UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

ESCUELA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MECATRÓNICA

TEMA:

"SISTEMA ABIERTO PARA SEGUIMIENTO DE POSICIÓN GLOBAL: SOFTWARE DE GESTIÓN DE DATOS"

AUTOR: KEVIN PAUL YÉPEZ RUIZ

DIRECTOR: CARLOS XAVIER ROSERO CHANDI

IBARRA-ECUADOR
2020

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. **IDENTIFICACIÓN DE LA OBRA**

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	**0401711759**		
APELLIDOS Y NOMBRES:	**YÉPEZ RUIZ KEVIN PAUL**		
DIRECCIÓN:	**Juana Atabalipa 15-03 y Hernán Gonzales de Saa**		
EMAIL:	kpyepezr@utn.edu.ec – paulye203@hotmail.com		
TELÉFONO FIJO:	**062 651 319**	**TELÉFONO MÓVIL:**	**0998675962**

DATOS DE LA OBRA	
TÍTULO:	**"SISTEMA ABIERTO PARA SEGUIMIENTO DE POSICIÓN GLOBAL: SOFTWARE DE GESTIÓN DE DATOS"**
AUTOR (ES):	**YÉPEZ RUIZ KEVIN PAUL**
FECHA: DD/MM/AAAA	**06 de febrero del 2020**
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	☐ **PREGRADO**
TITULO POR EL QUE OPTA:	**INGENIERO EN MECATRÓNICA**
ASESOR /DIRECTOR:	**CARLOS XAVIER ROSERO CHANDI**

2. **CONSTANCIAS**

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 06 días del mes de febrero de 2020

EL AUTOR:

(Firma)……
Nombre: Kevin Paul Yépez Ruiz

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CERTIFICACIÓN

En calidad de director del trabajo de grado "SISTEMA ABIERTO PARA SEGUIMIENTO DE POSICIÓN GLOBAL: SOFTWARE DE GESTIÓN DE DATOS", presentado por el egresado KEVIN PAUL YÉPEZ RUIZ, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, 6 de Febrero del 2020

Carlos Xavier Rosero Chandi
DIRECTOR DE TESIS

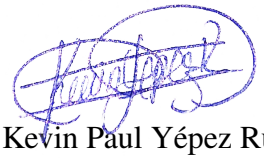
UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló sin violar derechos de autor de terceros, por lo tanto la obra es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, 6 de Febrero del 2020



Kevin Paul Yépez Ruiz
C.I.: 0401711759

UNIVERSIDAD TÉCNICA DEL NORTE

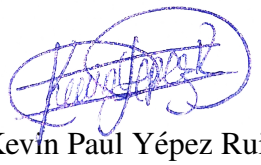
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DECLARACIÓN

Yo, Kevin Paul Yépez Ruiz con cédula de identidad Nro. 0401711759, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado 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 Universidad Técnica del Norte - Ibarra, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Ibarra, 6 de Febrero del 2020



Kevin Paul Yépez Ruiz
C.I.: 0401711759

Agradecimiento

A Dios y a la vida por darme la oportunidad de alcanzar una meta más y en el camino haber aprendido el valor de las cosas pequeñas y que la perseverenacia y esfuerzo vienen cargadas de cosas mejores.

A mi compañera de vida Cristina por ser mi apoyo y juntos haber trazado el camino para salir adelante, por siempre estar en los momentos duros y saber la manera de ser la guia y la calma para superar todo juntos.

A mis padres Jorge y Nora por ser la base y el ejemplo de vida en la cual siempre me he guiado, por el apoyo a lo largo de mi vida y por estar presentes en todos los pasos importantes, por ser una de las fuerzas para siempre salir adelante y no dejar de esforzarme para lograr mis objetivos.

A toda mi familia por ser parte importante de mi vida, de mi crecimiento como persona y el ejemplo para siempre avanzar paso apaso pero sin decaer.

A mis hijas Sophie y Paulie por ser la motivación más grande en mi vida, por ser la razon por la cual nunca dejaré de luchar y seguir avenzando, y para poder ser el ejemplo en el cual se guien para cumplir sus objetivos.

A mi director Xavier por haber sido una base en mi carrera universitaria, y por todo el apoyo y consejos que supo brindarme en mis años de estudiante, a mis amigos que se convirtieron en una parte importante para lograr este objetivo.

Dedicatoria

A mis padres con todo el cariño por nunca dejar de apoyarme y ser una motivación para haber culminado esta etapa tan importante de mi vida.

A Criss,Sophie y Paulie, mi pequeña y hermosa familia por la cual siempre lucharé para seguir siendo un buen ejemplo como padre, esposo y como persona.

A mis amigos Roberto, Israel y Eli por haber estado en todo este camino, por la amistad más allá de lo académico, por las risas, el estrés, y practicamente todo el tiempo que vivimos en la universidad y por que la amistad perdure por siempre.

A mi familia ya que todos son una parte importante y una razón para luchar por mis objetivos.

Kevin Paul Yépez Ruiz

Resumen

El presente trabajo busca aportar a aquellos estudiantes y profesionales que se dedican a la investigación e implementación de aplicaciones que tienen que ver con rastreo o monitoreo satelital, prestando servicio de manera gratuita para el uso de la plataforma, en tiempo real, durante un tiempo ilimitado.

Es también importante recalcar que el desarrollo del proyecto servirá para su uso a nivel universitario como soporte para la academia y podrá ser ampliado para tener más características de monitoreo sea de personas, animales, cosas como robots autónomos.

La relevancia de este nuevo enfoque es eliminar los problemas presentes an la actualidad, dando así acceso libre a aquellas personas que requieran el uso del sistema, y dando las libertades de usar una plataforma abierta en el momento que ellos lo deseen, ejecutando el sistema de una manera fácil de acceder y usar.

Abstract

This degree work aims to contribute students and professionals dedicated to the research and implementation of applications related to satellite tracking or monitoring, providing free service for the use of the platform, in real time, for an unlimited time.

It is also important to emphasize that the development of the project will be of use to the university as a support for the academy which be expanded to have more monitoring features of people, animals, and more things like autonomous robots.

The relevance of this new approach is to eliminate present-day problems, by giving free access to those who require the use of the system, and providing freedom to use an open platform at any time, by executing a system easy to access and use.

Índice general

Introducción **1**
Problema . . . 1
Objetivos . . . 2
Objetivo General . . . 2
Objetivos Específicos . . . 2
Alcance . . . 2
Justificación . . . 3

1. Revisión literaria **4**
1.1. Sistema de localización . . . 4
1.1.1. Tipos de sistemas de localización . . . 4
1.2. Dispositivos de monitoreo y seguimiento satelital . . . 5
1.2.1. OBDII GPS Tracker . . . 5
1.2.2. GPS Tracker 303H . . . 5
1.3. Software de desarrollo Open source . . . 5
1.3.1. Python . . . 6
1.3.2. Firebase . . . 6
1.4. Base de datos . . . 6
1.4.1. Bases de datos relacionales . . . 6
1.4.2. Bases de datos no relacionales . . . 7
1.5. Desarrollo web . . . 8
1.6. Sockets . . . 9
1.6.1. Tipos de socket . . . 10
1.6.2. El modelo cliente/servidor . . . 11
1.7. Web sockets . . . 15
1.7.1. Crear un websocket . . . 15
1.7.2. Websockets en el servidor . . . 16
1.7.3. Comunicación de origen cruzado . . . 16
1.7.4. Servidores proxy . . . 16
1.7.5. Visión general del diseño de un sistema basado en websocket . . . 16
1.8. HTTP . . . 17

1.8.1. La comunicación entre el navegador y el servidor mediante el protocolo HTTP . . . 17
1.8.2. Solicitud HTTP . . . 17
1.8.3. Respuesta HTTP . . . 18
1.9. Análisis de software de plataformas exixtentes a nivel local y nacional . . . 18
1.9.1. Introducción . . . 18
1.10. Plataformas de seguimiento de posición global a nivel local y nacional . . . 19
1.10.1. Sistemas de monitoreo y seguimiento satelital en Ecuador . . . 19
1.11. ¿Por qué comparar estas plataformas? . . . 19
1.12. Semejanzas y diferencias . . . 20
1.13. Propuesta del sistema a desarrollar . . . 20

2. Diseño **22**
2.1. Descripción del Sistema . . . 22
2.2. Diagrama de Bloques . . . 22
2.2.1. Requerimientos del Sistema . . . 22

3. Implementación y pruebas **25**
3.1. Montaje y Funcionamiento . . . 25
3.1.1. Configuración de la aplicación . . . 25
3.1.2. Indicaciones de funcionamiento . . . 29
3.2. Pruebas . . . 29
3.2.1. Verificación de funcionamiento . . . 29

Conclusiones y trabajo futuro **32**
Conclusiones . . . 32
Recomendaciones . . . 32
Trabajo futuro . . . 33

APÉNDICE **34**

A. Código **34**

Índice de figuras

1.1. Modelo MVC 9
1.2. Bloques modelo MVC 9
1.3. Seciencia de llamadas hecha por el servidor [19] 13
1.4. Secuencia de llamadas hecha por un cliente [19] 14
1.5. Etapas de comunicacion entre navegador y servidor en el protocolo HTTP . . . 17
1.6. Comparación de características de plataformas analizadas 20

2.1. Diagrama de bloques general del sistema 23

3.1. Creación e inicialización de proyecto en Firebase 26
3.2. Conexión del aplicativo con la plataforma de desarrollo Firebase 27
3.3. Acceso a la información almacenada en la base de datos 30
3.4. Prueba de datos de ubicación en Google Maps 31
3.5. Configuración de suguridad y parámetros de base de datos 31

Introducción

Este trabajo de grado ha sido realizado con el *Grupo de Investigación en Sistemas Inteligentes de la Universidad Técnica del Norte (GISI-UTN)*.

Con el desarrollo del presente proyecto se busca aportar significativamente al campo de la investigación académica en temas acordes al seguimineto de posición global, con la ventaja que nos ofrece el software libre al ser un proyecto que no necesita licencias y servirá como base para futuras investigaciones.

Problema

Hablar de software libre es hablar de herramientas que brindan ciertas libertades a los usuarios. El software libre es un movimiento global iniciado en 1983 por Richard Stallman, un hacker estadounidense que en aquel entonces trabajaba en el departamento de Inteligencia Artificial del Instituto Tecnológico de Massachusetts (MIT) [1]. La trilogía entre educación, creatividad y software libre no sólo da cuenta de las oportunidades que esto puede traer al contexto formativo de la academia actual, sino que además contribuye a evidenciar todas esas transformaciones que hasta ahora no se han puesto en marcha [2].

Por otro lado, el sistema de posicionamiento global mediante satélites (GPS: Global Positioning System) supone uno de los más importantes avances tecnológicos de las últimas décadas. Diseñado inicialmente como herramienta militar para la estimación precisa de posición, velocidad y tiempo, se ha utilizado también en múltiples aplicaciones las cuales se han proliferado a un ritmo exponencial [3].

A nivel local y nacional existen varias alternativas comerciales que brindan el servicio de rastreo y monitoreo a través de plataformas virtuales, las cuales tienen limitaciones y se rigen a ofrecer un servicio según la capacidad de pago del usuario. Algunas de las opciones son:

- Monitoreo de ubicación y rutas bajo suscripción y según plan contratado [4].
- Varios servicios de rastreo con un pago por contrato y suscripción mensual [5].
- Servicios de rastreo bajo suscripción y plan contratado [6].

El principal problema que se encuentra en estas plataformas es que no son abiertas ni gratuitas. Ofrecen sus servicios únicamente después del pago de una suscripción, ya sea mensual o

anual, y manejan costos que no son accesibles para el común de la sociedad. La única manera de obtener soporte es recurrir a los propios medios de las empresas.

En la literatura existen varias soluciones al problema planteado en: [7] [8] [9] , sin embargo, ninguna se encuentra disponible en la web y su código tampoco ha sido liberado. Se limitan a ser sólo trabajos finales de ingeniería.

De aquí surge la necesidad de aportar con una plataforma libre que sirva de soporte para nuevas investigaciones que surjan en el ámbito académico y que no requiera de licencias ni permisos. De esta forma podría constituirse en una herramienta base para el desarrollo y prueba de nuevos algoritmos y para la solución de problemas relacionados con el seguimiento de posición global.

Objetivos

Objetivo General

Desarrollar una plataforma abierta para la gestión de datos de posicionamiento global.

Objetivos Específicos

- Analizar el software de las plataformas de seguimiento de posición global que existen a nivel local y nacional para proponer una mejora respecto a éstas.
- Probar diferentes dispositivos localizadores existentes en el medio para determinar su funcionalidad y configuración para su acople con la aplicación.
- Desarrollar una aplicación que se ajuste a la funcionalidad de las plataformas y de los dispositivos estudiados.
- Realizar pruebas de funcionamiento de la aplicación.

Alcance

El presente proyecto plantea el desarrollo del software de gestión de datos de un sistema abierto para seguimiento de posición global. Se adquirirá hardware de posicionamiento global y se lo configurará para que se conecte con su propia plataforma y así inferir su funcionamiento.

En cuanto a la aplicación, se ubicará en un servidor virtual mediante el servicio de Web hosting. Ésta aplicación se realizará en software libre y básicamente manejará la comunicación con los dispositivos localizadores y la gestión de datos (almacenamiento, lectura y borrado). La gestión de datos permitirá al usuario manejar la información a través de un computador que tenga acceso a internet. Al ser desarrollada como un sistema abierto brindará las libertades para

ser reconfigurada según la necesidad de la investigación. Es importante aclarar que no se trabajará a fondo en la interfaz, ya que el objetivo del proyecto se limita a gestionar la información (guardar datos desde los dispositivos, acceder desde un computador).

Justificación

El presente trabajo busca aportar a aquellos estudiantes y profesionales que se dedican a la investigación e implementación de aplicaciones que tienen que ver con rastreo o monitoreo satelital, prestando servicio de manera gratuita para el uso de la plataforma, en tiempo real, durante un tiempo ilimitado.

Es indudable la necesidad de incluir nuevas prácticas y modelos en la educación que sean más pertinentes con la sociedad de la información.

Es también importante recalcar que el desarrollo del proyecto servirá para su uso a nivel universitario como soporte para la academia y podrá ser ampliado para tener más características de monitoreo sea de personas, animales, cosas como robots autónomos.

La relevancia de este nuevo enfoque es eliminar los problemas antes mencionados, dando así acceso libre a aquellas personas que requieran el uso del sistema, y dando las libertades de usar una plataforma abierta en el momento que ellos lo deseen, ejecutando el sistema de una manera fácil de acceder y usar.

A su vez otorga al usuario el control total del sistema, sin depender de pagos mensuales o renovaciones de licencias para la utilización del mismo, a excepción de los pagos mensuales de internet o de plan de datos de un teléfono móvil, que no entran dentro del costo del sistema.

El Ingeniero Mecatrónico es un profesional competente, crítico, humanista, líder y emprendedor, cuenta con una formación sólida en las diversas áreas del conocimiento lo que le permite involucrarse eficazmente y con responsabilidad social en actividades de investigación, diseño e innovación de productos mecatrónicos, manejo de equipo técnico relacionado con su profesión, transferencia y/o adaptación de tecnología, con cuidado del medio ambiente, por lo que el presente trabajo está dentro del campo de aplicación de la carrera y valida su sustentación.

Capítulo 1

Revisión literaria

1.1. Sistema de localización

Es un sistema creado para tener la capacidad de ofrecer servicios de monitoreo ya sea a personas o vehículos, siempre y cuando tengan incorporado un GPS a bordo y con conexión a una red de telefonía o de transferencia de datos (internet), por medio de un enlace inalámbrico para poder enviar y recibir información desde y hacia un servidor de datos.[8]

1.1.1. Tipos de sistemas de localización

Según [8], existen dos tipos de sistemas de monitoreo o localización, los cuales se describen a continuación:

Localización mediante GPS

El objeto que se desea localizar, deberá contar con un receptor GPS incorporado, el modem inalámbrico será el encargado de transmitir la información de la ubicación, a un servidor cada cierto tiempo, donde será almacenada la información y posteriormente darle tratamiento según sea requerido por el administrador.

Localización mediante GPS-GSM GPRS

El sistema está conformado por un módulo de micro-control GPS-GSM/GPRS que es instalado de forma estratégica en el objeto rastrear para no poder ser encontrado con facilidad, este módulo es a su vez la fusión de dos tecnologías que están en auge, una unidad de localización que hace uso de la red de satélites GPS y el dispositivo GSM/GPRS que permite enviar y recibir datos a través de la red de telefonía celular; el módulo se encarga de recibir y decodificar las señales provenientes de la constelación de Satélites GPS, proporcionando así datos GPS de posicionamiento, para posteriormente mediante el mismo módulo enviarlos a un Módulo Servidor

de Monitoreo mediante la tecnología GSM o GPRS según sea necesario a través de la red de telefonía celular, donde será almacenada en una base de datos.

1.2. Dispositivos de monitoreo y seguimiento satelital

Para el desarrollo del presente proyecto se usó dos tipos de dispositivos, lo que ayudó a la comprensión de su funcionamiento, caracteristicas de comunicación, entre otros, para así tener clara la lista de requerimientos del sistema que se realizó.

1.2.1. OBDII GPS Tracker

- Fácil de instalar Plug and Play
- Alerta de geofence
- alerta de velocidad
- Alerta de ralentí del motor
- Frenado fuerte / aceleración / alerta de giro
- Datos de diagnóstico del motor
- Informes
- Historial en línea de 30 días
- Garantía de por vida

1.2.2. GPS Tracker 303H

Este perseguidor es un nuevo producto basado en el sistema por satélite de la red del G/-M/GPRS y de colocación de GPS, que fijó funciones múltiples de la seguridad, de la colocación, de supervisar vigilancia, las alarmas de la emergencia y del seguimiento en su totalidad. Puede seguir y supervisar su blanco remoto por SMS o por internet.

1.3. Software de desarrollo Open source

En este apartado se procede a realizar una descripción de las diferentes tecnologías de desarrollo de software que pueden ser utilizadas en el proyecto, en donde se hace una mención especial al sistema de desarrollo para aplicaciones web o móviles de Google la cual facilita el desarrollo de apps de manera rápida y proporciona una base de datos adaptable al sistema a desarrollar.

1.3.1. Python

Python es un lenguaje de programación interpretado e interactivo, capaz de ejecutarse en una gran cantidad de plataformas. Se desarrolla como un proyecto de código abierto, administrado por PYTHON Software Foundation. Es un lenguaje de programación de código abierto que permite la ejecución en diversas plataformas, los usuarios que utilizan este lenguaje lo consideran el más elegante y a su vez amigable para la programación web, el principal objetivo de este lenguaje es buscar la factibilidad tanto para la lectura como el diseño, al ser un lenguaje multi paradigma brinda innumerables beneficios al permitir al usuario trabajar bajo varios estilos: programación orientada a objetos, programación funcional, entre otros. Otro aspecto importante a considerar es que permite la facilidad de extensión esto quiere decir que se puede escribir nuevos módulos de manera fácil en bajo lenguaje como C o C++ y se puede incluir para aplicaciones que necesiten una interfaz programable.[9]

1.3.2. Firebase

Firebase es la plataforma de desarrollo proporcionada por Google la cual está orientada a facilitar la creación de aplicaciones de manera eficiente y rápida, Firebase proporciona una base de datos en la nube que es adaptable a las necesidades del usuario, utilizala infraestructura de software de Google por lo que garantiza ser estable, esta disponible para diferentes plataformas de desarrollo como android, ios o web y se acopla a diferentes leguajes de programación.

1.4. Base de datos

Una base de datos es un almacenamiento de información organizado, estructurado y categorizado, que comparten entre sí un vínculo o relación en común al momento de acceder a ella. Las bases permiten integrar los datos y tener coherencia en los mismos, proporcionando un corto tiempo de respuesta para su acceso [10]. Dependiendo de las circunstancias, se puede usar un formato de base de datos compleja o un archivo de texto sencillo [11].

Los sistemas gestores de bases de datos DBMS (Database Management Systems), son software que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Cada gestor dispone de distintas interfaces con sus propias características [12].

Para relacionar la información los gestores de bases de datos se diferencian en dos: relacionales y no relacionales

1.4.1. Bases de datos relacionales

Una base de datos relacional consta de tablas que poseen información que las vincula, relacionándolas unas con otras. Esto permite almacenar datos eficientemente y sin redundancias. Poseen una estructura bidimensional donde constan varias filas y columnas llamadas registros y campos respectivamente [11].

En 1970 Edgar Frank Codd realiza la postulación de sus fundamentos, los que no tardaron en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de relaciones. En este tipo de modelo, la forma para almacenar los datos no tiene relevancia. Los datos pueden ser recuperados o almacenados mediante consultas, ofreciendo flexibilidad para administrar la información [12].

El lenguaje más utilizado para construir bases de datos relacionales es SQL (Structured Query Language o Lenguaje Estructurado de Consultas), declarado estándar internacional de comunicación dentro de las bases de datos [12]. Los gestores de bases de datos relacionales más destacados son:

- MySQL: Es una base de datos Open Source desarrollada por Oracle. Es un amplio subconjunto del lenguaje SQL, permite seleccionar distintos tipos de almacenamiento, replicación de los datos, búsqueda e indexación de los campos [12]. Poco a poco fue dotada de componentes esenciales del algebra relacional. En la actualidad, es una verdadera base de datos relacional que ofrece un buen rendimiento [10].

- PostgreSQL: Es un evolucionado sistema de administración de bases de datos objetorelacionales de código libre, entre sus características se encuentran: está basado en lenguaje C, posee interoperabilidad con SQL, de almacenamiento confiable y muy robusto, de manipulación potente, flexible y eficiente, además ofrece muchas funcionalidades para responder a necesidades muy avanzadas de manera eficaz [13].

- SQLite: Es un gestor de base de datos relacional escrito en C de código abierto, se diferencia de MySQL y PostgreSQL, ya que no funciona según el modelo cliente-servidor, sino que está embebida. Tiene una rápida configuración y sólo se necesita incluir una librería en la aplicación. Es ideal para peque˜nas aplicaciones, pero el rendimiento puede disminuir si el número de datos es excesivamente elevado [10].

1.4.2. Bases de datos no relacionales

Este tipo de bases de datos se diferencian de las bases de datos relacionales porque no usan SQL como lenguaje principal para consultas, llamándolas NoSQL. Las bases de datos NoSQL almacenan la información sin cumplir el esquema entidad–relación. Se las usa principalmente donde las bases de datos relacionales tienen problemas de escalabilidad y rendimiento, principalmente donde existen miles de usuarios concurrentes con millones de consultas [14].

NoSQL hace referencia a las bases de datos, en las que priman el manejo de grandes conjuntos de datos y la velocidad. Estas bases de datos permiten tener escalabilidad (ampliar rápidamente) y desligar el hardware del tipo de datos haciéndolo eficiente. Las bases de datos NoSQL son utilizadas cuando el volumen de datos es demasiado grande, mientras que SQL se utiliza para realizar análisis más detallados [12].

Se clasifican según su forma de almacenar los datos pues no lo hacen en forma de tabla, usan otros formatos como clave-valor, BigTable, bases de datos documentales y orientadas agrafos [14]. Entre las bases de datos no relacionales destacan las siguientes:

- Redis: Es una base de datos clave-valor, escrita en C, que provee un excelente rendimiento como base de datos en memoria. Útil para almacenar datos que no poseen relaciones complejas entre sí, pero que están indexados por una clave. Sus valores pueden ser del tipo: cadena de caracteres, listas, diccionarios y conjuntos [10]. Su uso es recomendable cuando los datos cambian rápidamente y tienen un tama˜no predecible, como el análisis en tiempo real [12].

- MongoDB: Es una base de datos orientada a gestionar y almacenar documentos, escrita en C++ de código abierto, se trata de un tipo de base clave-valor donde el valor es un documento. Así, la eficacia de esta tecnología radica en dar una buena estructura al valor y tener un uso correcto de claves [10]. Su uso es útil si se hacen consultas dinámicas [12].

- Cassandra: Apache Cassandra es Open Source implementado en Java. Desarrollado por Facebook que en 2010 fue donada a Apache para proyectos de primer nivel como software libre. Es una base de datos NoSQL importante y muy utilizada a nivel mundial, haciendo su uso, tecnologías relevantes como Netflix, eBay, Spotify, Twitter, Urban Airship, Constant Contact, entre otros. Cassandra es capaz de trabajar con varios terabytes de datos. La información es almacenada en columnas en modelo clave-valor [15].

1.5. Desarrollo web

El concepto framework se emplea un muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírsenos.

En general, con el término framework, nos estamos refiriendo a una estructurasoftware compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

Para comprender como trabajan los frameworks Web existentes es imprescindible conocer el patrón MVC.[16]

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada

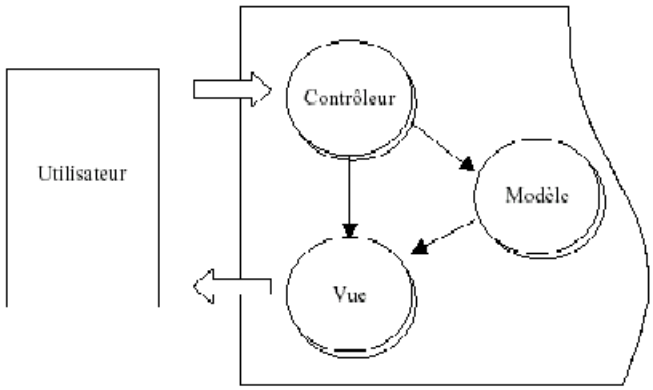


Figura 1.1: Modelo MVC

y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

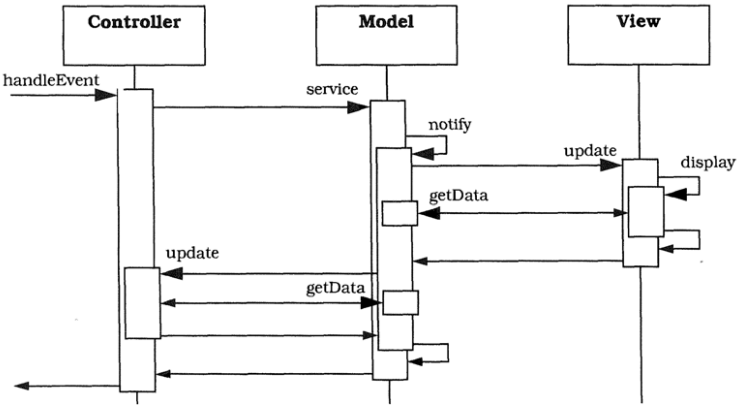


Figura 1.2: Bloques modelo MVC

1.6. Sockets

Aparecieron a principios de los 80 con el sistema UNIX de Berkeley, para proporcionar un medio de comunicación a los procesos, con el fin de proporcionar un medio de comunicación

entre ellos. Los sockets, si hacemos un símil, tienen la misma función que la que pudiera tener la comunicación por correo o por teléfono (de un buzón se extraen mensajes completos, mientras que el teléfono permite el envío de flujos de información que no tienen una estructura claramente definida; esta dualidad se encuentra en los sockets). El socket, por tanto, ofrece dos puntos de contacto entre distintas aplicaciones, a través de los cuales estas se comunican.[19]

Los sockets no son más que puntos o mecanismos de comunicación entre procesos que permiten que un proceso hable (emita o reciba información) con otro proceso incluso estando estos procesos en distintas máquinas. Esta característica de interconectividad entre máquinas hace que el concepto de socket nos sirva de gran utilidad.[17]

La comunicación entre procesos a través de sockets se basa en la filosofía CLIENTE-SERVIDOR: un proceso en esta comunicación actuará de proceso servidor creando un socket cuyo nombre conocerá el proceso cliente, el cual podrá "hablarçon el proceso servidor a través de la conexión con dicho socket nombrado.

El proceso crea un socket sin nombre cuyo valor de vuelta es un descriptor sobre el que se leerá o escribirá, permitiéndose una comunicación bidireccional, característica propia de los sockets y que los diferencia de los pipes, o canales de comunicación unidireccional entre procesos de una misma máquina. El mecanismo de comunicación vía sockets tiene los siguientes pasos: [17]

- 1º) El proceso servidor crea un socket con nombre y espera la conexión.
- 2º) El proceso cliente crea un socket sin nombre.
- 3º) El proceso cliente realiza una petición de conexión al socket servidor.
- 4º) El cliente realiza la conexión a través de su socket mientras el proceso servidor mantiene el socket servidor original con nombre.

1.6.1. Tipos de socket

Cada tipo de socket va a definir una serie de propiedades en función de las comunicaciones en las cuales está implicado:

- a) La fiabilidad de la transmisión. Ningún dato transmitido se pierde.
- b) La conservación del orden de los datos. Los datos llegan en el orden en el que han sido emitidos.
- c) La no duplicación de datos. Sólo llega a destino un ejemplar de cada dato emitido.
- d) La comunicación en modo conectado. Se establece una conexión entre dos puntos antes del principio de la comunicación (es decir, se establece un circuito virtual). A partir de entonces, una emisión desde un extremo está implícitamente destinada al otro extremo conectado.

- e) La conservación de los límites de los mensajes. Los límites de los mensajes emitidos se pueden encontrar en el destino.
- f) El envío de mensajes (urgentes). Corresponde a la posibilidad de enviar datos fuera del flujo normal, y por consecuencia accesibles inmediatamente (datos fuera de flujo).

Cabe reseñar que un cauce de comunicación normal tiene las cuatro primeras propiedades, pero no las dos últimas.[18]

A continuación se describe los tipos de socket disponible:

- SOCK STREAM: Los sockets de este tipo permiten comunicaciones fiables en modo conectado (propiedades a, b, c y d) y eventualmente autorizan, según el protocolo aplicado los mensajes fuera de flujo (propiedad f). El protocolo subyacente en el dominio Internet es TCP. Se establece un circuito virtual realizando una búsqueda de enlaces libres que unan los dos ordenadores a conectar (parecido a lo que hace la red telefónica conmutada para establecer una conexión entre dos teléfonos). Una vez establecida la conexión, se puede proceder al envío secuencial de los datos, ya que la conexión es permanente. Son streams de bytes full-dúplex (similar a pipes). Un socket stream debe estar en estado conectado antes de que se envíe o reciba en él.
- SOCK DGRAM: Corresponde a los sockets destinados a la comunicación en modo no conectado para el envío de datagramas de tamaño limitado. Las comunicaciones correspondientes tienen la propiedad e. En el dominio Internet, el protocolo subyacente es el UDP. Los datagramas no trabajan con conexiones permanentes. La transmisión por los datagramas es a nivel de paquetes, donde cada paquete puede seguir una ruta distinta, no garantizándose una recepción secuencial de la información.
- SOCK RAW: Permite el acceso a los protocolos de más bajo nivel (por ejemplo, el protocolo IP en el dominio Internet). Su uso está reservado al superusuario.
- SOCK SEQPACKET: Corresponde a las comunicaciones que poseen las propiedades a, b, c, d y e. Estas comunicaciones se encuentran en el dominio XNS.

Cabe recalcar que los tiposde socket mas usados son los dos primeros de esta lista.[18]

1.6.2. El modelo cliente/servidor

El modelo cliente/servidor, es el modelo de ejecución que siguen todas las aplicaciones de red. Un servidor es un proceso que se está ejecutando en un nodo de la red, y su función es gestionar el acceso a un determinado recurso. Un cliente es un proceso que se ejecuta en el mismo nodo, o en uno diferente, y que realiza peticiones al servidor. Las peticiones están originadas por la necesidad de acceder al recurso que gestiona el servidor.

La comunicación entre cliente y servidor, puede ser o bien orientada a la conexión, o bien sin conexión. En el caso de que la comunicación sea orientada a la conexión, esta se lleva a cabo

mediante el establecimiento de circuitos virtuales entre el cliente y el servidor. En este caso, el intercambio de información se realiza con una alta fiabilidad fluyendo la información a través del circuito virtual de una forma secuencial, es decir, tiene un flujo continuo. Esto no ocurre en el caso de que la comunicación sea sin conexión, puesto que aquí el intercambio de información se efectúa mediante el envió de datagramas. La fiabilidad es menor, y al contrario que en el caso anterior, los datagramas no siguen un flujo continuo.

La comunicación sin conexión presenta un aspecto simétrico en la medida de que el iniciador del diálogo puede ser cualquiera de los dos que intervienen. Esto no ocurre en el caso de la comunicación orientada a la conexión, ya que uno de los dos procesos (en posición de cliente) pregunta al otro (en posición de servidor) si acepta esta comunicación.

Dado que la comunicación que normalmente se utiliza es orientada a la conexión, describiremos a continuación cómo sería el comportamiento tanto del cliente como del servidor con este tipo de servicio, mostrando la secuencia de llamadas de cliente y servidor para un servicio sin conexión al final. [19]

El servidor

El servidor está continuamente esperando peticiones de servicio. Cuando se produce una petición, el servidor despierta y atiende al cliente. Cuando el servicio concluye, el servidor vuelve al estado de espera. De acuerdo con la forma de prestar el servicio, podemos considerar dos tipos de servidores:

- Servidores interactivos: El servidor no sólo recoge la petición de servicio, sino que él mismo se encarga de atenderla. Esta forma de trabajo presenta un inconveniente; si el servidor es lento en atender a los clientes y hay una demanda de servicio muy elevada, se van a originar unos tiempos de espera muy grandes.

- Servidores concurrentes. El servidor recoge cada una de las peticiones de servicio y crea otros procesos para que se encarguen de atenderlas. Este tipo de servidores sólo es aplicable en sistemas multiproceso, como UNIX. La ventaja que tiene este tipo de servicio es que el servidor puede recoger peticiones a muy alta velocidad, porque está descargado de la tarea de atención al cliente. En las aplicaciones donde los tiempos de servicio son variables, es recomendable implementar este tipo de servidores.

Su papel es pasivo en el establecimiento de la comunicación, ya que después de haber avisado al sistema al que pertenece de que está preparado para responder a las peticiones de servicio, el servidor se pone a la espera de peticiones de conexión que provengan de clientes. Para esto dispone de un socket de escucha, enlazado al puerto TCP correspondiente al servicio, sobre el que espera las peticiones de conexión. Cuando llega al sistema una petición de este tipo, se despierta al proceso servidor y se crea un nuevo socket, que se llama socket de servicio, el cual se conecta al cliente. Entonces el servidor podrá, por una parte delegar el trabajo necesario para la realización del servicio a un nuevo proceso (creado por fork) que utilizará entonces la conexión, y por otra parte volverá al socket de escucha.[19]

Después de la aceptación de la comunicación, el servidor tiene dos posibilidades:

- Hacerse cargo de ella, lo que significa eventualmente que otras conexiones pendientes no serán efectivamente aceptadas hasta que el servicio demandado haya sido satisfecho
- O bien subtratar la gestión de la conexión y la realización del servicio mediante un proceso hijo.

La secuencia de primitivas para la utilización de sockets que el servidor tiene que usar, y su orden, se muestra en el diagrama expresado a continuación.

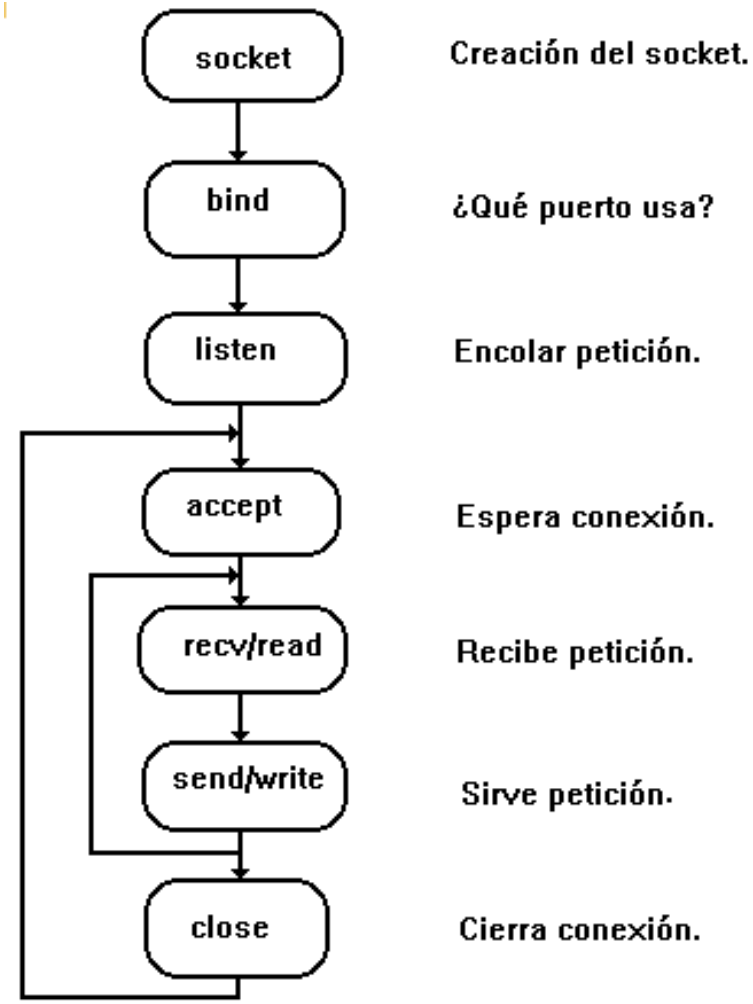


Figura 1.3: Seciencia de llamadas hecha por el servidor [19]

En el caso de que la comunicación sea sin conexión, la secuencia de llamadas varía sensiblemente, puesto que no es necesario encolar las peticiones que le llegan al servidor, ni esperar

a que la conexión se efectúe. El servidor, una vez conectado al puerto correspondiente, se bloquea hasta que recibe alguna petición por parte de un cliente, contestando a este, y retomando la escucha a la espera de nuevas peticiones.

El cliente

El cliente es la entidad activa en el establecimiento de una conexión, puesto que es el que toma la iniciativa de la demanda de conexión a un servidor. Esta demanda se realiza por medio de la primitiva connect, solicitando el establecimiento de una conexión que será conocida por los dos extremos. Además, el cliente está informado del éxito o del fracaso del establecimiento de la conexión.

Para que un proceso cliente inicie una conexión con un servidor a través de un socket, es necesario realizar una llamada a connect. Así, se crea un circuito virtual entre los dos procesos cuyos extremos son los sockets. La secuencia de primitivas para la utilización de sockets que el servidor tiene que usar, se muestra a continuación.

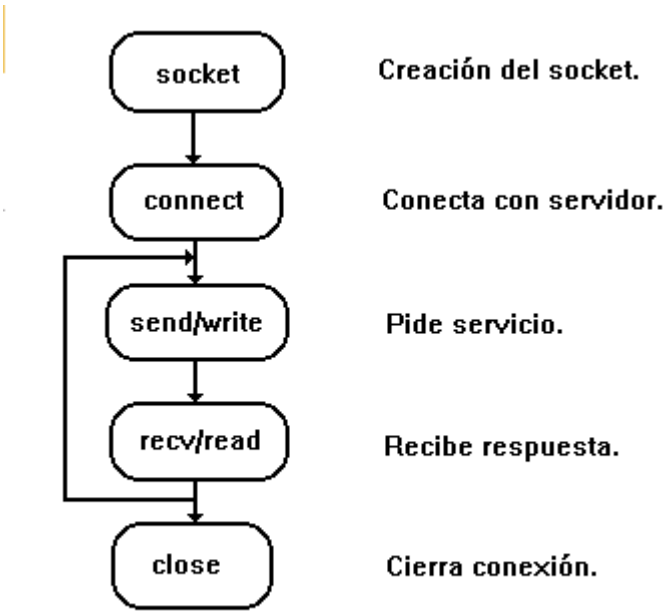


Figura 1.4: Secuencia de llamadas hecha por un cliente [19]

Pero en el caso de una comunicación sin conexión, el cliente no utiliza la llamada connect para establecer un circuito virtual entre él y el servidor, sino que lo único que hace es conectarse a un puerto por el cual envía peticiones de servicio a un servidor.

1.7. Web sockets

Internet se ha creado en gran parte a partir del llamado paradigma solicitud/respuesta de HTTP. Un cliente carga una página web, se cierra la conexión y no ocurre nada hasta que el usuario hace clic en un enlace o envía un formulario.

Hace ya algún tiempo que existen tecnologías que permiten al servidor enviar datos al cliente en el mismo momento que detecta que hay nuevos datos disponibles. Se conocen como ”Push.º Çomet”. Uno de los trucos más comunes para crear la ilusión de una conexión iniciada por el servidor se denomina Long Polling. Con el Long Polling, el cliente abre una conexión HTTP con el servidor, el cual la mantiene abierta hasta que se envíe una respuesta. Cada vez que el servidor tenga datos nuevos, enviará la respuesta. El Long Polling y otras técnicas funcionan bastante bien y de hecho ha sido utilizadas en muchas aplicaciones como el chat de Gmail.

Los WebSockets nos ofrecen una conexión bidireccional entre el servidor y el navegador. Esta conexión se produce en tiempo real y se mantiene permanentemente abierta hasta que se cierre de manera explícita. Esto significa que cuando el servidor quiere enviar datos al servidor, el mensaje se traslada inmediatamente. Efectivamente, esto es lo que sucedía al utilizar tecnologías como Comet, pero se conseguía utilizando una serie de trucos. Si esto no funcionada, siempre era posible utilizar Ajax para conseguir un resultado parecido, pero sobrecargando el servidor de manera innecesaria.

Si disponemos de un socket abierto, el servidor puede enviar datos a todos los clientes conectados a ese socket, sin tener que estar constantemente procesando peticiones de Ajax. La ventaja en cuanto a rendimiento y escalabilidad es bastante evidente al utilizar WebSockets.

La latencia en las comunicaciones es otro de los beneficios de utilizar WebSockets. Como el socket está siempre abierto y escuchando, los datos son enviados inmediatamente desde el servidor al navegador, reduciendo el tiempo al mínimo, en comparación con un paradigma basado en Ajax, donde hay que realizar una petición, procesar la respuesta y enviarla de nuevo de vuelta.

Finalmente, los datos a transmitir se reducen también de manera drástica, pasando de un mínimo de 200-300 bytes en peticiones Ajax, a 10-20 bytes utilizando websockets. [20]

1.7.1. Crear un websocket

El API de WebSocket es realmente sencillo de utilizar. Actualmente, los navegadores únicamente soportan el envío de cadenas de caracteres, y se realiza de una manera muy similar a la que utilizábamos para enviar mensajes en los Web Workers. El API está limitado a métodos para abrir la conexión, enviar y recibir datos y cerrar la conexión. La URL que utilizamos para conectarnos con el WebSocket no tiene por qué pertenecer al mismo dominio que nuestro documento, por lo que podemos conectarnos a servicios de terceros sin problemas, expandiendo las posibilidades de nuestra aplicación.[20]

1.7.2. Websockets en el servidor

Al utilizar los WebSocket, se crea un patrón de uso completamente nuevo para las aplicaciones de servidor. Aunque las pilas de servidor tradicionales como LAMP están diseñadas a partir del ciclo de solicitud-respuesta de HTTP, a menudo dan problemas si hay muchas conexiones WebSocket abiertas. Mantener un gran número de conexiones abiertas de forma simultánea requiere una arquitectura capaz de recibir un alto nivel de concurrencia sin consumir muchos recursos. Estas arquitecturas suelen estar basadas en subprocesos o sistemas de E/S asíncronos.[20]

1.7.3. Comunicación de origen cruzado

Las comunicaciones de origen cruzado son un protocolo moderno creado directamente para WebSocket. Aunque sigue siendo necesario que te asegures de que solo te comunicas con clientes y servidores de confianza, WebSocket permite la comunicación entre las partes de cualquier dominio. El servidor decide si desea poner su servicio a disposición de todos los clientes o solo de los que están alojados en un grupo de dominios bien definidos.[21]

1.7.4. Servidores proxy

Toda nueva tecnología trae consigo una nueva serie de problemas. En el caso de WebSocket, se trata de la compatibilidad con los servidores proxy que median las conexiones HTTP en la mayoría de las redes corporativas. El protocolo WebSocket utiliza el sistema de actualización de HTTP (normalmente utilizado para HTTP/SSL) para .ªctualizarüna conexión HTTP a una conexión WebSocket. A algunos servidores proxy no les gusta esto y cancelan la conexión. Por tanto, aunque un cliente dado utilice el protocolo WebSocket, es posible que no pueda establecer una conexión.[21]

1.7.5. Visión general del diseño de un sistema basado en websocket

La tecnología Websocket proporciona un canal de comunicación bidireccional mediante una única coneción TCP. Está diseñado para implementarse en aplicaciones como los navegadores web. Su API esta siendo estandarizado por el WC3. Las conexiones se establecen a través de un puerto TCP 80 regular, lo que garantiza que el sistema pueda ejecutarse detras de firewalls.

El ciclo de vida de una sesión de WebSocket

Primero, el cliente que admite el protocolo WebSockets solicita una conexión WebSocket. La respuesta del servidor indica el inicio de esta conexión. La conexión permanece abierta durante toda la sesión, hasta cualquier punto solicite su liberación con el procedimiento especificado. Como un WebSocket, permanece activo; Los marcos de WebSocket se pueden migrar desde servidor a cliente y viceversa sin solicitud previa, el servidor WebSockets también aloja la parte de lógica de servicio de nuestras aplicaciones web, que se encarga de mantener una

lista de clientes con WebSockets activos y gestión de sesiones. La aplicación del servidor tiene datos en tiempo real para mapear la situación de una región basado en los datos disponibles. La aplicación del servidor también es responsable de la minería de datos y la ejecución lógica de negocios basada en datos recopilados de varios clientes.

1.8. HTTP

El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML). entre un navegador (el cliente) y un servidor web (denominado, entre otros, httpd en equipos UNIX) localizado mediante una cadena de caracteres denominada dirección URL. [22]

1.8.1. La comunicación entre el navegador y el servidor mediante el protocolo HTTP

Dicha comunicación esta compuesta de dos etapas:

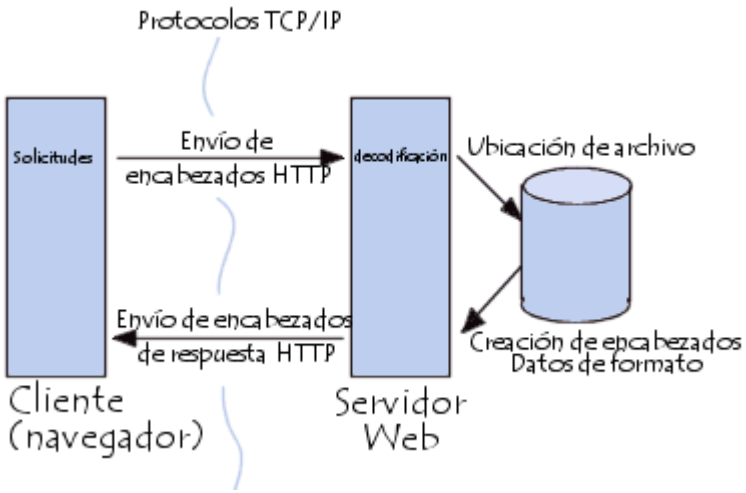


Figura 1.5: Etapas de comunicacion entre navegador y servidor en el protocolo HTTP

El navegador realiza una solicitud HTTP y el servidor procesa la solicitud y después envía una respuesta HTTP. En realidad, la comunicación se realiza en más etapas si se considera el procesamiento de la solicitud en el servidor.

1.8.2. Solicitud HTTP

Una solicitud HTTP es un conjunto de líneas que el navegador envía al servidor. Comprende: [22]

- Una línea de solicitud: una línea que especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada. La línea está formada por tres elementos que deben estar separados por un espacio: el método, la dirección URL y la versión del protocolo utilizada por el cliente (por lo general, HTTP/1.0)

- Los campos del encabezado de solicitud: un conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

- El cuerpo de la solicitud: un conjunto de líneas opcionales que deben estar separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

1.8.3. Respuesta HTTP

Una respuesta HTTP es un conjunto de líneas que el servidor envía al navegador. Está constituida por: [22]

- Una línea de estado: una línea que especifica la versión del protocolo utilizada y el estado de la solicitud en proceso mediante un texto explicativo y un código. La línea está compuesta por tres elementos que deben estar separados por un espacio: La línea está formada por tres elementos que deben estar separados por un espacio: la versión del protocolo utilizada, el código de estado y el significado del código.

- Los campos del encabezado de respuesta: un conjunto de líneas opcionales que permiten aportar información adicional sobre la respuesta y/o el servidor. Cada una de estas líneas está compuesta por un nombre que califica el tipo de encabezado, seguido por dos puntos (:) y por el valor del encabezado Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

- El cuerpo de la respuesta: contiene el documento solicitado.

1.9. Análisis de software de plataformas exixtentes a nivel local y nacional

1.9.1. Introducción

En la siguiente unidad se describe las características, diferencias y funcionalidad de las plataformas analizadas en este trabajo de grado, de igual manera se realizó una comparación entre ellas para de esta manera obtener los requerimientos del sistema a desarrollar, tamando

en cuenta parámetros de valoración tales como costo, facilidad de adqisición, programación, usabilidad, entre otros.

1.10. Plataformas de seguimiento de posición global a nivel local y nacional

1.10.1. Sistemas de monitoreo y seguimiento satelital en Ecuador

Tracklink

Tracklink es una empresa ecuatoriana que ofrece varios servicios ya sea de monitoreo o de administración de dispositivos de seguimiento, con alianzas con empresas del exterior. Actualmente es una empresa multinacional ubicada en varios países de Latinoamérica, ofrece varios tipos de servicios ya sea personales o empresariales con todas las garantías y recursos suficientes para satisfacer las necesidades del mercado al cual está enfocado. La empresa ofrece la opción de monitoreo a través de la aplicación móvil Trackitapp, que permite administrar información en tiempo real, así como otras funciones disponibles sin la necesidad de pagar valores adicionales a él plan contratado de manera inicial, con la facilidad para el cliente de administrar y controlar los dispositivos directamente.

Alerta GPS Ecuador S.A

Empresa ecuatoriana que ofrece el servicio de rastreo, recuperación y administración de varias opciones principalmente enfocado a vehículos. Se necesita adquirir el dispositivo propio de la empresa para poder acceder al servicio mensual el cual requiere de igual manera el pago de una suscripción, ofrece tambien la opción de administración a través de una aplicación móvil para dos tipos de suscripciones : cliente gold y gold server.

Hunter

Es una empresa multinacional ubicada en varios países a nivel mundial, ofrece una gran variedad de servicios de monitoreo ya sea para empresas o a nivel personal, ofrece servicios de monitoreo, inspección y administración de dispositivos e información a través de una plataforma o aplicativo móvil. Maneja varias tarifas que se ajustan a la necesidad de cada cliente.

1.11. ¿Por qué comparar estas plataformas?

Las plataformas que se comparó y analizó en este trabajo, fueron tomadas en cuenta por su facilidad de acceso a información, así como tambien por ser empresas consolidadas a nivel nacional por su garantía y años de experiencia. La funcionalidad y características de cada una se detalla en la figura siguiente.

	PLATAFORMAS DE SEGUIMIENTO DE POSICIÓN GLOBAL ANALIZADAS		
Funcionalidad	Tracklink	Alerta GPS EC	Hunter
Plataforma web	si	si	si
Comunicación GPRS	si	si	si
Comunicación GPS	si	si	si
Tiempo real	no	si	no
Acceso 24/7	no	si	no
Soporte	si	si	si
Garantía	si	si	si
Seguimiento personal	no	no	si
Historial de datos	no	si	no
Costo/Año	$648,48	$192	$616

Figura 1.6: Comparación de características de plataformas analizadas

1.12. Semejanzas y diferencias

las plataformas analizadas anteriormente cuentan con características en común, las cuales cabe recalcar que dependiendo la suscripcion a la que se acceda se tiene como beneficio o no están incluidas. De aquí podemos afirmar que el servicio .Alerta GPS EC", ofrece todos los servicios con un sólo pago emnsula sin tener que activarlos por un costo adicional y tambien conocer que no cuenta con el servicio de seguimiento de personas.

Las plataformas de Hunter y Tracklink ofrecen servicios como, seguimineto en tiempo real, acceso 24/7, acceso a historial, seguimiento de personas, únicamente con un pago adicional y con activación fuera del plan contratado, lo que conlleva a un incremento del valor del servicio para poder contar con todas las funcionalidades.

Las tres plataformas analizadas ofrecen comunicación via GPRS/GPS, de igual manera cuentan con plataforma web para que cada cliente acceda a su información de acuerdo a su plan contratado, los tres servicios ofrecen soporte técnico y garantía en su suscripción, asi como tambien el equipo de seguimeinto esta incluido en el pago sea anual o mensual.

Otra cosa a tomar en cuanta es que cada plataforma es compatible únicamente con el equipo propio y no puede usarse con dispositivos de venta libre.

1.13. Propuesta del sistema a desarrollar

En base a las plataformas analizadas se plantea una solución practica para el desarrollo de la plataforma de seguimineto de posición global, se ha tomado como factores de valoración :

- Costos

- Facilidad de adquisición
- Programación
- Usabilidad
- Versatilidad

Una vez analizado estos indicadores se puede proponer una plataforma abierta para el seguimiento de posición global, que cuente con un aplicativo que permita la gestión de datos en tiempo real, con acceso ilimitado a la información, para la comunicación se usará las funciones del módulo GPS/GPRS, los datos adquiridos generan una base de datos para tener acceso al historial de la información sin restricción, la plataforma es compatible con dispositivos de venta libre y al usar software libre para su desarrollo el código base de este proyecto es de libre acceso para desarrolladores, de ésta forma se tiene una base para el desarrollo de nuevos proyectos orientados a seguimiento de posición global, no unicamente de autos, sino con un campo de aplicación muy extenso, que va desde personas, animales, dispositivos autónomos, etc.

Capítulo 2

Diseño

2.1. Descripción del Sistema

Una vez finalizada la etapa de análisis de las plataformas de gestión de datos de posición global,ésta sección nos da una visión detallada de las características del sistema, nos ayuda a comprender sus funciones activas y pasivas y describe su funcionamiento para una fácil comprensión al lector.

2.2. Diagrama de Bloques

A continuación se presenta el diagrama de bloques general del sistema.

2.2.1. Requerimientos del Sistema

Los requerimientos son las necesidades de los usuarios; definen las funciones que el sistema será capaz de realizar. Los requerimientos o requisitos describen los servicios que ofrece el sistema y las restricciones asociadas a su funcionamiento. Los requerimientos son funcionales y no funcionales [23].

Requerimientos funcionales

Permiten conocer la naturaleza del funcionamiento del sistema. Describen cualquier actividad que el sistema realiza, el comportamiento o función particular cuando se cumplen ciertas condiciones [24].Dicho esto se describen los siguientes requerimientos funcionales para el presente trabajo:

- La aplicación está montada en la plataforma de desarrollo Firebase que usa la infraestructura de servidores de Google, la cual proporciona seguridad y estabilidad al sistema.
- Se requiere una conexión estable y continua a internet para su correcto funcionamiento.

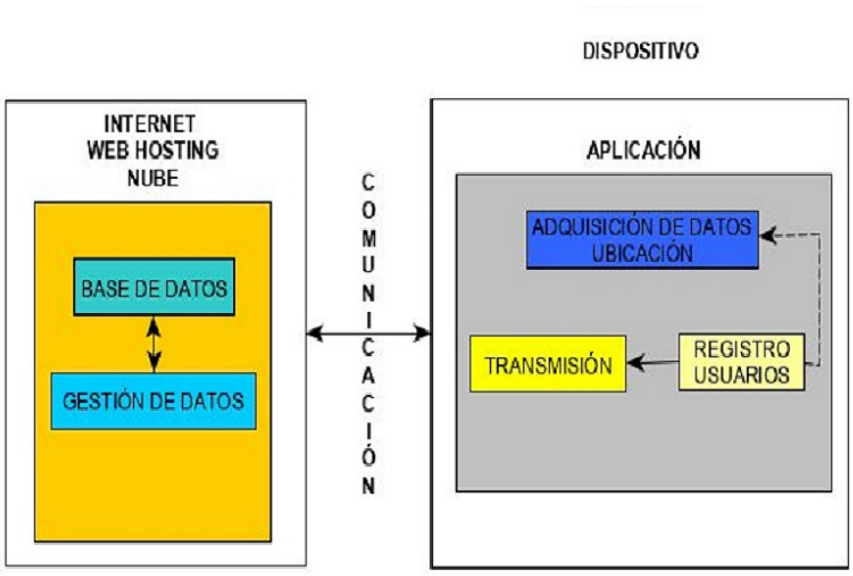


Figura 2.1: Diagrama de bloques general del sistema

- La aplicación almacena los datos obtenidos y los procesa en una base de datos virtual proporcionada por la plataforma de desarrollo firebase.
- Los datos almacenados se pueden gestionar a conveniencia del usuario.
- Con los datos almacenados se puede generar respaldos gracias a que los datos se archivan de manera ordenada.
- La aplicación proporciona un acceso al historial de datos almacenados en la base de datos de firebase.

Requerimientos no funcionales

Son características o cualidades que los usuarios requieren como parte del comportamiento y calidad del software. Las características proporcionan información del sistema. Este tipo de requerimientos no son parte fundamental del sistema, pero sí son necesarios para hacerlo funcionar de la manera deseada [25]. Para seleccionar los atributos que especifican las características útiles sobre la calidad del sistema, se toma en cuenta los criterios que se encuentran en [26], que muestra el estandar para la evaluación de calidad expuesto por la ISO/IEC 9126 y las siete categorias de atributos de calidad definidos por Bass, Clements y Kazman. Así podemos definir los requerimientos no funcionales del ”Sistema abierto para seguimiento de posición global: Software de gestión de datos”, como:

- Eficiencia: La comunicación entre la aplicación y el servidor se lleva a cabo de manera continua e ininterrunpida. la información se almacena automaticamente en la base de datos virtual de la plataforma de desarrollo firebase.
- Disponibilidad: La aplicación puede ser usada las 24 horas, pues ésta no depende de una suscripción y unicamente requiere conexión a internet.
- Seguridad lógica y de datos:
  - la aplicación usa la infraestructura de servidores de Google por lo que cuenta con estabilidad y fiabilidad en la etapa de comunicación y transporte de información.
  - Todos los archivos de la base de datos deben transferirse luego de cada adquisición para mayor seguridad. Los respaldos serán almacenados en una localidad segura ubicada en un servicio de almacenamiento en la nube.
- Usabilidad:
  - El aprendizaje del sistema por el usuario debe ser menor a un tiempo de 15 minutos ya que la aplicación es intuitiva.
  - La aplicación es de sencillo uso.
- Software: El sistema está basado en software libre, debido a que la mayoría de las herramientas están disponibles sin costo. Además, las diferentes aplicaciones pueden ser copiadas y compartidas sin ninguna restricción de uso.
- Hardware: El sistema trabaja continuamente con el dispositivo en donde el aplicativo se ejecute, el mismo que gracias a sus características es el cerebro del sistema.
- Soporte:
  - El software tiene una sencilla instalación, es gratuito y tiene facilidad de mantenimiento.
  - Es claro con la finalidad de permitir en un futuro desarrollar nuevos requerimientos, aislar y corregir los errores y atender las demandas del entorno cambiante.
- Modificable: Proporciona facilidad para hacer cambios en los requerimientos.
- Escalabilidad: El sistema puede tomar a consideración un crecimiento futuro, solo se necesitará una modificación en los campos de la base de datos y de su eficiencia en caso de requerir adaptarse a actualizaciones de cualquier índole.
- Extensibilidad: El sistema toma en cuenta un crecimiento en el futuro, realizando los códigos de programación con un bajo nivel de dificultad y comprensibles para facilitar que el software crezca en funcionalidades y gracias a la plataforma de desarrollo Firebase que proporciona una gran variedad de características adaptables al sistema de manera rapida y fácil.

Capítulo 3

Implementación y pruebas

La siguiente sección detalla el proceso de desarrollo del proyecto, y describe los pasos para que la aplicación cumpla con los requeriminetos descritos en secciones anteriores.

3.1. Montaje y Funcionamiento

En el desarrollo del proyecto se comprendió la importancia de que al desarrollar la aplicación, los datos que se obtengan queden registrados y almacenados de manera que se tenga acceso remoto a dicha información y se pueda acceder a ella no solo desde el dispositivo sino desde cualquier punto de acceso a la base de datos almacenada en la nube que proporciona firebase. Es importante tambien tener claro el enfoque de la aplicación ya que al hablar de seguimineto de posición global se puede mal interpretar de manera que se piense en crear una aplicacion invasiva, por lo contrario el enfoque de este proyecto es contar con una plataforma de gestion de datos de posición global que contribuya al desarrollo de nueva tecnología tanto en el campo académico como en el ambito social y de seguridad.

Dado que la base de datos virtual de firebase nos proporciona acceso en tiempo real de los datos que estén siendo transmitidos y almacenados, se podra acceder a las coordenandas de la ubicación exacta del dispositivo de posicionamineto unicamnete accediendo a la consola de nuestro proyecto en firebase.

3.1.1. Configuración de la aplicación

Crear aplicacion en firebase

La plataforma de desarrollo firebase nos brinda la facilidad de crear una aplicacion de manera rápida y eficiente, una vez que se accede a su cuenta de firebase realizamos los siguientes pasos.

- 1.Agregar proyecto y nombrarlo

- 2.Crear proyecto
- 3.Seleccionar la opción Autenticación
- 4.Para nuestro proyecto activamos la opcion de ingreso vía correo electrónico/ contraseña.
- 5.Nos dirigimos a la seccion Usuarios y agregamos un usuario el cual nos servirá para la autenticación de la aplicacion en firebase.

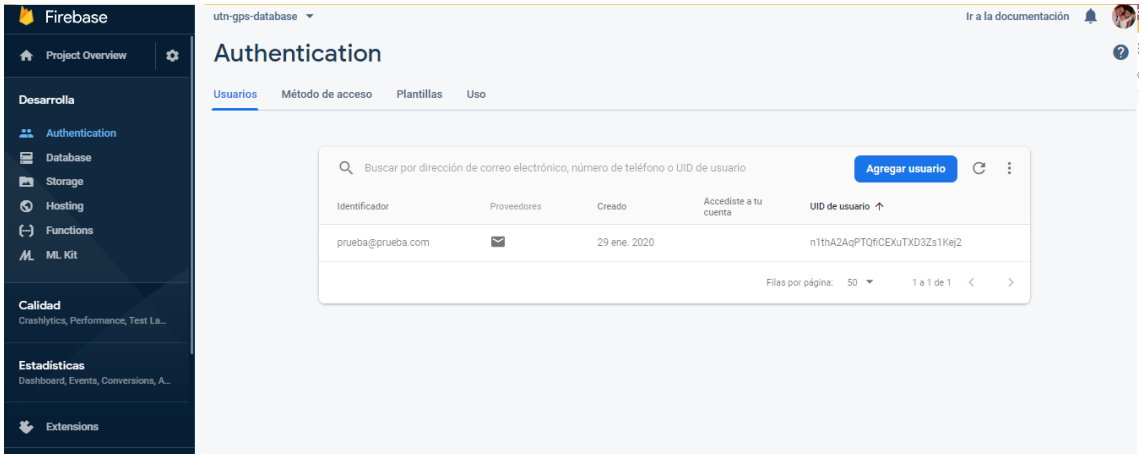


Figura 3.1: Creación e inicialización de proyecto en Firebase

Conectar la aplicación a firebase

Usando el asistente de firebase se conectó la aplicación para que tenga soporte con Firebase Realtime Database y Firebase Authentication.

Éste paso es de suma importancia ya que se configura las dependecias necesarias para nuestro aplicativo. Las dependencias más importantes son:

- com.google.firebase:firebase-auth: Es la encargada de que nuestra aplicación tenga como método de acceso correo electrónico y contraseña para ingresar como usuario.
- com.google.firebase:firebase-database: Es la encargada de que una vez se este ejecutando, la aplicación tenga acceso a la base de datos de firebase.

Al finalizar la configuración tenemos una aplicación vacía creada y configurada con el método de auntenticación y la base de datos de firebase.

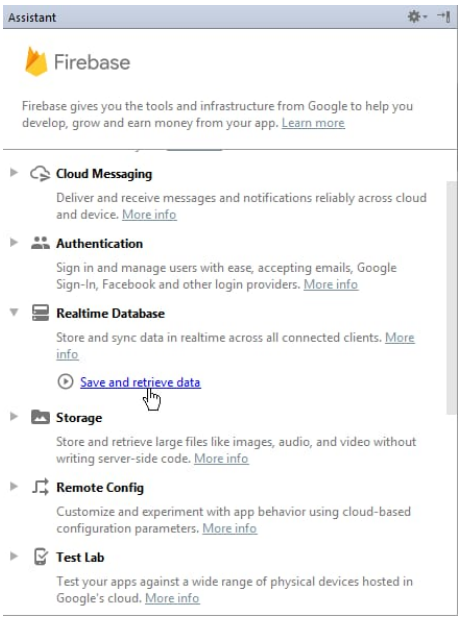


Figura 3.2: Conexión del aplicativo con la plataforma de desarrollo Firebase

Por último se configura la dependencia:

- com.google.android.gms:play-services-location: Es la encargada de solicitar persmisos de uso de los servivios de ubicación de Google.

Configurar el servivio de seguimiento

La aplicación rastreará la ubicación de un dispositivo y la almacenará en la base de datos de firebase.
Toda aplicación de seguimiento debe solicitar los permisos necesarios por lo que en el apartado de programación: .ªndroidManifest.xml", se detalla los comandos:

- a)android.permission.ACCESS.FINE.LOCATION: Una vez aprobado por el usuario, proporciona la capacidad de determinar de la manera más precisa posible la ubicación del dispositivo, incluyendo GPS así como tambien redes WIFI y red móvil.

- b)android.permission.INTERNET: Proporciona a la aplicación permiso para acceder a internet desde el dispositivo.

Al desarrollar una aplicacion de seguimiento de posición es importante que el usuario esté conciente de que se está rastreando su ubicación, por esto es importante crear una notificación pesistente que informe al usuario de la actividad y se cierre en cuanto se acceda a ella. En el

apartado ”strings.xml”: se detalla la información que se verá reflejada en la notificación persistente, que podría ser: nombre de usuario,ID, entre otros.

Inicio de rastreo

Una vez lista la configuración de permisos e inicialización del rastreo, tenemos un aplicativo que requeire que se otorgue permisos para acceder a la ubicación y que nos mostrará de manera activa que se esta siendo rastreado, lo que proporciona al usuario confianza al no ser invasiva en privacidad y con la opción de cerrarla cuando así se requiera de manera inmediata. Dicho esto nos dirigimos a la seccion principal de nustra programación donde podemos realizar la inicializacion de la aplicación de manera correcta.

La cual sigue la siguiente secuencia de programación:

- Inicialización del aplicativo.
- Comprueba si el rastreo GPS está habilitado.
- Comprueba si la aplicación cuenta con los permisos.
- Si la aplicación tiene permiso, inicia el rastreo
- Si la aplicacion no tiene permiso, solicita acceso.
- Si se autoriza el acceso se inicia el rastreo.
- Se tiene acceso a la ubicacioón y el rastreo está habilitado.
- El ususario está notificado en todo momento de que se encuantra activo.
- Se puede salir unicamente pulsando la notificación persistente.

Misma que está detallada en el apéndice de estre trabajo.

Registro de seguimiento

Ahora es necesario crear y configurar el servicio encargado de enviar los datos de ubicación del dispositivo a la plataforma de firebase.

Los comandos principales de ésta seccion son:

- logintoFirebase(): Al ejecutar esta sección se autentifica la aplicación en firebase con el usuario y contraseña antes creados, de ésta forma una vez que se ingrese la base de datos está lista para almacenar la información.
- requestLocationUpdates(): Una vez iniciada sesión, se crea la solicitud para iniciar el rastreo, si la sesión se inicia con éxito la transmisión de datos comienza. Si la sesión no se inicia con éxito se registra el error.

- FusedLocationProvider: Crea la solicitud para recibir actualizaciones, y al recibirlas almacenarlas en la base de datos de firebase.

De ésta manera cuando se ejecuta la aplicación, podemos acceder a la base de datos de firebase para verificar en tiempo real las actualizaciones de ubicación.

3.1.2. Indicaciones de funcionamiento

El uso de la aplicación es altamente intuitivo y de fácil comprensión, lo que se requeire es un dispositivo compatible con la aplicación ya que el enfoque de este trabajo no se adentra en la interfaz ni visualización de datos en un mapa, sino unicamnete en la gestion de la base de datos, los requerimientos de uso de la aplicaciós son basicos.

Instalación

Una vez se cuante con el archivo ejecutable, basta con realizar la intalación aceptando las solicitudes de permisos que ésta requiera y permitiendo el acceso dela aplicación.

Ejecución y uso

Para hacer uso del aplicativo hay que ejecutarlo, una vez se ingresa en la aplicación esta requiere el inicio de sesión , para lo cual se deberá contar con un usuario y contraseña registrados previamente en firebase, para así poder realizar la autenticación y la aplicación funcione adecuadamente.
Como se explicó anteriormente la aplicación requiere unicamente el inicio de sesión y en cuanto éste se realice con éxito, la aplicación comienza a transmitir datos de manera automática y sin interrupciones a menos de que así se requiera.
Para la visualización de datos, basta con ingresar a la consola de control de nuestro proyecto en firebase, y dirigirnos a la sección Database. En donde en tiempo real se registran los datos del dispositivo y se muestran almacenados.

3.2. Pruebas

En el desarrollo de aplicaciones de seguimiento de posición global, puede resultar útil usa datos falsos de ubicación, para de esta forma tener variación en los datos almacenados.

3.2.1. Verificación de funcionamiento

Para comprobar si la aplicación esta grabando datos de posición en la base de datos de firebase:

- Ejecutar el aplicativo

- Otorgar a la aplicación acceso a la ubocación
- Verificar que se haya creado una notificación persistente
- En el navegador web se abre la consola de control de nustro proyecto vinculado a firebase
- Ingresar en ”Base de datos”
- Seleccionar la ventana ”datos”para visualizar la información que se ha transmitido desde el aplicativo.

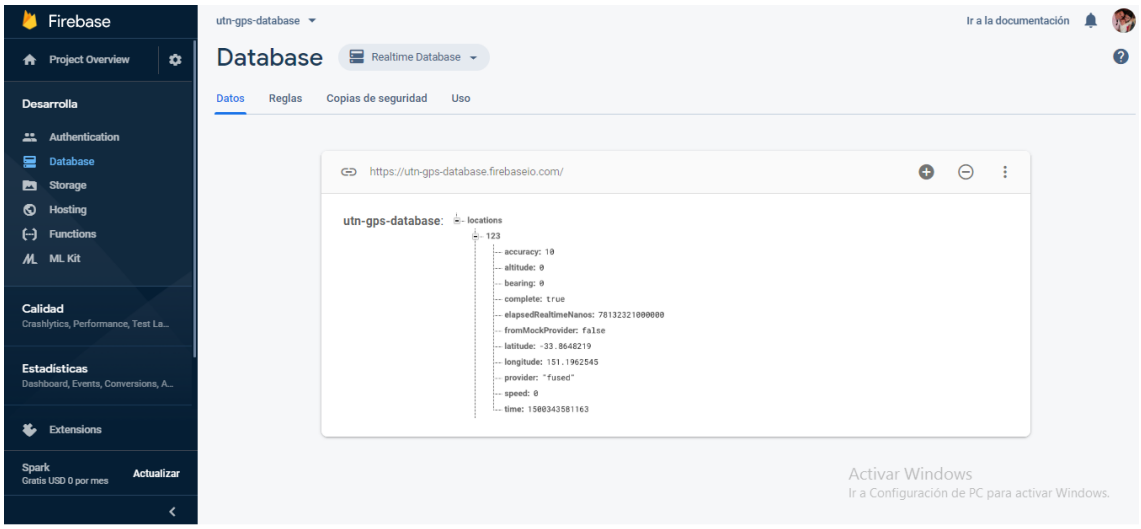


Figura 3.3: Acceso a la información almacenada en la base de datos

Es de suma importancia aclarar que fire nos presenta datos de tipo JSON, que son conjuntos de pares de atributos-valores. Estos pares pueden ser números, cadenas, booleanos verdadero / falso, matrices de cero o más valores, colecciones de pares nombre-valor y valores nulos, por lo que todos los siguientes son tipos de datos JSON válidos:

- precisión: 1.6430000066757202
- completa:
- rodamiento verdadero : 0
- proveedor: ”fusionado”

La manera más sencilla de comprobar que los datos registrados son correctos es copiando los valores de longitud y latitud en Google maps, lo que debería arroarnos la ubicación que estamos testeando en el mapa.

Figura 3.4: Prueba de datos de ubicación en Google Maps

Además, al utilizar la Base de datos en tiempo real de Firebase se desea configurar algunas reglas de seguridad de la base de datos, que definen cómo deben estructurarse, indexarse y cuándo pueden leer y escribirse datos. También se usó reglas para especificar quién tiene acceso a la base de datos de Firebase, aunque firebase ya restringe el acceso a usuarios autenticados de manera predeterminada.

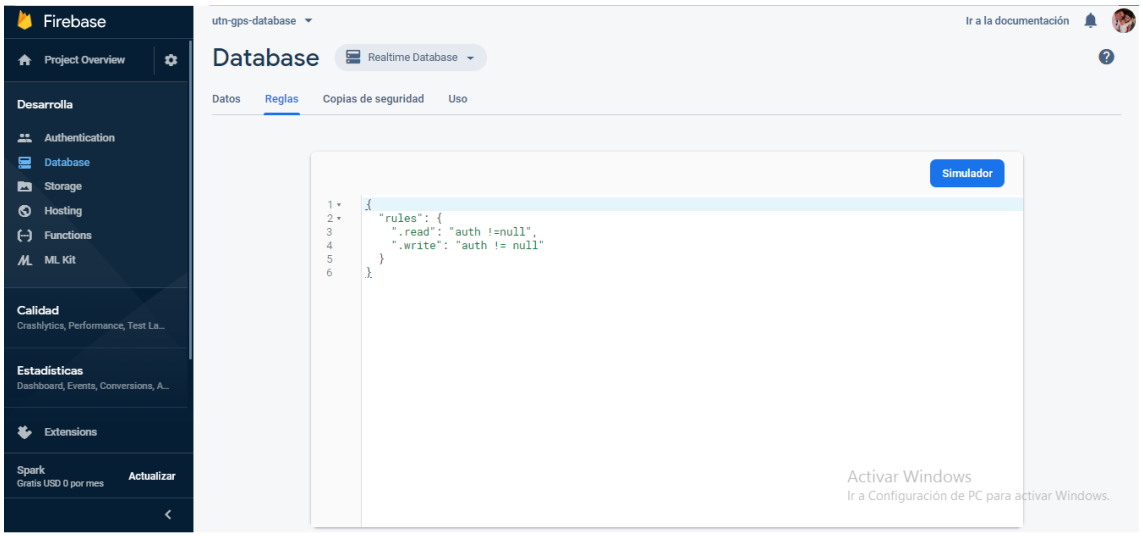


Figura 3.5: Configuración de suguridad y parámetros de base de datos

Conclusiones y trabajo futuro

Este capítulo muestra las conclusiones del presente proyecto y esboza algunas líneas posibles de trabajo futuro.

Conclusiones

- Como primera fase para el desarrollo del proyecto se analizó el software y funcionalidades de plataformas existentes en el mercado para asi proponer la aplicación a desarrollar, teniendo en cuanta indicadores como accesibilidad, costo, facilidad de modificaciones futuras, usabilidad, programación, entre otras.
- Se probó dispositivos de seguimiento disponibles en el mercado para comprender su funcionamiento y adaptabilidad al software de desarrollo.
- Se desarrolló la aplicación con los requerimientos y carazterísticas descritos a lo largo de este trabao escrito de tal manera que cumpla con la gestión de datos de posicionamiento global.
- La aplicación es capaz de almacenar datos en tiempo real en la base de datos de la plataforma de desarrollo firebase, la misma que proporciona acceso remoto unicamente accediendo a la cuanta firebase del ususario.
- Para someter a la aplicación a circunstancias reales de funcionamiento se realizo pruebas en recorridos largos los cuales proporcionaron datos variables en la base de datos, de ésta manera comprobando que la información se actualiza y se almacena de manera correcta y en tiempo real.
- Al ser una base de datos virtual se tiene la ventaja de poder acceder a la informacion de manera remota unicamente con un ordenador con acceso a internet.

Recomendaciones

- Es recomendado contar con una conexión estable a internet para que la aplicación se desempeñe de manera correcta.

- En el proceso de programación es importante tener en cuenta los parámetros necesarios tanto de software como de hardware, para que la sincronización no tenga inconvenientes.
- Se recomienda al lector estar al tanto de las actualizaciones en los sistemas de desarrollo , así como el los dispositivos y sus nuevas características para trabajos futuros.

Trabajo futuro

Gracias al desarrollo de un sistema con software libre para seguimiento de posición global, las opciones para un trabajo futuro son amplias ya que el sistema es adaptable, ampliable y puede ser modificado para cualquier tipo de necesidad en el campo de seguimiento de posicion global.

La aplicación puede ser replicada, mejorada y editada a conveniencia de quien lo requiera, ya que al ser software libre el código queda liberado en su totalidad.

La aplicación tiene gran escalabilidad en el campo de seguimiento de posición, por lo que en un trabajo futuro de podría ampliar en ámbitos de seguridad, tracking de personas, animales, o flotas de trasporte a gran escala, gracias a que la base de datos se adapta a la cantidad de ususarios que se registren y usa la infraestructura de software de Google para su funcionamiento en la nube.

Apéndice A

Código

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    implementation 'com.google.firebase:firebase-auth:11.8.0'
    implementation 'com.google.android.gms:play-services-location:11.8.0'
    implementation 'com.google.firebase:firebase-database:11.8.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="utn-gps-database.firebaseapp.com">

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET"/>
```

```
<string name="tracking_enabled_notif">Tracking is currently enabled. Tap to cancel.</string>
<string name="test_email">test@test.com</string>
<string name="test_password">testpassword</string>
<string name="firebase_path">location</string>
```

```
import android.app.Activity;
import android.support.v4.app.ActivityCompat;
import android.os.Bundle;
import android.support.v4.content.ContextCompat;
import android.content.Intent;
import android.location.LocationManager;
import android.Manifest;
import android.content.pm.PackageManager;
import android.widget.Toast;

public class MainActivity extends Activity {

    private static final int PERMISSIONS_REQUEST = 100;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
//Check whether GPS tracking is enabled//

        LocationManager lm = (LocationManager) getSystemService(LOCATION_SERVICE);
        if (!lm.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
            finish();
        }

//Check whether this app has access to the location permission//

        int permission = ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION);

//If the location permission has been granted, then start the TrackerService//

        if (permission == PackageManager.PERMISSION_GRANTED) {
            startTrackerService();
        } else {

            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                    PERMISSIONS_REQUEST);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
            grantResults) {

//If the permission has been granted...//

        if (requestCode == PERMISSIONS_REQUEST && grantResults.length == 1
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) {

//...then start the GPS tracking service//

            startTrackerService();
        } else {

//If the user denies the permission request, then display a toast with some more information//

            Toast.makeText(this, "Please enable location services to allow GPS tracking", Toast
                .LENGTH_SHORT).show();
        }
    }

//Start the TrackerService//

    private void startTrackerService() {
        startService(new Intent(this, TrackingService.class));

//Notify the user that tracking has been enabled//

       Toast.makeText(this, "GPS tracking enabled", Toast.LENGTH_SHORT).show();

//Close MainActivity//

        finish();
    }

}

import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.android.gms.location.FusedLocationProviderClient;
```

```
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;

import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.support.v4.content.ContextCompat;
import android.os.IBinder;
import android.content.Intent;
import android.content.IntentFilter;
import android.util.Log;
import android.Manifest;
import android.location.Location;
import android.app.Notification;
import android.content.pm.PackageManager;
import android.app.PendingIntent;
import android.app.Service;

public class TrackingService extends Service {

    private static final String TAG = TrackingService.class.getSimpleName();

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public void onCreate() {
        super.onCreate();
        buildNotification();
        loginToFirebase();
    }

//Create the persistent notification//

    private void buildNotification() {
        String stop = "stop";
        registerReceiver(stopReceiver, new IntentFilter(stop));
        PendingIntent broadcastIntent = PendingIntent.getBroadcast(
                this, 0, new Intent(stop), PendingIntent.FLAG_UPDATE_CURRENT);

// Create the persistent notification//
        Notification.Builder builder = new Notification.Builder(this)
                .setContentTitle(getString(R.string.app_name))
                .setContentText(getString(R.string.tracking_enabled_notif))
                .setOngoing(true)
                .setContentIntent(broadcastIntent)
                .setSmallIcon(R.drawable.tracking_enabled);
        startForeground(1, builder.build());
    }

    protected BroadcastReceiver stopReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {

//Unregister the BroadcastReceiver when the notification is tapped//
```

```
            unregisterReceiver(stopReceiver);

//Stop the Service//

           stopSelf();
         }
    };

    private void loginToFirebase() {

//Authenticate with Firebase, using the email and password we created earlier//

        String email = getString(R.string.test_email);
        String password = getString(R.string.test_password);

//Call OnCompleteListener if the user is signed in successfully//

        FirebaseAuth.getInstance().signInWithEmailAndPassword(
                email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(Task<AuthResult> task) {

//If the user has been authenticated...//

                if (task.isSuccessful()) {

//...then call requestLocationUpdates//

                    requestLocationUpdates();
                } else {

//If sign in fails, then log the error//

                   Log.d(TAG, "Firebase authentication failed");
                }
            }
        });
    }

//Initiate the request to track the device's location//

    private void requestLocationUpdates() {
        LocationRequest request = new LocationRequest();


      request.setInterval(10000);

//Get the most accurate location data available//

        request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
        FusedLocationProviderClient client = LocationServices.getFusedLocationProviderClient(
            this);
        final String path = getString(R.string.firebase_path);
        int permission = ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION);

      if (permission == PackageManager.PERMISSION_GRANTED) {


            client.requestLocationUpdates(request, new LocationCallback() {
                @Override
                public void onLocationResult(LocationResult locationResult) {

//Get a reference to the database, so your app can perform read and write operations//
```

```
                    DatabaseReference ref = FirebaseDatabase.getInstance().getReference(path);
                    Location location = locationResult.getLastLocation();
                    if (location != null) {

//Save the location data to the database//

                      ref.setValue(location);
                    }
                }
            }, null
        }
    }
}
```

Bibliografía

[1] R. M. Stallman, Software libre para una sociedad libre, Madrid: Traficantes de Sueños, 2004.

[2] J. C. Cobo Romaní, «Conocimiento, creatividad y software libre: una oportunidad para la educación en la sociedad actual.,» OUC Papers-Revista sobre la sociedad del conocimiento, nº 8, 2009.

[3] D. Cerda y I. Pazmiño, «Diseño e Implementacion de un Sistema con GPS y Control de Seguridad Vehicular,» Escuela Politécnica del Ejercito, Carrera de Ingeniería Automotriz, vol. 1, 2011.

[4] «tracklink.com.ec,» [En línea]. Available: http://tracklink.com.ec. [Último acceso: 18 Julio 2018].

[5] «www.alertagpsec.com,» [En línea]. Available: https://www.alertagpsec.com/. [Último acceso: 2018 Julio 18].

[6] C. 2011, «www.hunter.com.ec,» [En línea]. Available: http://www.hunter.com.ec/soluciones.aspx. [Último acceso: 2018 Julio 18].

[7] L. I. Chulde y G. D. Brito, «Sistema De Monitoreo Web Para Vehículos Mediante Hardware y Software Libre,» Universidad Técnica de Ambato, 2017.

[8] D. V. H. Oña, «Diseño, implementación de un sistema de monitoreo para el vehículo Mazda BT-50 de la escuela de conducción de la ESPE-L vol. 1, 2013.

[9] N. M. L. M. M. P. Z. O. E. L. L. S. Jimmy Rolando Molina Ríos, «Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,» vol. 4, nº 4, 2016.

[10] S. Chazallet, Python 3 los Fundamentos del Lenguaje. F. Piqueres, Ed. Barcelona: ENI,2015.

[11] H. Spona, Programación de Bases de Datos con MySQL y PHP. Marcombo, Ed. México: Alfaomega, 2012.

[12] J. Rea, “Dise˜no, desarrollo e implementación de un protocolo de comunicaciones entre sensores en red y computadores”, trabajo de fin de grado, Universidad Autónoma de Madrid,Madrid, 2006.

[13] M. Sabana, PHP con PostgreSQL 8. Lima: Megabyte, 2006.

[14] A. Castro, J. González y M. Callejas, “Utilidad y funcionamiento de las bases de datos NoSQL”, J. CEDEC, vol. 21, no. 33, pp. 21-32, dic. 2012.

[15] R. Herranz, “Bases de datos NoSQL: arquitectura y ejemplos de aplicación”, trabajo de fin de grado, Universidad Carlos III de Madrid, Madrid, 2014.

[16] Gutiérrez, Javier J. Qué es un framework web. Available in:www.lsi.us.es/javierj/investigacion-ficheros/Framework. pdf Accessed May12(2014).

[17] Rueda Miguel, ”Sockets: Comunicación entre procesos distribuidos.”, Doctorado en informática. Junio 1996.

[18] Tanembaum Andrew, Computer networks, 5th edition,2011.

[19] D.E. Comer, Ïnternetworking with TCP/IP”Vol 1. 2013.

[20] Arkaitz Garro, ”HTML5”, Cap13:Websockets, 2014.

[21] I. Fette, ”The websocket protocol”, version3, 2010.

[22] ZHAO, Yan. Client/server two-way communication system framework under HTTP protocol. U.S. Patent Application No 09/776,478, 8 Ago. 2002.

[23] M. Tabares, R. Anaya y F. Arango, “La ingeniería de requisitos orientada a aspectos: una experiencia de aplicación en un sistema de ayuda en línea”, Scielo, no. 153, pp. 285-299, nov. 2007. doi: ISSN 0012-7353

[24] V. Hernández y Y. Machado, “Describiendo requisitos verificables”, Innovation and Development for the Americas, jun. 2010.

[25] J. Rumbaugh, I. Jacobson y J. Booch, El Lenguaje Unificado de Modelado. Manual de Referencia. Madrid: Addison Wesley, 2000.

[26] H. Cervantes, P. Velasco y L. Castro, Arquitectura de Software, Conceptos y Ciclo de Desarrollo. R. Kinney, Ed. México: CENGAGE Learning, 2016.