

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Electrónica, Sistemas e Informática

Desarrollo tecnológico y generación de riqueza sustentable

PROYECTO DE APLICACIÓN PROFESIONAL (PAP)

PAP Programa Ciudades Inteligentes



4L05 Vida Digital

Desarrollo de Web Services en NodeJS para la red de sensores inalámbrica

PRESENTAN

Programas educativos y Estudiantes

Lic. en Ingeniería en Sistemas Computacionales. Gabriel Alejandro Gómez López

Profesor PAP: Luis Eduardo Pérez Bernal

Tlaquepaque, Jalisco, mayo 2020

ÍNDICE

Contenido

REPORTE PAP	2
Presentación Institucional de los Proyectos de Aplicación Profesional	2
Resumen	2
1. Introducción	3
1.1. Objetivos	3
1.2. Justificación.....	3
1.3 Antecedentes	4
1.4. Contexto.....	4
2. Desarrollo	5
2.1. Sustento teórico y metodológico	5
2.2. Planeación y seguimiento del proyecto	8
3. Resultados del trabajo profesional.....	17
4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto.....	20
5. Conclusiones	21
6. Bibliografía.....	22
Anexos (en caso de ser necesarios).....	22

REPORTE PAP

Presentación Institucional de los Proyectos de Aplicación Profesional

Los Proyectos de Aplicación Profesional (PAP) son una modalidad educativa del ITESO en la que el estudiante aplica sus saberes y competencias socio-profesionales para el desarrollo de un proyecto que plantea soluciones a problemas de entornos reales. Su espíritu está dirigido para que el estudiante ejerza su profesión mediante una perspectiva ética y socialmente responsable.

A través de las actividades realizadas en el PAP, se acreditan el servicio social y la opción terminal. Así, en este reporte se documentan las actividades que tuvieron lugar durante el desarrollo del proyecto, sus incidencias en el entorno, y las reflexiones y aprendizajes profesionales que el estudiante desarrolló en el transcurso de su labor.

Resumen

Dentro del presente reporte PAP se describe el proceso que se llevó a cabo a lo largo del semestre primavera 2020 referente al desarrollo de los *webservices* enfocados en el lado del servidor (*backend*). Así mismo, se describen sus finalidades, funcionalidades, implementaciones y tecnologías utilizadas.

Estos *webservices* reciben información de monitoreo ambiental e información interna de los nodos que se encuentran tanto en el bosque de la primavera, como dentro de las instalaciones del ITESO. Almacena dicha información en una base que puede ser consultada con finalidades de análisis de datos vía web por medio de *request* HTTP al servidor alojado en cPanel.

1. Introducción

1.1. Objetivos

Tener una infraestructura para el sistema de monitoreo mejorado utilizando NodeJS por parte del servidor. Esto con la finalidad de aumentar la robustez y rapidez del servidor en cuestión.

Desarrollar los servicios web requeridos para el funcionamiento del sistema de monitoreo utilizando NodeJS tomando como base a las funcionalidades ya implementadas dentro del sistema, al igual que las actualizaciones de los requerimientos por parte del equipo de *frontend*.

Los anteriores objetivos se estarán realizando por medio de una metodología ágil enfocada en atacar, o bien resolver, los temas de mayor prioridad. Los *webservices* serán catalogados en bloques de desarrollo con una ponderación de prioridad que depende estrictamente de la urgencia en la que los servicios serán requeridos a lo largo del desarrollo del proyecto.

1.2. Justificación

El monitoreo de las variables del medio ambiente nos provee un beneficio fundamental en el combate de los incendios dentro del Bosque de la Primavera. Esto es posible gracias a que podemos detectar focos de incendio de una manera mucho más rápida y eficiente. Además de que la Zona Metropolitana de Guadalajara obtendría un sistema de monitoreo general de variables ambientales que permitirá generar históricos en cuanto a la calidad del aire.

Así mismo el cambio de infraestructura dentro del servidor PHP con el que se encuentra actualmente, se realizará con la finalidad de mejorar la robustez y rapidez del mismo. Esto siguiendo los lineamientos actuales del desarrollo de tecnologías web. Con dicho cambio de recursos implementados se busca principalmente atacar el tema de la escalabilidad del

proyecto, puesto que Javascript es un lenguaje con mayores capacidades de lidiar con múltiples requisiciones al mismo tiempo.

1.3 Antecedentes

El ITESO tiene como antecedente el sistema de nodos que, por medio de un sistema web, provee la interfaz de monitoreo a los usuarios. El servicio que se encuentra actualmente activo está desarrollado con PHP a lo largo de diferentes semestres y solo una parte del servidor se encuentra desarrollado con *Javascript* en el semestre pasado al presente, él cuenta con una documentación un tanto pobre.

El objetivo principal de esta nueva etapa del proyecto es el de actualizar la infraestructura del servidor haciendo uso de *Javascript*. Aunado a esto se tiene planeado crear la documentación del *toolchain*, donde se podrá ver de manera más sencilla el cómo es que todos los datos pasan desde ser recogidos hasta ser desplegados en pantalla para el usuario final.

1.4. Contexto

Este proyecto está siendo desarrollado en un contexto socio-cultural donde, por cuestiones naturales, políticas o económicas, los pulmones de la ZMG (Bosque de la Primavera) ha sufrido múltiples incendios importantes a lo largo de los últimos años. Esto mismo ha provocado un desbalance en la flora y fauna del bosque, cambios en la temperatura de la misma ciudad y una baja significativa en cuestión de calidad del aire que respiramos día a día.

Teniendo esto en mente, una red de sensores ambientales nos puede dar indicativos de las fluctuaciones de todas estas variables ambientales que están ligadas a la problemática y, con ello, poder tomar cartas en el asunto para tomar medidas en el asunto.

2. Desarrollo

2.1. Sustento teórico y metodológico

Servidor:

Por lado del hardware es básicamente una computadora que alberga software dedicado para un servidor y los sus archivos correspondientes (documentos HTML, imágenes, hojas de estilo y archivos de JavaScript). Está conectado a Internet y soporta la transferencia de datos a otros dispositivos conectados a la web.

Del lado de software, es una pieza de software que entiende de direcciones web, del protocolo HTTP y maneja el flujo del cómo los usuarios finales podrán navegar entre los diferentes archivos del servidor.

Webservice:

Un servicio web es una aplicación funcional accesible mediante una interfaz de red creada utilizando tecnologías estándar de internet. Podría decirse que es una aplicación accesible utilizando protocolos mediante HTTP, XML, SMTP o Jabber.

HTTP:

HTTP o bien Protocolo de Transferencia de Hipertexto es, como lo indica su nombre, un protocolo de transferencia de documentos hipermedia como lo son los HTML. Está diseñado para generar la comunicación entre los navegadores web y los servidores web.

URL:

El Localizador Uniforme de Recursos una secuencia estandarizada de caracteres que identifica y permite localizar y recuperar una información determinada en Internet. Es comúnmente conocido como dirección web o enlace. Esta secuencia de caracteres es la dirección en la cual se alojará un servicio web dentro de Internet, accesible desde cualquier punto del planeta.

Request:

Es un comando del protocolo HTTP, el cual permite enviar todo tipo de petición HTTP a un URL específico, para que esta petición pueda ser recibida por un servidor y que éste mismo genere una respuesta. Existen varios tipos de *request*, entre los cuales están los siguientes: DELETE, GET, HEAD, OPTIONS, POST, PUT, TRACE, entre otros los cuales cumplen funciones diferentes dentro del paso de información por medio del protocolo HTTP.

Response:

Es la manera en la que el servidor responde al cliente sobre las peticiones que éste último genera. De esta manera el servidor le provee al cliente los recursos que éste ha solicitado, manda estatus de aprobación a las peticiones y códigos de error de ser necesarios. Estas respuestas contienen una línea de estado, *headers* que indican información del cómo la información que se está pasando tiene que ser tratada y además cuenta con un *body*, que es la parte del mensaje o información que se le manda al cliente.

Ciente web:

Puede ser llamado como aplicación o bien navegador web (como Google Chrome, Microsoft Edge, Firefox o Safari). Estos están instalados en la computadora de un usuario el cual interactuará con servidores web por medio de peticiones HTTP. El Servidor web será es capaz de entender estas peticiones y le brindará al usuario la información solicitada por medio del cliente web.

Frontend:

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

Backend:

Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El *Backend* también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador información

RESTful API:

Un RESTful API consta de 5 principios, Todo es un recurso, cada recurso tiene un identificador único, uso de métodos HTTP estándar (GET, POST, PUT, DELETE, HEAD,

OPTIONS, TRACE, CONNECT), los recursos pueden tener diferentes representaciones, la comunicación no guarda estado.

JavaScript:

Es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Javascript está incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. Es importante recalcar que es un lenguaje interpretado y basado en prototipos.

2.2. Planeación y seguimiento del proyecto

- Descripción del proyecto

El proyecto consiste en dos grandes entregables:

Toolchain: Diagrama donde se muestre de manera visual, acompañada de algo de texto, el flujo de trabajo del proyecto entero. Con esto mejoraría la trazabilidad de los elementos del proyecto para futuras implementaciones o cambios.

Servidor web en Javascript: Es la implementación de los servicios web que aún no están implementados en el lenguaje Javascript. Aunado a ello, revisarán y modificarán, de ser necesario, los servicios web ya implementados en este lenguaje de programación, con la única finalidad de hacer del servidor *failsafe*.

Para lograr la propuesta de proyecto se estará trabajando con un método ágil enfocado a la urgencia de los servicios web y concentrados en bloques de trabajo (aspecto discutido con los requerimientos de *frontend*). Cada uno de estos bloques tendrán un ciclo aproximado de dos semanas, donde se

programará, documentará y testeará cada uno de los bloques de servicios web implementados.

- Plan de trabajo

El plan de trabajo ha sido elaborado por el profesor que toma el papel de manager dentro del proyecto. Este se ha ido desarrollado de manera dinámica, por lo tanto, las tareas fueron creándose, dependiendo de las circunstancias del proyecto.

Actividad	Tipo de Actividad	Tipo de Recursos Necesarios
Revisión de requerimientos	Operativa	Humanos
Organización de trabajo con el equipo de frontend	Profesional	Humanos
Documentación de productos a desarrollar	Profesional	Humanos
Servicio web que entrega las lecturas generales de un nodo	Técnica	Tecnológicos
Crear servicios web para testeo y desarrollo de productos en frontend	Técnica	Tecnológicos
Servicio web que entrega lecturas de un nodo por día	Técnica	Tecnológicos
Servicio web que entrega lecturas de un nodo por semana	Técnica	Tecnológicos
Servicio web que entrega lecturas de un nodo por mes	Técnica	Tecnológicos
Servicio web que entrega lecturas de un nodo por año	Técnica	Tecnológicos
Servicio web sensiteso	Técnica	Tecnológicos

Servicios de manipulación de nodos	Técnica	Tecnológicos
Servicios de manipulación de variables	Técnica	Tecnológicos
Servicios de manipulación de usuarios	Técnica	Tecnológicos

Juntas y acuerdos

Frontend y diseño:

- Añadir columna “Tipo de nodo” dentro de la base de datos que especifique si el nodo es de tipo, calidad de aire, calidad del agua o de temperatura.
- Petición HTTP para conocer los datos de un nodo por día, por mes o por año. Siendo que el servidor recibe los datos del día, mes y año especificado por el usuario en la interfaz.
- Petición HTTP para conocer los datos de un nodo por semana. El servidor calculará una semana a partir de una fecha dada en la petición.

Desarrollo de propuesta de mejora

La propuesta de mejora es la siguiente:

En un principio se planeó el continuar con el esquema de trabajo de peticiones HTTP que se habían identificado viendo la documentación.

Generación de documentación de un toolchain para entender a grandes rasgos el trabajo que se realiza en el proyecto desde la información que capta el nodo hasta que se ve desplegada la información.

Desarrollo y testeo de bloques de peticiones HTTP tomando en cuenta como valor de prioridad la urgencia que el equipo de frontend tenía planeado en su plan desarrollo de propuesta de trabajo.

Surge un cambio en la dinámica de trabajo y ahora la prioridad de peticiones HTTP se tomarán conforme el project manager dictaminara, tomando como base los requerimientos esenciales del equipo de frontend, que el mismo project manager les marcaba.

Servicio web sensiteso

- Investigación sobre la mejor manera de implementar el servicio web de sensiteso
- Programación de servicio web sensiteso dentro de la cPanel (cambiar implementación)
- Levantar servidor dentro de cPanel específico para el servicio web sensiteso (error con cPanel, teniendo que postergar la actividad)
- Realizar pruebas locales con el servicio sensiteso (teniendo errores con el script de la base de datos)

Setup local del proyecto

- Instalación y configuración de NodeJS

Se tienen que seguir los pasos descritos dentro del programa descargado de la página

<https://nodejs.org/es/download/>

	32-bit	64-bit
Windows Installer (.msi)	node-v12.16.1-win32-x86.msi	node-v12.16.1-win32-x64.msi
Windows Binary (.zip)	node-v12.16.1-win32-x86.zip	node-v12.16.1-win32-x64.zip
macOS Installer (.pkg)	node-v12.16.1.pkg	
macOS Binary (.tar.gz)	node-v12.16.1.tar.gz	
Linux Binaries (x64)	node-v12.16.1.tar.gz	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.16.1.tar.gz	

- Instalación de Express y dependencia SQL

Se tiene que crear un archivo llamado package.json, donde se describen en formato JSON las dependencias que necesita el proyecto para correr de manera local.

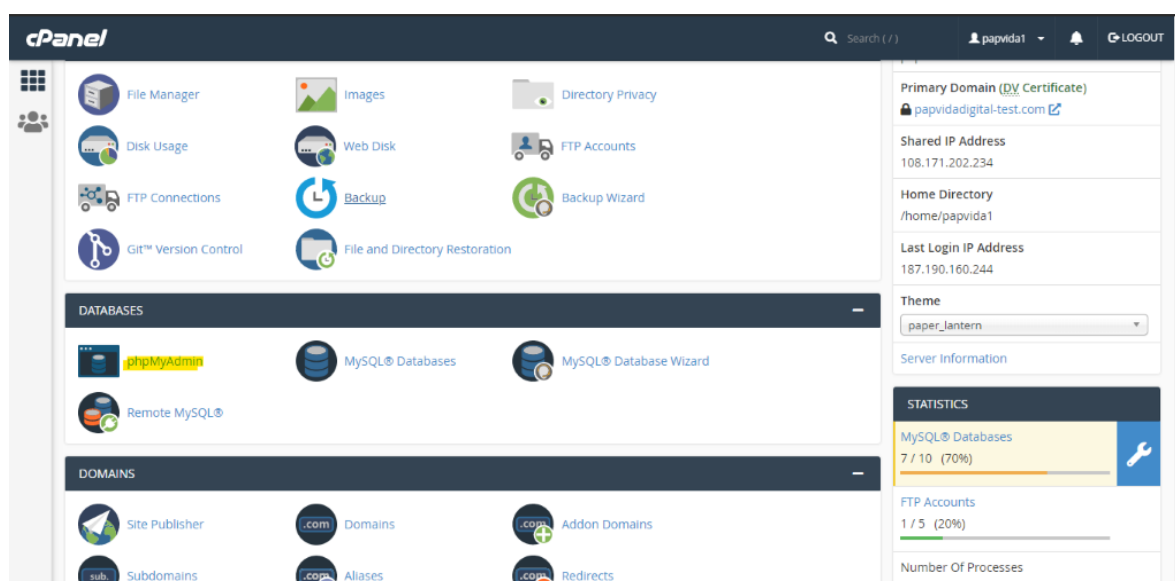
```
1  {
2    "name": "dmy_queries",
3    "version": "1.0.0",
4    "dependencies": {
5      "express": "^4.13.4",
6      "mysql": "^8.0.19"
7    }
8  }
```

- Instalación y configuración de MySQL Server

La instalación de MySQL Server se tiene que realizar por medio de la siguiente página:
<https://www.mysql.com/downloads/> ahí se descargará la versión que concuerde con la señalada dentro del JSON anterior.

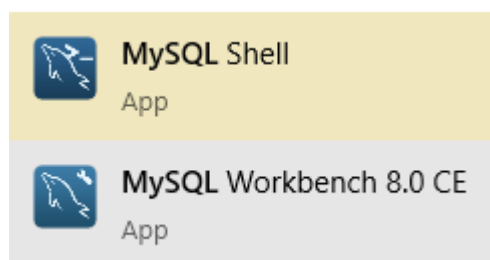
- Exportación del script que posee la base de datos papvida1_nodosP2020

La exportación e importación se puede lograr por medio del panel de administración de cPanel en el apartado de Databases dando click en phpMyAdmin. Los pasos concretos están descritos dentro del reporte técnico.



- Importar base de datos en SQL Server (errores en el script)

La importación de la base de datos se hare por medio de MySQL Shell o bien con MySQL Workbench.



El primero es por medio de comandos en consola y el segundo por medio de una interfaz de usuario.

Los comandos de configuración pertinentes a la importación de los datos se encuentran descritos con mayor detalle dentro del reporte técnico correspondiente al backend

Programación de los servicios web que devuelven los datos de un nodo por día, semana, mes y año.

Los servicios web correspondientes al día, mes y año fueron implementados de manera local, al momento de pasarlos al servidor fallaron y fueron cambiados por los que se encuentran actualmente corriendo en el servidor.

```
app.get('/:nodo/:year/:month/:day', jsonParser, function(req, resp) {  
  // MySQL per month  
  connection.query('SELECT * FROM pap_database where idnodo = "' + nodo +  
    '" AND feha_hora LIKE "' + year + '-' + month + '-' + day + '%" ', function(error, results, fields) {  
    if(!error) {  
      console.log('Error in the query');  
    } else {  
      console.log('Successful query');  
      console.log(results);  
      res.send(results);  
    }  
  });  
});
```

```
app.get('/:nodo/:year/:month', jsonParser, function(req, resp) {  
  // MySQL per month  
  connection.query('SELECT * FROM pap_database where idnodo = "' + nodo +  
    '" AND feha_hora LIKE "' + year + '-' + month + '%" ', function(error, results, fields) {  
    if(!error) {  
      console.log('Error in the query');  
    } else {  
      console.log('Successful query');  
      console.log(results);  
      res.send(results);  
    }  
  });  
});
```

```
app.get('/:nodo/:year/', jsonParser, function(req, resp) {  
  // MySQL per year  
  connection.query('SELECT * FROM pap_database WHERE idnodo = "' + nodo +  
    '" AND fecha_hora LIKE "' + year + '%" ', function(error, results, fields) {  
    if(!error) {  
      console.log('Error in the query');  
    } else {  
      console.log('Successful query');  
      console.log(results);  
      res.send(results);  
    }  
  });  
});
```

El servicio web correspondiente a los datos de un nodo por semana fue implementado primero de manera local y posteriormente dentro del servidor. Cabe resaltar que las versiones de javascript, al ser diferentes cambiaba la manera en que el objeto Date funcionaba en su manipulación.

```

app.get('/test/week/:nodo/:year/:month/:day', (req, res) => {

  let nodo = req.params.nodo;
  let year = req.params.year;
  let month = req.params.month-1;
  let day = req.params.day;

  var end_week = new Date(year, month, day);
  var start_week = new Date(year, month, day);
  start_week.setDate(start_week.getDate()-7);

  connection.query('SELECT * FROM LecturasNodos WHERE idNodo = "' + nodo + '" AND DATE(fecha_hora) BETWEEN "'
+ start_week.getFullYear()+ '-' +parseInt(start_week.getMonth()+1)+ '-' +start_week.getDate()+'" AND "'
+ end_week.getFullYear()+ '-' +parseInt(end_week.getMonth()+1)+ '-' +parseInt(end_week.getDate()-1)+'"', function (error, results) {
    if (error) throw error;
    console.log(results);
    res.send(results);
  });
});

```

Servicio web sensitoso, este webservice es utilizado únicamente por los nodos de manera autónoma. Con este servicio web, es posible ir registrando dentro de la base de datos, las lecturas que los nodos van registrando en tiempo real por medio de tramas que el servidor puede entender e interpretar.

```

"ID;NPL;AC;TD;TN;0;BAT;98";
"ID;I1;AC;TD;TN;0;TmpS;23.35;TmpI;0.00;HumS;87.52";
"ID;TT1;AC;TD;BT;33";
"ID;TT1;AC;TD;BT;33";
"ID;TS;AC;TD;TS;18:02:15:16:26:01;CO2;11.1;O2;21.1;TEMP;25.01;BAT;81";
"ID;TS;AC;TD;TS;18:02:15:16:26:01;CO2;11.1;O2;21.1;TEMP;25.01;BAT;81";
"ID;TS;AC;TD;TS;18:02:15:16:26:01;CO2;11.1;O2;21.1;TEMP;25.01;BAT;81";
"ID;TS;AC;TD;TN;0;CO2;10;O2;20;TEMP;23.33;BAT;78";

```

Las tramas son construidas a base de la información que leen los nodos y poseen en formato JSON:

```

{id:"HM1","ac":"td","tn":"2020-04-16T17:44:38-0500","temp":"28.50","hum":"27.90"}
{id:"HM1","ac":"td","tn":"2020-04-16T18:04:59-0500","temp":"28.60","hum":"27.80"}
{id:"HM1","ac":"td","tn":"2020-04-16T18:15:09-0500","temp":"28.70","hum":"27.90"}
{id:"HM1","ac":"td","tn":"2020-04-16T18:25:19-0500","temp":"28.80","hum":"27.90"}

```

La deconstrucción de la trama a una lista de variables debe de tomarse en cuenta para hacer sacar los datos necesarios dentro de la base de datos. Y que estos datos no están siendo usados como mapas o tablas hash. Para ello a cada variable se le hace un *LowerCase* con la finalidad de manipular las variables sin necesidad de estar haciendo la conversión cada vez que se requieran los datos.

```
app.get('/sensiteso/:data',function(req, resp) {

  if (data_0.length<= 0){
    console.log("BAD_REQUEST");
    exit(1);
  }
  // change data_0 for "data"
  data_list = data_0.substr(5, data_0.length).split(";");

  if(data_list.length % 2 != 0){
    console.log("BAD_REQUEST");
    exit(1);
  }

  for (i=0; i <data_list.length; i++) {
    data_list[i] = data_list[i].toLowerCase();
  }

  let id = data_list[0];
```

Manipulación de información (variables, nodos, usuarios)

Para el apartado de usuarios existe la implementación para poder recibir los datos de todos los usuarios registrados dentro del servidor y también se implementó el servicio web que los elimina de la base de datos permanentemente. Estos no fueron completamente implementados en conjunto con el equipo de *frontend*

```
app.get('/accounts', jsonParser, (req, res) => {

  connection.query('SELECT username, type FROM user;', function(error, results) {
    if(error) throw error;
    console.log(results);
    res.send(results);
  });
});

app.delete('/accounts/:username', (req, res) => {

  let username = req.params.username
  connection.query("DELETE FROM User WHERE username='"+username+"'";, function(error, results) {
    if(error) throw error;
    console.log(results);
  });
});
```


Para la manipulación de nodos y variables existen los HTTP Request GET, POST y DELETE, estos se encargan de insertar nueva información, actualizar una ya existente, obtener la información de la base de datos o bien eliminarla.

GET

```
app.get('/nodes', jsonParser, (req, res) => {  
  connection.query('SELECT * FROM Nodos;', function(error, results) {  
    if(error) throw error;  
    console.log(results);  
    res.send(results);  
  });  
});
```

POST (insertar nuevos datos)

```
app.post('/nodes', (req, res) => {  
  let id = req.body.id;  
  let name = req.body.nombre;  
  let dir = req.body.direccion;  
  let long = req.body.longitud;  
  let lat = req.body.latitud;  
  let desc = req.body.descripcion;  
  
  connection.query('SELECT * FROM Nodos WHERE id="'+id+'";', function(error, results) {  
    if(error) throw error;  
    console.log(results);  
  });  
  
  if (results == "") {  
    connection.query("INSERT INTO Nodos (id, nombre, direccion, longitud, latitud, descripcion) VALUES ('"+  
      id+"','"+name+"','"+dir+"','"+long+"','"+lat+"','"+desc+"');", function(error, results) {  
      if(error) throw error;  
      console.log(results);  
    });  
  }  
});
```

POST (modificación a la base de datos)

```
} else {  
  connection.query("UPDATE Nodos SET nombre='"+name+"',direccion='"+dir+"',longitud='"+lon+  
    "',latitud='"+lat+"',descripcion='"+desc+  
    "' WHERE id='"+id+"';", function(error, results) {  
    if(error) throw error;  
    console.log(results);  
  });  
}
```

DELETE

```
app.delete('/nodes/:id', (req, res) => {  
  
  let id = req.params.id;  
  connection.query("DELETE FROM LecturasNodos WHERE idNodo='"+id+"'"; function(error, results) {  
    if(error) throw error;  
    console.log(results);  
  });  
  
  connection.query("DELETE FROM Nodos WHERE id='"+id+"'"; function(error, results) {  
    if(error) throw error;  
    console.log(results);  
  });  
  
});
```

3. Resultados del trabajo profesional

1. Guía rápida de configuración del proyecto y criterios a tener en cuenta para poder desarrollar el proyecto de manera local.

Esta es una guía de configuración que únicamente sirve para dar inicio a la manipulación del proyecto. Se muestran configuraciones de NodeJS, MySQL y un poco del manejo de cPanel.

Con esto puede yo entender un poco el flujo de los datos que iba yo a modificar, si bien hubo problemas con respecto a algunas configuraciones, esta guía breve puede ayudar en un futuro a no perder el tiempo en la búsqueda y manejo de las herramientas empleadas entro del proyecto.

2. Breve guía de la importación de una base de datos:

Existen varios caminos para realizar una importación de la base de datos que se encuentra en cPanel. Dentro de la pequeña explicación trato de mostrar la manera más efectiva de realizar esta importación. Otras alternativas resultaron tener problemas relacionados a un mal manejo del script en Python que realiza el importado de los datos desde el archivo .sql.

3. Codificación de servicio web destinado a recuperar las lecturas de un nodo:
 - a. Por año

Este requisito del proyecto destaca la necesidad de conocer toda la información almacenada dentro de la base de datos a lo largo de un año especificado por el usuario. Este servicio GET es código que se puede usar como plantilla para generar otros servicios similares, lo cual es importante dentro de la programación el poder adaptar implementaciones funcionales con aspectos nuevos a desarrollar.

b. Por mes:

Este servicio al ser similar que el anterior fue, únicamente requiriendo una nueva variable llamada “*month*” dentro de la URL. Cabe resaltar que existe dentro de estos *request* un apartado que arroja dentro de consola si se ha producido un correcto *query* o si este ha fallado, con la única finalidad de tener un control por medio del desarrollador del flujo de la información.

c. Por día:

Al igual que fueron desarrollados los *request* para el mes y el año, este servicio necesita la entrada del usuario del día además de las otras dos variables. Con estas tres variables es sencillo el mandar a llamar los datos desde la base de datos para entregárselas al usuario.

Un aspecto para tener en cuenta es la facilidad con la que los JSON pueden ser creados para que los programadores *FrontEnd* puedan utilizar esta información a su gusto. También es importante la coordinación con otros equipos para saber qué cosas necesitas de ellos y qué cosas les vas a ofrecer como resultado.

d. Por semana (siete días):

Este servicio web utiliza las mismas variables que el utilizado para arrojar los datos de un día en específico, solamente que este tiene una URI llamada *week*

antes de todas las variables. Con esa simple URI se puede diferenciar de un servicio enfocado únicamente a un día de otro enfocado a los datos de una semana determinada.

Este servicio requirió de más investigación debido a la manera en que se realiza el cálculo de una semana. A la hora de obtener las fechas por medio de un objeto Date en Javascript este formato es diferente entre versiones al hacer una conversión a números enteros, de ahí en fuera solamente se hace una resta de siete días para poder realizar pedirle a la base de datos la información cargada por un nodo en el plazo de una semana.

4. Servicio sensiteso:

Este servicio web está enfocado en la recepción de tramas directamente de los nodos (*gateway*). Estas tramas poseen los datos necesarios para poder diferenciar a quién le pertenece una lectura, su fecha, hora de lectura y los datos censados.

Este servicio web ha sido el más trabajoso de manejar debido a las constantes verificaciones que tienen que realizar para tomar los datos de los nodos.

5. Manipulación de información:

a. Nodos:

Este conjunto de servicios permite la inserción, modificación y eliminación de algún nodo. Esto tomando en cuenta el nombre del nodo, su identificador, sus coordenadas geográficas y una breve descripción.

b. Variables:

Este conjunto de servicios permite la inserción, modificación y eliminación de las variables ambientales o no ambientales que se van insertando a un catálogo de estas, con la finalidad de tener un control de toda la información que la red de sensores inalámbrica puede otorgar.

c. Usuarios:

Aquí si bien no fueron finalizados los servicios correspondientes a la manipulación de usuarios, existe un modelo ya implementado que le

brinda al usuario un listado de todos los usuarios registrados dentro del servidor salvaguardando las contraseñas de estos.

Este servicio se puede utilizar como plantilla para el desarrollo de la modificación de algún usuario o la eliminación de este.

4. Reflexiones del alumno o alumnos sobre sus aprendizajes, las implicaciones éticas y los aportes sociales del proyecto

- Aprendizajes profesionales

Para mí, los aprendizajes profesionales radican en gran manera el salirme de la zona de confort, utilizar un lenguaje de programación que no utilizo en mi día a día, el trabajar en el backend de manera individual teniendo en frente a tres programadores de frontend. Esos primeros apartados los considero maneras difíciles de sobrellevar, aunado a ello el distanciamiento social y otros compromisos mermaron la eficiencia de mis actividades.

He aprendido también la importancia de la comunicación de los diferentes equipos (diseño, frontend y backend) ya que, toda la información que tenía que desplegar venía directamente de los requerimientos de diseño.

Cabe resaltar que este fue para mí un proyecto con una dinámica de trabajo fuera de lo habitual a lo largo de mis años como estudiante o como profesionista.

- Aprendizajes sociales

Si bien yo ya conocía la dinámica de este proyecto, su finalidad y sus implementaciones actuales, es interesante que este semestre me pude percatar de la verdadera escalabilidad del proyecto puede llegar a alcanzar viendo el cómo las variables del tiempo y localización de los nodos.

Este proyecto puede ser podría escalarse desde el ITESO o bien podría importarse en otra parte del mundo en un futuro. Aunque el periodo y mi aportación fueron cortas, siento que realmente ayudo de gran manera a que este proyecto pueda crecer exponencialmente, me parece que el redirigir las herramientas que anteriormente utilizaba a algo mejor, podrá impactar de mayor manera a la sociedad.

- Aprendizajes éticos

La ventaja de trabajar en un proyecto meramente de carácter social me hace tener mejor sensibilidad y compromiso con el contexto ambiental en el que se envuelve Guadalajara, el Bosque de la Primavera y otras metrópolis. Puedo comprender los deberes que tenemos como humanos para cuidar la flora y fauna con el que cohabitamos y me agrada el poder tener una manera de medir el actuar positivo o negativo de la sociedad en cuanto al ambiente se habla.

- Aprendizajes en lo personal

Si bien para mí, este proyecto se fue tornado en un trabajo de segundo plano debido al contexto en el que se desarrolló el semestre, me llevo muy en claro que el trabajo en equipo es lo más importante dentro de los proyectos, el tomar acuerdos en conjunto o bien poder realizar sincronización de avances es elemental en el desarrollo de trabajos.

También llevo en mi entendimiento que debe de existir un equilibrio de equipos de desarrollo, tener una fuerza equitativa entre diferentes departamentos o equipos.

6. Conclusiones

Estar involucrado en el desarrollo de este proyecto en conjunto con compañeros de otra carrera me deja muy en claro la importancia que existe en la completa colaboración y entrega que debe tener cada integrante del equipo para sacar a flote el proyecto entero. A lo largo del semestre pude notar, al menos en mi área, que en efecto existe una mejora a la hora de estar leyendo el código y poder hacerle modificaciones rápidas al mismo. Vaya el objetivo de tener una mejor legibilidad del proyecto se está cumpliendo.

También pude apreciar que la velocidad con la que manipulan los datos había mejorado con respecto a la implementación pasada en PHP. Y, a su vez, puede darme cuenta de que la escalabilidad del proyecto se puede ir apreciando cada vez mas una vez que el número de nodos y variables vaya incrementando con el paso del tiempo en el desarrollo de este proyecto de ITESO.

6. Bibliografía

thanhvk, sboroda, klez. (2019). What is a web server?. septiembre 12, 2019, de Mozilla Sitio web

chrisdavidmills. (2019). HTTP. septiembre 12, 2019, de Mozilla Sitio web: <https://developer.mozilla.org/en-US/docs/Web/HTTP>

Snell, J. (2002). Programming Web services with SOAP. Sebastopol, CA: O'Reilly & Associates.

A beginner's guide to web development. (2018). Birmingham, England: PACKT Publishing.

Hahn, E. V. (2016). Express in action : writing, building, and testing Node.js

Raffino, M. (2019). URL. septiembre 12, 2019, de Concepto Sitio web: <https://concepto.de/url>

Anónimo. (2019). HTTP Request. septiembre 12, 2019, de 4D DOC CENTER Sitio web: <https://doc.4d.com/4Dv17/4D/17.3/HTTP-Request.301-4620861.es.html>

Anónimo. (2019). HTTP Responses. septiembre 12, 2019, de IBM Sitio web: https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.3.0/com.ibm.cics.ts.internet.doc/topics/dfhtl22.html

Crockford D.. (2008). JavaScript: The Good Parts. Estados Unidos: O'Reilly.

Anexos (en caso de ser necesarios)

Reporte técnico de Backend Privamera 2020

