

Control de acceso y ambientación de una habitación

Meroni, Milton Alberto; Marón, Gastón Ariel

Taller de Proyecto II

Ingeniería en computación

Facultad de Informática – Universidad Nacional de La Plata

Contacto: miltonmeroni@gmail.com; gastonmaron@gmail.com

Fecha: 14 de Diciembre de 2015

Índice

Objetivos del proyecto	1
Diferencia entre lo propuesto y lo realizado	1
Intel Galileo GEN1	2
Características principales	2
Node.JS	2
Paquetes externos utilizados	3
Instalación de los paquetes.....	4
Aplicación - Control de acceso y variables de ambiente	5
Comunicación entre Node.js, programa en lenguaje C y el usuario	5
Sistema WEB.....	6
Páginas WEB	7
Envío de emails.....	14
Base de datos.....	15
Diseño	15
Comunicación con Base de Datos.....	17
Programa en lenguaje C	19
Detección del evento apropiado.....	19
Ejecución del programa en Lenguaje C desde Node.js.....	20
Sensores y actuadores	21
Esquema de conexiones	22
Funciones principales.....	23
Función “iniciarIntelGalileo()”	23
Función “main()”	23
Función “simuladorSensores”	24
PC Servidor.....	24

Aplicación Spotify	24
Instalación del módulo node-spotify.....	25
Instalación de la librería “libspotify”	25
Generación de una “llave de aplicación”	26
Configuración de las credenciales de Spotify.....	26
Métodos para la manipulación de lista de reproducción y música	27
Guía de instalación.....	30
Puesta en marcha de la PC 1	30
Entorno Intel ® XDK IoT Edition	30
Puesta en marcha de la PC 2	35
Instalación de servidor LAMP en PC servidor	35
Instalación Node.js v0.10.36 en Ubuntu 15.04.....	49
Ejecución del proyecto final	50

Objetivos del proyecto

El proyecto consta de un control de acceso a una habitación “privada” donde solamente una persona puede estar dentro en un momento determinado impidiendo el acceso a cualquier otro usuario que desee ingresar. Dentro de la habitación, se controlaron variables como: iluminación, música, sistema de ventilación y calefacción.

Un usuario registrado podrá ingresar a la habitación poniendo su clave de acceso única a través de un teclado numérico, viendo las teclas apretadas en un display LCD.

También, a través de una página web, tendrá la posibilidad de modificar su perfil, permitiéndole acondicionar su ambiente en tiempo real.

Existe además, un usuario administrador, el cual tendrá la posibilidad de observar quien se encuentra dentro de la habitación en cualquier momento, los usuarios registrados en el sistema, y un histórico de accesos.

Se utilizará la placa de desarrollo Intel Galileo GEN1 para crear el servidor web y realizar el control de las variables antes mencionadas, utilizando como lenguaje de programación Node.JS. Con respecto a la base de datos donde se encuentran los usuarios registrados y las auditorías utilizadas para generar el histórico, la misma se encontrará en una PC conectada a la misma red. Se optó por realizar este esquema, ya que la memoria que posee la placa está acotada como máximo a 32Gb y puede resultar insuficiente. La computadora además, es la utilizada para realizar la reproducción de lista musical seleccionada por el usuario.

Diferencia entre lo propuesto y lo realizado

La propuesta entregada fue la siguiente:

“El objetivo del proyecto es realizar un control de acceso a una habitación y dependiendo del perfil de la persona se ambientará el lugar de acuerdo a las preferencias de la misma. Las variables a controlar son temperatura y la intensidad de la luz.

El sistema permitirá a través de la página web que los usuarios ingresen sus preferencias y, a partir de ahí, el sistema les otorgará una clave única para que puedan acceder al ambiente. Una vez que los usuarios tienen su clave propia, deberán ingresarlas en el teclado numérico para acceder a la habitación.”

En diferencia con respecto a lo realizado es que se incorporó la posibilidad de que el usuario pueda elegir entre un grupo de listas de reproducción. Cuando el usuario ingresa a la habitación, comienza la reproducción de la lista de canciones, dándole la opción de realizar el control a través de su perfil.

Se incorporó un usuario Administrador, el cual puede ver los usuarios registrados, el actual en la habitación y un histórico de usuarios que hayan ingresado. El administrador también es un usuario más, por lo que goza de todas las funcionalidades convencionales.

El acondicionamiento de la temperatura se simuló con un ventilador y con un LED indicador de calefacción.

Intel Galileo GEN1

Características principales

Galileo es un microcontrolador basado en el procesador Intel® Quark SoC X1000 con una arquitectura de 32-bit. Es la primera placa basada en la arquitectura Intel® diseñada para ser pin-compatible tanto en hardware como software con los shields Arduino UNO. Posee 12 pines digitales de entrada/salida (numerados del 2 al 13), los pines digitales AREF y GND; 6 entradas analógicas, una tira de pins hembra para tomar tensión (5V – 3.3V – GND), pines ICSP, y los pines 0 y 1 del puerto UART. Todos los pines se encuentran ubicados de la misma manera que el Arduino UNO R3.

Galileo está diseñada para soportar shields que operan tanto con 3.3V o 5V. La tensión principal en la que opera Galileo es 3.3V. Sin embargo, un jumper en la placa habilita la tensión 5V en los pines de entrada/salida. Esto provee soporte para los 5V utilizados por los shields UNO.

Galileo contiene sobre su placa: un slot mini-PCI Express, un puerto Ethernet de 100Mb, un slot para tarjeta Micro-SD, un puerto serie RS232, un puerto Host USB, un puerto Cliente USB, y una memoria flash NOR de 8Mbyte.

Node.JS

Node.js es un framework asíncrono con una arquitectura orientada a eventos. Funciona con un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al

cambio de contexto. Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes, en el que toda operación que realice entradas y salidas debe tener una función callback.

Node.js incorpora varios "módulos básicos" compilados en el propio binario, como por ejemplo el módulo de red, que proporciona una capa para programación de red asíncrona y otros módulos fundamentales, como por ejemplo Path, FileSystem, Buffer, Timers y Stream.

Pese a que los módulos pueden instalarse como archivos simples, normalmente se instalan utilizando el Node Package Manager (npm) que nos facilitará la compilación, instalación y actualización de módulos así como la gestión de las dependencias.

Este framework es el utilizado para programar la placa de desarrollo utilizando el entorno Intel XDK IoT Edición.

Paquetes externos utilizados

Los paquetes utilizados dentro de Node.JS son los siguientes:

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

- **Sleep**
 - Agrega las funciones de `sleep()` y `usleep()` al Node.js. La única desventaja que tiene es que estas funciones bloquearán la ejecución de todo el programa.
- **request**
 - Este módulo está diseñado para ser la manera más simple para hacer llamados http. Soporta HTTPS y sigue redirecciones de forma predeterminada.
- **Sequence**
 - Este módulo ejecuta código asíncronico en un orden cronológico.
- **jsupm_grove**
 - Es una librería de sensores basados en MRAA que permite usar varios sensores en plataformas como Intel Galileo.
- **memwatch**
 - Esta herramienta ayuda a detectar y buscar las “fugas” (leaks) de memoria en el código de Node.js.

Instalación de los paquetes

Los módulos con sus respectivas versiones se encuentran en el archivo Package.JSON. Cuando el proyecto es subido a la placa Intel Galileo, se realiza la instalación de estos módulos mediante la ejecución del comando “`npm install`”. Esta instrucción busca la existencia del archivo Package.JSON e instala todos los módulos definidos en las opciones de dependencia (**Figura 1**). Para que pueda llevarse a cabo esta acción, la placa debe estar conectada a Internet.

package.json

```
{
  "name": "blankapp",
  "description": "",
  "version": "0.0.0",
  "main": "main.js",
  "engines": {
    "node": ">=0.10.0"
  },
  "dependencies": {
    "body-parser": "^1.14.1",
    "ejs": "^2.3.4",
    "emailjs": "^1.0.0",
    "express": "^4.13.3",
    "mysql": "^2.9.0",
    "sleep": "latest",
    "request": "latest",
    "sequence": "^3.0.0",
    "jsupm_grove": "latest",
    "memwatch": "latest"
  }
}
```

Figura 1. Dependencias del proyecto.

Luego de hacer la instalación de los módulos, se generará una nueva carpeta con el nombre “node_modules” donde serán instaladas todas las dependencias del proyecto. Este proceso demora varios minutos debido a que son muchos módulos que se deben descargar e instalar.

Aplicación - Control de acceso y variables de ambiente

A continuación, se explicarán las distintas secciones que forman la aplicación desarrollada.

Comunicación entre Node.js, programa en lenguaje C y el usuario

Con el esquema de la **Figura 2** se quiere ilustrar cómo son las comunicaciones entre el programa principal de Node.js, el lenguaje en C y el usuario al momento del ingreso a la habitación.

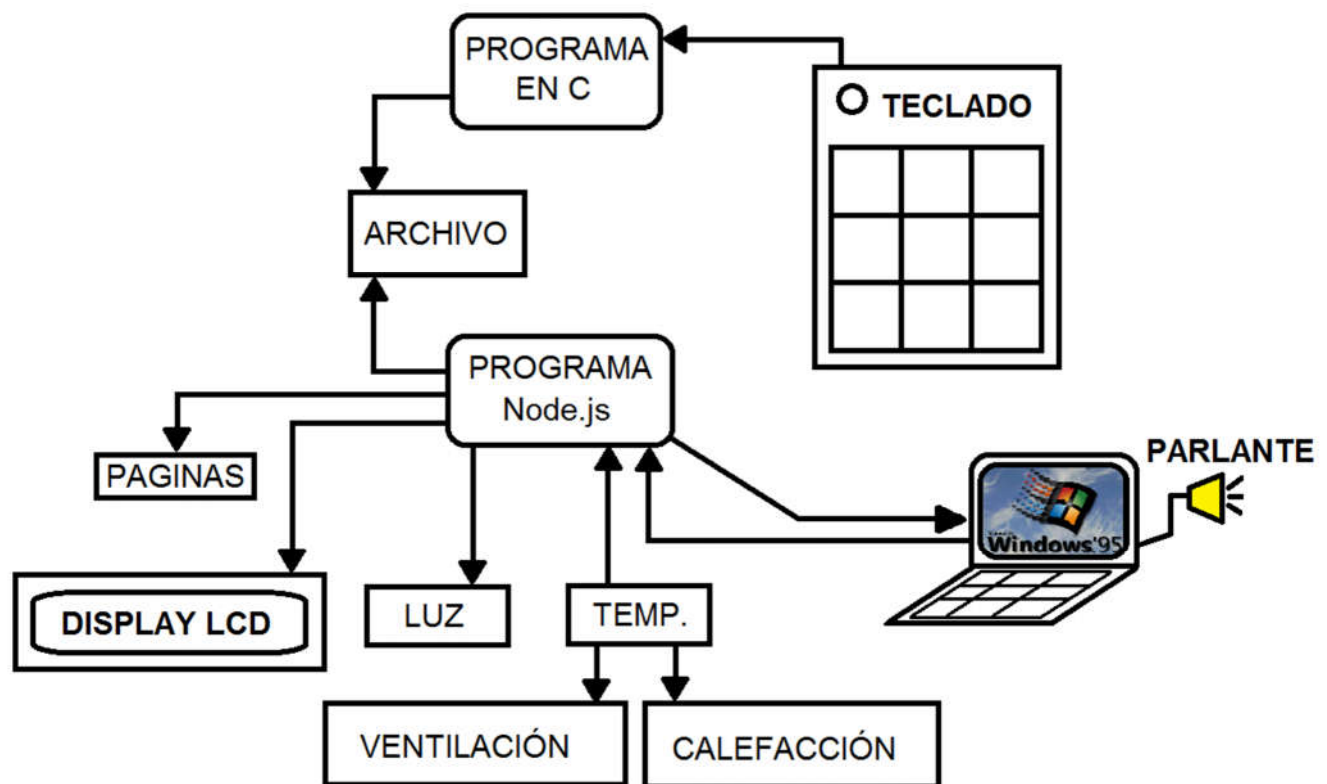


Figura 2. Circuito de comunicaciones entre usuario, Node.js y programa en C.

El circuito de comunicaciones es asimétrico por lo que, los usuarios pueden usar el sistema web mientras que otro puede ingresar a la habitación al mismo tiempo. Para que el usuario ingrese a la habitación, se debe acercar hasta el lugar donde se encuentra el teclado numérico e ingresar su clave única para que la aplicación le ambiente la temperatura, le acondicione la luz y comience a reproducir la lista de reproducción seleccionada. Los valores del teclado son capturados por el

programa en lenguaje C y son escritos en un archivo de texto que comparte con el Node.js. Cuando el Node.js detecta que hay nuevos caracteres, los imprime en el display LCD para hacerle saber al usuario de la tecla que presionó. Cuando el usuario presiona la tecla “Enter”, el programa Node.js limpia el archivo y realiza una búsqueda en la base de datos de la PC externa para saber si la clave corresponde a un usuario. La PC externa le responde con el usuario, en caso de que este existiera, sino le retorna vacío. En este momento, esta condición es visualizada en el display ya sea para un mensaje de bienvenida o para informarle que la contraseña ingresada era incorrecta.

Si el usuario introdujo la contraseña de manera satisfactoria, se acondicionará la habitación acorde a las variables que se habían seleccionado. Si el usuario, además de ingresar a la habitación, accede a su perfil a través de las páginas web, podrá cambiar sus variables como así controlar la reproducción de las canciones. Si el usuario cambia alguna de sus preferencias, éstas se verán reflejadas en tiempo real ya que el programa en Node.js actualizará la base de datos y tomará las decisiones adecuadas para el usuario. Por ejemplo, si cambia la lista de reproducción, la música cambiará, lo mismo con la temperatura y la luz.

Sistema WEB

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

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.

```
var express = require("express");
var app = express();

var port = process.env.PORT || 5000;
app.listen(port, function() {
  console.log("Listening on " + port);
});
```

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

```
var bodyParser = require('body-parser');
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({
  extend: true
}));
```

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

```
app.set('view options', { layout: false });
app.set('view engine', 'ejs');
```

Páginas WEB

El sistema cuenta con cuatro páginas web. A continuación se detalla la funcionalidad de cada una de ellas:

Página de Inicio (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.

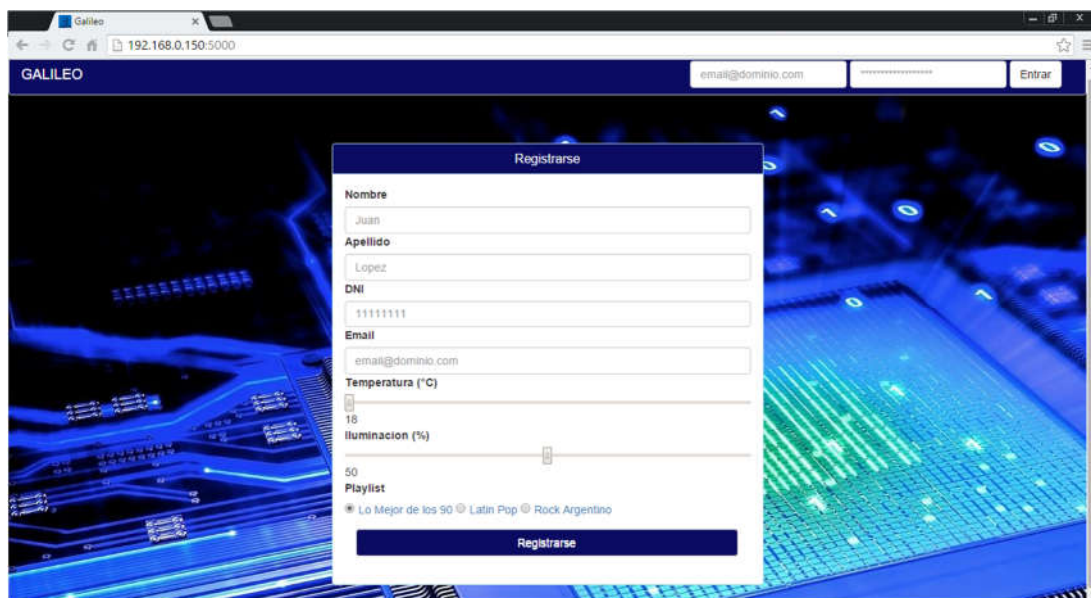
El formulario de registro, permite establecer además de los datos personales, el valor de temperatura, luz y lista de reproducción con lo que le gustaría que se acondicione el ambiente. Una vez registrado, se le enviará por mail la clave ÚNICA de acceso al sistema. Esta clave la podrá utilizar para ingresar 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 desde un ordenador se verá de la siguiente manera:

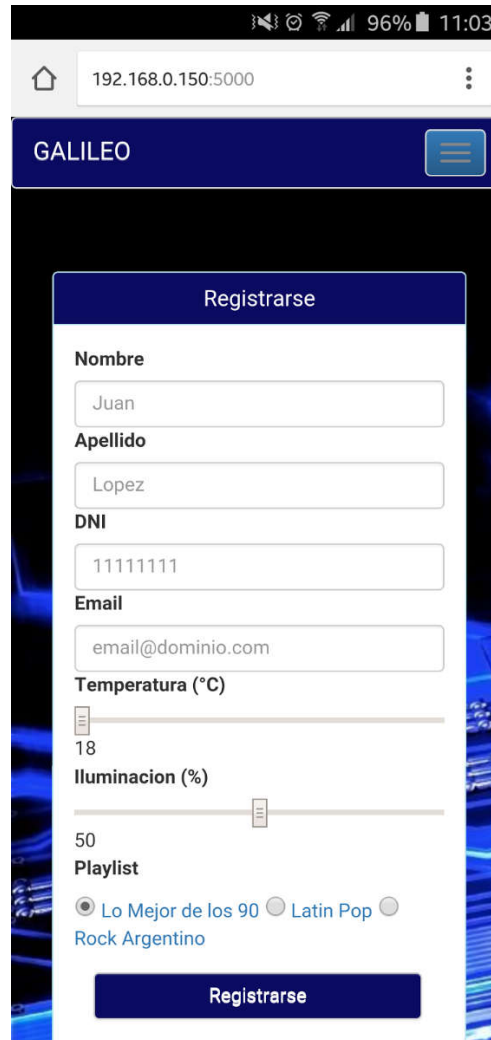


The screenshot shows a web browser window with the address bar displaying '192.168.0.150:5000'. The page title is 'GALILEO'. The main content area features a registration form titled 'Regístrate' with the following fields and controls:

- Nombre:** Input field with 'Juan' entered.
- Apellido:** Input field with 'Lopez' entered.
- DNI:** Input field with '11111111' entered.
- Email:** Input field with 'email@dominio.com' entered.
- Temperatura (°C):** A slider control with a value of 18.
- Iluminación (%):** A slider control with a value of 50.
- Playlist:** A dropdown menu showing 'Lo Mejor de los 90', 'Latin Pop', and 'Rock Argentino'.
- Regístrate:** A blue button at the bottom of the form.

The background of the page is a dark blue abstract image with glowing circuit lines and binary code.

Mientras que desde un teléfono celular se verá así:



The screenshot shows a mobile application interface with a dark blue header labeled 'GALILEO'. Below the header is a registration form titled 'Registrarse'. The form contains the following fields and controls:

- Nombre:** Text input field with the value 'Juan'.
- Apellido:** Text input field with the value 'Lopez'.
- DNI:** Text input field with the value '11111111'.
- Email:** Text input field with the value 'email@dominio.com'.
- Temperatura (°C):** A slider control with a value of 18.
- Iluminacion (%):** A slider control with a value of 50.
- Playlist:** Three radio button options: 'Lo Mejor de los 90' (selected), 'Latin Pop', and 'Rock Argentino'.

At the bottom of the form is a dark blue button labeled 'Registrarse'.

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.

```
app.get("/", function (req, res) {  
  res.render(appDir + '/inicio.ejs', {errorMessage: "", errorMessageRegister: "", successMessageRegister: ""});  
});
```

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:

```
var path = require('path');  
var appDir = path.dirname(require.main.filename);
```

- 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 módulo body-parser apuntando al parámetro contenido en el cuerpo del requerimiento. Si se desea obtener el dato “parametro” que proviene en un formulario se debe hacer con la siguiente línea de código:

- req.body.parametro.
- Asegurarse que el atributo “name” del input/checkbox/radiobutton, etc que se desea obtener tenga el nombre “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”.

```
<form class="form-horizontal" role="form" name="form1" method="post" action="/registro" target="_self" data-toggle="validator">
```

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

```
<label style="color:red"><%= errorMessageRegister %></label>  
<label style="color:green"><%= successMessageRegister %></label>
```

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. La password será establecida por defecto, mientras que el email será recuperado de la base de datos.

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.

Perfil de usuario (perfilUsuario.ejs)

El perfil de usuario, le mostrará a un usuario registrado todos sus datos y le dará la posibilidad de modificarlos.

El requerimiento “/log”, en caso de que el usuario exista en la base de datos, redireccionará al “/perfilUsuario.ejs” y además, renderizará esta página con los datos del mismo.

Teniendo en cuenta que el usuario puede estar dentro de la habitación o no, se visualizarán o ocultarán los botones para controlar la música del lugar. Por lo tanto, existen dos páginas de usuario posibles:

- Página sin control de música

The screenshot shows a web browser window with the address bar displaying "192.168.0.150:5000/log". The page title is "GALILEO". A "Cerrar sesión" button is in the top right corner. The main content area features a user profile form titled "Usuario Galileo" with the following fields and values:

Usuario Galileo	
Nombre:	Gaston Ariel
Apellido:	Maron
DNI:	36996998
Email:	gastonmaron@gmail.com
Contraseña:	6465
Temperatura seleccionada:	34
Intensidad de luz:	80
Playlist 1:	Lo Mejor de los 90
Playlist 2:	Latin Pop
Playlist 3:	Rock Argentino
Modificar perfil	

- Página con control de música

The screenshot shows the same web application but with a music control interface at the top. The header bar contains the text "GALILEO" and four navigation icons: a double left arrow, a double right arrow, a pause icon, and a single right arrow. Below the header, the user profile form "Usuario Galileo" is displayed with the following fields and values:

Usuario Galileo	
Nombre:	Gaston Ariel
Apellido:	Maron
DNI:	36996998
Email:	gastonmaron@gmail.c
Contraseña:	6465

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

- Cerrar sesión

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

- Modificar perfil

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

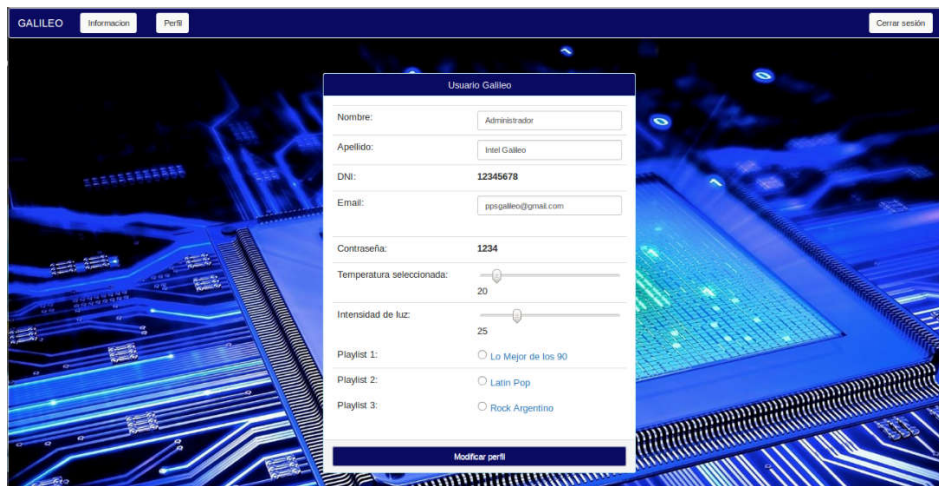
Este método tomará todos los datos del formulario, chequeara que el email ingresado sea el suyo o en caso de haberlo cambiado que no exista en la base de datos. De ser así, modificará el perfil y re direccionará a la página “perfilUsuario.ejs” completando el formulario con los nuevos datos. Si el email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

- Control de música (en caso que pudiera realizarlo)
 - Al presionar cada uno de los botones, se produce una petición desde la página al programa a través de un método “post”, el cual generará un requerimiento “http” al ordenador donde se encuentra la base de datos, pidiendo que realice la tarea correspondiente.

Perfil del Administrador (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: “Información” y “Perfil”. Además, en el caso de que el administrador ingrese a la habitación, éste puede controlar las canciones de la playlist que seleccionó, es decir, que tiene los controles de música a su disposición. Otra aclaración importante es que el administrador sólo podrá utilizar dichos controles en la página de “Perfil” y no en la de “Información”.

La página se ve de la siguiente manera:



The screenshot shows a web application interface for a user named 'Galileo'. At the top, there are tabs for 'Información' and 'Perfil', with 'Perfil' being the active tab. A 'Cerrar sesión' button is in the top right corner. The main content area is titled 'Usuario Galileo' and contains a form with the following fields and values:

- Nombre: Administrador
- Apellido: Intel Galileo
- DNI: 12345678
- Email: pps-galileo@gmail.com
- Contraseña: 1234
- Temperatura seleccionada: 20 (with a slider)
- Intensidad de luz: 25 (with a slider)
- Playlist 1: ☐ Lo Mejor de los 90
- Playlist 2: ☐ Latin Pop
- Playlist 3: ☐ Rock Argentino

At the bottom of the form is a button labeled 'Modificar perfil'.

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

- Los requerimientos “Modificar perfil” y “Cerrar sesion”
 - Los mismos que en el caso de “perfilUsuario.ejs”.
- Información
 - Mostrará la información relevante para el administrador.
 - Usuario actual (si es que lo hay).
 - Usuarios registrados.
 - Histórico de accesos (Auditorias).

En el caso del botón “Información”, producirá un requerimiento llamado “/histórico” mediante el método “post”. Este método nos redireccionará a una página llamada “historicoAdministrador.ejs”.

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 todavía no salió de la habitación. Los datos son enviados a la página WEB en formato “JSON”. En esta página hay un template “ejs” el cual se muestra al usuario actual (si es que hay uno dentro de la habitación) y se crea las dos tablas correspondientes a los usuarios registrados y a las auditorías.

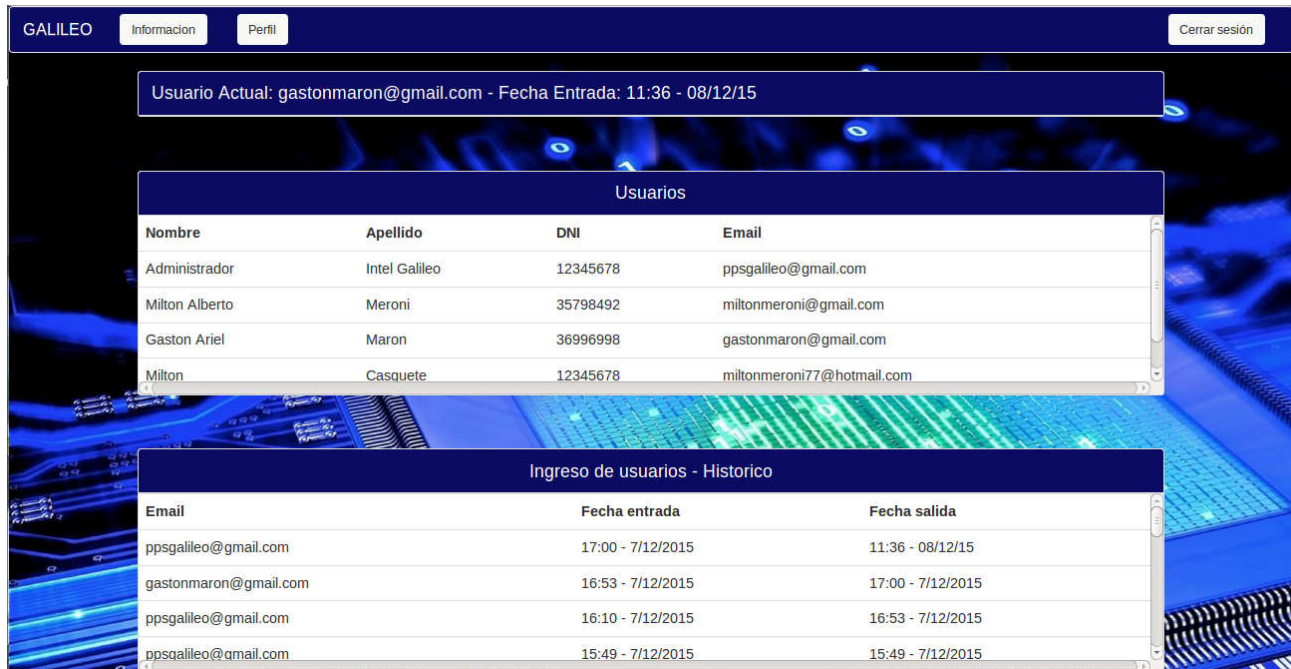
- Perfil

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”. Si el administrador se encuentra dentro de la habitación, cuando haga click en el botón “Perfil”, le aparecerán los botones para poder controlar las canciones.

Información de administrador (historicoAdministrador.ejs)

Esta página mostrará en forma de tablas información sobre los usuarios registrados, los accesos históricos y los actuales.

La página es la siguiente:



Usuarios			
Nombre	Apellido	DNI	Email
Administrador	Intel Galileo	12345678	ppsgalileo@gmail.com
Milton Alberto	Meroni	35798492	miltonmeroni@gmail.com
Gaston Ariel	Maron	36996998	gastonmaron@gmail.com
Milton	Casquete	12345678	miltonmeroni77@hotmail.com

Ingreso de usuarios - Historico		
Email	Fecha entrada	Fecha salida
ppsgalileo@gmail.com	17:00 - 7/12/2015	11:36 - 08/12/15
gastonmaron@gmail.com	16:53 - 7/12/2015	17:00 - 7/12/2015
ppsgalileo@gmail.com	16:10 - 7/12/2015	16:53 - 7/12/2015
ppsgalileo@gmail.com	15:49 - 7/12/2015	15:49 - 7/12/2015

Los requerimientos generados al programa Node.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

El envío de emails será necesario al realizar el registro de usuario. Para realizar esta operación a través de Node.js, 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.

```
var email = require("emailjs");
var serverEmail = email.server.connect({
  user: "ppsgalileo@gmail.com",
  password: "ingenieriaencomputacion",
  host: "smtp.gmail.com",
  ssl: true
});
```

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.

```
function sendEmail(email,nombre,pass){
  serverEmail.send({
    text: "Sr/a. "+nombre+" \n \nSu clave de acceso es: "+pass+"\n \nSaludos,\n \nGalileo",
    from: "Galileo <ppsgalileo@gmail.com>",
    to: nombre+" "+email,
    //cc: "else <else@your-email.com>",
    subject: "Clave de acceso - Galileo"
  }, function(err, message) { if (err!=null) console.log(err); });
}
```

El mensaje que se recibe es el siguiente:



Base de datos

Diseño

La Base de datos, tal como se mencionó anteriormente, se encuentra en un ordenador externo el cual está conectado a la misma red que Intel Galileo. Dicha base de datos 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)
idPlaylist	Identificación de la lista de reproducción elegida 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 auditoría	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)

Comunicación con Base de Datos

La creación de la conexión y finalización se realiza en todos los métodos de la misma manera. Será necesario conocer el “ip” y nombre de la base de datos, el nombre de usuario y password para acceder a la misma.

```
var mysql      = require('mysql');
var connection = mysql.createConnection({
  host      : ipDataBase,
  user      : usrDataBase,
  password  : passDataBase,
  database  : nameDataBase
});
connection.connect();

//se realiza la consulta
//correspondiente

connection.end();
```

A continuación, se detallaran todos los métodos creados junto a su correspondiente explicación.

Método	Función que realiza
saveUserDataBase(nombre,apellido,dni,email,pass,temp,luz,playlist)	Guarda un usuario en la base de datos.
updateUserDataBase(nombre,apellido,email,temp,luz,id,playlist)	Actualiza el usuario que tenga el “id” que proviene del parámetro.
recoveryAllUsers(callback)	Recupera todos los usuarios que están en la base de datos. En caso de que no haya ninguno o que se haya producido un error, retorna null.
recoveryUser(email,pass,callback)	Consulta si un usuario y su contraseña se encuentran en la base de datos. Si existe, devuelve el usuario; caso contrario, retorna null.
recoveryUserByEmail (email,callback)	Consulta si un usuario se encuentra en la base de datos discriminando por email. En caso de que exista, devuelve el usuario; caso contrario retorna null.
recoveryUserByPass (pass,callback)	Consulta si un usuario se encuentra en la base de datos discriminando por contraseña. En caso de que exista, devuelve el usuario; caso contrario retorna null.
saveAuditoriaDataBase (email)	Genera una nueva auditoría y la guarda en la base de datos.
recoveryAuditoriaNULL(callback)	Retorna la auditoría que contenga a la fecha de salida como NULL. Esto quiere decir que hay un

	usuario dentro de la habitación. Retorna la auditoría en caso de que haya una; caso contrario, retorna null.
recoveryAllAuditoria (callback)	Retorna todas las auditorías que se encuentren en la base de datos
updateAuditoriaDataBase(email)	Actualiza a la auditoría que contenga la fecha de salida como NULL y que el email sea el mismo que el que proviene del parámetro de la función.

Programa en lenguaje C

El programa “teclado.c” captura los eventos provenientes del teclado y los escribe en un archivo de texto. Este archivo se debe encontrar en el mismo directorio que la aplicación.

El programa en Lenguaje C “evtest.c” realiza la impresión de las capacidades en los dispositivos del kernel en /dev/input/eventX y sus eventos. El objetivo del programa “teclado.c” es modificar el programa mencionado anteriormente para la obtención de los eventos de un teclado numérico y la discriminación de 11 teclas (los números de 0 a 9 y la tecla “Enter”). Tanto los números como la palabra “Enter” son escritos en el archivo “teclas.txt”, estableciendo esta última un protocolo de comunicación entre el Programa principal del Node.js y el programa en Lenguaje C. Cuando el programa del Node.js detecta que está la palabra clave “Enter”, borra todo el contenido del archivo de texto y analiza los números ingresados, mientras que el programa en lenguaje C, se posiciona al inicio del archivo para nuevamente escribir los caracteres numéricos ingresados.

Detección del evento apropiado

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:

- `var ejecutarTeclas = "cd "+appDir+"; "+gcc+";"+perm+"; ./teclado "+evento;`
 - Ejecución de todos los comandos de manera secuencial.

Finalmente, se ejecutan todos los comandos de la variable “ejecutarTeclas”.

Nota: El archivo “teclas.txt” debe estar en el mismo directorio que el main.js y el programa en lenguaje C compilado.

Sensores y actuadores

Los dispositivos utilizados son:

- Sensor de temperatura
 - Este sensor es del “Grove Starter Kit”. Se conectó al pin A0 (Analógico). El cálculo realizado para tomar el valor de temperatura es el predefinido por Intel.
- Cooler con relay
 - El cooler simulará el sistema de ventilación de la habitación. Su control se realiza por el pin D6 (Digital) a través de un relay. Se encuentra conectado al terminal “Normalmente abierto”.
 - El pin D6 se configuró como salida y, escribiendo un “1”, se abre la llave (apaga el cooler), mientras que con un “0” se cierra (enciende el cooler).
 - La acción de encendido o apagado se tomará a partir de la comparación entre la temperatura actual en la habitación y la temperatura elegida por el usuario.
- LED Rojo (calefacción) con relay
 - Este LED indicará si la calefacción está encendida o apagada. Su control se realiza por el pin D4 (Digital) a través de un relay. Se encuentra conectado al terminal “Normalmente abierto”.
 - Para realizar el control del relay, teniendo en cuenta que en este caso es el relay del “Grove Starter Kit”, se pudo utilizar las funciones propias del mismo. Las acciones de encendido y apagado se simplificaron a un simple llamando a la función “on()” y “off()” respectivamente.
- LED Verde (luz ambiente)
 - El control de la intensidad de la luz ambiente se realizó mediante una señal PWM. Esto pudo realizarse a través del pin D3, el cual permite configurar este tipo de

señales. Será necesario habilitar el pin, para luego, establecer la intensidad escogida por el usuario, escribiendo un valor entre 0.0 (0%) y 1.0 (100%).

- Display LCD
 - A través del display, se podrá visualizar si la habitación está libre o ocupada. Además, le permitirá al usuario observar la clave que está ingresando. El control del display se realizó a través del paquete “jsupm_i2clcd”.
 - Se crearon funciones para simplificar la escritura en el mismo:
 - cleanLCD: limpia las dos líneas del display.
 - writeLCD(dato,fil,col): escribe el “dato” en la línea y columna enviada como parametro.
 - mensajeLCDconDelay(msj,msj2): escribe y mantiene por dos segundos el string “msj” en la primera línea del display y “msj2” en la segunda línea.
- Teclado numérico USB
 - Se encuentra conectado al puerto USB-Host de la placa.

Esquema de conexiones

En la **Figura 3**, se muestra un esquema de todas las conexiones de los distintos sensores y actuadores. Se destaca que el Shield está puesto en la placa Intel Galileo y que el teclado numérico es conectado, como se mencionó anteriormente, al USB Host de la placa.

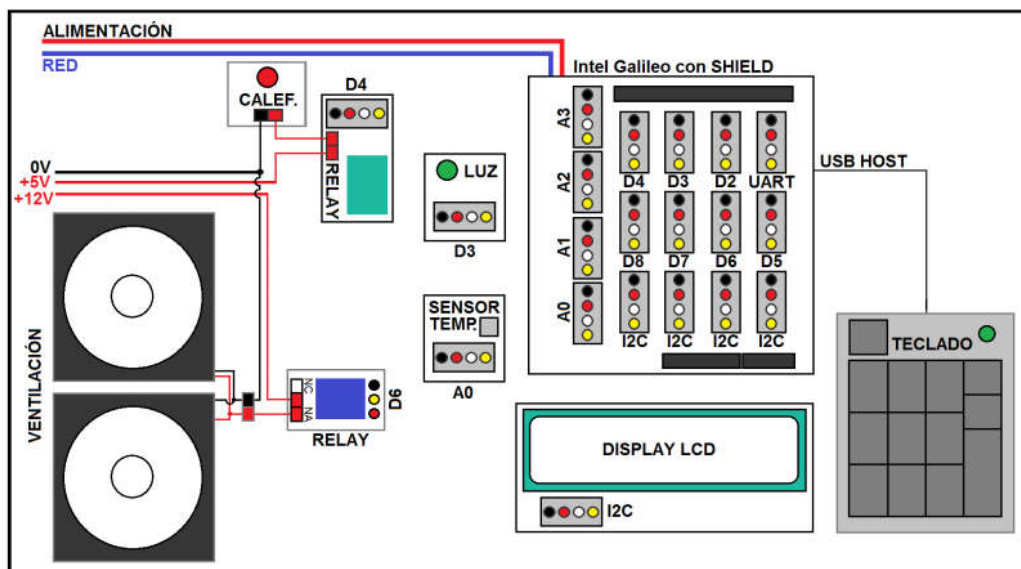


Figura 3. Esquema de conexión de los sensores y actuadores.

Funciones principales

Función “iniciarIntelGalileo()”

Para poder realizar un correcto inicio o reinicio del programa, se decidió agregar esta función para que prepare a la placa para su funcionamiento. Esto se logra con una consulta a la base de datos para saber si hay un usuario dentro de la habitación. En el caso de que no haya nadie en la habitación se inicializan los sensores apagados y el display LCD mostrará el mensaje de “Libre”. Por otro lado, si había un usuario al momento de que la placa se apagó, la Intel Galileo acondicionará la habitación para dicho usuario y en el display LCD se verá un mensaje de “Ocupado”. Una vez tomada la acción de acondicionar o de iniciar, se pone en marcha el sistema Web y la función principal “main()”.

Función “main()”

Esta función es llamada desde iniciarIntelGalileo(). Su tarea es la de, cada 200ms, abrir el archivo “teclas.txt” para ver las teclas presionadas del teclado numérico y buscar si se encuentra la palabra clave “Enter”. Si se encuentra, quiere decir que el usuario ha ingresado un conjunto de números que representarían su contraseña. En ese momento, se borra todo el contenido del archivo de texto “teclas.txt”, y se realizan las siguientes acciones:

- Si hay un usuario dentro y la contraseña ingresada es la misma que el residente de la habitación
 - Se apagan los sensores (se “destruye” la función “simuladorSensores()”), la música se detiene y se le escribe un mensaje de despedida.
- Si no hay un usuario dentro de la habitación, se verifica que la contraseña ingresada pertenezca a un usuario registrado.
 - Si pertenece a un usuario, se le acondicionará la habitación (Controlando la temperatura, la intensidad de la luz, mediante la función “simuladorSensores()”) y se comenzará a reproducir la lista de reproducción.
 - Si no pertenece a ningún usuario, se le mostrará un mensaje de contraseña incorrecta.
- Si un usuario quiere ingresar a la habitación, pero ya hay uno adentro, se le mostrará el mensaje de “Habitación ocupada” en el display LCD.

Si nadie aprieta la tecla “Enter”, el display LCD visualizará el estado actual de la habitación, es decir, si está “Libre” u “Ocupada”.

Función “simuladorSensores”

Esta función se encargará de realizar el acondicionamiento y control de temperatura (cooler y calefacción) y luz de la habitación. Recibe como parámetro la password del usuario que se encuentra dentro, para a partir de allí, obtener los datos establecidos por el mismo a través de una consulta a la BD.

En cuanto a los valores de “luz”, en la base de datos se encuentran cargados de “0” a “100”, por lo que será necesario dividirlo por “100” para llevarlo a los valores recibidos por la señal PWM que la representa.

Dentro de esta función, se realiza el llamado a una función periódica llamada “intervalSensores”. Esta última, se ejecuta cada 1000 ms (un segundo), y encenderá o apagará el cooler o calefacción dependiendo del valor de temperatura actual y la establecida por el usuario. Además, en caso de que el usuario haya cambiado su perfil, accederá nuevamente a la base de datos para obtener los nuevos parámetros y reacondicionará el ambiente.

PC Servidor

La computadora servidor cuenta con un sistema operativo Ubuntu 15.04. En esta se encontrará la base de datos, la cual se encuentra sobre un servidor LAMP y se administra a través de phpMyAdmin (Ver Guía de instalación - Puesta en marcha de la PC 2). Además, se encargará de correr la aplicación que controlará la reproducción de la lista de canciones elegida por el usuario.

Aplicación Spotify

Para la incorporación de las listas de reproducción, se decidió utilizar las herramientas de desarrollo de Spotify. Con estas herramientas, se puede crear un reproductor básico de música con listas de reproducciones elegidas por el usuario de la cuenta Spotify.

Aclaraciones:

- Para utilizar estas herramientas, se debe tener una cuenta “**Spotify Premium**”

- La versión del Node.js DEBE ser la **v0.10.x**. (Ver Guía de instalación - Puesta en marcha de la PC 2).

Instalación del módulo node-spotify

Para la instalación del módulo no se utilizará el NPM, sino que se descargará el archivo comprimido directamente desde la página. Para ello se deben seguir los pasos siguientes:

1. Crear una carpeta con el nombre “Spotify” y acceder a ella
 - a. Comando: `mkdir Spotify`
 - b. Comando: `cd Spotify`
2. Descargar el módulo node-spotify
 - a. comando: `wget http://www.node-spotify.com/builds/0.7.0/node-spotify-0.7.0-linux.zip`
3. Descomprimir el archivo “.zip”
 - a. comando: `unzip node-spotify-0.7.0-linux`

A partir de aquí para acceder al módulo, en el programa Node.js, se debe direccionar a la carpeta que generó la descompresión del archivo descargado.

Instalación de la librería “libspotify”

Esta librería se utiliza para poder utilizar las librerías de Spotify en el Node.js. Su instalación se detalla en los siguientes pasos. Se debe abrir una terminal del Linux Ubuntu.

1. Dirigirse al directorio “Spotify” creado en la sección anterior
 - a. `cd Spotify`
2. Obtener la librería “libspotify”
 - a. `Wget https://developer.spotify.com/download/libspotify/libspotify-12.1.51-Linux-x86_64-release.tar.gz`
3. Descomprimir el archivo
 - a. `tar -xvzf libspotify-12.1.51-Linux-x86_64-release.tar.gz`
4. Instalar la librería
 - a. `make install prefix=/usr/local`

Generación de una “llave de aplicación”

La denominada “llave de aplicación” sirve como autenticación para poder utilizar los servicios de Spotify en una aplicación externa. Para obtenerla, se debe ingresar a la página <https://developer.spotify.com/> y poner los datos de usuario y contraseña correspondientes al usuario “Premium”. Luego, se debe crear una aplicación (haciendo click en el botón “Create an app” que se encuentra en el margen superior de la página <https://developer.spotify.com/my-applications/#!/applications> para la utilización de las bibliotecas de Spotify.

My Applications

CREATE AN APP



TallerDeProyectoll

It's a Engineering in Computation's project from Facultad de Informatica - Universidad Nacional de La Plata

Una vez que se creó la aplicación, se debe ingresar a la página (<https://devaccount.spotify.com/my-account/keys/>) que contiene la “llave de aplicación” para la aplicación creada. Este sitio permite la descarga de la “llave” en formato binario o el código en C. Se debe descargar el archivo binario y guardarlo en la carpeta “Spotify” creada en la instalación del módulo “node-spotify”

Configuración de las credenciales de Spotify

Para poder comenzar a utilizar los servicios de Spotify, se debe crear una variable que contendrá las opciones para la ejecución de la aplicación, tales como la carpeta donde se guardarán los datos “Cache”, los ajustes y el archivo “llave de aplicación”.

```
var options = {  
  appkeyFile: './spotify_appkey.key',  
  cacheFolder: 'cache',  
  settingsFolder: 'settings'  
};
```

Luego, para poder utilizar el Spotify, se declaró una variable que realiza la inclusión del módulo y le incorpora las opciones. Esto se lleva a cabo con la siguiente línea de código:

```
var spotify = require('/node-spotify-0.7.0-linux/spotify')(options);
```

De ahora en más la variable “spotify” contendrá todas las funciones del node-spotify para la manipulación de las canciones. Como por ejemplo, el inicio de sesión, el cual es **OBLIGATORIO** para poder acceder a las canciones y listas de reproducciones del usuario con cuenta “Premium”.

```
spotify.login('usuarioPremium', 'contraseñaPreimum', false, false);
```

Métodos para la manipulación de lista de reproducción y música

Para que esta aplicación se pueda comunicar con la placa Intel Galileo, se decidió abrir un socket en el puerto 8000 para que la placa pueda enviarle peticiones. Esto se realizó de la misma manera que el Servidor Web, es decir, a través del módulo “Express”, el cual se instaló mediante el NPM con el comando:

- npm install express
 - Siempre en el mismo directorio que la aplicación

Se definieron siete métodos por “GET” para la comunicación entre la placa y la aplicación. Los parámetros son enviados como un “QUERY_STRING”, es decir, el programa de la Intel Galileo, para enviarle una petición tiene que hacer un requerimiento al puerto 8000 de la placa con el nombre del método seguido por el parámetro (en caso de ser necesario). Por ejemplo:

- musicOn?num=3
 - El método /musicOn, obtendrá del query_string el número de la playlist enviada.

Los métodos son:

- /musicOn
 - Parámetro
 - num: número de la Playlist a reproducir
 - Acción
 - Comenzar la reproducción de las canciones correspondientes a la Playlist elegida
 - Respuesta
 - vacío
- /musicOff

- Parámetro
 - Vacío
 - Acción
 - Detener la canción actual
 - Respuesta
 - Vacío
- /changePlayList
 - Parámetro
 - num: número de la Playlist a reproducir
 - Acción
 - Detener la lista de reproducción actual y comenzar la reproducción de las canciones correspondientes a la Playlist elegida
 - Respuesta
 - Vacío
- /damePlaylist
 - Parámetro
 - num: número de la Playlist
 - Acción
 - Obtener todas las canciones pertenecientes al número de la playlist enviada
 - Respuesta
 - Todas las canciones correspondientes a la playlist enviada
 - Formato: “nombre de la cancion - Artista”
- /dameNombrePlaylist
 - Parámetro
 - num: número de la Playlist
 - Acción
 - Obtener el nombre de la Playlist enviada
 - Respuesta
 - Nombre de la Playlist

- /pauseMusic
 - Parámetro
 - Vacío
 - Acción
 - Pone en pausa a la canción actual
 - Respuesta
 - vacío
- /resumeMusic
 - Parámetro
 - Vacío
 - Acción
 - Reanuda la canción actual
 - Respuesta
 - Vacío
- /changeMusicBack
 - Parámetro
 - Vacío
 - Acción
 - Cambia a la canción anterior. Si es la canción con índice 0, reproduce la última canción en la lista de reproducción actual.
 - Respuesta
 - Vacío
- /changeMusicNext
 - Parámetro
 - Vacío
 - Acción
 - Cambia a la canción siguiente. Si es la última canción, reproduce la primera canción en la lista de reproducción actual.
 - Respuesta
 - Vacío

Guía de instalación

En este apartado se explicarán todos los pasos para que el proyecto pueda ser reproducido. Se necesitarán dos computadoras (PC 1: con Windows 10; PC 2: con Ubuntu 15.04) y la placa Intel Galileo, todos con acceso a internet. Esta guía se dividirá en tres grandes partes que son:

- Instalación del entorno Intel XDK IoT Edition en la PC 1. Descarga, importación y acondicionamiento del proyecto para la placa Intel Galileo.
- Instalación de un servidor LAMP en la PC 2. Descarga e importación de la base de datos. Instalación de Node.js v0.10.36, descarga y ejecución del servidor que realizará el control de la música.
- Ejecución del proyecto final.

Puesta en marcha de la PC 1

Entorno Intel ® XDK IoT Edition

El entorno de desarrollo Intel XDK IoT en una solución completa para crear y probar aplicaciones en las plataformas de Intel IoT. Este entorno provee componentes hardware y software para que los desarrolladores puedan crear proyectos usando la placa y el Grove – Starter Kit (conjunto de herramientas diseñadas para minimizar la dificultad de utilizar componentes electrónicos (Figura 4).

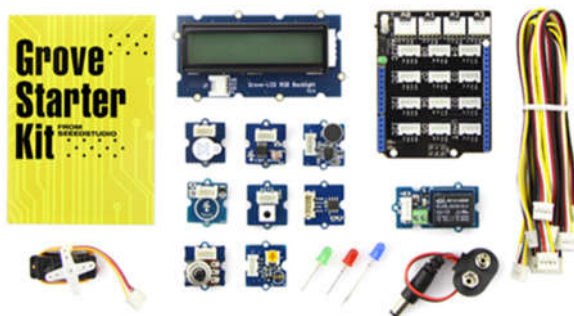


Figura 4. Grove Starter Kit con todos sus componentes.

El entorno de desarrollo permite programar en JavaScript, que está compuesto por tres partes principales: el Intel XDK IoT Edition, el sistema operativo Yocto Linux, y un conjunto de librerías. Estas librerías se crearon para proporcionar a los desarrolladores una manera fácil de controlar los componentes de hardware integrados en plataformas Intel IoT, así como el acceso de alto nivel a varios sensores y actuadores.

- El Linux Yocto es una distribución “meta” específicamente para proyectos embebidos. Es una distribución “meta” porque es más que una distribución Linux. Yocto es un entorno para la producción de sistemas operativos basados en Linux a medida. Además, proporciona herramientas para la selección de paquetes de software, la preparación de los ajustes y la creación de imágenes que se pueden utilizar para instalar este sistema operativo a medida. Cuando se compra una placa Intel Galileo, Yocto Linux es el sistema operativo disponible. Con el “Intel IoT Developer Kit”, otra versión es incluida, con muchas más librerías y recursos, para ofrecer a los desarrolladores más recursos para crear las aplicaciones en su lenguaje de programación favorito. Esta versión incluye compilador GCC, Python, Node.js y OpenCV, entre otros.
- El sistema Yocto instalado en la placa Intel Galileo provee un conjunto de librerías específicas especialmente diseñadas para el “Intel IoT Developer Kit” y el entorno de desarrollo Intel XDK IoT Edition. Estas librerías son las mencionadas MRAA y UPM.

Instalación de entorno de desarrollo Intel® XDK IoT Edition

1. Ingresar y descargar el instalador del Intel IoT XDK Edition¹ correspondiente al sistema operativo.
2. Abrir el instalador con permisos de administrador, si aparece un mensaje de confirmación hacer click en *yes*.
3. Instalar *Bonjour Print Services* (Solo Windows).
 - 3.1. Este programa habilita al *XDK IoT Edition* detectar automáticamente los dispositivos instalados en la red.
 - 3.2. Descargar *Bonjour*.²
 - 3.3. Click en *Download*.

¹ <https://mega.nz/#!r44jxaiL!0cDlvCO5DUIY72eWcj969ENfP1iv0DaPYckM8SBFB9s>

² <http://support.apple.com/kb/DL999>

- 3.4. Click derecho sobre *BounjourPSSetup.exe* y luego seleccionar **Ejecutar como administrador**. Si un mensaje de configuración aparece, hacer click en **YES** para continuar.
- 3.5. Seguir las instrucciones del asistente de instalación para instalar *Bonjour*.(Figura 5)



Figura 5. Instalador del “Bonjour Print Services”

Instrucciones para importar

Para poder ejecutar la aplicación se debe importar el proyecto desde un repositorio Github³. Se debe hacer click en el panel derecho donde dice “Download ZIP” (Figura6).

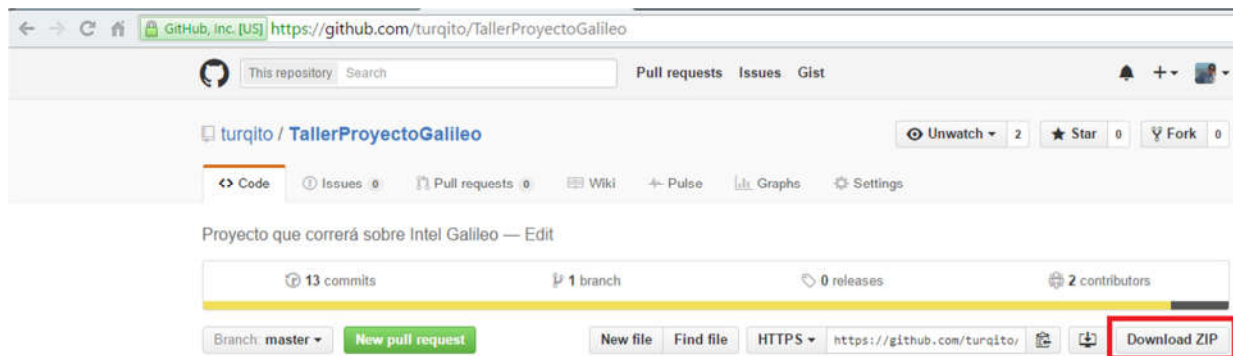


Figura 6. Repositorio donde se encuentra la aplicación

Luego, se debe extraer con la herramienta “7-zip”, instalada anteriormente. Una vez extraído debe aparecer una carpeta con los archivos del proyecto (Figura 7).

³ <https://github.com/turqito/TallerProyectoGalileo>

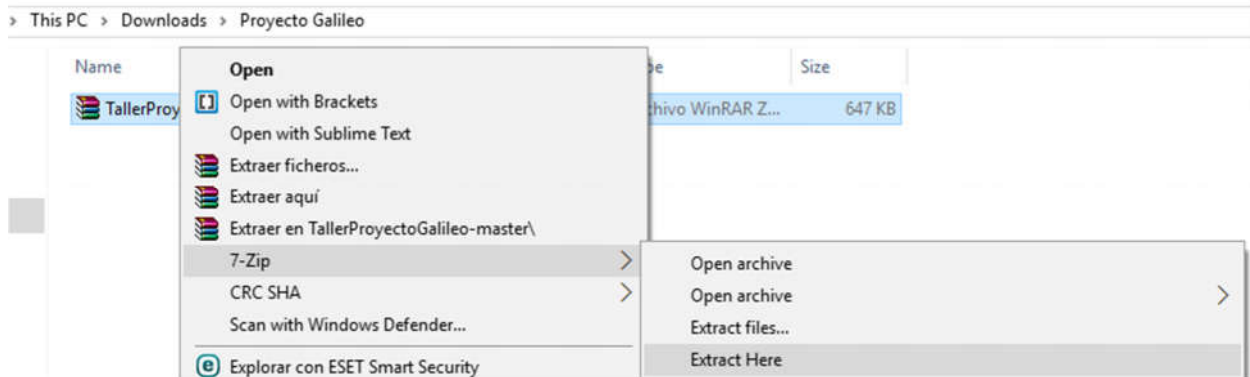


Figura 7. Extracción del proyecto

Para poder importar el proyecto, hay que abrir el Intel XDK IoT Edition y hacer click en “OPEN AN INTEL XDK PROJECT” y navegar hasta la carpeta donde se descomprimió el proyecto. Finalmente, seleccionar el proyecto “Practica2.xdk” y hacer click en “Open”. A partir de allí ya se tiene el proyecto cargado en el entorno (Figura 8).

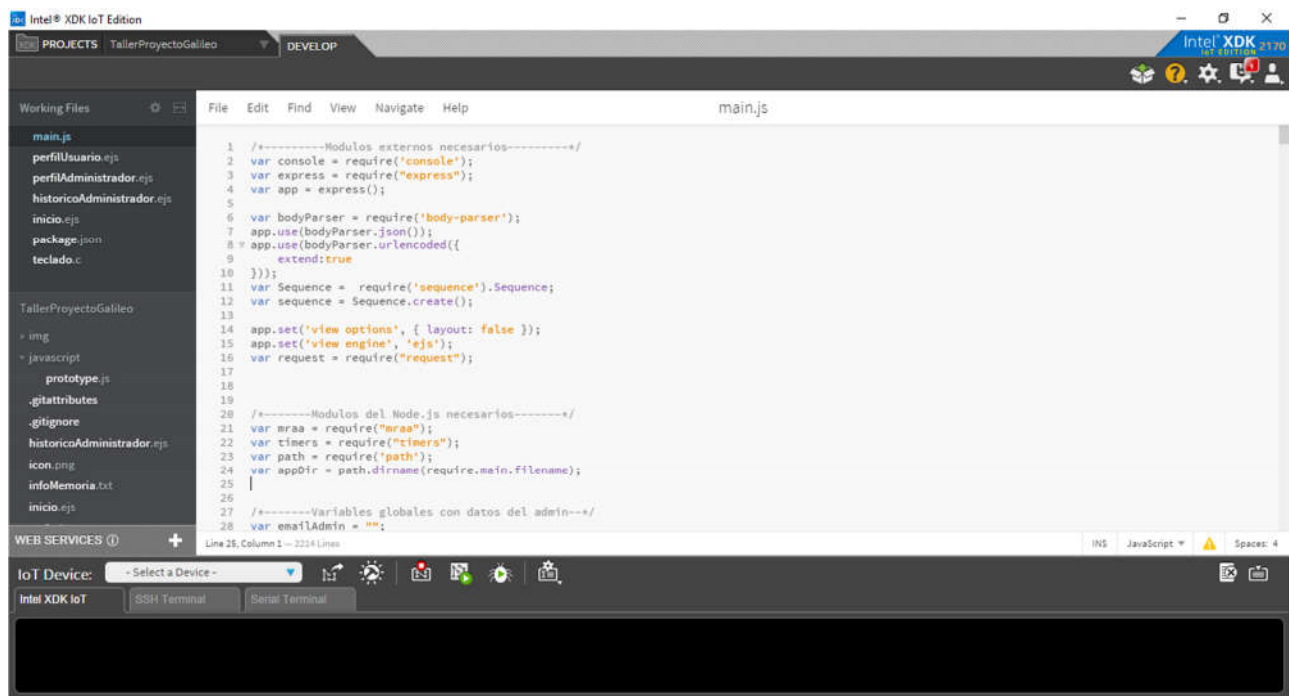


Figura 8. Entorno con el proyecto cargado.

Para el traspaso del código que se encuentra en el entorno a la placa hay que conectarse con la misma mediante Ethernet.

ACLARACIÓN: Es necesario que ambos dispositivos (PC 1 con entorno Intel XDK y placa Intel Galileo) estén conectados en la misma red.

Será necesario conectar el cable serie (Jack 3.5mm a USB) a la placa y, elegir en el menú contenedor Port en la esquina inferior izquierda de la pantalla, el dispositivo serie que representa la placa, además, se deberá configurar el Baud Rate en 115200, Data Bits en 8, Stop Bits en 1 y Parity en 0 (Figura 9).

Finalmente se presiona el botón Connect (se recomienda presionar el botón Reboot de la placa para que inicie el sistema operativo).



Figura 9. Configuración para la conexión serie.

En la terminal se verá el booteo del sistema operativo y la posibilidad de iniciar sesión con el usuario root.

Luego, simplemente se debe conectar un cable Ethernet (Rj-45) a la placa y el otro extremo a un access point disponible sin la necesidad de una configuración adicional.

Una vez conectado el cable, para ver la dirección IP asignada a la placa deberá ingresar el comando ifconfig (en la terminal) para luego realizar la conexión IoT (Figura 10).

```
root@galileo:~# ifconfig
enp0s20f6 Link encap:Ethernet  HWaddr 98:4F:EE:01:0E:77
        inet addr:192.168.0.117 Bcast:192.168.0.255 Mask:255.255.255.0
        inet6 addr: fe80::984f:eeff:fe01:e77/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:46801 errors:0 dropped:11 overruns:0 frame:0
        TX packets:14646 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:52940637 (50.4 MiB) TX bytes:1268945 (1.2 MiB)
        Interrupt:43
```

Figura 10. Obtención de la dirección IP de la placa.

Para cargar el programa se necesitará establecer una conexión con la placa a través de Ethernet. Para ello en el menú contenedor IoT Device se elige la opción “Add manual connection” que desplegará una nueva ventana como indica la Figura 11.

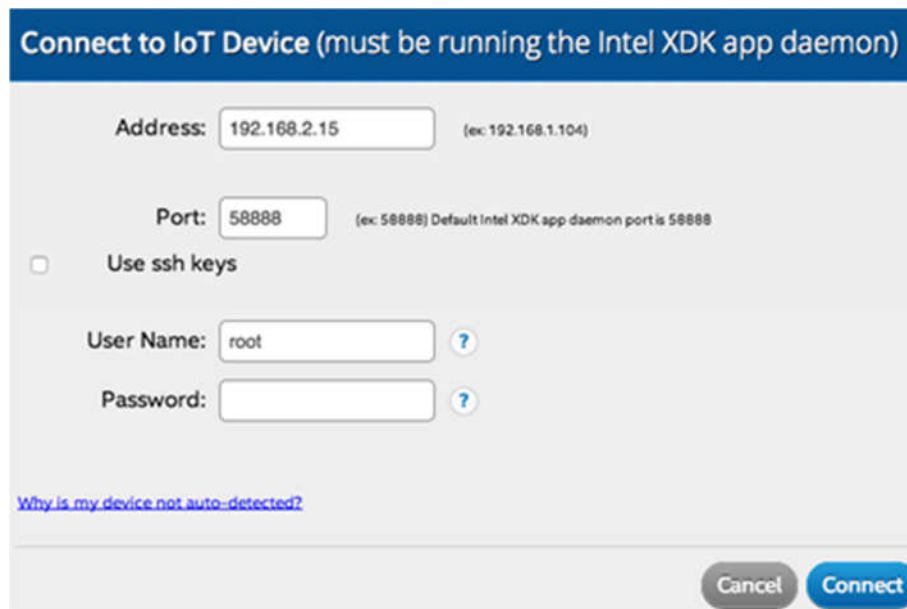


Figura 11. Configuración para la conexión con la placa.

Dentro del campo Address se coloca la IP previamente obtenida a través del comando ifconfig, luego en los campos User y Password se deberá ingresar el usuario y contraseña siendo el usuario default root y sin contraseña.

Al botón Connect aparecerá un cartel indicando si la conexión fue establecida (Figura 12).

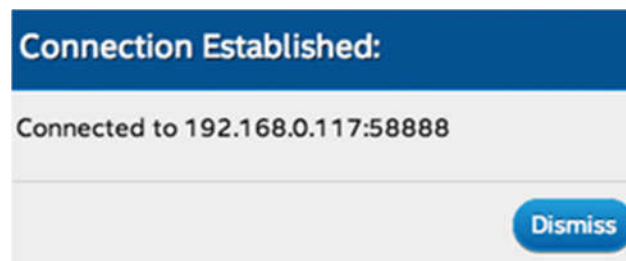


Figura 12. Mensaje de conexión establecida.

Puesta en marcha de la PC 2

Instalación de servidor LAMP en PC servidor

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 MySQL y el contenido dinámico es procesado con PHP.

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`

```
ubuntu@ubuntu:~$ sudo apt-get update
[sudo] password for ubuntu:
```

Actualiza los repositorios del Sistema Operativo.

- `sudo apt -get install apache2`

```
ubuntu@ubuntu:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0
0 upgraded, 9 newly installed, 0 to remove and 270 not upgraded.
Need to get 1,541 kB of archives.
After this operation, 6,356 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

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”.

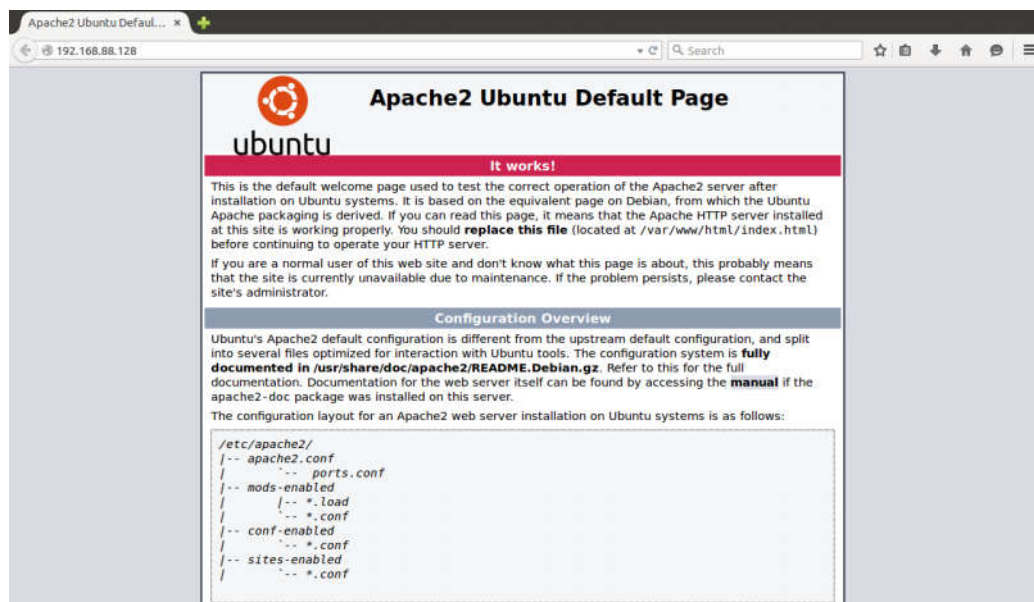

```

ubuntu@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:8e:8b:cd
          inet addr:192.168.88.128  Bcast:192.168.88.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe8e:8bcd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:69070 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31649 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:90430391 (90.4 MB)  TX bytes:2793238 (2.7 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2147 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2147 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1034560 (1.0 MB)  TX bytes:1034560 (1.0 MB)

```

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.

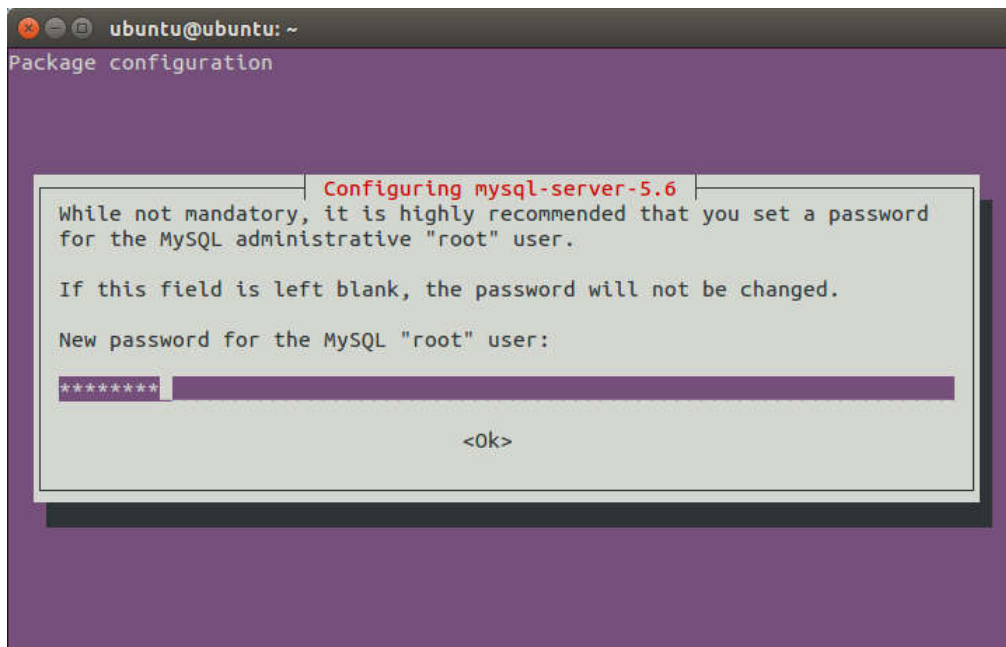

```

ubuntu@ubuntu:~$ sudo apt-get install mysql-server libapache2-mod-auth-mysql php
5-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
  libterm-readkey-perl mysql-client-5.6 mysql-client-core-5.6 mysql-common
  mysql-server-5.6 mysql-server-core-5.6 php5-common
Suggested packages:
  libmldbm-perl libnet-daemon-perl libsql-statement-perl
  libipc-sharedcache-perl mailx tinyca php5-user-cache
The following NEW packages will be installed:
  libaio1 libapache2-mod-auth-mysql libdbd-mysql-perl libdbi-perl
  libhtml-template-perl libmysqlclient18 libterm-readkey-perl mysql-client-5.6
  mysql-client-core-5.6 mysql-common mysql-server mysql-server-5.6
  mysql-server-core-5.6 php5-common php5-mysql
0 upgraded, 15 newly installed, 0 to remove and 270 not upgraded.
Need to get 21.2 MB of archives.
After this operation, 156 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

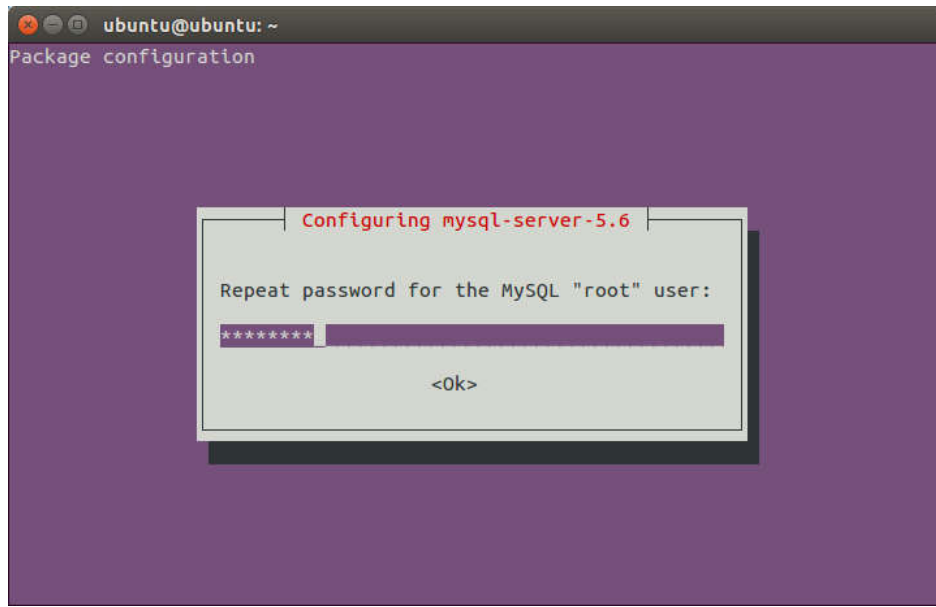
```

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`

```
ubuntu@ubuntu:~$ sudo mysql_install_db
```

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`

```
ubuntu@ubuntu:~$ sudo /usr/bin/mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.
```

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”.

```
By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MySQL comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!
```

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`

```

ubuntu@ubuntu:~$ sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libmcrypt4 php5-cli php5-json php5-readline
Suggested packages:
  php-pear libmcrypt-dev mcrypt
The following NEW packages will be installed:
  libapache2-mod-php5 libmcrypt4 php5 php5-cli php5-json php5-mcrypt
  php5-readline
0 upgraded, 7 newly installed, 0 to remove and 270 not upgraded.
Need to get 4,636 kB of archives.
After this operation, 20.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

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`

```

ubuntu@ubuntu:~$ sudo apt-get install php5-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
php5-mysql is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 270 not upgraded.

```

Instalación de PhpMyAdmin

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

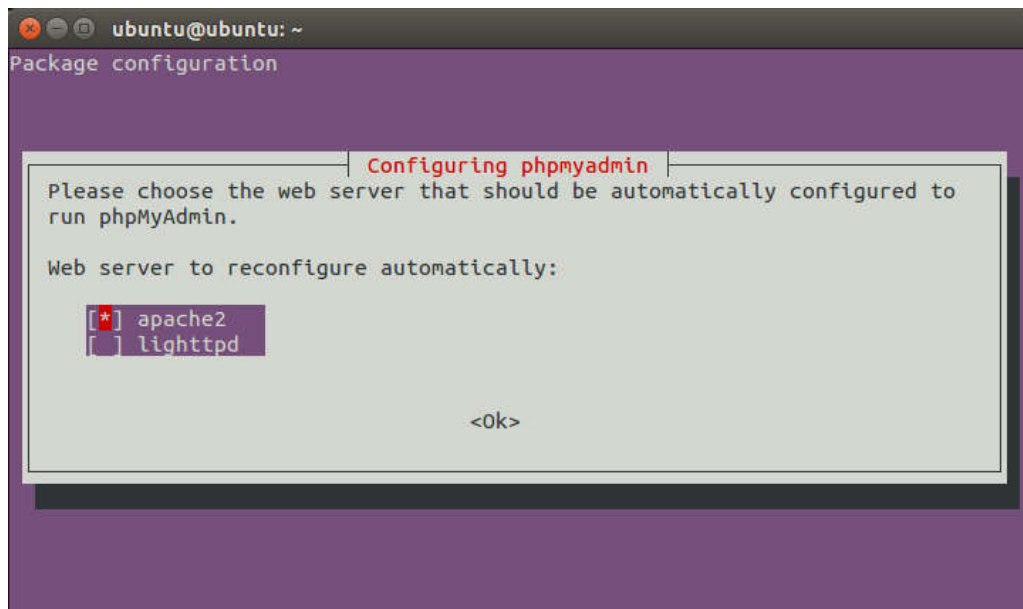
- `sudo apt-get install phpmyadmin`

```

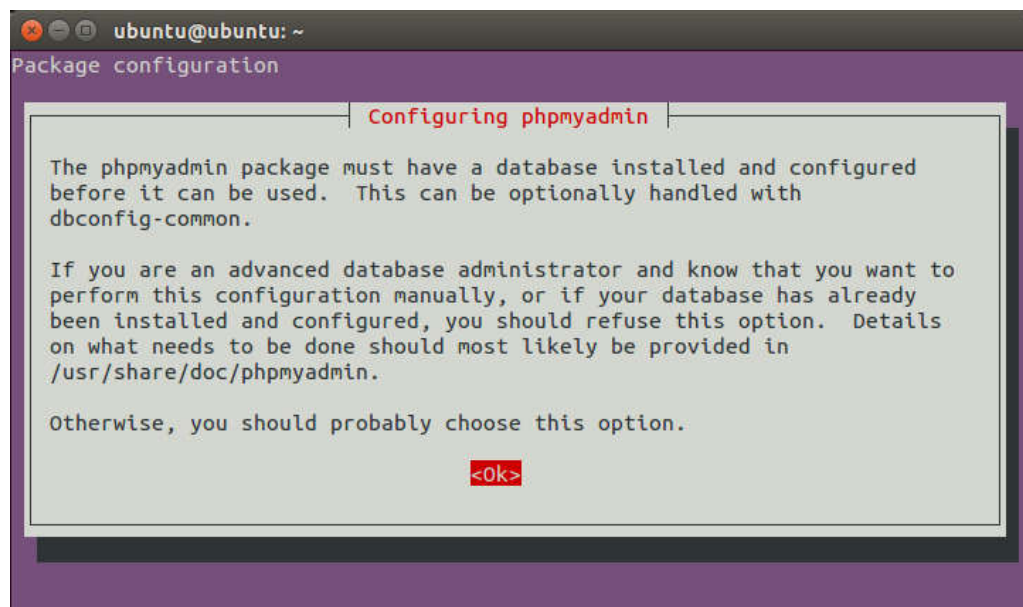
ubuntu@ubuntu:~$ sudo apt-get install phpmyadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  dbconfig-common libjs-sphinxdoc php-gettext php-tcpdf php5-gd
Suggested packages:
  php5-imagick
The following NEW packages will be installed:
  dbconfig-common libjs-sphinxdoc php-gettext php-tcpdf php5-gd phpmyadmin
0 upgraded, 6 newly installed, 0 to remove and 270 not upgraded.
Need to get 12.4 MB of archives.
After this operation, 46.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

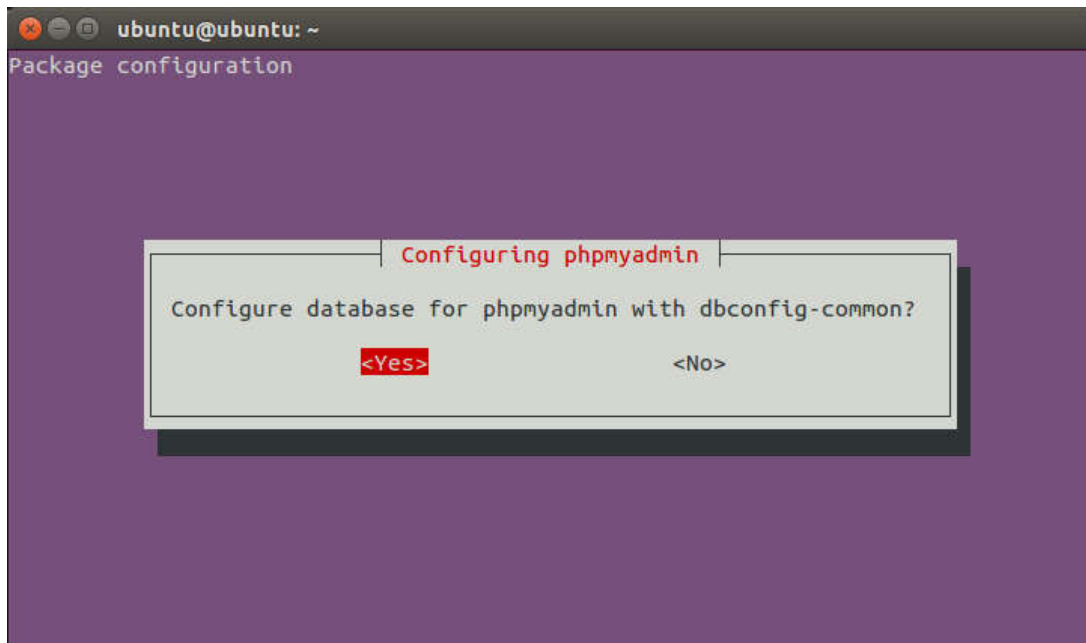
Debemos seleccionar que servidor se debe configurar, en este caso, se seleccionará “apache2” con la barra espaciadora y se presionará “Enter”.



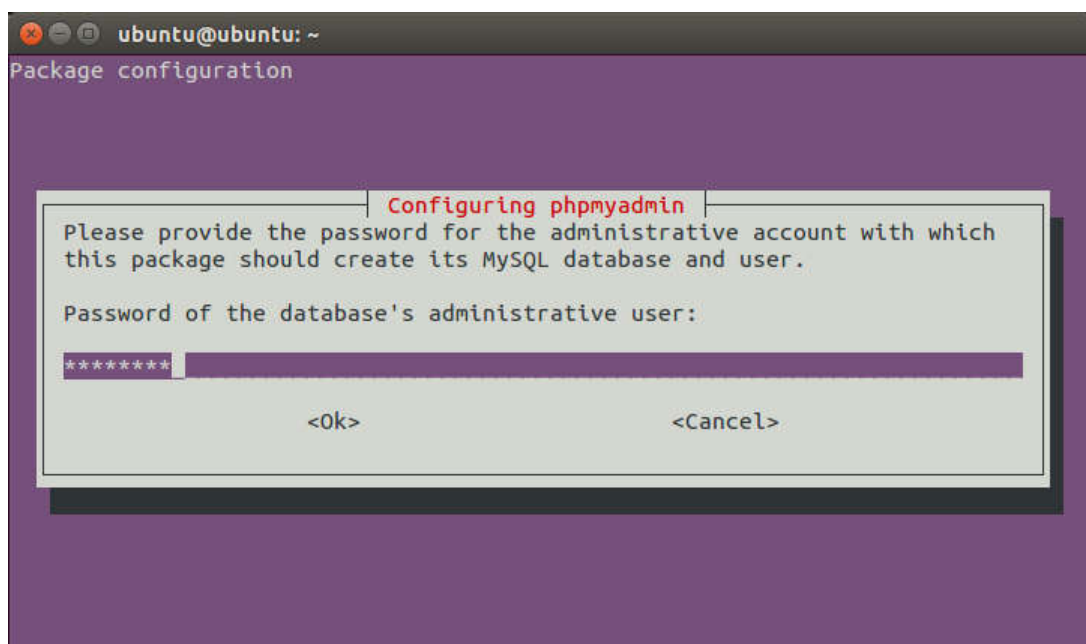
En la imagen siguiente, se realiza la configuración de PhpMyAdmin por lo que se debe presionar “Enter”.



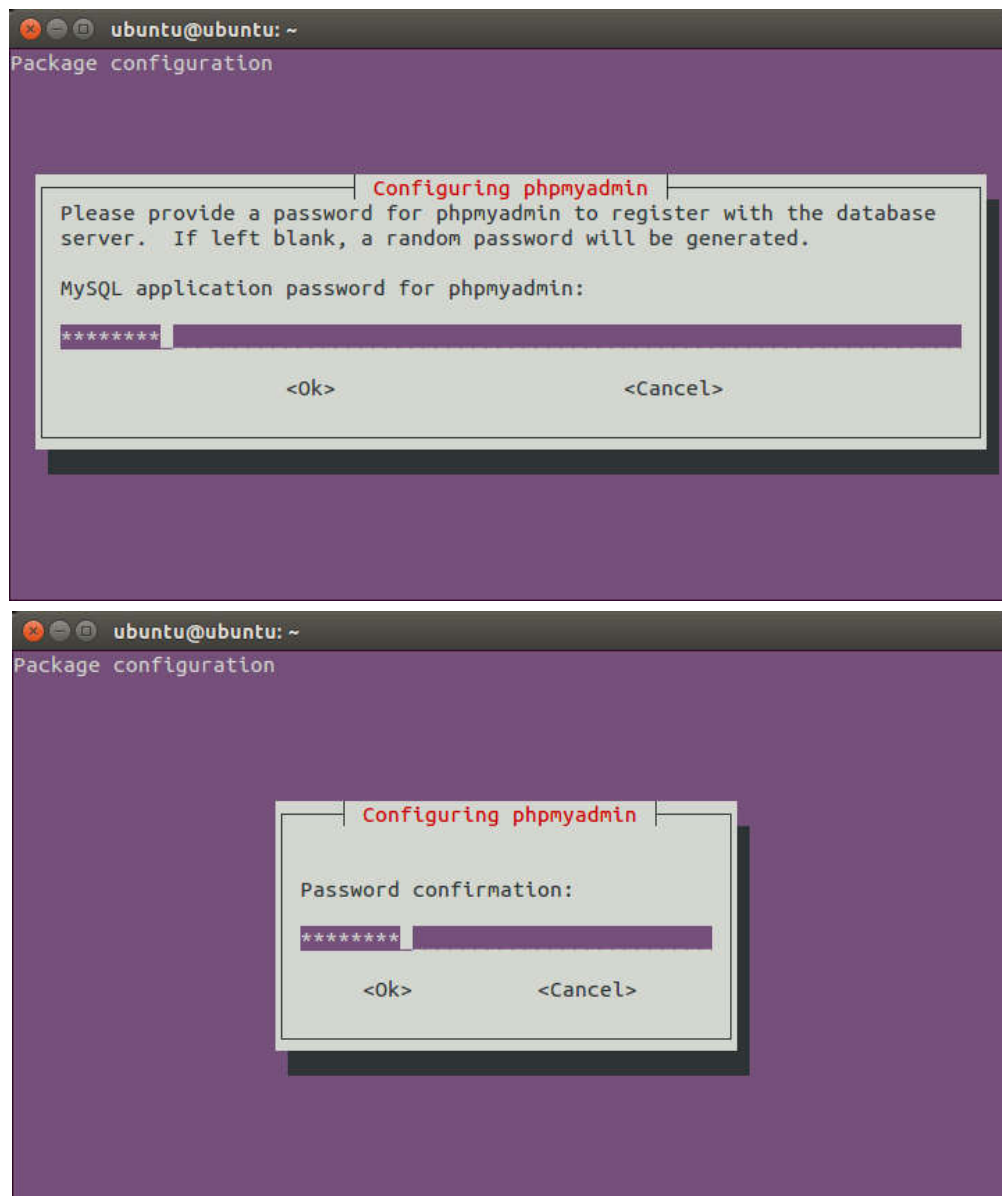
A continuación, se selecciona “Yes” y se presiona “Enter”.



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



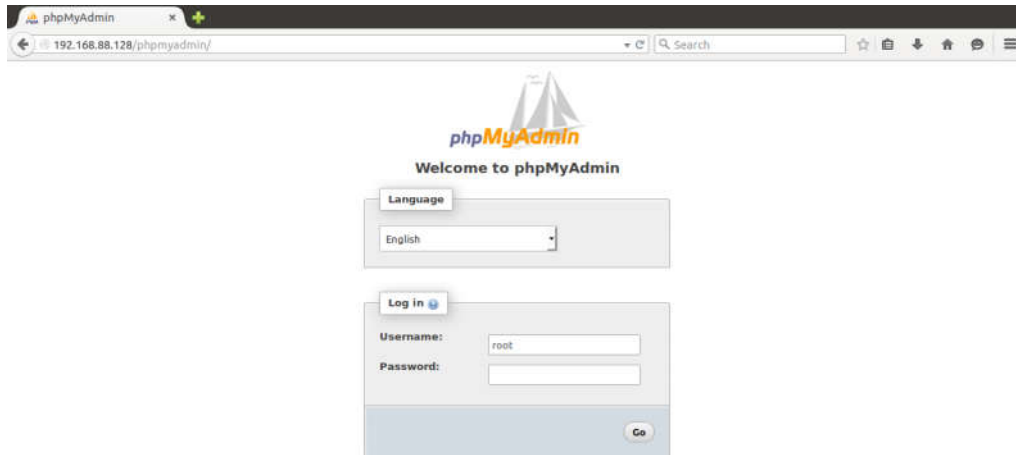
Y finalmente, se le pedirá que elija y confirme una contraseña para la aplicación phpMyAdmin.



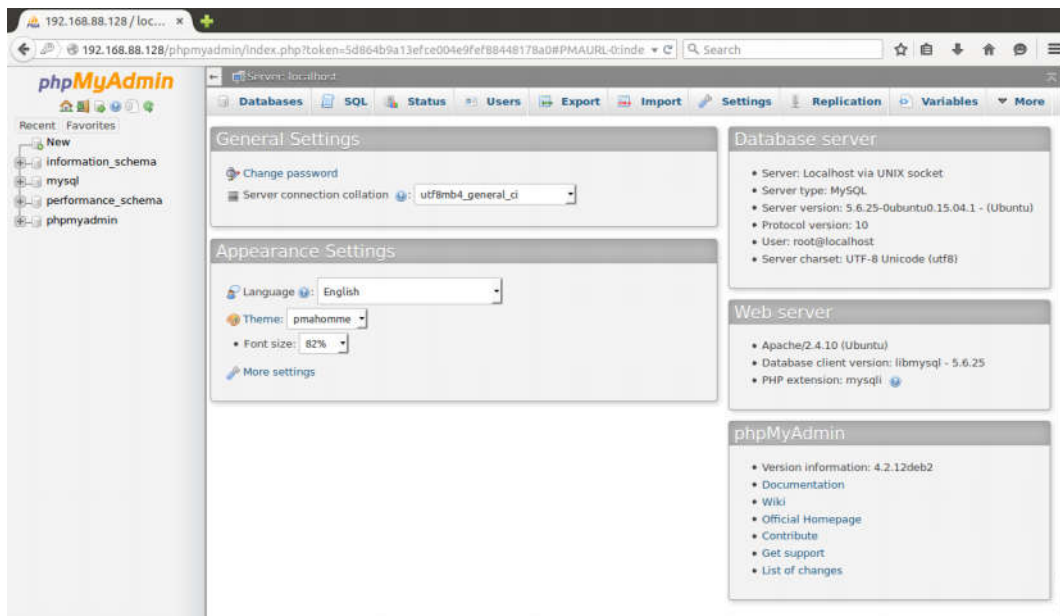
Finalmente, para corroborar el correcto funcionamiento del administrador de base de datos, se debe ingresar en la barra de direcciones del navegador la siguiente dirección:

- *ipDelServidor/phpmyadmin*

Deberá aparecer la siguiente ventana



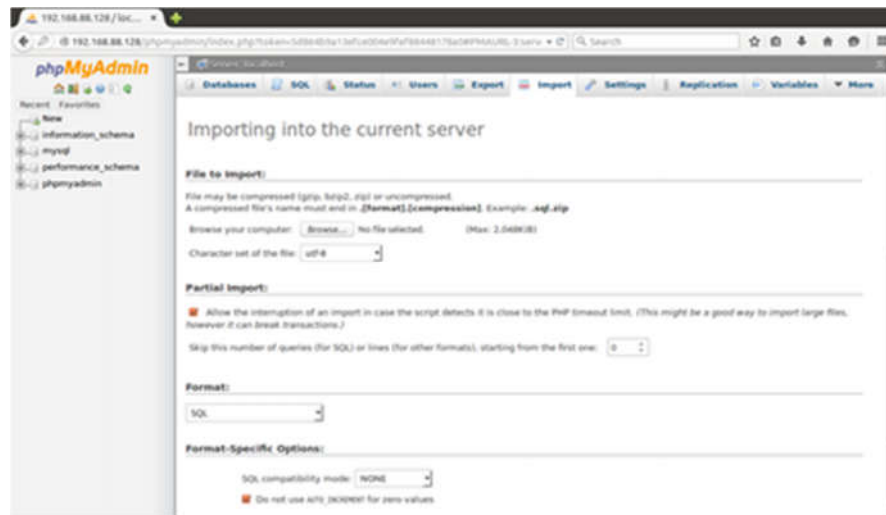
Para ingresar al sistema, se debe poner el usuario root, junto con la contraseña del servidor de mysql que se proporcionó en la instalación. Una vez que se ingresó, se debe ver la siguiente ventana:



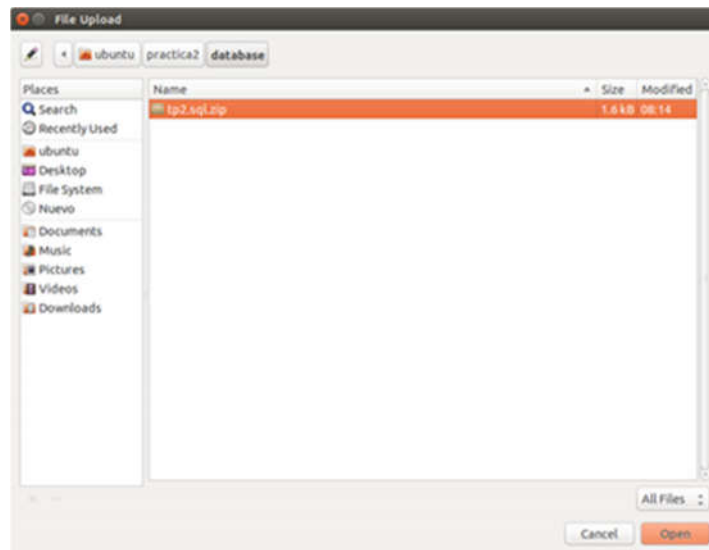
Importación de la base de datos al phpmyadmin

En el caso de que se quiera importar la base de datos sin tener que definir las entidades anteriores, se puede descargar el archivo llamado “tp2.sql.zip”⁴ que contiene la base de datos “tp2” con las tablas mencionadas en el apartado “Base de Datos - Diseño”.

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

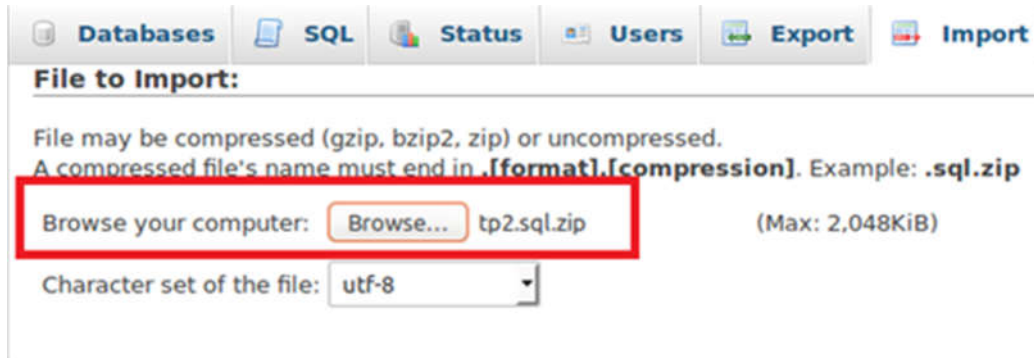


Luego, al lado de donde dice “Browse your computer:”, hay un botón, “Browse...” que permite buscar en el sistema de archivos una base de datos. Navegar hasta el directorio que se mencionó y elegir el archivo comprimido.



⁴ <https://mega.nz/#!/MAXjUJQb!CTWngJb3e57VuKT7iY8riaj4B3tqEksQdMC4GZiNT3U>

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



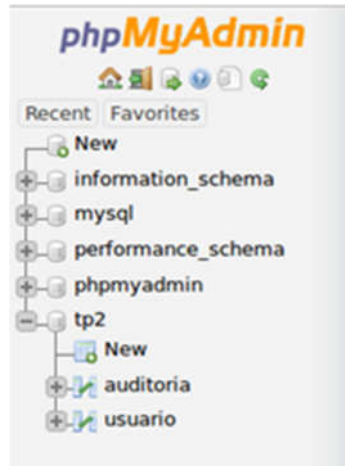
File to Import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **[format].[compression]**. Example: **.sql.zip**

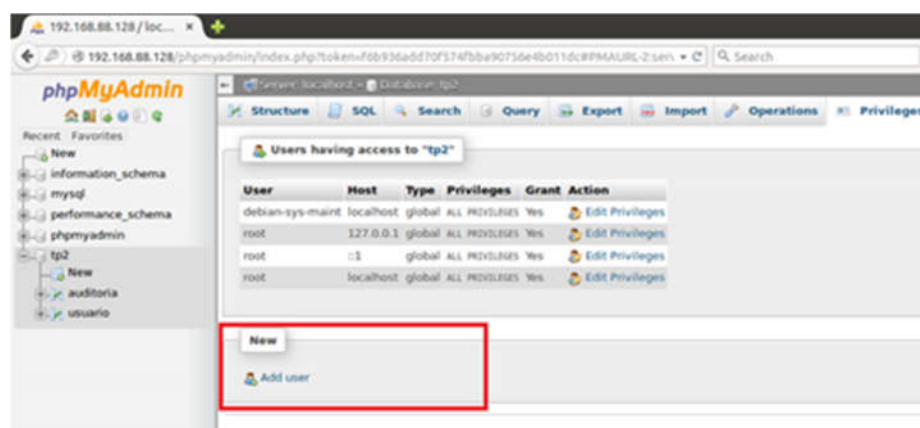
Browse your computer: tp2.sql.zip (Max: 2,048KiB)

Character set of the file:

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.

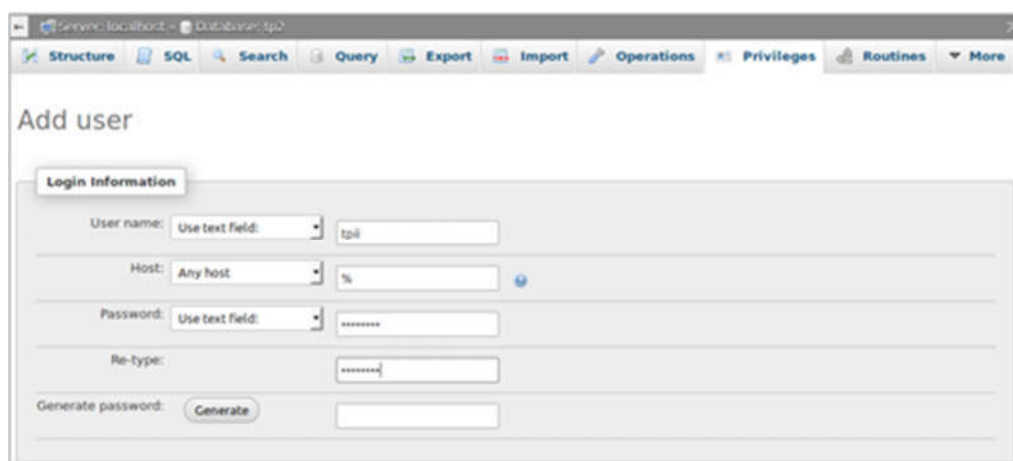


Para poder acceder a esta nueva base de datos desde la aplicación, se debe crear un usuario con ciertos privilegios. Para ello se debe ingresar a la solapa “Privileges” y agregar un nuevo usuario, haciendo click en “Add User”.

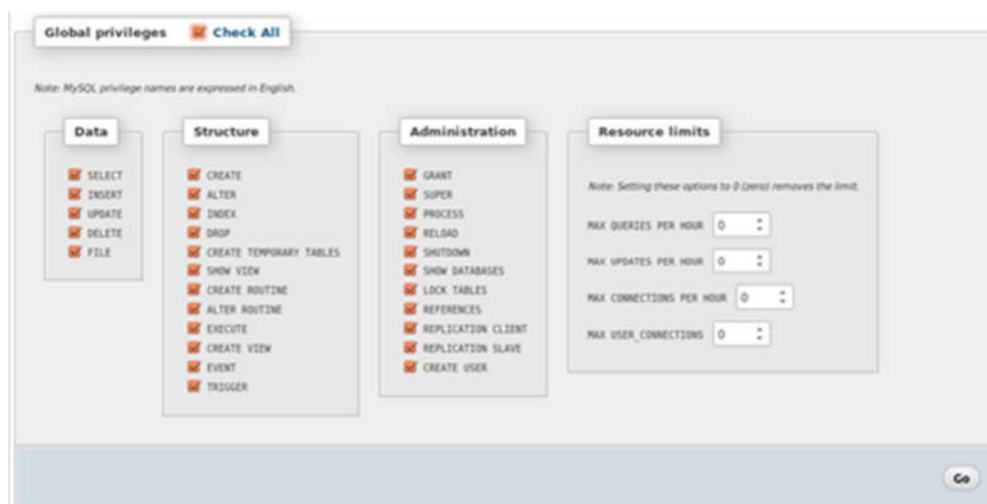


Se deben completar los siguientes campos:

- User name: nombre de usuario
 - En este caso: tpri
- Host: desde donde se conectará el usuario
 - En este caso: % (desde cualquier IP)
- Password: contraseña del usuario
 - En este caso: “tpri2015” (sin comillas)
- Re-type: poner de nuevo la contraseña

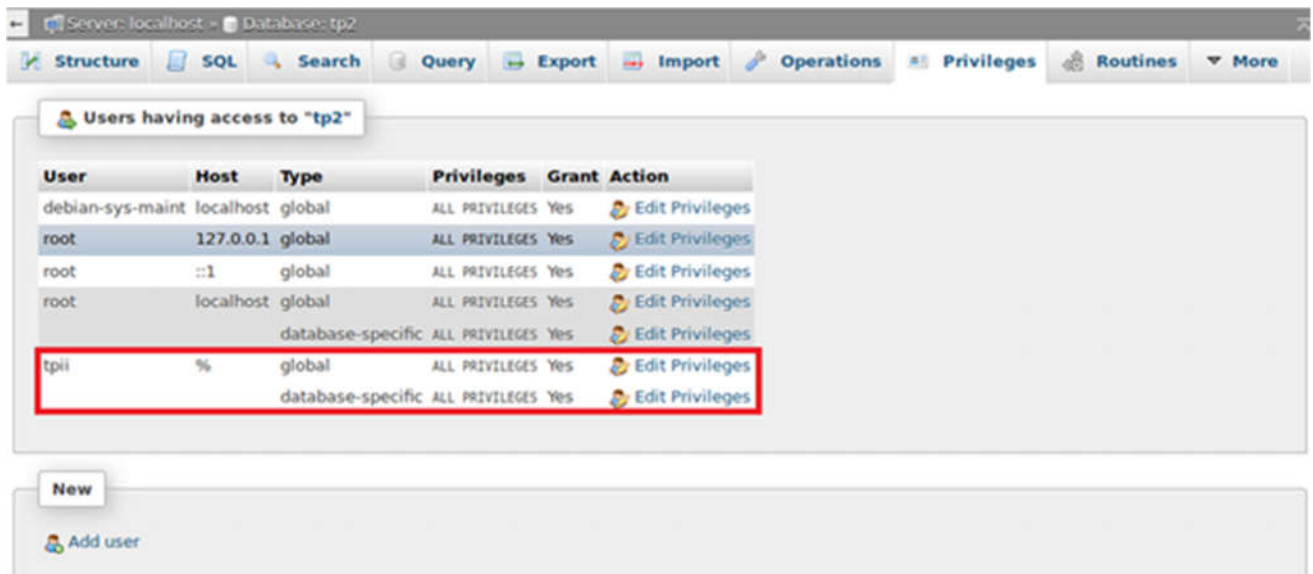


Siguiendo con la página, se puede asignar los privilegios que va a tener este usuario en la base de datos.



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.



Instalación Node.js v0.10.36 en Ubuntu 15.04

Aclaraciones:

- **Debe estar conectado a internet al momento de realizar estos pasos**
- **Debe tener posibilidad de usuario administrador**

Para la instalación la versión v0.10.36 del Node.js se deben realizar los siguientes pasos:

1. Crear un directorio “instalarNodejs”.
 - a. comando: `mkdir instalarNodejs`
 - b. comando: `cd instalarNodejs`
2. Descargar la versión v0.10.36 del Node.js
 - a. comando: `wget https://nodejs.org/dist/v0.10.36/node-v0.10.36.tar.gz`
3. Descomprimir el archivo tar.gz
 - a. comando: `tar -xvzf node-v0.10.36.tar.gz`
4. Ingresar al directorio node-v0.10.36
 - a. comando: `cd node-v0.10.36`
5. Ejecutar el archivo de configuración del Node.js
 - a. comando: `./configure`

- b. Ejecutar el comando make
 - c. comando: make
 - d. Ejecutar el comando “make install”
- 6. En caso de que no funcione, anteponer la orden “sudo” e introducir la clave del usuario administrador
 - a. comando: [sudo] make install
- 7. Verificar si se instaló correctamente
 - a. comando: node -v
 - i. respuesta: “v0.10.36”

Descarga del proyecto servidor

Se deberá descargar el archivo “spotify.js”⁵ y “Package.JSON”⁶ en la misma carpeta que se instaló el módulo “node-spotify”. Realizar la instalación del módulo externo necesario, que en este caso es el “Express” para poder realizar el servidor. Esto se realiza mediante el comando:

- npm install
 - Esto leerá las dependencias del Package.JSON e instalará el módulo “express”

Por cuestiones de seguridad, será necesario abrir el archivo para modificar los datos del usuario premium (usuario y contraseña).

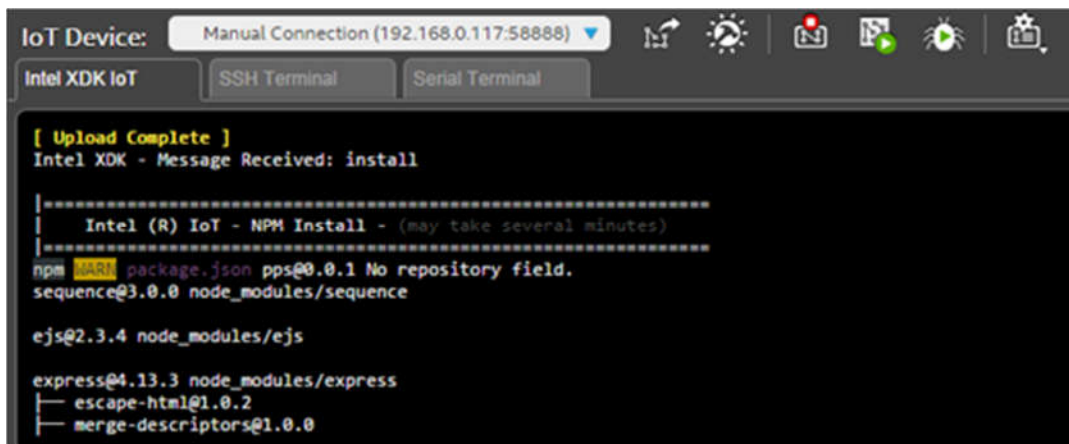
Ejecución del proyecto final

Para llevar a cabo la ejecución del proyecto final, será necesario realizar la configuración de dos variables en el código, estas son, “ipMusic” y “ipDataBase”. A estas, se deberá asignar el “ip” de la computadora servidor (PC 2). Para obtenerlo, será necesario ejecutar el comando “ifconfig” en la terminal.

Finalmente, hay que hacer click en el ícono  para que se construya el proyecto y se instalen los módulos externos (Figura 10).

⁵ <https://mega.nz/#!JdhnyAzQ!zuWbkB26VEO0VxuGlo56viN4SwNYLM4vMfW67UglGX0>.

⁶ <https://mega.nz/#!NMgRBbKB!00RicO51LJYzQtqost4021GAhx4i4LZvVyubvg9XmPA>

The image shows a screenshot of the Intel XDK IoT interface. At the top, there's a header with 'IoT Device:' and a dropdown menu set to 'Manual Connection (192.168.0.117:56888)'. Below this are three tabs: 'Intel XDK IoT', 'SSH Terminal', and 'Serial Terminal'. The 'SSH Terminal' tab is active, displaying a terminal window with the following text:

```
[ Upload Complete ]
Intel XDK - Message Received: install

=====
| Intel (R) IoT - NPM Install - (may take several minutes)
|=====
npm WARN package.json pps@0.0.1 No repository field.
sequence@3.0.0 node_modules/sequence

ejs@2.3.4 node_modules/ejs

express@4.13.3 node_modules/express
├─ escape-html@1.0.2
└─ merge-descriptors@1.0.0
```

Figura 10. Comandos ejecutados desde Intel XDK en el Linux Yocto.

Hasta este momento, la placa se encuentra con todos los módulos instalados. Ahora, será necesario poner en marcha el servidor en la PC 2. Para esto, se debe acceder desde la terminal al directorio donde se encuentra descargado y ejecutar el programa a través del comando “nodejs spotify.js”.

Una vez realizada esta tarea, se subirá el proyecto a la placa a través del botón que se encuentra a la izquierda del botón construir.

Para asegurarse que la aplicación comience a correr, hacer click en el ícono de “Stop” y luego hacer click en el botón “Run”. Se pondrá en un estado inicial el sistema, ya sea si había un usuario antes de que la placa se apague, acondicionará el recinto, o si tiene que poner todos los sensores y actuadores apagados. Además, levantará el servidor web para escuchar peticiones del navegador en el puerto 5000.

Entonces, abrir un navegador y poner en la barra de direcciones la siguiente información:

- direcciónIpDeLaPlaca:puertoQueEstaEscuchando
 - Ej: 192.168.0.150:5000

Cuando se presione la tecla Enter, aparecerá una página como en la Figura 11.

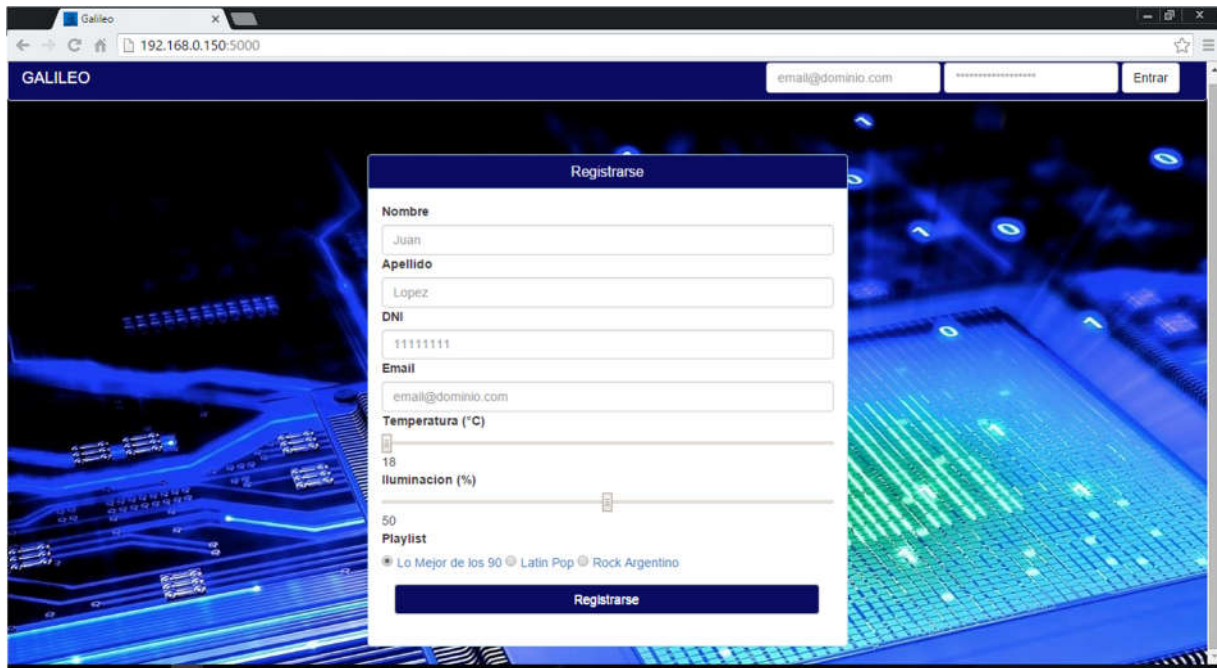


Figura 11. Página principal de la aplicación.