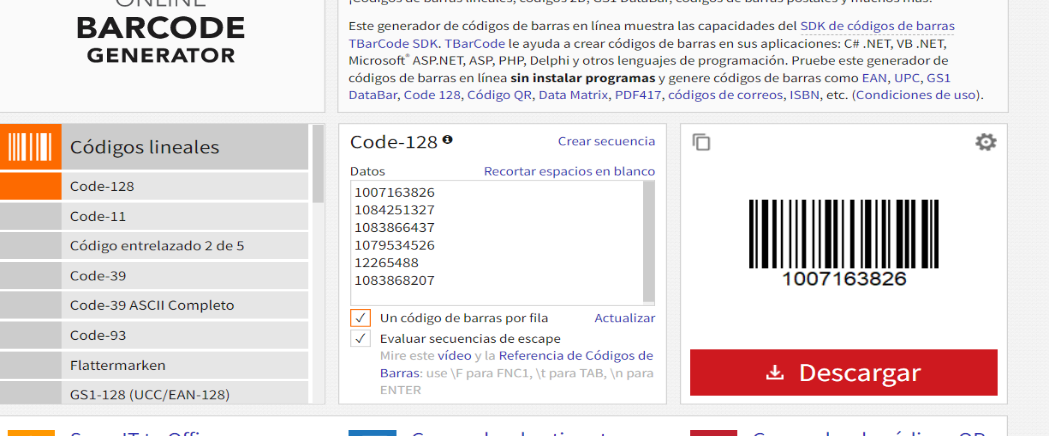


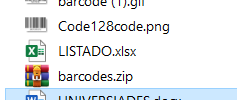
Se puede dirigir a la video tutorial “configuración de scanner” donde encontrara el paso a paso de para el funcionamiento del scanner.

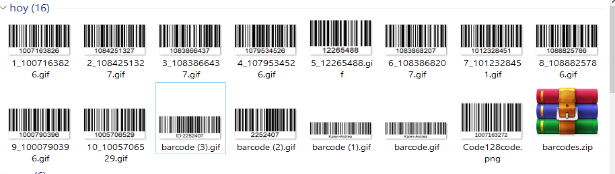
**Crear Códigos De Barra.**

1. Se solicita a los aprendices que registren la información referente al id ficha, nombre completo y identificación en un Excel organizado con el fin de generar la el código de barras.
2. Se ingresa al navegador a la herramienta que se encuentra en el siguiente link <https://barcode.tec-it.com/es/Code128> la cual permite generar los códigos en el formato code128.



1. Copiar la información que se desea convertir a código de barras.
2. Una vez convertido se procede a descargar el archivo.



1. Se descomprime el archivo.
2. Se ubica el código de barras en el espacio de Excel.

****

**Nota:**  El sistema es sensible a mayúsculas y minúsculas.

1. **BIBLIOGRAFÍA**

<https://es.wikipedia.org/wiki/Node.js>

<https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript>

<https://seopromarketing.online/que-es-mvc-ventajas/>

<https://raiolanetworks.es/blog/bootstrap/>

<https://es.wikipedia.org/wiki/JQuery>

<https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>

<https://ejs.co/>