

Universidad Nacional Autónoma de Nicaragua-León

Facultad de Ciencias y Tecnología

Departamento de Computación

Ingeniería en Telemática



Monografía para Optar al Título de Ingeniero en Telemática

Desarrollo de una aplicación móvil para dispositivos con Sistema Operativo Android que servirá como diccionario digital español-inglés-miskito-mayangna

Autor:

Br. Dayan Ariel Campos Ramos

Tutor:

M.Sc. Valeria Mercedes Medina Rodríguez

León, Nicaragua, mayo 2022

“A la Libertad por la Universidad”

Dedicatorias

Dedico esta monografía a mi familia que siempre estuvieron brindándome su apoyo a lo largo de toda mi carrera y formación día a día.

A mi madre Cristina Xiomara Ramos Munguía por su ayuda y consejos que me impulsaban cada vez más adelante hasta llegar a este punto.

A mi padre Luis Ángel Campos Saavedra que con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño más.

A mi hermana Estrellita Lisbeth Campos Ramos quien siempre me brindaba su apoyo las veces que lo necesitaba.

Agradecimientos

Le agradezco a Dios en primer lugar, por los pasos y pruebas que tuve que superar para llegar donde me encuentro en este preciso momento. Cada trabajo, cada examen, cada prueba lo encomendé a él.

Todo resulto de la mejor manera por eso le doy las gracias a Dios que estuvo conmigo en cada momento, con la presentación de esta monografía estoy dando uno de los pasos muy importante de mi vida, con esta monografía estoy cumpliendo uno de mis grandes sueños, obtener mi título de ingeniero en Telemática.

A cada uno de los maestros que fueron aportando su granito de arena desde el inicio de mi carrera, desde el primer año de Ingeniería en Telemática donde no tenía ni el más mínimo conocimiento sobre el despliegue de las redes de telecomunicaciones y el desarrollo de una aplicación. En especial a la profesora Valeria Medina, en primer lugar, por haber accedido a ser mi tutor y estar en momentos muy importantes en el desarrollo del presente trabajo. Por brindarme su tiempo fuera del aula de clases y seguir enseñándome.

A todas las demás personas que me apoyaron de una u otra manera, muchas gracias por todo su apoyo, ¡Dios los bendiga!

Contenido

1. Introducción	1
2. Antecedentes.....	2
3. Planteamiento del problema	3
3.1. Pregunta General.....	3
3.2. Preguntas específicas.....	3
4. Justificación	5
5. Objetivos.....	6
5.1. Objetivo general	6
5.2. Objetivos específicos	6
6. Marco teórico	7
6.1. Costa Caribe de Nicaragua	7
6.2. Dialectos e idiomas en Nicaragua.....	8
6.3. Diccionario	12
6.4. Aplicaciones de diccionarios y traductores	13
6.5. Características que debe presentar una aplicación	14
6.6. Componentes de una aplicación	15
6.7. Tecnologías para el desarrollo de la aplicación	17
6.7.1. Java.....	17
6.7.2. Android Studio	21
6.7.3. Cliente-Servidor	25
6.7.4. Firebase.....	28
7. Diseño metodológico	31
7.1. Descripción general del proyecto	31
7.2. Fases del proyecto.....	31
8. Desarrollo	35
8.1. Plantilla utilizada para el diseño de la aplicación.	35
9. Resultados.....	41
9.1. Recopilación de la información	41
9.2. Construcción del diagrama	41
9.2.1. Casos de uso.....	41
9.2.2. Diagrama de las bases de datos.....	43
9.2.3. Apartados de la aplicación.....	45

9.2.4.	Diagrama de clases	46
9.2.5.	Pruebas del funcionamiento de la aplicación	46
10.	Conclusiones.....	48
11.	Recomendaciones.....	49
12.	Bibliografía	51
13.	Anexos	53

Lista de Tablas

Tabla 1: Palabras reservadas	19
Tabla 2: Tipos de datos en JAVA.....	19
Tabla 3: Funciones de Firebase	28
Tabla 4: Requisitos para instalar Android Studio	33
Tabla 5: Recurso hardware	34
Tabla 6: Dispositivos móviles y emuladores.....	34

Lista de Figuras

Ilustración 1: Mapa de las lenguas indígenas de la Costa Caribe de Nicaragua.....	8
Ilustración 2: Sintaxis básica de java.....	21
Ilustración 3: Los archivos de proyecto en la vista de Android.....	23
Ilustración 4: Interfaz de usuario	24
Ilustración 5: Esquema del modelo cliente servidor	27
Ilustración 6: Vista bienvenida.....	35
Ilustración 7: Vista principal.....	35
Ilustración 8: Menú desplegable.....	36
Ilustración 9: Vista de enviar traducción.....	36
Ilustración 10: Vista idiomas seleccionados	37
Ilustración 11: Vista cuenta	37
Ilustración 12: Vista información	38
Ilustración 13: Vista iniciar sesión	38
Ilustración 14: Vista registrarse	39
Ilustración 15: Vista recuperar contraseña	39
Ilustración 16: Caso de uso de las distintas funciones de la aplicación	42
Ilustración 17: Estructura de la base de datos principal	43
Ilustración 18: Estructura de la base de datos secundaria	44
Ilustración 19: Diagrama de clases	46
Ilustración 20: Cronograma de actividades.....	53
Ilustración 21: Vista de bienvenida.....	54
Ilustración 22: Vista principal.....	54
Ilustración 23: Búsqueda de palabras	55
Ilustración 24: Menú desplegable.....	55
Ilustración 25: Vista idiomas seleccionados	56
Ilustración 26: Vista enviar traducción.....	56
Ilustración 27: Vista iniciar sesión	57
Ilustración 28: Vista recuperar contraseña	57
Ilustración 29: Vista registrarse	58
Ilustración 30: Iniciar sesión con Facebook.....	58

Ilustración 31: Iniciar sesión con Google.....	59
Ilustración 32: Vista cuenta	59
Ilustración 33: Vista información	60
Ilustración 34: Métodos habilitados para el inicio de sesión.....	69
Ilustración 35: Usuarios registrados	69
Ilustración 36: Base de datos principal.....	70
Ilustración 37: Base de datos secundaria.....	70



1. Introducción

Nicaragua es el país más grande de la región centroamericana, con una extensión territorial de 130,373 km². La espectacular geografía de Nicaragua incluye cerca de 800 km de costas en el Pacífico y Caribe, 25 formaciones volcánicas, más de 20,000 km de reservas naturales, cerca de 10,000 km² de lagos, lagunas y ríos, y un impresionante 7% de la biodiversidad del mundo. (PRONicaragua, 2017)

La Costa Caribe de Nicaragua ocupa el 46% del territorio nacional y representa para el país el 35% del hato ganadero, 23% del total del área agrícola, más del 80% del área forestal, 70% de la producción pesquera, 60% de los recursos mineros, 45,000 Km² con alto potencial de explotación de hidrocarburos y más de 450 Km de costas en una de las zonas turísticas más reconocidas del mundo. La principal actividad empresarial de la Costa Caribe es el comercio al por mayor y al detalle, ésta compone el 61% en la Región Autónoma de la Costa Caribe Sur (RACCS) y el 69% en la Región Autónoma de la Costa Caribe Norte (RACCN). (PRONicaragua, 2017)

En las Regiones Autónomas de Nicaragua habita el 13% de la población total del país, lo que la convierte en la segunda región más poblada de Nicaragua, después del departamento de Managua. En esta región del país reside la mayor parte de los pueblos indígenas y afrodescendientes, los que enriquecen y aportan de manera significativa al desarrollo con su identidad. Esta combinación de riquezas naturales e históricas, diversidad cultural y posición geográfica permite que la Costa Caribe nicaragüense aporte sustancialmente al país en términos: sociales, culturales, políticos y económicos, teniendo aún un gran potencial por desarrollar. (PRONicaragua, 2017)

En este trabajo monográfico se desarrolló un diccionario digital lingüístico que facilitará la comunicación en las regiones de la Costa Caribe nicaragüense principalmente en las comunidades rurales de cada uno de sus municipios, que en un futuro puede ayudar a fomentar el comercio, el turismo, la cultura y la educación atreves del dialecto.



2. Antecedentes

- ❖ “Fundación Tuahka”, Diccionario Cuatrilingüe Tuahka Tuahka-Miskito-Español-Inglés. La meta de la Fundación Tuahka es Investigar, Recopilar y Promulgar la cultura indígena Tuahka. Asimismo, apoyar y facilitar recursos a las víctimas de desastres naturales y fortalecer las capacidades físicas y espirituales de las personas que viven en situaciones de extrema pobreza al interior de esta comunidad indígena.

Este Diccionario Cuatrilingüe Tuahka es el segundo que se publica en idioma Tuahka con palabras simples y compuestas, nombres de lugares y sus significados, términos y frases que habían desaparecido y fueron rescatados a través de una investigación minuciosa que se realizó en las comunidades Tuahkas.

En las traducciones al castellano y al inglés se dan algunas imperfecciones, principalmente en los ejemplos, ya que existen algunas palabras que no tienen un equivalente realmente satisfactorio en estos idiomas europeos. Esto es debido a que el Tuahka es un idioma indígena original de la Región Autónoma del Atlántico Norte de Nicaragua y de Honduras. (Tuahka, Mayo de 2014)

- ❖ Desarrollo de una app móvil para dispositivos con Sistema Operativo Android, que servirá como diccionario digital español-mayangna. El pueblo indígena Mayangna es muy poco conocido en el mundo e incluso en el mismo territorio nacional ya que muy poco se habla de ellos en las escuelas y las universidades.

Es por ello la necesidad de hacer una aplicación móvil que sirva como traductor y diccionario de español a mayangna y viceversa, para que el idioma llegue a conocerse de manera más sencilla, para ofrecerles a los turistas locales y extranjeros una herramienta digital con la cual van a poder interactuar con los aborígenes de esa zona del país y conocer más sobre su cultura e idioma y así rescatar esta lengua indígena que es muy poco conocida dentro de nuestra cultura nicaragüense. (Villanueva Mejía, Sánchez Corea, & Frank, 2019)



3. Planteamiento del problema

En las regiones de la Costa Caribe nicaragüense principalmente en las comunidades rurales de cada uno de los municipios, existe el problema para la comunicación entre las diferentes etnias.

Los idiomas que se hablan son: el español, miskito, mayangna e inglés, en la Costa Caribe predominan el miskito y el mayangna, a los turistas nacionales y extranjeros que visitan la Costa Caribe de Nicaragua se les dificulta la comunicación ya que no dominan estos idiomas lo que puede crear un obstáculo para el desarrollo en el comercio, turismo, cultura y la educación.

En el ámbito académico existen docentes que hablan solo un idioma (español) cuando el Ministerio de Educación asigna un docente a una de las comunidades lejanas la comunicación entre los estudiantes, padres de familia y el mismo docente se les dificulta ya que no hablan el mismo idioma.

Otra de las problemáticas vista en nuestro país es la falta de conocimiento sobre la existencia de estos lenguajes y la cultura de los habitantes de la zona, por lo que estos son conservadores en cuanto a socializar con personas aledañas a sus comunidades.

3.1. Pregunta General

- ❖ ¿Cómo facilitar la comunicación entre los turistas nacionales y extranjeros con los habitantes de la Costa Caribe de Nicaragua mediante una aplicación Android?

3.2. Preguntas específicas

- ❖ ¿Qué herramientas software se deben utilizar para desarrollar una aplicación diccionario español-inglés-miskito-mayangna de tal forma que sea intuitiva y de fácil uso?
- ❖ ¿Qué tipo de base de datos se utilizará para el almacenamiento de las traducciones?



- ❖ ¿De dónde se obtendrán todas las palabras y expresiones con sus respectivas traducciones para agregarlas a la base de datos de tal forma que estas sean suficiente para un uso cotidiano?
- ❖ ¿De qué forma los usuarios podrán ayudar a enriquecer los datos almacenados en la base de datos?



4. Justificación

La comunicación escrita sirve para informar una idea o pensamiento por medio de un texto ya sea por: periódicos, cartas, Internet, aplicaciones, etc. Los dialectos miskito y mayangna son de las lenguas más antiguas en la historia de Nicaragua, sin embargo, son pocos conocidos y estudiados ya que el idioma que más predomina en Nicaragua es el español.

La creación de un diccionario digital lingüístico que facilite la comunicación de las personas que no dominan algunos de los siguientes idiomas: español, inglés, miskito y mayangna es de suma importancia para poder interactuar, aprender y descubrir otras culturas.

Los diccionarios existentes que podrían servir para la traducción y el aprendizaje de los dialectos miskito y mayangna son muy pocos y no son conocidos por la mayoría de la población nicaragüense por lo cual es necesario un diccionario digital lingüístico de fácil acceso y entendimiento para ayudar a la comprensión de estos mismos.



5. Objetivos

5.1. Objetivo general

- ❖ Desarrollar una aplicación móvil que sirva como diccionario digital lingüístico de los idiomas español, inglés, miskito y mayangna para facilitar la comunicación entre los turistas nacionales y extranjeros con los habitantes de la Región Autónoma de la Costa Caribe Norte de Nicaragua.

5.2. Objetivos específicos

- ❖ Escoger el software empleado para desarrollar la aplicación y establecer los requerimientos mínimos para el óptimo funcionamiento de la aplicación en los dispositivos con sistema operativo Android 5.0 y posterior.
- ❖ Escoger que tipo de base de datos se utilizara para el almacenamiento de las palabras y frases de los idiomas: español, inglés, miskito y mayangna.
- ❖ Seleccionar palabras y expresiones más usadas en los idiomas con sus respectivas traducciones de tal forma que estas sean suficiente para el uso cotidiano.
- ❖ Crear una sección en la aplicación donde los usuarios puedan contribuir con nuevas traducciones.



6. Marco teórico

6.1. Costa Caribe de Nicaragua

La Costa Caribe de Nicaragua, geográficamente está constituida por las Regiones Autónomas Atlántico Norte (RAAN) y Atlántico Sur (RAAS) y el Departamento del Río San Juan. La RAAN y RAAS ocupan el 47% del territorio nacional y en ellas habita cerca del 13% de la población nacional aproximadamente 620,400 personas. Esta macro región del país constituye el mayor crecimiento demográfico de Nicaragua. Las características fundamentales de la Costa Caribe es su carácter multiétnico, pluri cultural y multilingüe por la presencia de comunidades Miskitas, Creoles, Garífunas, Sumu/Mayangnas (Twahka, Panamahka, Ulwa), Rama y Mestizas. (Mundial, 2011)

La Región Autónoma del Atlántico Norte (RAAN) se encuentra ubicada en el sector noreste del país. Posee una superficie de 32,819.68 km² integrada por 8 municipios (Waspám, Puerto Cabezas, Prinzapolka, Rosita, Bonanza, Siuna, Mulukukú y Waslala). La sede del gobierno regional se encuentra en la Ciudad de Bilwi (Puerto Cabezas). La población regional asciende a 314,130 habitantes y representa el 7% del total de la población del país, de este total de población el 57% se identifica como mestizo, un 36% como miskito, un 1% como creole o negro y un 6% como mayangna. La densidad poblacional es de 12 habitantes por kilómetro cuadrado. El 33.3% de la población vive en zonas urbanas y el 66.7% en el área rural. (Mundial, 2011)

La Región Autónoma del Atlántico del Sur (RAAS) posee una superficie de 27,260 km², integrada por 12 municipios (Nueva Guinea, El Rama, Bluefields, Bocana de Paiwas, Muelle de Los Bueyes, La Cruz del Río Grande de Matagalpa, El Tortuguero, El Ayote, Kukra Hill, Laguna de Perlas, Corn Island y Desembocadura del Río Grande de Matagalpa). La sede del gobierno regional se ubica en la Ciudad de Bluefields. La población regional asciende a 306,510 (2005); un 60% de esta población se autoidentifica como mestizos, un 22.5% como creoles o negros, un 10% como miskitos, un 2% como Rama, un 0.1% como Ulwa y el resto no se identificó con ningún grupo étnico. El 36.9% de la población de la RAAS vive en zonas urbanas y el 60.4% vive en zonas rurales de la región. (Mundial, 2011)



6.2. Dialectos e idiomas en Nicaragua

El español es el idioma oficial de Nicaragua, aunque el inglés también es un idioma importante que se habla en la nación. El país tiene varias lenguas indígenas como el inglés criollo nicaragüense, miskito, sumo, garífuna y rama, muchas de las cuales se han extinguido, mientras que otras son habladas por pequeños grupos de nicaragüenses.

Los habitantes de la Costa Caribe de Nicaragua además de hablar su idioma o lengua natal la mayoría de ellos también dominan el español; de igual manera los turistas extranjeros que visitan esta tierra de lagos y volcanes en su mayoría también además de dominar su idioma natal dominan entre un 40% o 50% el idioma español.



Ilustración 1: Mapa de las lenguas indígenas de la Costa Caribe de Nicaragua

En Nicaragua han dejado de existir un total de 4 idiomas: náhuat, mangué, subtiaba y matagalpa. El español es hablado por la mayoría de la población, en un 90%. A lo largo y ancho del país, el idioma adopta diversas variedades en cuanto a vocabulario, acentos y coloquialismos. (QUEIDIOMA, n.d.)



Idiomas hablados en la Costa Caribe de Nicaragua

En Nicaragua se hablan dos formas de inglés: el inglés criollo, con aproximadamente 30,000 hablantes y el inglés estándar con aproximadamente 20,334 hablantes. El inglés criollo prevalece a lo largo de la costa caribeña de Nicaragua debido a la influencia del patrimonio africano e inglés en la región. Grandes secciones de la población de las ciudades de esta región, como Bluefields, hablan inglés criollo. El inglés estándar lo hablan los emigrantes de países de habla inglesa que se establecieron en otras naciones como Estados Unidos, el Reino Unido y Canadá. El inglés también es ampliamente utilizado por quienes participan en el sector turístico de Nicaragua. (Wise, 2021)

Español: En el español de Nicaragua el sustrato náhuatl se presenta como la característica lingüística en el nivel léxico más identitaria. La herencia lingüística que dejaron los grupos nahuas en la cultura, específicamente en la gastronomía, en nombres de plantas, animales y topónimos como nombres de ríos, lagos, montañas y los nombres de ciudades y pueblos es una herencia que en la actualidad continúan vigentes, pero que en la medida que pasa el tiempo está siendo absorbida por la variante dominante esto es el español estándar. Los cambios culturales y tecnológicos han provocado un desplazamiento de las palabras procedentes o de origen náhuatl, así palabras como: chicle (goma de mascar), chibola (canica), chapa (arete), zacate (hierba), entre otros son términos de uso común en el español nicaragüense, sin embargo, estos cada vez más están siendo desplazados por los equivalentes en el español estándar. (Úbeda, 2020)

Ingles Creole: Muchos caribeños hablan, un idioma que comenzó en África cuando los primeros europeos llegaron al continente africano. Esto ocurrió hace 350 años. Los africanos y los europeos no se entendían, por eso tuvieron que inventar una manera de comunicarse. Africanos y europeos construyeron un nuevo idioma mezclando palabras de las lenguas de cada uno. Escogían palabras que les sirvieran para comprar y vender. (Saldaña, 2002)

Para construir el idioma creole, la gente tomó algunas palabras de sus idiomas africanos. Tomaron, igualmente, muchas palabras del idioma de sus amos. Arreglaron las palabras



de manera que guardaran un orden parecido al de sus idiomas africanos. Pronunciaban las palabras tomadas prestadas, de una manera africana. (Saldaña, 2002)

El idioma creole era importante para los esclavos. Ellos necesitaban encontrar una manera de comunicarse entre ellos. Pero a los dueños de los esclavos no les interesaba el creole. Ellos continuaron comunicándose en sus idiomas europeos. Los blancos podían utilizar sus propios idiomas entre ellos, los negros no podían hacerlo con los suyos. Por eso los esclavos crearon el idioma creole, para que reemplazara a sus idiomas africanos que no podían ya más utilizar. (Saldaña, 2002)

Miskito: Es el idioma indígena más hablado de Nicaragua, con aproximadamente 154,400 hablantes nativos. El idioma lo hablan los miskitos que habitan en la región costera del Caribe, al noreste de Nicaragua. La población más alta de hablantes de Miskito se concentra en la Región Autónoma de la Costa del Caribe Norte. El idioma pertenece a la familia de idiomas nativos de Misumalpan. (Wise, 2021)

El miskito es una lengua morfológica y sintácticamente muy parecida a la lengua tawahka, pero ambas poseen diferencias marcadas; el idioma miskito moderno o criollo que hablan actualmente los miskitos de Nicaragua posee una gran cantidad de extranjerismos provenientes del idioma inglés, español y africano. (EcuRed, n.d.)

Sumo o Mayangna: El Sumo es hablado por el pueblo Sumo de Nicaragua. Hay varios dialectos del idioma Sumo, algunos son tan diferentes de los otros que pueden tratarse como idiomas separados. (Wise, 2021)

La lengua Mayangna la hablan unos 25 mil habitantes de la Costa Atlántica de Nicaragua y Honduras. Su lengua está clasificada hasta donde se ha estudiado como una lengua del grupo MISUMALPA a la que pertenecen las lenguas: Miskito, Sumu-Mayangna y Matagalpa, que tienen a su vez una estrecha relación con otras lenguas de Centro y Sur América que conforma la gran familia Chibcha, pues el miskito y el Mayangna tienen una misma manera de armar las frases y oraciones. (Patrón, 2007)

Actualmente existen tres formas distintas de la lengua Mayangna: El Panamaka, Tuahka y el Ulwa, este último, lo hablan principalmente en la comunidad de Karawala y algunos



habitantes de la comunidad de Kara, ambas comunidades se encuentran en la Región Autónoma del Atlántico Sur, RAAS. (Patrón, 2007)

Rama: Es un idioma de la familia Chibchan, es una lengua indígena severamente amenazada de Nicaragua. El lenguaje fue descrito como "muriendo rápidamente" desde 1862. Los esfuerzos de reactivación para salvar los idiomas no tuvieron éxito. El idioma fue hablado previamente por la gente Rama que vive en la costa caribeña de Nicaragua y la isla Rama Cay. Sin embargo, a lo largo de los años, la gente de Rama comenzó a hablar una nueva forma de inglés criollo, el creole de Rama Cay, en lugar del idioma Rama. (Wise, 2021)

El territorio Rama está constituido por la pequeña isla de Rama Cay, ubicada a unos 15 kilómetros al sur de Bluefields, y las comunidades La Zompopera, El Coco, Wiring Cay, Punta de Águila, río Maíz y río Indio. La población rama es uno de los pueblos indígenas de la Costa Caribe, con la menor cantidad porcentual de la población regional (aproximadamente 0.55%) y ubicada en la Región Autónoma Atlántico Sur –RAAS-. (Morales, McCrea, & Gutiérrez Soza, 2009)

El territorio rama administrativamente se encuentra en el municipio de Bluefields. El pueblo rama tiene una notable variedad de recursos naturales, entre ellos el recurso forestal forma parte importante en sus actividades de desarrollo ya que dependen del bosque y lo utilizan dándole múltiples usos como alimento para consumo humano, medicina y en menor grado como fuente de ingreso. La mayoría de la población está concentrada en la isla de Rama Cay. Está compuesta por dos pequeñas islas unidas por un banco formado por conchas de ostiones y caracoles (Morales, McCrea, & Gutiérrez Soza, 2009)

Garífuna: Es un pueblo transfronterizo que habita en Nicaragua, Belice, Guatemala y Honduras. Descienden de pueblos africanos e indígenas Arawak y Caribes que entraron en contacto en la época colonial en las Antillas Menores, de donde fueron desplazados. Su nombre deriva de una palabra garífuna: garinagu, que significa "caribes negros". Su lengua original era la Garífuna, pero actualmente hablan el inglés creole y el castellano. (gobernanza, 2020)



Sus territorios se encuentran en la Región Autónoma del Atlántico Sur (RAAS), junto a la Laguna de Perlas. Se concentran en las comunidades costeras, principalmente en la Comunidad Orinoco; y en segundo lugar en las comunidades de San Vicente, Justo Point, La Fe y Brown Bank. También existe un número considerable de familias Garífunas en Bluefields y Corn Island. (gobernanza, 2020)

Según el VIII Censo de Población realizado en el año 2005, en Nicaragua habitan 3.271 Garífunas. En conjunto, el Pueblo Garífuna está conformado por 450.000 personas en todo el mundo. (gobernanza, 2020)

6.3. Diccionario

Un diccionario es un libro que recopila, en orden alfabético, las palabras de una lengua y explica sus distintos significados. Los diferentes significados de cada palabra se denominan acepciones. El diccionario se puede utilizar tanto para consultar el significado de una palabra como su ortografía y pronunciación. También nos puede aportar la etimología de la palabra (origen) e información de tipo gramatical (clase de palabra, género, etc.) (Clavero)

¿Qué tipos de diccionarios hay?

- ❖ **Diccionarios de usos:** Son los fundamentales y los primeros que a temprana edad comenzamos a utilizar, principalmente en las escuelas. Se trata del diccionario en el que podemos encontrar las palabras que necesitamos para comunicarnos y su significado en nuestra lengua.
- ❖ **Diccionarios visuales:** Recopilan una serie de palabras fundamentales con su significado, pero son bastante prácticos, ya que se acompañan de fotos con las que los usuarios pueden esclarecer claramente el o los significados de las distintas palabras.
- ❖ **Diccionarios bilingües:** Son los que utilizamos cuando estudiamos una lengua extranjera. Se trata de un diccionario que recoge palabras en los dos idiomas de referencia, para que saber cómo se dice en ambas lenguas la misma palabra.



- ❖ **Diccionarios de sinónimos:** Se trata de un diccionario que contiene una lista de palabras junto a otras con el mismo significado. Esto permite encontrar grupos de palabras muy distintas, pero que significan lo mismo.
- ❖ **Diccionarios de antónimos:** Son exactamente igual que los anteriores, pero lo que recogen son listas de palabras junto a otras con significados completamente opuestos, palabras que significan justo lo contrario.
- ❖ **Diccionarios normativos:** En este caso se trata del diccionario que podríamos considerar como oficial, ya que recoge las palabras que las academias de las lenguas aceptan por su extendido uso entre los hablantes.
- ❖ **Diccionarios etimológicos:** Son aquellos que recogen el origen de las distintas palabras.
- ❖ **Diccionarios enciclopédicos:** Se trata de un tipo de diccionario muy práctico, ya que recoge el saber ya sea por etapas históricas, disciplinas de estudio, con información ampliada y detallada. Siempre van acompañados de imágenes, gráficos, esquemas y todos los elementos visuales que hacen el estudio más ilustrado y fácil.
- ❖ **Diccionarios técnicos:** Recogen palabras propias de una disciplina. Es decir, son diccionarios con un vocabulario especializado, para todos aquellos que trabajen con un sector determinado puedan encontrar de forma fácil todos los términos que le pueden servir para su trabajo. (JPimat, 2018)

6.4. Aplicaciones de diccionarios y traductores

Diccionario RAE y ASALE (DLE): DLE es la aplicación oficial que la Real Academia Española (RAE) y la Asociación de Academias de la Lengua Española (ASALE) ponen a su disposición para consultar la 23.^a edición del Diccionario de la lengua española.

Resuelve al instante y de forma fácil cualquier duda sobre el significado de una palabra y sus características ortográficas y gramaticales, accediendo directamente a la base de datos de la Real Academia Española. Tiene una interfaz ágil e intuitiva, las múltiples posibilidades de consulta al diccionario más importante de la lengua española.



Inglés Español Diccionario & Traductor: La aplicación permite buscar palabras en Inglés & Español con definiciones, ejemplos, pronunciación, etc. Las características que posee son definiciones detalladas de palabra y frases, sugerencias relacionadas de búsqueda, analizador de frases avanzado, palabras preferidas e historial de búsqueda, tamaños de fuente personalizables para una mejor legibilidad.

Traductor de Google: un sistema multilingüe de traducción automática, desarrollado y proporcionado por Google, para traducir texto, voz, imágenes o video en tiempo real de un idioma a otro. Ofrece interfaz web, así como interfaces móviles para iOS y Android, y una API, que los desarrolladores pueden utilizar para construir extensiones de navegador, aplicaciones, y otros softwares. El Traductor de Google posee la capacidad de traducir en más de 100 idiomas en distintos niveles, el sistema provee un servicio gratuito y diariamente es utilizado por más de 200 millones de personas.

Traductor U: incluye varios diccionarios, entre ellos, el diccionario de Oxford, el Collins COBUILD Advanced, el diccionario WordNet, diccionarios etimológicos, de sinónimos, de antónimos. Es capaz de traducir a más de 100 idiomas, tanto texto como voz y foto, la aplicación que tiene también lo que llaman traductor mágico y que puede traducir sin conectarse a Internet. El traductor mágico es una función disponible para Android que muestra una lupa que, al colocarla en el texto dentro de otras apps, traduce automáticamente su contenido. La aplicación incluye también una función de vocabulario en pantalla bloqueada, el cual hace que aprendamos palabras con sólo ver la pantalla bloqueada de nuestro móvil.

6.5. Características que debe presentar una aplicación

Diseño atractivo: El diseño gráfico es uno de los elementos clave en las aplicaciones. Da un aspecto fresco y diferenciador siguiendo las líneas de cada sistema operativo. En el éxito de una aplicación es el cuidado de los pequeños detalles, la creatividad, el diseño intuitivo y especialmente la usabilidad de la que depende la aplicación. Diseñar implica pensar en todos los aspectos clave para satisfacer y cubrir las necesidades de los usuarios con la mayor calidad.



Escalabilidad: Se entiende por escalabilidad a la capacidad de adaptación y respuesta de un sistema con respecto al rendimiento de este a medida que aumentan de forma significativa el número de usuarios.

Buen marketing: El marketing móvil supone un canal entre empresas, clientes y usuarios que favorece el crecimiento y expansión del negocio. Para ello se emplean los mejores medios y elementos para promocionar los productos y servicios. Esto supone un cambio de paradigma, acercándose a la tecnología y el uso de sus herramientas para llegar a un público objetivo.

Valor agregado: El valor agregado es la característica extra que un producto o servicio ofrece con el propósito de generar mayor valor comercial dentro de la percepción del consumidor. Este término puede ser el factor determinante entre el éxito o el fracaso que una empresa tenga, debido a que se encarga, también, de diferenciarla de la competencia.

6.6. Componentes de una aplicación

Existe una serie de elementos clave que resultan imprescindibles para desarrollar aplicaciones en Android. Lo siguiente es una descripción inicial de algunos de los más importantes. (Valencia, 2017)

View: Las vistas son los elementos que componen la interfaz de usuario de una aplicación: por ejemplo, un botón o una entrada de texto. Todas las vistas van a ser objetos descendientes de la clase View y por tanto pueden ser definidas utilizando código Java. Sin embargo, lo habitual será definir las vistas utilizando un fichero XML y dejar que el sistema cree los objetos por nosotros a partir de este fichero. (Valencia, 2017)

Layout: Un layout es un conjunto de vistas agrupadas de una determinada forma. Se puede disponer de diferentes tipos de layouts para organizar las vistas de forma lineal, en cuadrícula o indicando la posición absoluta de cada vista. Los layouts también son objetos descendientes de la clase View. Los layouts pueden ser definidos en código, aunque la forma habitual de definirlos es utilizando código XML. (Valencia, 2017)



Activity: Una aplicación en Android va a estar formada por un conjunto de elementos básicos de visualización, coloquialmente conocidos como pantallas de la aplicación. En android cada uno de estos elementos o pantallas se conoce como actividad. Su función principal es la creación de la interfaz de usuario. Una aplicación suele necesitar varias actividades para crear la interfaz de usuario. Las diferentes actividades creadas serán independientes entre sí, aunque todas trabajarán para un objetivo común. (Valencia, 2017)

Fragment: Un fragment está formado por la unión de varias vistas para crear un bloque funcional de la interfaz de usuario. Una vez creados los fragments, podemos combinar uno o varios fragments dentro de una actividad, según el tamaño de pantalla disponible. (Valencia, 2017)

Service: Un servicio es un proceso que se ejecuta “detrás”, sin la necesidad de una interacción con el usuario. Se utilizan cuando quiere tener en ejecución un código de manera continua, aunque el usuario cambie de actividad. En Android disponemos de dos tipos de servicios: servicios locales, que son ejecutados en el mismo proceso, y servicios remotos, que son ejecutados en procesos separados. (Valencia, 2017)

Intent: Una intención representa la voluntad de realizar alguna acción, como realizar una llamada de teléfono, visualizar una página web. Se utiliza cada vez que queramos: lanzar una actividad, lanzar un servicio, enviar un anuncio de tipo broadcast, comunicarnos con un servicio. (Valencia, 2017)

Broadcast Receiver: Un receptor de anuncios recibe anuncios broadcast y reacciona ante ellos. Los anuncios broadcast pueden ser originados por el sistema (por ejemplo: Batería baja, Llamada entrante) o por las aplicaciones. Las aplicaciones también pueden crear y lanzar nuevos tipos de anuncios broadcast. Los receptores de anuncios no disponen de interfaz de usuario, aunque pueden iniciar una actividad si lo estiman oportuno. (Valencia, 2017)

Content Provider: En muchas ocasiones las aplicaciones instaladas en un terminal Android necesitan compartir información. Android define un mecanismo estándar para



que las aplicaciones puedan compartir datos sin necesidad de comprometer la seguridad del sistema de ficheros. Con este mecanismo podremos acceder a datos de otras aplicaciones, como la lista de contactos, o proporcionar datos a otras aplicaciones. (Valencia, 2017)

6.7. Tecnologías para el desarrollo de la aplicación

6.7.1. Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes. (Java, n.d.)

La tecnología Java se usa para desarrollar aplicaciones para un amplio alcance de entornos, desde dispositivos del consumidor hasta sistemas empresariales heterogéneos. (IBM, n.d.)

Como cualquier lenguaje de programación, el lenguaje Java tiene su propia estructura, reglas de sintaxis y paradigma de programación. El paradigma de programación del lenguaje Java se basa en el concepto de programación orientada a objetos (OOP), que las funciones del lenguaje soportan. (IBM, n.d.)

El lenguaje Java es un derivado del lenguaje C, por lo que sus reglas de sintaxis se parecen mucho a C: por ejemplo, los bloques de códigos se modularizan en métodos y se delimitan con llaves ({ y }) y las variables se declaran antes de que se usen. (IBM, n.d.)

Estructuralmente, el lenguaje Java comienza con paquetes. Un paquete es el mecanismo de espacio de nombres del lenguaje Java. Dentro de los paquetes se encuentran las clases y dentro de las clases se encuentran métodos, variables, constantes, entre otros. (IBM, n.d.)



Cuando se programa para una plataforma Java, se escribe el código de origen en archivos .java y luego los compila. El compilador verifica su código con las reglas de sintaxis del lenguaje, luego escribe los códigos byte en archivos .class. Los códigos byte son instrucciones estándar destinadas a ejecutarse en una Java Virtual Machine (JVM). Al agregar este nivel de abstracción, el compilador Java difiere de los otros compiladores de lenguaje, que escriben instrucciones apropiadas para el chipset de la CPU en el que el programa se ejecutará. (IBM, n.d.)

Al momento de la ejecución, la JVM lee e interpreta archivos .class y ejecuta las instrucciones del programa en la plataforma de hardware nativo para la que se escribió la JVM. La JVM interpreta los códigos byte del mismo modo en que una CPU interpretaría las instrucciones del lenguaje del conjunto. La diferencia es que la JVM es un software escrito específicamente para una plataforma particular. La JVM es el corazón del principio «escrito una vez, ejecutado en cualquier lugar» del lenguaje Java. Su código se puede ejecutar en cualquier chipset para el cual una implementación apropiada de la JVM está disponible. Las JVM están disponibles para plataformas principales como Linux y Windows y se han implementado subconjuntos del lenguaje Java en las JVM para teléfonos móviles y aficionados de chips. (IBM, n.d.)



Palabras reservadas

Ciertas palabras están reservadas para uso interno por Java y no se pueden utilizar como nombres de variables.

Tabla 1: Palabras reservadas

assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Algo importante para tener en cuenta es que las palabras reservadas no pueden ser utilizadas como nombres de variables, clases o métodos.

Las palabras reservadas `const` y `goto` son reservadas, estas no son utilizadas en la actualidad.

Tipos de datos

Los tipos de datos más simples en Java son: enteros, coma, flotante, boolean, lógicos y caracteres, en la siguiente tabla se muestran todos los tipos de datos en java:

Tabla 2: Tipos de datos en JAVA

		NOMBRE	TIPO	OCUPA	RANGO APROXIMADO
	TIPOS PRIMITIVOS	byte	Entero	1 byte	-128 a 127
		short	Entero	2 bytes	-32768 a 32767



TIPOS DE DATOS EN JAVA	(sin métodos; no son objetos; no necesitan una invocación para ser creados)	int	Entero	4 bytes	2*10 ⁹
		long	Entero	8 bytes	Muy grande
		float	Decimal simple	4 bytes	Muy grande
		double	Decimal doble	8 bytes	Muy grande
		char	Carácter simple	2 bytes	---
		boolean	Valor true o false	1 byte	---
	TIPOS OBJETO (con métodos, necesitan una invocación para ser creados)	Tipos de la biblioteca estándar de Java	String (cadenas de texto) Muchos otros (p.ej. Scanner, TreeSet, ArrayList...)		
		Tipos definidos por el programador / usuario	Cualquiera que se nos ocurra, por ejemplo, Taxi, Autobus, Tranvia		
		arrays	Serie de elementos o formación tipo vector o matriz. Lo consideraremos un objeto especial que carece de métodos.		
		Tipos envoltorio o wrapper (Equivalentes a los tipos primitivos, pero como objetos.)	Byte		
			Short		
			Integer		
			Long		
			Float		
			Double		
			Character		
			Boolean		



Sintaxis básica de java



Ilustración 2: Sintaxis básica de java

6.7.2. Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan la productividad cuando se desarrolla apps para Android, como las siguientes:

- ❖ Un sistema de compilación flexible basado en Gradle.
- ❖ Un emulador rápido y cargado de funciones.
- ❖ Un entorno unificado donde se puede desarrollar para todos los dispositivos Android.



- ❖ Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla.
- ❖ Integración con GitHub y plantillas de código para ayudar a compilar funciones de apps comunes y también importar código de muestra.
- ❖ Variedad de marcos de trabajo y herramientas de prueba.
- ❖ Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros.
- ❖ Compatibilidad con C++ y NDK.
- ❖ Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine. (Developers, 2021)

Estructura del proyecto

Cada proyecto de Android Studio incluye uno o más módulos con archivos de código fuente y archivos de recursos. Entre los tipos de módulos se incluyen los siguientes:

- ❖ Módulos de apps para Android
- ❖ Módulos de biblioteca
- ❖ Módulos de Google App Engine

De manera predeterminada, Android Studio muestra los archivos de tu proyecto en la vista de proyecto de Android, como se ve en la ilustración 3. Esta vista está organizada en módulos para que puedas acceder rápidamente a los archivos fuente clave de tu proyecto.

Puedes ver todos los archivos de compilación en el nivel superior de Secuencias de comando de Gradle y cada módulo de app contiene las siguientes carpetas:

- ❖ **manifests:** contiene el archivo AndroidManifest.xml.
- ❖ **java:** contiene los archivos de código fuente Java, incluido el código de prueba de JUnit.
- ❖ **res:** contiene todos los recursos sin código, como diseños XML, strings de IU e imágenes de mapa de bits.



La estructura del proyecto de Android en el disco difiere de esta representación plana.

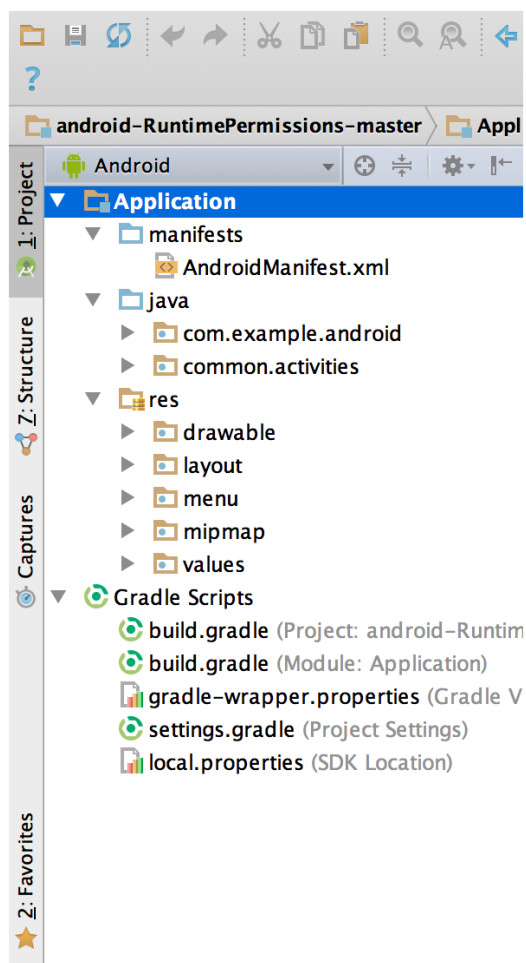


Ilustración 3: Los archivos de proyecto en la vista de Android



Interfaz de usuario: La ventana principal de Android Studio consta de varias áreas lógicas identificadas en la ilustración 4.

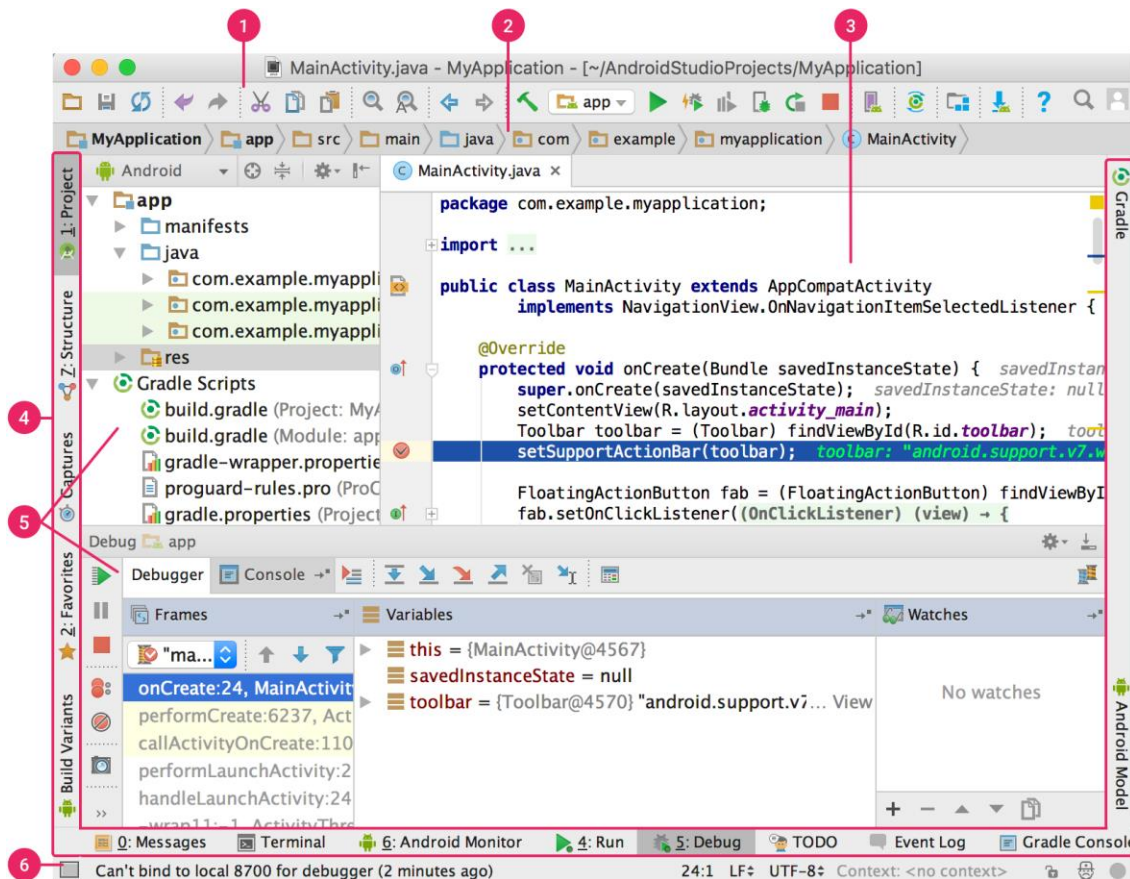


Ilustración 4: Interfaz de usuario

1. La barra de herramientas permite realizar una gran variedad de acciones, como ejecutar la app e iniciar las herramientas de Android.
2. La barra de navegación ayuda a explorar tu proyecto y abrir archivos para editar. Proporciona una vista más compacta de la estructura visible en la ventana Project.
3. La ventana del editor es el área en la que se puede crear y modificar código. Según el tipo de actividad actual, el editor puede cambiar. Por ejemplo, cuando se observa un archivo de diseño, el editor muestra el Editor de diseño.
4. La barra de la ventana de herramientas se encuentra afuera de la ventana del IDE y contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales.



5. Las ventanas de herramientas brindan acceso a tareas específicas, como la administración de proyectos, la búsqueda, el control de versiones, entre otras. Se pueden expandir y contraerlas.
6. En la barra de estado, se muestra el estado del proyecto y el IDE, además de advertencias o mensajes.

Sistema de compilación de Gradle

Android Studio usa Gradle como base del sistema de compilación, y el complemento de Android para Gradle proporciona capacidades específicas de Android. Este sistema de compilación se ejecuta en una herramienta integrada desde el menú de Android Studio, y lo hace independientemente de la línea de comandos. Puedes usar las funciones del sistema de compilación para lo siguiente:

- ❖ Personalizar, configurar y extender el proceso de compilación.
- ❖ Crear varios APK para tu app; diferentes funciones usan el mismo proyecto y los mismos módulos.
- ❖ Volver a usar códigos y recursos en conjuntos de archivos fuente.

Gracias a la flexibilidad de Gradle, se puede lograrlo sin modificar los archivos fuente de tu app. Los archivos de compilación de Android Studio se denominan `build.gradle`. Son archivos de texto sin formato que usan la sintaxis Groovy a fin de configurar la compilación con elementos que proporciona el complemento de Android para Gradle. Cada proyecto tiene un archivo de compilación de nivel superior para todo el proyecto y archivos de compilación de nivel de módulo independientes para cada módulo. Cuando se importa un proyecto existente, Android Studio genera automáticamente los archivos de compilación necesarios.

6.7.3. Cliente-Servidor

La arquitectura cliente servidor tiene dos partes claramente diferenciadas, por un lado, la parte del servidor y por otro la parte de cliente o grupo de clientes donde lo habitual es que un servidor sea una máquina bastante potente con un hardware y software



específico que actúa de depósito de datos y funcione como un sistema gestor de base de datos o aplicaciones. (Schiaffarino, 2019)

Componentes:

- ❖ **Red:** Una red es un conjunto de clientes, servidores y base de datos unidos de una manera física o no física en el que existen protocolos de transmisión de información establecidos.
- ❖ **Cliente:** El concepto de cliente hace referencia a un demandante de servicios, este cliente puede ser un ordenador como también una aplicación de informática, la cual requiere información proveniente de la red para funcionar.
- ❖ **Servidor:** Un servidor hace referencia a un proveedor de servicios, este servidor a su vez puede ser un ordenador o una aplicación informática la cual envía información a los demás agentes de la red.
- ❖ **Protocolo:** Un protocolo es un conjunto de normas o reglas y pasos establecidos de manera clara y concreta sobre el flujo de información en una red estructurada.
- ❖ **Servicios:** Un servicio es un conjunto de información que busca responder las necesidades de un cliente, donde esta información pueden ser mail, música, mensajes simples entre software, videos, etc.
- ❖ **Base de datos:** Son bancos de información ordenada, categorizada y clasificada que forman parte de la red, que son sitios de almacenaje para la utilización de los servidores y también directamente de los clientes.

Diferencia entre cliente y servidor

El cliente es un computador pequeño con una estructura al igual a la que tenemos en nuestras oficinas u hogares la cual accede a un servidor o a los servicios de este a través de Internet o una red interna. (Schiaffarino, 2019)

El servidor al igual que el cliente es una computadora, pero con diferencia de que tiene una gran capacidad que le permite almacenar gran cantidad de diversos de archivos, o correr varias aplicaciones en simultaneo para así nosotros los clientes poder acceder los servicios. (Schiaffarino, 2019)

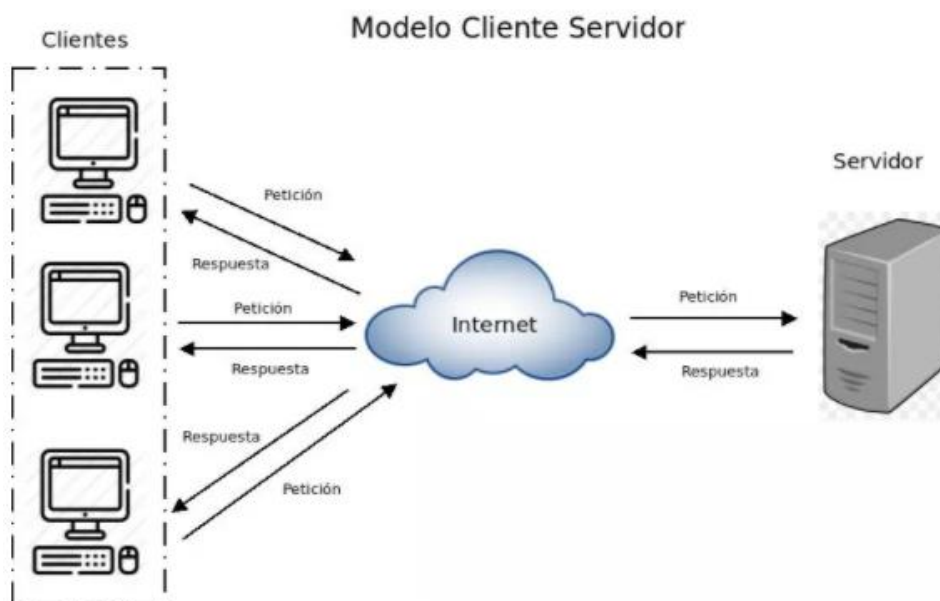


Ilustración 5: Esquema del modelo cliente servidor

Ejemplos de modelo cliente servidor

Algunos ejemplos de la arquitectura cliente servidor pueden ser:

- ❖ **Navegar una web:** Funciona basándonos en un cliente web (navegador) y un servidor web como Apache, Nginx o LiteSpeed
- ❖ **Protocolo FTP:** Funciona de idéntica forma, se utiliza un cliente de FTP (como Filezilla) para conectar a un servidor FTP (como Pure-FTPd, Proftpd, etc)
- ❖ **SSH:** Es idéntico también, se utiliza un cliente SSH para conectar al servidor SSH que corre en una red remota.
- ❖ **Juegos en red:** Existen clientes que permiten a jugadores online jugar desde sus casas conectándose a servidores de juegos remotos.
- ❖ **Sistema DNS:** El famoso servidor DNS interactúa con clientes DNS también, es decir, basa su arquitectura en el modelo cliente servidor
- ❖ **Servidor de Correo:** Donde los clientes consultan el correo al servidor de correo remoto, tanto desde móvil o una computadora de escritorio o laptop.



6.7.4. Firebase

Firebase de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo. (López, 2020)

Tabla 3: Funciones de Firebase

Funciones de Firebase	
Desarrollo	El primer grupo de funciones es conocido como Desarrollo o Develop en Firebase. Como su nombre indica, incluye los servicios necesarios para el desarrollo de un proyecto de aplicación móvil o web. Estos contribuyen a que el proceso sea más rápido, puesto que se dejan determinadas actividades a mano de Firebase, mientras que otras permiten optimizar diversos aspectos para conseguir la calidad deseada.
Realtime database	Una de las herramientas más destacadas y esenciales de Firebase son las bases de datos en tiempo real. Estas se alojan en la nube, son No SQL y almacenan los datos como JSON. Permiten alojar y disponer de los datos e información de la aplicación en tiempo real, manteniéndolos actualizados, aunque el usuario no realice ninguna acción.
Autenticación de usuarios	Firebase ofrece un sistema de autenticación que permite tanto el registro propiamente dicho (mediante email y contraseña) como el acceso utilizando perfiles de otras plataformas externas (por ejemplo, de Facebook, Google o Twitter), una alternativa muy cómoda para usuarios reacios a completar el proceso.



Almacenamiento en la nube	Firestore cuenta con un sistema de almacenamiento, donde los desarrolladores pueden guardar los ficheros de sus aplicaciones y sincronizarlos. Este almacenamiento es de gran ayuda para tratar archivos de los usuarios (por ejemplo, fotografías que hayan subido), que se pueden servir de forma más rápida y fácil. También hace la descarga de referencias a ficheros más segura.
Crash Reporting	Firestore ofrece Crash Reporting, que detecta y ayuda a solucionar los problemas de la app, consiguiendo un informe de errores muy detallado (con datos como el dispositivo o la situación en la que se da la excepción) y organizado, puesto que los agrupa por similitud y los clasifica por gravedad.
Test Lab	El Laboratorio de pruebas permite testear la app en dispositivos Android virtuales basados en los parámetros que configuremos. De esta forma, es mucho más sencillo detectar posibles errores antes de lanzar la aplicación.
Remote Config	La configuración remota sirve para modificar ciertas funciones, aspectos o incluso la apariencia de la aplicación sin que sea necesario publicar una actualización de la misma. De esta forma, no se requiere ningún tipo de acción por parte del usuario y se trata de cambios mucho más dinámicos.
Cloud Messaging	Su utilidad es el envío de notificaciones y mensajes a diversos usuarios en tiempo real y a través de varias plataformas.



Hosting	Firestore también ofrece un servidor para alojar las apps de manera rápida y sencilla, esto es, un hosting estático y seguro. Proporciona certificados de seguridad SSL y HTTP2 de forma automática y gratuita para cada dominio, reafirmando la seguridad en la navegación.
----------------	--



7. Diseño metodológico

7.1. Descripción general del proyecto

El diccionario es uno de los materiales o instrumentos clásicos, debidamente utilizado constituye un recurso valioso para el aprendizaje de otra lengua o el de buscar palabras cuyo significado no es conocido.

La búsqueda rápida, exacta y completa son las ventajas más provechosas de diccionarios electrónicos. En el presente trabajo se desarrolló una aplicación Android que será un diccionario digital lingüístico, la cual está destinada a proporcionar apoyo a las personas que no dominan los idiomas: español, inglés, miskito y mayangna.

La aplicación será personalizada y consta con cuatro opciones; en la primera sección se muestra el diccionario donde el usuario podrá buscar una palabra que desconozca en los otros idiomas, en la segunda sección los usuarios pueden ayudar enviando nuevas traducciones, en la tercera sección pueden visualizar su cuenta si iniciaron sesión en la aplicación, en la última sección tiene una pequeña descripción de la aplicación, la aplicación consta de una amplia base de datos donde se almacenan todas las traducciones.

7.2. Fases del proyecto

Etapas 1

En esta sección se toma en cuenta el problema que se presenta en la actualidad en las regiones de la Costa Caribe nicaragüense principalmente en las comunidades rurales de cada uno de los municipios, existe el problema para la comunicación entre las diferentes etnias, turistas nacionales y extranjeros que visitan la Costa Caribe, a los habitantes y turistas se les dificulta la comunicación ya que no dominan alguno de los idiomas: español, inglés, miskito y mayangna, lo que crea un obstáculo para el desarrollo en el comercio, el turismo, la cultura y la educación.



Etapas 2

En esta segunda etapa se decidió que software se utilizaría para el desarrollo de la aplicación Android, se escogió Android Studio como el IDE para programar y Firebase como base de datos en la nube, también se diseñó la plantilla que se utilizaría para el diseño final de la aplicación para que esta sea intuitiva y fácil de utilizar, también se consultaron diccionarios para elegir que palabras y expresiones se mostrarían en el diccionario digital lingüístico, se seleccionaron 770 traducciones las que sirven para un uso cotidiano.

Etapas 3

Se desarrolló el diccionario digital lingüístico para apoyar a personas que se les hace difícil comprender cualquiera de los idiomas: español, inglés, miskito, mayangna. Esta cuenta con una amplia base de datos de las traducciones de los idiomas antes mencionados. Esta aplicación pretende ser de gran utilidad para las personas habitantes de la Costa Caribe de Nicaragua, docentes y estudiantes de las diferentes escuelas y universidades, turistas nacionales y extranjeros, y para el público en general.

Diseño de interfaz gráfica

En este proyecto se desarrolló el diccionario digital lingüístico bajo la versión 2021.1.1 Bumblebee de Android Studio (*ver la tabla 4*), cuenta en la actualidad con múltiples servicios y herramientas para facilitar el desarrollo y gestión del proyecto y con un emulador para poder ejecutar la aplicación en diferentes versiones de Android.

La interfaz de la aplicación está compuesta por actividades y fragmentos con el uso de la plantilla que se muestra en la sección de “Desarrollo” fue posible el desarrollo de los distintos módulos que posee la aplicación. También la aplicación cuenta con una sección donde el usuario puede enviar nuevas traducciones que posteriormente se agregan a la base de datos principal por un administrador.



Tabla 4: Requisitos para instalar Android Studio

Requisitos para instalar Android Studio	
Windows	❖ Microsoft Windows 8/10 de 64 bits ❖ Arquitectura de CPU x86_64; Intel Core de segunda generación o más reciente, o CPU AMD con soporte para un hipervisor de Windows ❖ 8 GB de RAM o más ❖ 8 GB de espacio disponible en disco como mínimo (IDE + SDK de Android + Emulador de Android) ❖ Resolución de pantalla mínima de 1280 x 800
Mac	❖ MacOS 10.14 (Mojave) o superior ❖ Chips basados en ARM o Intel Core de segunda generación o más nuevos con soporte para Hypervisor. ❖ 8 GB de RAM o más ❖ 8 GB de espacio disponible en disco como mínimo (IDE + SDK de Android + Emulador de Android) ❖ Resolución de pantalla mínima de 1280 x 800
Linux	❖ Cualquier distribución de Linux de 64 bits que admita Gnome, KDE o Unity DE; Biblioteca GNU C (glibc) 2.31 o posterior. ❖ Arquitectura de CPU x86_64; Procesador Intel Core de segunda generación o posterior, o procesador AMD compatible con virtualización AMD (AMD-V) y SSSE3 ❖ 8 GB de RAM o más ❖ 8 GB de espacio disponible en disco como mínimo (IDE + SDK de Android + Emulador de Android) ❖ Resolución de pantalla mínima de 1280 x 800



Requerimientos de la aplicación Diccionario.

Para que la aplicación se ejecute en el dispositivo son necesario los siguientes requisitos mínimos:

- ❖ Almacenamiento de 35 MB de espacio disponible.
- ❖ Memoria RAM: 1 GB.
- ❖ Sistema Operativo Android: Lollipop (5.0) o superior.
- ❖ Conexión a internet.

Recursos del Hardware

Equipo hardware donde se desarrolló el proyecto y dispositivos Android donde se ensayó la aplicación.

Tabla 5: Recurso hardware

Ordenador				
Equipo	CPU	RAM	Almacenamiento	S. O
Lenovo ideapad 330s- 15IKB	Intel Core i5- 8250U 1.60 GHz	8.00 GB	1000 GB	Windows 11 Home 64 bits

Tabla 6: Dispositivos móviles y emuladores

Dispositivos móviles y emuladores Android			
Dispositivos	S. O	RAM	Almacenamiento
Samsung Galaxy j6	Android 10	2 GB	32 GB
Emulador Pixel 2	5.0 Lollipop	2 GB	8 GB
Emulador Pixel 2	6.0 Marshmallow	2 GB	8 GB
Emulador Pixel 2	7.1.1 Nougat	2 GB	8 GB
Emulador Pixel 2	8.0 Oreo	2 GB	8 GB
Emulador Pixel 2	9.0 Pie	2 GB	8 GB
Emulador Pixel 2	Android 11.0	2 GB	8 GB
Emulador Pixel 2	Android 12.0	2 GB	8 GB



8. Desarrollo

8.1. Plantilla utilizada para el diseño de la aplicación.



Ilustración 6: Vista bienvenida

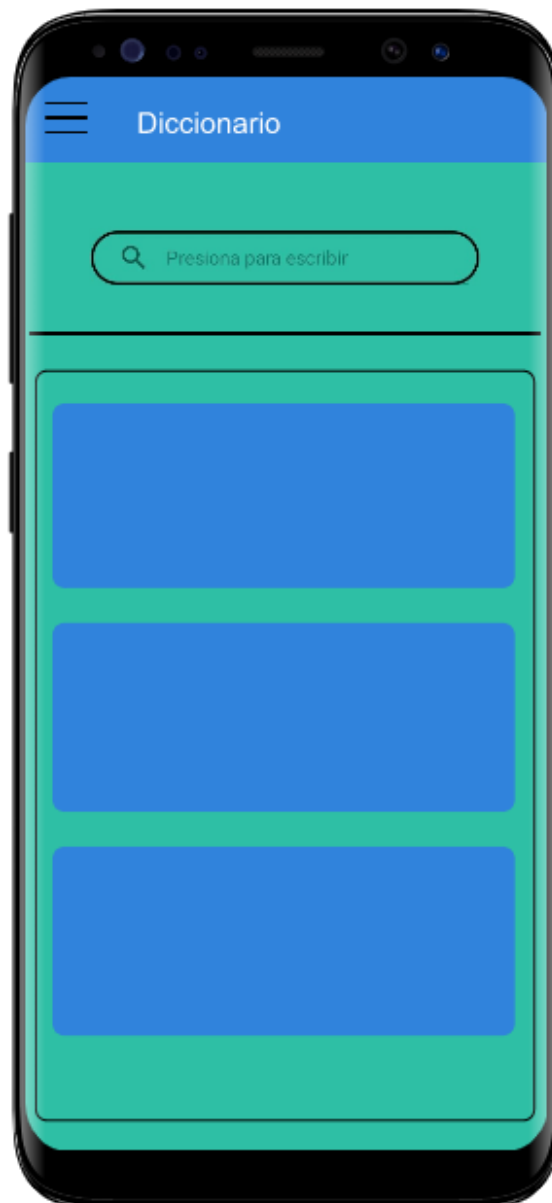


Ilustración 7: Vista principal

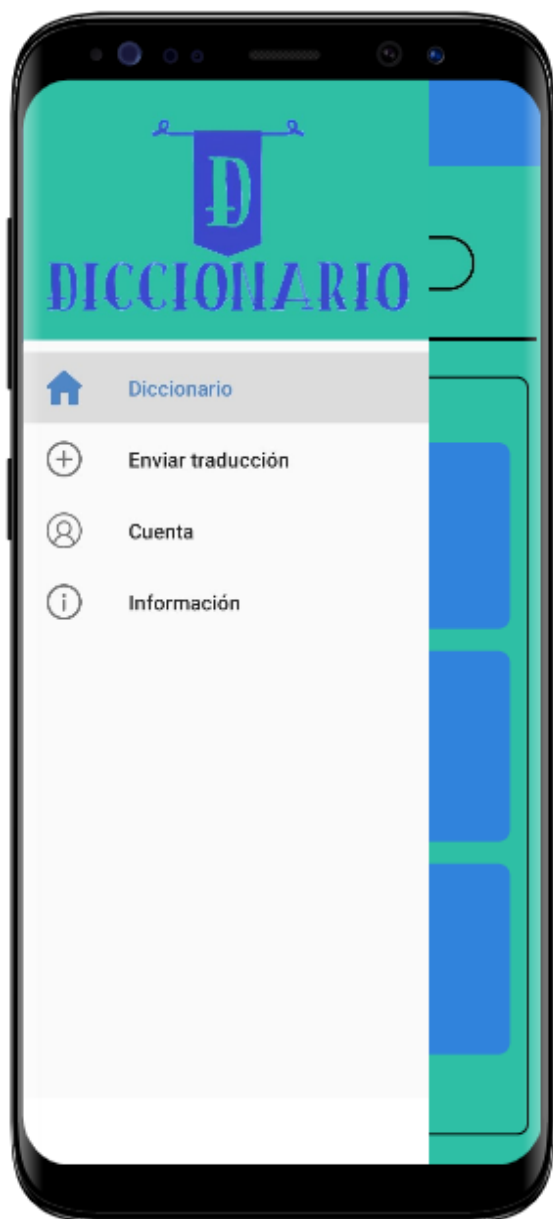


Ilustración 8: Menú desplegable



Ilustración 9: Vista de enviar traducción

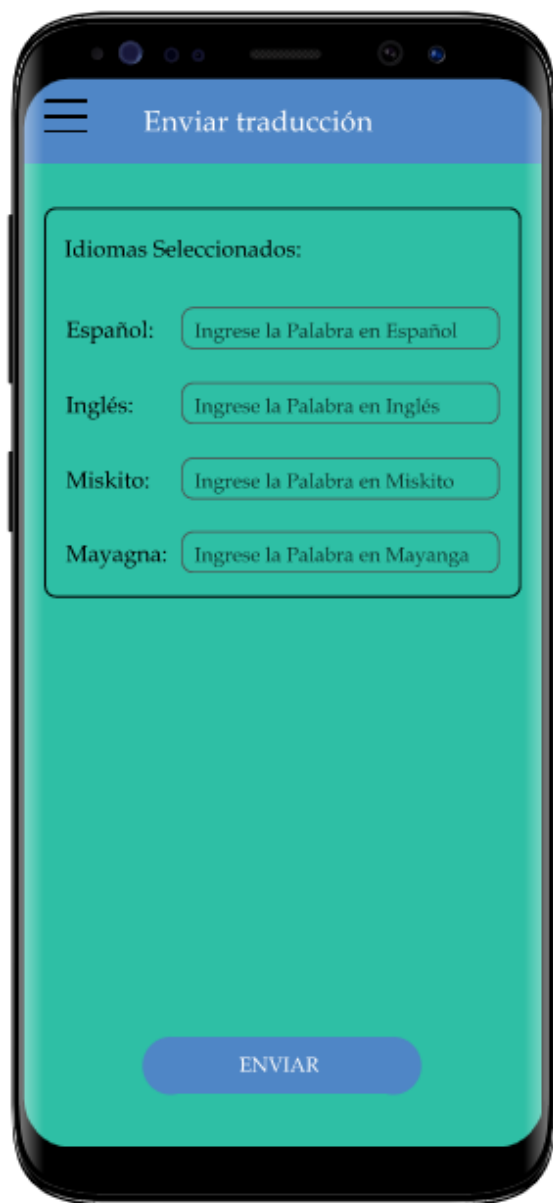


Ilustración 10: Vista idiomas seleccionados

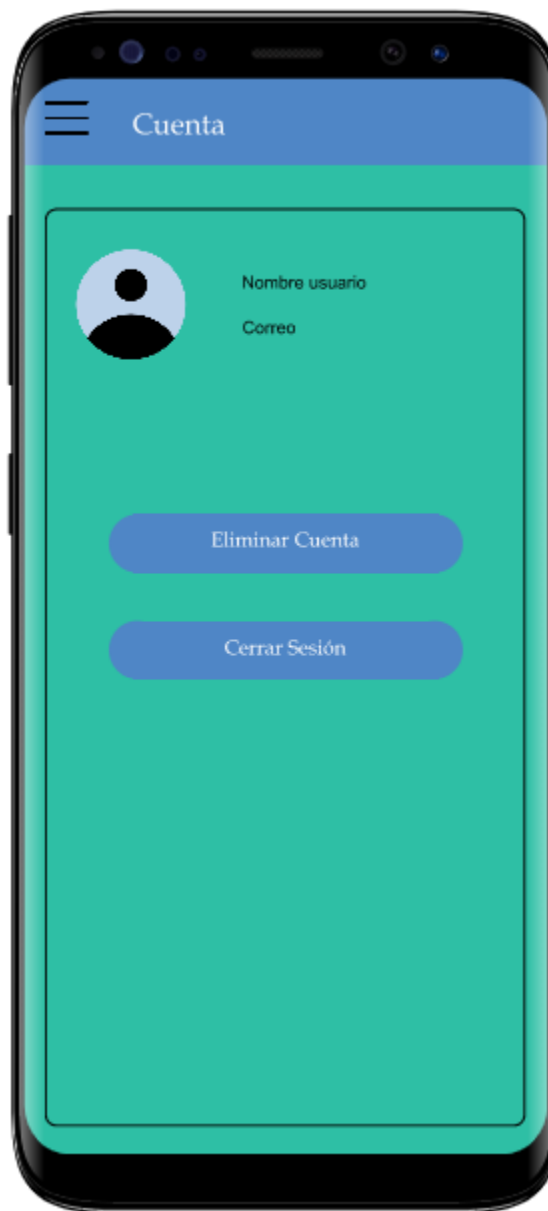


Ilustración 11: Vista cuenta



Ilustración 12: Vista información

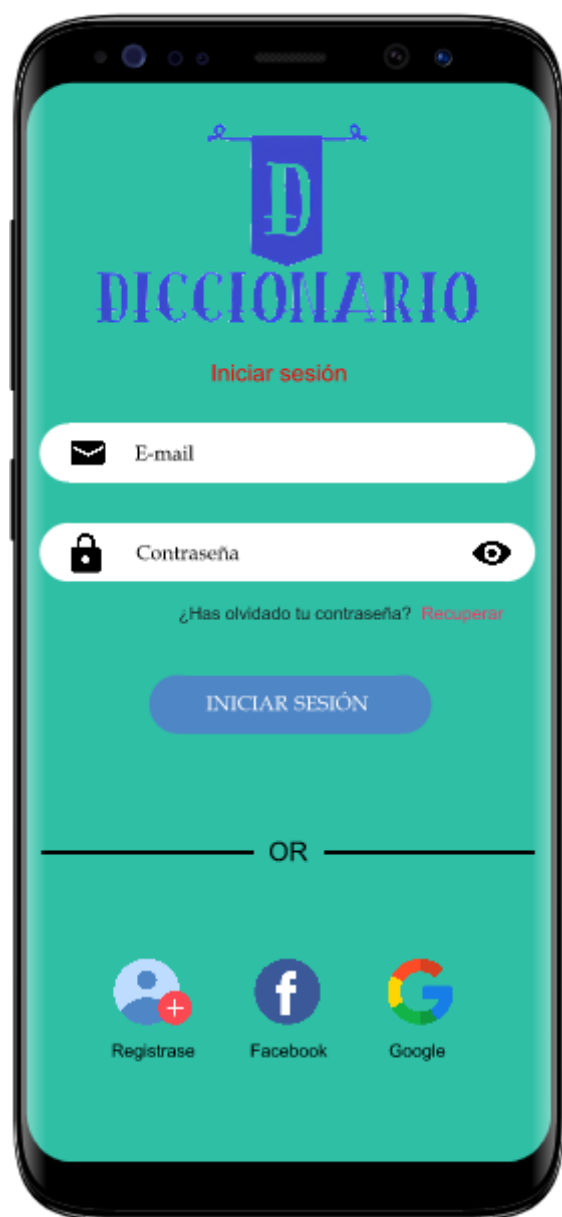


Ilustración 13: Vista iniciar sesión

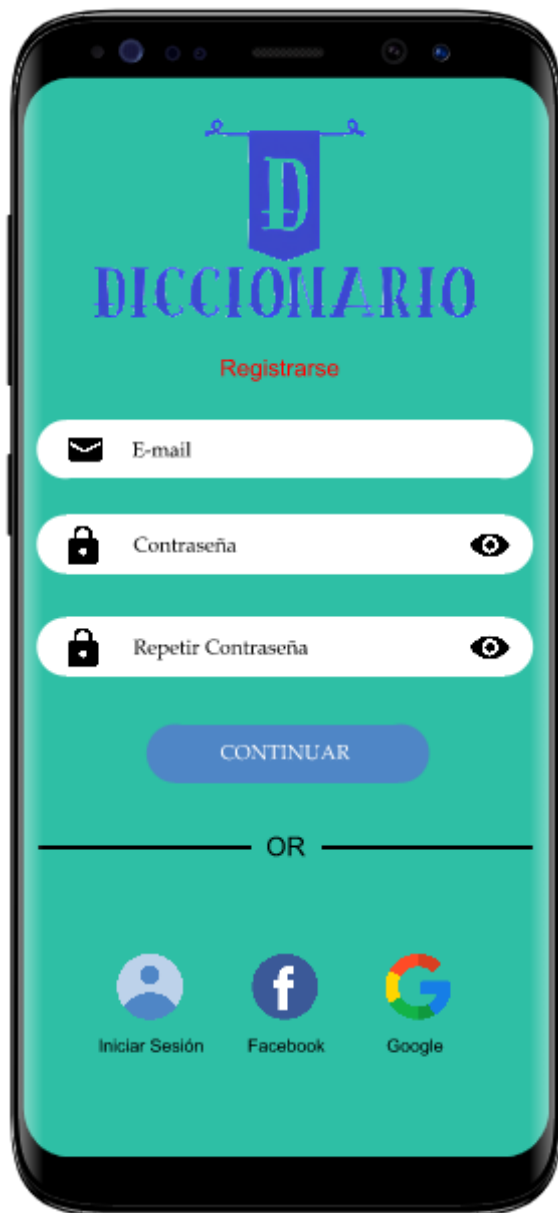


Ilustración 14: Vista registrarse



Ilustración 15: Vista recuperar contraseña



La siguiente lista describe los tipos de Layout utilizados en el proyecto:

- ❖ **LinearLayout:** Dispone los elementos en una fila o en una columna.
- ❖ **RelativeLayout:** Dispone los elementos en relación con otro o al padre.
- ❖ **ConstraintLayout:** Versión mejorada de RelativeLayout, que permite una edición visual desde el editor.
- ❖ **CoordinatorLayout:** Proporciona un nivel adicional de control sobre eventos táctiles entre vistas secundarias.

La siguiente lista describe los tipos de controles utilizados en el proyecto:

- ❖ **Searchview:** Proporciona una interfaz de búsqueda donde los usuarios pueden ingresar una consulta de búsqueda y luego enviar una solicitud al proveedor de búsqueda.
- ❖ **Button:** Este control es un simple botón.
- ❖ **ImageButton:** Tiene la misma funcionalidad que Button. La diferencia respecto a éste es que se le puede asignar una imagen cubriendo la superficie del ImageButton.
- ❖ **EditText:** Este control es la típica caja de texto que se emplea para la introducción de datos por teclado.
- ❖ **Chechbox:** Son casillas que se pueden marcar o desmarcar y que se utilizan para indicar que se ha seleccionado o deseleccionado una opción.
- ❖ **TextView:** Su única función es la mostrar texto en pantalla.
- ❖ **ImageView:** Este control no es más que una imagen establecida en una view.
- ❖ **RecyclerView:** Permite mostrar un listado de elementos.
- ❖ **View:** Representa el bloque de construcción básico para los componentes de la interfaz de usuario.
- ❖ **CardView:** Proporciona un diseño para tarjetas.
- ❖ **NavigationView:** Representa un menú de navegación estándar para la aplicación.
- ❖ **ScrollView:** permite albergar una jerarquía de views con el fin de desplazar su contenido a lo largo de la pantalla, cuando sus dimensiones exceden el tamaño de la misma.



9. Resultados

9.1. Recopilación de la información

Para obtener toda la información referente a las traducciones entre los idiomas de español, inglés, miskito y mayangna se consultaron diccionarios donde se proporcionaban las traducciones de las diferentes palabras y expresiones para una buena comunicación entre las personas que no dominan alguno de los idiomas anteriores, también se consultó a estudiantes de la UNAN-León que su lengua nativa es el miskito o el mayangna, para verificar la correcta escritura de las palabras que aparecen en los diccionarios consultados ya que estos presentaban errores ortográficos.

9.2. Construcción del diagrama

Se procedió a realizar el diagrama de procesos para definir cuál sería el flujo secuencial que seguiría la aplicación del diccionario: español, inglés, miskito y mayangna.

9.2.1. Casos de uso

A continuación, se muestra el diagrama de los casos de uso que muestra las funcionalidades del diccionario digital lingüístico y el acceso de los usuarios a cada una de ellas.

El usuario registrado puede consultar el diccionario, enviar traducción, iniciar sesión, ver su cuenta, eliminarla o cerrar la sesión y puede ver información de la aplicación. El usuario anónimo puede consultar el diccionario, registrarse y ver la información de la aplicación.



Acceso a las distintas funciones de la aplicación

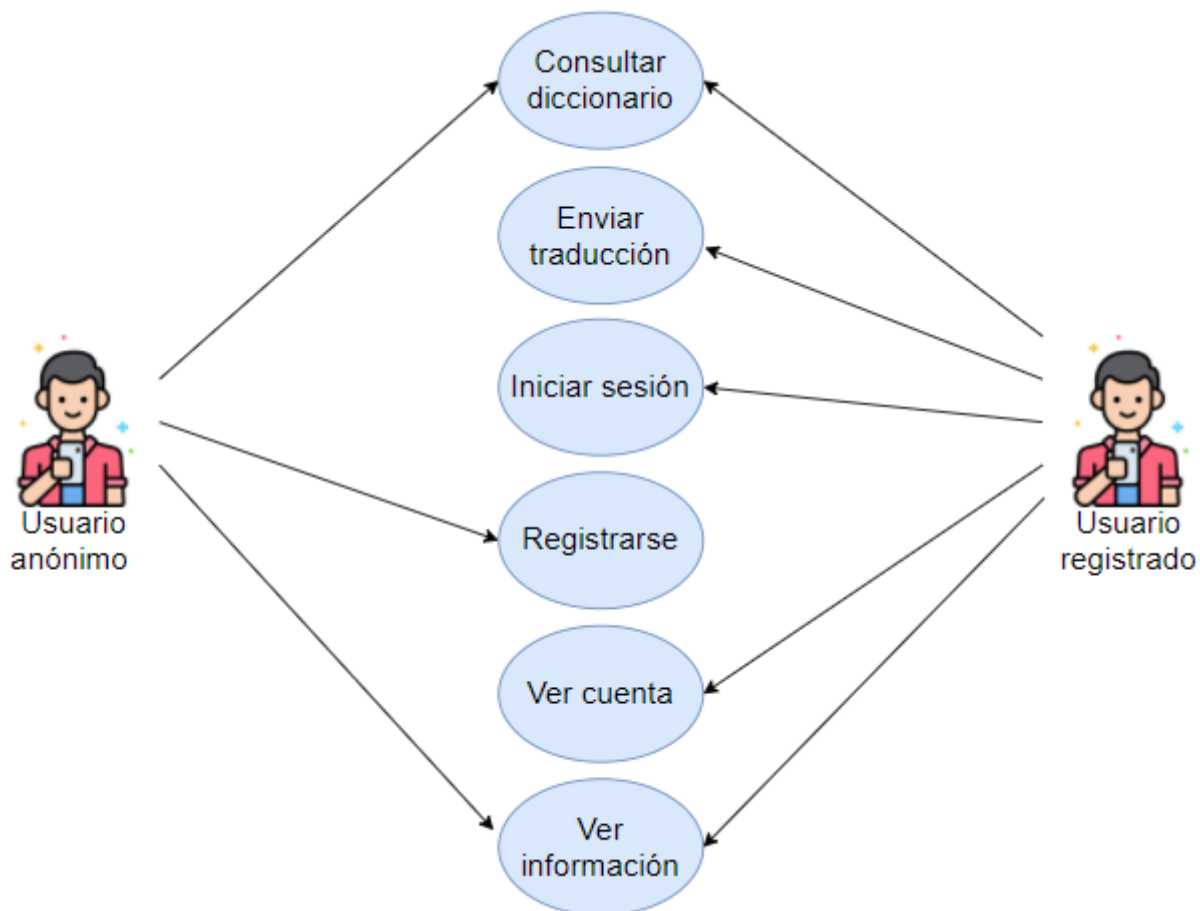


Ilustración 16: Caso de uso de las distintas funciones de la aplicación



9.2.2. Diagrama de las bases de datos

Base de datos principal “Diccionario”: Para el almacenamiento de las palabras y expresiones se eligió el servicio de Firebase “Realtime Database” la cual es una base de datos a tiempo real alojada en la nube con formato JSON.

Los datos almacenados se sincronizan a tiempo real con la aplicación. La comunicación es bidireccional, es decir que cuando se modifica o se agrega datos estos cambios se ven reflejados directamente en la web (consola de firebase) y en la aplicación “Diccionario”.

Esta base de datos cuenta con cinco campos de tipo string los cuales son:

- ❖ **id:** Identificador de los datos almacenados.
- ❖ **esp:** Campo donde se guarda la palabra o frase en español con su respectivo ejemplo de uso.
- ❖ **ing:** Campo donde se guarda la palabra o frase en inglés con su respectivo ejemplo de uso.
- ❖ **mis:** Campo donde se guarda la palabra o frase en miskito con su respectivo ejemplo de uso.
- ❖ **may:** Campo donde se guarda la palabra o frase en mayangna con su respectivo ejemplo de uso.

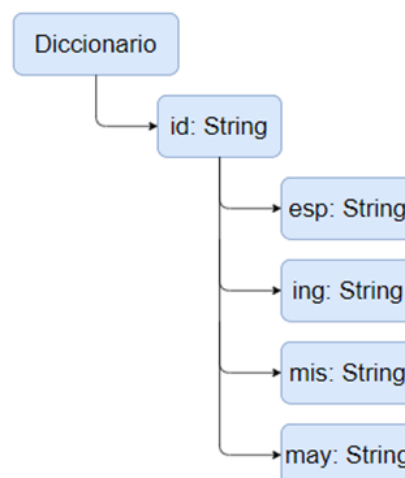


Ilustración 17: Estructura de la base de datos principal



Base de datos secundaria “traducciones_agregadas”: Para el almacenamiento de las traducciones enviadas por los usuarios se eligió el servicio de Firebase “Cloud Firestore” la cual es una base de datos NoSQL flexible, escalable y en la nube a fin de almacenar y sincronizar datos para el desarrollo tanto del lado del cliente como del servidor.

Esta base de datos cuenta con siete campos de tipo string los cuales son:

- ❖ **id:** Identificador de los datos almacenados.
- ❖ **correo:** Campo donde se guarda el correo del usuario que envió las frases o palabras.
- ❖ **fecha:** Campo donde se guarda la fecha que se enviaron las frases o palabras.
- ❖ **esp:** Campo donde se guarda la palabra o frase en español con su respectivo ejemplo de uso.
- ❖ **ing:** Campo donde se guarda la palabra o frase en inglés con su respectivo ejemplo de uso.
- ❖ **mis:** Campo donde se guarda la palabra o frase en miskito con su respectivo ejemplo de uso.
- ❖ **may:** Campo donde se guarda la palabra o frase en mayangna con su respectivo ejemplo de uso.

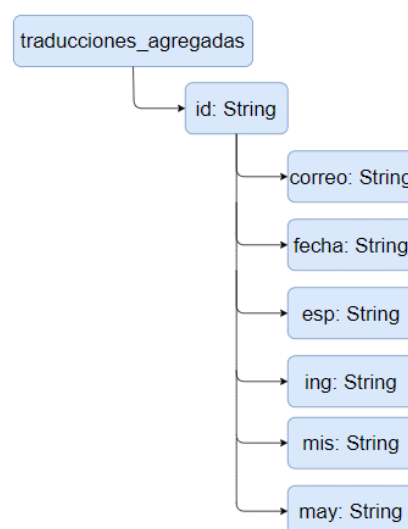


Ilustración 18: Estructura de la base de datos secundaria



9.2.3. Apartados de la aplicación

El diccionario digital lingüístico es una aplicación que no muestra grados de complicación, cuenta con distintas opciones de acuerdo con el tipo de usuario que la esté utilizando que pueden ser el anónimo o el registrado.

Apartado de diccionario: Donde se muestran las acciones principales de la aplicación, donde se traducen las palabras que busque el usuario entre los idiomas: español, inglés, miskito, mayangna; también se muestra un ejemplo de uso de la palabra o frase buscada en los cuatro idiomas, las traducciones realizadas se podrán copiar para que sean utilizadas donde lo desee el usuario.

Apartado de enviar traducción: Para poder ingresar a este apartado los usuarios tienen que iniciar sesión o registrarse para que puedan contribuir con la base de datos enviando nuevas traducciones de palabras y estas se guardarán en una base de datos secundaria para después procesarlas y agregarlas a la base de datos principal por un administrador manualmente.

Apartado de cuenta: En este apartado los usuarios tienen que iniciar sesión para poder entrar, y pueden eliminar la cuenta o cerrar la sesión de la aplicación.

Apartado de información: Donde se muestra que tipo de diccionario utiliza la aplicación.



9.2.4. Diagrama de clases

El siguiente diagrama de clases muestra cómo se relacionan las distintas clases del diccionario digital lingüístico.

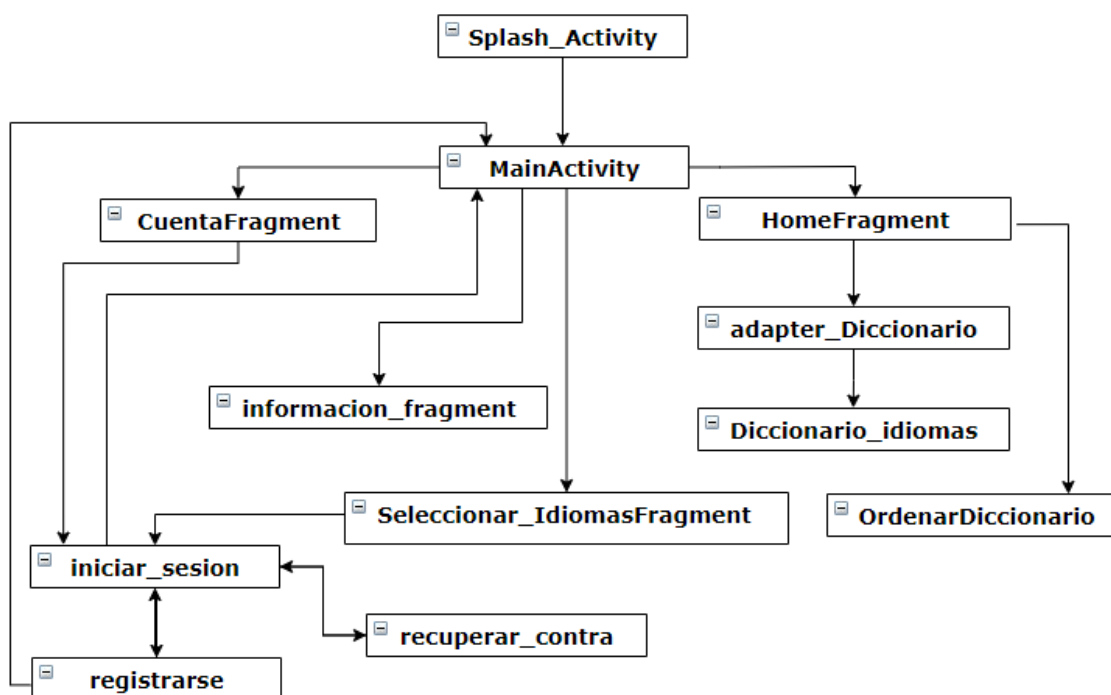


Ilustración 19: Diagrama de clases

9.2.5. Pruebas del funcionamiento de la aplicación

Un punto clave para comprobar el funcionamiento del diccionario digital lingüístico es la prueba de esta. Todas las pruebas realizadas por los usuarios se realizaron en un ambiente controlado, con el fin de no comprometer la información que está almacenada en la base de datos principal en Firebase. Los resultados fueron satisfactorios, durante los días que se realizaron las pruebas.

Todas las validaciones realizadas a la aplicación resultaron ser las apropiadas para los distintos tipos de usuarios ya sean los que iniciaron sesión y los que no. Cada una de las vistas de la aplicación tienen distintos controles que a su vez resultan ser intuitivos y de fácil uso.



Pruebas realizadas

- ❖ **Instalación y desinstalación:** La aplicación se instaló y desinstaló correctamente desde la API 21 (Lollipop) a la API 31 (Android 12) de Android.
- ❖ **Funcionalidad:** Todas las funciones programadas se ejecutaron correctamente en cada una de las vistas.
- ❖ **Usabilidad:** Los usuarios al interactuar con la aplicación lo hicieron de forma intuitiva y no se les dificultó utilizarla.
- ❖ **Interrupciones:** La aplicación soportó todo tipo de interrupciones de las otras aplicaciones.
- ❖ **Compatibilidad de hardware y software:** La aplicación es compatible con dispositivos Android que cuentan con 8GB de almacenamiento y 1GB de RAM como mínimo y desde las API 21 (Lollipop) a la API 31 (Android 12) tuvo una instalación y ejecución exitosa.
- ❖ **Conexión de red:** La aplicación mantuvo una conexión exitosa con los servicios de Firebase, Facebook y Google en todo el tiempo de ejecución de esta.
- ❖ **Rendimiento:** En la prueba de rendimiento no presentó problemas de: bloqueo de desplazamiento, latencia de inicio, transiciones sin interrupciones.



10. Conclusiones

Con el desarrollo de la aplicación móvil “Diccionario” que cuenta con los idiomas: español, inglés, miskito y mayangna, se concluye lo siguiente:

- ❖ Se eligió el software adecuado para desarrollar la aplicación y se estableció los requisitos mínimos para que esta funcionara correctamente en los dispositivos móviles.
- ❖ Se implementó la base de datos no relacional Realtime Database de Firebase para el almacenamiento de las palabras en la nube para que los usuarios tengan siempre acceso a las últimas palabras agregadas.
- ❖ Se seleccionaron con éxito las palabras y expresiones más usadas de los cuatros idiomas de tal forma que permitan una comunicación correcta en el uso cotidiano.
- ❖ Se creó una sección en la aplicación donde los usuarios que hayan iniciado sesión pueden enviar nuevas traducciones y así lograr que en la base de datos se esté agregando constantemente nuevas traducciones (previa aprobación por el administrador).
- ❖ Se desarrolló una aplicación móvil capaz de mostrar las diferentes traducciones entre los idiomas: español, inglés, miskito y mayangna; para facilitar la comunicación entre las personas que viven o visitan la costa caribe de Nicaragua. Después de consultar varios diccionarios y a estudiantes de la UNAN-León que su lengua nativa es el miskito o mayangna se eligió el Diccionario Cuatrilingüe Thuaka echo por la Fundación Tuahka para obtener las distintas palabras y frases que son mostradas por este diccionario digital lingüístico ya que este cumplía con lo que se quería mostrar en la aplicación. Esta aplicación se desarrolló usando tecnologías conocidas por muchos como: Android Studio, Java y Firebase, que facilitó toda la codificación de una manera ordenada y sencilla. Usando todas estas tecnologías como un solo conjunto permitió alcanzar el fin deseado, obteniendo como resultado la creación de una aplicación Android destinada a los usuarios que deseen consultar este diccionario.



11. Recomendaciones

Una vez concluido el presente trabajo monográfico, se considera tomar en cuenta lo siguiente:

Recomendaciones para el uso de la aplicación:

- ❖ Todo usuario que tenga acceso a la aplicación deberá tener conocimientos básicos sobre el uso de diccionarios digitales.
- ❖ Tener Android 5.0 (Lollipop) como mínimo en su dispositivo móvil.
- ❖ Para su uso se necesita conexión a Internet ya sea wi-fi o datos móviles.
- ❖ Para enviar nuevas traducciones el usuario tiene que crear una cuenta vinculada con la aplicación, o iniciar sesión con una existente, además de esto también puede iniciar sesión con Facebook o Google.
- ❖ Para una buena recopilación de datos el usuario deberá tener conocimientos de los idiomas en los que desea ayudar enviando nuevas traducciones ya que estas se revisan exhaustivamente por un administrador y evalúa si se agrega a la base de datos principal.

Recomendaciones para trabajos futuros:

- ❖ Se recomienda migrar la aplicación a ambiente web utilizando nuevas tecnologías.
- ❖ Se puede implementar una base de datos más robusta.
- ❖ Mejoramiento el aspecto visual de las distintas vistas de la aplicación.
- ❖ Para llegar a más usuarios, se recomienda agregar nuevas formas de iniciar sesión en la aplicación.
- ❖ Agregar un apartado donde los usuarios que iniciaron sesión puedan ver las palabras o frases que han enviado y poder editarlas o eliminarlas.
- ❖ Conocer el estado en que se encuentra alguna palabra o frase enviada por un usuario, como pueden ser: aprobada, rechazada, pendiente de aprobación.
- ❖ Al escribir una palabra a buscar dentro del diccionario, agregar la opción de autocompletado/sugerencias/similares mientras se está escribiendo.



- ❖ Traducir los textos de las distintas vistas en los idiomas inglés, miskito y mayangna.
- ❖ Adaptar las vistas de la aplicación a diferentes tamaños de pantallas y orientaciones.
- ❖ Verificar e indicar al usuario si la palabra que está enviando ya se encuentra agregada en el diccionario.
- ❖ Agregar las pronunciaciones de las palabras que son mostradas en el diccionario.
- ❖ Crear una interfaz web de administración para que el administrador seleccione las palabras enviadas por los usuarios para posteriormente agregarlas a la base de datos principal.



12. Bibliografía

- Clavero, D. (s.f.). *Google Sites*. Obtenido de <https://sites.google.com/site/usodeldiccionario/introduccion>
- Developers, G. (20 de Abril de 2021). *developer android*. Obtenido de <https://developer.android.com/studio/intro>
- EcuRed. (s.f.). *ecured.cu*. Obtenido de [https://www.ecured.cu/Miskitos_\(etnia_de_Honduras_y_Nicaragua\)](https://www.ecured.cu/Miskitos_(etnia_de_Honduras_y_Nicaragua))
- gobernanza, T. i. (2020). *territorioindigenaygobernanza.com*. Obtenido de https://www.territorioindigenaygobernanza.com/web/nnic_03/#:~:text=Es%20un%20pueblo%20transfronterizo%20que,%2C%20Belice%2C%20Guatemala%20y%20Honduras.&text=Su%20nombre%20deriva%20del%20una,ingl%C3%A9s%20creole%20y%20el%20castellano.
- IBM. (s.f.). *developer.ibm.com*. Obtenido de <https://developer.ibm.com/es/languages/java/tutorials/j-introtojava1/>
- Java. (s.f.). *java*. Obtenido de https://www.java.com/es/download/help/whatis_java.html#:~:text=Java%20es%20un%20lenguaje%20de,es%20r%C3%A1pido%2C%20seguro%20y%20fiable.
- JPimat. (19 de Marzo de 2018). *Como escribir bien*. Obtenido de <https://comoescribirbien.com/tipos-de-diccionarios/>
- López, S. (17 de mayo de 2020). *digital55*. Obtenido de <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- Morales, M. L., McCrea, M. T., & Gutiérrez Soza, N. (2009). APOORTE DE LA MUJER RAMA AL DESARROLLO IDENTITARIO DE LAS COMUNIDADES DE RAMA CAY Y LA ZOMPOPERA, BLUEFIELDS. *GÉNERO E IDENTIDAD*, 83.
- Mundial, B. (2011). *Análisis Social Programa de Desarrollo en la Costa Caribe y Alto Coco Banco Mundial – Departamento para el Desarrollo Internacional (Dfid)*.
- Patrón, T. (2007). Centro Cultural DITALYANG y Fundación TUAHKA. *Sahlai*, 22.
- QUEIDIOMA. (s.f.). *queidioma*. Obtenido de <https://queidioma.com/hablan-en-nicaragua/>
- Saldaña, R. B. (2002). *ecampus.uraccan.edu.ni*. Obtenido de <http://ecampus.uraccan.edu.ni/index.php/Caribe/article/view/218/1034>
- Schiaffarino, A. (12 de marzo de 2019). *infranetworking*. Obtenido de <https://blog.infranetworking.com/modelo-cliente-servidor/>



- Tuahka, F. (Mayo de 2014). *Diccionario Cuatrinlingüe Tuahka*. Región Autónoma del Atlántico Norte de Nicaragua, RAAN.: Bolonia Printing, Managua, Nic.
- Úbeda, Z. C. (2020). Situación actual del sustrato náhuatl en el español de Nicaragua. *Revista Lengua y Literatura* , 43.
- Villanueva Mejía, G. J., Sánchez Corea, D. M., & Frank, M. T. (febrero de 2019). Desarrollo de una app móvil para dispositivos con Sistema Operativo Android, que servirá como diccionario digital español-mayangna; en el periodo de octubre del 2018 a febrero del 2019. León, Nicaragua.
- Wise, J. (enero de 2021). *ripleybelieves*. Obtenido de <https://es.ripleybelieves.com/what-languages-are-spoken-in-nicaragua-1406>



13. Anexos

Cronograma de actividades

Actividades	Marzo 2021	Abril 2021	Mayo 2021	Junio 2021	Julio 2021	Agosto 2021	Septiembre 2021	Octubre 2021	Noviembre 2021
Antecedentes									
Planteamiento del problema									
Justificación									
Objetivos									
Marco Teórico									
Diseño Metodológico									
Desarrollo de la aplicación									
Conclusiones									
Recomendaciones									

Ilustración 20: Cronograma de activades



Interfaces de la aplicación

Vista de bienvenida: Muestra el título del diccionario y un mensaje de bienvenida a la aplicación.

Vista principal: Muestra una lista de palabras guardadas en la base de datos ordenada alfabéticamente. Contiene un SearchView para realizar una búsqueda en tiempo real en la lista, una vez que el usuario acceda a esta actividad se actualizará la lista del diccionario automáticamente obteniendo las nuevas palabras de Firebase.



Ilustración 21: Vista de bienvenida

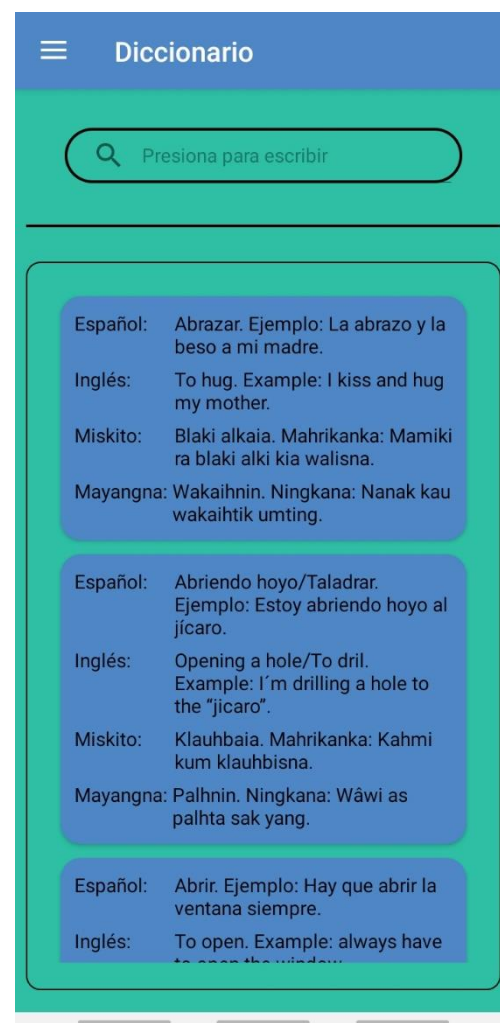


Ilustración 22: Vista principal



Búsqueda de palabras: Las palabras que busque el usuario se filtrarán y mostrarán si la palabra se encuentra en la base de datos.

Menú desplegable: Muestra los distintos apartados de la aplicación a los que los usuarios pueden acceder.

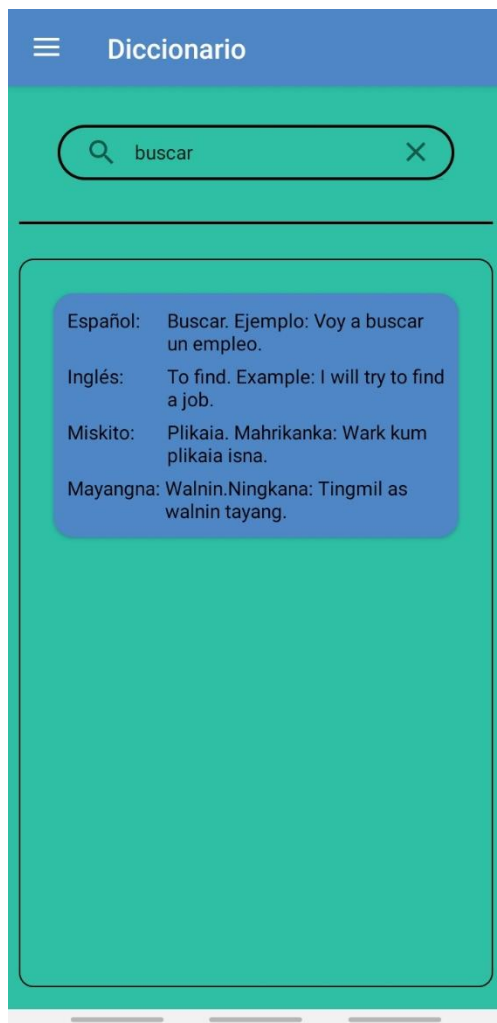


Ilustración 23: Búsqueda de palabras

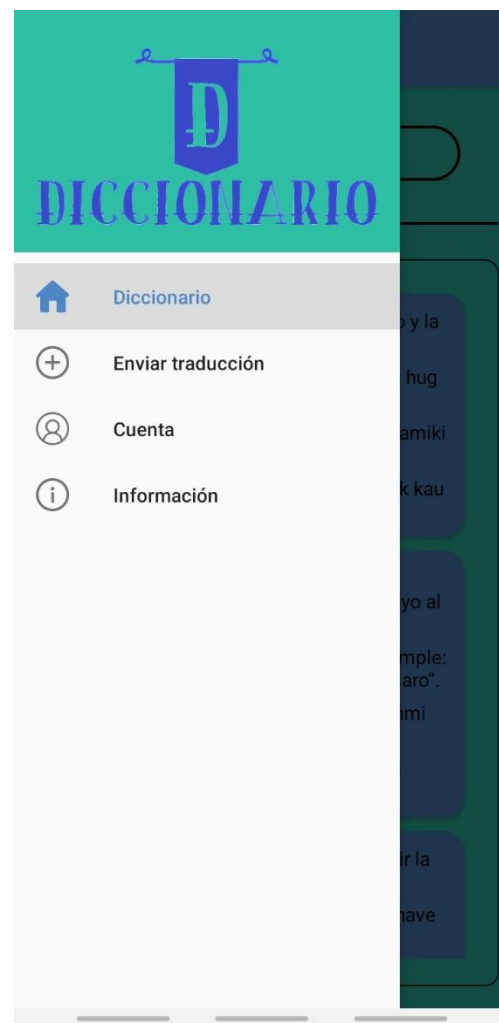


Ilustración 24: Menú desplegable



Vista idiomas seleccionados: Donde el usuario puede seleccionar los idiomas en los que quiere enviar nuevas traducciones.

Vista enviar traducción: Donde el usuario envía las traducciones de los idiomas que anteriormente seleccionó.

Esta es una interfaz de usuario para un sistema de traducción. El encabezado es una barra azul con un icono de menú a la izquierda y el título "Enviar traducción". El fondo principal es verde. Un recuadro blanco con borde negro contiene el texto "Seleccione los idiomas:" seguido de una lista de idiomas con casillas de verificación: Español, Inglés, Miskito y Mayangna. Todos los idiomas están seleccionados. En la parte inferior, hay un botón azul con el texto "CONTINUAR".

Ilustración 25: Vista idiomas seleccionados

Esta es una interfaz de usuario para un sistema de traducción. El encabezado es una barra azul con un icono de menú a la izquierda y el título "Enviar traducción". El fondo principal es verde. Un recuadro blanco con borde negro contiene cuatro secciones, cada una con un idioma y un campo de texto: "Español:" con "Ingrese la palabra en Español", "Inglés:" con "Ingrese la palabra en Inglés", "Miskito:" con "Ingrese la palabra en Miskito" y "Mayangna:" con "Ingrese la palabra en Mayangna". En la parte inferior, hay un botón azul con el texto "ENVIAR".

Ilustración 26: Vista enviar traducción



Vista iniciar sesión: Los usuarios podrán iniciar sesión con un correo y contraseña que este previamente registrada en la aplicación o con Facebook o Google, también posee un botón que lo envía al formulario de registro o al de recuperar contraseña.

Vista recuperar contraseña: El usuario puede recuperar su contraseña al ingresar el correo con el que creo una cuenta en la aplicación para que Firebase le envíe un correo para restablecer su contraseña.

Ilustración 27: Vista iniciar sesión

Ilustración 28: Vista recuperar contraseña



Vista registrarse: Los usuarios pueden registrarse con cualquier correo o pueden iniciar sesión con Facebook o Google, cuenta con un botón que lo envía a la vista de iniciar sesión.

Iniciar sesión con Facebook: Los usuarios pueden registrarse e iniciar sesión con su cuenta de Facebook que tengan vinculada a su dispositivo móvil.

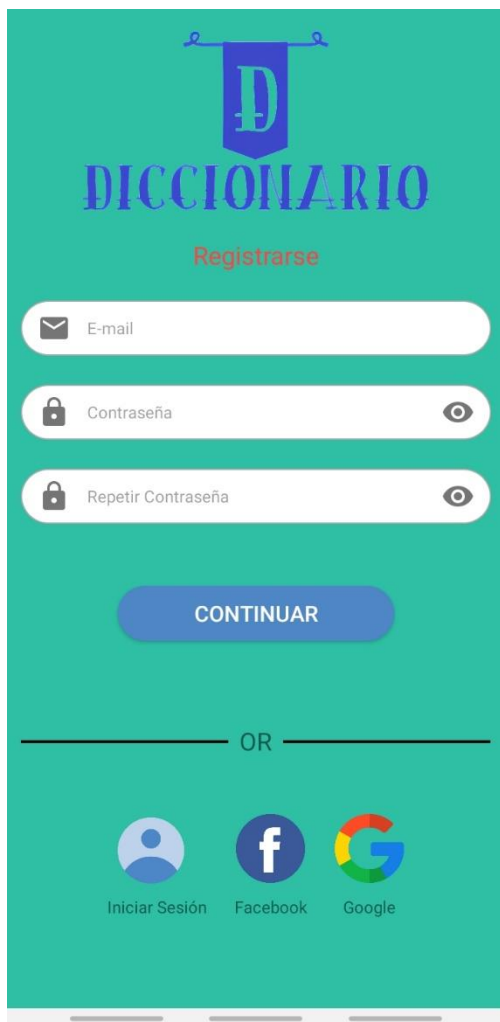


Ilustración 29: Vista registrarse

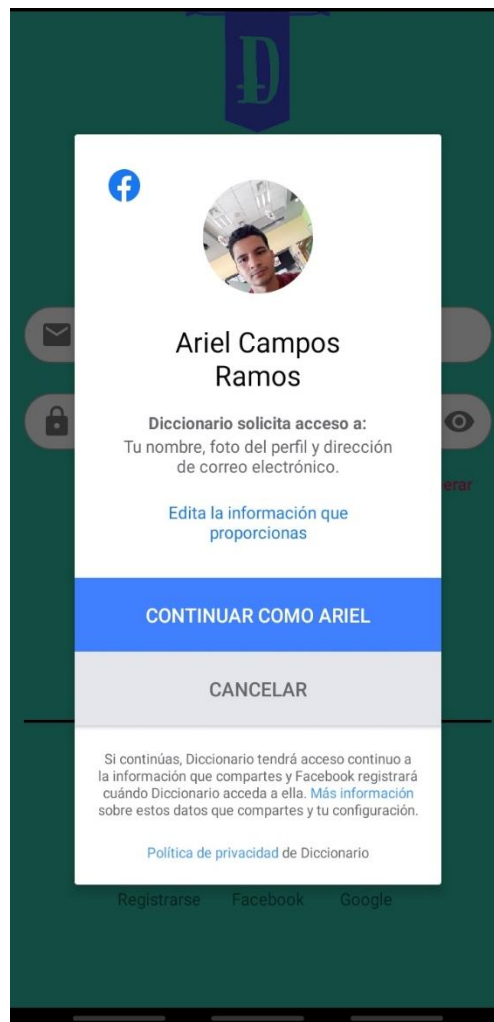


Ilustración 30: Iniciar sesión con Facebook



Iniciar sesión con Google: Los usuarios pueden registrarse e iniciar sesión con su cuenta de Google que tengan vinculada a su dispositivo móvil.

Vista cuenta: Se visualiza la foto de perfil ya sea de Facebook o Google, nombre de usuario y correo, cuenta con las opciones de eliminar cuenta o cerrar sesión en la aplicación.

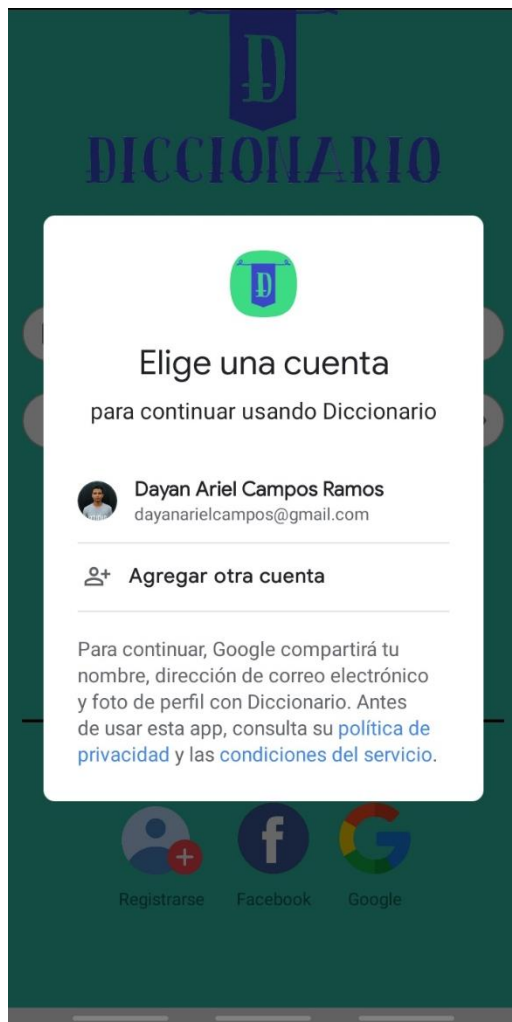


Ilustración 31: Iniciar sesión con Google

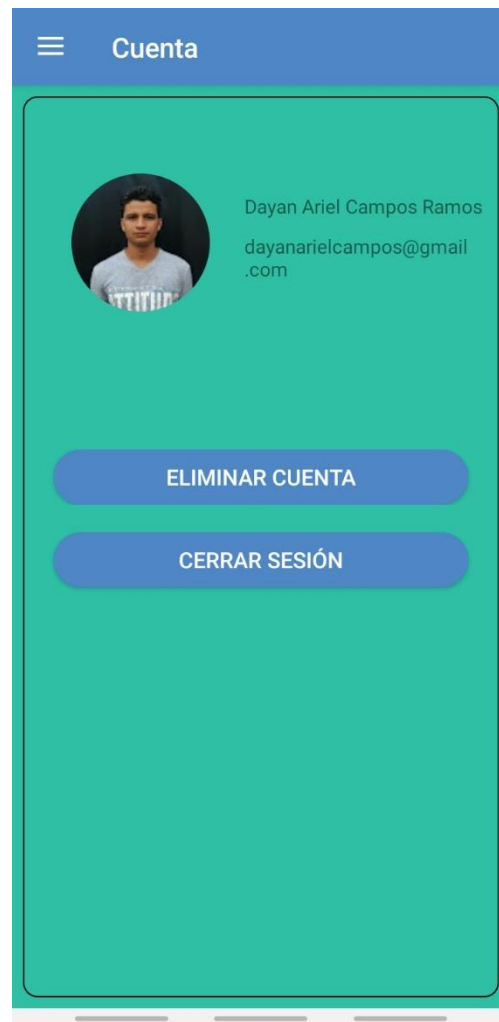


Ilustración 32: Vista cuenta



Vista información: En esta vista se explica que tipo de diccionario muestra la aplicación y porque algunas palabras no tienen una traducción equivalente al español y al inglés.

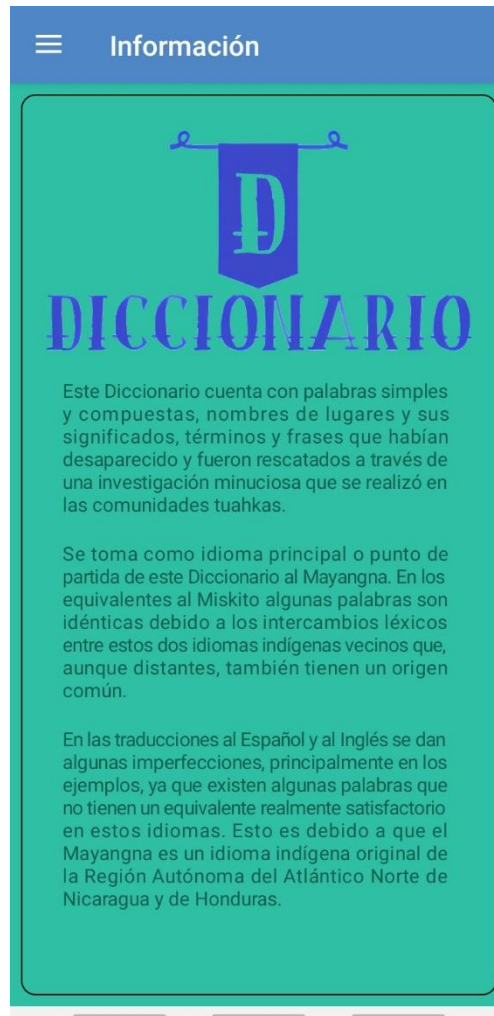


Ilustración 33: Vista información



Agregar Firebase al proyecto de Android

Requisitos previos

- Instala Android Studio o actualízalo a su versión más reciente.
- Asegúrese de que el proyecto cumpla con estos requisitos:
 - Se orienta al nivel de API 19 (KitKat) o superior.
 - Usa Gradle 4.1 o una versión posterior.
 - Utiliza Jetpack (AndroidX), que incluye el cumplimiento de los siguientes requisitos de versión:
 - `com.android.tools.build:gradle 3.2.1` o una versión posterior
 - `compileSdkVersion 28` o una versión posterior
- Configura un dispositivo físico o usa un emulador para ejecutar la aplicación. Hay que tener cuenta que los SDK de Firebase con una dependencia en Servicios de Google Play requieren que el dispositivo o el emulador tenga instalados los Servicios de Google Play.
- Acceder a Firebase con una Cuenta de Google.

Para conectar la aplicación Android a Firebase existen dos opciones:

- **Opción 1:** Usar el flujo de trabajo de configuración de Firebase console.
- **Opción 2:** Usa Firebase Assistant de Android Studio (es posible que requiera configuración adicional).

Opción 1: Agrega Firebase mediante Firebase console

Agregar Firebase a la aplicación implica realizar tareas en Firebase console y en el proyecto abierto de Android (por ejemplo, descargar archivos de configuración de Firebase desde la consola y transferirlos al proyecto de Android).



Paso 1: Crea un proyecto de Firebase

Antes de poder agregar Firebase a la aplicación para Android, se debe crear un proyecto de Firebase y conectarlo a la aplicación.

Crear un proyecto de Firebase

Paso 2: Registrar la aplicación con Firebase

Para usar Firebase en la aplicación para Android, se debe registrar la aplicación con el proyecto de Firebase. El registro de la aplicación a menudo se denomina “agregarla” al proyecto.

1. Dirigirse a Firebase console.
2. En el centro de la página de descripción general del proyecto, se hace clic en el ícono de **Android** o en **Agregar aplicación** para iniciar el flujo de trabajo de configuración.
3. Se Ingresa el nombre del paquete de la aplicación en el campo **Nombre del paquete de Android**.
4. Se Ingresa otra información de la aplicación como el **sobrenombre** y el **certificado de firma SHA-1 de depuración** (*opcional*).
5. Dar clic en **Registrar aplicación**.

Paso 3: Agrega un archivo de configuración de Firebase

1. Agregar el archivo de configuración de Firebase para Android a la aplicación, como se indica a continuación:
 - a. Dar clic en **Descargar google-services.json** a fin de obtener el archivo de configuración de Firebase para Android (google-services.json).
 - b. Transferir el archivo de configuración al directorio del módulo de la aplicación.



2. Agregar el complemento de google-services a los archivos Gradle a fin de habilitar los productos de Firebase en la aplicación.
 - a. Agregar las reglas para incluir el complemento de servicios de Google al archivo Gradle (build.gradle) de nivel de raíz (a nivel de proyecto). Además, revisar que este el repositorio Maven de Google.

```
buildscript {  
    repositories {  
        google()  
    }  
    dependencies {  
        classpath 'com.google.gms:google-services:4.3.5'  
    }  
}  
allprojects {  
    repositories {  
        google()  
    }  
}
```

- b. En el archivo Gradle (generalmente app/build.gradle) del módulo (a nivel de aplicación), aplicar el complemento Gradle de los servicios de Google:

```
apply plugin: 'com.android.application'  
apply plugin: 'com.google.gms.google-services'  
android {  
}
```

Paso 4: Agrega los SDK de Firebase a tu app

1. Usar la BoM de Firebase para Android y declarar las dependencias de los productos de Firebase que se quiere usar en la aplicación. Declarar en el **archivo Gradle (generalmente app/build.gradle) del módulo (nivel de aplicación)**.



Si Analytics está habilitada

```
dependencies {  
    implementation platform('com.google.firebase:firebase-bom:27.0.0')  
    implementation 'com.google.firebase:firebase-analytics'  
    implementation 'com.google.firebase:firebase-auth'  
    implementation 'com.google.firebase:firebase-firestore'  
}
```

Si Analytics está inhabilitada

```
dependencies {  
    implementation platform('com.google.firebase:firebase-bom:27.0.0')  
    implementation 'com.google.firebase:firebase-auth'  
    implementation 'com.google.firebase:firebase-firestore'  
}
```

2. Sincronizar la aplicación para garantizar que todas las dependencias tengan las versiones necesarias.

Opción 2: Agregar Firebase mediante Firebase Assistant

Firebase Assistant registra la aplicación con un proyecto de Firebase y agrega los archivos, los complementos y las dependencias necesarios de Firebase al proyecto de Android, todo desde Android Studio.

1. Abrir el proyecto de Android en Android Studio y accede a Firebase Assistant:
 - a. Ir a **Archivo > Buscar actualizaciones** para asegurarte de estar usando las versiones más recientes de Android Studio y Firebase Assistant.
 - b. Ir a **Herramientas > Firebase** para abrir el panel *Asistente*.
2. Escoger un producto de Firebase para agregarlo a la aplicación. Expandir su sección y hacer clic en el vínculo del instructivo (por ejemplo, **Analytics > Registra un evento de Analytics**).



- a. Hacer clic en **Conectarse a Firebase** para conectar el proyecto de Android con Firebase.
 - b. Hacer clic en el botón para agregar un producto de Firebase deseado (por ejemplo, **Agrega Analytics a la aplicación**).
3. Sincronizar la aplicación para garantizar que todas las dependencias tengan las versiones necesarias.
4. En el panel de Firebase Assistant, seguir las instrucciones de configuración restantes para el producto de Firebase que se seleccione.
5. Agrega todos los demás productos de Firebase que se quiera mediante Firebase Assistant.



Código para consultar y obtener la base de datos del servidor de Firebase, esta se guardar en un ArrayList la cual se ordenará alfabéticamente para después pasarla a un RecyclerView y se muestren por pantalla todas las palabras y frases con sus ejemplos.

```
ref = FirebaseDatabase.getInstance().getReference().child("Diccionario");
ref.keepSynced(true);
ref.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if (snapshot.exists()) {
            for (DataSnapshot snap : snapshot.getChildren()) {
                Diccionario_idiomas di = snap.getValue(Diccionario_idiomas.class);
                list.add(di);
            }
            OrdenarDiccionario.ordenar(list);
            adapter.notifyDataSetChanged();
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});
```



Función para buscar en tiempo real las palabras o frases con sus ejemplos en el ArrayList, busca coincidencias en los cuatros idiomas al mismo tiempo, las búsquedas pueden ser escritas en mayúsculas o minúsculas y no afectara el filtrado en la lista.

```
private void buscar(String text) {  
    ArrayList<Diccionario_idiomas> milista = new ArrayList<>();  
    for (Diccionario_idiomas obj: list){  
        if(obj.getEsp().toLowerCase(Locale.ROOT).contains(text.toLowerCase(Locale.ROOT)) ||  
            obj.getIng().toLowerCase(Locale.ROOT).contains(text.toLowerCase(Locale.ROOT)) ||  
            obj.getMis().toLowerCase(Locale.ROOT).contains(text.toLowerCase(Locale.ROOT)) ||  
            obj.getMay().toLowerCase(Locale.ROOT).contains(text.toLowerCase(Locale.ROOT))){  
                milista.add(obj);  
            }  
        }  
        adapter_Diccionario adapter = new adapter_Diccionario(milista);  
        recyclerView.setAdapter(adapter);  
    }  
}
```



Ejemplo de la estructura del archivo JSON que se realizó para almacenar todas las palabras y frases en los cuatros idiomas para posteriormente subir el archivo a Firebase y sea consultado por la aplicación móvil.

```
{
  "Diccionario" : {
    "001" : {
      "esp" : "Solamente éste. Ejemplo: Mi madre solamente este hijo varón tiene.",
      "ing" : "Only this. Example: Mi mother has this only son.",
      "may" : "Adik manh. Ningkana: Nanak adik manh mim al dûwih.",
      "mis" : "Naha baman. Mahrikanka: Mamiki naha baman luhpa waitna brisa."
    },
    "002" : {
      "esp" : "Ésta/Éste. Ejemplo: ¿Ésta es tu hija Yaring?",
      "ing" : "This. Example: Is this your daughter Yaring?",
      "may" : "Adiyah. Ningkana: Adiyah man mimim wana Yaring?",
      "mis" : "Naha sa. Mahrikanka: Naha sa man luhpiam mairin Yaring?"
    },
    "003" : {
      "esp" : "Quejarse. Ejemplo: La persona se queja cuando tiene dolor en el cuerpo.",
      "ing" : "Complain. Example: The person complains when the body aches.",
      "may" : "Ahnin. Ningkana: Muih kidi muinh ranghwa puyun ahwi.",
      "mis" : "Ahbaia. Mahrikanka: Upla ba ai wina klauhwi pyuara ahbisa."
    }
  }
}
```



Vista desde Firebase de los métodos de iniciar sesión habilitados, con los cuales los usuarios pueden iniciar sesión en la aplicación “Diccionario”.

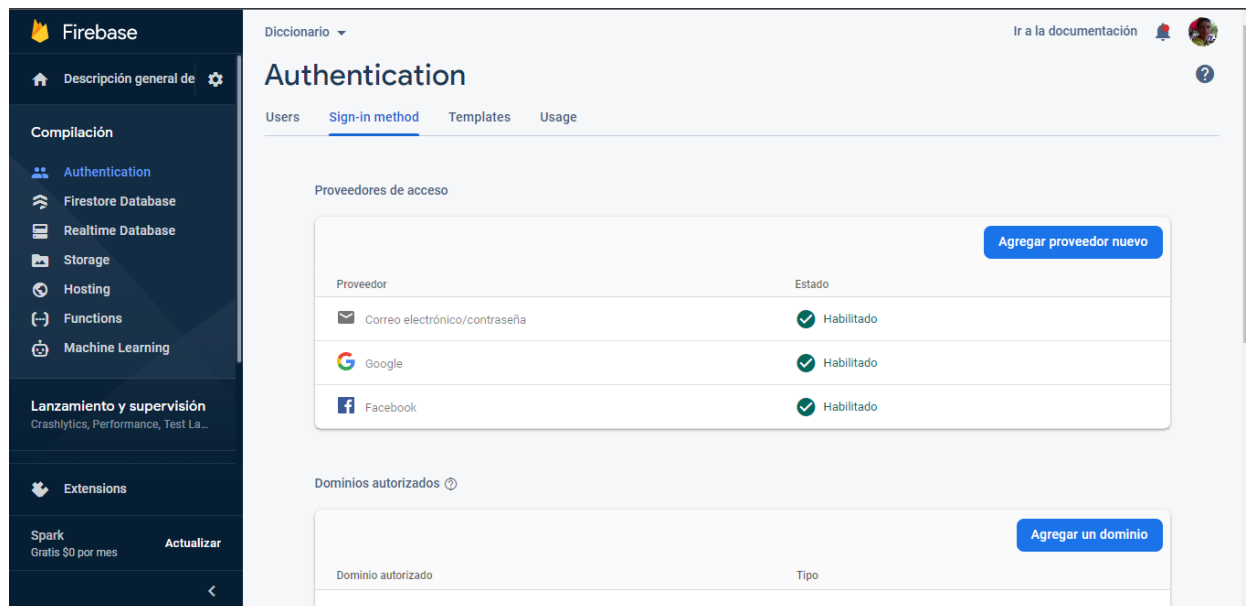


Ilustración 34: Métodos habilitados para el inicio de sesión

Vista desde Firebase de los usuarios registrados en la aplicación “Diccionario”, donde el administrador puede: restablecer la contraseña, inhabilitar o borrar las cuentas.

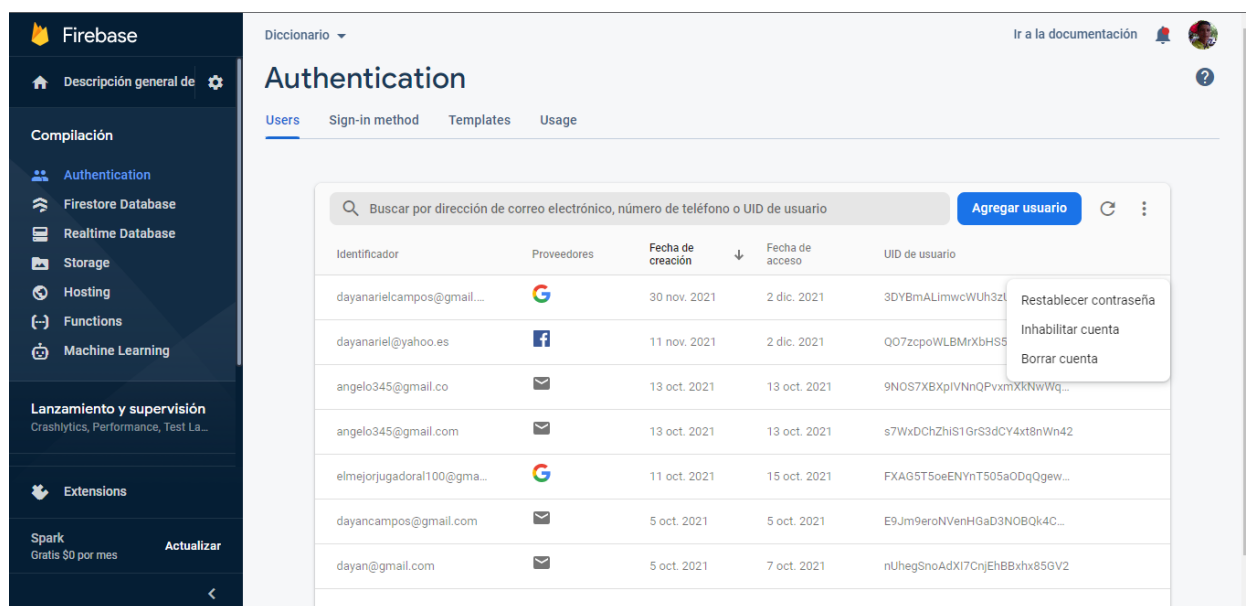


Ilustración 35: Usuarios registrados



Vista desde Firebase de la base de datos principal, se utiliza el servicio de Realtime Database para el almacenamiento de las distintas traducciones.



Ilustración 36: Base de datos principal

Vista desde Firebase de la base de datos secundaria, se utiliza el servicio de Cloud Firestore para el almacenamiento de las traducciones que envíen los usuarios.

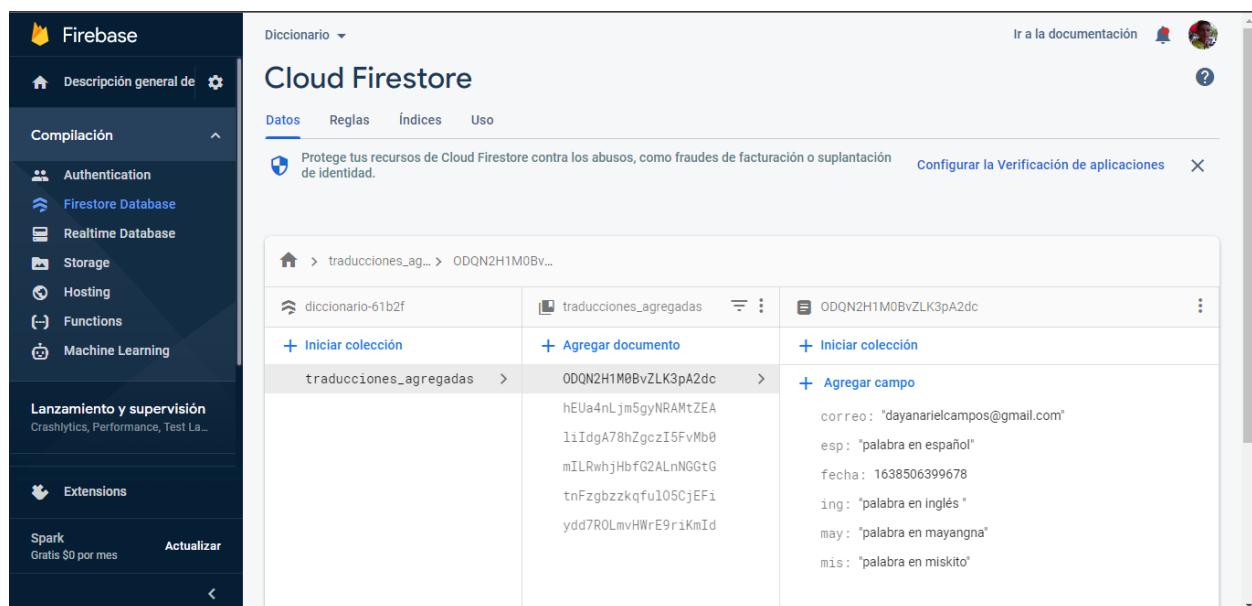


Ilustración 37: Base de datos secundaria