REPÚBLICA DEL ECUADOR
Escuela Politécnica Nacional
"E SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.

- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.

- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE RECOLECCIÓN DE DATOS SÍSMICOS BASADOS EN TECNOLOGÍA GPRS Y LA PLATAFORMA RASPBERRY PI 3

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

FRANKLIN FERNANDO NAVARRETE JUNGAL

franklin.navarrete@epn.edu.ec

DIRECTOR: MSc. Ricardo Xavier Llugsi Cañar

CODIRECTOR: MSc. Andrés Fernando Reyes Castro

Quito, noviembre 2019

AVAL

Certifico que el presente trabajo fue desarrollado por Franklin Fernando Navarrete Jungal, bajo mi supervisión.

MSc. RICARDO XAVIER LLUGSI CAÑAR
DIRECTOR DEL TRABAJO DE TITULACIÓN

MSc. ANDRÉS FERNANDO REYES CASTRO
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Franklin Fernando Navarrete Jungal, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____

FRANKLIN FERNANDO NAVARRETE JUNGAL

DEDICATORIA

El presente trabajo está dedicado a todas aquellas personas que siempre estuvieron ahí, dándome ánimo para seguir adelante en cada etapa de la vida en especial va dedicado a mis padres de manera muy especial.

AGRADECIMIENTO

En toda esta etapa universitaria, el más sincero agradecimiento a mi madre y padre quienes fueron las personas que siempre estuvieron ahí apoyándome de ante mano agradeciéndoles por todo lo que ellos han sembrado en mí.

Un agradecimiento a mi hermano y hermana que más que hermanos son amigos en las buenas y en las malas.

Una especial referencia a todas aquellas personas que siempre estuvieron ahí a mi lado en las buenas y en las malas grandes amigos y compañeros ya sean pocos y contados pero nobles y leales.

ÍNDICE DE CONTENIDO

AVAL....I
DECLARACIÓN DE AUTORÍA....II
DEDICATORIA....III
AGRADECIMIENTO....IV
ÍNDICE DE CONTENIDO....V
RESUMEN....VII
ABSTRACT....VIII
1. INTRODUCCIÓN....1
1.1 OBJETIVOS....2
1.2 ALCANCE....2
1.3 MARCO TEÓRICO....3
1.3.1. MOVIMIENTOS SÍSMICOS....3
1.3.2. SISTEMAS CELULARES....4
1.3.2.1. PRIMERA GENERACIÓN (1G)....4
1.3.2.2. SEGUNDA GENERACIÓN (2G)....4
1.3.2.3. GENERACIÓN INTERMEDIA (2.5G)....6
1.3.2.4. SERVICIOS DE GPRS....7
1.3.2.5. INFRAESTRUCTURA DE UNA RED GPRS....8
1.3.2.6. INTERFAZ DE LA RED GPRS....9
1.3.3. HARDWARE Y SOFTWARE DEL SISTEMA....10
1.3.3.1. SOFTWARE DEL SISTEMA....11
1.3.3.2. HARDWARE DEL SISTEMA....11
1.3.3.2.1. MODELOS DE RASPBERRY PI....12
1.3.3.2.2. MÓDULOS DE COMUNICACIÓN GPRS....16
1.3.3.2.3. SENSORES DE MOVIMIENTO....17
1.3.3.3. SISTEMA DE ENERGÍA....18
1.3.3.4. PLATAFORMA HEROKU....22
1.3.3.5. POSTGRETS SQL....23
1.3.3.6. MODELO DE COMUNICACIÓN ISO/OSI....24

2. METODOLOGÍA....29
2.1. REQUERIMIENTO DEL PROTOTIPO....29
2.2. DISEÑO DEL PROTOTIPO....31
2.3. FASE DE IMPLEMENTACIÓN DE LA ESTRUCTURA DEL PROTOTIPO.35
2.4. SOFTWARE DEL SISTEMA....46
2.5. SISTEMA DE ENERGIZACIÓN....62
3. RESULTADOS Y DISCUSIÓN....65
3.1. PRUEBAS APLICADAS AL HARDWARE....65
3.2. PRUEBA APLICADAS AL SOFTWARE....66
3.3. COSTO REFERENCIAL DEL PROTOTIPO....71
4. CONCLUSIONES Y RECOMENDACIONES....73
4.1. CONCLUSIONES....73
4.2. RECOMENDACIONES....74
5. REFERENCIAS BIBLIOGRÁFICAS....75
ANEXOS....78

RESUMEN

Los movimientos sísmicos son fenómenos de la naturaleza que se pueden dar en cualquier momento. Es prescindible especificar que los movimientos sísmicos, se cuantifican con diferentes parámetros como lo es el nivel sísmico medido en escala Richter. Ante una ocurrencia de movimientos sísmicos es imposible evitar o incluso prevenirlos.

A partir de las circunstancias mencionadas se presenta un sistema que permite tener un análisis más exhaustivo de estos fenómenos, por lo cual se plantea la construcción de un prototipo que almacena datos sísmicos con la ayuda de la tecnología GPRS y la plataforma Raspberry Pi 3B.

El primer capítulo analiza la tecnología que ayudará a acoplar diferentes dispositivos para proporcionar un correcto funcionamiento. Los dispositivos que se usan en la construcción son el Módulo GSM/GPRS, la Plataforma Raspberry Pi 3B, los sensores de movimiento sísmico, Paneles Solares, Controladores de Carga y baterías.

En el segundo capítulo se explica la construcción metódica del prototipo en donde el proyecto se enfoca en la parte de software, este permite la unificación de los diferentes equipos en un solo sistema. Se procede a determinar los equipos útiles según la necesidad.

En el tercer capítulo se realiza una discusión de resultados y comparación de este sistema con otro de tipo más profesional, cabe recalcar que este sistema es un prototipo en el cual habrá obviamente pequeñas fallas, defectos o incluso problemas para poder mejorarlos en un futuro. Dichos problemas pueden ser la conectividad a una red, polvo y lluvia hacia los equipos principales e incluso un poco de inestabilidad en la medición ya que se recuerda aquí se trabaja con sensores de tipo comercial mas no sensores especializados.

El sistema implementado permite la recolección de datos, adecuación de los datos, almacenamiento de datos y al final la visualización de los datos en una interfaz web, cabe destacar el uso de la base de datos, la cual permite que los datos adquiridos por los sensores sean debidamente almacenados, por lo tanto, se habla de la integración de todas estas herramientas para la construcción de este sistema.

PALABRAS CLAVE: GPRS, Sismos, Plataforma Raspberry Pi 3B, interfaz web, Base de Datos.

ABSTRACT

Seismic movements are phenomena of nature that develop at any time. Something important is that Seismic movements are quantified with different parameters such as the seismic level measured in Richter scale. When a seismic movement is developed, is impossible to prevent it.

Due to this mentioned, there is a System which allows an analysis more exhaustive about these phenomena proposing the construction of a prototype that stores seismic dates whit the help of the GPRS technology and the Raspberry pi 3B platform.

In the first chapter, it is allowed to analyze the technology that will help to link different devices for they give a correct operation. The devices needed for the construction are: Module GSM/GPRS, Raspberry Platform Pi 3B, Seismic Motion Sensors, Solar Panels, Charge Controllers and batteries.

In the second chapter, the project will focus on the software for the methodical construction of the prototype. Also, determinate the useful equipment according to the need.

In the third chapter, there are a discussion of results and comparison of this system with another professional system type. However, probably in this System, there will have failures or even problems to improve them in the future. These problems can be the connectivity to a network towards the main equipment even instability in the measurement working with commercial type sensors but not specialized sensors.

The System has been implemented of this way, data collection, correct data storage and finally visualization in a web interface, notably the use of database allow that data has been gotten by sensors will be correct store. In this case, there a focus on the integration of all these tools for the construction of this system.

Keywords: GPRS, Raspberry Pi 3B Platform, web interface, Database, Python

1. INTRODUCCIÓN

Los sismos son generados por vibraciones de la tierra los cuales pueden ser de alta o baja magnitud. La tecnología ha permitido crecer cada día más estableciendo nuevos avances dentro de diferentes áreas de estudio o inclusive en combinación con ellas, como por ejemplo un Sistema de adquisición de datos sísmicos basado en tecnología GPRS y la plataforma Raspberry Pi 3.

Este prototipo permite tener una estimación sobre los movimientos sísmicos que puedan ocurrir, con la ayuda de este prototipo se espera establecer un estudio mediante el análisis de los datos sensados en beneficio de poder precautelar a la ciudadanía en general a futuro.

Los datos se toman mediante sensores de vibración, en este caso, sensores genéricos de vibración como lo son el SW-420, el mismo que pasa los datos a la plataforma Raspberry Pi 3B. Los datos son procesados para posteriormente transmitirlos y almacenarlos en la nube, estos datos son enviados desde la plataforma Raspberry pi 3B con la ayuda de un módulo de comunicación GSM/GPRS, el cual se conecta a la red 2.5G.

Los datos, después de ser almacenados en una Base de Datos en la nube se podrán visualizar de manera adecuada en una interfaz Web.

Cabe destacar que este es un prototipo el cual se lo implementa a manera de prueba por lo cual el mismo puede tener en ocasiones algún desperfecto ya sea por condiciones ambientales, polvo en los equipos, corrosión, entre otros factores que pueden perjudicar al buen funcionamiento del prototipo.

El sistema de energía provisto para este sistema se da mediante el uso de paneles solares, controlador de carga y batería, lo cual proporciona que el sistema funcione de manera autónoma ante las necesidades energéticas del mismo.

El prototipo establece un costo menor ante equipos similares, obviamente este sistema trabaja con equipos comerciales mas no equipos profesionales y de precisión, ante esto el sistema de medición y precisión posee algunas diferencias debido a que los dispositivos con los que se trabaja son de menor fiabilidad.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es:

- Construir un sistema para recolección e información de datos sobre movimientos telúricos.

Los objetivos específicos del Proyecto Técnico son:

- Describir los fundamentos teóricos y las herramientas disponibles para el uso y la gestión del servicio del sistema.
- Describir los elementos fundamentales de una solución para el sistema a implementar.
- Diseñar un sistema para recolección de datos sísmicos, a un bajo costo económico.
- Verificar la funcionalidad del sistema mediante la realización de pruebas.
- Presentar el costo referencial del sistema propuesto.

1.2 ALCANCE

Como parte inicial de este proyecto se procede a la construcción de un prototipo que permite la recolección de datos sísmicos, basado en un sistema que permitirá sensar movimientos telúricos dentro de un área establecida. El sistema principalmente estará constituido por dos sensores de vibración acoplados cada uno al Raspberry Pi 3. Entonces se establecerá tres Raspberry Pi 3 ubicados de manera independiente en lugares establecidos de forma arbitraria. La información sensada se la procesará y enviará a un servidor para poderla observar en una interfaz web de manera clara y concisa, dirigida a los usuarios del sistema. Estos datos serán almacenados para así poder tener una estimación adecuada ante estos fenómenos naturales.

En sí, en el proyecto se establecerá un sistema involucrado con la información de los elementos necesarios en el diseño y construcción, se realizará el presupuesto incluyendo todos los parámetros que involucrarán en la construcción del sistema. Por lo tanto, el costo será detallado para así poder establecer su beneficio económico.

1.3 MARCO TEÓRICO

1.3.1. MOVIMIENTOS SÍSMICOS

Los fenómenos naturales son cambios que se dan en la naturaleza, estos pueden afectar positiva o negativamente a los seres humanos. Uno de los fenómenos naturales que más llama la atención en nuestra cotidianidad son los sismos. El término “sismo” se emplea para describir el efecto de un movimiento súbito de la corteza terrestre que genera ondas elásticas, las cuales se propagan a grandes distancias [1].

Es importante el estudio de estos fenómenos sísmicos debido a que muchos traen bastante destrucción y pérdidas humanas, se debe tomar en cuenta que estos fenómenos naturales no se pueden evitar ni tampoco prevenir, pero se puede entender a través del uso de la tecnología, esto con el fin de comprender y explicar su actividad. La representación de un movimiento sísmico puede ver en la figura 1.1.

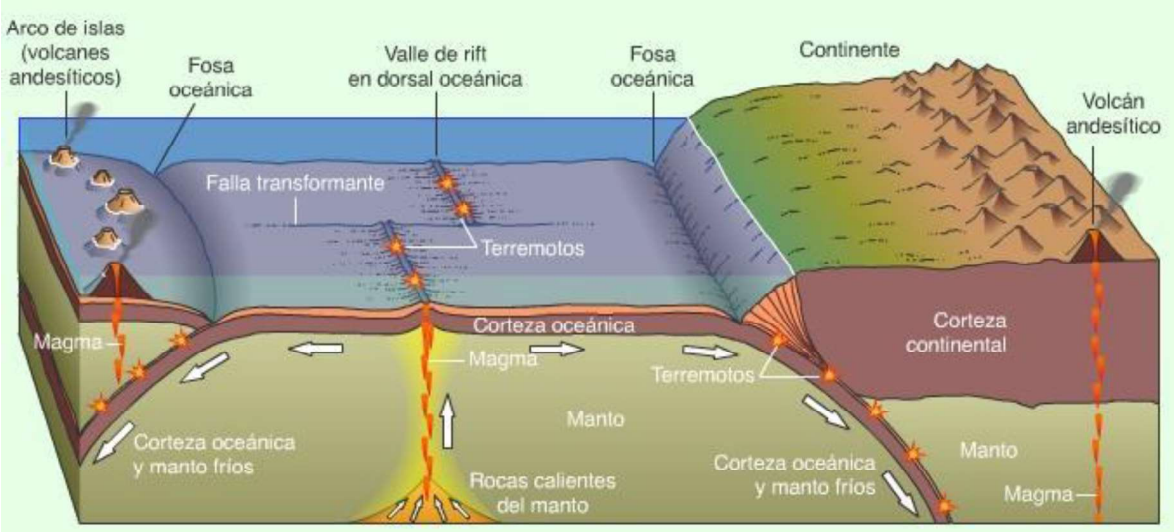


Figura 1.1. Movimientos de las placas tectónicas [2]

Los sismógrafos son equipos que se encargan de monitorear los movimientos telúricos de las placas tectónicas, estos pueden ser usados para localizar a precisión, en tres dimensiones, la fuente del terremoto, usando el tiempo que toma a las ondas sísmicas propagarse hacia fuera desde el epicentro, en el punto de la ruptura de la falla. Los sensores especializados para este tipo de detección de fallas son llamados geófonos estos perciben ondas sísmicas, es decir, ondas elásticas provocadas por la liberación de energía que se produce por los movimientos de las placas tectónicas. En los estudios sobre terremotos estos equipos son muy sensibles a los movimientos del terreno, pero tal

característica también implica un alto costo. En la Figura 1.2. se visualiza un sismógrafo de tipo eléctrico.

Figura 1.2. Sismógrafo Eléctrico [3]

1.3.2. SISTEMAS CELULARES

Actualmente los Sistemas Celulares se los considera como la estructura de redes de intercomunicación global en el mundo. Los sistemas celulares han acaparado gran interés, ya que proporcionan una extensa cobertura, pues se basan en la reutilización de frecuencias y manejan la división de celdas dentro de un mismo sitio geográfico. Esta división de celdas permite que se abastezca un área muy significativa en cuanto a la cobertura celular, no obstante día a día se va actualizando la tecnología conforme avanza el ritmo de vida digital existente, por lo cual es primordial crecer con las tecnologías existentes y evolucionar con estas hacia el futuro.

A través del tiempo la tecnología celular ha evolucionado desde su lanzamiento en los 80s con la primera generación pasando por la segunda generación, tercera generación, cuarta generación actualizándose gradualmente ofreciendo mejores servicios hasta llegar en la actualidad a la quinta generación, mediante la evolución se mejora varias características en cuanto al servicio que se ofrece por ejemplo velocidad de transmisión, ancho de banda, disponibilidad, cobertura, entre otras, Obviamente esto dependerá de la tecnología con la que se trabaja.

1.3.2.1. SEGUNDA GENERACIÓN (2G)

La segunda generación fue la primera en utilizar tecnología digital, la que proporcionó mejor calidad de voz, la transmisión de datos fue más rápida en comparación con el servicio de velocidad de la primera generación, se dio una enorme popularidad al envio de mensajes de texto y progresivamente las actualizaciones dieron como resultado la posibilidad del

acceso a internet, posteriormente se mejora con la llegada de la tecnología GPRS (*General Packet Radio Service, Servicio General de Paquetes vía Radio*).

La principal mejora que proporciona GPRS es la transmisión y recepción de datos, a continuación, se detallan las características más importantes de esta tecnología, las cuales se describen en la tabla 1.1.

Tabla 1.1. Descripción general de las características redes 2G [4]

ESPECIFICACIONES TÉCNICAS	
Tecnología	Digital
Velocidad	14kbps a 64 Kbps
Banda de frecuencia	- 850 - 1900 MHz (GSM (*Global System for Mobile communications, Sistema Global para las Comunicaciones Móviles*)) y 825 - 849 MHz (CDMA, (*Code Division Multiple Access, Acceso múltiple por división de código*)
Ancho de banda	canal - GSM divide cada canal de 200 kHz en bloques de 25 kHz El canal CDMA es nominalmente de 1,23 MHz
Modulación	TDMA *(Time Division Multiple Access, Acceso Múltiple por División de Tiempo*) y CDMA (*Code Division Multiple Access, Acceso múltiple por división de código*)
Conmutación	Conmutación de circuitos
Estándares	GSM, IS-95 (CDMA) - utilizado en América y partes de Asia, GSM (TDMA) originalmente en Europa, JDC (Celular Digital Japonés) (basado en TDMA), utilizado en Japón.
Servicios	Voz Digital, SMS, roaming internacional, conferencia, llamada en espera, retención de llamada, transferencia de llamadas, bloqueo de llamadas, número de identificación de llamadas, autenticación, facturación basada en los servicios prestados a sus clientes, cargos basados en llamadas locales, llamadas de larga distancia, llamadas con descuento, en tiempo real de facturación

VENTAJAS DE UNA RED 2G.

- Las ventajas de esta tecnología fueron considerables en cuestión de la seguridad a través de los sistemas de encriptación.
- Se obtuvo nuevas opciones ante los servicios de mensajes de texto.
- Las actualizaciones en los teléfonos móviles mejoraron el rendimiento llegando a proporcionar una mejor calidad ante el servicio prestado de las operadoras.
- Los usuarios gozaban de una tecnología que podía ofréceles acceso a navegar por internet.

DESVENTAJAS DE UNA RED 2G

- Notablemente, la desventaja más grande se dio ante el manejo de señales digitales que se transmitían desde el móvil, generalmente no alcanzaban los niveles de potencia requeridos para proveer la creciente demanda tecnológica de los usuarios.
- Se dan problemas de cobertura, el sistema funcionara lento e inclusive muchas ocasiones llega a caerse el servicio en su totalidad
- El servicio se veía decadente y dependía mucho de la cercanía a las redes de telefonía celular, así si un usuario se encontraba fuera del alcance, los servicios en su telefonía móvil llegaban a ser nulos.

1.3.2.3. SERVICIO GENERAL DE PAQUETES POR RADIO

Esta tecnología es la transición hacia la tecnología 3G (Tercera Generación), que comparte el rango de frecuencias de la red GSM, conocida como GPRS (*General Packet Radio Service, Servicio General de Paquetes vía Radio*).

Esta generación proporciona mejoras significativas a la tecnología con el objetivo de incrementar las tasas de transmisión de datos, la velocidad que se manejaba en esta tecnología llamada GPRS va desde los 56 kbps hasta los 115 kbps. Muchos operadores dentro de las telecomunicaciones procedieron a migrar a este tipo de tecnología para tener mayores prestaciones hacia sus usuarios finales.

La tecnología 2.5G posee algunos de los servicios que se incorporan en la expansión de la tecnología 3G razón por la cual los costos de migración para los proveedores serían menores ante un cambio radical de 2G a 3G.

En la tabla 1.2. se establece las características más importantes referente a la Generación Intermedia definida como 2.5G.

Tabla 1.2. Descripción General de las Características Redes 2.5G [4]

ESPECIFICACIONES TÉCNICAS	
Año	2003
Tecnología	Digital
Velocidad	115kpbs (GPRS) / 384 kbps (EDGE, (Enhanced Data Rates for GSM Evolution, Tasas de Datos Mejoradas para la Evolución del GSM))
Banda de frecuencia	850 -1900 MHz
Conmutación	Conmutación de paquetes
Estándares	Servicio General de Paquetes de Radio (GPRS) (Velocidades de datos mejoradas en GSM)
Servicios	Pulsar para hablar, multimedia, información basada en la web de entretenimiento, soporte WAP (Wireless Application Protocol, Protocolo de Aplicaciones Inalámbricas), MMS (Multimedia Messaging Service, Servicio de Mensajería Multimedia), SMS (*Short Message Service, Servicio de Mensajes Cortos*), juegos móviles, búsqueda y directorio, acceso a correo electrónico, videoconferencia.

1.3.2.4. CARACTERÍSTICAS DE LA RED GPRS

Esta tecnología permitió acceso a los usuarios a redes intranet mientras estos poseían movilidad, mediante la conexión a internet, pues fue muy importante en el aspecto corporativo ya que el usuario puede estar en un lugar diferente y poder revisar documentos importantes, finanzas, contratos. Todo esto con el respaldo de acceder a las bases de datos en donde se almacenan todos estos documentos referentes a una empresa corporativa.

- Se introduce el uso del protocolo IP (*Internet Protocol, Protocolo de Internet*)
- Sirve como ruta de migración hacia las redes 3G.
- Permite atender a múltiples usuarios en un mismo canal.

- Establece una nueva infraestructura dentro del manejo del protocolo de Internet.
- Genera el acceso mediante el Protocolo de Aplicaciones Inalámbrico (WAP), el SMS y MMS, acceso a Internet y correo electrónico.

1.3.2.5. INFRAESTRUCTURA DE UNA RED GPRS

GPRS es muy importante ya que en su época llegó a ser el sistema de conmutación de datos de mayor aceptación alrededor del mundo. En un principio fue desarrollado para Europa, pero fue adoptado por muchos más países. Los elementos más importantes en la infraestructura de la tecnología GPRS son:

- GGSN (*Gateway GPRS Support Node, Nodo de Soporte GPRS de puerta de enlace*) interconecta el sistema con redes de datos externas. El GGSN, se encarga además de proveer la salida a la PDN (*Packet Data Network, Red de Paquete de Datos*) dependiendo del APN (*Access Point Network, Punto de Acceso a la Red*), así como de la parte de facturación y aplicación de políticas y reglas de navegación que interconecta el sistema con redes de datos externas.
- SGSN (*Serving Gateway Support Node, Nodo de soporte de puerta de enlace de servicio*) se encarga de dar acceso, autentificación, asignación de calidad de servicio, cuando el usuario se ha conectado a la red de datos móviles, también lleva una parte de facturación de los roamers, proporciona conectividad a las BSS (*Base Station Subsystem, Subsistema de Estación Base*) de GSM.
- EL BG (*Border Gateway, Puerta de borde*) Proporciona seguridad y se encuentra ubicado en la conexión con la red troncal.

Otros elementos que son comunes en las redes de datos, dentro de esta tecnología se puede mencionar.

- DNS (*Domain Name System, Sistema de Nombres de Dominio*).
- CG (*Charging Gateway, Puerta de carga*) el cual consolida y procesa todos los datos pertinentes antes de pasar al proceso de facturación con el fin de alivianar la carga de procesamiento.

El Firewall que básicamente constituyen la barrera de seguridad su función es proteger a la red de accesos indeseados.

- El protocolo de comunicación de la red GPRS soporta tanto el intercambio de informaciones de control como de paquetes, PDP-PDU(Packet Data Protocol-Protocol Date Unit) entre el móvil y el Nodo que este se encuentre conectado,

La Figura 1.3. Muestra la infraestructura de los elementos de mayor importancia dentro de la red GPRS.

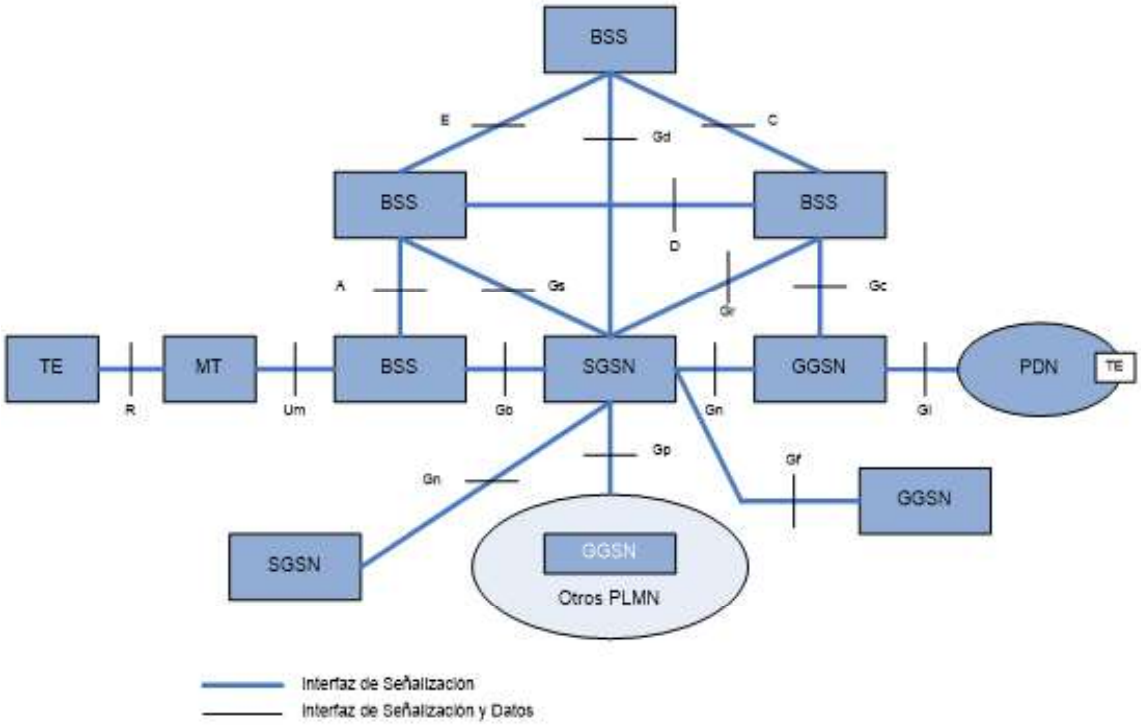


Figura 1.3. Elementos de una red GPRS [5]

1.3.2.6. INTERFAZ DE LA RED GPRS

Dentro de la Tecnología GPRS se introduce dos nodos GGSN y SGSN que actúan como interfaz lógica hacia las redes de paquetes externos conocido como router para lograr la conmutación de paquetes, no obstante, se debe hablar de las interfaces que actúan directamente, como son:

- **Interfaz Gn:** Se encarga de la comunicación entre el SGSN y el GGSN y el tráfico de señalización y datos planos.
- **Interfaz Gb:** Se encarga de establecer una comunicación diferente sobre el PLMN (Plublic Land Mobile Network, Red móvil terrestre pública).
- **Interfaz Ga:** Se encarga de la comunicación hacia el servidor de cobro
- **Interfaz Gc:** Actúa sobre el intercambiador de protocolo usando GTP
- **Interfaz Gi:** Se usa para la comunicación de la red GPRS con las redes de paquetes exteriores.

- **Interfaz Gp:** proporciona la misma funcionalidad que la interfaz Gn, también roporciona seguridad y encaminamiento como para proporcionar la conexión a la Inter-PLMN.
- **Interfaz Gr:** es aquella que existe entre el SGSN y el HLR (Home Location Register, o registro de ubicación base), para el intercambio de señalización.
- **Interfaz DHCP** actúa como medio de comunicación entre GGSN hacia DHCP

En la Figura 1.4. se visualiza el modo de operación para la comunicación de las diferentes interfaces.

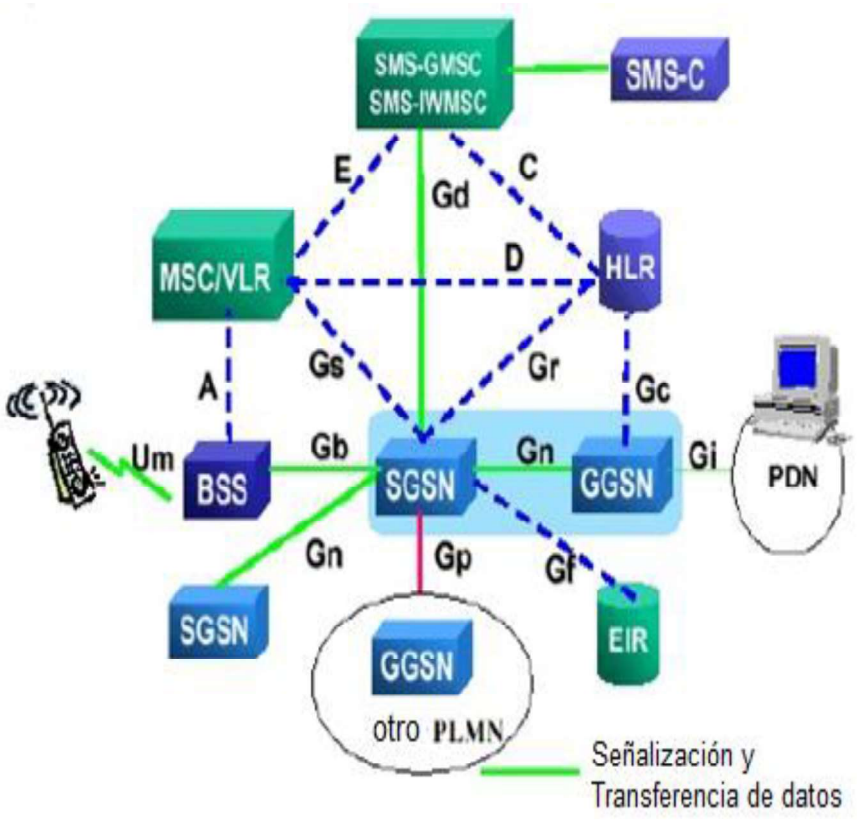


Figura 1.4. Interfaces de la red GPRS [6]

1.3.3. HARDWARE Y SOFTWARE DEL SISTEMA

Para tratar este tema se especifica el uso de la plataforma Raspberry Pi 3B y equipos físicos que permiten conectarse y trabajar conjuntamente con esta plataforma, mientras que en la parte de software se plantea el uso del lenguaje Python para realizar los programas que permiten el buen funcionamiento del prototipo.

1.3.3.1. SOFTWARE DEL SISTEMA

Software de sistema o software de base es la serie de programas preinstalados en el minicomputador denominado Rapberry Pi 3B. Su característica principal es gestionar tanto los recursos de hardware como los programas desarrollados, su ejecución es bastante privilegiada en comparación al resto del software, es por eso que se puede distinguir diferentes tipos de softwares para cada uso, como son el software de programación el cual permiten la construcción y diseño de otros programas que usan protocolos formales para la organización de algoritmos, que son de mucha utilidad para la programación en los proyectos.

- **Ruby. -** Este es un lenguaje de programación ágil, dinámico y de código abierto, es simple de utilizar debido a la sintaxis que maneja, un leguaje que va en crecimiento gracias a su framework de desarrollo web Ruby on Rails.[7]
- **Ruby on Rails (RoR). -** Ruby on Rails, un framework de aplicaciones web, el cual permite el manejo de código abierto, se fundamenta en el lenguaje de programación definido como Ruby, su entorno trabaja con el Modelo Vista Controlador [7]
- **Python. -** Este es un lenguaje de programación multiparadigma, orientado a objetos de mucha importancia, ya que trabaja fundamentándose en el manejo de código legible.
- **Raspbian. –** Es el sistema operativo de distribución libre para uso del equipo Raspberry pi 3B, es una distribución basada en Linux.

1.3.3.2. HARDWARE DEL SISTEMA

Dentro del Hardware del equipo hace referencia a los elementos electrónicos que permiten acoplarse al Raspberry Pi 3B para proporcionar el funcionamiento del prototipo, estos dispositivos son los siguientes:

- Sensores de movimiento, módulo GSM/GPRS, controlador de carga, batería y Panel Solar

También cabe mencionar que para poder controlar al Raspberry pi 3B como un minicomputador se hizo uso de:

- Teclado, mouse y monitor

Algunos de estos elementos que forman parte del prototipo y que se encuentran dentro del Hardware del sistema se pueden observar en la Figura 1.5.

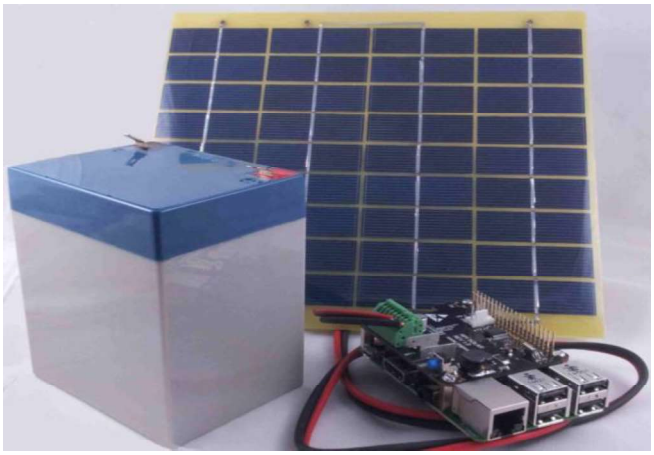


Figura 1.5. Ejemplo de Hardware del sistema [8]

1.3.3.2.1. MODELOS DE RASPBERRY PI

El Raspberry Pi, es un dispositivo tecnológico básico, pequeño, sin monitor, teclado o mouse, de hardware libre. Este es considerado como una de las computadoras más pequeñas del mundo, es muy popular por su bajo costo y la gran utilidad que puede proporcionar, detrás de este pequeño ordenador se encuentra la fundación que lleva su mismo nombre en el Reino Unido Raspberry Pi Foundation quien lanzó su primer prototipo para el año 2011 enfocado a la enseñanza de la informática en las escuelas, llegando a su comercialización en el año 2012.

- **Raspberry Pi 2 Modelo B**

Este modelo de Raspberry Pi 2B da un gran paso al cambiar el uso de la SD a Micro-SD. Se especifica sus características adicionales en la tabla 1.3.

Tabla 1.3. Características de Raspberry Pi 2 Modelo B [9]

Datos Técnicos	
Procesador	Broadcom BCM2836 4 núcleos @ 900 MHz
RAM	1 GB LPDDR2 SDRAM 450 MHz
USB	2 puertos USB 2.0
Salida de video	HDMI 1.4 @ 1920x1200 píxeles
Almacenamiento	MicroSD
Ethernet	1 puerto de 10/100 Mbps
Dimensiones	85,60 x 56,5 mm
Consumo	5V, 900mA, aproximadamente

Este modelo mejora en cuanto a prestaciones ante modelos anteriores, además se enfoca en la introducción de estas plataformas al área educativa asi como para usos prácticos de personal aficionado.

- **Raspberry Pi Zero W**

Anteriormente, salió al mercado una versión denominada Raspberry Pi Zero. La versión Raspberry Pi Zero posee un hardware limitado que representaba menos potencia, pero es más barato, no obstante, el Rapsberry Pi Zero se actualiza lanzando un nuevo modelo el Rapsberry Pi Zero W, que incorpora acceso Wi-Fi y Bluetooth. Figura 1.6.

Figura 1.6. Raspberry Pi W [9]

Tabla 1.4. Características de Raspberry Pi Zero W[9]

DATOS TÉCNICOS	
Procesador	Broadcom BCM2835 @ 1Ghz
Memoria	512 MB de memoria RAM LPDDR2.
Tipo de Tarjeta	Micro-SD.
Salida de vídeo	mini-HDMI a 1080p.
Tamaño	65mm de largo, 30mm de ancho y tan solo 5mm de grosor.

- **Raspberry Pi 3B**

En el 2016 el Raspberry da un enorme paso en su versión que cambia enormemente el modo de usabilidad generando mayores beneficios.

El Raspberry Pi 3B es uno de los que poseen mejores condiciones de trabajo debido a sus características las mismas que se observa en la tabla 1.5.

Tabla 1.5. Características de Raspberry Pi 3B [10]

ESPECIFICACIONES TÉCNICAS	
Conectividad	Via WIFI y Bluetooth
CPU + GPU	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz.
Memoria RAM	1GB de 4 núcleos
Conexión a video	Via HDMI.
USB	4 puertos USB 2.0
Frecuencia	1,2 GHz

El puerto ethernet permite conectividad en la red, además maneja cuatro puertos USB, los que da accesos a múltiples periféricos teclado, ratón, discos duros, adaptadores de wifi bluetooth GPS. Su estructura se visualiza en la figura 1.7.

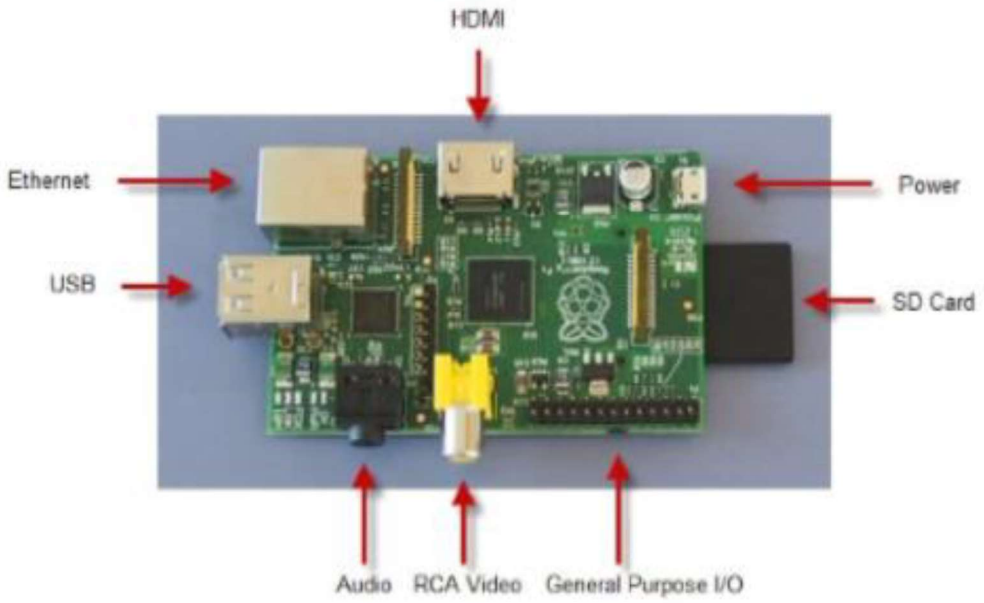


Figura 1.7. Raspberry Pi 3B [10]

- **Raspberry Pi 4**

Raspberry Pi 4, es una actualización de los anteriores modelos con un procesador de cuatro núcleos, un hardware actualizado, permite tres opciones de configuración: 1GB, 2GB o 4GB en la RAM, lo que se traduce en una mayor capacidad gráfica, soporta hasta dos pantallas

4K con un bajo consumo de energía viene con Gigabit Ethernet, junto con Wi-Fi 802.11ac para conexiones inalámbricas y Bluetooth. Ver Figura 1.8.

Figura 1.8. Raspberry Pi 4 modelo B [11]

Tabla 1.6. Características de Raspberry P4 Modelo B [15]

DATOS TÉCNICOS	
Procesador	ARM Cortex-A72 (v8) 64-bit SoC @ 1.5GHz (frecuencia de reloj)
Memoria	1 GB / 2 GB / 4 GB LPDDR4 SDRAM
Conectividad	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac Wireless LAN, Bluetooth 5.0
Resolución	4K a 60 fps
Puertos	GPIO 40 pines, 2 micro HDMI, 2 USB 2.0, 2 USB 3.0, CSI (cámara Raspberry Pi) DSI (pantalla tácil) Micro SD Conector de audio jack USB-C (alimentación)
Video y Sonido	Dual-Display Micro HDMI Ports which supports H 265 Decode for 4K Video @60p

1.3.3.2.2. MÓDULOS DE COMUNICACIÓN GPRS

Al hablar de módulos GSM/GPRS se establece que son dispositivos de comunicación pequeños y sencillos que nos permiten mediante el desarrollo de programas, proceder a ejecutarlos en el Raspberry Pi 3B lo que podría permitir controlar, comunicar, y usar redes GSM/GPRS con el fin de transmitir información. La transmisión de datos se da por códigos que utilizan un protocolo de comunicación, esto quiere decir que establece la comunicación hacia una central receptora utilizando como parámetros indispensables la dirección IP y un identificativo de puerto. Hay que tener en cuenta que la programación de este no debe tener ningún error o no se dará la comunicación.

El módulo GSM/GPRS A6 se considera como uno de los más pequeños del mundo cuenta con una interfaz estándar industrial ideal para la transmisión remota de datos GPRS y SMS soporta doble banda de red, se comunica a través del puerto UART. El módulo A6 ofrece un rendimiento en redes GSM/GPRS para SMS, datos y voz con un consumo de energía extremadamente bajo y a un nivel de temperatura aceptable para condiciones de trabajo que van de -30 ℃ a + 80 ℃.

Cabe destacar que este módulo tiene además otras características primordiales como:

- Cuenta con micro USB para alimentación externa
- También se puede instalar la tarjeta micro SIM la misma que debe ser adquirida a cualquier proveedor de una red de datos.

La estructura del dispositivo se puede observar en a la Figura 1.9.

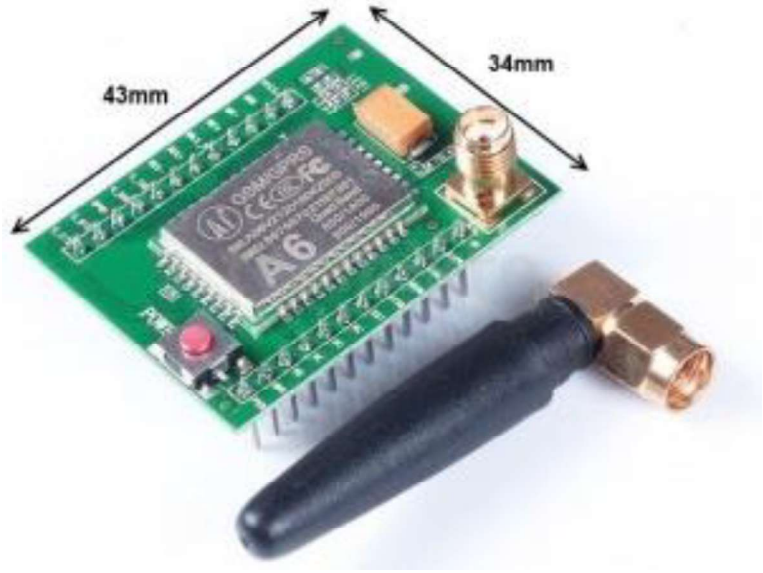


Figura 1.9. Módulo de comunicación GSM/GPRS A6 [12]

En la tabla 1.7. Se define las características más relevantes de este dispositivo, las cuales son de importancia en la construcción de nuestro sistema.

Tabla 1.7. Detalle y Especificaciones de Módulo de comunicación GSM/GPRS A6 [13]

DETALLE	ESPECIFICACIÓN
Frecuencia de trabajo:	Cuatri-Banda de 850/900/1800/1900MHz
Voltaje de funcionamiento:	4.5 ~ 5.5V DC
Consumo energía en espera	5 mA
Banda de Frecuencia:	780MHz ~ 960MHz, 1710MHz ~ 2170Mhz
Ganancia de la antena:	2,0 ± 0,7 dBi a 880 MHz, 2,0 ± 0,7 dBi a 1800 MHz - ROE <= 2,0 MHz ~ a 2,100 2,500 MHz de salida
Impedancia:	50 ohm
Tensión de lógica de interfaz:	3,3V
Dimensiones:	22,8×16,8×2,5mm
Peso:	19,4 g
Antena:	SMA (*SubMiniature version A, Subminiatura version A*)
Capacidad de tráfico de datos:	descarga 85.6 Kbps, subida 42.8 Kbps
Velocidad de transmisión:	115200bps y también puede ser ajustado por el comando Al Thinker

1.3.3.2.3. SENSORES DE VIBRACIÓN

Muchos de estos sensores de vibración se usan para diferentes aplicaciones industriales o el laboratorio. Estos sensores manejan diferentes tipos de sensibilidad es por lo cual que existen sensores de precisión, o de trabajo en laboratorio y comerciales. Estos sensores son aptos para ser utilizados en rangos térmicos de hasta +400°C, este valor puede variar dependiendo del fabricante y el tipo de sensor.

Los sensores de vibración generalmente son usados en mediciones sobre movimientos dentro de edificaciones, maquinas industriales y propiamente en movimientos telúricos leves, con un rango cercano al origen del movimiento, eso sí, dependiendo del equipo y la magnitud del sismo.

El sensor SW-420, es un circuito compuesto por un detector de vibración constituido interiormente de un resorte y un pequeño poste, por lo que cada vez que el sensor sea

sometido a un movimiento o golpe, su salida se verá afectada, los sensores de vibración sirven para registrar la aceleración, la velocidad o el desplazamiento, [14] el diseño de este sensor se observa en la Figura 1.10.

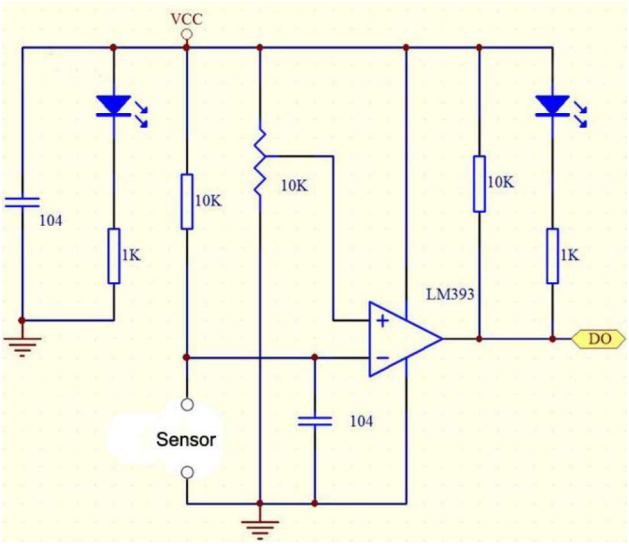


Figura 1.10. Diseño del Sensor de Vibración SW420 [14]

Tabla 1.8 Especificaciones Técnicas Sensor de Movimiento Sw- 420 [15]

ESPECIFICACIONES TÉCNICAS	
Modelo:	SW-420
Material:	placa PCB
Voltaje de funcionamiento:	3.3V ~ 5V
Corriente:	15mA
Salida de conmutación dígitos:	0, 1
Utiliza el comparador	LM393
Dimensiones:	(3,8 cm x 1,3 cm x 0,7 cm)
Peso:	0,18 oz (5 g)

1.3.3.3. SISTEMA DE ENERGÍA

La energía solar es aquella que proviene gracias a tratamientos mediante procesamientos y adquisición de la energía del sol, ante el aprovechamiento de la radiación electromagnética que este pueda proporcionar y mediante diferentes equipos que permiten la transformación en energía eléctrica. Figura 1.11.

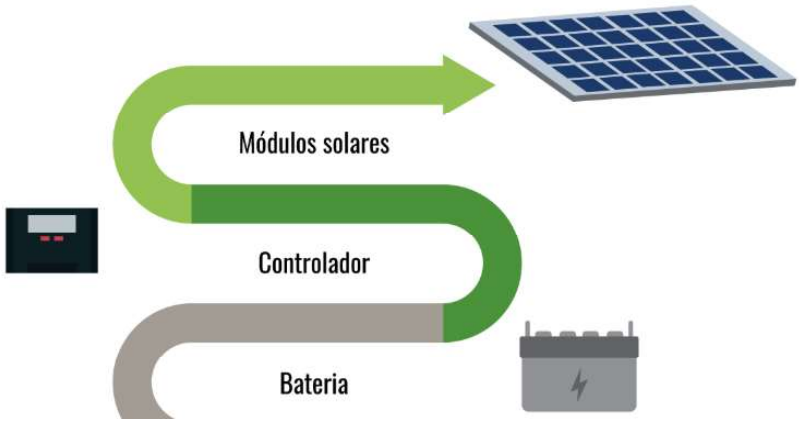


Figura 1.11. Elementos del Sistema de energía solar

Este sistema permite hacer uso de la energía receptada a partir de la radiación solar, refiriéndose al tipo de energía proveniente del sol, siendo está una manera ecológica de administrar la energía solar, la cual no contamina al ambiente, es inagotable y abundante. Se considera que la cantidad de energía que el sol provee a la tierra es 10 mil veces mayor a la que se consume en un solo día en toda la superficie terrestre, al ser un tipo de energía renovable en la actualidad el único elemento que puede trasformar la radiación solar a energía consumible es el panel solar, el mismo que tiene diferente mecanismo de acuerdo a la necesidad y utilidad, es asi que se puede encontrar en el mercado los siguientes mecanismos según el usos de los paneles solares.

A.- Captadores solares térmicos que se emplean para la energía solar, Figura 1.12.

Figura 1.12 Captadores solares Térmicos [16]

B.- Módulos fotovoltaicos utilizados para la energía solar fotovoltaica. Figura 1.13.

Figura 1.13 Captadores solares fotovoltaicos [17]

C.- Paneles solares termodinámicos utilizados comúnmente ya que la ventaja más grande está en su costo y su utilidad sin importar el estado climático. Figura 1.14.

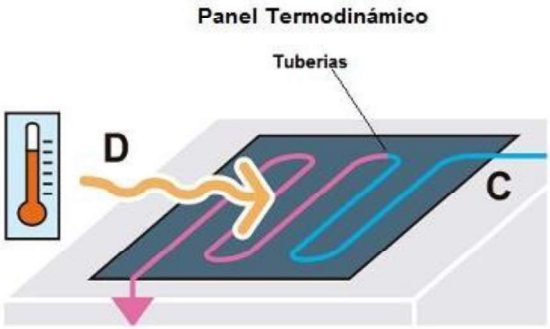


Figura 1.14 Panel solar termodinámico [18]

SISTEMA DE ENERGIZACIÓN MEDIANTE PANEL SOLAR

De estos tres mecanismos encargados de captar la energía solar antes mencionados se analiza a profundidad los módulos fotovoltaicos, que son utilizados para receptar energía solar fotovoltaica. Se empieza por definir que es el efecto fotovoltaico, el mismo que se refiere a la conversión de la radiación solar a electricidad consumible. El método de funcionamiento se puede observar en la Figura 1.15.

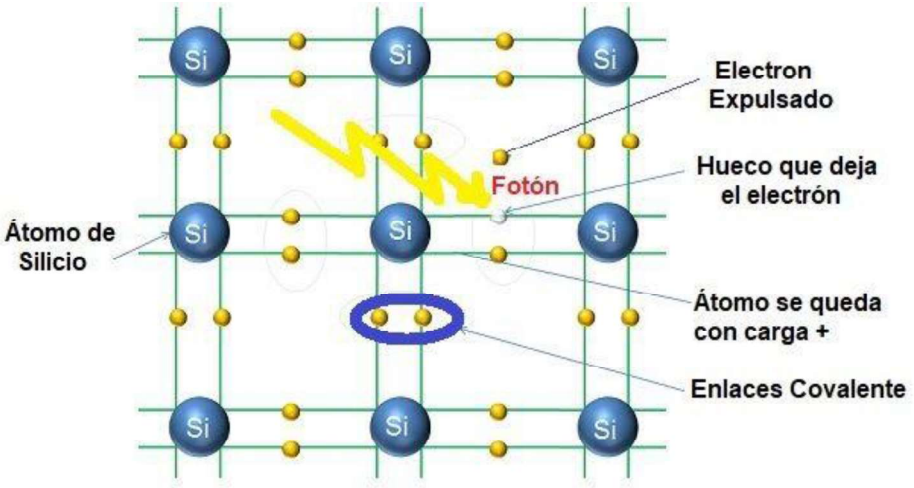


Figura 1.15. Funcionamiento de Efecto fotovoltaico [19]

Su funcionamiento se basa en una reacción nuclear de fusión y debido a las condiciones de intensa gravedad a las que está sometido. El sol que está compuesto principalmente por hidrógeno y en condiciones específicas se funde para producir helio. [20]

Una celda solar o célula solar, es una placa que puede variar en sus dimensiones, está formada de silicio cristalino. Debido a esta constitución permite convertir la luz del Sol en energía eléctrica gracias a un proceso denominado Efecto Fotoeléctrico, por tal motivo se menciona que un panel solar no es nada más que un conjunto de celdas solares unidas dentro de una gran placa, la cual las contiene. Si una celda solar convierte la energía del Sol en electricidad, un panel solar convierte mucha más energía que una sola celda solar. Las celdas se conectan unas con otras en serie dentro del panel.

Las células fotovoltaicas elaboradas de Silicio puro son montadas sobre módulos solares encargados de conseguir un voltaje adecuado, el panel solar está constituido por un marco de aluminio y cristal. Las celdas solares poseen un encapsulado, y otro encapsulado la cubierta posterior, también poseen la caja de conexiones, todo esto constituye el panel de energía solar, cabe destacar que son antireflectivos y antiadherentes con el fin de protegerlos de las condiciones ambientales, al hablar de la instalación del panel solar se debe tomar en cuenta otros mecanismos externos para su funcionalidad como lo son una batería de almacenamiento de la energía y un regulador de carga solar. Figura 1.16.

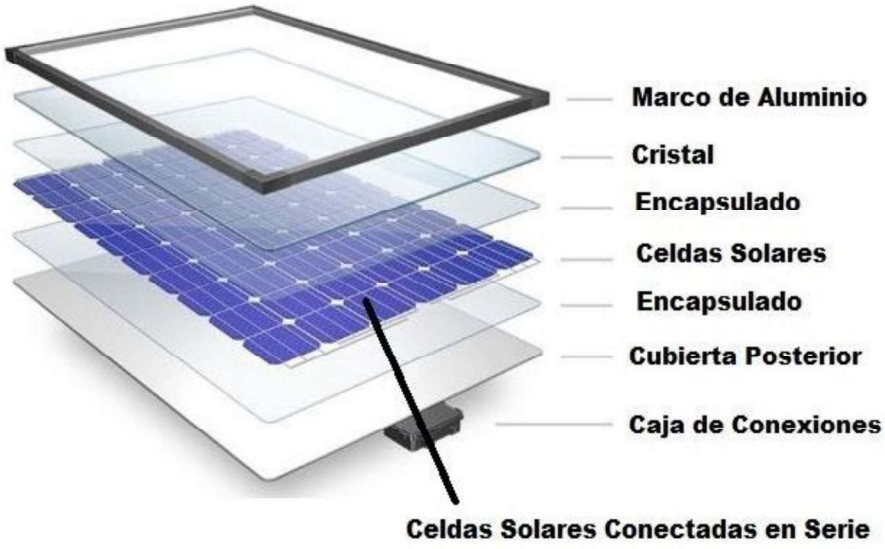


Figura 1.16. Partes de un panel solar fotovoltaico [21]

1.3.3.4. PLATAFORMA HEROKU

Heroku, es una plataforma que ofrece servicios de computación en la nube [22], por lo tanto, se define a la nube como un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet. La plataforma Heroku proporciona beneficios en el manejo de servidores, al cual se pueden agregar servicios de administración de base de datos, mensajería o instalar aplicaciones de diferentes lenguajes de programación como son Python, Java, PHP, SQL etc. Ahora en cuanto a la infraestructura de Heroku se establece que utiliza AWS *(Amazon Web Services, Servicios Wed de Amazon*) y el servicio se da por medio de contenedores Linux, que son denominados “dynos”, los cuales se encuentran aislados entre sí y cuentan con una buena capacidad de procesamiento, a esto se le añade la posibilidad de add-ons que permite agregar funcionalidades a dichos contenedores, si se habla de los comandos Heroku se puede instalar en Windows, Linux y Mac ya que la integración es transparente.

Heroku en general se define como, una de las plataformas como servicio más utilizadas en la actualidad, pues permite con facilidad el manejo de los servidores y la relación que se les pueda dar ante características de escalamiento, administración y manejo de configuraciones para el funcionamiento.

A continuación, en la Figura 1.17. se observa la cuenta de Heoku ya activada para proporcionar los servicios en la nube

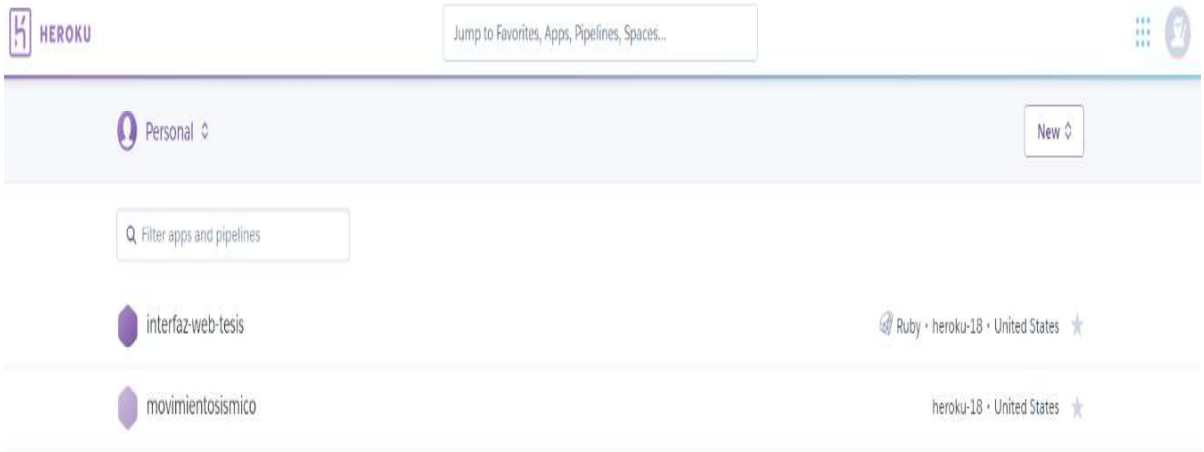


Figura 1.17. Interfaz de Heroku

1.3.3.5. POSTGRETS SQL

Es un sistema que permite la gestión de base de datos con características de orientación a objetos, en donde se puede trabajar con funciones, restricciones, herencia, tipos de datos, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD (*Berkeley Software Distribution, Distribución de software de Berkeley*), un servidor de base de datos muy usado ya que es de tipo open source dirigido por una comunidad de desarrolladores orientado a la organización y buen funcionamiento del mismo abierto a todo tipo de proyecto que use una base de datos denominado (PostgreSQL Global Development Group). Ver Figura 1.18.

Entre sus características se destaca las siguientes:

- Permite escalabilidad
- Trabaja con bases de datos más potentes y robustas del mercado.
- El uso de rollback, subconsultas y transacciones mejoran su funcionamiento.
- La capacidad de almacenamiento de procedimientos en la base de datos es óptima y de alto nivel se equipará a Oracle.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Copias de seguridad en caliente (Online/hot backups)
- Múltiples métodos de autentificación (cobian, backup4all con licencia)

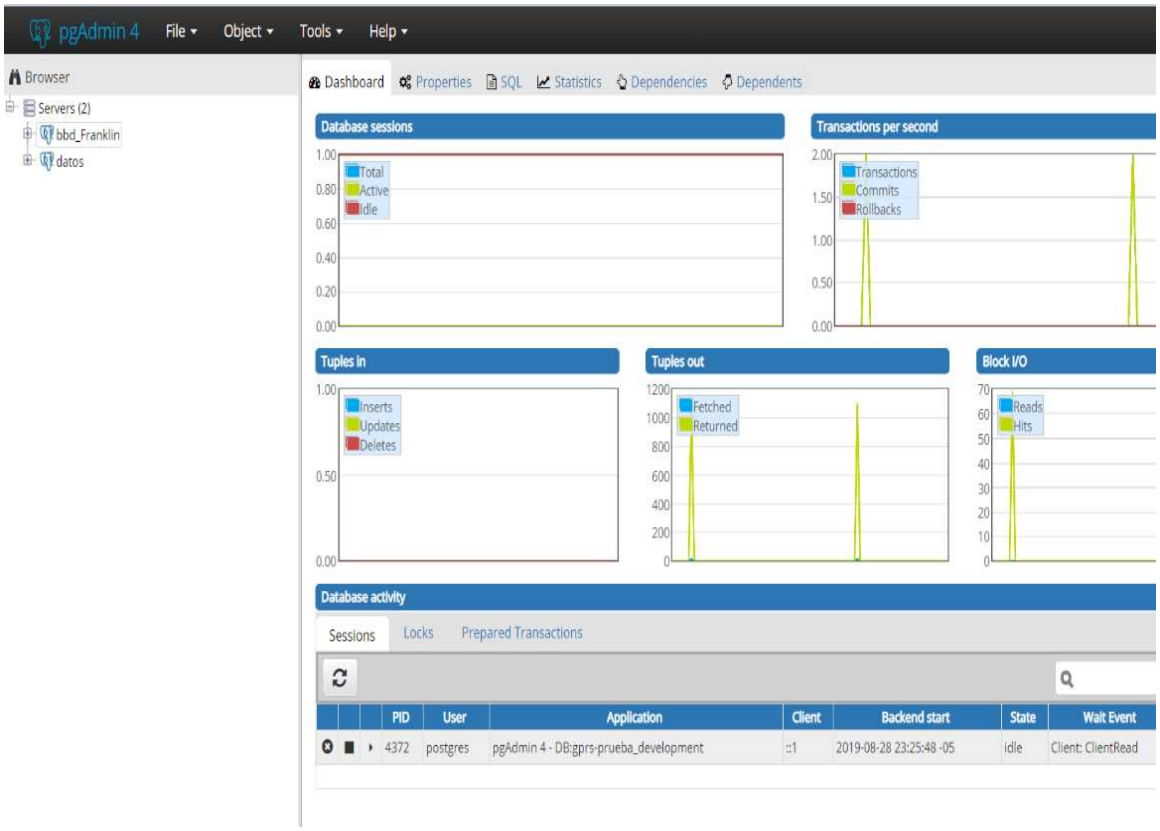


Figura 1.18. Sistema de Gestión de Base de Datos pgAdmin 4

1.3.3.6. MODELO DE REFERENCIA ISO/OSI

El modelo OSI (*International Organization for Standardization, Organización internacional para la estandarización*) se basa en siete niveles de abstracción que a su vez trabajan con sus propias funciones específicas de operación lo que hace posible la intercomunicación entre distintos protocolos, los niveles se distribuyen del siguiente modo:

1. **Nivel 1 física:** esta capa se encarga de gestionar la parte física todo lo referente a transmisión de bits, manejo de señales eléctricas y cableado.

2. **Nivel 2 enlace de datos:** se encarga del direccionamiento físico de los datos, el acceso al medio y en especial de la detección de errores en la transmisión.

3. **Nivel 3 red:** se encarga del enrutamiento entre dos o más redes conectadas.

4. **Nivel 4 transporte:** se encarga del transporte de los datos que se encuentran dentro del paquete de transmisión de origen hacia su destino dependiendo con el tipo de protocolo que se emplee.

   Si trabaja con el protocolo UPD (*User Datagram Protocol, Protocolo de Datagramas de Usuario*) orientado al envío sin conexión, si trabaja con el protocolo TCP

(*Transmission Control Protocol, Protocolo de Control de Transmisión*) orientado a la conexión.

Los siguientes niveles están orientados a la aplicación

5. **Nivel 5 sesión:** en este nivel se puede controlar y mantener activo el enlace para su transmisión lo que asegura que la conexión se mantenga hasta que se finalice
6. **Nivel 6 presentación:** se encarga de la representación de los datos y que estos sean de forma sencilla y entendible para el usuario.
7. **Nivel 7 aplicación:** aquí los usuarios pueden interactuar con el sistema permitiendo que se pueda ejecutar acciones en los comandos.

La Figura 1.19. muestra la representación del Modelo de Referencia ISO/OSI

Figura 1.19. Modelo de Referencia ISO/OSI [23]

1.3.3.7. ARQUITECTURA TCP/IP

El Protocolo de control de transmisión (TCP) es un protocolo utilizado para todos los nodos conectados a internet de forma que estos se puedan comunicar entre sí de manera fiable. Se trata de un protocolo orientado a la conexión que junto con el protocolo IP ha servido

de base para el modelo TCP/IP [24]. La arquitectura TCP/IP combina varias capas del modelo ISO/OSI en una única capa, y también deja de utilizar determinadas capas, por tal situación esta arquitectura se compone de cuatro niveles.

- Un primer nivel de acceso a la red: este nivel está compuesto por la capa Física y la capa de Red del modelo de referencia ISO/OSI.
- Un segundo nivel de red: ofrece un servicio de transmisión de la información entre dos puntos remotos de la red, no orientado a conexión y sin fiabilidad. Dentro de este nivel trabaja el protocolo IP (*Internet Protocol, Protocolo de Internet*).
- Un tercer nivel de transporte: suministra un transporte de información entre procesos funcionando en estaciones remotas, con o sin fiabilidad. En este nivel trabaja el protocolo TCP.
- Finalmente, el cuarto nivel o nivel de aplicación, nos da la posibilidad de abrir y controlar una sesión con un nodo remoto para transferir información, Este nivel de la arquitectura TCP/IP engloba aproximadamente los servicios de sesión, presentación y aplicación del modelo ISO/OSI.

Para poder realizar cada una de las funciones en cada nivel o capa se habla de una pila de protocolos. En una pila de protocolos, cada nivel soluciona una serie de problemas relacionados con la transmisión de datos, y proporciona un servicio bien definido a los niveles más altos. Los niveles superiores son los más cercanos al usuario y tratan con datos más abstractos, dejando a los niveles más bajos la labor de traducir los datos de forma que sean físicamente manipulables.

A continuación, se presenta una comparación de capas del modelo ISO/OSI y niveles de la arquitectura TCP/IP y la pila de protocolos. Figura 1.20.

Ahora se procede a la descripción de algunos protocolos de importancia que son usados en la realización del prototipo. Los protocolos de nivel de enlace de datos se encargan de codificar los paquetes recibidos del nivel de red para que puedan ser enviados a través del medio físico como señales electromagnéticas.

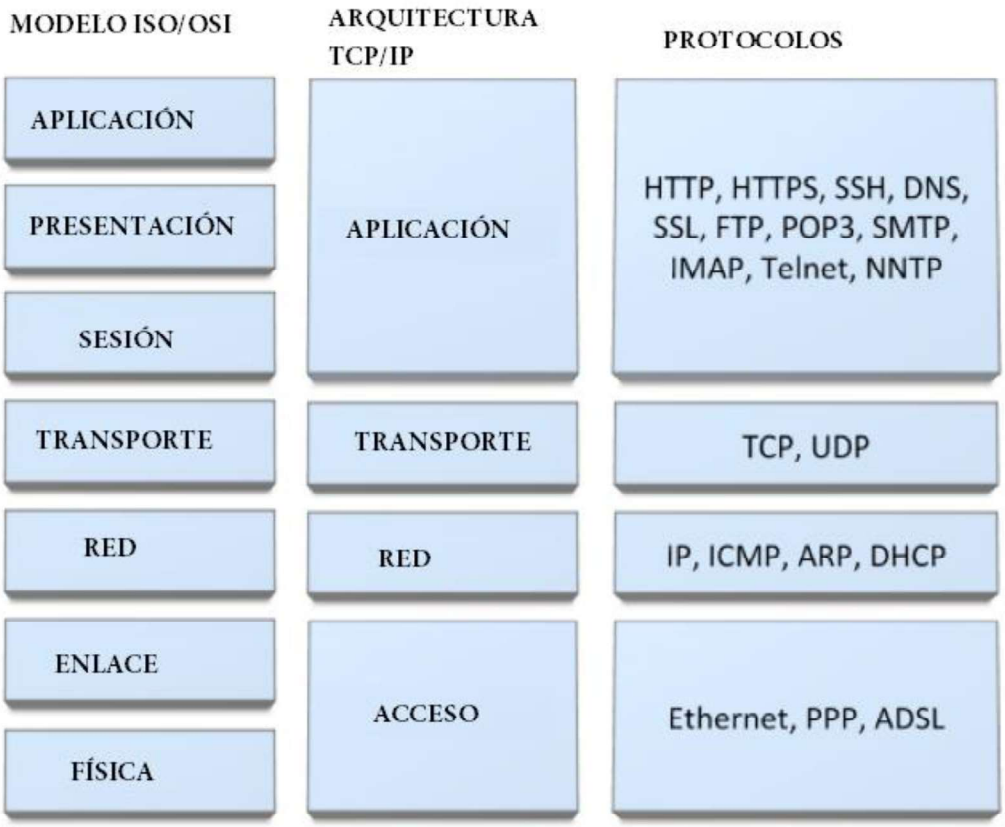


Figura 1.20. Modelo ISO/OSI y Arquitectura TCP/IP

- **Protocolo de enlace PPP**

El protocolo PPP (Point to Point Protocol, Protocolo Punto a Punto) permite encapsular tramas de datos listos para ser transmitidos mediante la conexión directa a través de cables seriales líneas telefónicas líneas troncales teléfonos celulares. Mantiene la compatibilidad con el hardware que le admite tiene tres componentes principales

- Entramado de estilo de HDLC (*High-Level Data Link Control, Control de Enlace de Datos de Alto Nivel*)): transporte de paquetes.
- Protocolo de control de enlace (LCP): conexión de paquetes.
- Familia de protocolos de control de red (NCP): configurar protocolos

- **Protocolos de capa de red y transporte.**

IP: se coloca en a capa tres de modelo ISO/OSI y su principal función es el direccionamiento de los paquetes; El protocolo IP no es fiable porque no asegura la entrega de datagramas. No está orientado a conexión porque el protocolo IP no

mantiene la situación de los datagramas sucesivos enviados, de forma que los datagramas pueden llegar al destino duplicados, en orden incorrecto, etc.

TCP: se encarga del transporte en la capa cuatro del modelo OISO/SI, establece control flujo, y conexión, el trasporte de este se da por datagramas mediante dirección IP y puerto en conclusión se encargan de la comunicación de entre las redes. En sí, da un servicio orientado a conexión para el transporte fiable de datos extremo a extremo, es decir, capaz de asegurar la entrega de información sin errores.

- **Protocolo de Aplicación HTTP**

HTTP (*HyperText Transfer Protocol o Protocolo de Transferencia de Hiper Textos*), este protocolo posibilita la comunicación dentro de un mismo "idioma", es el intermediario entre las peticiones del cliente y las respuestas del servidor en la Internet para que se puedan dar criterios de sintaxis y semántica informática, definido como el protocolo de transmisión de información de la World Wide Web, es decir, el código que se establece para que el solicitante y el que contiene la información solicitada puedan comunicarse.

El servidor brinda una respuesta estructurada de modo puntual y dotada de una serie de metadatos, que establecen las pautas para el inicio, desarrollo y cierre de la transmisión de la información. [25]

2. METODOLOGÍA

En esta sección se explica la forma en la que se desarrolló el proyecto. Este proyecto se basa en generar un prototipo, el cual se fundamenta en tecnologías existentes, tanto en la parte software como de hardware, se trata de minimizar los costos para la construcción y funcionamiento del sistema.

En el desarrollo de este prototipo se usa documentación de hardware y software libre, que generalmente se puede encontrar en los diferentes sitios oficiales.

El desarrollo del proyecto pasará por tres etapas fundamentales:

- Requerimientos del Prototipo
- Diseño del Prototipo
- Implementación del Prototipo

2.1. REQUERIMIENTO DEL PROTOTIPO

1.1.1 REQUISITOS DEL PROTOTIPO

Para el desarrollo del prototipo se considera la construcción como un pequeño equipo, cuyo funcionamiento sea básico y especifico, de tal manera se especifica los siguientes puntos a cumplir:

- El prototipo debe tener la capacidad de sensar datos sobre movimientos sísmicos.
- Los datos deben ser procesados por la plataforma Raspberry Pi 3B en un contexto entendible para el usuario final.
- Se deberá establecer una conexión a la red GPRS para poder enviar los datos procesados.
- Se trabaja con tres plataformas Raspberry Pi 3 a la cuales van conectados dos sensores por cada plataforma.
- La información recolectada se almacena en una base de datos la misma que se encuentra como servicio en la nube debido a la cantidad de datos que se va a recolectar.
- Se crea una interfaz web para poder visualizar los datos que se han sensado y procesado por el Raspberry Pi 3, los cuales están almacenados en la base de datos.

1.1.2 VISIÓ GENERAL DEL PROTOTIPO

Se procede a realizar el prototipo con la ayuda de tres equipos determinados por cada Raspberry pi 3B, en los que se conectan dos sensores, los mismos que recolectan los datos, al Raspberry pi 3B también se encuentra conecta el módulo GSM/GPRS el cual permite establecer la comunicación con la red 2.5G (GPRS), basada en los protocolos TCP e IP. La conexión a la red GPRS se la realiza mediante el protocolo PPP, el cual toma los datos provenientes de capas superiores, y una ves debidamente configurado y levantado el servicio pasa dichos datos a la capa inferior para poder ser trasmitidos.

La información recopilada se sube a una base de datos levantada en la nube para luego poder ser visualizada en la interfaz web desarrollada.

El sistema se basa en el diagrama de bloques, que se muestra en la figura 2.1.

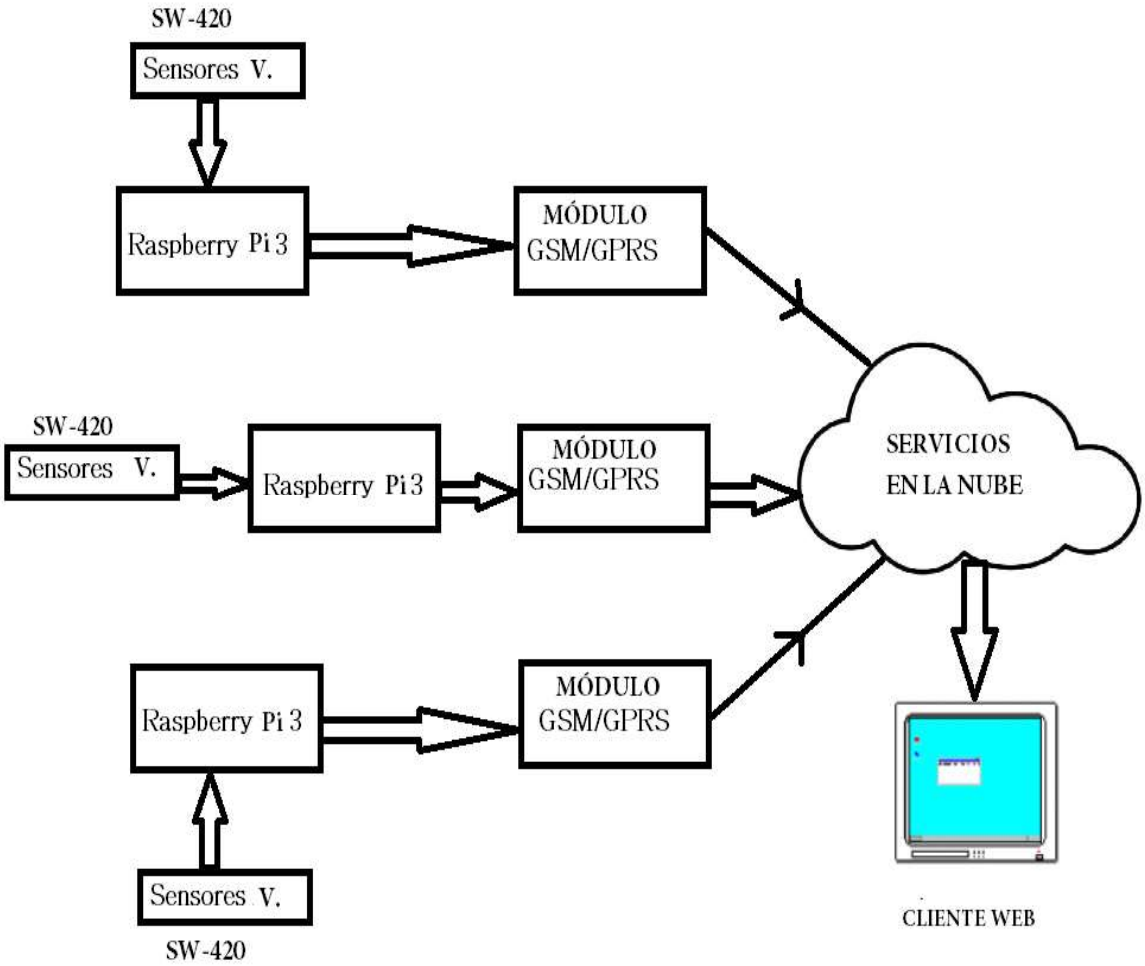


Figura 2.1. Diagrama de Bloques del Prototipo

2.2. DISEÑO DEL PROTOTIPO

En la parte del diseño se enfoca prioritariamente al manejo adecuado del Hardware y la parte de software, con el fin de poder explicar de mejor manera la construcción del prototipo.

2.2.1. CONSTRUCCIÓN DEL PROTOTIPO

En el diseño del prototipo se explica las partes más importantes del sistema construido, el mismo que se basa como herramienta principal en la plataforma Raspberry Pi 3B, el módulo GSM/GPRS, los sensores de movimiento y el manejo de la comunicación GSM/GPRS. El diagrama de los dispositivos que se usan en el prototipo se presenta en la Figura 2.2

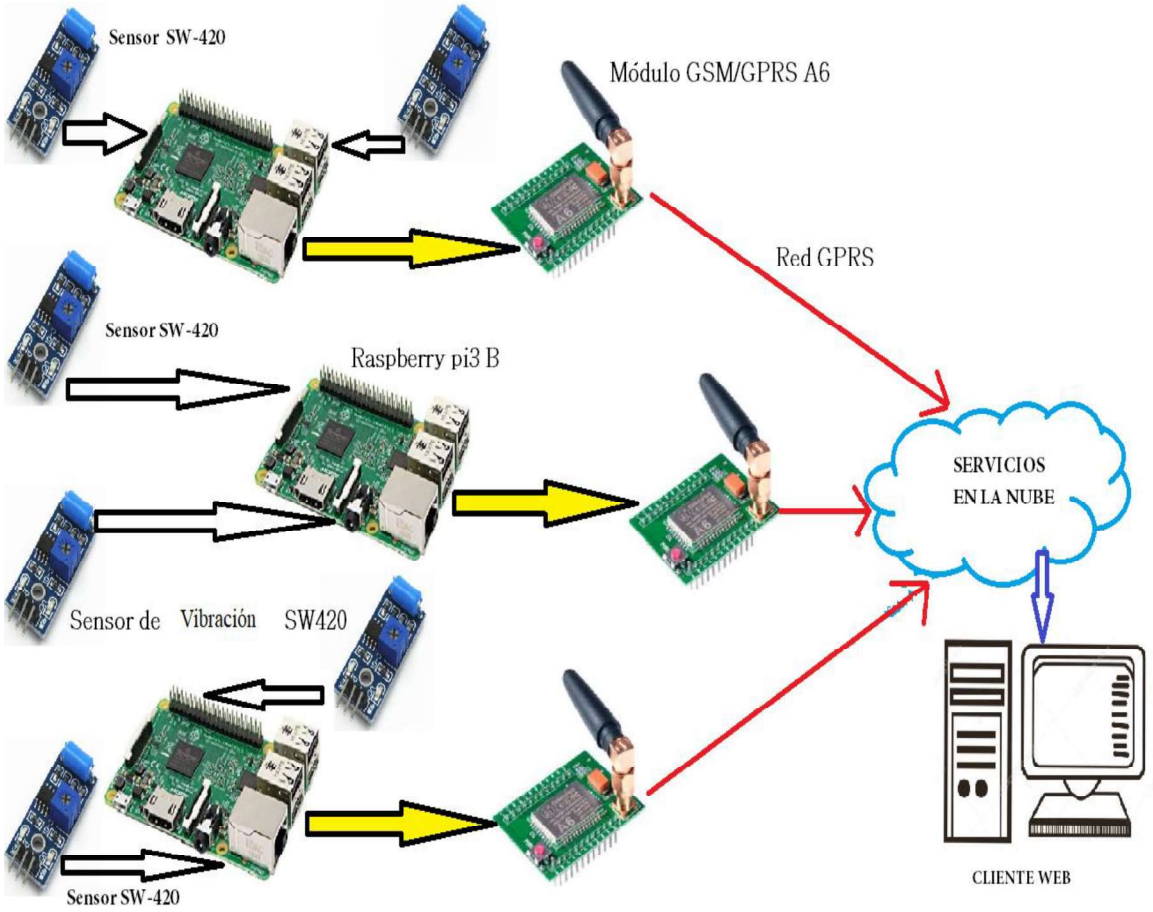


Figura 2.2. Diagrama esquemático del Hardware Utilizado.

En esta parte se empieza a explicar sobre el uso y configuración de la Plataforma Raspberry Pi 3B en cuanto concierne a la realización del prototipo, por tal razón se debe conocer algunos datos importantes que se presentan a continuación.

2.2.2. CONFIGURACIÓN PLATAFORMA RASPBERRY PI 3B

El Raspberry Pi 3B funciona con un sistema operativo Raspbian el cual debido a su característica de Open Source, permite trabajar en aplicaciones definidas en la plataforma para el debido funcionamiento del prototipo, permitir procesar las señales de los movimientos telúricos sensados mediante la realización de un determinado programa.

La capacidad de procesamiento en el Raspberry Pi 3B y el tiempo de conexión a la red GPRS es de gran importancia al analizar los datos, pues la información debe ser presentada lo más pronto posible ante el acontecimiento de un movimiento telúrico, es asi que el rendimiento del prototipo se establece ante estos parámetros.

El sistema operativo se instala en una Micro SD, con la ayuda del programa Win32 Disk Imager y SDFormatter (figura 2.3), se descarga del sistema Operativo Raspbian de la página oficial:

https://www.raspberrypi.org/downloads/

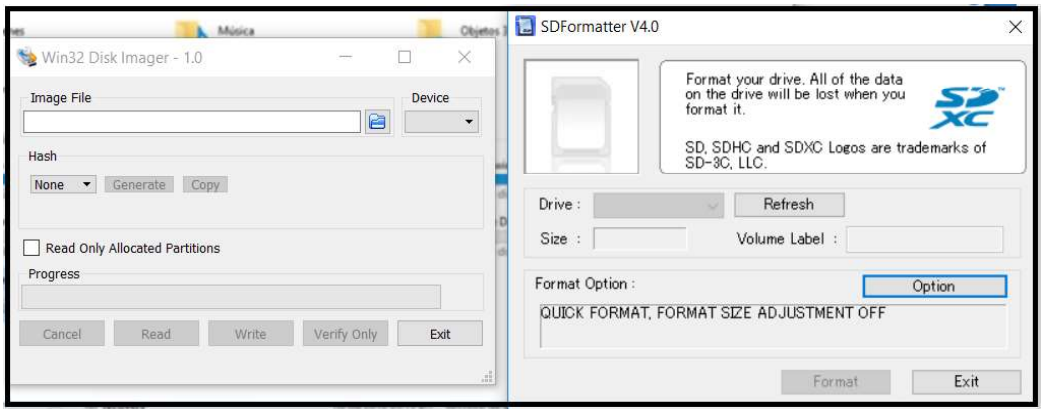


Figura 2.3 Aplicaciones usadas para instalar el Sistema Operativo

En el sistema operativo se procede a habilitar la configuración para comunicación serial, lo cual permite manejar el módulo GSM/GPRS A6.

El módulo GSM/GPRS A6 trabaja mediante unca conexión PPP, la misma que establece un nodo de comunicación mediante una interfaz serial, esta conexión permite la comunicación entre el módulo usando un chip telefónico y la red de telefonía celular, por lo tanto, es ahí donde se procede a realizar una configuración al Raspberry Pi 3 para que este pueda manejar la comunicación serial, se procede a cambiar la interfaz UART a ttyS0, esta se convierte en un nuevo puerto de comunicación y permite manejar la interfaz serial.

Una vez realizado este cambio se empieza abriendo la consola del Linux instalado en nuestro Raspberry Pi 3B y se ejecuta el comando: ***#dmesg | grep tty***

El cual permite verificar los puertos seriales conectados. Después de haber realizado esta verificación se procede al cambio en la configuración para que el sistema no use al UART por lo que se ejecuta: *#sudo raspi-config*

Ejecutando esta instrucción, se ingresa al panel de configuraciones del Raspberry pi 3B como se ve en la figura 2.4. Aquí se selecciona la opción 5 definida por las opciones de interfaz.

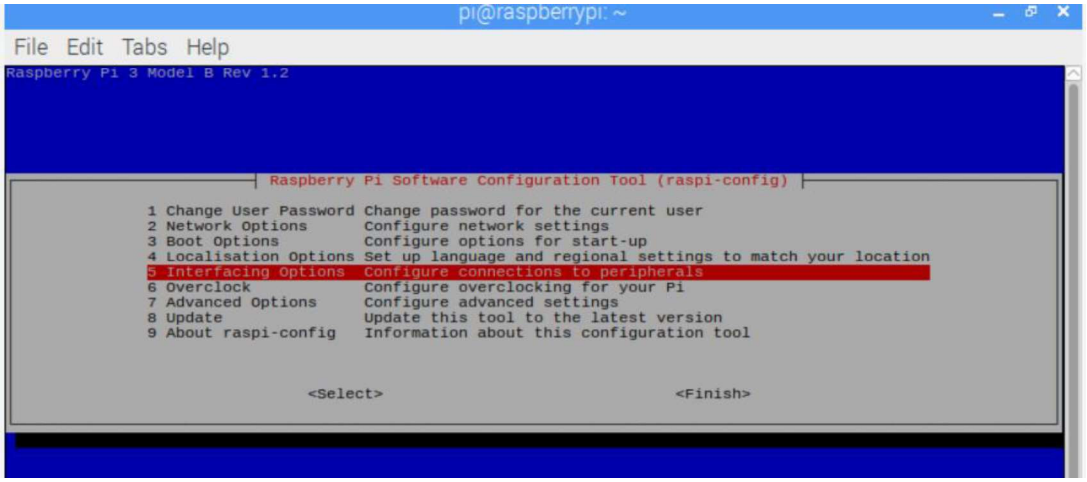


Figura 2.4. Panel de Configuración Raspberry Pi 3B

A continuación, se abre una nueva ventana con diferentes opciones, seleccionamos la opción P6 Serial, la misma que permite establecer la comunicación y a la vez la transmisión de los datos. Como se muestra a continuación en la Figura 2.5.

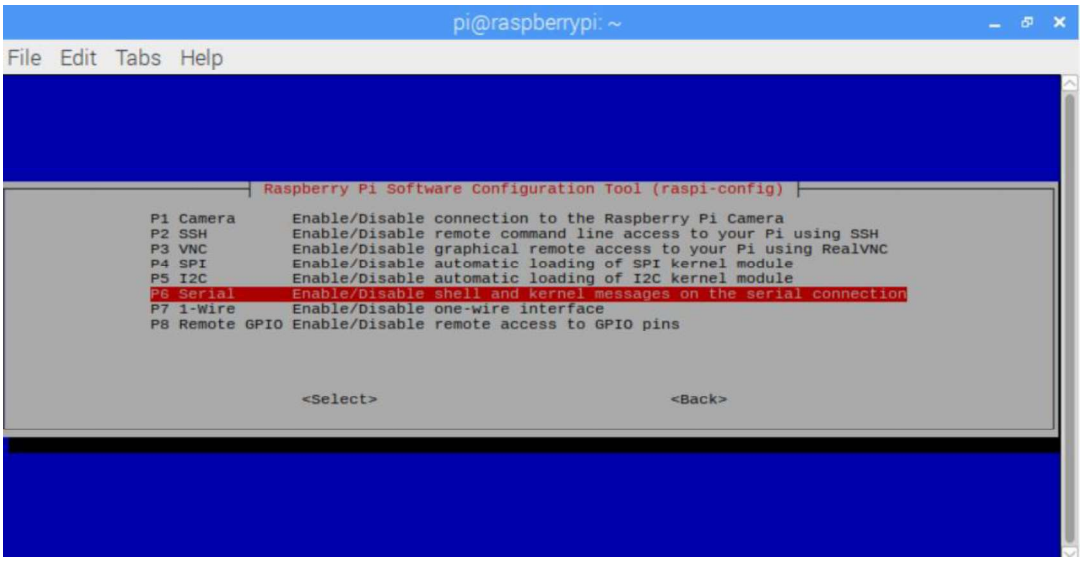


Figura 2.5. Configuración Serial

Después de esto, se debe mantener habilitado el Hardware de la UART. Una vez hecho los ajustes necesarios se sale de este panel de configuración y se procede a reiniciar el Raspberry Pi 3B. Luego de reiniciado el sistema operativo se procede a abrir una consola para hacer un cambio en un archivo de configuración. Por lo tanto, se ejecuta el siguiente comando: *#sudo nano /boot/config.txt*

En dicho archivo se configura el reloj de la UART, para proceder a añadir las siguientes líneas de configuración:

```
core_freq=250
enable_uart=1
```

Esta configuración debidamente implementada se puede visualizar en la Figura 2.6, con los cambios que se realizó adecuadamente.

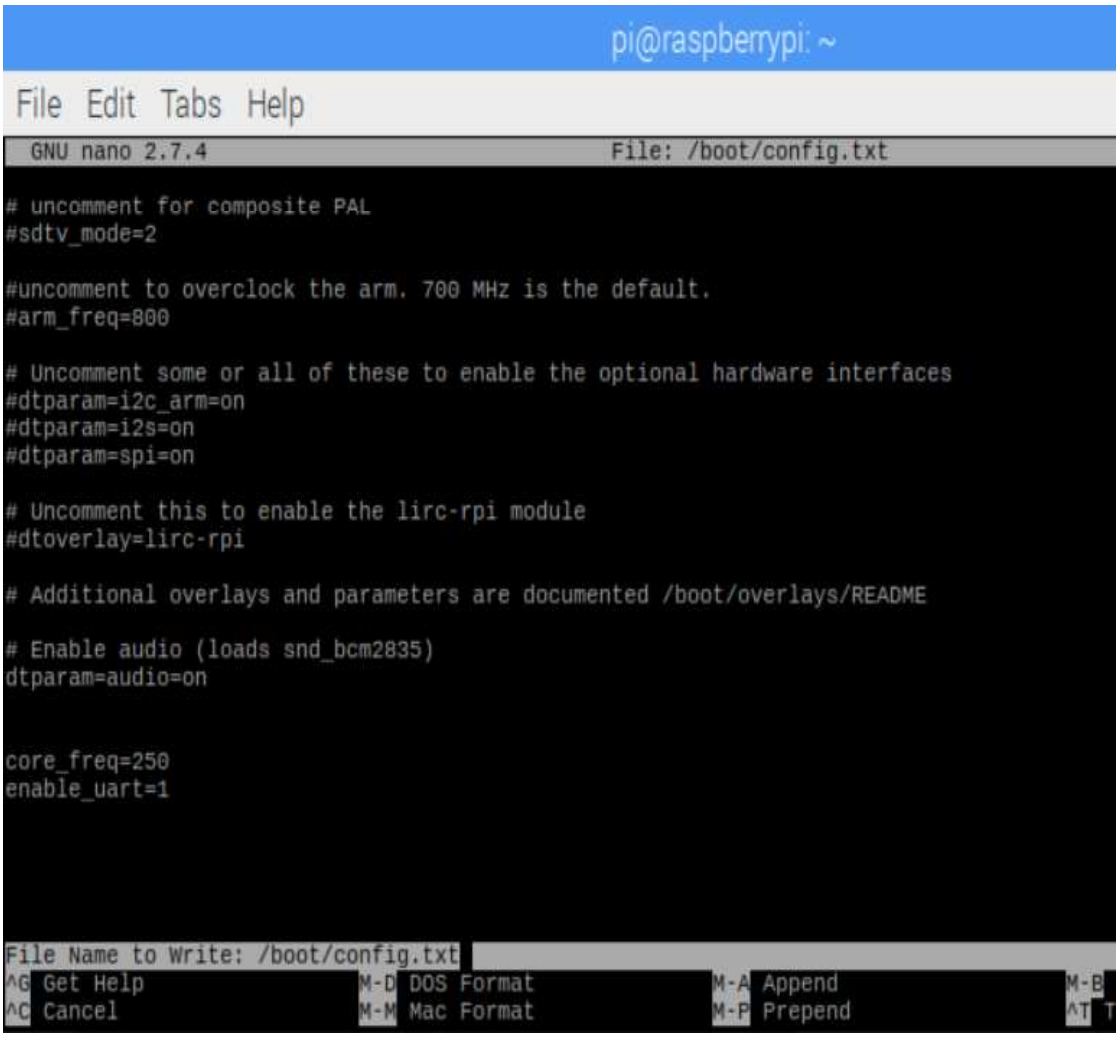


Figura 2.6. Configuración de reloj de UART

Una vez hecho estos cambios se guarda y se procede a cerrar esta ventana de configuración, de esta manera se usan comandos de referentes al lenguaje de Python que manejen la comunicación serial.

- Después de instalar el sistema operativo se abre una interfaz de comandos para instalar PPP mediante la siguiente instrucción:

  #apt-get install ppp.cserial

- Continuando con la instalación se procede a instalar el paquete “Screen” con el siguiente comando:

  #apt-get install screen

- A continuación, se instala el paquete “Elinks” de la siguiente manera:

  #apt-get install elinks

Todos estos paquetes instalados ayudan a establecer la comunicación entre el Raspberry Pi 3B y el módulo GSM/GPRS A6, esto permite transmitir los datos sensados después de haber sido procesados.

En la siguiente sección se analiza todos los dispositivos que interactúan en la construcción del Prototipo como son sensores, modulo GSM/GPRS y Contenedor de Dispositivos.

2.3. FASE DE IMPLEMENTACIÓN DE LA ESTRUCTURA DEL PROTOTIPO

En esta fase, se describe la implementación de las funcionalidades del prototipo, se procede a verificar y validar la implementación.

Se determina que la construcción del prototipo se da de forma práctica, entonces se describe los siguientes pasos para la construcción de este:

- Construcción del contenedor de cada módulo que formará el prototipo, se define de la siguiente manera pues este contenedor almacena el equipo Raspbrry pi 3B, los sensores de movimiento, la batería, el controlador de carga y el panel solar el cual se lo ubica en la parte exterior del contenedor. Dicho contenedor posee las siguientes dimensiones: (28x38x11) cm.

El prototipo diseñado de la caja contenedora se observa en la figura 2.7.

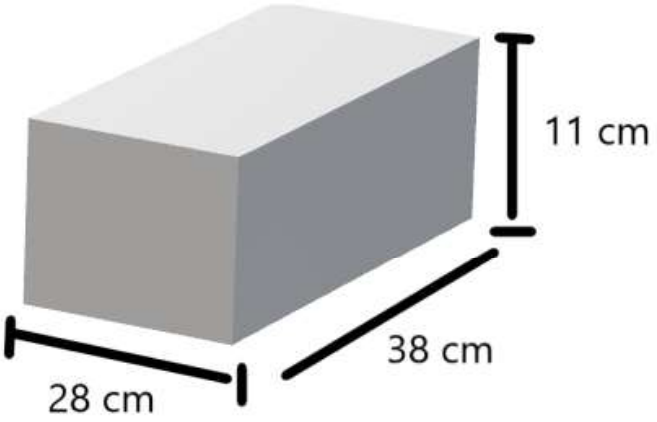


Figura 2.7. Dimensiones de caja contenedora de equipos

- El contenedor se lo construye de una manera simple con las medidas especificadas anteriormente, con planchas de 2 cm. de grosor, a este se le realiza uniones que permitan tener una cobertura resistente y que sea de utilidad para lo requerido, deseando también que sea duradero, este contenedor construido se lo observa en la Figura 2.8.

Figura 2.8. Caja contenedora de equipos

- **Controlador de Carga Solar**. – se encarga de controlar el flujo de energía que circula hacia los equipos instalados, lo parámetros que controla son la cantidad de flujo de electrones que circula por un conductor es decir la corriente eléctrica asi como también el voltaje con el cual se desea trabajar para los equipos, tomando como factor primordial la potencia de trabajo de los equipos. Este controlador de carga permite regular el voltaje y corriente hacia los equipos y a demás optimizar el

modo de carga de las baterías para asi poder obtener una mayor vida útil de las mismas.

El controlador de carga a usar se muestra en la figura 2.9.

Figura 2.9. Controlador de Carga Solar [26]

- **Panel Solar.-** este equipo permite obtener enegía aprovechando la radiacion solar. Esta formado por celdas fotovoltaicas de alto rendimiento las cuales convierten la luz en electricidad. Al estar los equipos en un ambiente aislado el uso de los paneles solares es un buen método para suministrar energía.

Este panel tiene las siguientes caracteristicas del fabricante, las cuales estan se representan en la Tabla 2.1.

Tabla 2.1. Carateristicas del fabricante del panel solar [27]

Carateristicas	Rango
Potencia máxima	10 w
Voltaje Maximo	18 v
Intensidad Maxima	0.56 A
Voc (Voltalje en circuito abierto)	21.2 v
Isc (Intensidad por Corto Circuito)	0.61 A

La razon de uso de este panel se la da debido a los calculos que se realizaron y su explicacion se dara posteriormente. Ver Figura 3.10.

Figura 2.10. Panel solar [28]

En el momento de conectar los equipos se debe se tomar en cuenta las siguientes instrucciones:

1.- Conectar la bateria, primero terminal positivo (+) y luego terminal negativo (-).

2.- Conectar el panel solar, primero terminal positivo (+) y luego terminal negativo (-).

3.- Conectar la carga, primero terminal positivo (+) y luego terminal negativo (-).

Para proceder a desconectar los equipos se lo realizara en secuencia inversa a lo ya determinado.

- **El sensor sw420**.- es un sensor que puede funcionar para monitoreo de movimientos telúricos y terremotos, tiene un umbral de medición que puede ser ajustado con un potenciómetro, el cual se encuentra incorporado. En este sensor es muy útil su señal de salida digital la misma que es tratada por un microcontrolador, en el prototipo construido quien procesa dichas señales es la plataforma Raspberry Pi 3B, el sensor proporciona salidas digitales tal es el caso que, la salida digital tiene 1L como nivel alto y 0L como nivel bajo.

A continuación, se visualiza en la figura 2.11. la forma de onda cuando ocurre un movimiento.

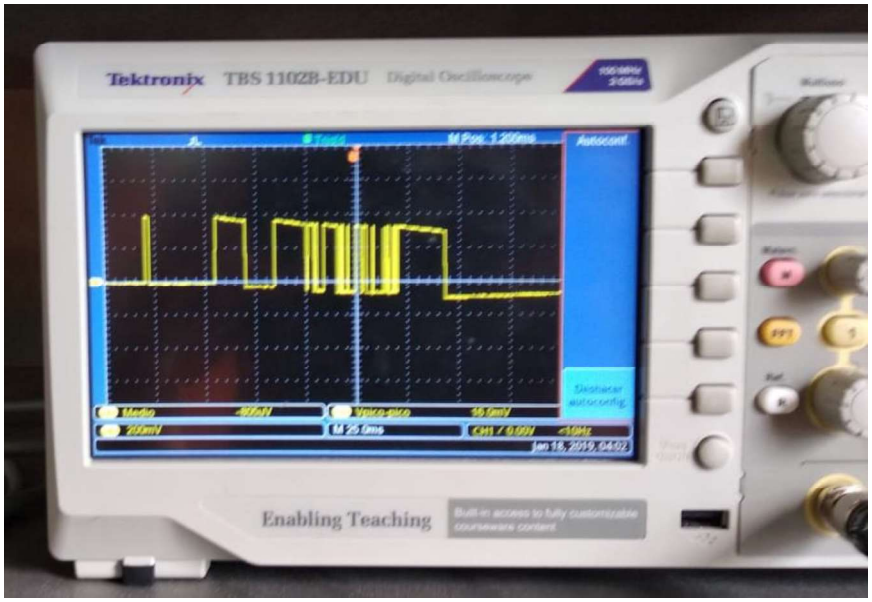


Figura 2.11. Señal de movimiento del sensor SW420

2.3.1 MANEJO DE SENSORES DE VIBRACIÓN Y DEL MODULO GSM/GPRS A6

En esa sección se analiza el funcionamiento de los dispositivos concernientes a la construcción del Prototipo como lo son sensores, modulo GSM/GPRS.

En primer lugar, se procede a la conexión física del módulo GSM/GPRS A6 y la plataforma Raspberry pi 3B como se verifica en la figura 2.12.

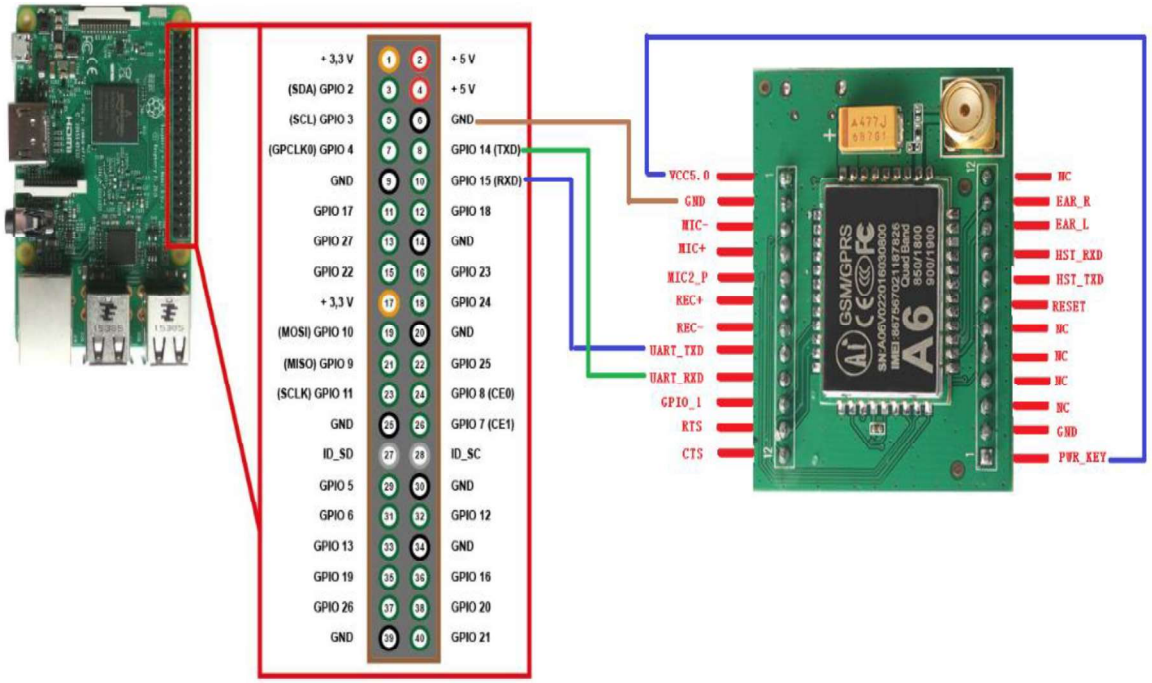


Figura 2.12. Conexión de Módulo GSM/GPRS A6 y Raspberry pi 3B [29]

Se define los siguientes pasos para la conexión:

- Paso 1: El UART (Tx) del módulo GSM/GPRS al pin GPIO 15 de (Rx) del Raspberry pi 3B.
- Paso 2: El UART (Rx) del módulo GSM/GPRS al pin GPIO 14 de (Tx) del Raspberry pi 3B.
- Paso 3: El pin GND del módulo GSM/GPRS al GND del Raspberry pi 3B y, por último.
- Paso 4: Se procede a cortocircuitar el PWR_KEY con el pin VCC 5.0 del módulo GSM/GPRS.

El módulo GSM/GPRS pose pines de transmisión y recepción una conexión de VCC y GND para su correcto funcionamiento. Luego de conectar el módulo GSM/GPRS se realiza la conexión física de los sensores de movimiento como se representa en la Figura 2.13.

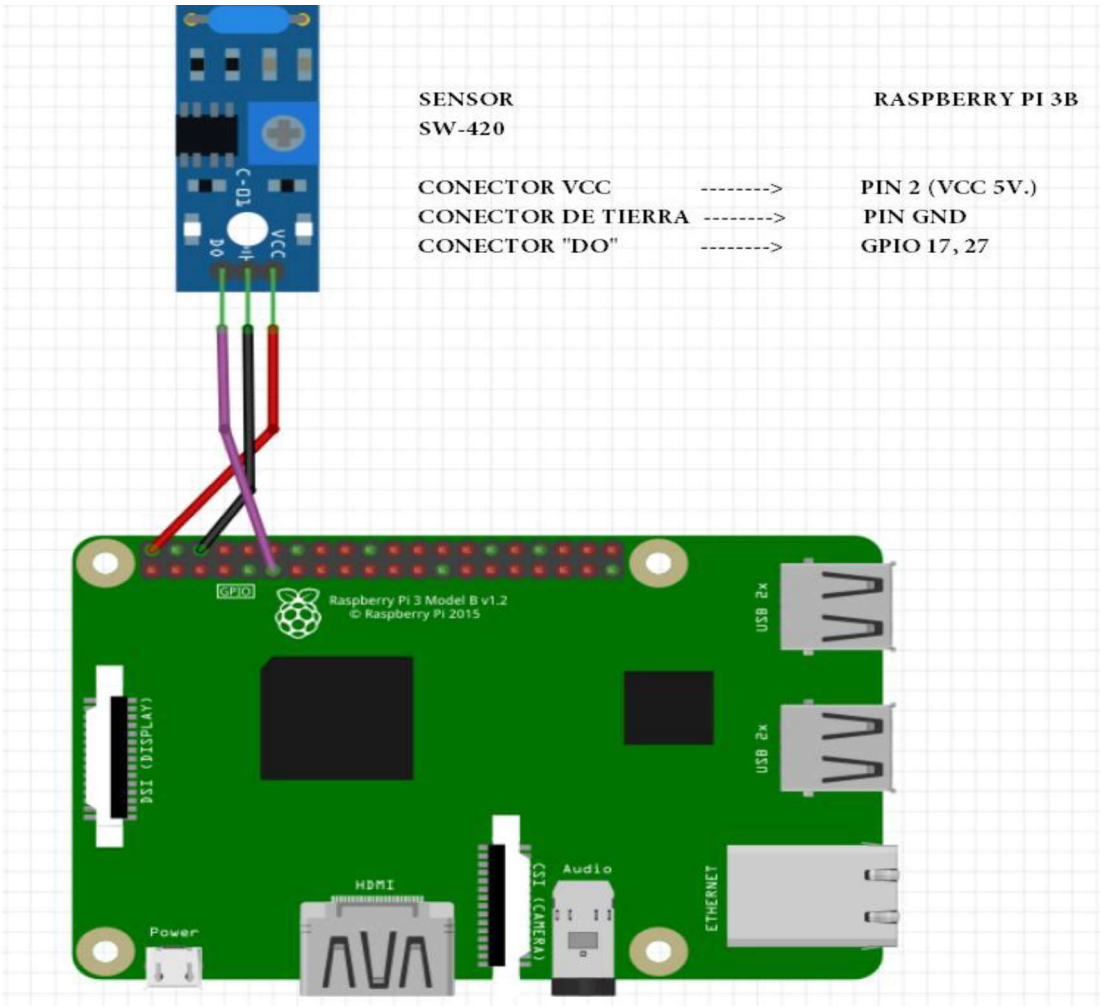


Figura 2.13 Conexión de Raspberry pi 3 y sensores de movimiento [30]

Para el proceso de sensamiento se usa los pines 17 y 27 del Rasberry pi 3B.

Este sensor como ya se había especificado en sus características anteriormente, Tabla 1.9. funciona con una alimentación VCC y su debida conexión a GND, su señal digital por el pin determinado como DO, por la cual se obtiene la señal, la misma que debe ser procesada en el módulo Raspberry pi 3B.

A continuación, se muestra en la figura 2.14. el prototipo con su debida conexión con los módulos GSM/GPRS A6 y sensores SW 420 que forman parte del prototipo que se construye.

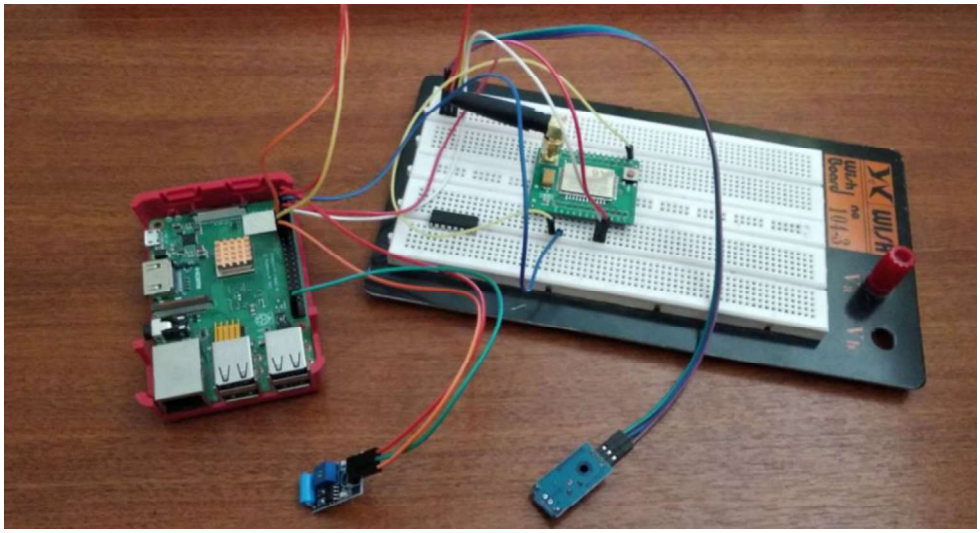


Figura 2.14. Conexión Física de sensores y modulo GSM/GPRS A6

2.3.2. DISEÑO DEL PROGRAMA DE CONTROL DE LOS SENSORES

El programa que controla los dispositivos y procesa la señal de los movimientos sísmicos, se basa en el monitoreo y la toma de datos por parte de los sensores, para luego a estos datos procesarlos y adecuarlos debidamente mediante el Raspberry Pi 3B, según esto se establece la lógica general para el funcionamiento de la aplicación. La idea central se basa en la forma de recolectar los datos mediante los sensores, estos sensores trabajan simultáneamente, si un evento se da los sensores realizan la medición cada diez segundos.

Dentro de los diez segundos los dos sensores trabajan y toman varios datos de los cuales se procede a realizar un promedio entre los dos y sacar un valor definido en los diez segundos, esto se repite durante cada minuto por lo que se llega a coleccionar seis datos en ese minuto. Estos datos se almacenan en una lista, luego se procede a enviarlos a la base de datos mediante la conexión a la red de tecnología GPRS. La conexión a la red GPRS dependerá de la cobertura y la ubicación de los equipos.

La importancia de la programación para el correcto funcionamiento del prototipo se basa en los diagramas de flujo y tomando en consideración los puntos antes expuestos, se procede a incluir las variables de uso y la estructura secuencial que se da para unificar los conocimientos de software hardware y programación para crear el sistema que permite construir el prototipo.

En el diagrama de flujo se indica el funcionamiento, esto se representa en la figura 2.15.

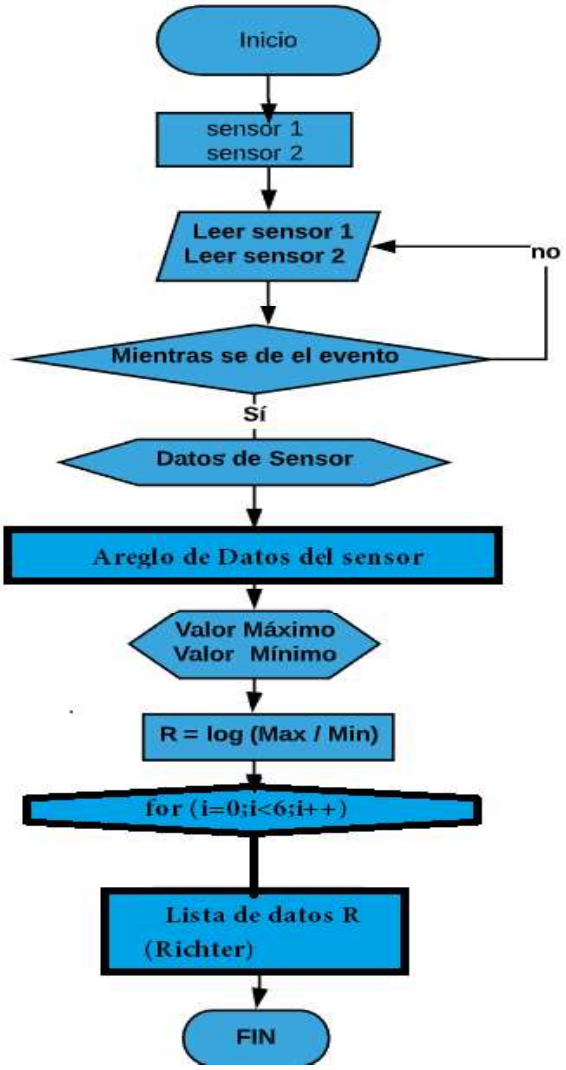


Figura 2.15. Diagrama de Flujo del Programa General del Prototipo.

La importancia de organizar la idea de funcionamiento gira en torno a los componentes y la unificación de estos ante el sistema que se está construyendo, este sistema es versátil y simple en cuanto al funcionamiento de software es decir cumple con la funcionalidad definida en los requerimientos. La parte que el software no puede cumplir adecuadamente es la de sensar, es decir, la precisión en cuanto a la toma de magnitudes en el momento de sensar ante la ocurrencia de un movimiento telúrico bien definido, ya sea por motivos

físicos de ubicación del sistema como distancia, profundidad o frecuencia de las ondas telúricas, pero es importante recalcar que esto se da principalmente por las características de precisión en la medición del sensor comercial usado en el prototipo construido. Otro factor que se toma en cuenta es también la capacidad que posee la plataforma Raspberry pi 3B para el procesamiento y transmisión de datos a la red de tecnología GPRS, la cual se lo realiza con el módulo de comunicación GSM/GPRS A6.

Dentro de la lógica del programa general cabe recalcar que una parte importante del sistema es la función sensar, ya que ésta lo que permite es armar una lista analizando los pulsos que se dan referente a los movimientos dentro de los diez segundos y además permite recolectarlos con lo cual después al procesarlos poder tener un valor definido en escala Richter, esto se puede observar en el diagrama de flujo de la figura 2.16.

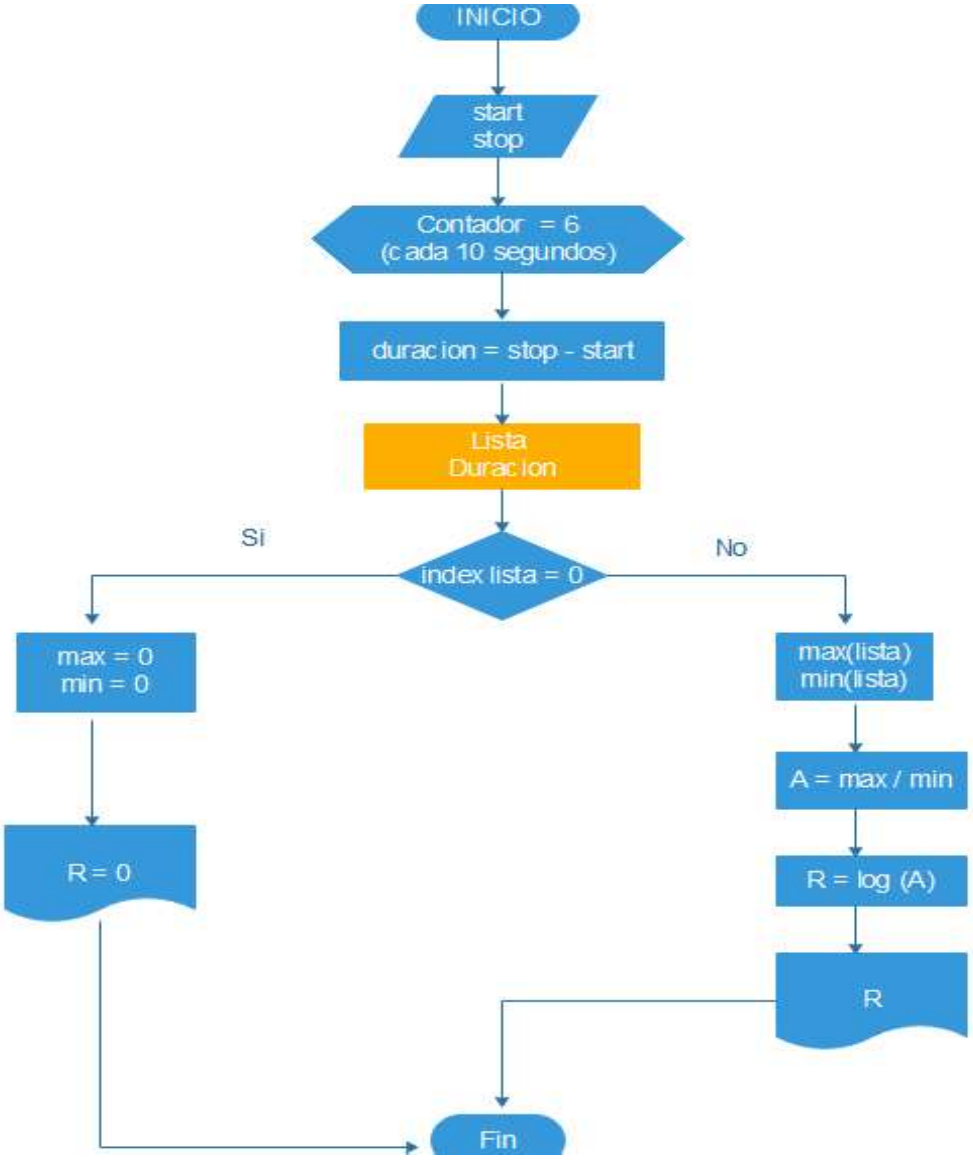
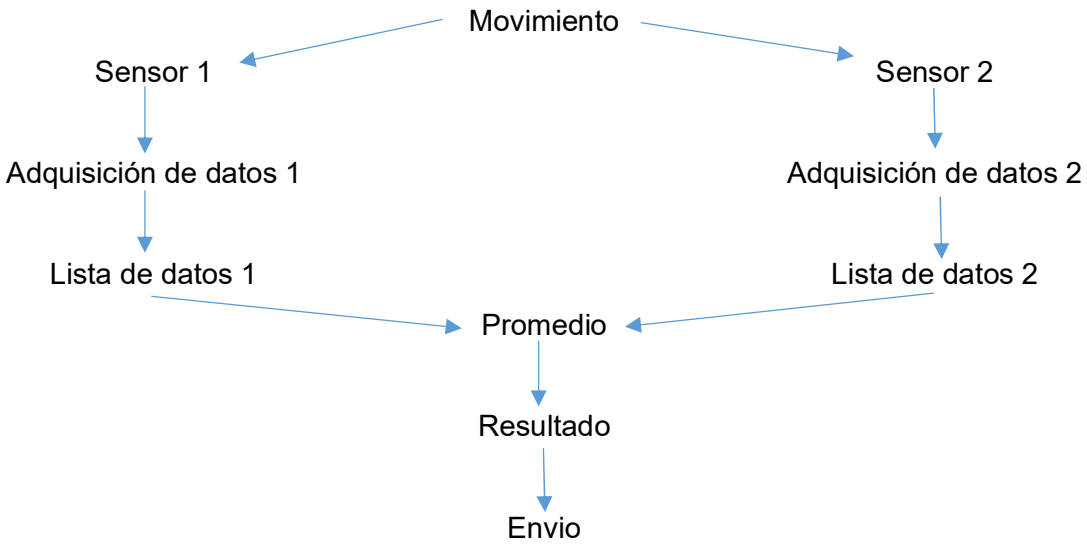


Figura 2.16. Diagrama de flujo función sensar

La clase principal, es la que después de haber sincronizado y obtenido los valores sensados, permite que se los organice, para ser enviados a la base de datos y poderlos presentar al usuario final. Aquí, se colecciona una lista de seis datos los cuales han sido tomados en un minuto cada diez segundos, todo esto con la ayuda de una función definida como sincronizar.

La función sincronizar, permite trabajar con un sensado continuo dentro del tiempo definido para sensar que se le ha dado al programa construido, cabe mencionar que se denomina sincronizar ya que establece que los dos sensores trabajen conjuntamente es decir al mismo tiempo, para realizar esto el programa se ayuda del objeto pool quien permite un manejo denominado como paralelismo de datos.

La programación se realiza con lenguaje Python, haciendo el llamado de un módulo de procesamiento, dicho módulo permite al programador aprovechar al máximo múltiples procesadores en una máquina determinada mediante esta ayuda, la función lo que hace es obtener el promedio de los datos tomados por el sensor uno y el sensor dos, a ese dato lo llama promedio, a continuación se crea una lista para ir guardando esto valores que se obtienen cada 10 segundos durante un minuto, llamándolo resultado, obviamente a cada dato se le anexa la fecha actual y la hora. El siguiente mapa conceptual explica lo ya mencionado.



La lista que se guarda con los datos se reescribe cada periodo de sensado, ya que esa lista se la guarda en el archivo de texto denominado **datos.txt,** este mismo archivo se lo abre nuevamente para leerlo y tomar los datos para poderlos enviar a la base de datos, una vez adecuados los mismos, se destaca que también a esta organización se le anexa el valor de modulo que define cuál de los tres módulos que forman parte del prototipo está enviando los datos. Este arreglo se lo lee línea por línea y se lo envía en una sola variable

denominada respuesta la cual se asocia con la URL (*Uniform Resource Locator, Localizador Uniforme de Recursos*) y los parámetros de los datos que se requiere visualizar. En la figura 2.17. se observa el diagrama de flujo referente al código ha implementado para esta funcionalidad.

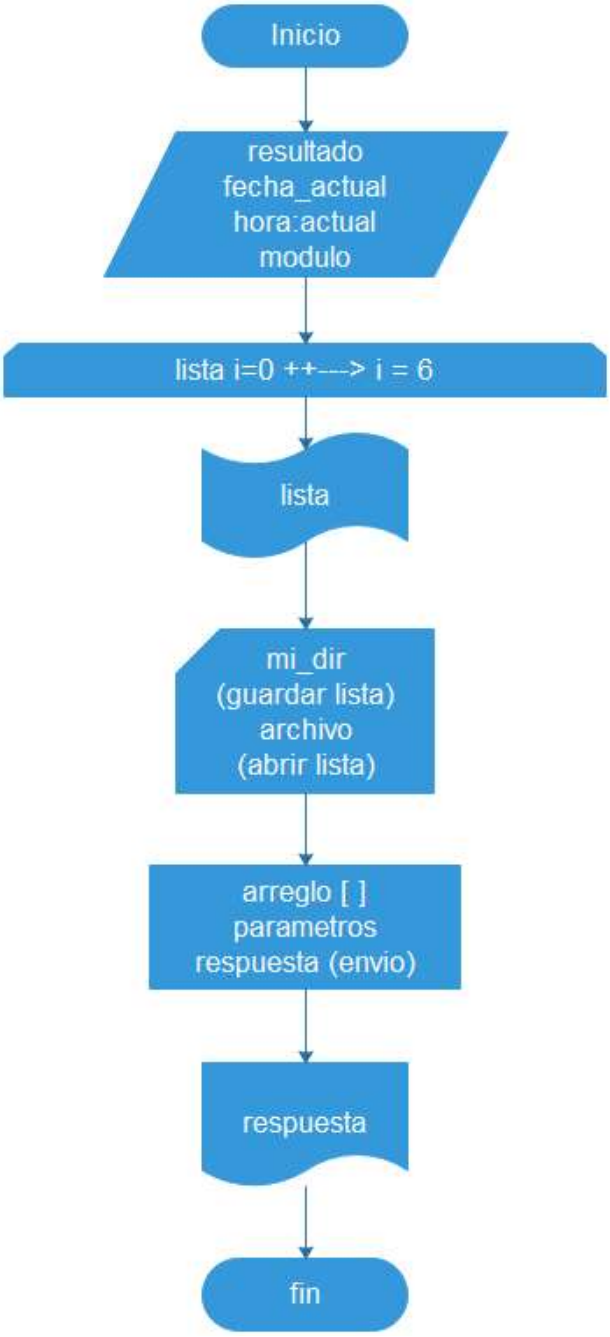


Figura 2.17. Diagrama de flujo, acondicionamiento de datos para enviar

2.4. SOFTWARE DEL SISTEMA

Para el desarrollo del software del prototipo se toma en cuenta las siguientes herramientas como lo es lenguaje de programación y los frameworks usados para el desarrollo de la aplicación como lo son: Python, Ruby on Rails, PostgreSQL, SQL.

- SQL (Structured Query Language). - Es un lenguaje el cual permite un uso a gran escala en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla. [31]

Para proceder a realizar una explicación sobre la programación implementada en el funcionamiento del sistema se procede a comprender el uso de la función logarítmica, la cual permite obtener una escala denominada Richter en función de las magnitudes de la onda sísmica. La magnitud de un terremoto se relaciona con cuánta energía libera.

Instrumentos llamados sismógrafos detectan el movimiento de la tierra; el movimiento más pequeño que puede detectarse en un sismógrafo tiene una amplitud A0. La ecuación define el cálculo de la escala Richter en función de la magnitud de sus ondas. Ecuación (2.4.1) [32]:

$$R = \log (A/Ao) \tag{2.4.1}$$

En la ecuación 2.4.1 se tiene los siguientes parámetros:

- **R** es la variable que define la escala Richter, la cual ayudada de una función logarítmica establece la magnitud de los movimientos sísmicos.
- **A** establece la mayor magnitud sensada durante el movimiento sísmico.
- **Ao** establece la menor magnitud sensada durante el movimiento sísmico.

Esta ecuación permite el procesamiento de los datos tomados por cada uno de los sensores dentro de la aplicación, esta es una ecuación la cual sirve de herramienta para el análisis y cálculo de las medidas mediante un programa estructurado adecuadamente, ya que se debe aclarar que también existen otros métodos para análisis de este tipo se datos sobre movimientos sísmicos asi como equipos especializados con mayor eficiencia y precisión dentro de rangos bien definidos.

Cabe destacar que estas medidas no serán exactas, debido a que los sensores de movimiento genéricos son muy diferentes a los sensores de precisión, esto en cuanto a la

utilización especifica que se le requiere dar a cada uno de estos, en el caso movimientos sísmicos, la precisión dependerá del tipo de tecnología, el costo, la sensibilidad y el funcionamiento. Los sensores denominados geófonos de alto costo son los que exactamente se usan en los sismógrafos profesionales.

Ahora se procede a crear la función sensar la cual nos permite tomar los datos mediante los sensores y una vez tomados poderlos procesar adecuadamente para tener mediciones establecidas de los movimientos telúricos que se dan en un respectivo intervalo de tiempo, las mediciones se las hacen cada 10 segundos como se explicó de acuerdo con el diagrama de flujo 2.16. El bloque de código de la función sensar se puede observar en la figura 2.18.

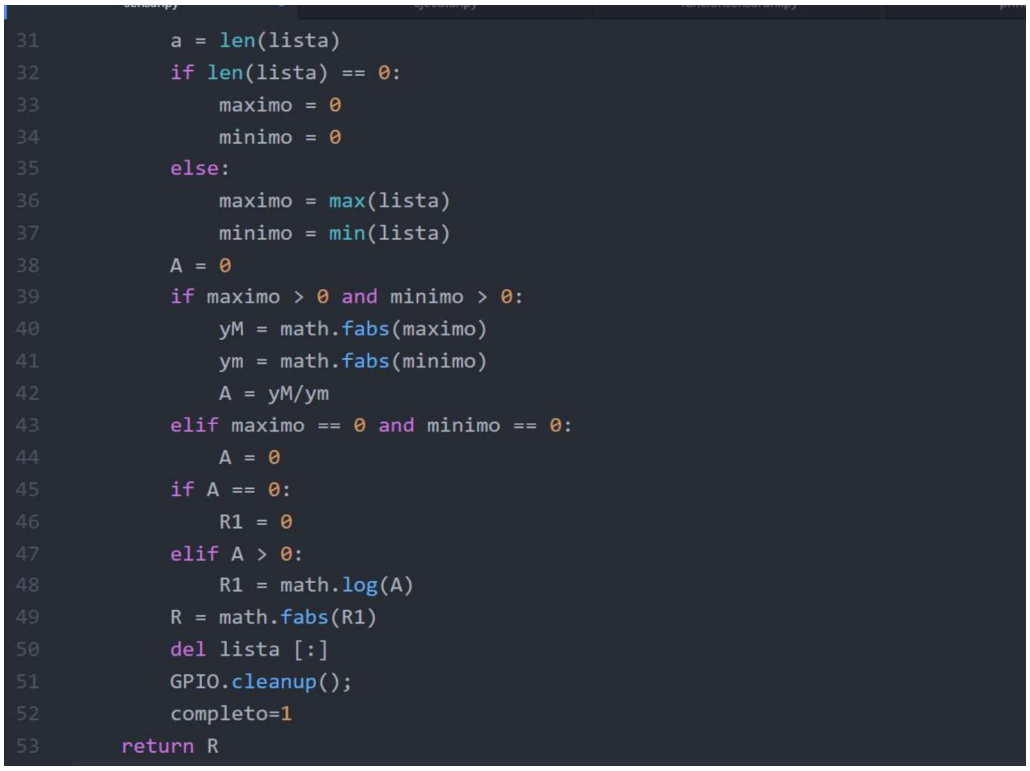
```
def sensar(channel):
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(channel, GPIO.IN)
    completo = 0
    while completo ==0:
        def callback(channel):
            start = time.time()
            stop = time.time()
            while GPIO.input(channel)==0:
                start = time.time()
            while GPIO.input(channel)==1:
                stop = time.time()
            if stop == None and start == None:
                duracion = 0
            elif stop != None and start != None:
                duracion = stop-start
            total = duracion
            lista.append(total)
        GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)
        GPIO.add_event_callback(channel, callback)
        time.sleep(10)
```

Figura 2.18. Bloque de código de la Función sensar

Como se puede observar en el bloque de código anterior se toma señales sísmicas en intervalos de tiempo definidos, en los cuales si se da un sismo el sensor tomará las señales en forma de pulsos los mismos que indican que los sensores registran dicho movimiento, ese pulso se lo analiza mediante tiempos de duración. Y posteriormente se va tomando cada muestra como un valor a almacenar del movimiento sísmico.

Mediante el método **append** se va almacenado dichos valores en una lista, para posteriormente poder determinar el máximo valor, el mínimo valor o por su parte determinar si no existió ningún tipo de valor sensado.

Después de almacenar los valores en una lista, se procede a analizar la lista, si la lista está vacía o si tiene valores almacenados para hacer su respectivo tratamiento mediante la ecuación explicada anteriormente, (ecuación 2.4.1). Para definir claramente los valores máximos o mínimos dentro de la lista se explica el siguiente bloque de código establecido en la figura 2.19.



```
31          a = len(lista)
32          if len(lista) == 0:
33              maximo = 0
34              minimo = 0
35          else:
36              maximo = max(lista)
37              minimo = min(lista)
38          A = 0
39          if maximo > 0 and minimo > 0:
40              yM = math.fabs(maximo)
41              ym = math.fabs(minimo)
42              A = yM/ym
43          elif maximo == 0 and minimo == 0:
44              A = 0
45          if A == 0:
46              R1 = 0
47          elif A > 0:
48              R1 = math.log(A)
49          R = math.fabs(R1)
50          del lista [:]
51          GPIO.cleanup();
52          completo=1
53      return R
```

Figura 2.19. Bloque de código para tratamiento de la ecuación escala Richter

Mediante el método **len** que se encuentra en la línea 31 definido en la figura 2.19. Del bloque de código se verifica si nuestra lista está vacía o posee valores recolectados referentes al análisis de las señales sísmicas. Después de esto se verifica los valores máximo y mínimo, de la lista recolectada, que se obtienen mediante los métodos **max**(lista) y **min**(lista), los valores máximo y mínimo, son los datos necesarios para poder trabajar con la ecuación 2.4.1. Cuya ecuación permite obtener el valor en escala Richter de los datos ya sensados.

A continuación, se explica la sincronización de los sensores es decir el sensor 1 y el sensor 2 trabajan simultáneamente en la recolección de datos, para lo cual se hace uso de la función sincronizar.

La función sincronizar está basada en el funcionamiento del objeto pool de Python, el cual permite paralelizar la ejecución de una función en múltiples valores de entrada, distribuyendo los datos de entrada a través de procesos (paralelismo de datos). [32]

Con la ayuda de este objeto se puede a la vez manejar los pines de entrada en nuestro caso pin 17 y 27. La función sincronizar se observa en la Figura 2.20.

```
import RPi.GPIO as GPIO
import time
import math
import funcionsensaruni
from multiprocessing import Pool
##global lista

def sumarLista1(A):
        suma = 0
        for m in range(0,len(A)):
            suma=suma+A[m]
        promedio = suma/2
        return promedio
def sincronizar():

##    if __name__ == '__main__':
    p= Pool(2)
    resp = p.map(funcionsensaruni.sensar,[27,17])
####    print (result)
    valor = sumarLista1(resp)
    return valor
```

Figura 2.20 Funcion Sincronizar

En esta función se destaca también el manejo de los datos de los dos sensores a los cuales también se los hace un promedio para tener un valor referencial en si una medida especifica en un mismo intervalo de tiempo derivado del procesamiento de los datos tomados.

Esta función sincronizar es de mucha utilidad ya que como se menciona ayuda a poder manipular diferentes datos en un mismo instante de tiempo, es algo necesario dentro del prototipo para poder trabajar a la vez con los dos sensores conectados en el raspberry i 3B, el proceso simultaneo a estos valores ayuda para poder tener mas muestras en un mismo instante.

En el siguiente bloque de código se puede analizar la estrutura de la lista que se forma para poder almacenar los datos, los mismos que estaran adecuadamente estructurados, esto permite que se pueda asociar y guardar como una lista, a continuacion se visualiza el bloque de código de la figura 2.21.

```
import requests
from datetime import datetime
from datetime import timedelta
import math
global lista
import sincronismo
lista = []

completo = 0
i=0

while completo ==0:
    resultado = str(sincronismo.sincronizar())
    print (resultado)
    fecha_actual = datetime.now().strftime("%Y-%m-%d")
    hora_actual = datetime.now().strftime("%H:%M:%S")
    fecha_a = str(fecha_actual)
    hora_a = str(hora_actual)
    datoF = fecha_a + ' ' + hora_a + ' ' + resultado
    i=i+1
    lista.append(datoF)

    if i==6:
        print (lista)
```

Figura 2.21. Variables de los datos a almacenar

Las variables que se usa en la aplicación son los siguientes:

- Fecha = fecha_a
- Hora = hora_a
- Valor = resultado
- Módulo = Módulo 1; 2 o 3

El valor de la variable resultado proviene del procesamiento de la función sincronizar y sumar lista, la cual como se menciona anteriormente, devuleve un promedio de la recolección de valores de los dos sensores en un determinado tiempo.

La variable fecha_a, permite almacenar la fecha referente a cada dato sismico tomado por los sensores. La variable hora_a, almacena la hora referente en la cual se produce la toma de los valores sismicos del promedio determinado entre los sensores.

En cada módulo de medición sísmico se añade un parámetro más que es el valor del módulo especificando si es uno, dos o tres, este identifica cual módulo envió la información al servidor. Todas estas variables se agrupan y se crea una lista para almacenar la información, esa información cada determinado tiempo de sensamiento forman la lista y se guarda en un archivo .txt dentro del Rapsberry pi 3B para luego al ser leido enviarlo a

la base de datos, la estructuración de los datos para poderlos enviar se da de la siguiente manera como se observa en la figura 2.22.

```
archivo = open("datos.txt", "r")
arreglo = []
for linea in archivo.readlines():
    arreglo += linea.rstrip().split(" ")
archivo.close()
print(arreglo)

modulo = 1
url = "https://interfaz-web-tesis.herokuapp.com/registros/guardar_registro/"
parametros = {'fecha1': arreglo[0], 'hora1': arreglo[1], 'valor1': arreglo[2], 'modulo1': modulo,
              'fecha2': arreglo[3], 'hora2': arreglo[4], 'valor2': arreglo[5], 'modulo2': modulo,
              'fecha3': arreglo[6], 'hora3': arreglo[7], 'valor3': arreglo[8], 'modulo3': modulo,
              'fecha4': arreglo[9], 'hora4': arreglo[10], 'valor4': arreglo[11], 'modulo4': modulo,
              'fecha5': arreglo[12], 'hora5': arreglo[13], 'valor5': arreglo[14], 'modulo5': modulo,
              'fecha6': arreglo[15], 'hora6': arreglo[16], 'valor6': arreglo[17], 'modulo6': modulo}

respuesta = requests.get(url = url, params = parametros)
print("Respuesta: ", respuesta)
```

Figura 2.22. Arreglo de datos

A continuación se da la explicación de la interfaz web, la cual se la realiza con la ayuda de la plataforma Heroku, este framework trabaja con la arquitectura Modelo Vista Controlador, la misma que permite diferentes servicios en la nube, en este entorno se crea la aplicación web el mismo que con la ayuda de Ruby and Rails permite la establecer la funcionalidad adecuada ante la necesidad de la creación de la aplicación web.

Para la base de datos se usa Posgrets SQL tal como se visualiza Figura 2.23.

database.yml

```
# gem 'pg'
#
default: &default
  adapter: postgresql
  encoding: unicode
  host: localhost
  port: 5432
  username: postgres
  password: [redacted]
  # For details on connection pooling, see Rails configuration guide
  # http://guides.rubyonrails.org/configuring.html#database-pooling
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>

development:
  <<: *default
  database: gprs-prueba_development
```

Figura 2.23. Configuración para el uso de la Base de Datos en Rails

Ante la administración de la base de datos se procede al uso de Posgrets SQL, este es un gestor de base de datos que permite trabajar como un sistema relacional orientado a objetos y lo principal es que es de código abierto.

pgAdmin.- es la herramienta de administración con un entorno visual libre y de codigo abierto instalable en diferentes sistemas operativos como en el caso de la construccion del prototipo se le instala en el sistema operativo windows. Esta herramienta proporciona la facilidad de gestionar y administrar las bases de datos ya sea mediante sentencias SQL, o de una mejor manera con la ayuda de su entorno gráfico. A partir de eso se puede acceder a todas las funcionalidades de la base de datos, consultas, manipulación y gestión de datos, incluso opciones avanzadas. En la siguiente figura se observa la interfaz gráfica del gestor de base de datos, Figura 2.24.

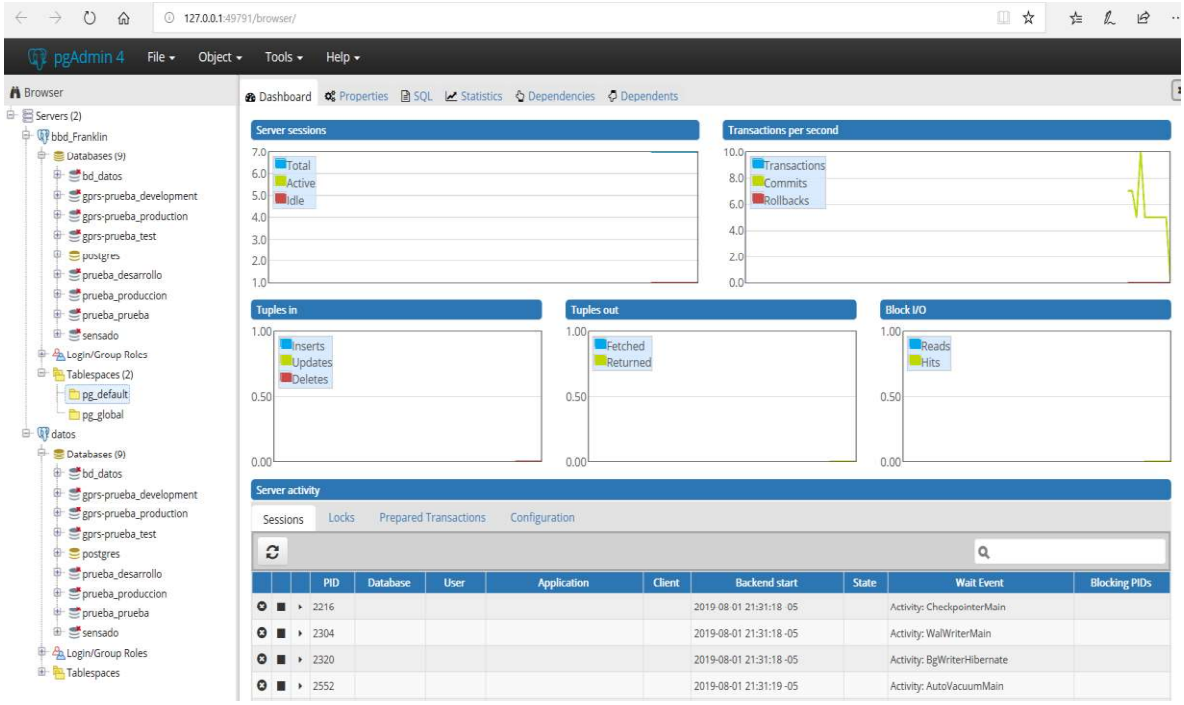


Figura 2.24. Interfaz Visual de administracion pgAdmin 4.

La interfaz web es una estructura formada por una serie de elementos gráficos que permiten a los usuarios acceder a los contenidos de un sitio web.

Para la interfaz web se procede a la instalacion de debian, una ves instalada la version debian para windows se procede a abrir una terminal en la cual se empieza a crear los archivos necesarios para la interfaz gráfica del prototipo. Lo mencionado se puede visualizar en la figura 2.25.

Figura 2.25. Terminal Debian en Windows

Para proceder a montar los archivos necesarios para la interfaz grafica se digita el siguiente comando con su ruta establecida, la cual es donde se van a guardar todos los archivos necesarios para su creación:

franklin@DESKTOP-S8HDOP6:*/mnt/c/Users/Asus/Desktop/Tesis Interfaz Web$ rails new interfaz_web -d postgretsql*

Una vez aplicado la instrucción mencionada anteriormente se puede ahora visualizar en la figura 2.26. El directorio y los archivos creados de de esta interfaz web.

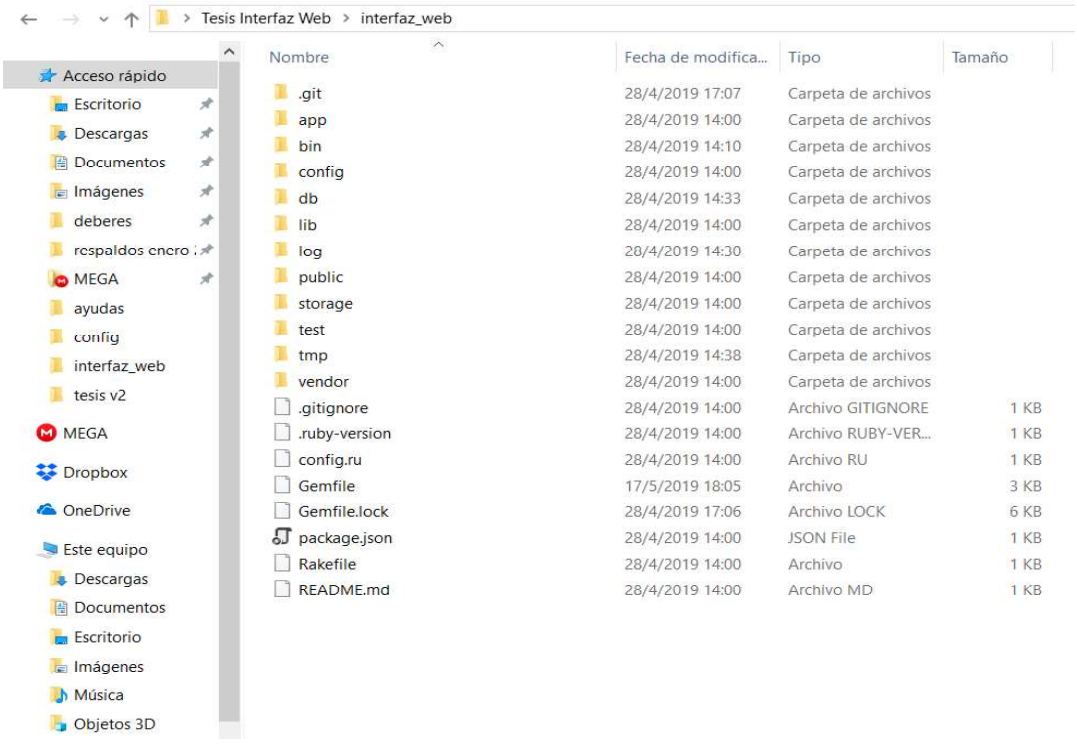


Figura 2.26. Archivos de creacion de la Interfaz Web

Dentro de los archivos creados se realiza unos cambios en el archivo llamado Gemfile.

El archivo Gemfile describe las dependencias del programa desarrollado para que se ejecute adecuadamente. Las gemas dentro de Ruby son paquetes de librerias, los cuales se instalan en el sistema y permiten usarse para que el mecanismo que se este desarrollando mediante la programación haga uso de esto.

Dentro de las librerias, se menciona algunas con las cuales se relaciona y que son de mayor utilidad pues proporcionan gran ayuda para la construccion de nuestra interfaz web, estas son las siguientes: bootstrap, jquery-rails, mini_racer, si estas no estan en el archivo se las añade caso contrario ya las estan listas para usarlas, el archivo Gemfile deben estar de la siguiente manera figura 2.27.

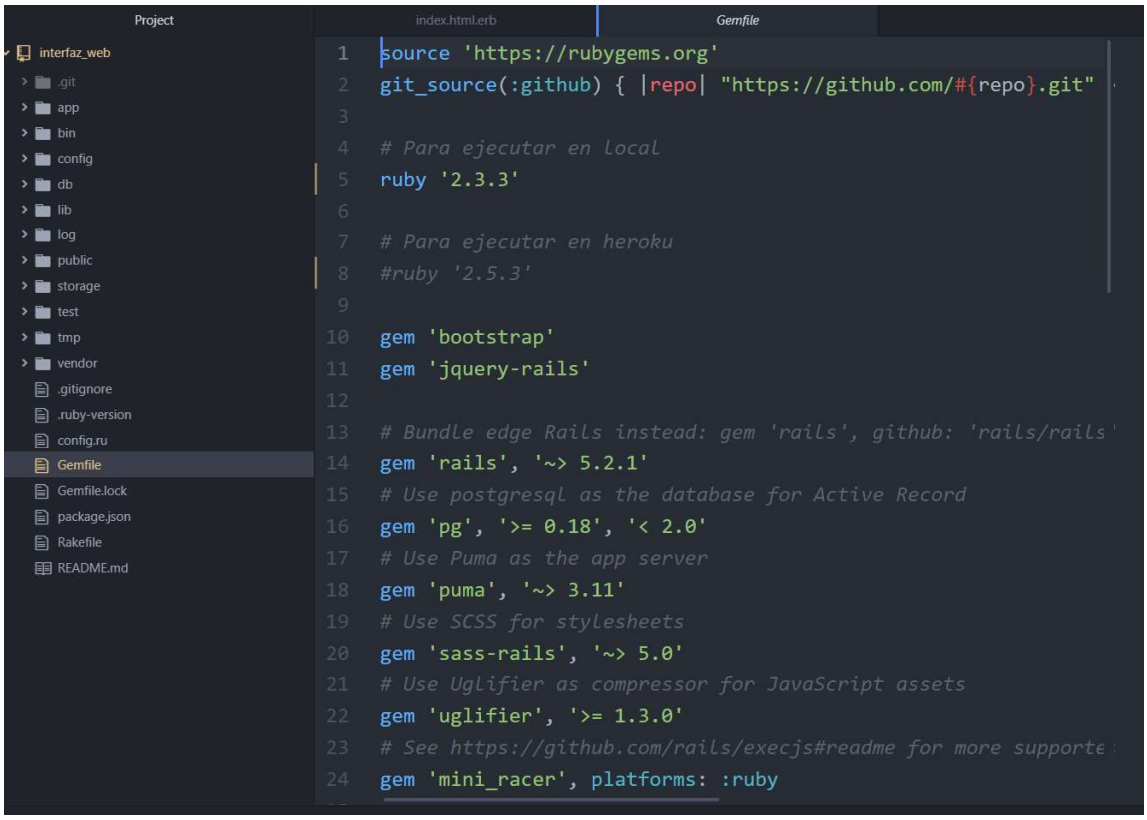


Figura 2.27. Archivo Gemfile

Gem bootstrap: al instalar esta gema proporciona una librería en la cual se trabaja con herramientas de CSS (Cascading Style Sheets, Hojas de Estilo en Cascada) y Javascript lo que permite dar estilo y moldear la interfaz web, esta gema organiza todo el entorno de trabajo.

Gem jquery-rails: esta gema es una ayuda complementaria para trabajar con la gema Bootstrap.

Gem mini_racer: esta gema proporciona la librería que permite un pequeño enlace entre javaScrip y ruby.

Luego se ejecuta el comando ***bundle install***, para instalar la gema de jQuery y bootstrap permitiendo instalar y ejecutar las diferentes herramientas relacionadas a dichas librerias.

Se procede ahora a la creacion de la base de datos, al manejar un modelo vista controlador es escencial establecer la configuracion adecuada de la Base de datos de Desarrollo, Prueba y Producción.

Ahora dentro de la misma carpeta de archivos de configuracion de la aplicacion se procede a ubicar en la carpeta ***config*** y luego se continua a abrir el archivo ***database.yml*** en el cual se hacen los siguientes cambios:

host: localhost

port: 5432

username: postgres

Las configuraciones realizadas hacen referencia a la ubicación de la base de datos (localhost), además al instalar estas librerias tambien se da la certeza de poder trabajar adecuadamente con protocolos TCP/IP, en si poder relacionarse con la pagina web, la otra configuración es el puerto en este caso se trabaja con el puerto 5432 el cual permite correr el servicio POSTGRESQL. Además de esto se definine la contraseña como se puede ver en la Figura 2.28.

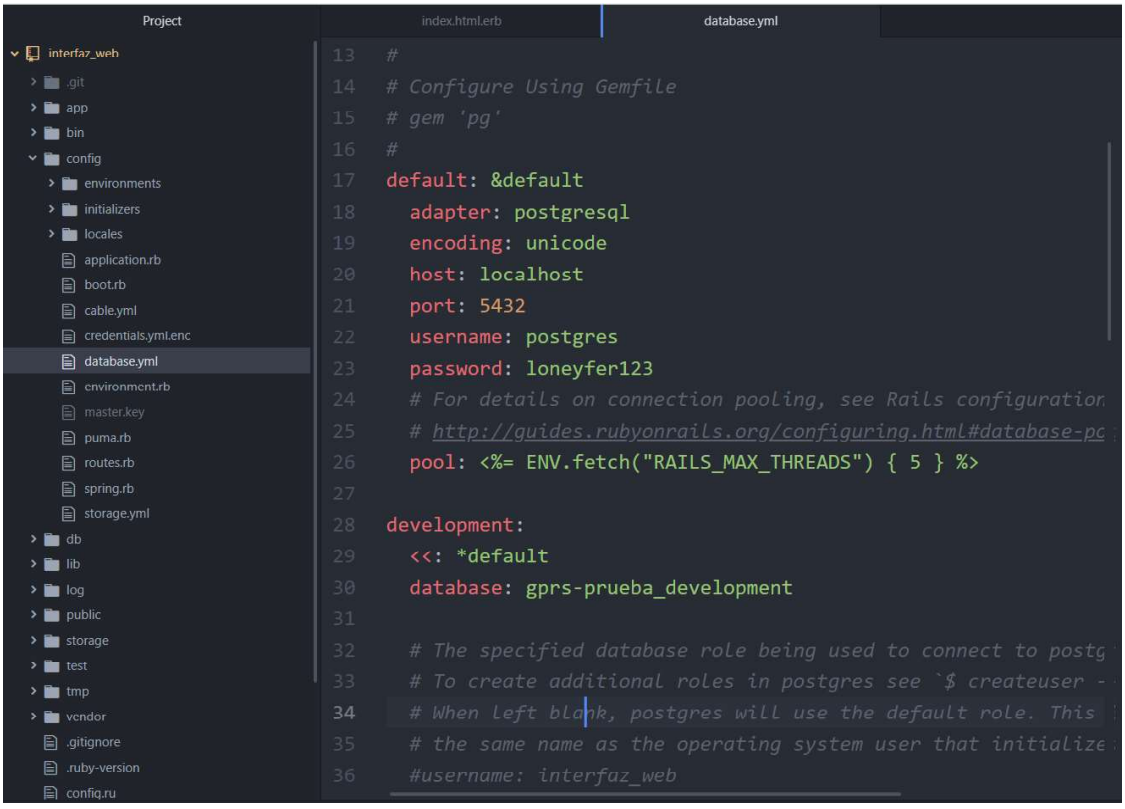


Figura 2.28. Configuración Base de Datos

También se asigna el nombre de la base de datos de la siguiente manera:

```
development:
  <<: *default
  database: gprs-prueba_development
test:
  <<: *default
  database: gprs-prueba_test
production:
  <<: *default
  database: gprs-prueba_production
```

Haciendo referencia a las bases de datos que maneja este modelo de desarrollo, prueba y producción. Cabe mencionar que la base de datos de producción es la más importante pues es esa la que corre y es la que define la aplicación optimizada para ejecutarse con el mejor desempeño posible y poder responder a solicitudes de usuarios reales. Una vez realizados estos cambios se abre una ventana de comandos en Debian sobre Windows estableciendo la ruta de la ubicación de la interfaz web y se procede a la creación de las tablas de datos a almacenar los mismo que en este caso se denominan Registros.

Continuando en la consola de Debian sobre Windows se procede a correr la siguiente instrucción:

franklin@DESKTOP-S8HDOP6:/*mnt/c/Users/Asus/Desktop/Tesis Interfaz Web$ rails g model Registro*

Al ejecutar esta instrucción se crea una migración en la cual se tiene el archivo para creación de registros a almacenar en la base de datos, esto se localiza en la carpeta *db/migrate/* y a continuación se procede a buscar el archivo *create_registros.db.* Dentro de este archivo se observa la clase **CreateRegistros** aquí se define los campos que se van a almacenar con su respectivo tipo de dato haciendo referencia a la tabla que pertenece, como se observa en la figura 2.29.

Figura 2.29. Creación de la tabla de registros

Una vez realizados estos cambios en la consola de Debian para Windows se digita el siguiente comando:

franklin@DESKTOP-S8HDOP6:/*mnt/c/Users/Asus/Desktop/Tesis Interfaz Web$ rake db:migrate*

Con lo cual se conecta automaticamente con la base de datos y procede a ejecutar la migracion de la base de datos, en si ejecuta la tabla con los campos ya definidos.

Ahora se va a crear el controlador de registros, el cual permite determinar la url a la cual se va a apuntar para almacenar los datos, esto se establece aplicando la siguiente linea de comando:

franklin@DESKTOP-S8HDOP6:/*mnt/c/Users/Asus/Desktop/Tesis Interfaz Web$* rails g controller registros index guardar_registro

Gracias a este comando se genera un controlador el cual maneja una clase derivada la misma que en el Modelo Vista Controlador permite acceder a la base de datos. Esta clase tomó todos los valores de la clase Registro a esos datos los asoció con el controlador de registros. La creación de esta clase permite guardar los registros, esto se puede visualizar en la Figura 2.30.

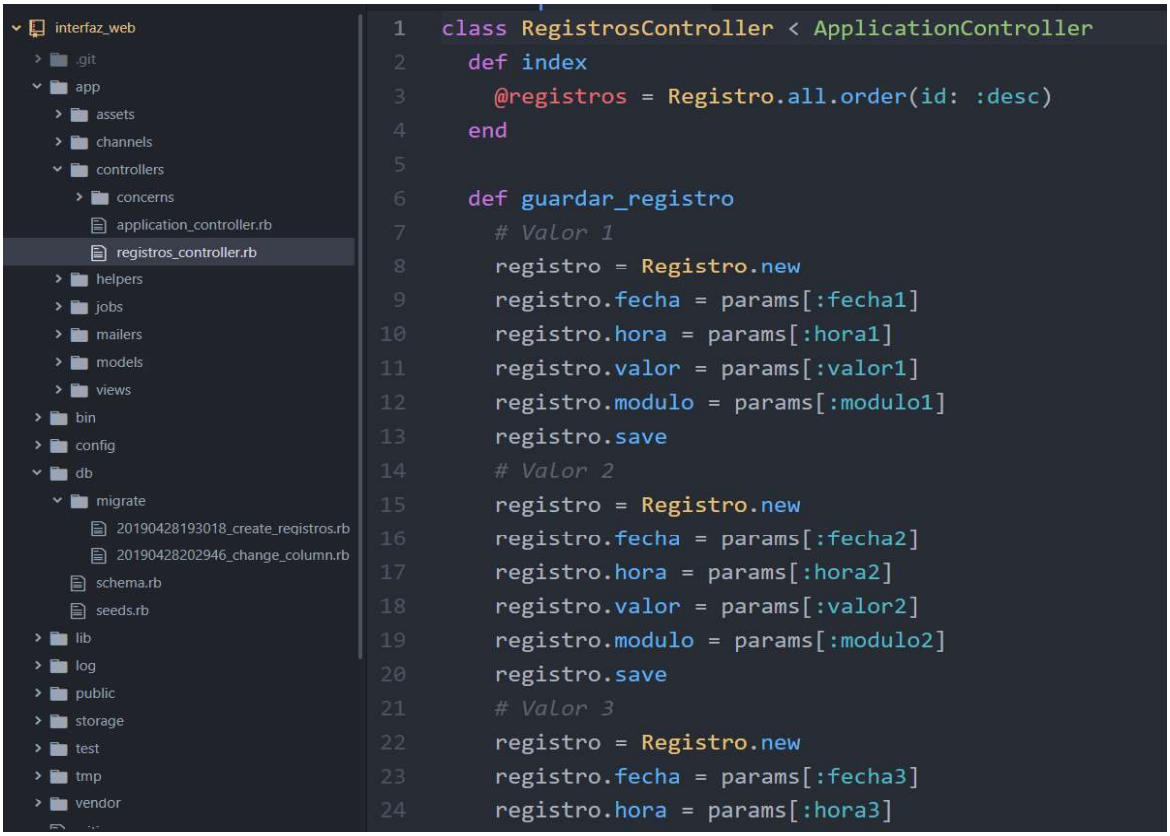


Figura 2.30. Clase para guardar los registros de los datos procesados

Ahora también se puede visualizar el archivo index.html.erb este permite manejar la parte visual de la aplicación web, establecer las columnas y definir lo que se desea visualizar en la aplicación web. Esto se visualiza en la figura 2.31.

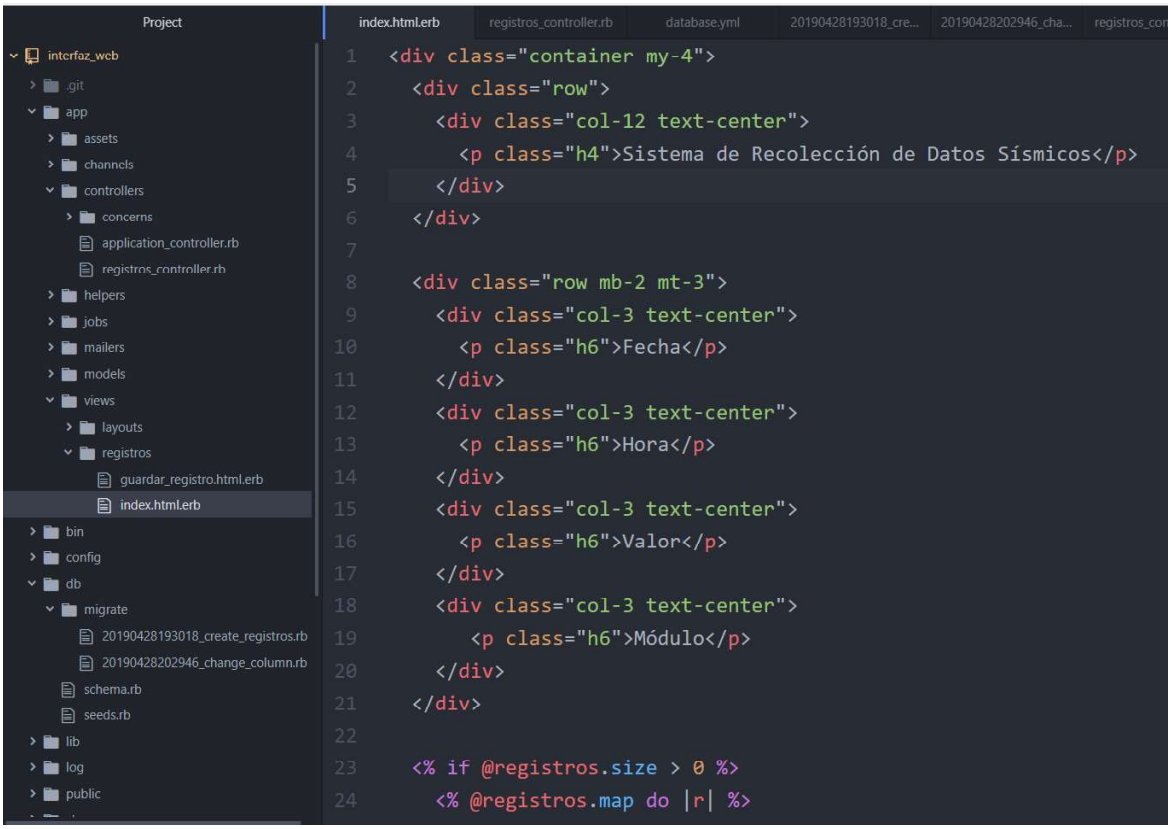


Figura 2.31. Index de aplicación web

Aquí se maneja lenguaje Html (HyperText Markup Language), el cual permite el desarrollo de la pagina web establece la estética de la página web hacia el usuario, es decir lo que va a visualizar el usuario al abrir la aplicación web.

Luego de realizar estos cambios procede a ubicar la carpeta ***views / layouts*** y a continuación se abre el archivo ***application.html.erb***, esta sera la pagina html principal, dentro de este archivo se tiene una instrucción definida de la siguiente manera <%= yield %> la misma que permite ejecutar el codigo localizado en ***index.html.erb*** el cual contiene la estetica de los atributos a mostrar en nuestra aplicación web, esto se visualiza en la figura 2.32.

En este archivo tambien se establece mediante código que la página web se refresque cada cinco segundos lo que permite automatizar dentro de ese periodo la pagina web para el usuario final.

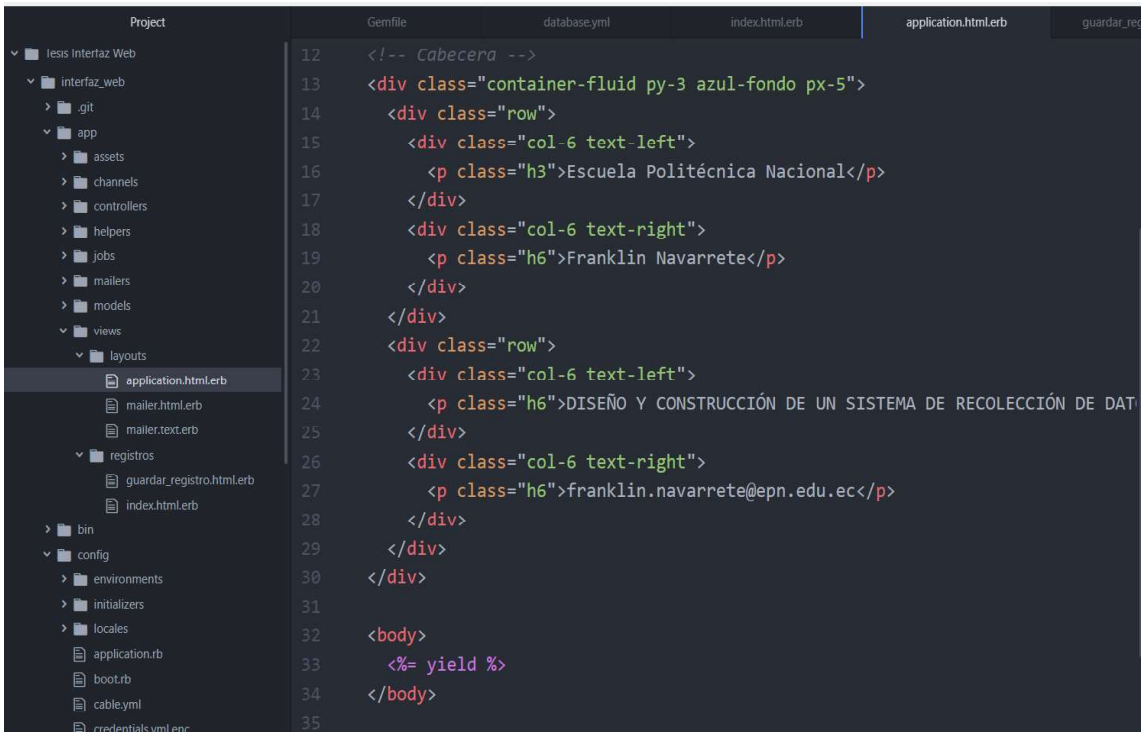


Figura 2.32. Código HTML de la Aplicación Web Principal

Para poder guardar todos y cada uno de los datos en la base de datos y luego poder visualizarlos en la aplicación web, hay que dirigirse a la carpeta de ***la interfaz web***, luego se busca la carpeta ***config*** y posteriormente el archivo ***routes.rb*** como se visualiza en al siguiente figura 2.33.

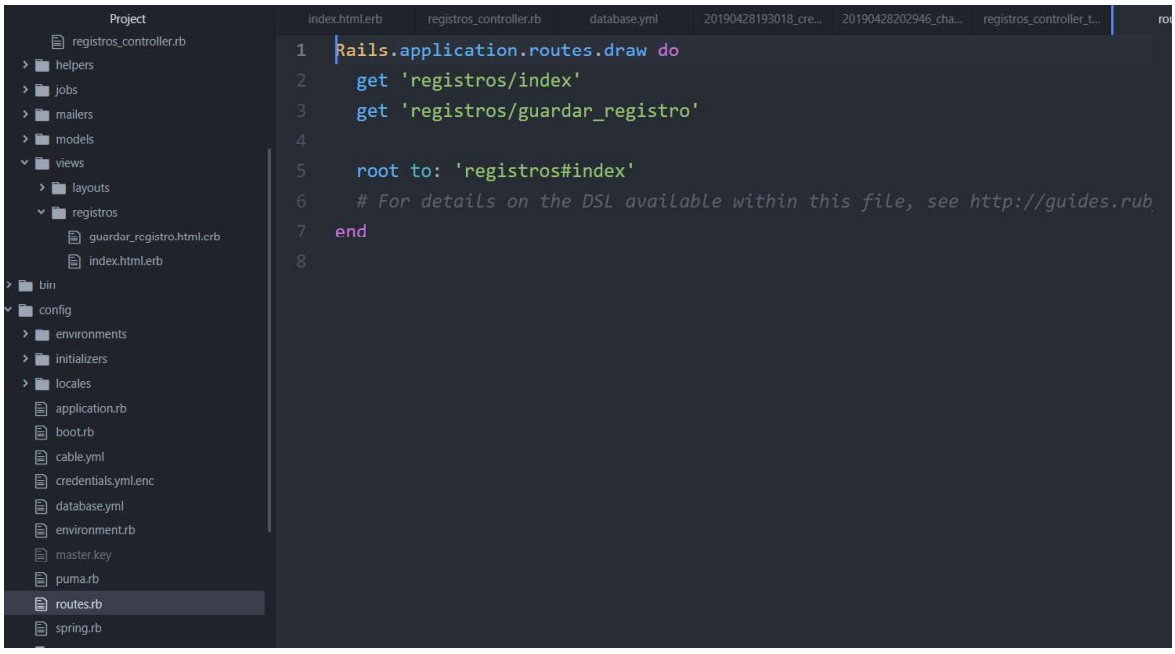


Figura 2.33. Ruta para almacenamiento de registros

Dentro de este archivo se configura la ruta que en este caso es “registros#index” con lo cual se especificara a cual se va a dirigir siempre que se procede a abrir la pagina web.

Para verificar que la pagina web y los controladores de rails esten adecuadamente implementados se escribibe el siguiente comando para observar el correcto funcionamiento.

franklin@DESKTOP-S8HDOP6:/*mnt/c/Users/Asus/Desktop/Tesis Interfaz Web$* rails s

Con lo cual se verifica el funcionamiento de la interfaz web es decir se podra ver los datos que ya se van alacenando en la base de datos y se muestran al usuario mediante la interfaz web, también se establece que la interfaz web actualice la pagina web cada tres segundos dando origen a la toma de valores que esten por venir de manera continua para poder mostrarse en la interfaz

Los parametros que se visualizan referente a los datos tomados son los siguientes datos:

- Fecha, Hora, Valor, Módulo.

Estos valores son los que se asociaron a las variables de almacenamiento de la base de datos. Se ha optado por definir color blanco y azul a la interfaz web esto con la intención de proporcionar una grata experiencia al usuario que visita la misma.

El modelado para la construcción y la experiencia visual de la interfaz hacia el usuario se la hace de la manera más sencilla y entendible para que cualquier persona pueda relacionarse adecuadamente con la información mostrada.

Para poder resumir la utilidad de la interfaz web establece que los datos sensados han pasado diferentes etapas desde el sensado, la adecuación de los mismos, el almacenamiento de los datos y al fin llegar a la etapa de poder de ser mostrados en la interfaz web, es decir la interfaz web es el paso definitivo y de culminación para el funcionamiento del prototipo, la aplicación web que visualizara el usuario con los datos se muestran en la figura 2.34.

Una vez concluido todos estos cambios se prueba la aplicación web localmente se procede a subir el proyecto a la nube para su respectivo funcionamiento gracias a los servicios que proporciona la plataforma Heroku.

Figura 2.34. Interfaz Web

Esta plataforma proporciona que los servicios corran desde uno de sus servidores, para lo cual se proporciona hacer el uso de Heroku DNS Server con el fin de apuntar al dominio de la aplicación típicamente definido de la siguiente manera (*nombreaplicacion.herokuapp.com*).

Para hacer el uso de esta plataforma se ha creado una cuenta la misma que en este caso se define como campo de estudio, anexado a esto correo electrónico y una contraseña luego se procede a crear el proyecto dentro de la plataforma heroku con la siguiente instrucción

franklin@DESKTOP-S8HDOP6:/*mnt/c/Users/Asus/Desktop/Tesis Interfaz Web$ heroku create*

Ahora se sube el proyecto a la plataforma Heroku mediante el siguiente comando desde interfaz de Debian para Windows

franklin@DESKTOP-S8HDOP6:/*mnt/c/Users/Asus/Desktop/Tesis Interfaz Web$ git push heroku master*

Al correr esta instrucción, se define toda la aplicación y quedara lista para su funcionamiento. A continuación, se visualiza en la figura 2.35. la aplicación web dentro de la plataforma de Heroku

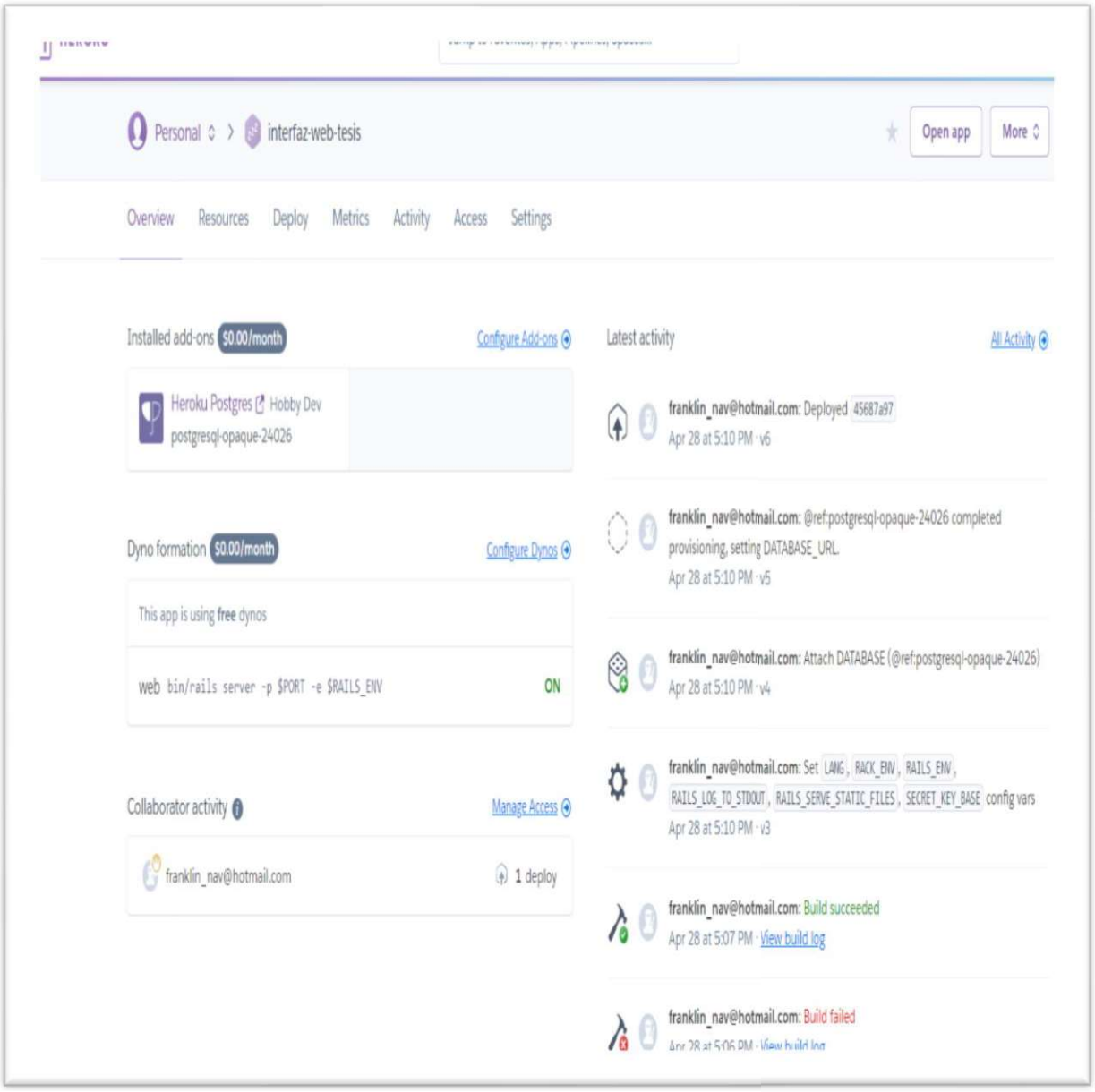


Figura 2.35. Aplicación Web en la plataforma Heroku

Para poder testear adecuadamente el funcionamiento del proyecto dentro de la plataforma Heroku sería primordial probar ingresando a la url del sistema y que se muestre la aplicación web del Sistema de recolección de datos sísmicos levantado el servicio correctamente, con el fin de que cualquier usuario pueda visualizar los datos sísmicos que sean pertinentes.

2.5. SISTEMA DE ENERGIZACIÓN

El sistema de energización para el prototipo se detalla mediante los valores con los que los equipos trabajan, los cuales se detallan la Tabla 2.2.

Tabla 2.2. Valores de Intensidad voltaje y potencia para el funcionamiento de dispositivos

Elemento	Cantidad	Amperaje [mA]	Voltaje [V]	Potencia [mW]
Raspberry Pi 3B	1	100	5	500
Sensor Sw 420	2	30	4.5	135
Modem GSM/GPRS	1	100	4	400
TOTAL				1.035

A partir de estos valores se realiza el siguiente dimensionamiento. Ecuación (2.5.1):

$$Pdiaria = 1{,}035W * \frac{24h}{dia} = 24.84\frac{Wh}{dia} \qquad (2.5.1)$$

A este valor se añade un margen de consumo adicional equivalente a 14 Wh/dia, se procede a calcular la IT. Ecuación (2.5.2)

$$P\ total\ diaria = 24{,}84 + 14 = 38{,}84\frac{Wh}{dia}$$

$$IT = \frac{(P\ total\ diaria)}{V} \qquad (2.5.2)$$

$$IT = \frac{38{,}84}{12} = 3{,}23\frac{Ah}{dia}$$

Ahora, a la corriente total, se procede a multiplicar por un factor de seguridad. Ecuación (2.5.3)

$$IT\ corregido = 3{,}23\frac{Ah}{dia} * 1{,}2 = 3{,}87\frac{Ah}{dia} \qquad (2.5.3)$$

Para obtener el valor de la capacidad de la batería se considera una autonomía de 1 día. Ecuación (2.5.4)

$$Cap.Bateria = 3{,}87\frac{Ah}{dia} * 1\ día = 3{,}876\frac{Ah}{dia} \qquad (2.5.4)$$

En estas condiciones es rentable obtener en em mercado una batería de 12 v y de 7 Ah.

Para determinar el tipo de paneles solares se necesitó saber el índice de radiación solar en la ciudad de Quito. Se analiza con el indicie de radiación solar en las peores condiciones es decir el más bajo el cual es de 4,14 $[\frac{KWh}{m^2}/dia$

Tabla 2.3. Índice de radiación solar [33]

Mes	Ene	Feb	Mar	Abr	May	Junio	Julio	Agos	Sep	Oct	Nov	Dic
Insolación Global $[\frac{KWh}{m^2}/dia]$	4,41	4,48	4,65	4,36	4,27	4,14	4,30	4,62	4,97	4,88	4,94	4,83

Según la tabla 2.3. la peor condición referente al factor de índice de radiación solar es 4,14 $[\frac{KWh}{m^2}/dia]$ por lo tanto, para determinar la corriente pico del sistema se usa este valor. Ecuación (2.5.5)

$$Ipico = \frac{\left(3{,}87\left[\frac{Ah}{dia}\right]\right)*(1\frac{KW}{m^2})}{(4{,}14[\frac{KWh}{m^2}/dia])} \qquad (2.5.5)$$

$$Ipico = 0{,}93A$$

Ante los valores calculados, se busca en el mercado un panel solar que cumpla las expectativas pues es esencial también verificar el tamaño de este, por lo tanto, se adquiere el panel solar de carácter comercial con las especificaciones siguientes:

- P = 10W - V = 12V - Dimensiones: (403 x 288 x 17) mm

Entonces se procede a determinar la configuración y conexión del sistema de energización a nuestro prototipo el cual se lo realiza paso a paso como se indica en la figura 2.36. Aquí se puede decir que la cantidad de energía depende del estado de la batería y el regulador de carga solar ya que es el que permite controlar constantemente que las baterías estén llenas.

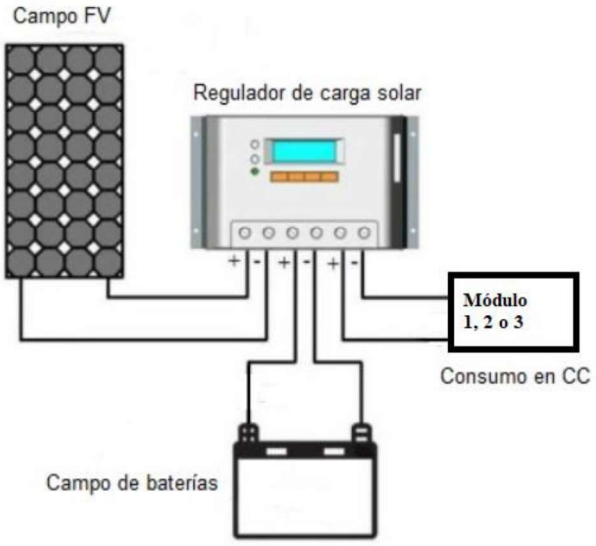


Figura 2.36. Diagrama de conexión al Regulador de Carga Solar

3. RESULTADOS Y DISCUSIÓN

En este capítulo se procederá a realizar diferentes pruebas para el correcto funcionamiento del sistema, cabe mencionar claramente que al ser un proyecto de prueba siempre va a existir una probabilidad de error o falla en alguno de los componentes o servicios de dicho sistema.

3.1. PRUEBAS APLICADAS AL HARDWARE

Los principales componentes de hardware que permiten el correcto funcionamiento del sistema son el equipo transmisor GSM/GPRS A6, los sensores sw420, la plataforma Raspberry Pi 3B, cada uno de estos dispositivos referentes a los tres módulos los cuales forman todo el sistema, esto se describirá en la tabla 3.1.

Tabla 3.1 Prueba de encendido de dispositivos

Dispositivo	Modulo	Tipo de Prueba	Observación
Raspberry Pi 3B	1	Encendido	Ninguna
Modulo GSM/GPRS A6	1	Encendido	Ninguna
Sensores	1	Encendido	Ninguna
Panel solar	1	Encendido	Ninguna
Controlador de carga	1	Encendido	Ninguna
Raspberry Pi 3B	2	Encendido	Ninguna
Modulo GSM/GPRS A6	2	Encendido	Ninguna
Sensores	2	Encendido	Ninguna
Panel solar	2	Encendido	Ninguna
Controlador de carga	2	Encendido	Ninguna
Raspberry Pi 3B	3	Encendido	Ninguna
Modulo GSM/GPRS	3	Encendido	Ninguna
Sensores	3	Encendido	Ninguna
Panel solar	3	Encendido	Ninguna
Controlador de carga	3	Encendido	Ninguna

A continuación, se procede a realizar la prueba de funcionamiento de cada dispositivo en nuestro caso es primordial hacer la prueba a los sensores y al dispositivo de transmisión y recepción GSM/GPRS A6. Esto se verifica en la tabla 3.2.

Tabla 3.2. Prueba de funcionamiento

Dispositivo	Modulo	Tipo de Prueba	Observación
Modulo GSM/GPRS A6	1	Conexión a red GPRS	Ninguno
Sensor 1	1	Sensar movimiento	Ninguno
Sensor 2	1	Sensar movimiento	Ninguno
Modulo GSM/GPRS A6	2	Conexión a red GPRS	Ninguno
Sensor 1	2	Sensar movimiento	Ninguno
Sensor 2	2	Sensar movimiento	Ninguno
Modulo GSM/GPRS A6	3	Conexión a red GPRS	ninguno
Sensor 1	3	Sensar movimiento	Ninguno
Sensor 2	3	Sensar movimiento	Ninguno

Cabe recalcar que adicional a las comprobaciones de los sensores también se hizo una prueba adicional con la ayuda de la plataforma Arduino para poder visualizar la forma de pulsos en el momento que se produce un movimiento figura 3.1. El método en el que se realizó dicha prueba fue sensando valores mediante un pequeño programa de Python en Arduino y se usó la herramienta plotter de la interfaz gráfica de uso de Arduino.

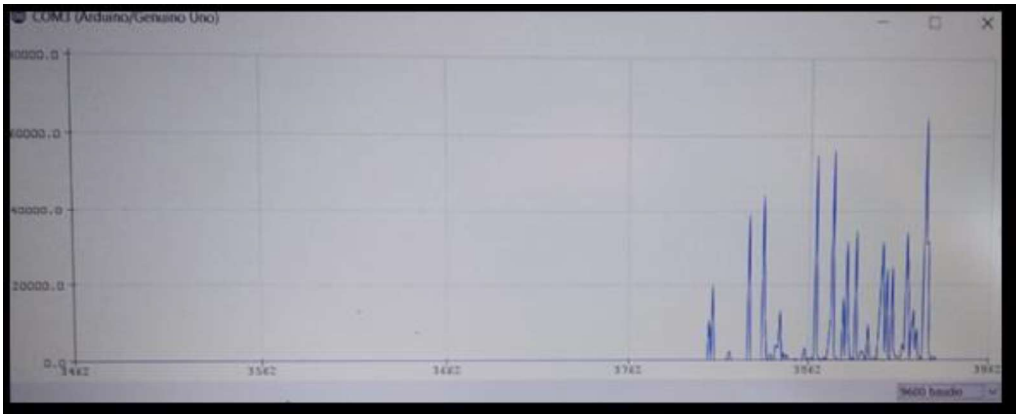


Figura 3.1. Pulsos de un movimiento tomado por el sensor SW420

3.2. PRUEBA APLICADAS AL SOFTWARE

En esta sección hay que enfocarse en la instalación de paquetes y los programas para el correcto funcionamiento de todos y cada uno de los componentes que formaran nuestro sistema, por ejemplo, uno de ellos es los paquetes que se instalaron para la conexión a la red GPRS para lo cual se puede hacer una prueba de conexión en una terminar de Raspbian con los siguientes comandos:

```
$ sudo su
$ cd/etc/ppp/peers
```

$ pon rnet

Después de esto se revisa si se levanta la interfaz PPP esto debe verificarse como se muestra en la figura 3.2.

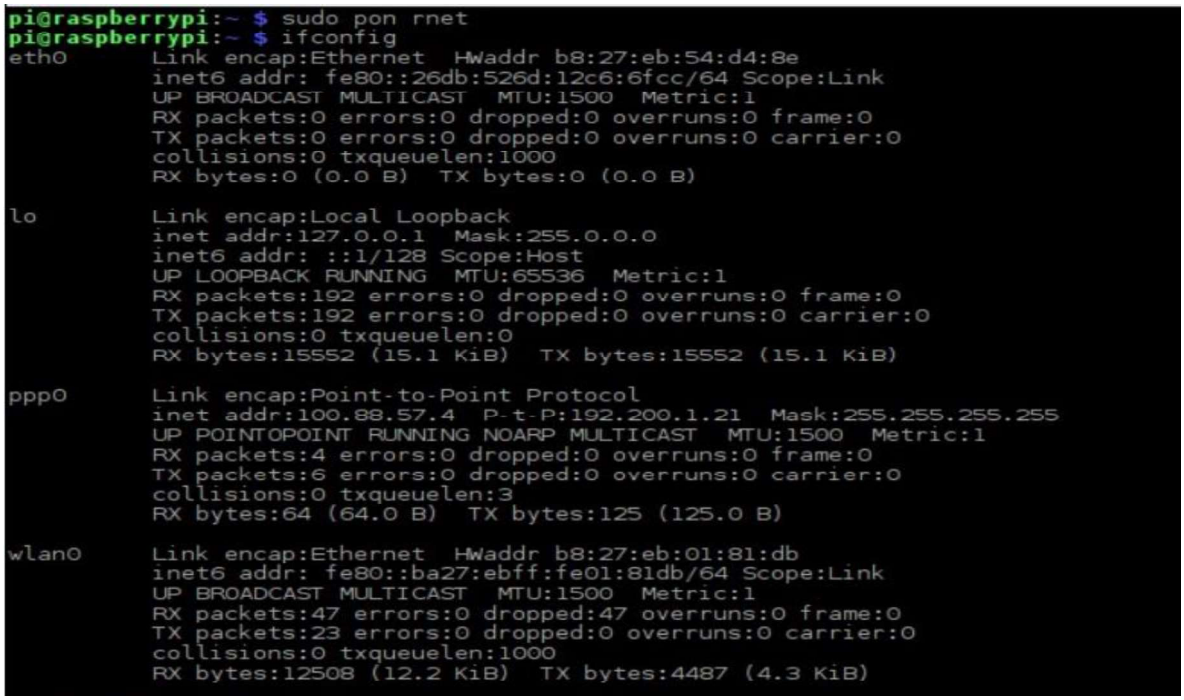


Figura 3.2. Interfaz PPP levantada

Es importante destacar el funcionamiento de la lógica programada para el correcto funcionamiento del sistema de recolección de datos sísmicos. Ya que nos permite centrar en el análisis de las funciones y procesos implementados específicamente dentro del software Tabla 3.3.

Tabla 3.3. Pruebas de funcionamiento de software

Función / Proceso	Tipo de Prueba	Observación
Sensar	Recolección de datos	Ninguna
Sincronizar	sincronismo entre sensor 1 y 2	Ninguna
Sumar_Lista1	Promedio	Ninguna
Lista1	Colección de datos en 10 s.	Ninguna
Lista 2	Colección de datos en 1 m.	Ninguna
Archivo	Lectura de datos del .txt	Ninguna
Parámetros	Adecuación de datos para envio a la Base de Datos	Ninguna
Respuesta	Envio de url y parámetros a la base de datos	Ninguna

Para comprobar el correcto funcionamiento de la base de datos se procede a verificar los valores sensados, estos valores son los que se leen en el archivo de texto que se almacena primero en el Raspberry pi 3B, como se observa en la figura 3.2

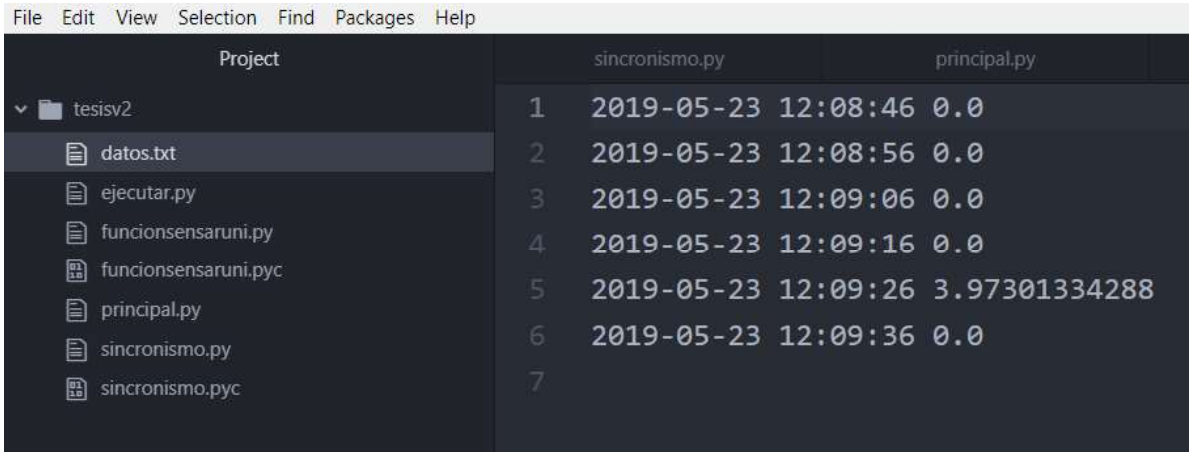


Figura 3.3. Archivo de texto de datos sensados por el módulo 1

Para verificar el funcionamiento de la base de datos se ingresa a la aplicación PGAdmin 4 que se muestra en la figura 3.4. La cual verifica la tabla de registros donde se irán guardando los diferentes valores que se envían desde la plataforma Raspberry Pi 3B ya sea perteneciente a nuestro módulo 1, 2 o 3.

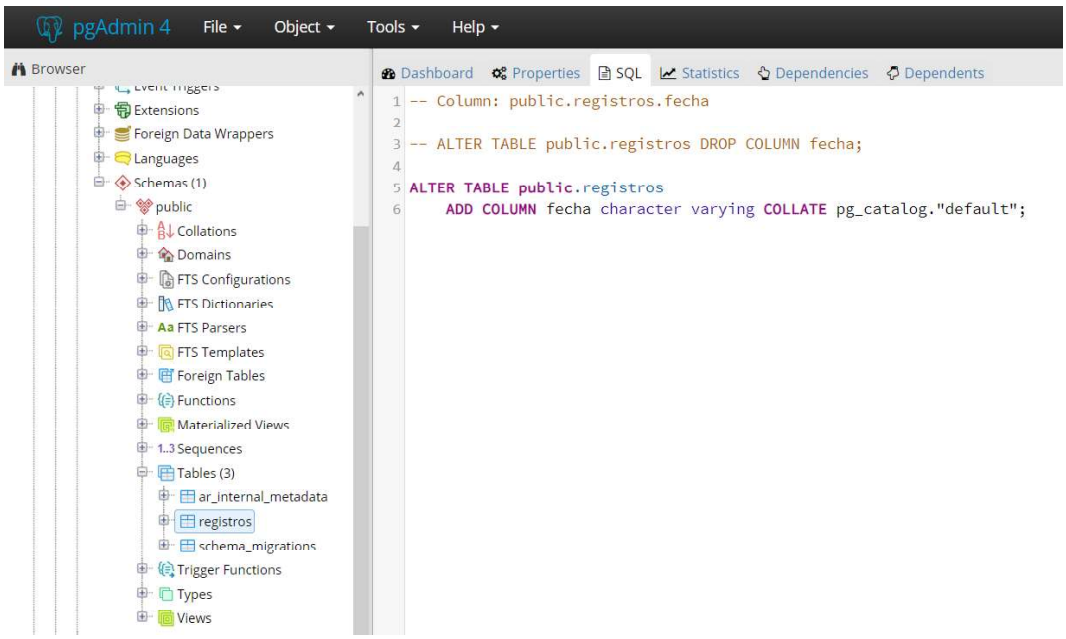


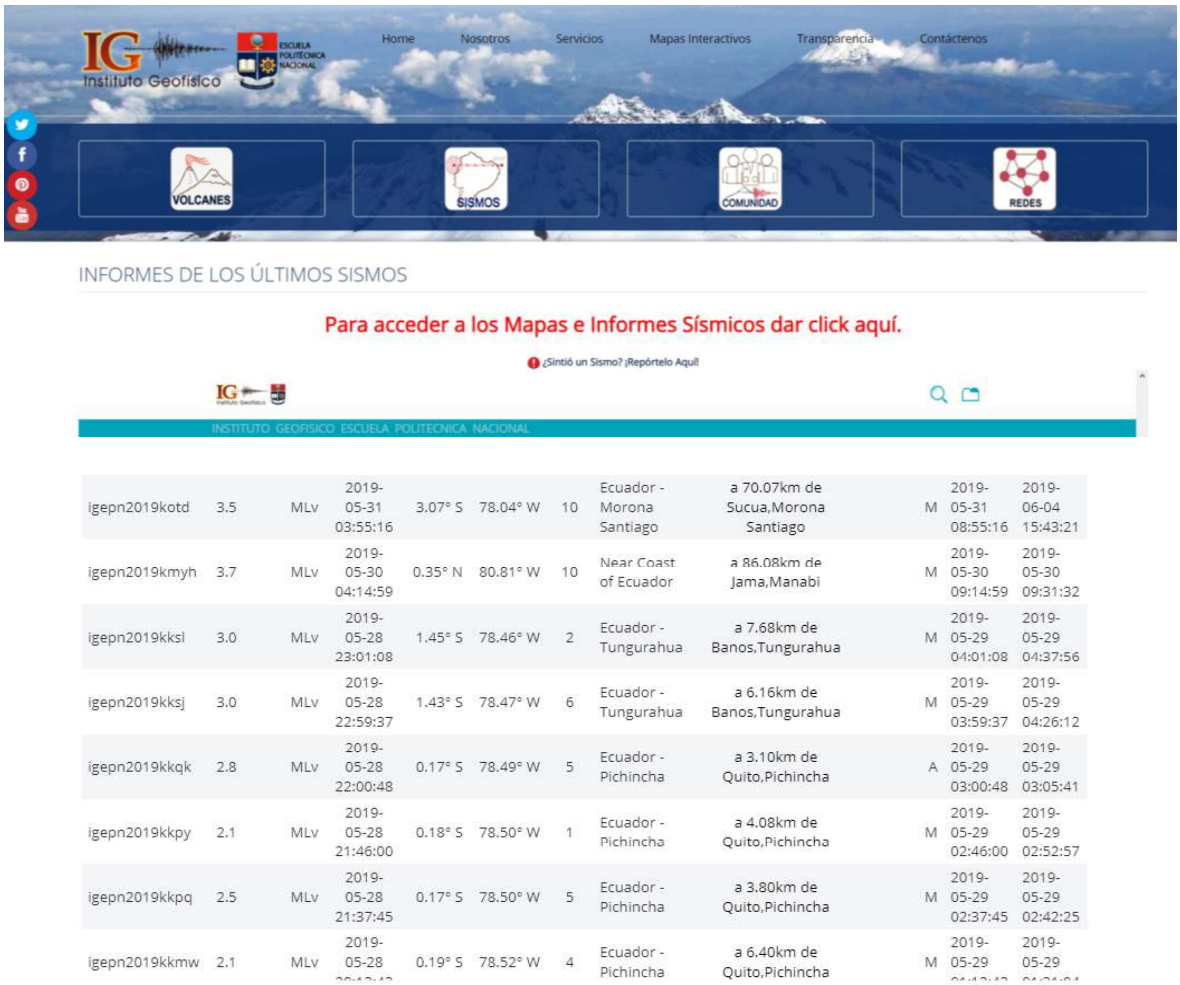
Figura 3.4. Tabla Registro de la Base de Datos

El sistema que se implementa es un prototipo que podría ser una pequeña ayuda a sistemas reales por lo tanto se destaca que en el sistema no se usó sensores de precisión ni equipos especiales como sismógrafos se optó por una alternativa diferente mediante el

uso de la plataforma Raspberry pi 3B, para poder comparar las mediciones establecidas entre el uno y el otro sistema se recalca en tratar de entender que el uno es un sistema especializado y el otro se podría decir que es un primer prototipo, el cual obviamente puede tener muchas mejoras asi como también puede tener algunos fallos en trabajo de campo.

Algunos valores que se ha podido comparar se observan con mayor especificación en la tabla 3.4.

La comparación se hace desde la página web del Sistema del Instituto Geofísico de la Politécnica Nacional. Cuya página web se puede fácilmente visualizar ingresando a la siguiente dirección URL: https://www.igepn.edu.ec/ultimos-sismos. Esta página web se visualiza en la Figura 3.5. Mientras que el Sistema de recolección de datos sísmicos basado en tecnología GPRS y la plataforma Raspberry Pi 3B posee la siguiente dirección URL: https://interfaz-web-tesis.herokuapp.com Esta página web se visualiza en la Figura 3.6.



igepn2019kotd	3.5	MLv	2019-05-31 03:55:16	3.07° S	78.04° W	10	Ecuador - Morona Santiago	a 70.07km de Sucua,Morona Santiago	M	2019-05-31 08:55:16	2019-06-04 15:43:21
igepn2019kmyh	3.7	MLv	2019-05-30 04:14:59	0.35° N	80.81° W	10	Near Coast of Ecuador	a 86.08km de Jama,Manabi	M	2019-05-30 09:14:59	2019-05-30 09:31:32
igepn2019kksl	3.0	MLv	2019-05-28 23:01:08	1.45° S	78.46° W	2	Ecuador - Tungurahua	a 7.68km de Banos,Tungurahua	M	2019-05-29 04:01:08	2019-05-29 04:37:56
igepn2019kksj	3.0	MLv	2019-05-28 22:59:37	1.43° S	78.47° W	6	Ecuador - Tungurahua	a 6.16km de Banos,Tungurahua	M	2019-05-29 03:59:37	2019-05-29 04:26:12
igepn2019kkqk	2.8	MLv	2019-05-28 22:00:48	0.17° S	78.49° W	5	Ecuador - Pichincha	a 3.10km de Quito,Pichincha	A	2019-05-29 03:00:48	2019-05-29 03:05:41
igepn2019kkpy	2.1	MLv	2019-05-28 21:46:00	0.18° S	78.50° W	1	Ecuador - Pichincha	a 4.08km de Quito,Pichincha	M	2019-05-29 02:46:00	2019-05-29 02:52:57
igepn2019kkpq	2.5	MLv	2019-05-28 21:37:45	0.17° S	78.50° W	5	Ecuador - Pichincha	a 3.80km de Quito,Pichincha	M	2019-05-29 02:37:45	2019-05-29 02:42:25
igepn2019kkmw	2.1	MLv	2019-05-28	0.19° S	78.52° W	4	Ecuador - Pichincha	a 6.40km de Quito,Pichincha	M	2019-05-29	2019-05-29

Figura 3.5. Sistema del Instituto Geofísico [34]

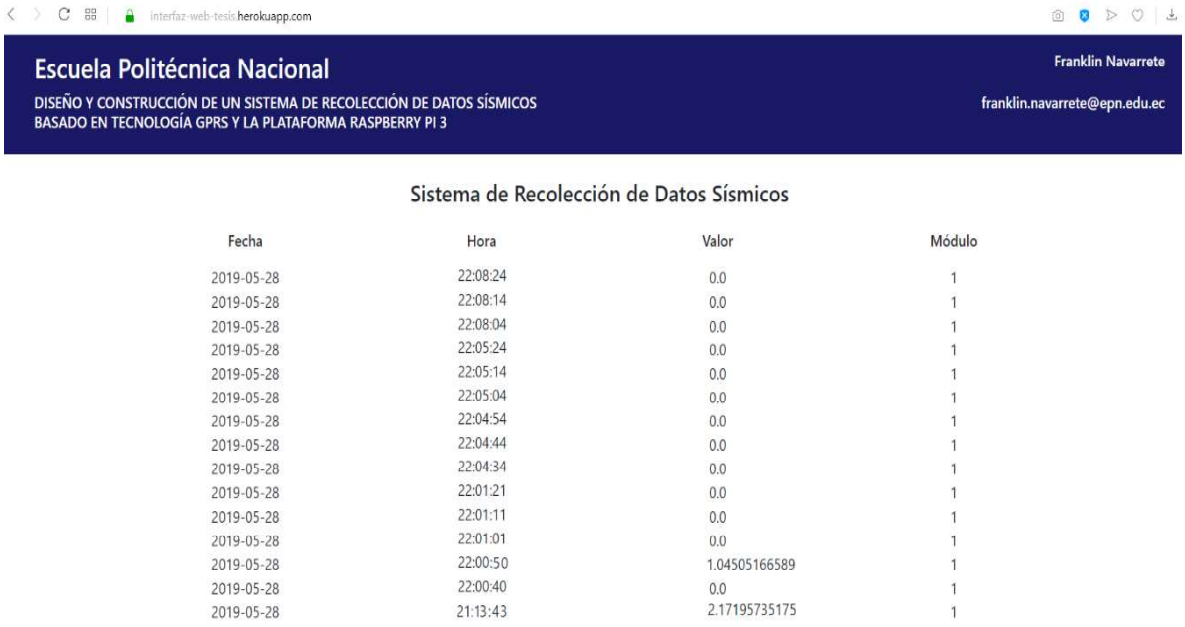


Figura 3.6. Sistema de Recolección de datos sísmicos (Prototipo)

Los valores que se pueden comparar muchas veces serán difíciles de comparar ya que los sensores que se usan son de sentido muy comercial y más no de precisión. Las muestras para la comparación de valores medidos referentes al prototipo construido se lo hacen en base a los valores que se dan en la página del Instituto Geofísico, ya que al observar esos valores se permite establecer una comparación cuando se da un movimiento telúrico simplemente observando la página web pues en su base de datos se almacenan las mediciones. Se realizó las pruebas según recomendaciones de personal del Instituto Geofísico ya que debido a problemas de logística no se pudo establecer una comparación entre equipos ubicados en el mismo lugar geográfico, es por eso que las pruebas se dan de la siguiente manera el prototipo se lo mantenía encendido durante gran periodo de tiempo y los valores que se pude captar se compararon con los valores sísmicos recolectados de la página web del Instituto Geofísico. Tabla 3.4.

Tabla 3.4. Comparación de medidas sismográficas

Fecha	Hora	Medida Prototipo (Richter)	Medida I. Geofísico (Richter)
2019-11-09	08:37:20	3,8	4,1
2019-11-06	08:02:27	3,0	3,6
2019-11-05	23:02:18	3,3	3,6
2019-11-02	21:15:40	3,3	3,6
2019-10-30	04:29:18	3,3	3,5

Los valores captados se los puede visualizar en el ANEXO D, estos datos como se observa no son exactamente iguales y eso se da ante lo ya mencionado que no se puede obtener valores precisos usando sensores comerciales en comparación con sensores de precisión.

También se explica que mientras el prototipo no tiende a sensar algún movimiento sísmico su valor será cuantificado como cero, razón por la cual en la Tabla 3.4. se especifican solo los valores medidos por el prototipo y que permiten hacer una comparación adecuada referente a los valores suministrados por la página del Instituto Geofísico.

3.3. COSTO REFERENCIAL DEL PROTOTIPO

El costo referencial del Sistema se basa en la tabla 3.5.

Tabla 3.5. Costo referencial

Equipo	Cantidad	Valor unitario	Valor total
Kit Raspberry Pi 3B (Memoria SD, Carcaza, cables conectores)	3	86,00	258,00
Modulo GSM/GPRS	3	18,00	54,00
Sensores de movimiento	6	5,00	30,00
Controlador de carga Solar	3	18,00	54,00
Panel Solar	3	20,00	60,00
Batería	3	24,00	72,00
Cable USB de carga	3	3,00	9,00
Sub TOTAL			537,00
I.V. A			64,44
TOTAL			601,44

También se debe tomar en cuenta los costos de materiales adicionales que fueron parte del proceso de elaboración del prototipo, es decir, otros elementos que significaron algún

costo en la construcción del sistema para determinar el precio total, esto se refleja en la tabla 3.6

Tabla 3.6. Costo de materiales

Equipo	Cantidad	Valor unitario	Valor Total
Kit de cables macho hembra	2	3,50	7,00
Placa de Baquelita	3	0,60	1,18
Estaño	1	1,00	1,00
Acido para baquelita	1	0,50	0,50
Conectores	3	0,20	0,60
Chips telefónicos	3	5,00	15,00
SUBTOTAL			15,28
I.V. A			1,83
TOTAL			17,11

Además, se determina que el costo de mano de obra se establece por las horas trabajadas en el desarrollo del software dentro de la implementación del sistema. El tiempo invertido está en alrededor de 130 horas, se estima que el costo de hora por trabajo en software se basa entre $10,00 a $15,00, para este caso se estima un costo de $12,00. Por lo tanto, en la tabla 3.7. se define el costo por desarrollo del sistema, en la parte de software.

Tabla 3.7. Costo por Desarrollo de Software

Especificación	Tiempo estimado (h.)	Valor unitario por hora ($)	Costo total $
Desarrollo de software	130,00	12,00	1.560,00

Ahora, se procede a definir el coto total del sistema el cual se lo hace tomando los costos de cada tabla, por lo que se tiene un **Costo Total de $2.178,55**.

Este precio comparado con valores extremadamente altos es rentable, pero cabe mencionar que sistemas de este tipo profesional usan sensores de precisión y específicamente sensores sísmicos denominados geófonos de precisión sísmica en los cuales se puede verificar profundidad, distancia, frecuencia sísmica y velocidad de propagación.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El sistema permitirá que cualquier persona tenga acceso a los datos de este, puede funcionar como dispositivo de información ante alertas sísmicas sin limitaciones a los altos costos que este tipo de equipos implica.
- El sistema permitirá contribuir con el medio ambiente porque ya no se generaría energía con combustible fósil debido a que el sistema está constituido con energía renovable.
- El sistema permitirá toma de decisiones en base a los análisis de resultados obtenidos que permitan a entidades y personas que lo requieren desarrollar mecanismos e instrumentos para prevenir, mitigar, atender e incorporar acciones para reducir desastres y emergencias.
- Este sistema podría ser una base para futuros estudios de riesgos sísmicos para la elaboración de planes de emergencia ya que permite una recolección de datos los mismos que pueden ser analizados por especialistas en el tratamiento de datos de Movimientos Sísmicos.
- Dentro del prototipo los servicios en la nube facilitan el manejo y desarrollo de la base de datos y la aplicación web, evitando asi tener gastos adicionales que se darían al implementar estos físicamente.

4.2 RECOMENDACIONES

- En cuanto a mejoras ante el sistema construido se puede mencionar que sería ideal trabajar con sensores de precisión lo cual es adecuado para registros de movimientos sísmicos, pero en cuanto a esta idea también involucra un razonamiento ante el tema costo beneficio.

- Referente al software del sistema en la experiencia que se ha tenido se puede determinar que también se le podría hacer varios cambios para un mejor rendimiento y funcionamiento, en cuanto a tiempos de tratamiento de las señales y otras funciones que permiten una mejor automatización y autonomía del sistema.

- En este proyecto se propuso la conexión para el envió de datos con la Tecnología GPRS, pero cabe mencionar que en la actualidad se tiene disponibles más tecnologías y con mejores servicios los cuales permitirían optimizar el sistema en cuanto a la transmisión y recepción de los datos recolectados por los sensores de movimientos.

- Sería factible proporcionar un sistema de respaldo energético para los módulos del proyecto ya que, si en caso falla el suministro de energía del sistema, poder proveer este respaldo seria esencial ya que no dejaría de funcionar el prototipo.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] Martínez C, “Teorías que marcaron el estudio científico de los sismos”, 18-sep-2017, [En línea]. Disponible en: http://ciencia.unam.mx/leer/647/teorias-que-marcaron-el-estudio-cientifico-de-los-sismos

[2] Granados K, “Placas Tectónicas”, 18-sep-2017, [En línea]. Disponible en: https://magicanaturaleza.com/c-sismos/placas-tectonicas/

[3] Martínez C, “Teorías que marcaron el estudio científico de los sismos”, 10-sep-2010 [En línea]. Disponible en: http://ciencia.unam.mx/leer/647/teorias-que-marcaron-el-estudio-cientifico-de-los-sismos

[4] EcuRed,” Telefonía móvil 2G”, 12-ene-2019, [En línea]. Disponible en: https://www.ecured.cu/Telefonía_móvil_2G

[5] Prieto F, “Elementos de una red GPRS”, versión 1, página 70, 11-may-2014.

[6] Hernández Ángela, “REDES Y SERVICIOS MÓVILES GSM Y GPRS”, 18-nov-2019, [En linea]. Disponible en: https://docplayer.es/14160231-Redes-y-servicios-moviles-gsm-y-gprs.html

[7] Ruby on Rails, [En linea]. Disponible en: https://rubyonrails.org, [Consultado: 10-agosto-2018].

[8] Instructables circuits, [En línea]. Disponible en: https://www.instructables.com/id/All-Weather-Solar-Powered-Raspberry-Pi/

[9] Llamas L, “Modelos y Características de Raspberry Pi”, 17-nov-2017, [En Línea]. Disponible en: https://www.luisllamas.es/modelos-de-raspberry-pi/

[10] Raspberry Pi, Disponible en: https://www.raspberrypi.org/products/raspberry-pi-1-model-a-plus/

[11] Cristian Rus, “Raspberry Pi 4”, 24-jun-2019, [En Línea]. Disponible en: https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica

[12] User Manual, Modulo GSM/GPRS A6, página 16

[13] AG Electrónica1 S.A., “Manual de Usuario modulo GSM/GPRS”, 09-ene-2017.

[14] Manual de Usuario, Sensor SW-420, página 3,4,6

[15] Createc, Manual de usuario, Sensor SW-420, [En Línea], Disponible en: https://createc3d.com/shop/es/modulos-y-sensores/403-comprar-sensor-vibracion-sw-420-compatible-arduino-precio-oferta.html

[16] Energía Solar, “Diferencia existente entre paneles solares”, 21-sep-2017, [En línea], Disponible en: https://autosolar.es/blog/aspectos-tecnicos/que-diferencia-existe-entre-los-paneles-solares-termicos-y-los-paneles-fotovoltaicos

[17] Especificaciones Panel Solar, Manual de Usuario, [En línea], Disponible en: https://www.linio.com.mx/p/panel-solar-fotovoltaico-policristalino

[18] Arancibia C, “Energías del Sol”, 23-jun-2010, versión 1.2, página 3.

[19] Best R, “Paneles Solares”, [En Línea], Disponible en: https://www.areatecnologia.com/electricidad/paneles-solares.html

[20] Ruiz Gonzalo, “¿Cómo funcionan los paneles Solares?”, [En Línea], Disponible en: https://erenovable.com/como-funcionan-los-paneles-solares/

[21] Best R, “Paneles Solares”, [En Línea], Disponible en: https://www.areatecnologia.com/electricidad/paneles-solares.html

[22] Heroku, [En linea]. Disponible en: https://www.heroku.com/platform. [Consultado 12-Julio-2019]

[23] Feldgen M, “Modelo de Referencia OSI y capa de Aplicación”, pagina 2, cuadro 3, año 2002.

[24] Hernández E, “COMPARACIÓN DE LOS MODELOS OSI Y TCP/IP”, [En línea], Disponible en: https://www.uaeh.edu.mx/scige/boletin/huejutla/n10/r1.html

[25] Raffino M, “Conceptos de Http”, 29-Ago-2019, [En Línea], Disponible en: https://concepto.de/http/

[26] Controlador de carga, [En linea]. Disponible en: https://cumbrepuebloscop20.org/energias/solar/controlador-carga/ [Consultado 18 - septiembre - 2019].

[27] Manual de instalación de usuario, “Panel solar 10W”

[28] Panel solar, [En línea]. Disponible en: https://www.bateriadelitiorecargable.com/es/policristalino-panel-solar-12v24v/panel-solar-policristalino-12v-10w-718html [Consultado 20-julio-2019]

[29] MICROLAB, “Modulo GSM/GPRS A6 con Raspberry”, 25-sep-2018, [En Línea], Disponible en: https://makers-blog.azurewebsites.net/index.php/2018/09/25/modulo-gsm-gprs-a6-con-raspberry

[30] Datasheet, “Vibration Sensor Module SW420”, 06-ene-2018, [En Línea], Disponible en: http://makersindia.org/t77-2/

[31] Plasencia Eduardo, “Que es y porque aprender SQL”, [En Línea], Disponible en: https://devcode.la/blog/que-es-sql/ [Consultado 7-septiembre-2019].

[32] https://www.montereyinstitute.org/courses/DevelopmentalMath/TEXTGROUP-15-19_RESOURCE/U18_L4_T2_text_container_es.html

[33] Stackoverflow, Python, [Consultado 25-junio-2019].

[34] CONELEC, Atlas solar del Ecuador con fines de generación eléctrica, Quito: Corporación para la Investigación Energética, 2008.

[35] Instituto Geofísico, [En línea], Disponible en: https://www.igepn.edu.ec/ultimos-sismos

[Consultado 12-octubre-2019].

ANEXOS

ANEXO A. Clase Sincronismo

ANEXO B. Clase Principal

ANEXO C. Clase Registro Controlador

ANEXO D. HTML de la aplicación web principal

ANEXO E. Comparación de Medidas de Movimientos Sísmicos

ANEXO A: CLASE SINCRONISMO

```
import RPi.GPIO as GPIO
import time
import math
import funcionsensaruni
from multiprocessing import Pool

def sumarLista1(A):
        suma = 0
        for m in range(0,len(A)):
            suma=suma+A[m]
        promedio = suma/2
        return promedio
def sincronizar():

    p= Pool(2)
    resp = p.map(funcionsensaruni.sensar,[27,17])
    valor = sumarLista1(resp)
    return valor
```

ANEXO B: CLASE PRINCIPAL

```
import RPi.GPIO as GPIO
import time
import requests
from datetime import datetime
from datetime import timedelta
import math
global lista
import sincronismo
lista = []

completo = 0
i=0

while completo ==0:
    resultado = str(sincronismo.sincronizar())

    print (resultado)
    fecha_actual = datetime.now().strftime("%Y-%m-%d")
    hora_actual = datetime.now().strftime("%H:%M:%S")
    fecha_a = str(fecha_actual)
    hora_a = str(hora_actual)
    datoF = fecha_a + ' ' + hora_a + ' ' + resultado

    i=i+1
    lista.append(datoF)

    if i==6:
        print (lista)

        completo =1

mi_dir = "/home/pi/Desktop/proyectoT/tesisv2/datos.txt"
f = open(mi_dir, 'w')
for l in lista:
    f.write(l + '\n')
f.close()

archivo = open("datos.txt", "r")
arreglo = []
for linea in archivo.readlines():
    arreglo += linea.rstrip().split(" ")
archivo.close()
print(arreglo)

modulo = 1
url = "https://interfaz-web-
tesis.herokuapp.com/registros/guardar_registro/"
parametros = {'fecha1': arreglo[0], 'hora1': arreglo[1], 'valor1':
arreglo[2], 'modulo1': modulo,
                'fecha2': arreglo[3], 'hora2': arreglo[4], 'valor2':
arreglo[5], 'modulo2': modulo,
                'fecha3': arreglo[6], 'hora3': arreglo[7], 'valor3':
arreglo[8], 'modulo3': modulo,
```

```
                'fecha4': arreglo[9], 'hora4': arreglo[10], 'valor4':
arreglo[11], 'modulo4': modulo,
                'fecha5': arreglo[12], 'hora5': arreglo[13], 'valor5':
arreglo[14], 'modulo5': modulo,
                'fecha6': arreglo[15], 'hora6': arreglo[16], 'valor6':
arreglo[17], 'modulo6': modulo}

respuesta = requests.get(url = url, params = parametros)

print("Respuesta: ", respuesta)
```

ANEXO C: CLASE REGISTRO CONTROLADOR

```
class RegistrosController < ApplicationController
  def index
    @registros = Registro.all.order(id: :desc)
  end

  def guardar_registro
    # Valor 1
    registro = Registro.new
    registro.fecha = params[:fecha1]
    registro.hora = params[:hora1]
    registro.valor = params[:valor1]
    registro.modulo = params[:modulo1]
    registro.save
    # Valor 2
    registro = Registro.new
    registro.fecha = params[:fecha2]
    registro.hora = params[:hora2]
    registro.valor = params[:valor2]
    registro.modulo = params[:modulo2]
    registro.save
    # Valor 3
    registro = Registro.new
    registro.fecha = params[:fecha3]
    registro.hora = params[:hora3]
    registro.valor = params[:valor3]
    registro.modulo = params[:modulo3]
    registro.save
    # Valor 4
    registro = Registro.new
    registro.fecha = params[:fecha4]
    registro.hora = params[:hora4]
    registro.valor = params[:valor4]
    registro.modulo = params[:modulo4]
    registro.save
    # Valor 5
    registro = Registro.new
    registro.fecha = params[:fecha5]
    registro.hora = params[:hora5]
    registro.valor = params[:valor5]
    registro.modulo = params[:modulo5]
    registro.save
    # Valor 6
    registro = Registro.new
    registro.fecha = params[:fecha6]
    registro.hora = params[:hora6]
    registro.valor = params[:valor6]
    registro.modulo = params[:modulo6]
    registro.save

    redirect_to controller: :registros, action: :index
  end
end
```

ANEXO D: HTML DE LA APLICACIÓN WEB PRINCIPAL

```
<div class="container my-4">
  <div class="row">
    <div class="col-12 text-center">
      <p class="h4">Sistema de Recolección de Datos Sísmicos</p>
    </div>
  </div>

  <div class="row mb-2 mt-3">
    <div class="col-3 text-center">
      <p class="h6">Fecha</p>
    </div>
    <div class="col-3 text-center">
      <p class="h6">Hora</p>
    </div>
    <div class="col-3 text-center">
      <p class="h6">Valor</p>
    </div>
    <div class="col-3 text-center">
       <p class="h6">Módulo</p>
    </div>
  </div>

  <% if @registros.size > 0 %>
    <% @registros.map do |r| %>
      <div class="row">
        <div class="col-3 text-center">
          <%= r.fecha %>
        </div>
        <div class="col-3 text-center">
          <%= r.hora %>
        </div>
        <div class="col-3 text-center">
          <%= r.valor %>
        </div>
        <div class="col-3 text-center">
           <%= r.modulo %>
        </div>
      </div>
    <% end %>
  <% else %>
    <div class="row">
      <div class="col-12 text-center">
        No existen registros
      </div>
    </div>
  <% end %>

</div>
```

ANEXO E: COMPARACIÓN DE MEDIDAS DE MOVIMIENTOS SÍSMICOS

INSTITUTO GEOFISICO ESCUELA POLITECNICA NACIONAL

ID Evento	Mag	Tipo	Hora Local	Lat	Long	Prof	Region	Ciudad mas cercana	Modo	Hora UTC	Update
igepn2019vzbw	4.1	MLv	2019-11-09 08:37:15	3.34° S	80.87° W	5	Peru-Ecuador Border Region	a 73.24km de Huaquillas,El Oro	A	2019-11-09 13:37:15	2019-11-09 22:56:40
igepn2019vtkh	3.6	M	2019-11-06 06:02:27	1.45° S	80.90° W	10	Near Coast of Ecuador	a 15.53km de Puerto Lopez,Manabi	M	2019-11-06 11:02:27	2019-11-06 11:35:18
igepn2019vswt	3.9	MLv	2019-11-05 23:11:27	0.44° N	80.52° W	10	Near Coast of Ecuador	a 58.50km de Muisne,Esmeraldas	M	2019-11-06 04:11:27	2019-11-06 04:23:09
igepn2019vswl	3.6	MLv	2019-11-05 23:02:18	2.24° S	79.72° W	50	Ecuador - Guayas	a 13.73km de Duran,Guayas	M	2019-11-06 04:02:18	2019-11-06 18:43:26
igepn2019vnps	3.6	M	2019-11-03 01:56:22	0.97° S	78.77° W	5	Ecuador - Cotopaxi	a 7.77km de Pujili,Cotopaxi	M	2019-11-03 06:56:22	2019-11-03 07:04:17
igepn2019vngl	3.6	MLv	2019-11-02 21:15:39	2.42° S	80.21° W	16	Ecuador - Guayas	a 30.47km de Playas,Guayas	M	2019-11-03 02:15:39	2019-11-03 02:24:09
igepn2019vmew	4.3	MLv	2019-11-02 07:20:04	2.57° S	81.04° W	2	Near Coast of Ecuador	a 39.28km de Salinas,Santa Elena	M	2019-11-02 12:20:04	2019-11-02 12:30:48
igepn2019vkwl	2.3	MLv	2019-11-01 13:57:08	1.36° S	78.48° W	14	Ecuador - Tungurahua	a 6.60km de Banos,Tungurahua	M	2019-11-01 18:57:08	2019-11-01 19:06:22
igepn2019vgms	3.5	MLv	2019-10-30 04:29:18	2.86° S	77.64° W	5	Ecuador - Morona Santiago	a 74.05km de Sucua,Morona Santiago	M	2019-10-30 09:29:18	2019-10-30 09:38:52

Sistema de Recolección de Datos Sísmicos

Fecha	Hora	Valor	Módulo
2019-11-09	08:38:00	0.0	1
2019-11-09	08:37:50	0.0	1
2019-11-09	08:37:40	0.0	1
2019-11-09	08:37:30	0.0	1
2019-11-09	08:37:20	3.8	1
2019-11-09	08:37:10	0.0	1

Escuela Politécnica Nacional
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECOLECCIÓN DE DATOS SÍSMICOS BASADO EN TECNOLOGÍA GPRS Y LA PLATAFORMA RASPBERRY PI 3
Franklin Navarrete
franklin.navarrete@epn.edu.ec

Sistema de Recolección de Datos Sísmicos

Fecha	Hora	Valor	Módulo
2019-11-06	06:02:37	0.0	2
2019-11-06	06:02:27	3.0	2
2019-11-06	06:02:17	0.0	2
2019-11-06	06:02:07	0.0	2
2019-11-06	06:01:57	0.0	2
2019-11-06	06:01:47	0.0	2

Escuela Politécnica Nacional
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECOLECCIÓN DE DATOS SÍSMICOS BASADO EN TECNOLOGÍA GPRS Y LA PLATAFORMA RASPBERRY PI 3
Franklin Navarrete
franklin.navarrete@epn.edu.ec

Sistema de Recolección de Datos Sísmicos

Fecha	Hora	Valor	Módulo
2019-11-05	23:02:58	0.0	3
2019-11-05	23:02:48	0.0	3
2019-11-05	23:02:38	0.0	3
2019-11-05	23:02:28	0.0	3
2019-11-05	23:02:18	3.3	3
2019-11-05	23:02:08	0.0	3

Franklin Navarrete
franklin.navarrete@epn.edu.ec

Sistema de Recolección de Datos Sísmicos

Fecha	Hora	Valor	Módulo
2019-11-02	21:15:40	3.2	2
2019-11-02	21:15:30	0.0	2
2019-11-02	21:15:20	0.0	2
2019-11-02	21:15:20	0.0	2
2019-11-02	21:15:10	0.0	2
2019-11-02	21:14:59	0.0	2

Escuela Politécnica Nacional
DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE RECOLECCIÓN DE DATOS SÍSMICOS BASADO EN TECNOLOGÍA GPRS Y LA PLATAFORMA RASPBERRY PI 3
Franklin Navarrete
franklin.navarrete@epn.edu.ec

Sistema de Recolección de Datos Sísmicos

Fecha	Hora	Valor	Módulo
2019-10-30	04:29:38	0.0	1
2019-10-30	04:29:28	0.0	1
2019-10-30	04:29:18	3.3	1
2019-10-30	04:29:08	0.0	1
2019-10-30	04:28:58	0.0	1
2019-10-30	04:28:48	0.0	1

ORDEN DE EMPASTADO