

Contenido

[Instalación Node Js 2](#_Toc484455549)

[App 1. Hello world avanzado 8](#_Toc484455550)

[1. Requerimientos 8](#_Toc484455551)

[2. Creación del servidor 9](#_Toc484455552)

[3. Instalación de dependencias 10](#_Toc484455553)

[4. Creación del index 11](#_Toc484455554)

[5. Creación del archivo index.js 12](#_Toc484455555)

[6. Modificación en el servidor 13](#_Toc484455556)

[7. Estructura de directorios 14](#_Toc484455557)

[8. Ejecución 15](#_Toc484455558)

[Creación de tunel 15](#_Toc484455559)

[Tips informativos 16](#_Toc484455560)

# 

# Instalación Node Js

Se requiere de la instalación del framework Node Js para programar aplicaciones web basadas en el lenguaje javascript. A continuación se presentan los pasos para instalar el framework en un dispositivo Windows 10, junto con las instrucciones para usarlo en el entorno de desarrollo integrado (*IDE*) Netbeans 8.1.

Buscar en Google *Node Js* e ingresar en la pestaña *Download* de la página [*https://nodejs.org/es/*](https://nodejs.org/es/) posteriormente elegir y descargar el framework adecuado para el sistema operativo del equipo donde se va a instalar. Para este caso Windows 10 (Véase *Fig 2. Descargar para sistema operativo seleccionado).*

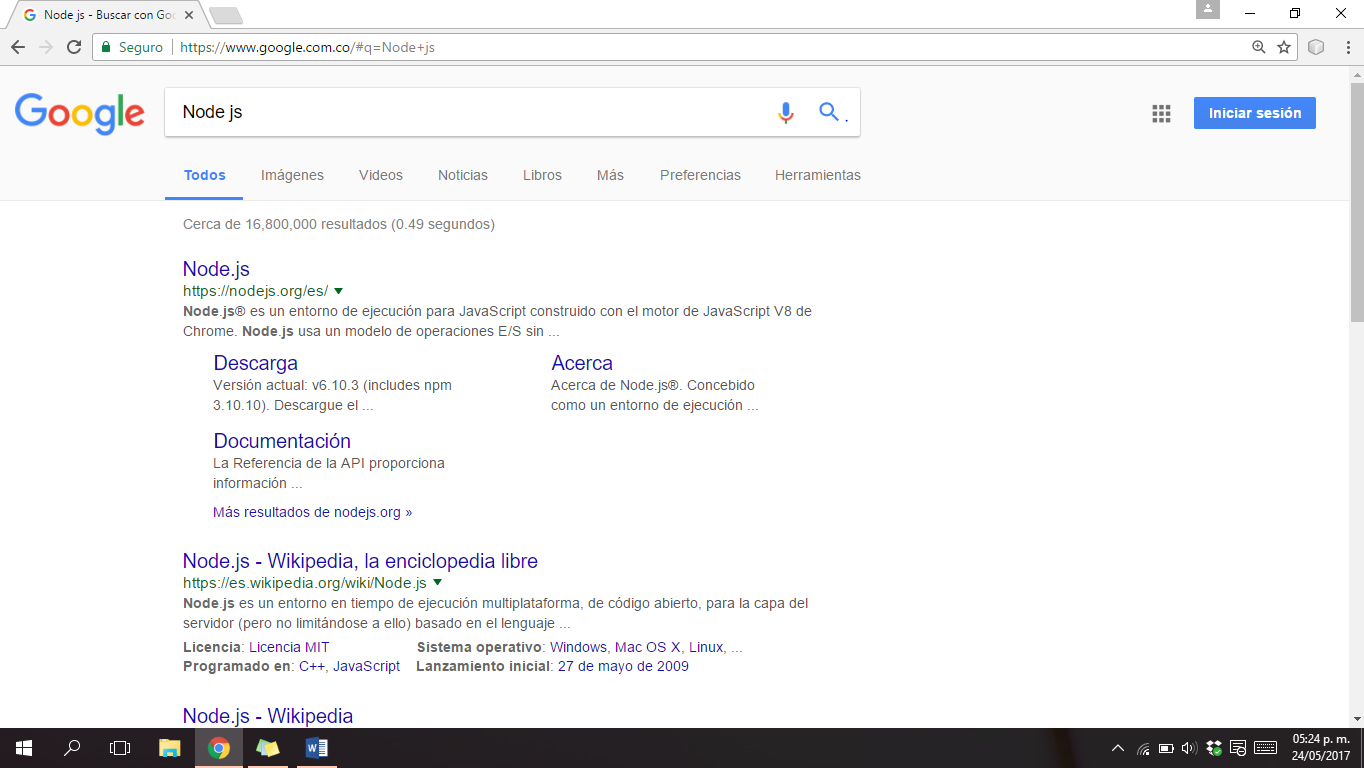


Fig . Google búsqueda

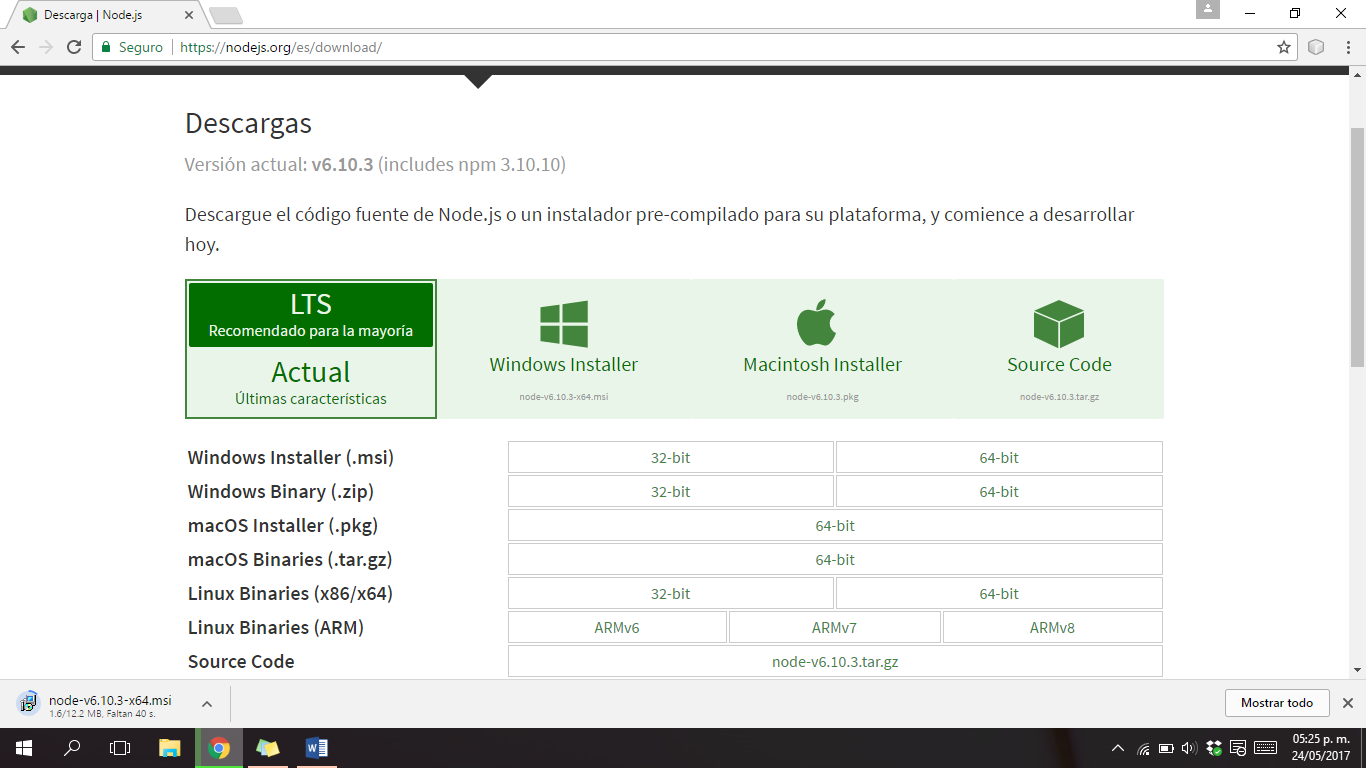


Fig . Descargar para sistema operativo seleccionado

Posterior a descargar el instalador de Node Js se ejecuta en el equipo. Para verificar que se ha realizado la instalación correcta en el equipo se ingresa a la consola de Windows ( + r, ingresar la palabra *cmd* y dar clic en ejecutar), se digita la palabra *node* y debe aparecer una flecha (>) tal como se muestra en la *Fig 4. Verificar instalación.*

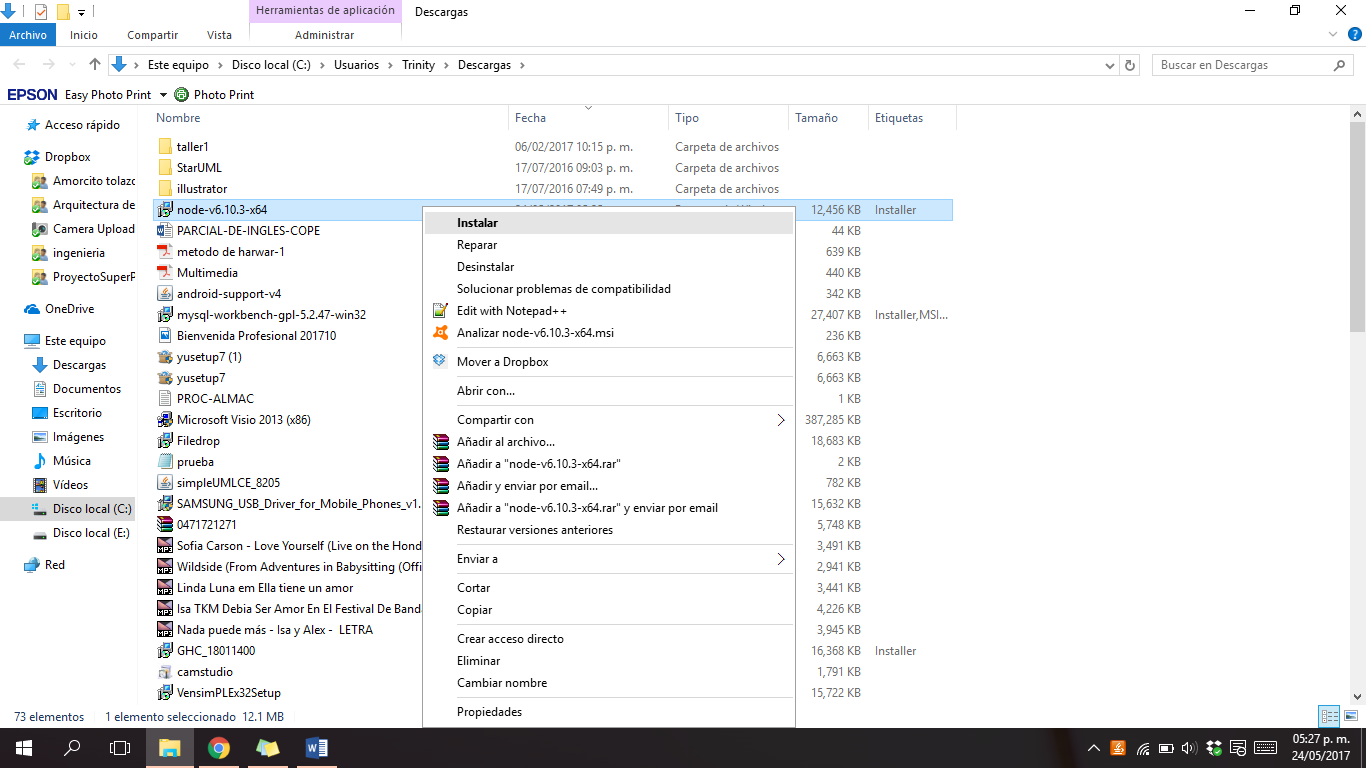


Fig 3. Ejecutar en el equipo

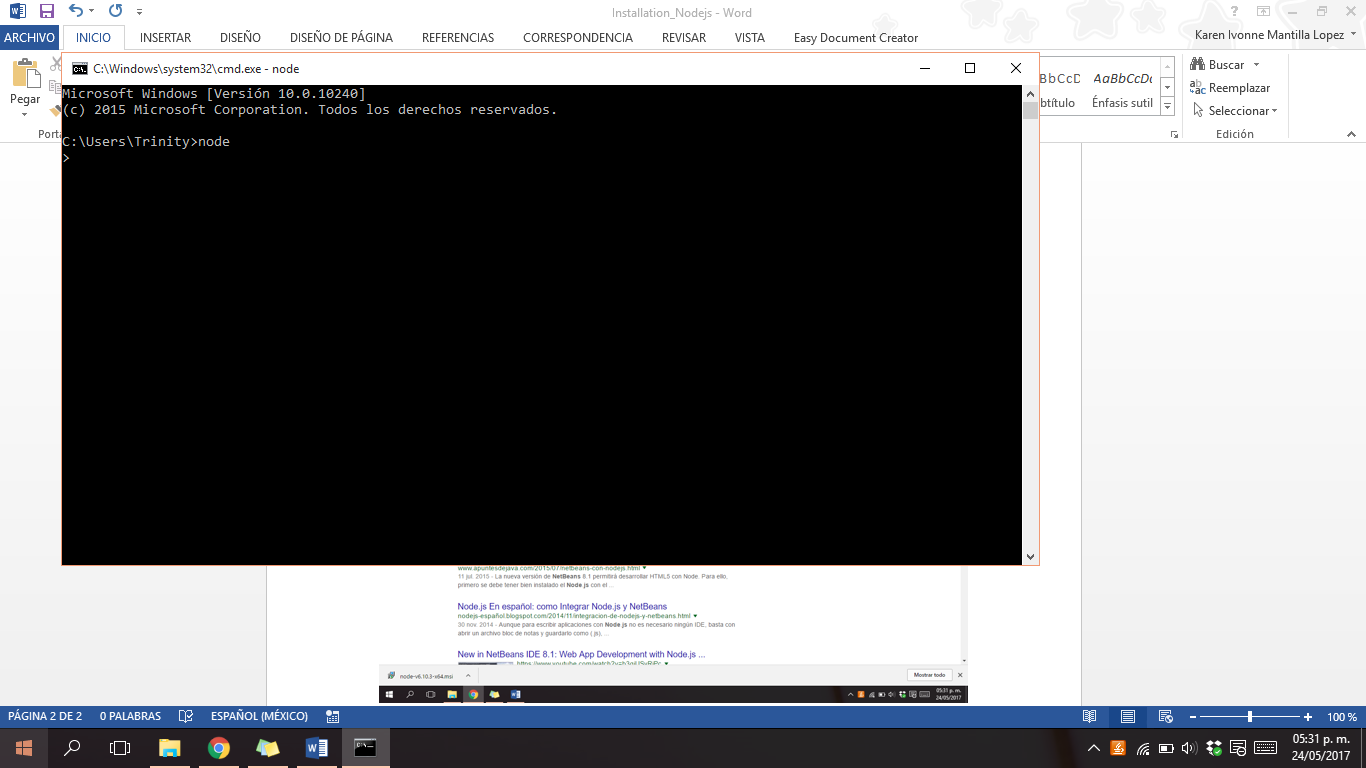


Fig 4. Verificar instalación

Para desarrolla aplicaciones en Node usando el IDE Netbeans 8.1 se requiere descargar un plugin, para ello se digita en Google *node js netbeans* y se ingresa al primer enlace (Véase *Fig 5. Plugin).*

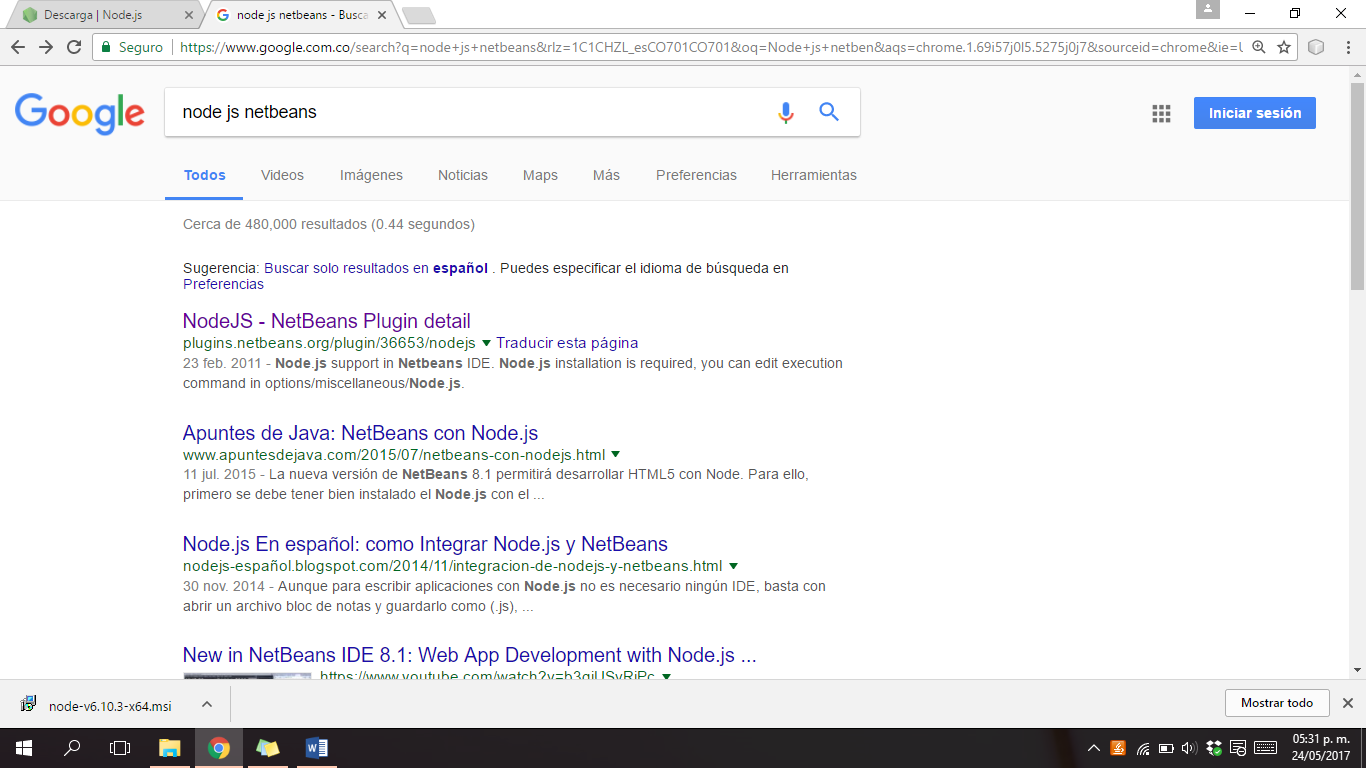


Fig 5. Plugin

Se descarga el plugin según la versión del IDE, en este caso el plugin disponible es para Netbeans 8.0, sin embargo se puede hacer uso de éste.

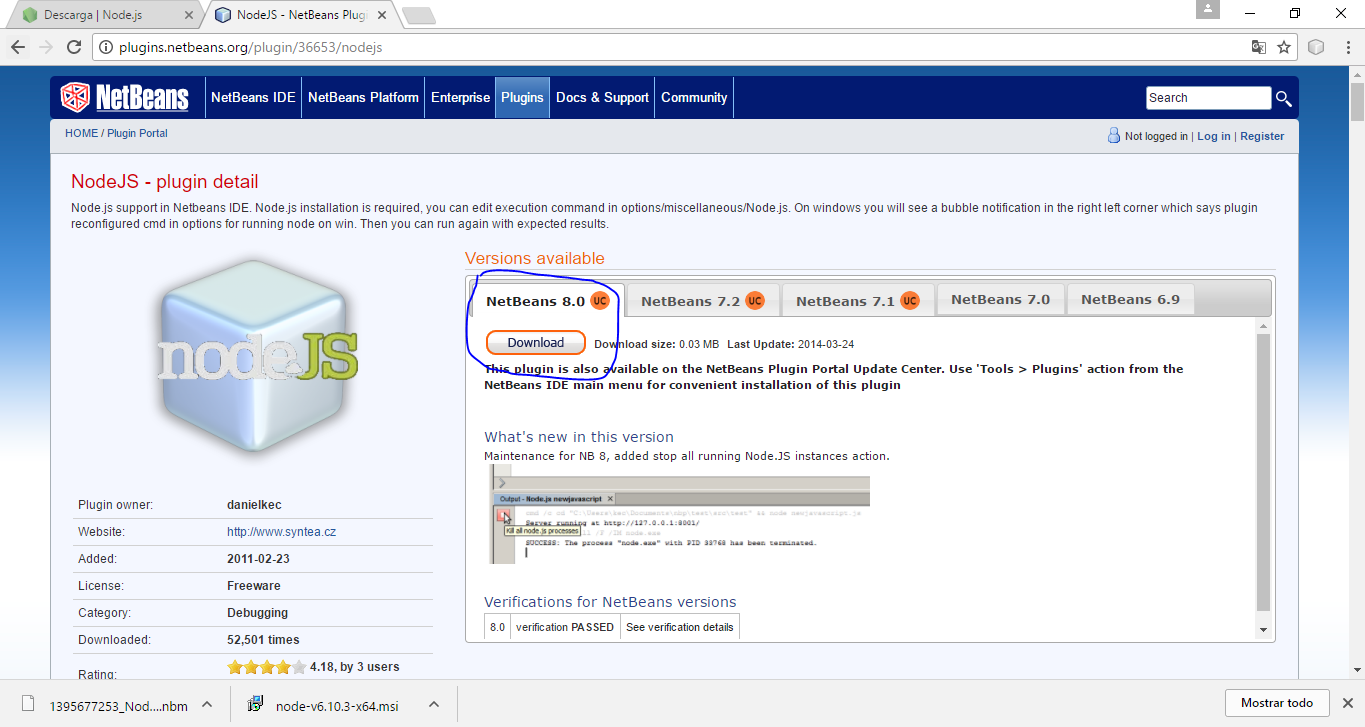


Fig . Descarga plugin

Se abre Netbean 8.1 y en la pestaña *Tools* se da clic en la opción *Plugins*.

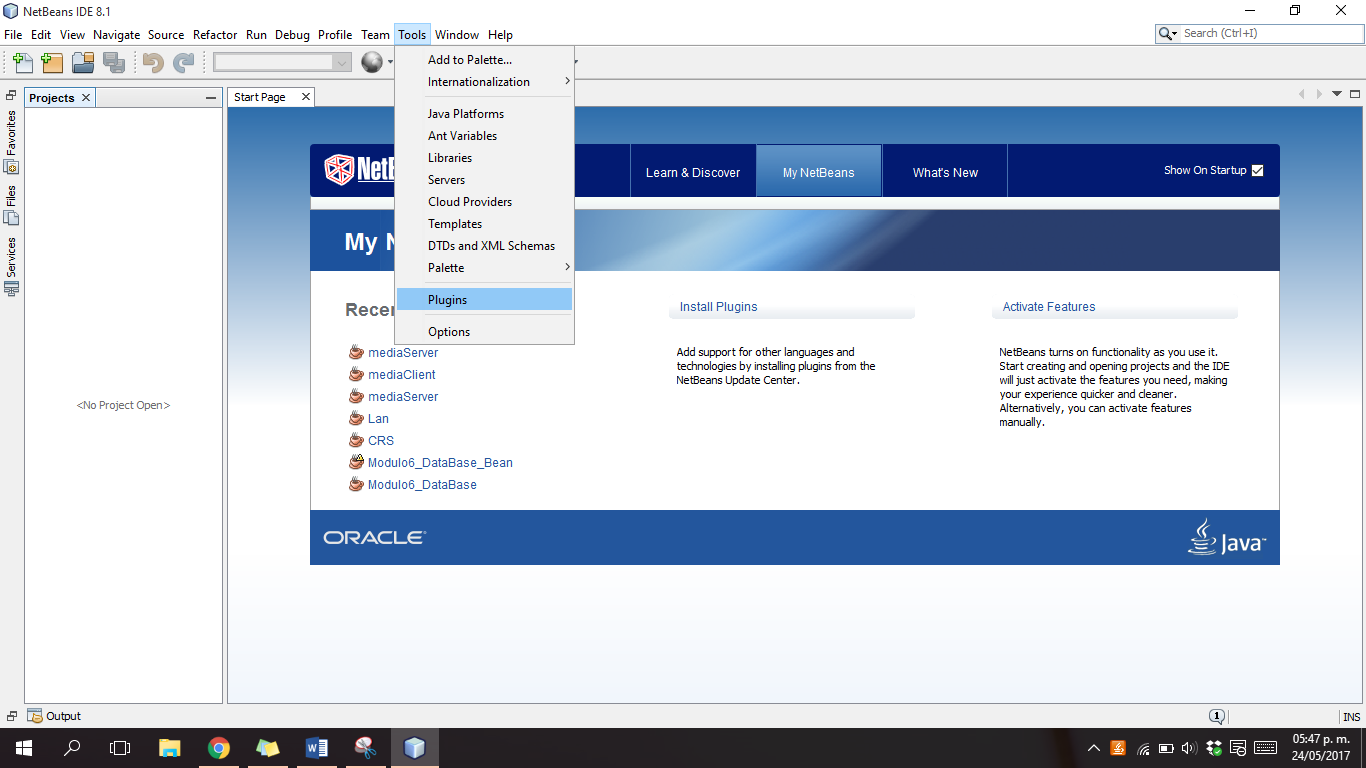


Fig . Panel para añadir plugin

Al dar clic en *Plugins* aparece una ventana (Véase *Fig 8. Añadir plugin)* en la que se puede visualizar la pestaña *Downloaded*, se da clic en el botón *Add plugins* y se procede a buscar la ubicación en donde se encuentre el plugin que se descargó anteriormente.

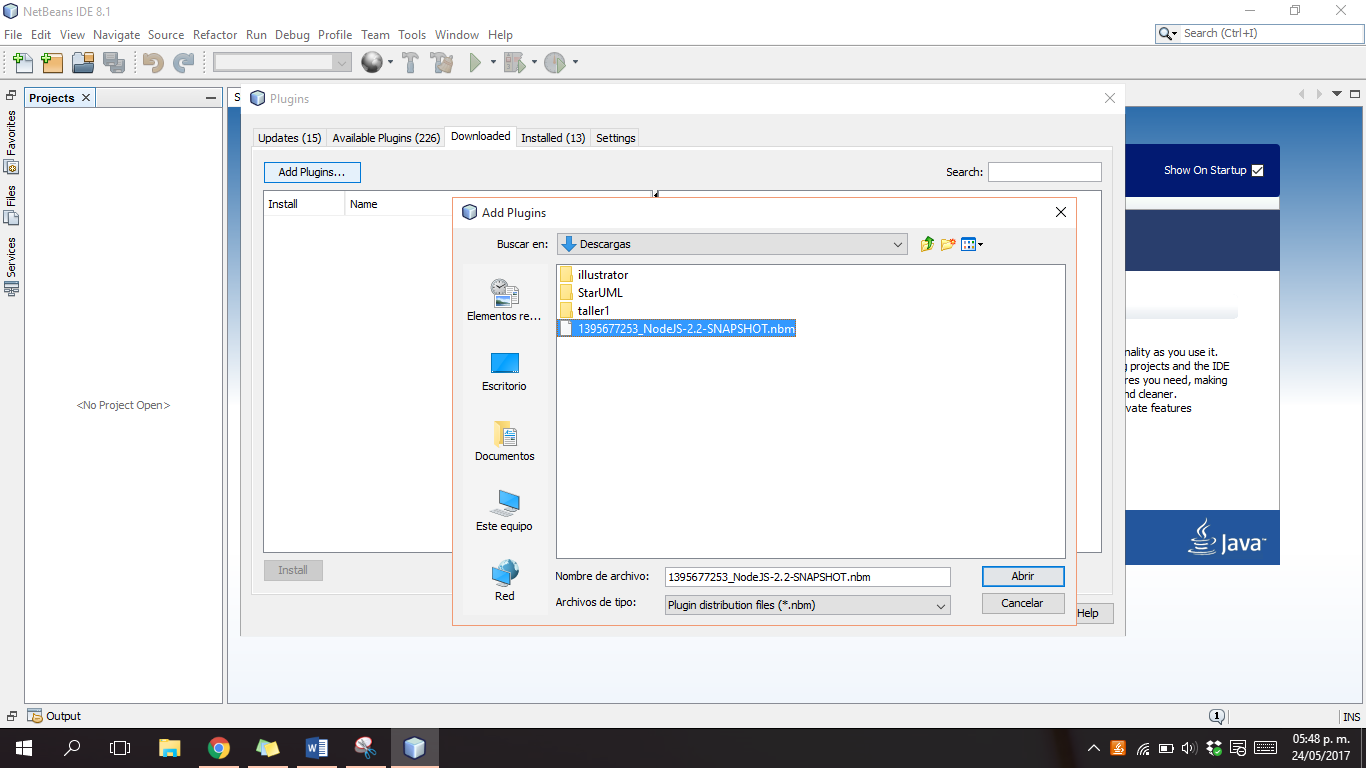


Fig . Añadir plugin

En caso de que al crear un nuevo proyecto se genere el error *Node Js: Missing node.js sources (in TestApp1)* (Véase *Fig 9. Error),* se da clic en la pestaña *Tools* posteriormente en *Options* para ingresar a la pestaña *HTML/JS* que aparece en la ventana (Véase *Fig 10. Ventana Options)*. En dicha pestaña se debe abrir el enlace *Install Generator,* el cual dirige al siguiente link: <http://expressjs.com/en/starter/generator.html>

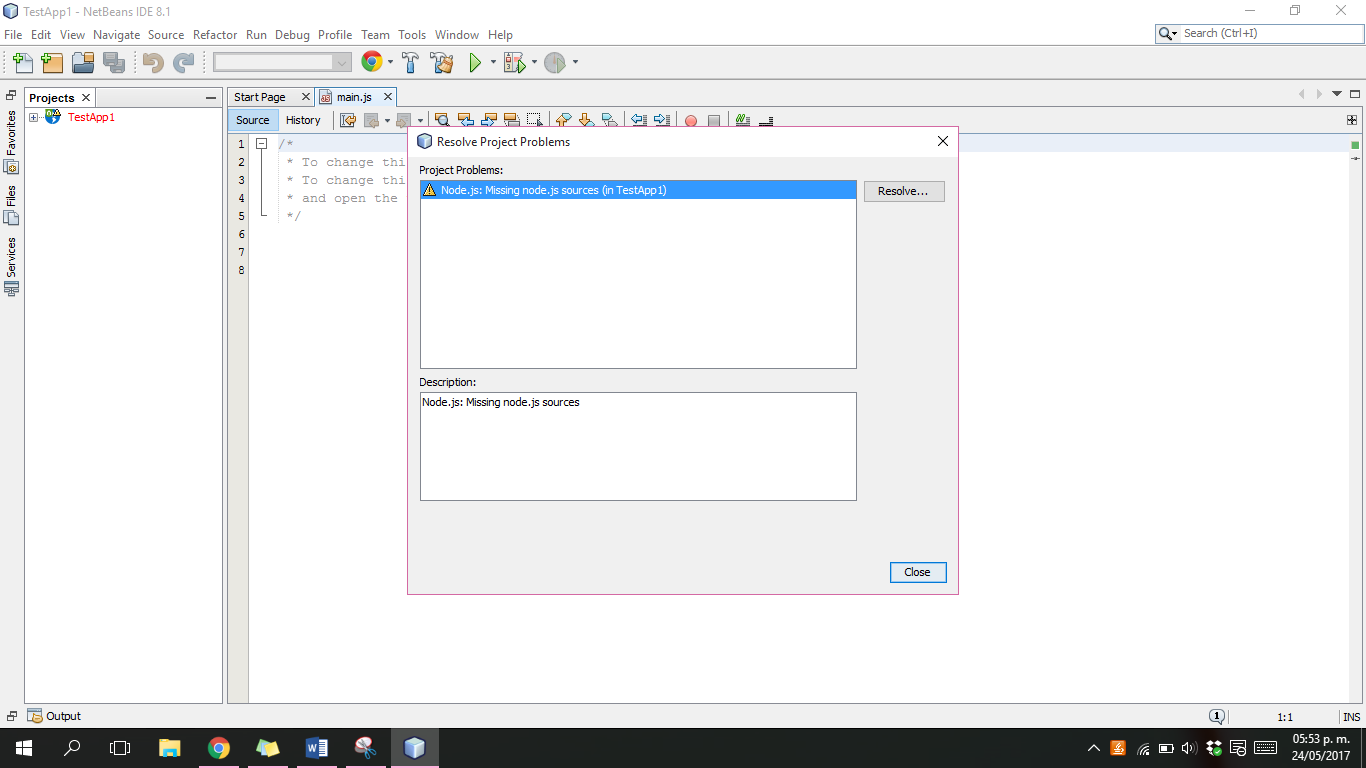


Fig . Error

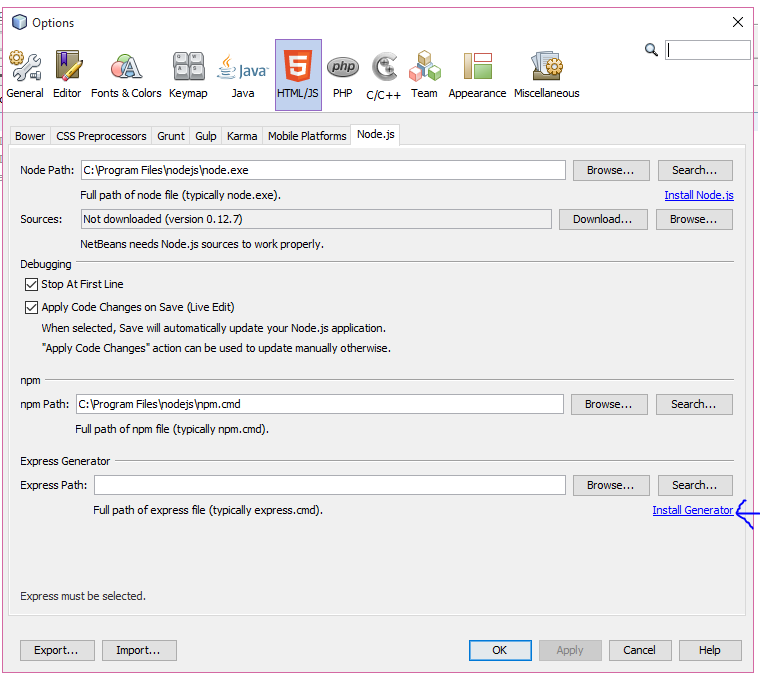


Fig . Ventana Options

A continuación se presentaran los pasos del enlace anteriormente mencionado (<http://expressjs.com/en/starter/generator.html>), sin embargo, lo más pertinente es guiarse por el enlace.

Se inicia ejecutando la consola de comandos de Windows ( + r, ingresar la palabra *cmd* y dar clic en ejecutar) para digitar el comando *npm install express-generator –g,* el cual se compila dando clic en la tecla enter (Véase *Fig 11. Primer comando*).

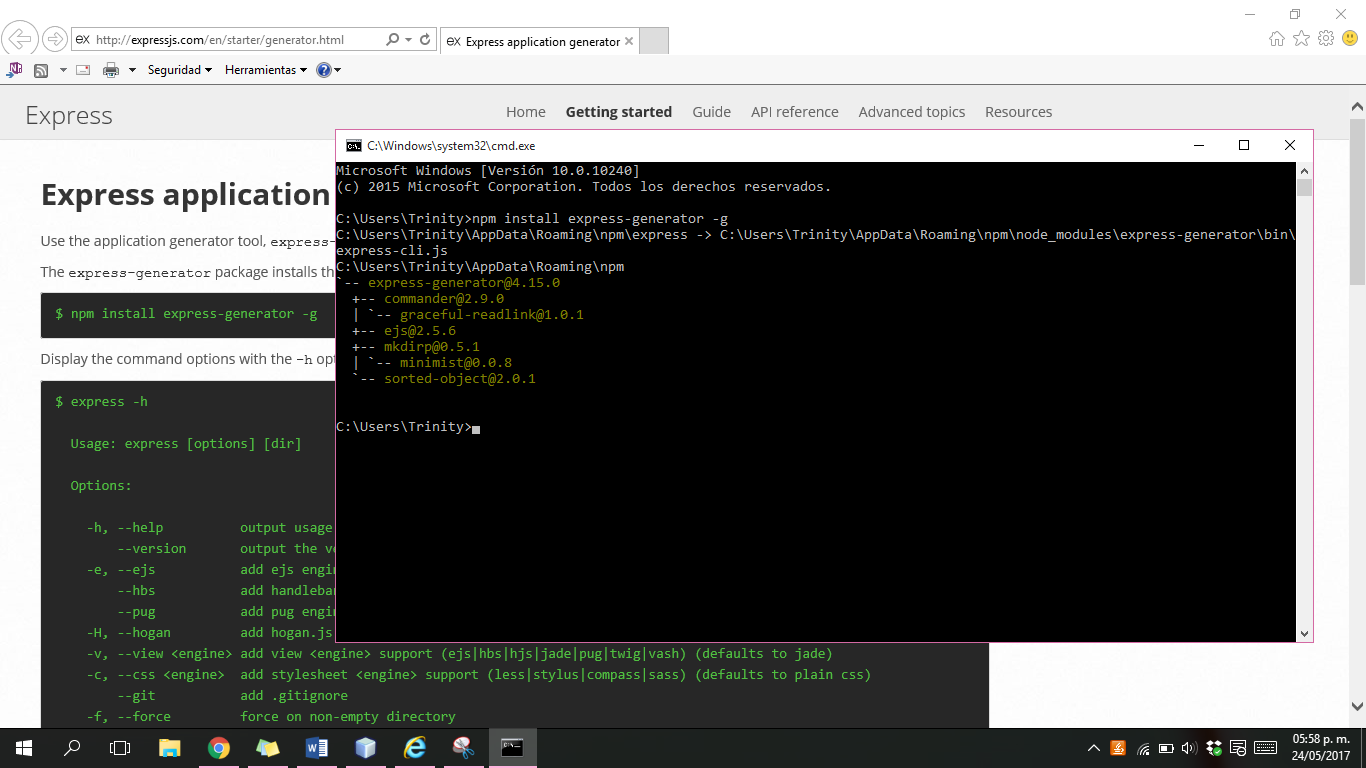


Fig . Primer comando

Posteriormente se compila el comando *express –view=pug myapp*, tal como se muestra en la siguiente imagen.

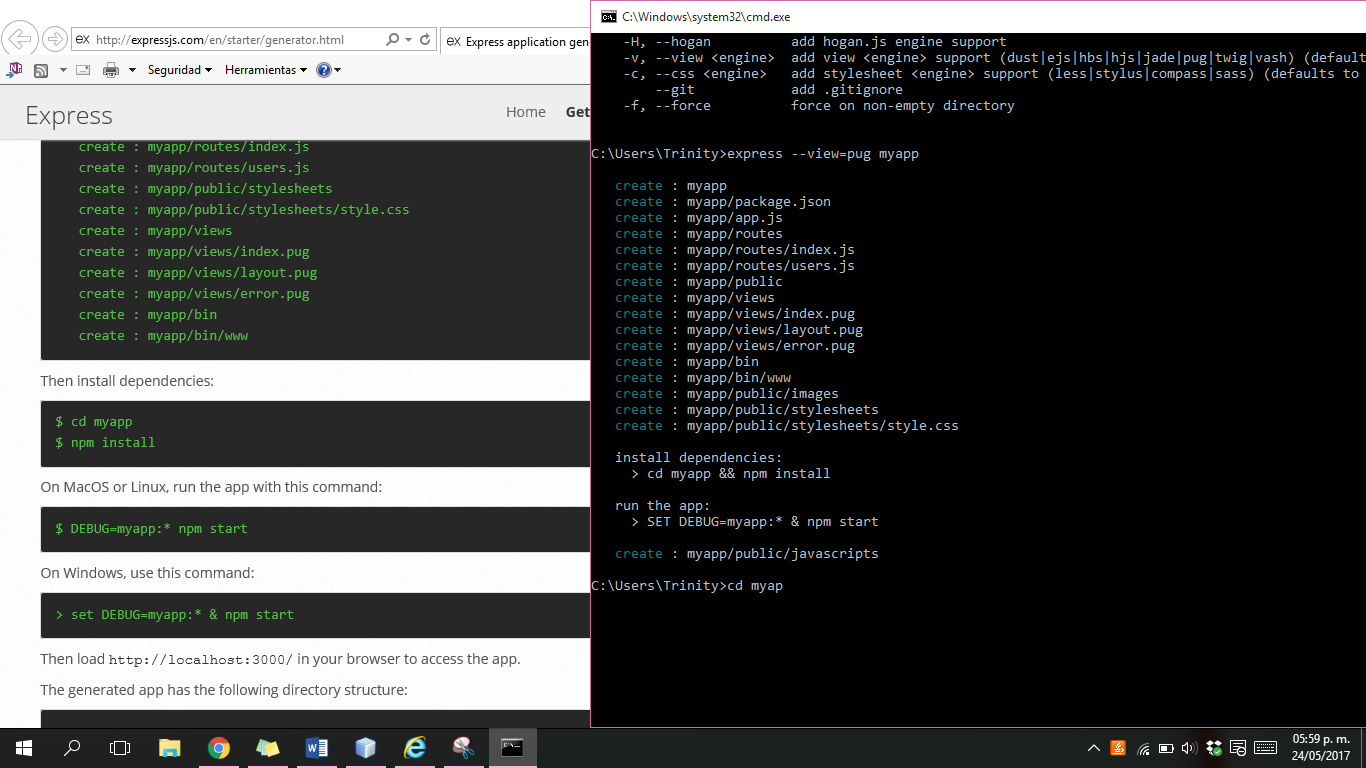


Fig . Segundo comando

Finalmente se digita el comando *cd myapp,* y al ingresar al proyecto en Netbeans en la ventana (Véase *Fig 10. Ventana Options)* aparece una ruta en el campo *Express Path*, se da clic en el botón *OK* y se puede iniciar a programar.

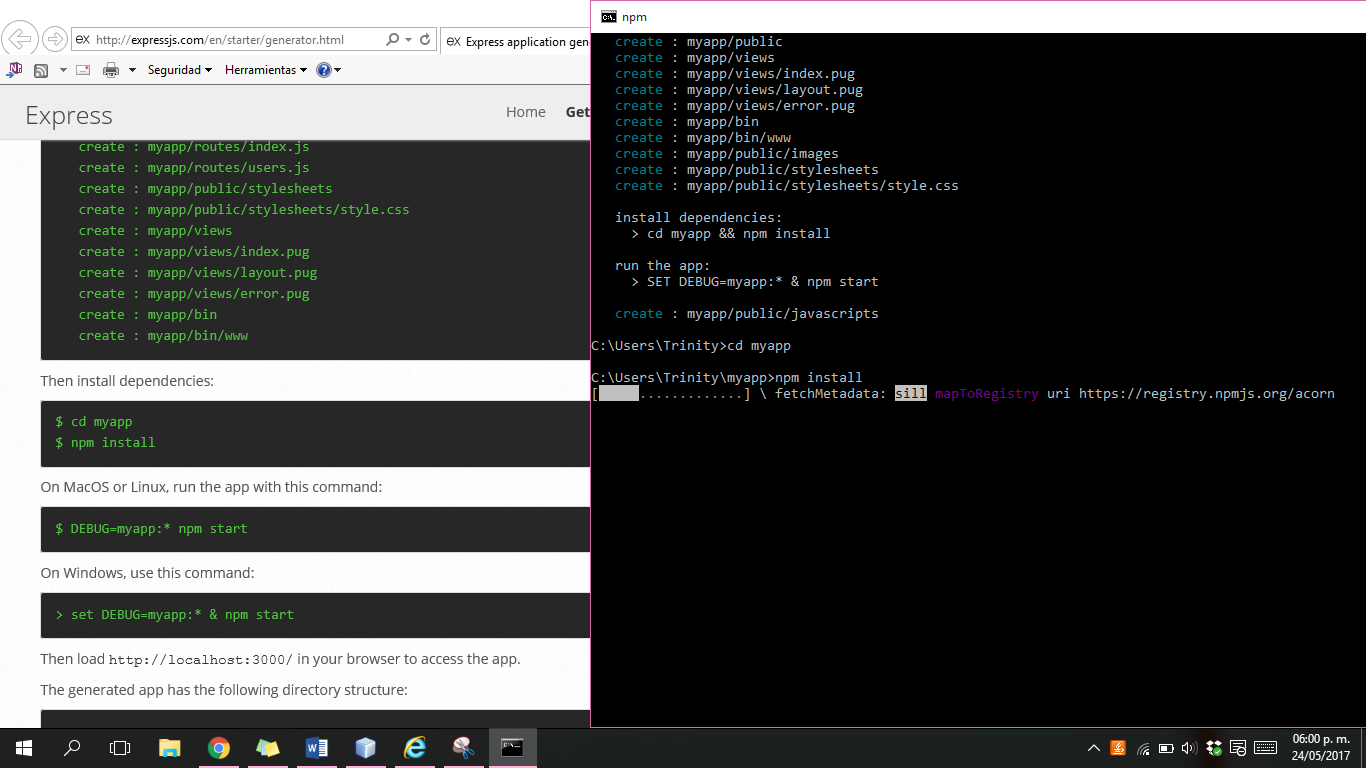


Fig . Tercer comando

# App 1. Hello world avanzado

Más adelante se van a presentar los pasos a seguir para elaborar una primera aplicación en Node Js, que va cumplir con unos requerimientos determinados que permitirán iniciar con el aprendizaje de este framework, dando bases para posteriormente desarrollar aplicaciones web complejas.

## Requerimientos

El app debe contar con un index principal que use *Bootstrap* y *javascript*, además que contenga:

* Un botón que capture y envíe lo escrito por el usuario en una caja de texto desde el frontend e imprimirlo en la consola de Netbeans usando *Ajax*.
* Un botón que solicite al servidor el valor de una variable y aumente su valor 1 luego de retornar el valor actual y mostrarlo en un label en el frontend, *usando Ajax*.

Inicialmente se crea un nuevo proyecto en Neatbeans, dando clic en *File* y luego en la opción *New Project.*

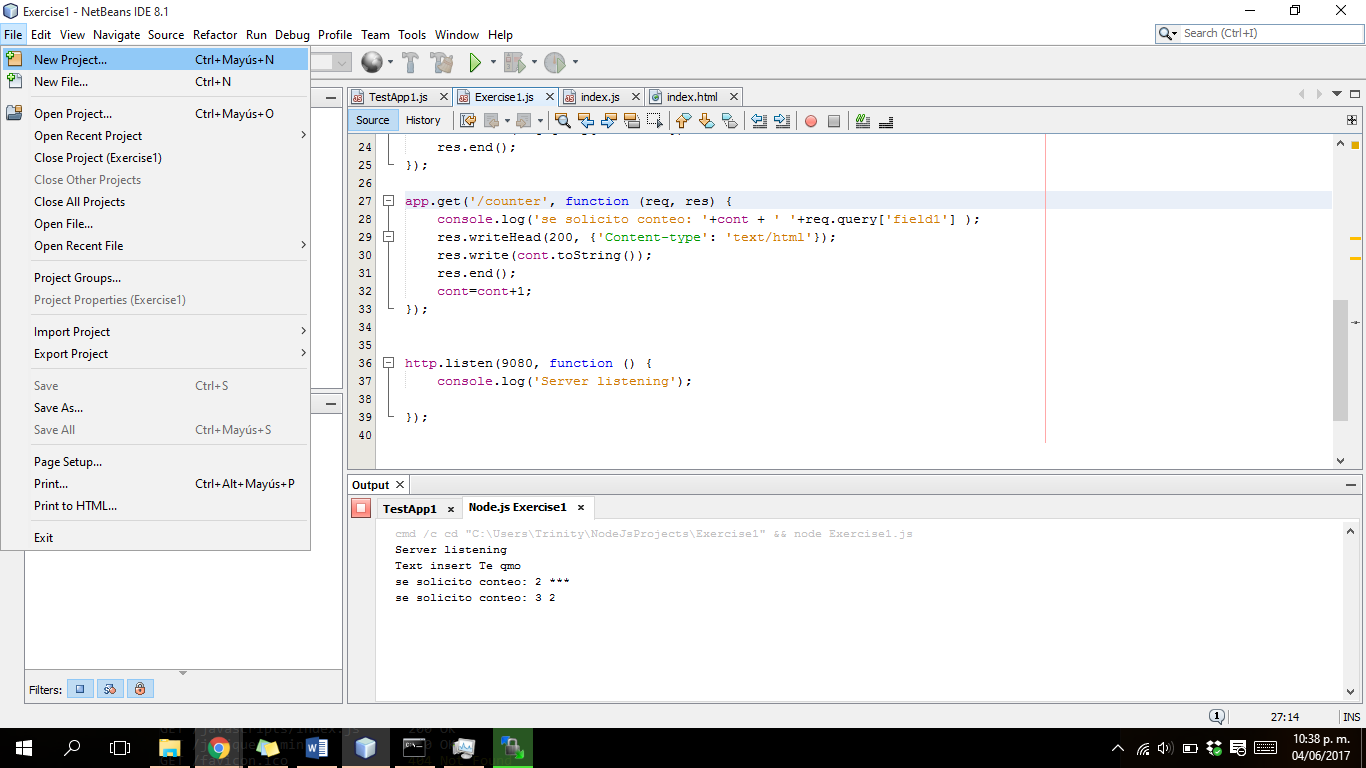


Fig . Crear proyecto paso 1

A continuación se visualiza la siguiente ventana (Véase *Fig 15. Crear proyecto paso 2*) donde se escoge el lenguaje en el que se va a desarrollar el proyecto, en este caso se selecciona en NodeJS Projects y damos clic en *Next*.

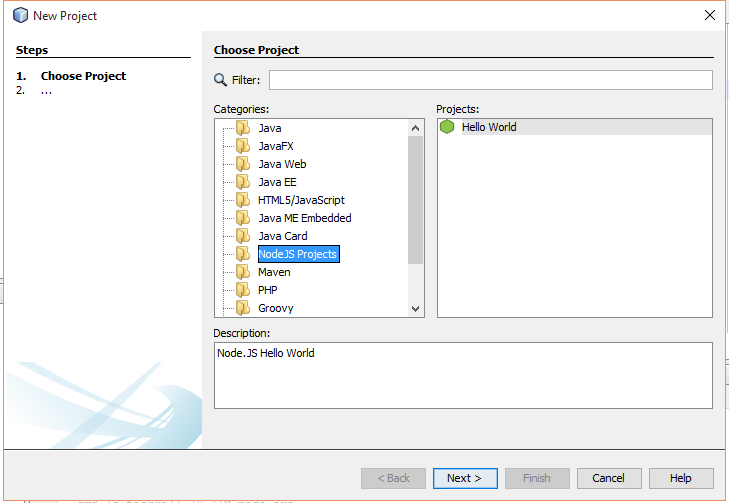


Fig . Crear proyecto paso 2

## Creación del servidor

Al crear un nuevo proyecto se crea un archivo javascript que incluye un código por omisión para crear un servidor con un puerto especifico en escucha, a través del cual se enviaran y responderán las peticiones que hagan los usuarios a la página web. En este archivo se usará más adelante para añadir código que permita el manejo de directorios.



Fig . Archivo .js

Existen diferentes formas de crear el servidor en el archivo .js, a continuación se presentaran y explicaran cada uno de los fragmentos de código. Para el desarrollo del ejercicio se usó *Fig 19. Forma 3.*

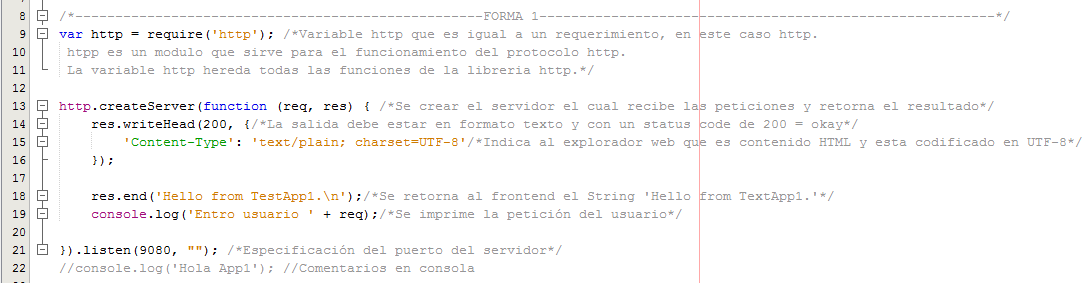


Fig . Forma 1

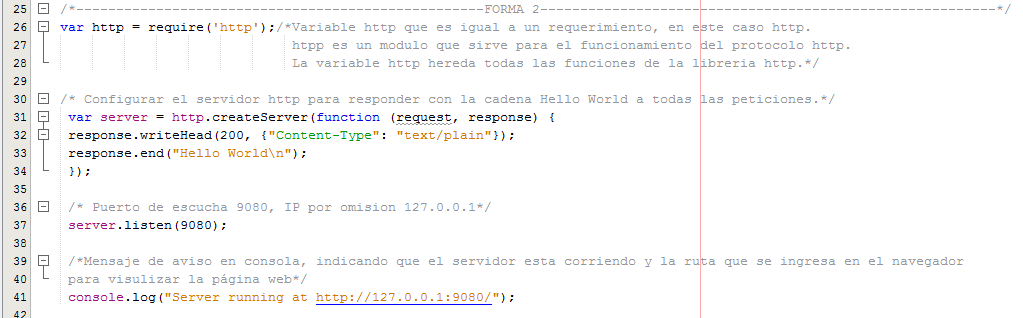


Fig . Forma 2

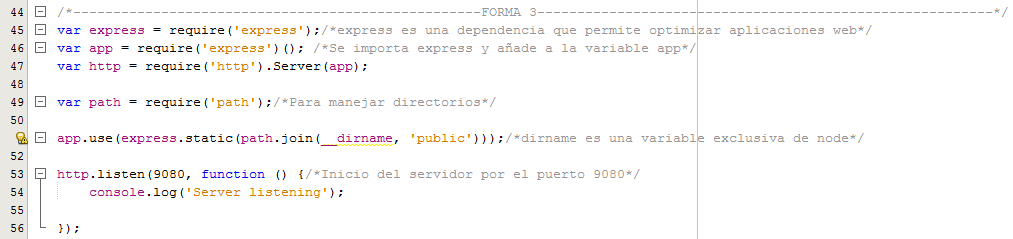


Fig . Forma 3

## Instalación de dependencias

Para que la creación del servidor funcione, se necesita instalar las dependencias en el proyecto según la forma (Véase *Fig 17, Fig 18, Fig 19)* que se haya elegido, si es la forma 1 o la forma 2 se necesita instalar la dependencia *http,* si es la forma 3 se requieren 3 dependencias: *path, express, http.*

En la siguiente imagen se presenta como debe realizarse la instalación de las dependencias, iniciando con la ejecución de la consola de comandos de Windows ( + r, ingresar la palabra *cmd* y dar clic en ejecutar) luego se procede a entrar en la ruta donde se encuentra el proyecto al que se le instalará las dependencias y se ejecuta el comando *npm install http –save*;en caso de ser otra dependencia se cambian en el comando http por el nombre de la que se va instalar (Ejemplo: *npm install path –save*).

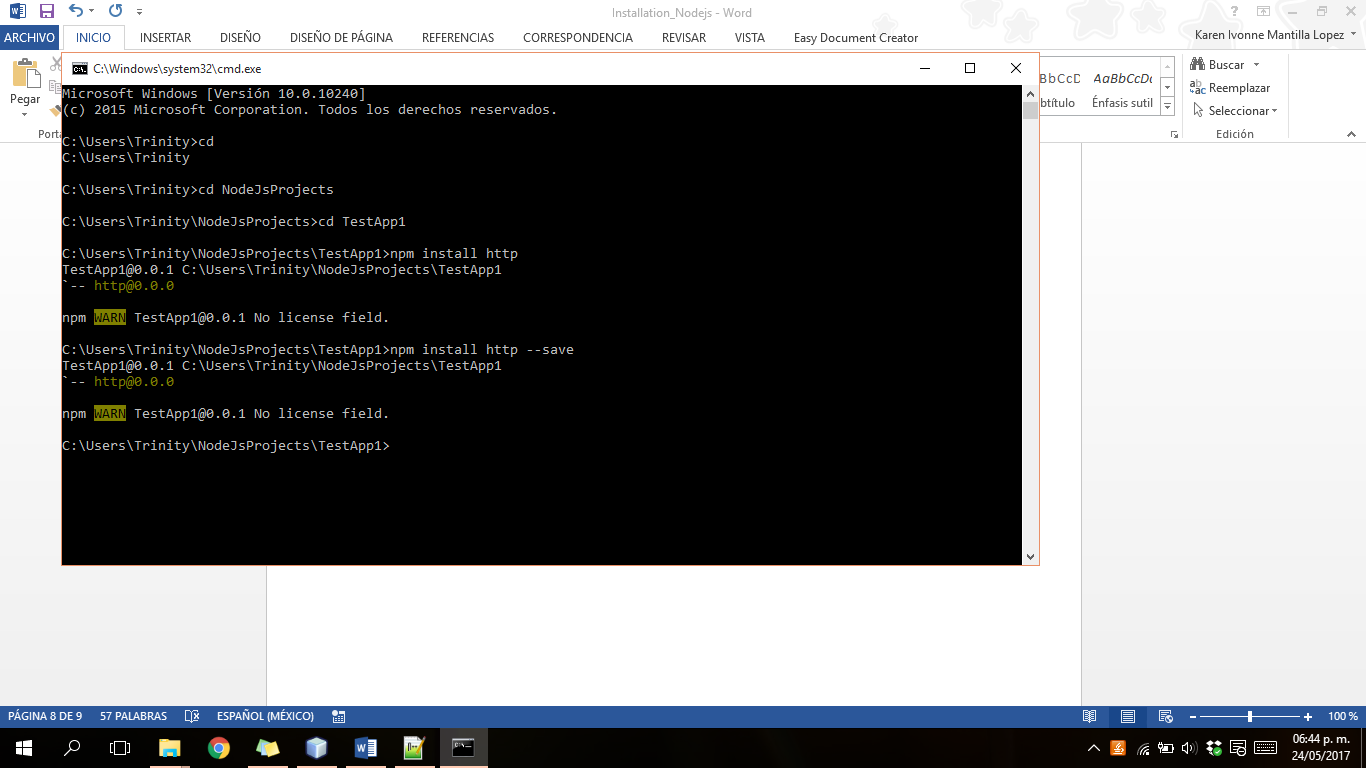


Fig . Instalación de dependencias

Para verificar la correcta instalación de las dependencias se abre el archivo package.json que se encuentra en la carpeta del proyecto; en éste debe visualizarse un json llamado *dependencies* que contenga el nombre de todas las que se instalaron. (Véase *Fig 21*)

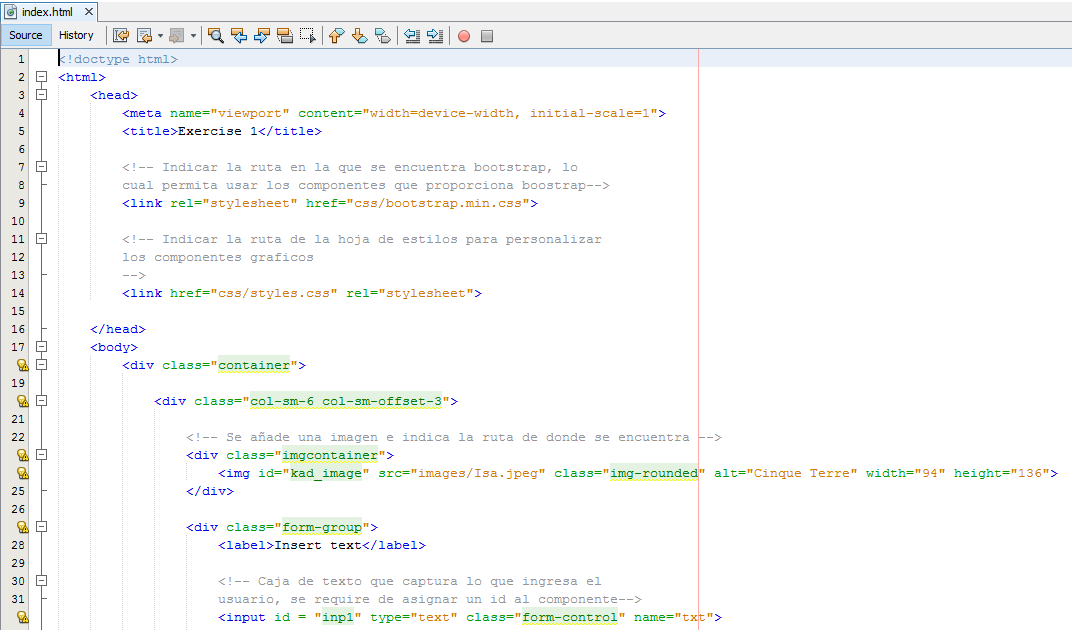


Fig . Verificar instalación

***NOTA:*** Por cada módulo que se necesite hay que instalar una dependencia

## Creación del index

Dentro de los requerimientos se estableció el diseño del frontend es decir, la parte gráfica de la aplicación; para ello se creó se requiere contar con la librería *bootstrap* y añadirla dentro de la estructura de directorios (Véase *Fig 20. Estructura de directorios*); además de la creación de un archivo html que se presentará y explicará en la siguiente imagen:



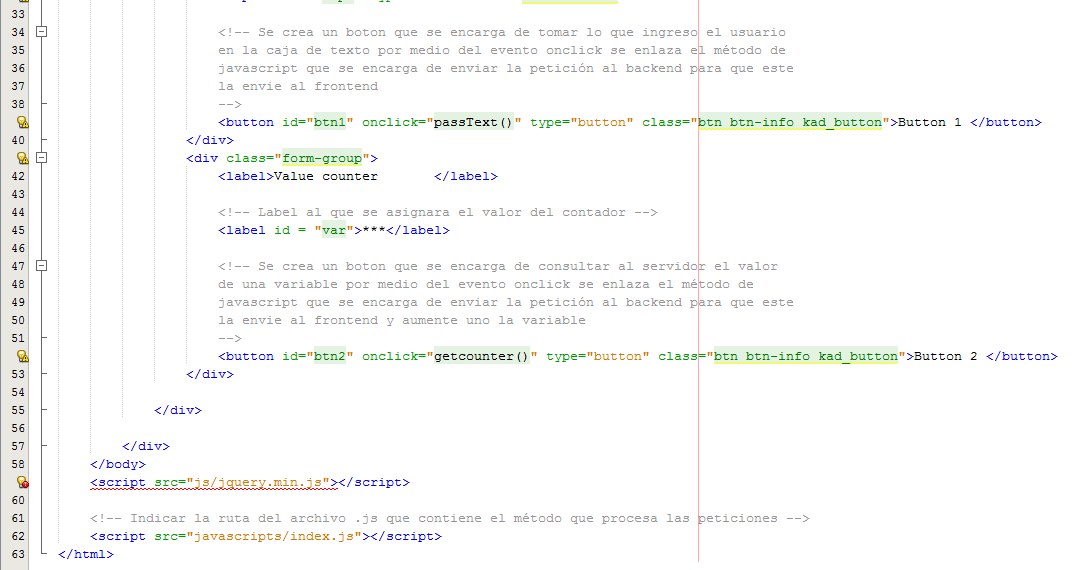


Fig . Index.html

## Creación del archivo index.js

Para procesar los parámetros enviados por el usuario se requiere de un archivo en javascript que contenga los métodos que son llamados por medio de eventos desde el frontend; estos se encargan de enviar al backend para que retorne una respuesta. A continuación se explicaran los métodos que se encuentran en index.js para atender los eventos de cada uno de los botones.

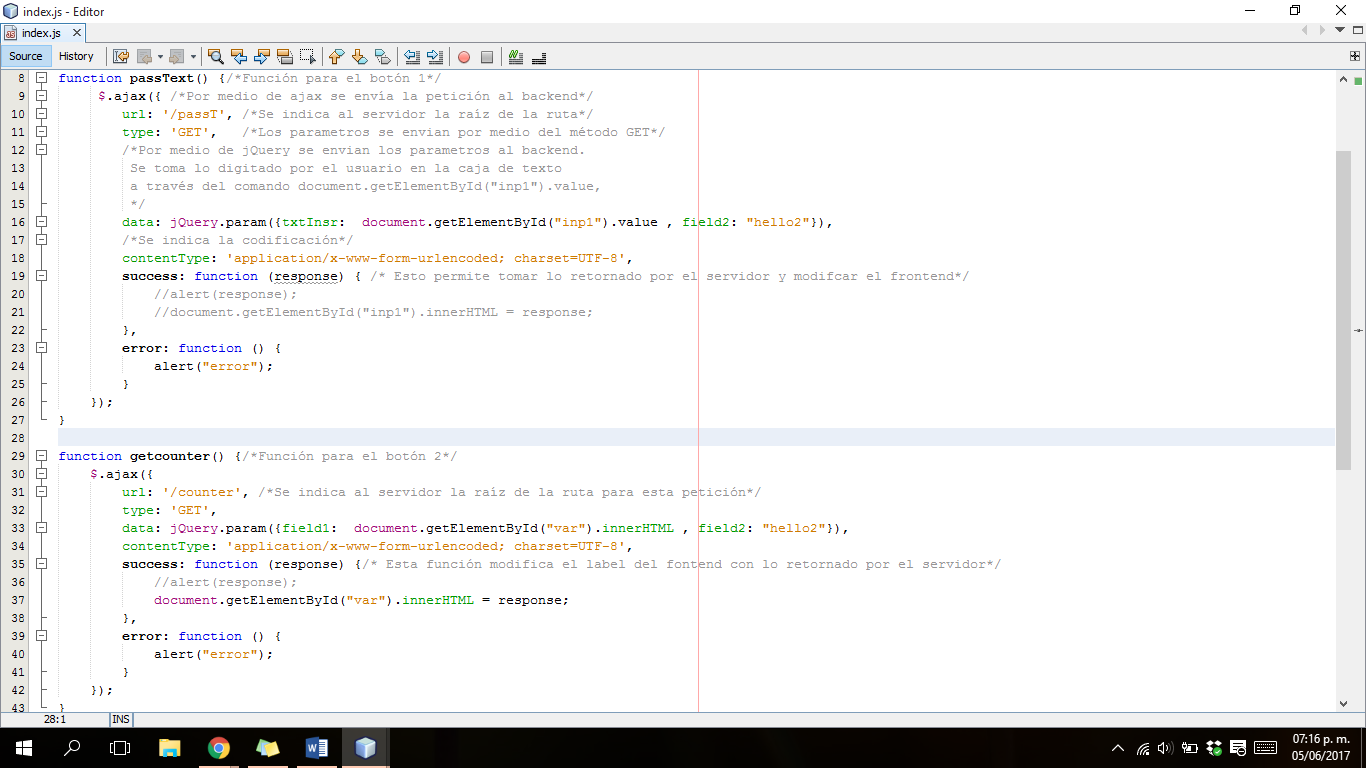


Fig . index.js

## Modificación en el servidor

El servidor es el encargado de retornar las respuestas de las peticiones de los usuarios, para ello se va a modificar el archivo creado por omisión cuando se crea la aplicación. Más adelante se va a presentar la imagen con las modificaciones hechas en el servidor y su explicación.

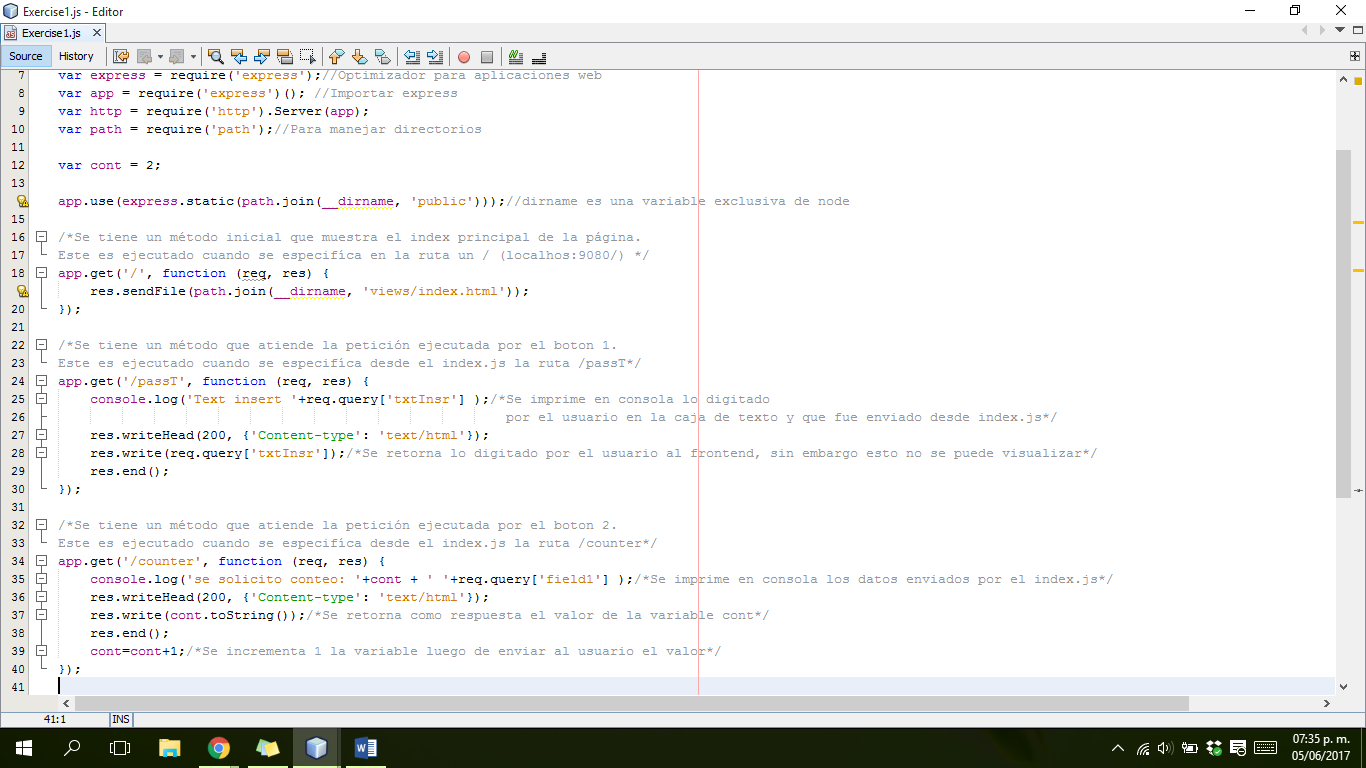


Fig . Exercise1.js

## Estructura de directorios

En la siguiente imagen se presenta la estructura de directorios del proyecto desarrollado, se encuentra el archivo .js que se crea por omisión con el mismo nombre de la aplicación, en el cual se crea el servidor con el puerto específico para responder a las solicitudes de los usuarios. Además se cuenta con la carpeta *public,* que contiene varias carpetas: *css* (en esta se encuentra la librería bootstrap)*, fonts* (se encuentran los archivos que personalizan los estilos de las letras)*, images* (se almacenan las imágenes que se usan la parte grafica de la aplicación), *javascript* (contiene el archivo index.js explicado anteriormente *Creación del archivo index.js*), *js* (archivos javascript que requieren algunos componentes del frontend)*, lib* (librerías añadidas). Por otro lado en la carpeta *views* se almacenan los archivos .html que modelan cada una de las partes de la página.

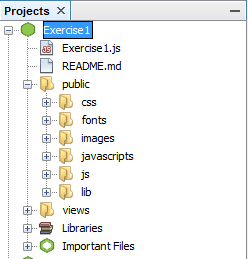
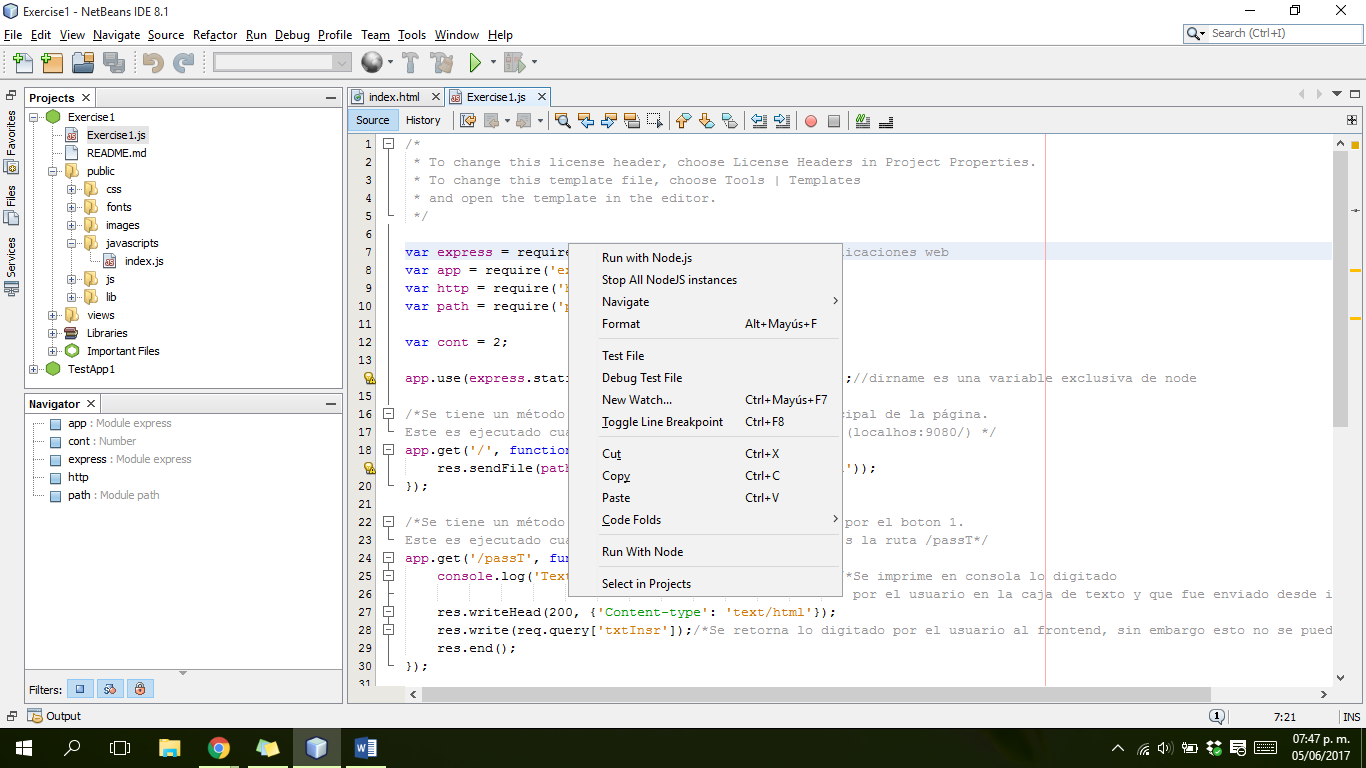


Fig . Estructura de directorios

## Ejecución

Para la ejecución de la aplicación, se ubica en el archivo creado por omisión .js con el nombre de la aplicación, hacer clic derecho e ir a la opción *Run with Node.js.* Y finalmente dirigirse al navegador e ingresar localhost:9080.



# Creación de tunel

La creación de un túnel permite a un programador tener acceso a una aplicación web desde un dispositivo diferente al que se encuentre alojada, sin requerir de un dominio de internet. Para ello se requiere instalar *ngrok* (<https://ngrok.com/> link para descargar el instalador del software)*.*

Para crear el túnel se ejecuta la consola de comandos de Windows ( + r, ingresar la palabra *cmd* y dar clic en ejecutar) y se ingresa a la ruta donde se encuentra ubicado el software, posteriormente digitar el comando *ngrok http 9080* (El valor 9080 corresponde al puerto que está usando la aplicación)*.*

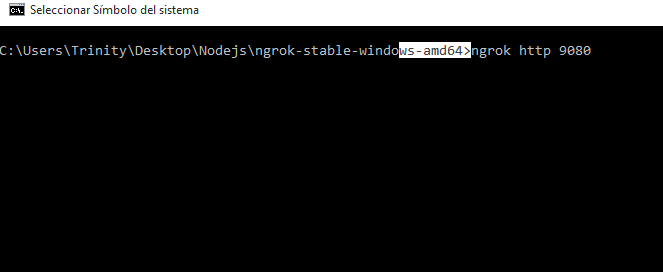
* 

Fig . Ejecutar ngrok

Tras ejecutar el comando aparece la siguiente información (Véase *Fig 14. Información ngrok*), de la que se toma la ruta que aparece en frente a *Forwarding*: <http://926b6de6e.ngrok.io>. Se ingeresa esta ruta en el navegador y se puede acceder a la página siempre y cuando esta se esté ejecutando en el dispositivo donde esta almacenada.

***NOTA:*** La ruta que proporciona ngrok cambia cada vez que se ejecuta.

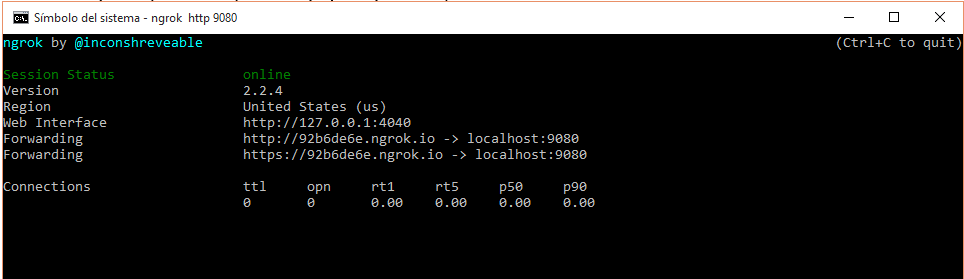
* 

Fig . Información ngrok

# Tips informativos

* Node Js trabaja con módulos, cada módulo es un script en javascript.
* Node es un intérprete de java script, todo lo que sea java se puede ejecutar con javascript.
* Npm es el instalador de módulos para Node Js.
* Para imprimir en consola se usa el comando *console.log('Hola App1');* (*Hola App1* corresponde al texto que se quiere imprimir).