# Introducción

Para la simulación del proyecto propuesto, se buscó la mejor forma de poder trasladar las características del Sistema Operativo que tiene la placa Intel Galileo (Linux Yocto) a un Sistema Operativo que pueda ser instalado en una PC. Para ello, se instalaron los módulos que el Yocto necesita en una máquina virtual con un Sistema Operativo Ubuntu.

Los módulos que difieren entre ambos sistemas operativos son:

* Node.js
* Npm

Se debe aclarar que en el proyecto, se mencionarán las instalaciones de otros módulos pero que deberán ser instalados también en la placa. Entre ellos se encuentran:

* Express
* Body-parser
* EmailJS
* Mysql
* EJS

Además, para poder llevar a cabo la simulación, se explicará cómo se debe instalar un servidor LAMP (Linux, Apache, Mysql, Php), utilizado para poder manipular la base de datos.

# Instalación de los módulos principales

Antes que nada, para tener una instalación limpia y sin problemas con los privilegios de administrador o super usuario, es recomendable que se cree una carpeta en el directorio del usuario actual. Sin embargo, es necesario tener una contraseña para el super usuario o para poder realizar acciones que impliquen privilegios administrativos, por ejemplo, para ejecutar el comando *sudo*. En este caso el nombre del usuario es “ubuntu”.

C:\Users\turco\Desktop\Pictures\Selection_001.png

El siguiente paso, aunque no es necesario, es dirigirse a la carpeta y desde allí comenzar la instalación:

* *cd /home/Ubuntu/practica2*

Luego, se deben actualizar los repositorios propios de Ubuntu con el comando:

* *sudo apt-get update*

C:\Users\turco\Desktop\Pictures\Selection_003.png

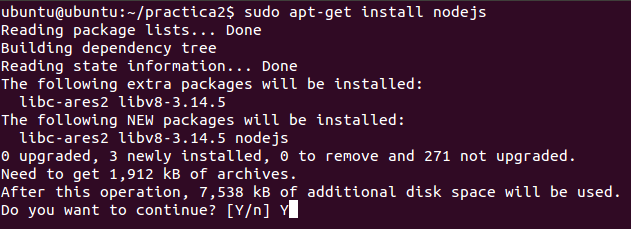
Una vez que el comando devuelve el prompt, se estará en condiciones para instalar los módulos Node.js y Npm.

## Node.js

Para realizar la instalación del programa Node.js se debe ejecutar el siguiente comando:

* *sudo apt-get install Node.js*

Cuando haga click en “Enter”, se iniciará el programa de instalación. Cuando pregunte por si quiere continuar, deberá poner la letra “Y” (en referencia a *yes*) y nuevamente, apretar la tecla “Enter”.



Para verificar la correcta instalación del Node.js, se averiguará cual es la versión con el comando:

* *Node.js –v*

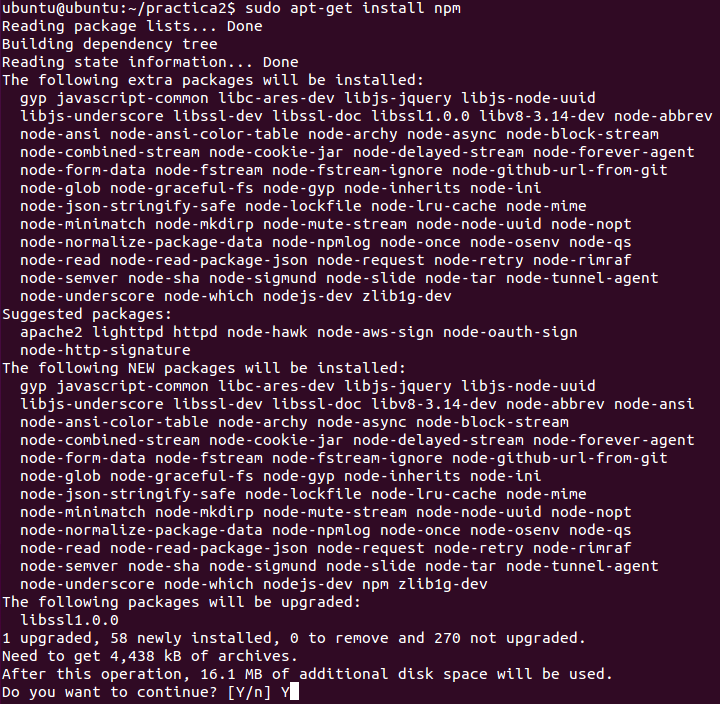


En este caso, la versión instalada es la v0.10.25.

## Npm

Ahora se debe instalar el manejador de paquetes para Node.js, npm. El comando es:

* *sudo apt-get install npm*



Nuevamente, poner la letra “Y” y apretar “Enter” para su instalación. Para corroborar que está instalado en npm, se averiguará cuál es su versión:

* *npm –v*

**

Con esta instalación terminada, ambos sistemas operativos están en las mismas condiciones.

# Instalación de los paquetes para el Node.js

La instalación de estos paquetes se realizará a través del manejador de paquetes npm. Para la correcta instalación de estos paquetes, NECESARIAMENTE se debe estar en la carpeta donde se va a trabajar con el Node.js. La carpeta propuesta en este informe es la que se encuentra en la siguiente ruta:

* */home/ubuntu/practica2*

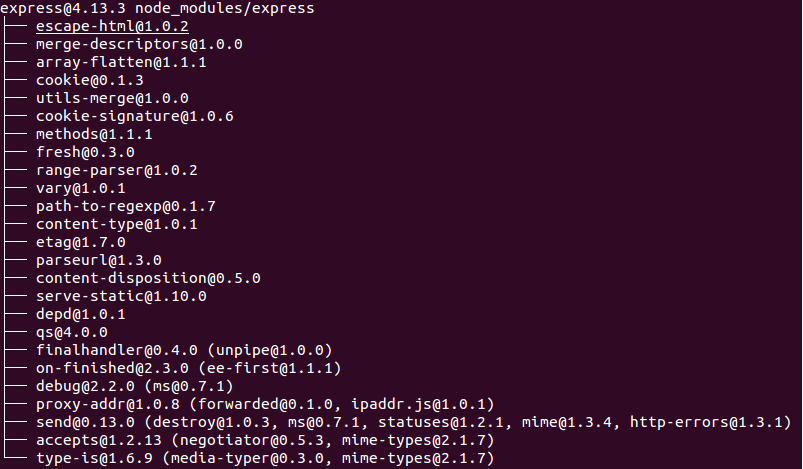
Una vez dentro del directorio se procederá a la instalación de los paquetes necesarios para la simulación.

## Express

“Express“ es un framework web rápido y minimalista para Node.js que proporciona un conjunto robusto de características para aplicaciones web, con una gran variedad de métodos de utilidad HTTP.

Para la instalación se debe ejecutar el siguiente comando:

* *npm install express*



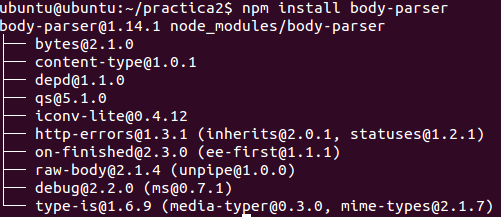
Que generará todo el árbol de módulos que contiene el “express”. Cabe destacar que en la carpeta donde se ejecutó el comando se generó una nueva carpeta con el nombre “node\_modules” donde serán instalados este y todos los módulos siguientes.

## Body-parser

Es el encargado de tomar todos los parámetros en formato JSON, provenientes de un método “post” desde un formulario html.

Para la instalación se debe ejecutar el siguiente comando:

* *npm install body-parser*

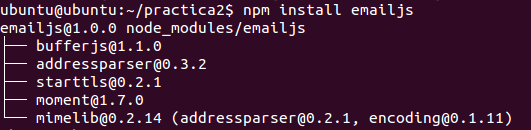


## EmailJS

Es una herramienta para poder mandar emails, con texto plano o con formato html y archivos adjuntos (archivos) desde un servidor Node.js a cualquier servidor smtp.

Para la instalación se debe ejecutar el siguiente comando:

* *npm install emailjs*

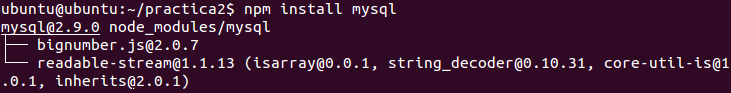


## Mysql

Es un driver de Node.js para utilizar mysql. Con esta herramienta se puede crear una conexión a una base de datos y poder realizar consultas (query).

Para la instalación se debe ejecutar el siguiente comando:

* *npm install mysql*



## EJS

Es una herramienta que permite desarrollar la visual de un código html. Combina datos y un template html para generar páginas web de manera dinámica.

Para la instalación se debe ejecutar el siguiente comando:

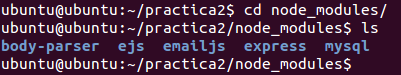
* *npm install ejs*



Si ahora observamos el directorio actual se podrá ver todos los módulos instalados. Si se ejecutan los siguientes comandos:

* *cd node\_modules*
* *ls*

Se observará las siguientes carpetas que corresponden a cada uno de los módulos instalados:



# Instalación de servidor LAMP

Se denomina "LAMP" a un grupo de software de código libre que se instala normalmente en conjunto para habilitar un servidor para alojar sitios y aplicaciones web dinámicas. Este término en realidad es un acrónimo que representa un sistema operativo **L**inux con un servidor **A**pache, el sitio de datos es almacenado en base de datos **M**ySQL y el contenido dinámico es procesado con **P**HP.

## Instalar Apache

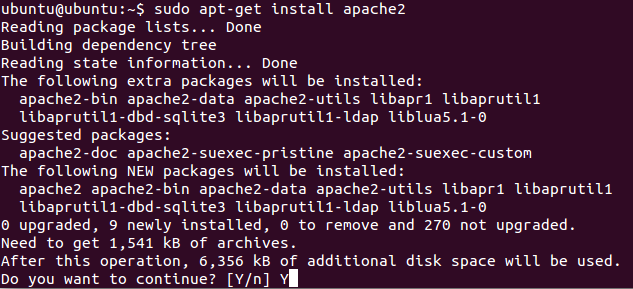
Podemos instalar Apache fácilmente desde el gestor de paquetes de Ubuntu, apt. El gestor de paquetes nos permite instalar con mayor facilidad un software desde un repositorio conservado por Ubuntu.

* *sudo apt -get update*

https://lh6.googleusercontent.com/KGkI1tIhEboDTuRr9CGUmVJf-9-dZLag_bpnf8tfMXB9KjAMIARNpG2s3w10wBFLJRRpr68P2r0fG5W-zPZjid7ItxVFUvrbey8eOCO8zp6e91cE1TUxV72vqTOKeAVr9xrueO9g

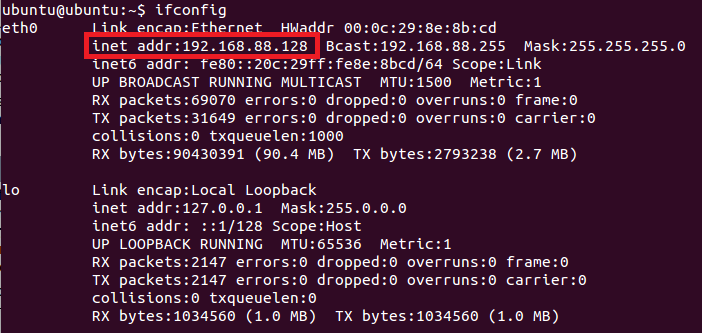
Actualiza los repositorios del Sistema Operativo.

* sudo apt -get install apache2

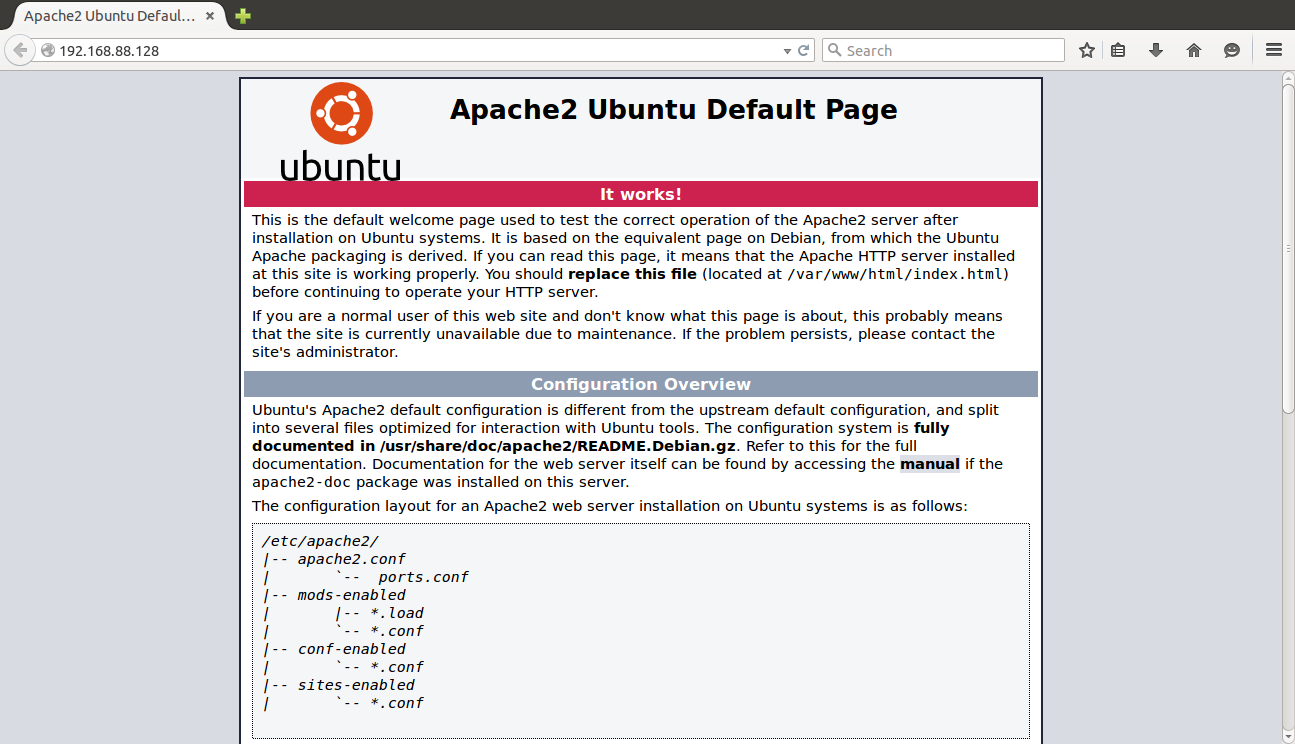


Como se mencionó antes, para continuar la instalación, se debe poner la tecla “Y” y luego presionar “Enter”.

Una vez finalizada la instalación del apache, podremos ver si el mismo se instaló correctamente. Para realizar esto, tendremos que conocer el “ip” de nuestro SO. Podemos obtenerlo a través del comando “ifconfig”.



Una vez realizado esto, se debe abrir el navegador y poner el “ip” obtenido. Si el servidor apache está corriendo, se verá la siguiente página, lo que indica el correcto funcionamiento.

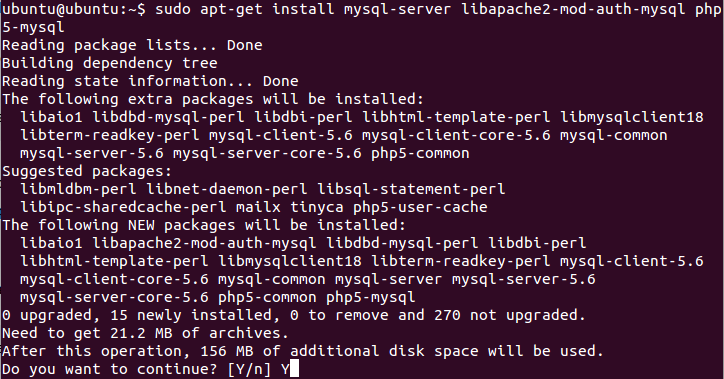


## Instalar MySQL

Ahora que ya tenemos nuestro servidor web configurado y corriendo, es el momento de instalar MySQL.

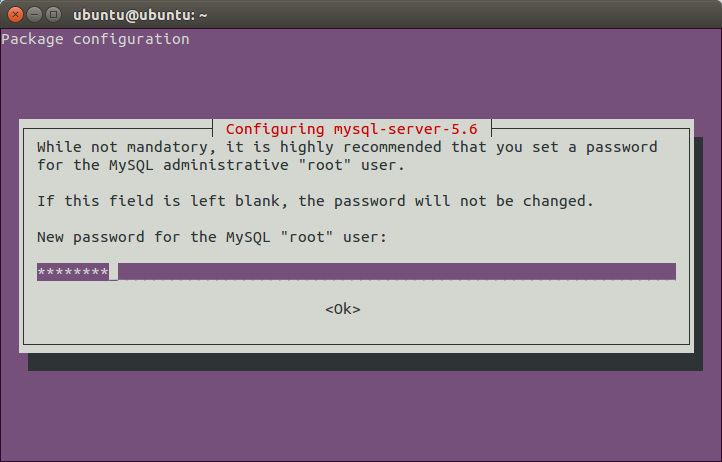
* *sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql*

MySQL es un sistema de gestión de base de datos. Básicamente, se encarga de organizar y facilitar el acceso a las bases de datos donde nuestro sitio puede almacenar información.

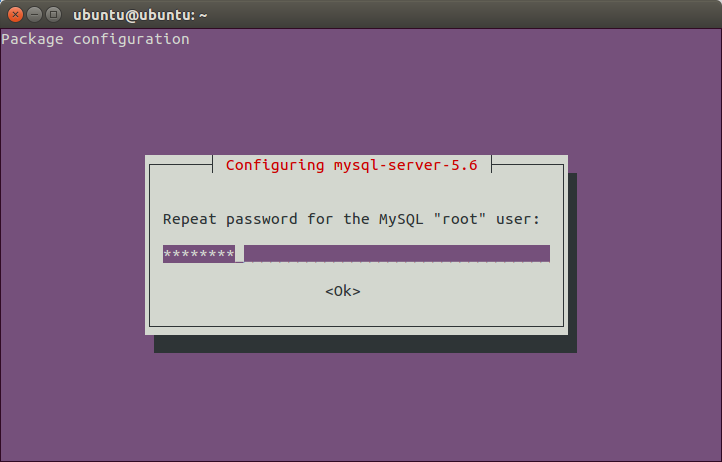


Tal como ya realizamos antes, debemos presionar la tecla “Y” y luego presionar “Enter” para continuar.

La instalación, nos pedirá que ingresemos una contraseña para un usuario “root”.



Luego, se debe repetir la clave antes ingresada.



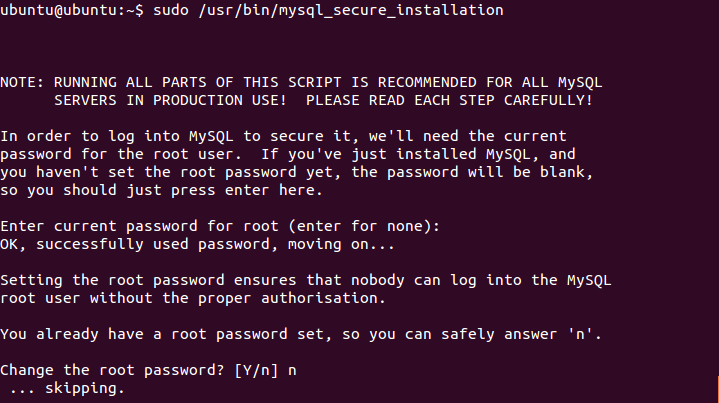
Cuando la instalación esté completa, debemos ejecutar algunos comandos adicionales para conseguir nuestro entorno MySQL configurado de forma segura.

* *sudo mysql\_install\_db*

https://lh5.googleusercontent.com/fs2Knt2LBMdi2E8WYx1p9KnjAlgAQKyL61eeTPi-dF1DKcjL09GOAsKisQe_jJf_YQrgw6DZwv_p7RaamRdZerdhCryC5KWpjGl8Cn5RKBxeAq0LRUmbsUIP6end0uX_ImLyMAN7

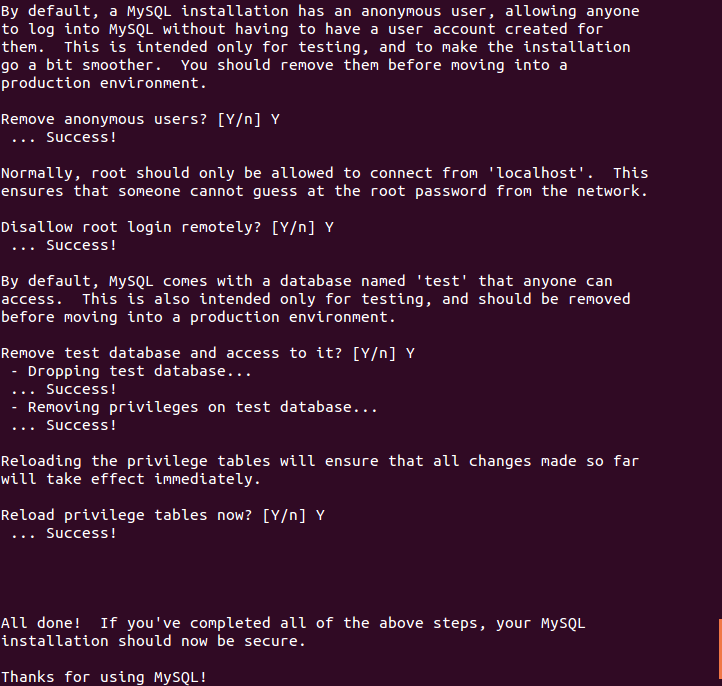
El comando antes ingresado, le indica a MySQL que tiene que crear su propia base de datos para la estructura del directorio donde se almacenará la información.

* *sudo /usr/bin/mysql\_secure\_installation*



Si no queremos cambiar la contraseña “root”, debemos poner “n” como se muestra en la imagen anterior y siguiendo, presionar la tecla “Enter”.

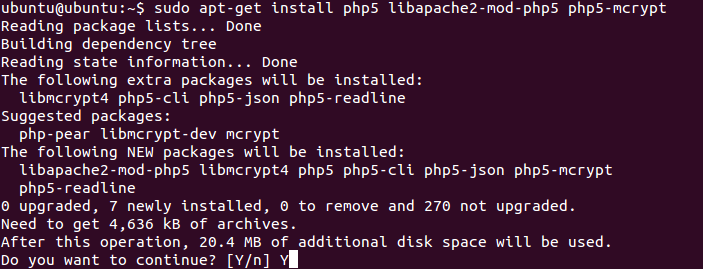
Luego, en las demás preguntas que nos realice la instalación, se deberá presionar “Y” y luego “Enter”.



## Instalar PHP

PHP es el componente de nuestra configuración que procesará código para mostrar contenido dinámico. Puede ejecutar secuencias de comandos, conectarse a nuestras bases de datos MySQL para obtener información, y entregar el contenido procesado a nuestro servidor web para mostrarlo.

* *sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt*

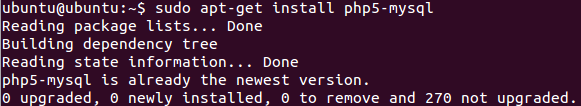


### Instalación de módulos PHP

Para mejorar la funcionalidad de PHP, podemos instalar opcionalmente algunos módulos adicionales.

En nuestro caso, se instalará el módulo “php5-mysql”.

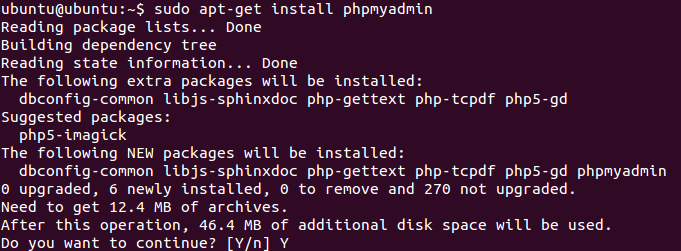
* *sudo apt-get install php5-mysql*



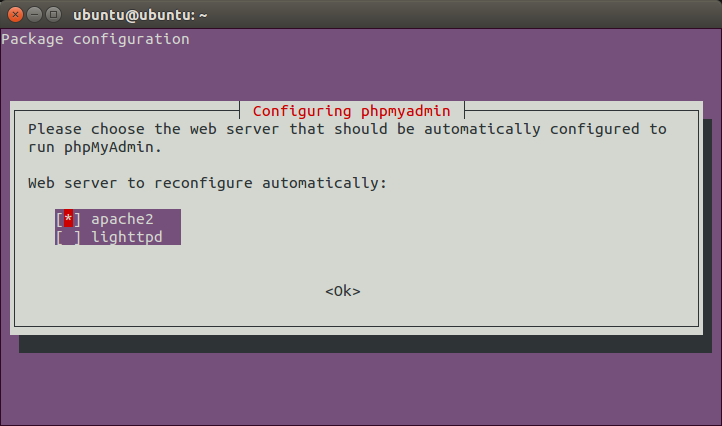
### Instalación de PhpMyAdmin

PhpMyAdmin nos da la posibilidad de administrar la BD mysql a través de un entorno amigable al usuario.

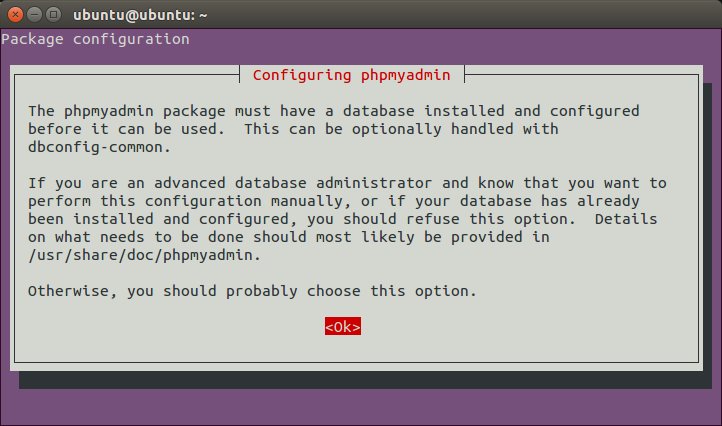
* *sudo apt-get install phpmyadmin*



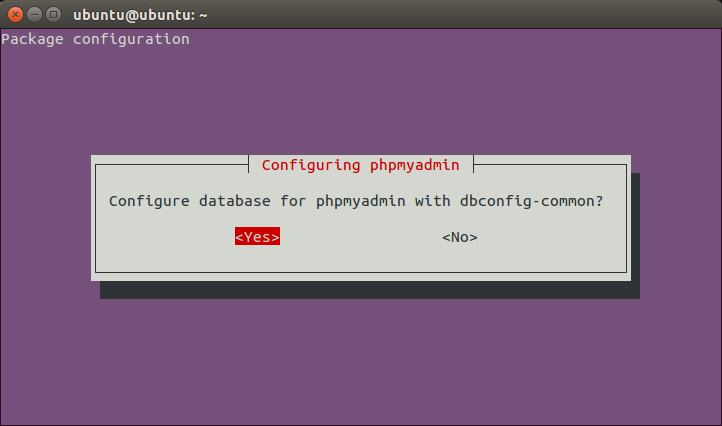
Debemos seleccionar que servidor se debe configurar, en este caso, se seleccionará “apache2” con la barra espaciadora y se presionará “Enter”.



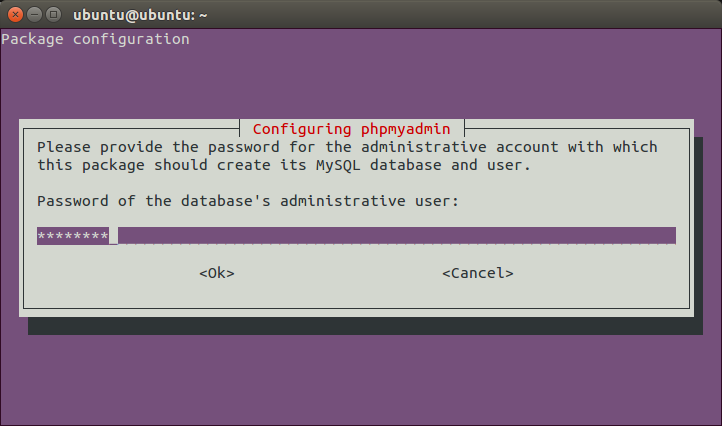
En la imagen siguiente, se realiza la configuración de PhpMyAdmin por lo que se debe presionar “Enter”.



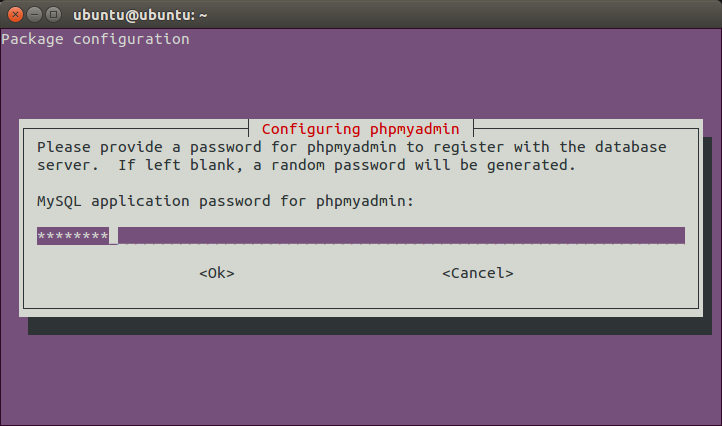
A continuación, se selecciona “Yes” y se presiona “Enter”.

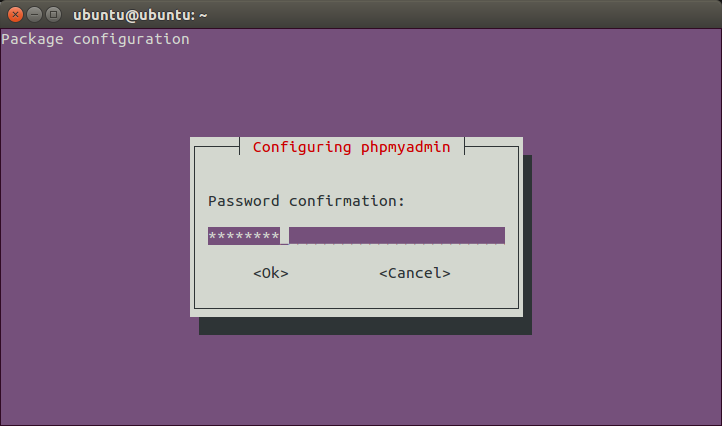


Se le pedirá la contraseña de su administrador de la base de datos.



Y finalmente, se le pedirá que elija y confirme una contraseña para la aplicación phpMyAdmin.





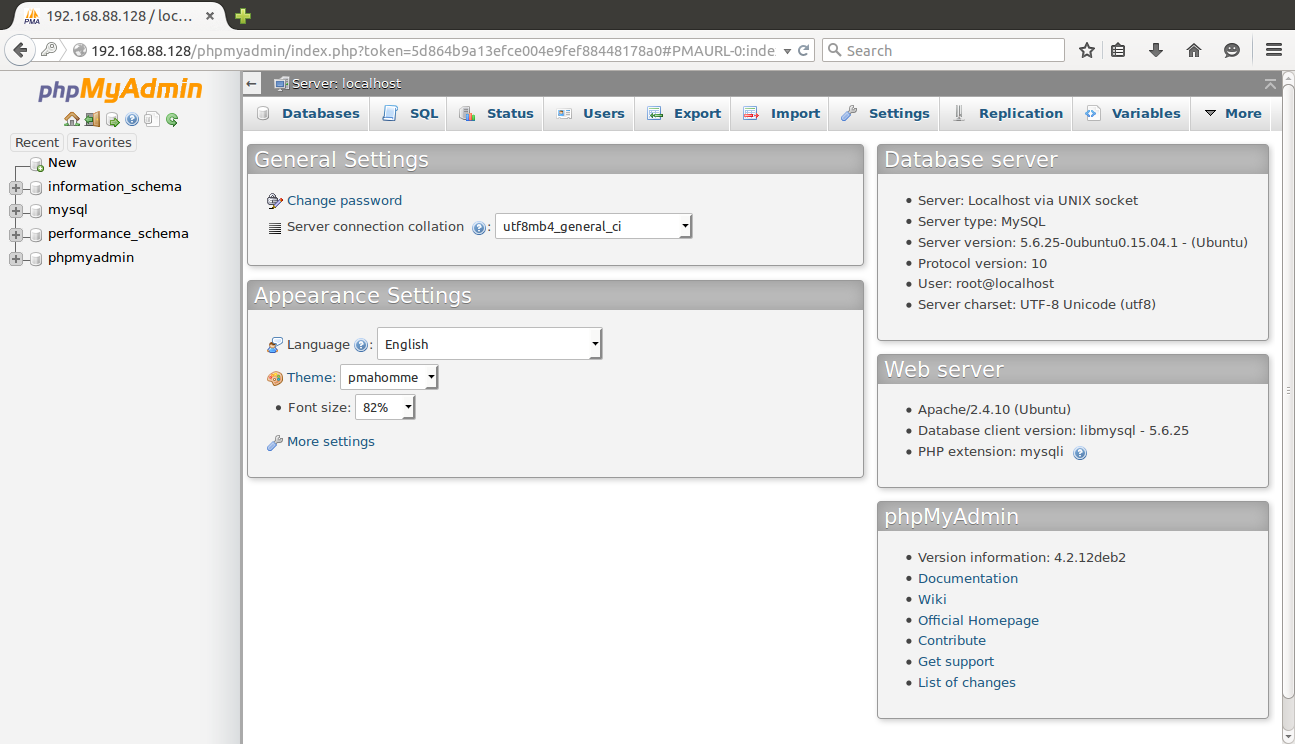
Finalmente, para corroborar el correcto funcionamiento del administrador de base de datos, se debe ingresar en la barra de direcciones del navegador la siguiente dirección:

* *ipDelServidor/phpmyadmin*

Deberá aparecer la siguiente ventana



Para ingresar al sistema, se debe poner el usuario root, junto con la contraseña del servidor de mysql que se proporcionó en la instalación. Una vez que se ingresó, se debe ver la siguiente ventana:



# Diseño de la base de datos

La comunicación entre la placa Intel Galileo y la PC será a través de un servidor que aloja la base de datos a utilizar. Para ello se propuso una base de datos que contiene dos tablas Usuario y Auditoria.

La tabla Usuario es aquella que albergará todos los datos del usuario, es por ello que tiene los siguientes atributos:

|  |  |  |
| --- | --- | --- |
| Atributos de la tabla Usuario | Función | Tipo de dato |
| id | Id único de la entidad usuario | Int-autoincremental |
| nombre | Nombre del usuario | Varchar(255) |
| apellido | Apellido del usuario | Varchar(255) |
| dni | DNI único del usuario | Varchar(255) |
| email | Email único del usuario | Varchar(255) |
| password | Contraseña única del usuario | Varchar(255) |
| temp | Temperatura elegida por el usuario | Varchar(255) |
| luz | Porcentaje de luz elegido por el usuario | Varchar(255) |

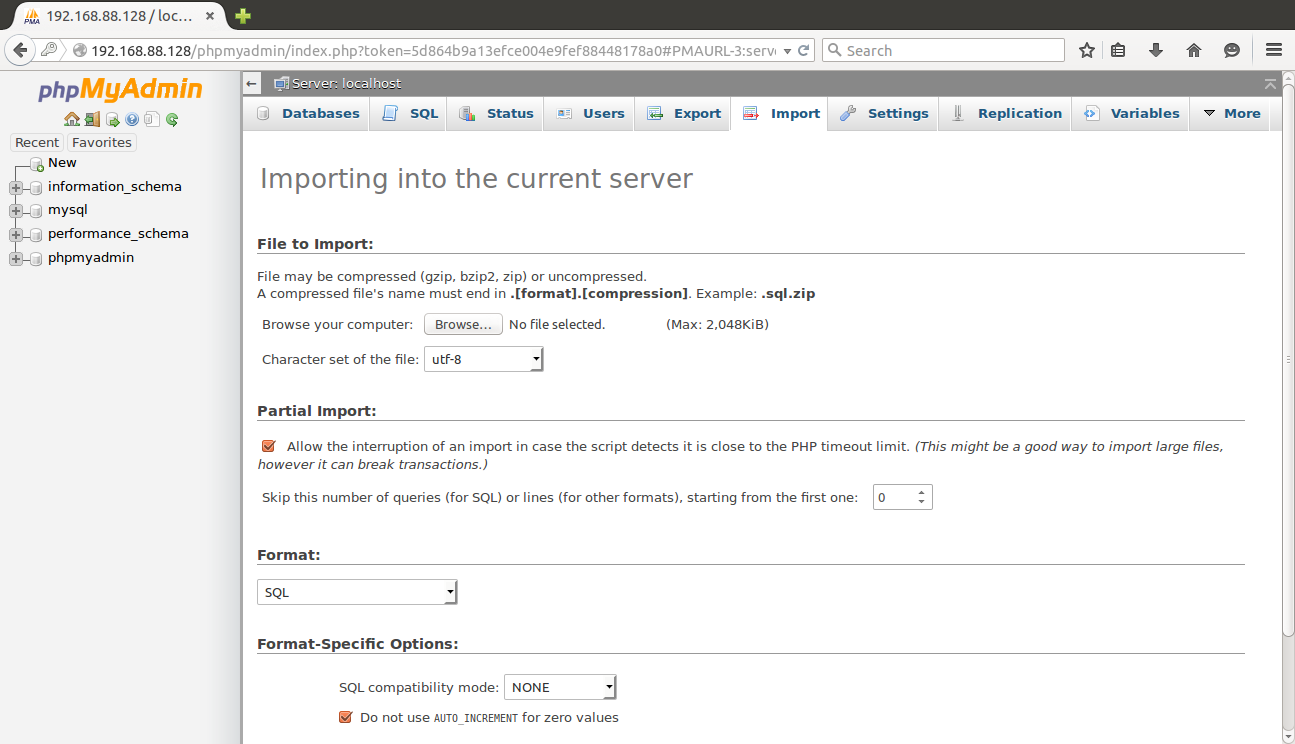
Mientras que la tabla Auditoria es aquella que albergará todos los datos de entrada y salida de los usuarios. Es usada para llevar el control de usuarios activos y un registro de los usuarios que han ingresado a la habitación. Tiene los siguientes atributos:

|  |  |  |
| --- | --- | --- |
| Atributos de la tabla Auditoria | Función | Tipo de dato |
| id | id único de cada auditoria | Int-Autoincremental |
| email | Email del usuario que ingreso a la habitación | Varchar(255) |
| fechaEntrada | Hora y fecha en la que el usuario ingresó a la habitación. | Varchar(255) |
| fechaSalida | Hora y fecha en la que el usuario sale de la habitación. Inicia como NULL e indica que el usuario todavía no salió de la habitación. | Varchar(255) |

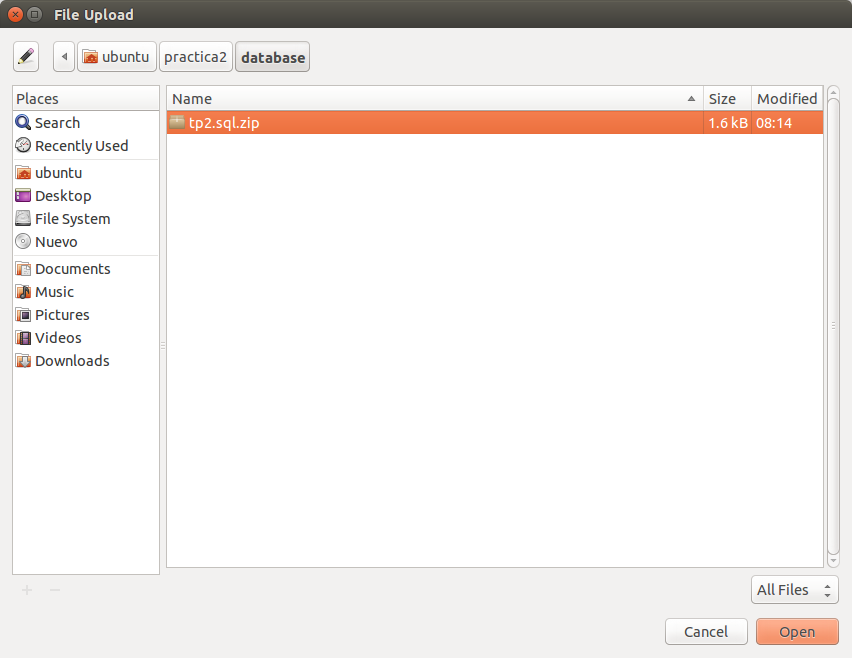
## Importación de la base de datos al phpmyadmin

En el caso de que se quiera importar la base de datos sin tener que definir las entidades anteriores, en la carpeta “database”, ubicada en el mismo directorio que este documento, se encuentra un archivo llamado “tp2.sql.zip” que contiene la base de datos “tp2” con las tablas anteriormente mencionadas.

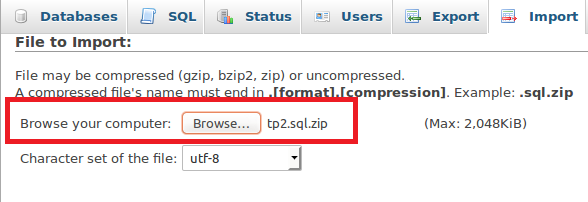
Para importarla, se debe ingresar al sistema “phpmydamin” y hacer click en la solapa “import”.



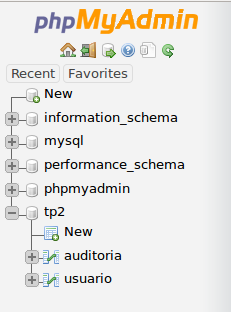
Luego, al lado de donde dice “Browse your computer:”, hay un botón, “Browse…” que permite buscar en el sistema de archivos una base de datos. Navegar hasta el directorio que se mencionó y elegir el archivo comprimido.



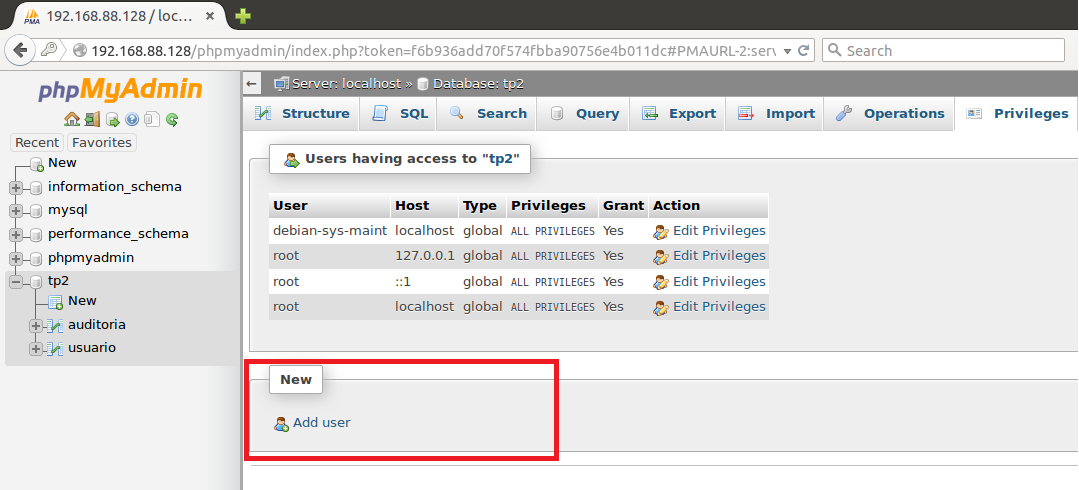
Una vez escogido, se pondrá el nombre del archivo junto al botón.



Finalmente, hacer click en el botón “Go” que está al final de la página de importación. Automáticamente se actualizará la página con la base de datos ya importada y lista para utilizar.

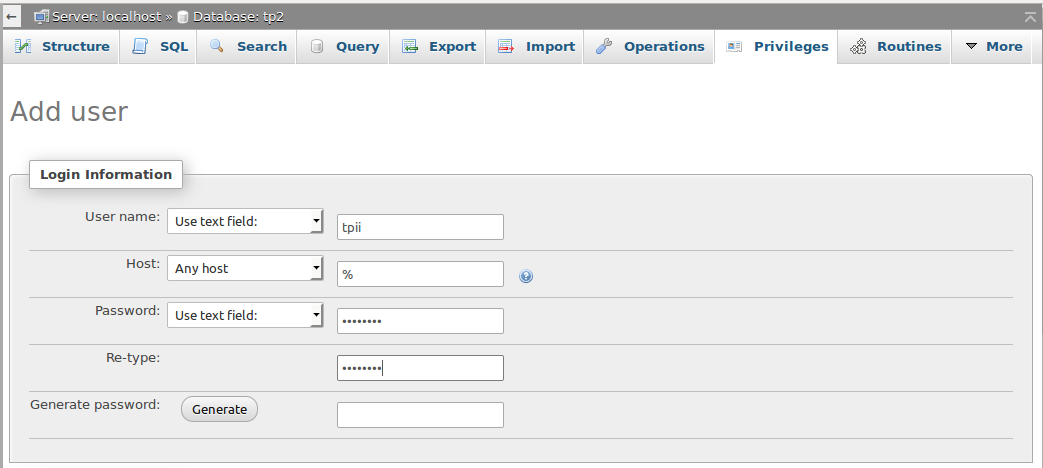


Para poder acceder a esta nueva base de datos desde la aplicación, se debe crear un usuario con ciertos privilegios. Para ello se debe ingresar a la solapa “Privileges” y agregar un nuevo usuario, haciendo click en “Add User”.

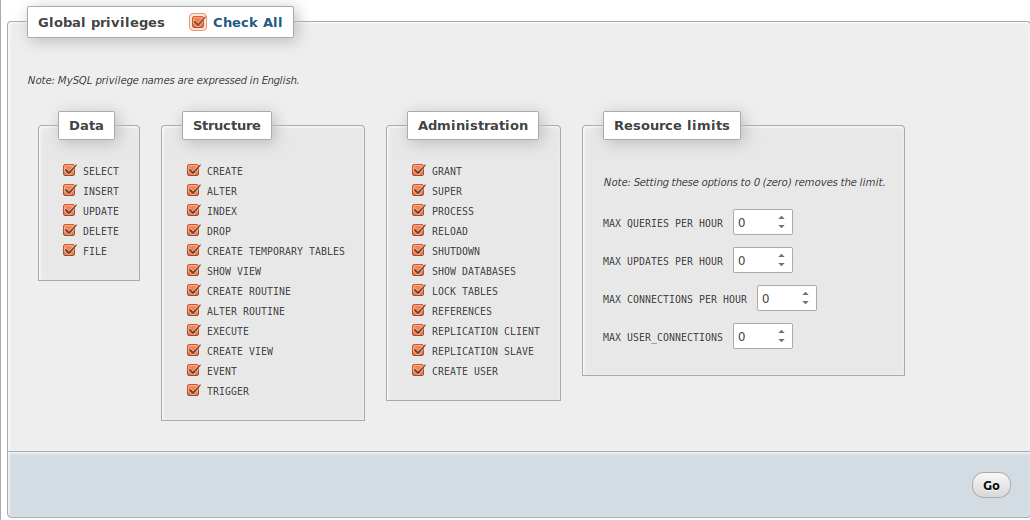


Se deben completar los siguientes campos:

* User name: nombre de usuario
  + En este caso: tpii
* Host: desde donde se conectará el usuario
  + En este caso: % (desde cualquier IP)
* Password: contraseña del usuario
  + En este caso: “tpii2015” (sin comillas)
* Re-type: poner de nuevo la contraseña

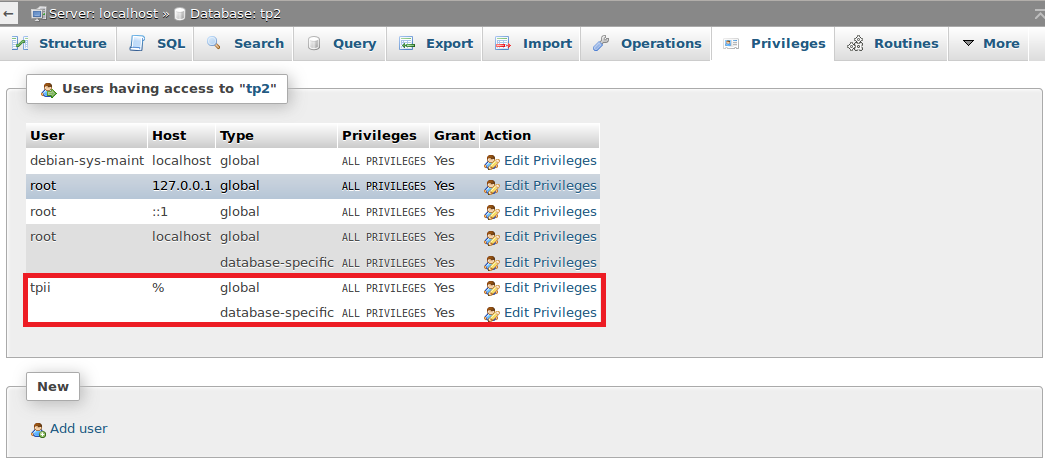


Siguiendo con la página, se puede asignar los privilegios que va a tener este usuario en la base de datos.



Para completar el registro, hacer click en “Go”.

Finalmente, en la pestaña de “Privileges” se podrá observar el nuevo usuario creado con los privilegios establecidos.



# Sistema WEB

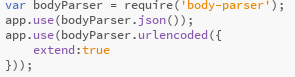
El servidor web en el cual se levantaran las páginas, se creó de la siguiente manera:

https://lh6.googleusercontent.com/e0AGSGFjITAQ_z7psfvcWHjycj_WTZgVqqXhJU8nwDr0iC3RYelTL_i-_nU9FXAAtv7f_s5Ya3DTWCzq07WatR7kweVSmNVCBYzYvul_cqsItbBRIGpCLcpGxuaoMuFcMMGZChGC

https://lh3.googleusercontent.com/3f6QAcE0RRvPVYK-x2dAPzK-KdyDLqBO2sJSFBkIBDJ9R02tHYvf-7G6eCV1FSFeMpliRBr6nmqPyzGcODd3R4_poQn-8Kc-2Lk0GTDJjRlZEOOk1yyRb0qpFhw0Fmo7CTcmbVuU

Todas las peticiones serán vistas a través del puerto 5000. La variable “app” representa la aplicación WEB que capturara los requerimientos de los diferentes formularios.

Se realizó la configuración del “body-parser” para que se puedan enviar parámetros a través de los métodos “post” de las páginas web.



Se configuró como motor de vistas a la herramienta “ejs”:

https://lh6.googleusercontent.com/GWkKGquALLLZFo-AKfdSRtHwsXEFTUKJzttOKaHZMzmfFdCe32VNKL5oRaHbh39dwGMCOIacM8mQqolD9mqtPCXISdmTYvSWOKhC-bD89XP_n2GusxVcR6xBivF6MxEeK-b0nwHH

## Páginas WEB

El sistema cuenta con cuatro páginas web:

1. inicio.ejs

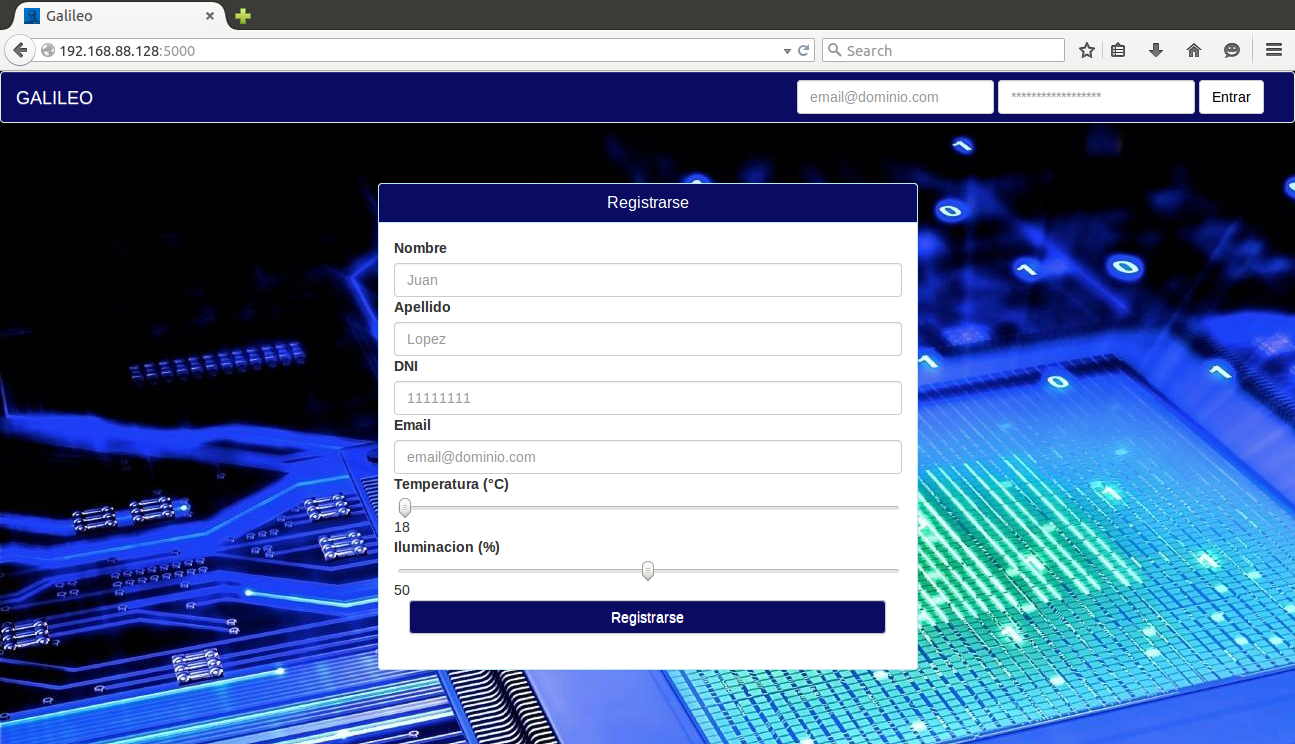
Esta página cuenta con un formulario de registro, el cual permite a un usuario registrarse al sistema y en caso de ya estar registrado, iniciar sesión.

El formulario de registro, permite establecer además de los datos personales, el valor de temperatura y luz con el que le gustaría que se acondicione el ambiente. Una vez registrado, se le enviará por mail la clave ÚNICA de acceso al sistema. Esta clave la podrá utilizar para ingresar al sistema web y modificar su perfil o parámetros, y para ingresar a la habitación, ingresándola a través de un teclado numérico.

Esta página mostrará tres mensajes de error.

* Indicación de registro exitoso
* Indicación de fallo de registro
* Indicación de error de usuario/contraseña

La página se verá de la siguiente manera:



Requerimientos generados al programa “js” desde la página “inicio.ejs”:

* Acceso a la página inicio.ejs a través de la dirección “ip:5000”.

Tal como se mencionó antes, los requerimientos serán capturados a través de la variable “app”. En este caso, se captura el “get” que realiza el navegador para obtener la página web. Al ingresar la método, se le envía la página web sin mensajes de error ya que es la primera petición.

https://lh3.googleusercontent.com/lhvnHD0xfzGwgu4mHCqXTrtj09GxjFLiIWhtazfEJV_WotBAHhxJmOPbWRZVhOt2jVG_ocxGHt6j7PJMNY54WoUWiHtD0H88QkDD9OVkf2Y_MFco2Hyslhdy7a92R1zp5xZ_ii3C

La variable appDir es la que nos da la dirección donde se encuentra el archivo sobre el cual estamos programando, por lo que sí a las páginas web las ponemos siempre en el mismo directorio, no es necesario configurar la ruta manualmente.

Lo antes mencionado puede realizarse de la siguiente manera:

https://lh3.googleusercontent.com/72yxkc6tyEKUo41ySAd6YkMiUItkZimrVhsgOh4ZCmYzF0bPdqiTXwPla0tZMczUweaBEOjG81xNqCJaLC6hAgt4bt7AQF72wCmWwr6kjxfdda2OvT9IF9V9iOkKWN__j9Kisiwp

* Registro de usuario

Cuando el usuario presiona el botón “Registrarse”, se genera una petición a través de un método “post” llamada “/registro”. De esta petición, se deben obtener todos los datos que ingreso el usuario. Los mismos, pueden obtenerse a través del parámetro “req.body.*parametro*”.

En este caso, se debía obtener en primera instancia el email para corroborar que el mismo ya no esté registrado en el sistema, lo mismo ocurriría con el password ya que ambos son únicos. Una vez se haya chequeado esto, el usuario será registrado y se le enviará un email a su casilla con el password generado.

Terminado esto, se responderá al navegador con la página “inicio.ejs”, y un mensaje de “éxito” en caso de que se haya registrado el usuario, o de error en caso contrario.



A continuación, se muestra en forma de ejemplo, el encabezado del “form” que se utilizó en la página “inicio.ejs” para el “/registro”, con la finalidad de visualizar cómo se deben configurar los parámetros “method” y “action”.

https://lh5.googleusercontent.com/o0c_k6aJB1kQyV9EdBZEoqFfNYKNstMGsYayUPvN0CcMkPKaHGRKnjL5Lyctn4Bt2CcvJQrJujFz2aNRxDOz4zl9ofzEeA3RQRB5BrEPnfVpjvxFe8O8_moyctNI8M0hz7dACnqY

Los mensajes de error que se mencionaron anteriormente, se visualizan en la página de la siguiente manera:

https://lh5.googleusercontent.com/-_rEz31y2SN2ACq9XmH0TMrqvvHZhkmFqQ2gqvO3gSVFRYiDAt_IMTLlMhUe0CDryiFW35TZJm5lgysK5hGpxTBP4Du2sEGlcjFM00whPSU3mMOwxG7FpF94e5WWMaYg2nVZa6GA

Las variables que están dentro del tag **<%=   %>** son las enviadas cuando se responde la petición al navegador.

* Inicio de sesión

Este requerimiento se recibirá a través de un método “post” llamado “/log” cuando el usuario presione el botón “Entrar”.

En el sistema existirá un “email” y “password” de administrador. Los mismos fueron seteados de la siguiente manera:

https://lh3.googleusercontent.com/wkHwMX3y-xRyO-y6cZoLdV69tmtlCu-vApBkkOGU0yqhQ_AgCKQ3mKRS5PoKh72wpkcCnkDXq92aL4KQKH7qkpFVpS-UIEj8V42u75xziHTv_RirGJyU92CctWYepl5WGG4vVjSf

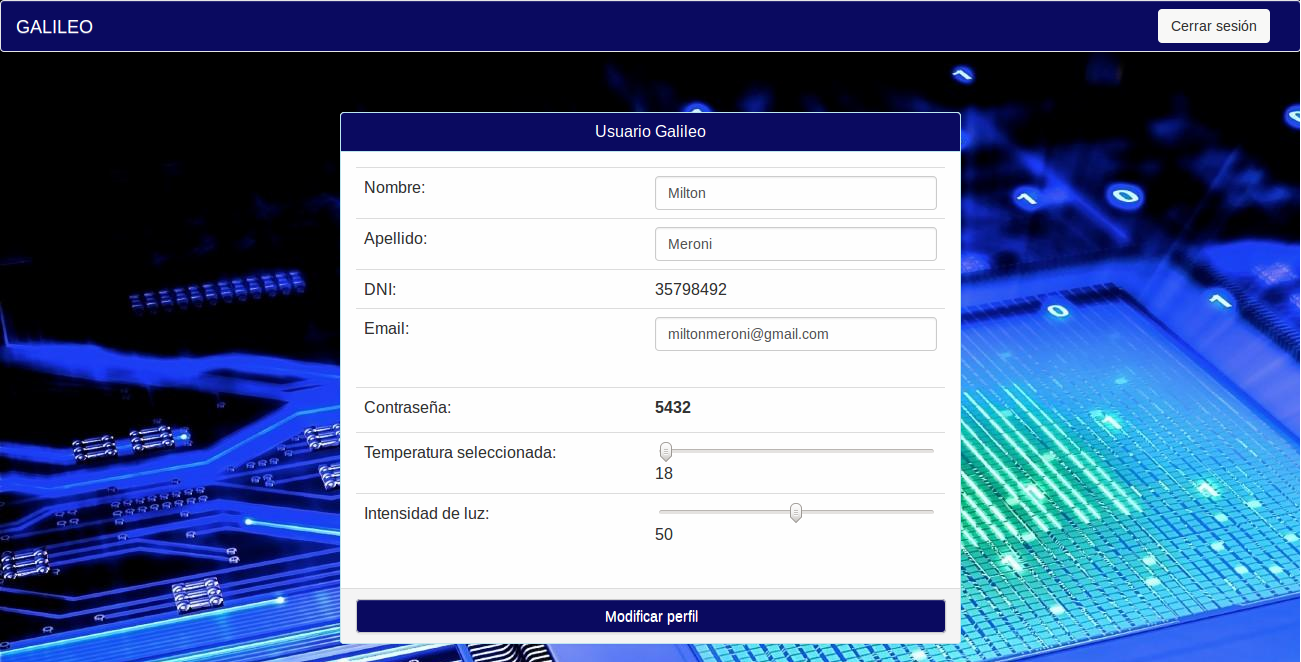
De la misma manera que en el formulario de registro, los datos ingresados se obtienen a través de req.body.*parametro*.

Esta petición chequeará que el email ingresado este en la base de datos y corresponda con la password ingresada. Luego, verificará si corresponde con el email del administrador, de ser así, redireccionará a la página “/perfilAdministrador.ejs”, sino, a la página “/perfilUsuario.ejs”. Si el email no corresponde con ningún usuario registrado, se redireccionará nuevamente a la página “inicio.ejs” y se enviará un mensaje de error indicando usuario/contraseña inválidos.

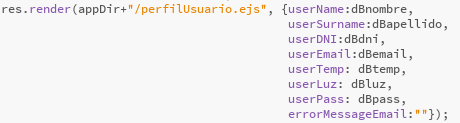


1. perfilUsuario.ejs

El perfil de usuario, le mostrará a un usuario registrado todos sus datos y le dará la posibilidad de modificarlos. La página es la siguiente:



Tal como se pudo visualizar en el método “post /log” la redirección al perfil de usuario, no solo contiene la página “perfilUsuario.ejs”, sino que además se agregan datos. Los mismo son utilizados para completar cada campo del formulario que posee esta pagina.



A continuación se muestra un ejemplo de uno de los “input” de la página “perfilUsuario.ejs” para que se visualice como se envia el dato “userName”.

https://lh5.googleusercontent.com/060oU5Z7IcTWSjds6OIQrIU1TrrXjKe8A6Jrz6AdyXXdW98ydZQgVp-S9d0ZCyHFiB0X5U9ySzUqQuboZDRwtNPgAX1tM-y0yI1QuCmTxqQtUT54ffA6bEj5AfDAFIJ9uX4urVxJ

Requerimientos generados al programa “js” desde la página “perfilUsuario.ejs”:

* Cerrar sesión

Esta petición es generada al presionar el boton “Cerrar sesion”. Se produce a través de un método “post” llamado “/cerrarSesion”. Tiene como finalidad redireccionar a la página “inicio.ejs”.

https://lh4.googleusercontent.com/eQg6DUAWXVHNXchjVOybsHRJh5PEeqP66kSF4j7Ok2fvww9YgQmkmtJ48kuPB5yxl92-IR9HvfZZa4nFjmPSEkwF6t0P2fAm_VNFV-xU_kEZ-Tq0BLd4eY5SqlHdigwY_Ws5AYTg

* Modificar perfil

Esta petición se genera al presionar el botón “Modificar perfil” y se produce a través de un método “post” llamado “/modPerfil”.

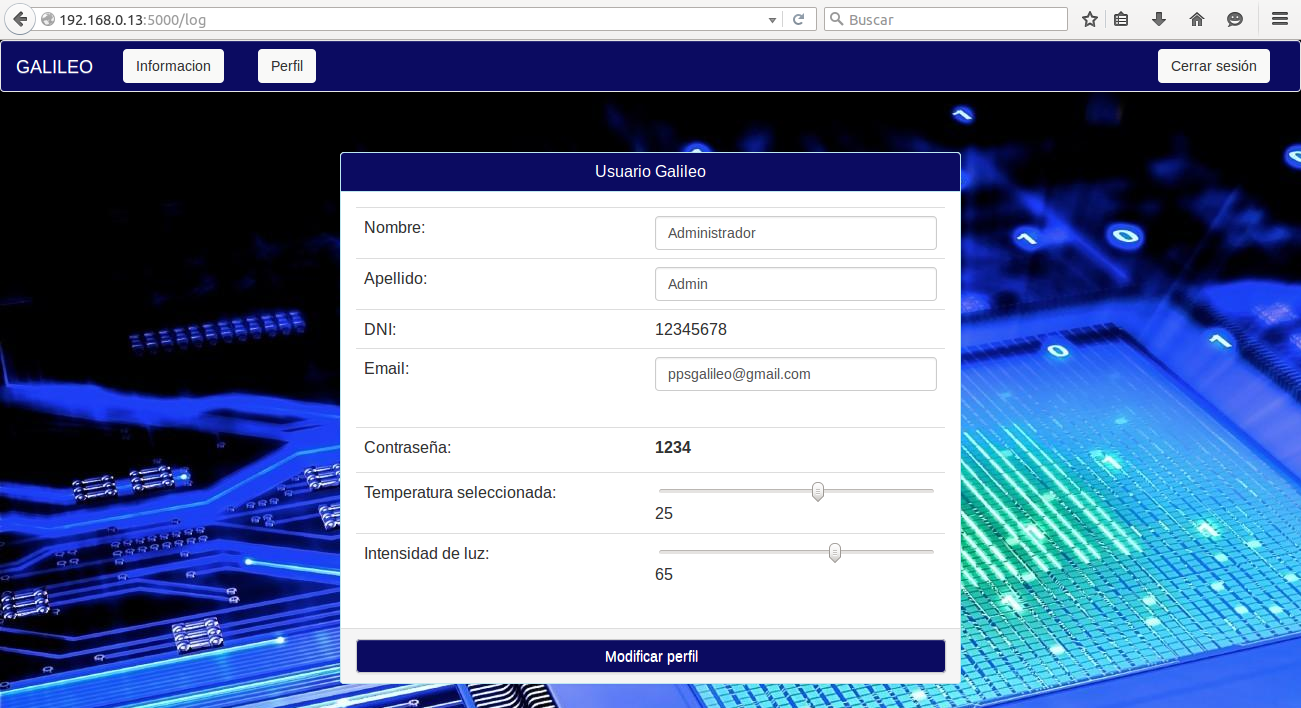
Este método tomara todos los datos del formulario, chequeara que el email ingresado sea el suyo o en caso de haberlo cambiado que no exista en la base de datos. De ser así, modificará el perfil y re direccionará a la página “perfilUsuario.ejs” completando el formulario con los nuevos datos. Si el email ingresado ya está registrado, re direccionará a la página indicando este error.



1. perfilAdministrador.ejs

El perfil administrador se ve de manera muy similar al del usuario, pero en este caso, se agregan dos botones más a la barra superior. Los mismos son “Información” y “Perfil”.

La página se ve de la siguiente manera:



Al igual que sucedió con la página “perfilUsuario.ejs” al redireccionar al perfil de administrador se envían datos y los mismos cumplen la misma función que en el caso anterior, completar el formulario de perfil.



Requerimientos generados al programa “js” desde la página “perfilAdministrador.ejs”:

* Los requerimientos “Modificar perfil” y “Cerrar sesion” son los mismos que en el caso de “perfilUsuario.ejs”.
* Información

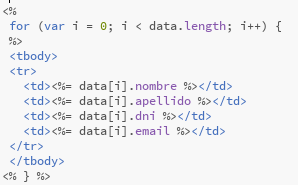
Este requerimiento se produce al presionar el botón “Información”. La petición viene a través de un método “post” llamado “/historico”.



Este método nos redireccionará a una página llamada “historicoAdministrador.ejs”. En esta página se mostrará información de los usuarios registrados en el sistema, los accesos históricos y los usuarios que actualmente están en la habitación.

Para enviar la información, se realizan dos consultas. La primera es para obtener todos los usuarios, y la segunda, para obtener los registros de accesos a la habitación. En el caso de esta última, se discrimina por la fecha de salida (si es NULL) si el registro es histórico o actual, entendiéndose por actual a un usuario que todavia no salio de la habitación. Los datos son enviados a la página WEB en formato “JSON”. En esta página hay un template “ejs” el cual crea las tablas a partir de los datos enviados por el programa.

A continuación, se muestra una porción de la página “historicoAdministrador.ejs” donde se muestra como arma las tablas a partir de los datos que se envían.



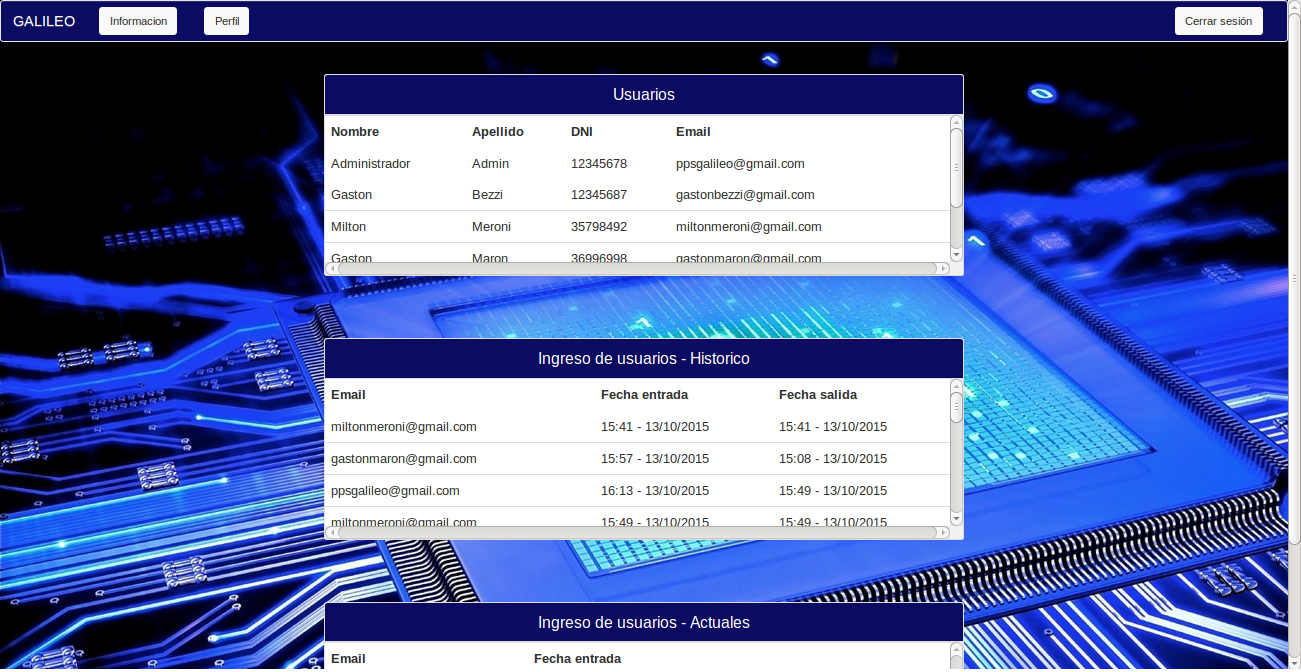
* Perfil

Esta petición es generada al presionar el botón perfil, y llega a través de un método “post” llamado “/perfil”. Esto tiene como función regresar al perfil del administrador, en caso de que se encuentre en la página “historicoAdministrador.ejs”.



1. historicoAdministrador.ejs

Tal como se mencionó en el punto anterior, esta página mostrará en forma de tablas información sobre los usuarios registrados, los accesos históricos y los actuales. La página es la siguiente:



Los requerimientos generados al programa “js” desde la página “historicoAdministrador.ejs” son “Cerrar sesion”, “Información” y “Perfil”. Estos son los mismos que se explicaron en los puntos anteriores.

## Envío de emails

Para enviar emails a través de nodejs, tal como se explicó en el comienzo del informe, es necesario la instalación de la librería “emailjs”. A continuación, se muestra la configuración del servidor de email.



Se creó una función que recibe el email del destinatario, el nombre y la password que deberá utilizar para ingresar al sistema. Esta función envía al usuario la clave ÚNICA de acceso.



El mensaje que se recibe es el siguiente:

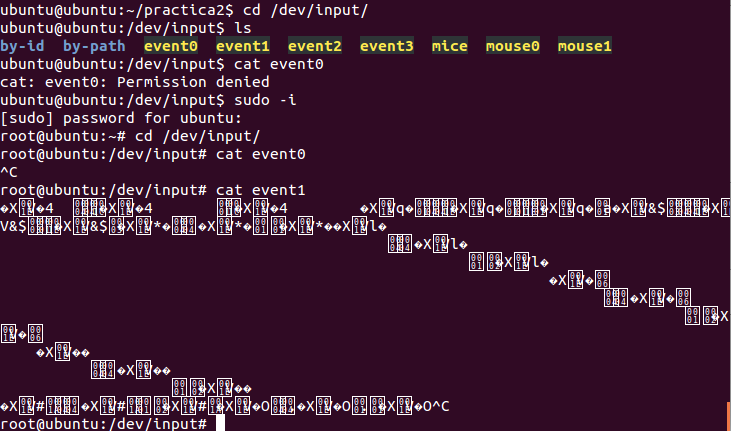


# Uso del microcontrolador y simulación de sensores y actuadores

Para tarea del microcontrolador, se tuvo que combinar la programación en lenguaje C y en el Node.js. Esto se debe a que el programa en lenguaje C captura los eventos del teclado y los escribe en un archivo, mientras que el programa principal del Node.js es una función que se ejecutará cada un segundo, leyendo ese archivo y verificando que la contraseña sea la de algún usuario.

## Programa en lenguaje C

Este programa captura los eventos provenientes del teclado y los pone en un archivo de texto. Este archivo se debe encontrar en el mismo directorio que la aplicación. Para que pueda detectar el teclado, hay que averiguar a qué evento dentro del directorio “*/dev/input/*” ya que el sistema operativo tiene varios eventos que atender. Para ello **antes de ejecutar el programa** se debe probar en una consola cual es el evento buscado:



Debe notarse que para visualizar el evento debe tenerse privilegios de usuario administrador. Luego con el comando “cat” se podrá observar lo que están arrojando los eventos. El teclado puede ser cualquiera de los eventos, por lo que habrá que ver los eventos que tiene la distribución de Linux que se esté usando. Cuando se observe que al apretar las teclas aparezca esos caracteres ilegibles, quiere decir que ese es el evento que tenemos que observar. En este caso, es el “event1”. En el programa “express.js”, en la parte en la que se ejecuta el programa en lenguaje C, se debe cambiar la variable evento ubicado en la línea 636:



Donde dice “event1” poner el evento que corresponda el evento del teclado.

## Programa principal en Node.js

En este programa, además de crear el servidor web y atender las peticiones, también realiza la ejecución del programa en lenguaje C y una tarea que se repite cada un segundo. Para la ejecución del programa se tiene el siguiente bloque de código:



El programa principal va abrir el archivo “teclas.txt” en modo lectura, y verificar si se presionó la tecla “Enter”.

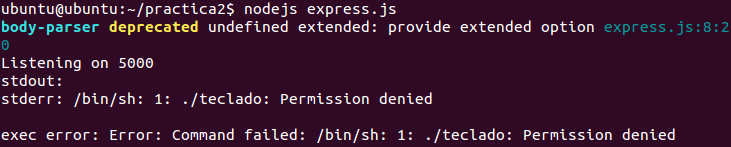
En caso de que se haya presionado, se obtendrá la contraseña del archivo y se limpiara el mismo. Una vez realizado esto, se obtendrá el usuario de la base de datos. Cuando se ingresa esta nueva clave, en caso de que el usuario exista, pueden haber ocurrido dos situaciones. Que el usuario esté ingresando a la habitación, o que el mismo esté saliendo. Por lo tanto, al obtener al usuario, se debe chequear que este no tenga una auditoría activa, si esto NO es así, se debe crear una nueva auditoría para registrar el ingreso y se da inicio a la función “simulacionSensores”. Esta función recibe los parámetros de temperatura y luz que el usuario predefinio en su perfil, y tiene como objetivo simular el acondicionamiento de la habitación.

Si existe una auditoría activa por este usuario, se actualiza la misma con la fecha de salida, y se detiene la función de acondicionamiento.

Como primera medida, la función “simulacionSesores” simula el encendido de la luz al porcentaje indicado por el parámetro “luz”. Luego, dentro de sí misma, tiene una sección que se repetirá cada un segundo con la finalidad de simular el acondicionamiento de la temperatura del ambiente. En esta sección se chequeara si se debe bajar, aumentar, o mantener la temperatura. La temperatura actual (sensor) se generó con una función que aporta valores random entre 18 y 30 grados.

### Posibles errores

Error de permiso denegado de ejecución de ./teclado:

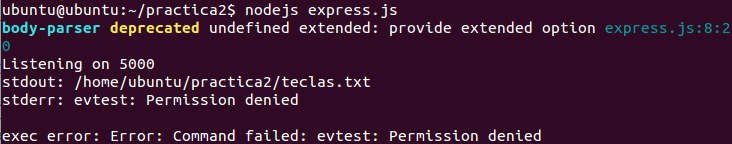


Solución:

Dentro del directorio, cambiar los permisos del programa ejecutable con el siguiente comando:

* *chmod 775 teclado*

Error de permiso denegado del evtest:



Solución:

* Ejecutar el programa como administrado anteponiendo el comando sudo
  + *sudo nodejs express.js*

# Comunicación con la PC

Para la comunicación con la PC (que en nuestro caso es la base de datos) se implementaron una serie de variables, que representan los parámetros para la conexión con la base de datos; y una serie de métodos que establecen la conexión y realizan consultas en la base de datos.

Las variables son las siguientes:

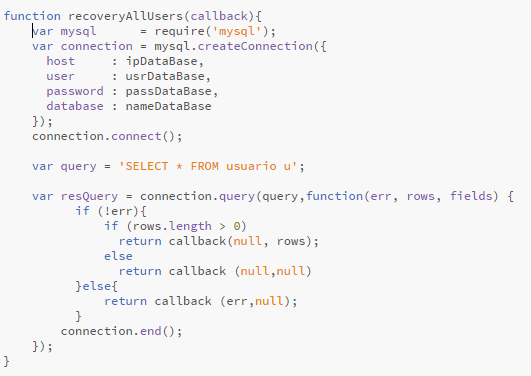
* var ipDataBase = '192.168.88.128'; // ip de la base de datos
* var usrDataBase = 'tpii'; // nombre de usuario
* var passDataBase = 'tpii2015'; // contrasena
* var nameDataBase = 'tp2'; // nombre de la base de datos

Estas variables van a ser configuradas cada vez que se lo necesite, en caso de que se haya cambiado la IP de la base de datos o si, cuando se realizó la importación de la base de datos, el usuario y/o la contraseña que se establecieron sean diferentes.

Los métodos son sobre las entidades de la base de datos, Usuario y Auditoria. Cabe destacar que en todos los métodos, se configura los parámetros para la conexión con la base de datos, se efectúa la conexión, se realiza la consulta y, al finalizar el método, se cierra la conexión.

Sobre la entidad Usuario, se establecieron los siguientes métodos:

recoveryAllUsers(callback)

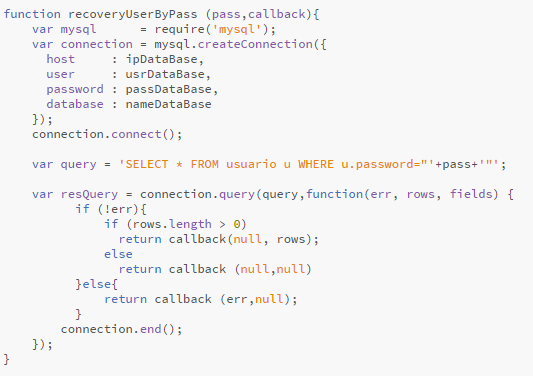
* Recupera todos los usuarios que existen en la base de datos. En caso de que no haya ninguno retorna (NULL, NULL). En caso de error, devuelve el error producido.

recoveryUser(email,pass,callback)

* Consulta si existe el email y el password de un usuario en particular. Retorna el mismo resultado que el método recoveryAllUser.

recoveryUserByEmail(email,callback)

* Recupera el usuario que tenga el email enviado por parámetro. Retorna el mismo resultado que el método recoveryAllUser.

recoveryUserByPass(pass,callback)

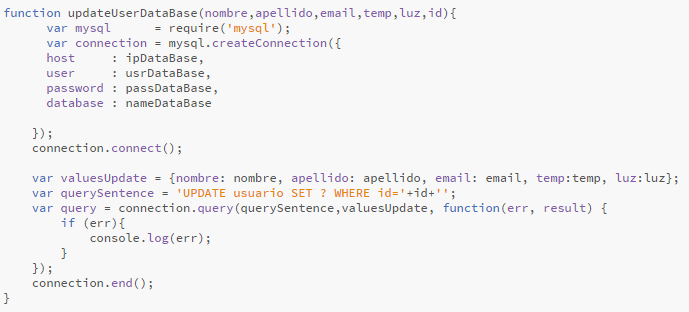
* Recupera el usuario que tenga la contraseña enviada por parámetro. Retorna el mismo resultado que el método recoveryAllUser

saveUserDataBase(nombre,apellido,dni,email,pass,temp,luz)

* Guarda un usuario con todos los valores que vienen por parámetro.



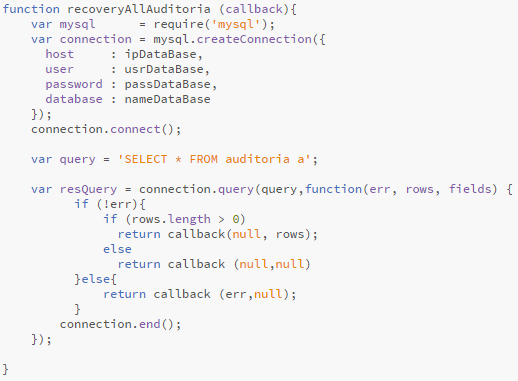
updateUserDataBase(nombre,apellido,email,temp,luz,id)

* Actualiza los atributos que el usuario puede llegar a modificar. Entre estos se encuentran el nombre, apellido, email, temperatura y luz. 

Sobre la entidad auditoria se realizaron los siguientes métodos:

recoveryAllAuditoria(callback)

* Recupera todas las auditorias que se encuentran en la base de datos. Retorna las filas obtenidas o NULL en caso de que no haya. En caso de error, envía el error que se ocasionó.



saveAuditoriaDataBase (email)

* Genera la fecha de entrada con el formato “hh:min – dd/mm/yy” e inserta dicha fecha junto con el email que viene por parámetro.



updateAuditoriaDataBase(email)

* Genera la actualización de la auditoria, es decir, le asigna una fecha a la fecha de salida que esté establecida como NULL en una auditoria que contenta el email que viene por parámetro.



**Correccion de error de conexión con la base de datos.**