UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENIERÍA Y ARQUITECTURA

Escuela Profesional de Ingeniería de Sistemas

Aplicación móvil para el registro automático de rutas y lugares visitados durante las experiencias turísticas, basado en el paradigma offline first

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Autor:

Rosbel Neftali Ccana Yucra

Asesor:

Mg. Roel Dante Gomez Apaza

Co-Asesor:

Dra. Sc. Nélida Gladys Maquera Sosa

Juliaca, julio de 2019

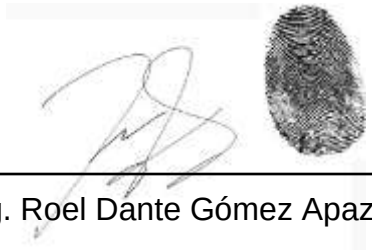
DECLARACIÓN JURADA DE ORIGINALIDAD DE TESIS

Yo Mg. Roel Dante Gómez Apaza, docente de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que la presente investigación titulada: **"APLICACIÓN MÓVIL PARA EL REGISTRO AUTOMÁTICO DE RUTAS Y LUGARES VISITADOS DURANTE LAS EXPERIENCIAS TURÍSTICAS, BASADO EN EL PARADIGMA OFFLINE FIRST"** del autor **Rosbel Neftali Ccana Yucra**, tiene un índice de similitud de 15% verificable en el informe del programa Turnitin, y fue realizada en la Universidad Peruana Unión bajo mi dirección.

En tal sentido asumo la responsabilidad que corresponde ante cualquier falsedad u omisión de los documentos como de la información aportada, firmo la presente declaración en la ciudad de Juliaca, a los 10 días del mes de junio del año 2026.



Mg. Roel Dante Gómez Apaza
Asesor

LIMA - PERU

044

ACTA DE SUSTENTACIÓN DE TESIS

En Puno, Juliaca, Villa Chullunquiani, a, dos .....dia(s) del mes de julio .....del año 20 19 siendo las 11:30 horas, se reunieron en el Salón de Grados y Títulos de la Universidad Peruana Unión, Filial Juliaca, bajo la dirección del Señor Presidente del jurado: Mg. Jhonson Henry Centurión Tulca ....., el secretario: Ing. Angel Rosendo Condori Choquera ..... y los demás miembros: Ing. David Mamani Pari .....y el asesor: Mg. Roel Dante Gómez Apaza ..... con el propósito de administrar el acto académico de sustentación de la tesis titulada: Aplicación móvil para el registro automático de rutas y lugares visitados durante las experiencias turísticas, basado en el paradigma offline first. .....de el(los)/la(las) bachiller/es: a) Rosbel Neftali Ecana Yucra .....b) .....conducente a la obtención del título profesional de Ingeniero de Sistemas .....

(Nombre del Título Profesional)

con mención en.....

El Presidente inició el acto académico de sustentación invitando al (los)/a(la)(las) candidato(a)/s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por el(los)/la(las) candidato(a)/s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado.

Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato (a): Rosbel Neftali Ecana Yucra

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
Aprobado	16	B	Bueno	Muy Bueno

Candidato (b): .....

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) *Ver parte posterior*

Finalmente, el Presidente del jurado invitó al(los)/a(la)(las) candidato(a)/s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

Presidente

Secretario

Asesor

Miembro

Miembro

Candidato/a (a)

Candidato/a (b)

ÁREA TEMÁTICA

Área: Ingeniería y Tecnología.

Sub área: Ingeniería Eléctrica, Electrónica e Informática

Disciplina: Ingeniería de Sistemas y Comunicaciones

DEDICATORIA

Este trabajo se la dedico principalmente a Dios por su compañía y bendición divina.

A mi familia por ser el motor para alcanzar mis logros, en especial a mi mamá **Lidia Yucra Cordova** por ser ese apoyo incondicional en cada etapa de mi vida, por motivarme en todo momento a que siga con mis sueños sin importar mis errores y enseñarme la valentía de seguir adelante a pesar de las adversidades.

AGRADECIMIENTOS

Primeramento doy gracias a Dios por protegerme durante mi camino y darme fuerzas para superar obstáculos y dificultades.

A mis padres por su apoyo incondicional, cuidado y ejemplo me ha inculcado que los anhelos y metas con esfuerzo y dedicación se llegan alcanzar.

A mi asesor de tesis, Mg. Roel Dante Gomez Apaza por brindarme su conocimiento, por su guía en el desarrollo de este proyecto con disposición, confianza y apoyo profesional.

Un profundo agradecimiento a Dra.Sc. Nélida Gladys Maquera Sosa por el apoyo y colaboración en el desarrollo de este proyecto, como coordinadora del proyecto de investigación “Plataforma Digital Inteligente y Big Data para el Turismo Rural Comunitario en la Región de Puno”

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURA

ÍNDICE DE ANEXOS

ABREVIATURAS Y ACRÓNIMOS

- API: Interfaz de programación de aplicaciones
- GPS: Sistema de Posicionamiento Global
- OMT: Organización Mundial de Turismo

RESUMEN

El objetivo del presente trabajo de investigación es desarrollar una aplicación móvil para el registro automático de rutas y lugares visitados durante las experiencias turísticas, el enfoque de desarrollo está basado en offline-first. Para lograr dicho objetivo se hizo uso de tecnologías tales como Nativescript, Vue.js y Firebase, porque están orientadas a aquellos desarrolladores que se dedican mayormente al desarrollo web. Con todo esto la aplicación se desarrolló bajo la metodología ágil, dentro del marco de scrum. La aplicación móvil TourSteps está destinada a dispositivos iOS y Android, con un soporte offline y protocolo de resolución de conflictos de sincronización de firebase; se incorporó el api de mapbox para trazar y visualizar el recorrido de un usuario en el mapa, con indicadores que muestran las experiencias vividas en un lugar determinado. Así mismo la aplicación en cuestión realiza registros automáticos de puntos (coordenadas) por el que un usuario recorre con su respectivo nombre del lugar, para lograr esto se incorporó la API de geocoding de Google Maps.

Palabras claves:

Desarrolló movíl, Nativescript, Nativescript-vue, Vue.js, Scrum, nativescript-mapbox, nativescript-firebase, offline-first,

CAPÍTULO I. El problema

1.1. Descripción del problema

En la actualidad la tecnología se ha vuelto parte de la vida cotidiana de las personas y en especial de la generación de los millennials o generación Y, quienes están muy involucrados con las nuevas tecnologías que les ayude optimizar su tiempo, ahorrar dinero y sobre todo que sean fáciles de usar. Las aplicaciones móviles son una de sus herramientas principales porque son simples de usar, dinámicas, cumplen tareas específicas, brindan información relevante y les ayudan a solucionar problemas de la vida diaria (Fombona, Pascual y Madeira como se citó en Hurtado, 2016).

En la investigación realizada por (DITENDRIA, 2018) menciona que: "En 2017 se descargaron 178,1 miles de millones de aplicaciones móviles y se espera que en 2022 la cifra asciende a 258,2 miles de millones de descargas". El crecimiento interanual del número de sesiones que se realizaron en el mundo durante el 2017 se fija en un 6%, la figura 1 nos muestra las evaluaciones de las descargas de aplicaciones móviles a nivel mundial.

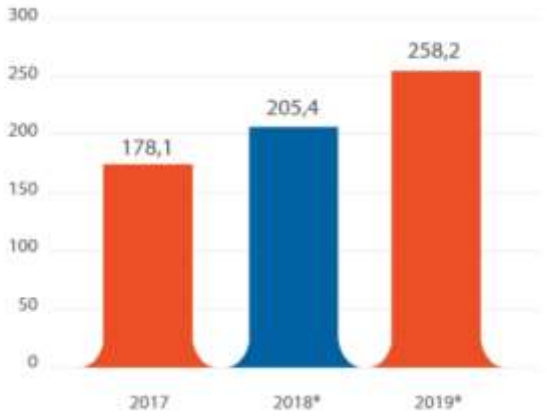


Figura 1 Evaluaciòn de las descargas de aplicaciones móvil a nivel mundial, la evaluación se realizó en descargas anuales en miles de millones
Fuente: DITENDRIA, (2018) Informe Mobile en España y en Mundo

Hoy en día las personas cuando usan el celular, dedican su tiempo en un 90% a las aplicaciones móviles y solo un poco más del 10% de su tiempo que pasamos los dedicamos al navegador, una referencia en latinoamérica tenemos a Argentina y Brasil como se puede observar en la gráfica 2, donde Argentina es el país que más tiempo invierte en el uso aplicaciones móviles (95%) frente a los navegadores (DITRENDIA, 2018).

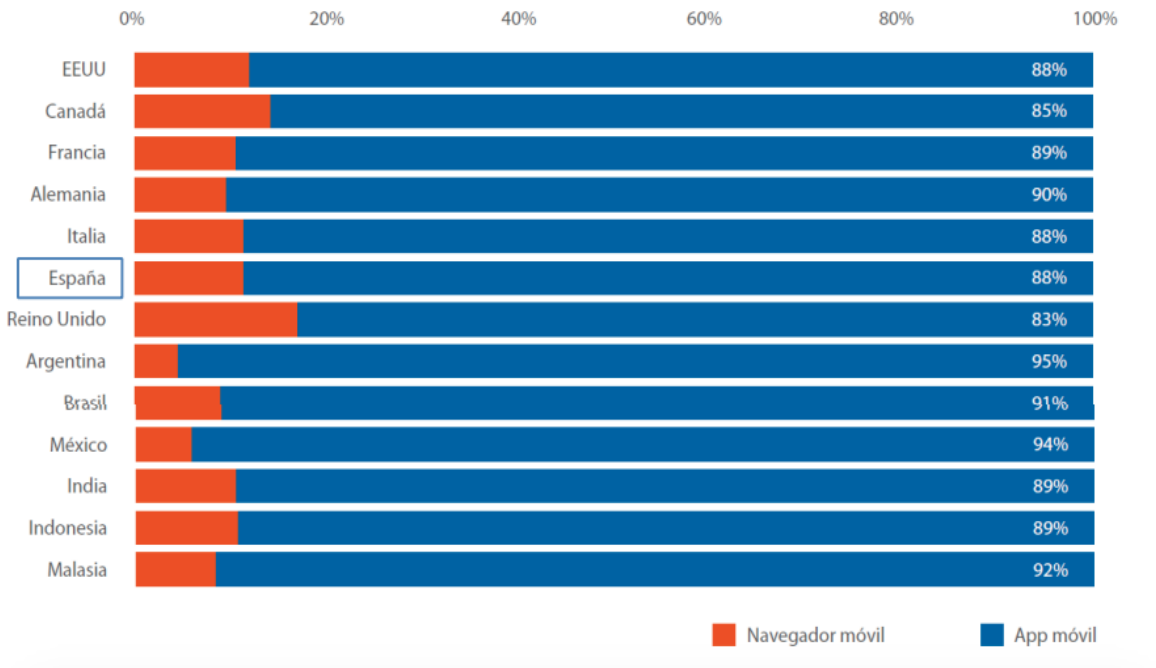


Figura 2. Participación de minutos móviles total por navegador/aplicación, gráfica editado por DITENDRIA a partir de datos de Cosmcore
Fuente: DITENDRIA, (2018) Informe Mobile en España y en Mundo

Como se pudo observar la mayoría de las personas invertimos el tiempo en el uso de aplicaciones móviles en el teléfono, también hay un grupo de personas que deseamos tener el registro de cada actividad que realizamos, ahora si se trata de un viaje que va ser una experiencia única hacemos todo lo posible para registrar cada momento. Y para esto también existen aplicaciones móviles, orientadas al turismo que dan de alguna forma soluciones a estos inconvenientes, pero la exigencia de tener un aplicativo que automatice el registro de experiencias de momentos únicos durante los viajes cada vez es mayor. Fuller (como se citó en D'Angelo, 2015) menciona: "Los tiempos cambian y las preferencias y gustos de los turistas también. A medida que el turismo se generalizó como una opción viable para amplias poblaciones, los viajeros diversificaron sus intereses y se volvieron más exigentes con respecto a la calidad de la experiencia".

Los millennials son los que usan a diario las aplicaciones móviles, este segmento de mercado crece de manera rápida y cada vez tiene mayores exigencias en sus viajes, tienen la necesidad de contar con una amplia variedad de herramientas que le brinden informaciòn detallada de la situaciòn actual de sus destinos, que la informaciòn esté actualizada y que enumere una lista de actividades a realizar durante el viaje, ya que les gusta tomar riesgos y

probar distintas situaciones y lo más importante compartir cada experiencia mediante las redes sociales y otros medios (Hurtado, 2016).

Generación Millennials o Generación Y, segùn Ipsos, (como se citò en Hurtado, 2016) menciona: “los Millennials suelen ahorrar un tercio de la parte de sus ingresos y su ingreso ha aumentado respecto al año anterior por lo que puede gastar en entretenimiento y mantener un equilibrio en su vida diaria. Están muy ligados a la tecnología, usan frecuentemente internet para informarse, buscan compartir experiencias a través de redes sociales, blogs y cualquier medio informativo”. Los millennials son muy buenos referentes al momento de querer obtener informaciòn, ya que ellos van registrando y compartiendo cada experiencia vivida.

En la actualidad el uso de las aplicaciones móviles nos facilita realizar actividades como: acceso a datos de geolocalización, muestran informaciòn de nuestros puntos de interés, permiten trazar nuestro recorrido en mapas, ayudan a reservar habitaciòn en hoteles y entre otras. Todo esto gracias a que los dispositivos de hoy en dìa estàn implementadas con las últimas tecnologìas como son: GPS integrado, acelerómetro, giroscopio, sensor de proximidad, sensor de luz, cámara frontal, cámara posterior, conexión Wi-Fi, conexión a redes de datos, y otras funcionalidades que podemos usar en las distintas aplicaciones (Robles, 2015).

Sin embargo, el uso de estas características no se realiza de la manera óptima cuando se implementa en un aplicativo. La mayoría de las aplicaciones que integran estas herramientas muchas veces son un obstáculo para el turista al momento de registrar su experiencia, porque se tiene que seguir muchos pasos para lograr el objetivo. Como turistas nuestra prioridad es tener experiencias y no estar registrando, pero a la vez requerimos tener registros cada lugar donde tuvimos una experiencia.

Viendo todas estas dificultades que los turistas tienen se planteó el problema: ¿Como el registro automático de rutas y lugares visitados mediante una aplicación móvil con conectividad offline first mejorará, la calidad del registro de experiencias durante el viaje turístico?

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar una aplicación móvil para el registro automático de rutas y lugares visitados durante las experiencias turísticas, basado en el paradigma offline first.

1.2.2. Objetivo Específico

- Definir los requerimientos funcionales y no funcionales para el desarrollo de la aplicación móvil
- Implementar el registro automático de lugares mediante geolocalización por GPS y la visualización en tiempo real del recorrido turístico con Mapbox.
- Realizar pruebas de funcionamiento offline.

1.3. Justificación

1.3.1. Justificación Social

Twain (2018) menciona: “Dentro de veinte años estarás más decepcionado por las cosas que no hiciste que por las que hiciste. Así que suéltate de las ataduras y navega lejos del puerto seguro. Atrapa los alisios en tus velas. Explorar. Sueño. Descubrir”. Twain es un viajero que le gusta registrar cada experiencia vivida, y en la actualidad existe muchos turistas como él. Es por tal motivo que nuestra aplicación permitirá al turista disfrutar más sus viajes, pero a la vez tener un registro detallado de cada lugar donde estuvimos.

También está enfocado para aquellos viajeros solitarios que inician un viaje sin ningún plan, solo deseando disfrutar cada experiencia. El aplicativo les ayudará en ubicar fácilmente puntos de interés como pueden ser: hoteles, museos, parques o atractivos turísticos, restaurantes entre otros, gracias al sistema de georreferenciación que tendrá implementado; lo más importante el usuario que en este caso son los turistas principalmente tendrán una ruta registrada automáticamente de todo su recorrido, sin que se den cuenta con todos las experiencias captadas mediante fotos, reacciones y otros; al final permitiéndole tener un álbum de recuerdos de cada experiencia vivida.

1.3.2. Justificación Económica

La aplicación a desarrollar tiene como un objetivo crear un álbum de recuerdos de las rutas recorridas por el turista, aquí se plantea tener un álbum básico sin costo, pero para aquellos usuarios que deseen tener el álbum personalizado con más contenido tendrá un costo de acuerdo a sus páginas. De esta forma tendremos un recuerdo de todo lo que hicimos de manera online o impreso y a la vez generamos ingreso para nuestro país.

Por otra parte, las herramientas a usar para el desarrollo de la aplicación son de costo cero, el cual no generará gastos adicionales.

1.3.3. Justificación Tecnológica

En el presente proyecto nos planteamos utilizar la última tecnología para el desarrollo de nuestro aplicativo móvil, a pesar de que existe aplicaciones móviles similares a la que nos planteamos desarrollar, Becerra, Silva y Rocha (2012) menciona: "La innovación va más allá de la creación de algo excepcionalmente nuevo. El factor más importante para analizar el proceso de innovación es entender qué nuevas características existentes se incorporan a un producto, proceso, servicio o idea, de modo que la innovación no siempre implica la creación de algo inédito sino la reestructuración de modelos ya existentes, generalmente, porque la innovación está condicionada a la tecnología disponible en una determinada época y, además del contexto temporal, también está relacionada con los contextos socioculturales y económicos".

Con lo mencionado en el anterior párrafo se decidió que el desarrollo del aplicativo móvil se realice con la tecnología Nativescript-vue. Es una extensión de Nativescript, este es un framework creado por Telerik para crear aplicaciones multiplataforma nativas para iOS y Android sin vistas web (aplicaciones móviles interpretadas). Permite utilizar Angular, TypeScript o JavaScript moderno para obtener una interfaz de usuario y rendimiento verdaderamente nativos, al tiempo que reutiliza las habilidades y el código de nuestros proyectos web (Telerik, 2017).

Por otra parte, también se desarrolló basándose en el paradigma offline first, siendo este algo nuevo para los desarrolladores de software porque la preocupación primordial es que la aplicación funcione fuera de línea, Feyerke (2013) menciona: "Cuando se trata de crear aplicaciones, a menudo suponemos que nuestros usuarios se parecen mucho a nosotros. Los

imaginamos con los últimos dispositivos, el software más reciente y las conexiones más rápidas". Nuestro desarrollo estuvo enfocado en la disponibilidad de la aplicación en cada lugar y momento requerido, porque nuestros usuarios no siempre estarán en lugares con conexión rápida.

La aplicación nos permite registrar de manera automática cada lugar relevante que el usuario haya visitado durante su viaje y así visualizar durante el viaje podrá su ruta recorrida, ya que estos se registraron de manera automática mediante la geolocalización por GPS. Al mismo tiempo en la ruta se muestra las fotos tomadas, alguna nota del lugar turístico, comentarios que el usuario haya realizado y también estadísticas de viaje que se recopilaran de manera automática.

Como ya sabemos google es capaz de organizar nuestras fotos usando una combinación de fuentes de información: etiquetas geolocalizadas en las fotos (información incrustada en las fotos sobre las coordenadas geográficas en las que fueron tomadas), información de ubicaciones de Google Now o Google Maps, y datos GPS.

1.4. Presunción filosófica

Como estudiante formado bajo los principios bíblicos y teniendo durante 5 años una educación adventista, puedo afirmar que el motivo principal para desarrollar un aplicativo móvil es el servicio. Apoyar a los turistas a tener una mejor calidad de vida en sus viajes y también contribuir a los emprendedores a tener mayor ingreso con la afluencia de turistas en la región de Puno.

CAPÍTULO II. Marco teórico

2.1. Revisión de la literatura

Sánchez (2012) en su tesis titulada: "Integración de Foursquare y Geolocalización en una Aplicación Móvil para la Creación de Rutas Turísticas", tiene como objetivos: primeramente, crear una aplicación móvil que permita al usuario crear rutas turísticas, luego publicarlas en un servidor Web, y descargar rutas de otros usuarios desde el servidor Web para que éstas sean recorridas punto a punto. La metodología utilizada fue la siguiente: en primer lugar, la aplicación se integró con la red social Foursquare, está integración permitió que la aplicación muestre lugares registrados, para la creación de rutas turísticas. Seguidamente la aplicación localiza la ubicación del usuario, esto significa que los puntos o lugares de ocio se obtienen en función de la posición de este mediante GPS del celular. Por último, se realizó la integración del aplicativo móvil, con un servidor web en el que los usuarios pueden publicar las rutas creadas. En conclusión, del proyecto se desarrolló para dispositivos Android integrándose a la red social Foursquare mediante la Api que proporciona dicha red social. La implementación de geolocalización en el aplicativo permite al usuario crear rutas de manera fácil porque le muestra los puntos de interés durante su recorrido. Por último, las rutas creadas por el usuario se almacenan en un servidor web al cual se comunican mediante una API REST permitiendo tener una bitácora de rutas que otros usuarios pueden elegir y guiarse de ello.

Robles (2015) en su Tesis titulada: "Desarrollo de una Aplicación para equipos Android, basada en Geolocalización para Obtener Información de Atractivos Turísticos en la Ciudad de Tulcán", tiene como objetivo desarrollar la aplicación móvil "Conoce Tulcán" para equipos Android, basada en geolocalización para obtener información de atractivos turísticos en la Ciudad de Tulcán. El aplicativo se desarrolló para poder brindar información a los turistas de los distintos sitios, por falta de guías turísticos en la ciudad, para lograr el objetivo se desarrolló el sitio web en google site donde se almacenan la información, fotografías e imágenes de cada lugar turístico, esta información ser consumida por el aplicativo móvil que fue implementada en Android para lo cual primero se hizo el levantamiento de requerimientos, luego el diseño de prototipos y al final la codificación y generación del apk. Para terminar, se realizaron pruebas de usabilidad, funcionalidad y técnica de la aplicación

implementada. En conclusión, la aplicación funciona con geolocalización el cual permite ver los puntos de interés de la ciudad que están registradas en la página web creada, se comprobó que el funcionamiento de la app es más efectivo con red de datos móvil.

Prithvi Spandana, Shreyas KA y Anand Kumar M (2015) en el Artículo titulado: "Sistema de Rastreo Eficaz y Fiable para Monitorear a los empleados de campo mediante Móviles con GPS", el objetivo es determinar si el ingeniero encargado de un sitio visita al lugar ínsito en el tiempo indicado. Esto a raíz de que en los empleados de la organización no cumplen sus labores de manera responsable y esto afecta a la productividad de la empresa. Es por tal motivo que el sistema propuesto hace uso de la aplicación móvil Android para monitorear las actividades de los empleados y, por lo tanto, aumentar el rendimiento. Los registros y las notas se reemplazan por teléfonos inteligentes Android. La ubicación del usuario también llamada ubicación GPS se enviaría a un servidor remoto cuando el usuario realice un "REPORTS IN" desde la ubicación de su sitio, donde el servidor realiza un seguimiento de esta ubicación y calcula el área o el sitio (lugar) donde estaba el usuario. Cuando se llevó a cabo la notificación, se verifica si la ubicación del usuario esta en las coordenadas donde se registró la ubicación del sitio, entonces la presencia de los usuarios en la ubicación se considerará como presente y si la ubicación no está en las coordenadas donde se registró la ubicación del sitio, se consideran ausentes. En conclusión, se pudo observar que la aplicación realiza de manera automática el registro de asistencia de un empleado, mediante sus coordenadas de GPS y este los almacena en el servidor web.

Vanhala J. (2017) en su tesis titulada: "Implementing an Offline First Web Aplication", tiene como objetivos responder a las siguientes preguntas: "¿Qué características tienen los navegadores que permiten las aplicaciones web sin conexión? ¿Cómo se pueden utilizar estas características y Orbit para implementar una aplicación web con un enfoque offline first?". Para poder responder las preguntas se realizó primeramente la revisión literaria de offline first, en segundo lugar, se diseñó los requerimientos, seguidamente se desarrolló e implementó la aplicación y por último se realizó las pruebas correspondientes. De todo ello se tuvo como resultados que no todas las tecnologías que soportan offline-first son compatibles con los navegadores, para la sincronización de los datos al momento de cambiar de offline a online se tiene que gestionar la resolución de conflictos.

Gill O. (2018) en su tesis titulada: “Using React Native for mobile software development”, tiene como objetivo: “demostrar cómo se puede utilizar React Native como marco para el desarrollo de software móvil”. Para lograrlo primeramente realizó la revisión literaria de las tecnologías a utilizar, luego comparó react-native con otros frameworks, seguidamente implementó un prototipo y por último realizó las pruebas correspondientes de la aplicación. Como resultado la aplicación fue desarrollada en una plataforma cruzada React-Native, la aplicación permite al usuario administrar toda su agenda diaria y completar la información requerida de una manera fácil y por último se decidió que los desarrolladores de software de frontend en Nokia deben tener la habilidad de desarrollar en React-Native, porque la empresa eligió como su herramienta principal de desarrollo de interfaces de usuario de sus productos.

2.2. Marco teórico

2.2.1. Turismo

Según lo define la OMT (1998) el turismo "comprende las actividades que realizan las personas durante sus viajes y estancias en lugares distintos a su entorno habitual, por un período de tiempo consecutivo inferior a un año con fines de ocio, por negocios y otros".

En un sentido más amplio se le define como, “un fenómeno social que consiste en el desplazamiento voluntario y temporal de individuos o grupos de personas que, fundamentalmente con motivo de recreación, descanso, cultura o salud, se trasladan de su lugar de residencia habitual a otro, en el que no ejercen ninguna actividad lucrativa ni remunerada, generando múltiples interrelaciones de importancia social, económica y cultural” (Gurría, 1999:14).

El turismo se concibe hoy como una manifestación del comportamiento humano, resultado de la interacción social que implica desplazamiento voluntario y temporal de las personas, estableciendo interrelaciones socioculturales y también económicas. Éstas motivan a los actores económicos involucrados a incrementar las actividades que estimulen la permanencia de las personas y, de ese modo, generar mayores beneficios.

2.2.1.1. Turismo de experiencias

En la actualidad en los distintos ámbitos como: ferias, encuentros comerciales de turismo el ámbito académico y científico del turismo las forma de ver al turismo está cambiando de

manera radical y la palabra de moda es el turismo de experiencia. Este concepto, de acuerdo con el mismo diccionario de la Real Academia de la Lengua Española (RAE) es: “entre otras cosas, una circunstancia o acontecimiento vivido por una persona y también el hecho de haber sentido, conocido o presenciado alguien algo”.

“La experiencia turística no es otra cosa que un conjunto de impresiones físicas, emocionales, sensoriales, espirituales y/o intelectuales, que son percibidas de manera diferente por los turistas, desde el mismo momento en que planifican su viaje, lo disfrutan en el destino elegido e incluso cuando vuelven a su lugar de origen y recuerdan su viaje” (Otto and Ritchie, 1995). Los turistas de hoy al momento de tomar su decisiòn de elegir un destino toman en cuenta las consideraciones tangibles e intangibles.

Gândara, Mendes, Moital, Ribeiro, Souza y Goulart (como se citò en Rivera, 2016) menciona: “La experiencia es, en definitiva, una vivencia personal que interfiere en lo cotidiano del sujeto, reflejo de aspectos tangibles e intangibles que, en diferentes grados, impactan y sufren el impacto de acontecimientos únicos y memorables. Y, así, acaba generando emociones, encantamiento, historias, sueños y vivencias que son utilizados para entretener, fascinar o cautivar al turista”.

2.2.2. Desarrollo de aplicaciones móviles

Durante muchos años, la industria de los teléfonos inteligentes ha estado dominada por dos principales competidores, Android y Apple. A partir de 2016, estas dos empresas representan más del 99% de todas las ventas de teléfonos inteligentes a nivel mundial. Google compró el sistema operativo Android en 2005 a Android Inc. y aunque los dispositivos Android son distribuidos por muchos fabricantes diferentes, todos usan el mismo sistema operativo.

Los dispositivos móviles de Apple usan el sistema operativo iOS, que durante su concepción se basó en su MacOS existente. Esto significaba que muchas de las características que tenía MacOS durante la década de 2000 se incorporaron a iOS, entonces conocido como OS X, haciéndolos sentir familiares a los usuarios de iPhone. Debido a que Apple fabrica sus propios dispositivos, Apple puede crear un software específico para sus dispositivos. Mientras que a Android le cuesta más ajustar su sistema operativo y las

actualizaciones de software para admitir múltiples dispositivos con diferentes capacidades de hardware.

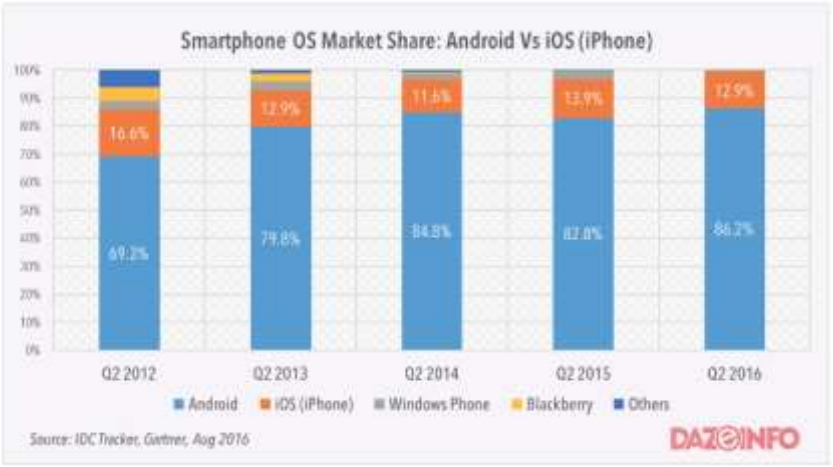


Figura 3. Smartphone OS Market Share.
Fuente: Apple Vs Android - A comparative study (2017) [online]. recuperado de https://android.jlelse.eu/apple-vs-android-a-comparative-study-2017 -c5799a0a1683.

La Figura 3 demuestra el aumento en el porcentaje total de participación de mercado de Android de 2012 a 2016. Sin embargo, el gráfico de barras solo representa un porcentaje del total de usuarios de teléfonos inteligentes y no representa el número cada vez mayor de propietarios de teléfonos inteligentes. Por lo tanto, aunque parece que el porcentaje de participación de mercado de Apple ha disminuido, la cantidad real de personas que poseen dispositivos de Apple ha aumentado en esos cuatro años (Apple Vs Android - A comparative study, 2017).

Figura 4. Distribución geográfica de los teléfonos inteligentes 2017.
Fuente: Apple Vs Android - A comparative study (2017) [online]. recuperado de https://android.jlelse.eu/apple-vs-android-a-comparative-study-2017 -c5799a0a1683.

La Figura 4 destaca las diferentes regiones en las que la mayoría de los usuarios de teléfonos inteligentes poseen un dispositivo iOS o Android. Es evidente que la mayoría de los países del Primer Mundo tienden a favorecer los dispositivos con iOS, mientras que los países en desarrollo prefieren los dispositivos con Android. Esto puede deberse principalmente a la diferencia de precio en los dispositivos, con dispositivos iOS que en promedio son significativamente más caros.

La Figura 4 también puede ayudar a explicar la gran diferencia entre los porcentajes de participación de mercado total, que se observan en la Figura 3, ya que los países con poblaciones extremadamente altas, como China, India y Rusia, parecen tener una mayor proporción de personas usando un dispositivo Android.

Con los dispositivos Android y Apple cada vez más popular, es más importante que los desarrolladores de software puedan admitir ambos tipos de dispositivos. Sin embargo, puede ser costoso y lento para las empresas tratar de crear y mantener aplicaciones de software que están codificadas individualmente para dos plataformas diferentes.

Como desarrollador, se requiere tener un buen canal de comunicación con el diseñador de la interfaz del usuario, ya que puede haber ocasiones en que el desarrollador puede enfocarse en una determinada forma de implementación y no ser capaz de considerar otros diseños. Sin embargo, los roles se pueden revertir y un desarrollador puede señalar las mejoras en la experiencia del usuario que podrían realizarse, ya que están trabajando más con la aplicación durante el desarrollo.

2.2.2.1. Consideraciones para el desarrollo Móvil

Como el objetivo del proyecto es mejorar la experiencia del turista durante su viaje mediante el uso de aplicación móvil se consideró la importancia del diseño, esta sección de la tesis se centrará en los conceptos básicos y los requisitos de conocimiento para desarrollar aplicaciones móviles. Canavesi destaca la importancia de enfocarse en la plataforma móvil en su artículo y afirma: "El marketing móvil se ha convertido en una de las partes más importantes, si no la más importante, de casi cualquier estrategia de marketing. La gente confía en sus dispositivos móviles para casi cualquier actividad imaginable y cualquier empresa que no sea parte de esta tendencia mundial parece estar fuera de contacto ".

A medida que el público confía más en sus dispositivos móviles, la industria continúa creciendo y aumenta la demanda de que las empresas tengan aplicaciones que funcionen bien. La revolución de la tecnología móvil ha resaltado la importancia de la usabilidad en una aplicación. Esto implica garantizar que el usuario tenga una experiencia agradable mientras usa las aplicaciones. Los firmantes de experiencia de usuario (UX) se han convertido en una parte fundamental del buen desarrollo de aplicaciones, especialmente para aplicaciones móviles, ya que deben considerar todas las diferentes posibilidades en las que un usuario puede usar la aplicación de manera más efectiva.

2.2.2.2. Paradigmas de desarrollo Móvil

Hoy en día, los entornos de desarrollo donde se generan las aplicaciones móviles son dinámicos e inciertos. La mayoría de estos son aplicaciones pequeñas, no críticas, pero los usuarios finales son un gran número, gracias a los métodos ágiles los desarrolladores liberan versiones rápidas con el fin de satisfacer las grandes demandas del mercado. Pero también están las apps de mayor tamaño, algunas incluso tienen funcionalidades offline, las cuales requieren métodos de sincronización con base de datos u otras apps.

Todas las particularidades mencionadas anteriormente hacen que el desarrollo de software para los dispositivos móviles sea muy distinto al tradicional. Por tales motivos se requiere de nuevas prácticas y metodologías que nos permitan promover las mejoras continuas en la Ingeniería de Software.

2.2.2.2.1. Desarrollo de aplicaciones nativas

Desarrollar aplicaciones nativamente nos brinda un conjunto de características ventajosas, las que sobresalen son: el acceso a todas las capacidades del dispositivo, un rendimiento alto, la posibilidad de desarrollar app offline es mayor y por ultimo funcionamiento en segundo plano realizando acciones necesarias o brindando notificaciones push al usuario cuando este se requiera. La distribución de estas aplicaciones se realiza en sus respectivas tiendas. Sin embargo, el único inconveniente es su alto costo en el desarrollo y mantenimiento porque el código fuente no se puede reusar en una plataforma diferente, se tiene que realizar esfuerzos dobles y por ende los costos incrementan, actualización y distribución de nuevas versiones se tiene que hace para cada una de las plataformas (Lisandro D, Galdamez N, Thomas P, 2016).

2.2.2.2.2. Desarrollo de aplicaciones móviles híbridas

Este tipo de aplicaciones se desarrollan con las tecnologías web (HTML, JavaScript y CSS). Estas aplicaciones corren dentro de un contenedor web especial con mayor acceso a las capacidades del dispositivo a través de una API específica. Las apps híbridas permiten que en nuestro desarrolló el código sea reutilizado en las distintas plataformas, tienen mayor acceso al hardware del dispositivo, y se distribuyen en sus tiendas respectivas. Aunque el rendimiento aceptable tiene las siguientes desventajas: Cuando usamos componentes que no sean nativos en la interfaz, esto perjudica la experiencia de usuario, y como se carga en un contenedor web para ser ejecutado por otra aplicación nativa la ejecución es relativamente lento (Lisandro D, Galdamez N, Thomas P, 2016).

2.2.2.2.3. Desarrollo de aplicaciones móvil interpretadas

Este tipo de aplicaciones son desarrolladas utilizando un lenguaje base, al final del desarrollo se compila en su mayor parte a código nativo, mientras el resto es interpretado en tiempo de ejecución. La implementación de este tipo de aplicaciones es de manera independiente, de las plataformas utilizando diversas tecnologías y lenguajes, tales como JavaScript, Java, Ruby y XML, entre otros.

La mayor ventaja de este tipo de aplicaciones es que se obtienen interfaces de usuario totalmente nativas. Sin embargo, los desarrolladores experimentan una dependencia total con el entorno de desarrollo elegido (Lic. Lisandro N, 2017). Los framework más populares para este tipo de desarrollo son Appcelerator Titanium, Nativescript, React-native, Flutter y otros.

2.2.2.2.4. Desarrollo de aplicaciones móvil generadas por compilación cruzada

Este tipo de aplicaciones al término del desarrollo se compilan de manera nativa y gracias a eso se tiene un producto en una versión específica con alto rendimiento para cada plataforma destino. Los entornos de desarrollo más utilizados para generar este tipo de aplicaciones son Applause, Xamarin, Embarcadero Delphi 10 Seattle y RubyMotion.

Por ejemplo, el entorno de desarrollo abierto Applause se escribe en un lenguaje específico del dominio basado en el framework Xtext, este está diseñado explícitamente para aplicaciones móviles orientadas a datos, y generar código fuente en Objective C, Java, C# o Python (Lic. Lisandro N, 2017).

2.2.2.2.5. Desarrollo de Aplicaciones Web Progresivas

En pocas palabras, las aplicaciones web progresivas son aplicaciones web que se cargan en un navegador web al igual que las páginas web o sitios web. Le brinda una experiencia móvil enriquecida a través de funcionalidades nativas, como la capacidad de trabajar sin conexión, las notificaciones push y la accesibilidad del hardware del dispositivo.

Se siente como una aplicación nativa y ofrece la misma experiencia que una nativa. No hay necesidad de descargarlo desde una tienda de aplicaciones. Se carga, ejecuta y funciona en un navegador web.

Los desarrolladores de PWA se plantearon alcanzar los siguientes objetivos: tener un rendimiento de calidad en dispositivos móviles, carga instantánea de la aplicación, una interfaz de usuario semejante en lo máximo posible a una nativa, funcionamiento offline (offline-first) y que el envío de notificaciones push sea idéntica a las apps nativas.

2.2.3. Paradigma de desarrollo offline first

Cuando se trata de crear aplicaciones, a menudo suponemos que nuestros usuarios se parecen mucho a nosotros. Los imaginamos con los últimos dispositivos, el software más reciente y las conexiones más rápidas. Y si bien podemos mantener un verdadero zoológico de dispositivos y navegadores más antiguos para realizar pruebas, pasamos la mayor parte de nuestro tiempo construyendo desde la comodidad de nuestros dispositivos de escritorio modernos, siempre en línea. (Feyerke, 2014)

La conectividad a internet es solo una parte del rompecabezas. Los dispositivos modernos se basan en diferentes tipos de redes inalámbricas: GPS, LTE, 4G, 3G, Edge, GPRS, Wi-Fi, Bluetooth y RFID, entre otros.

2.2.3.1. Principios de un buen diseño Offline First

2.2.3.1.1. Dame acceso ininterrumpido al contenido que me importa

Cómo pasamos mayor parte de nuestro tiempo en el teléfono queremos que nuestras aplicaciones móviles favoritas funcionen con normalidad aún si estamos fuera de línea. Es frustrante abrir una aplicaciòn y ver que no tiene ningun contenido para visualizar o interactuar, como usuarios queremos tener un montón de contenido disponible. Como ejemplo extremo, un navegador web que te permita navegar por Internet, tal como apareció

cuando estuviste conectado por última vez, sería una experiencia fantástica. Esto no existe porque las leyes de la física previenen tal experiencia (Sauble D., 2015).

Sauble D. (2015) menciona lo siguiente: “Naturalmente, no puede proporcionar una cantidad infinita de contenido a las personas. En su lugar, puede proporcionar la mayor cantidad de contenido posible sin dañar la experiencia del usuario, es decir, no robe todo su ancho de banda, no ocupe todo el espacio en su dispositivo, no permita que el rendimiento sufra...”

2.2.3.1.2. El contenido es mutable no dejar que mi estado cambie cuando pase de online a offline

Para lograr esto una práctica común es hacer que el contenido sea de solo lectura mientras está fuera de línea y cuando está en línea, puede editar, eliminar o manipular el contenido. Con esto generamos un malestar en la gente, porque ellos quieren seguir trabajando, pero les está diciendo que no pueden, generalmente sin una buena razón.

Sauble D. (2015) nos da un consejo a los desarrolladores: “Haga que el comportamiento en línea coincida con el comportamiento fuera de línea. Una vez que haya decidido qué pueden hacer las personas con sus datos, intente aplicar estas decisiones a todas las situaciones. Harás que sus vidas sean más fáciles y también la tuya, ya que solo tienes que diseñar y construir un paradigma de interacción único.”

2.2.3.1.3. Los mensajes de error no deberían dejar al usuario adivinando o preocupado innecesariamente.

Un mensaje de error inentendible, es un mensaje de error malo. Las aplicaciones bien diseñadas deben ser claras, utilizando lenguaje humano y un vocabulario consistente. Los mensajes de error deberían ser considerados importantes. Desafortunadamente, los mensajes de error a menudo se pasan por alto en el proceso de diseño. Cuando esto sucede, la aplicación asume dos caras: una cara amistosa cuando todo está bien y una cara angustiante cuando las cosas van mal.

El efecto puede ser discordante e incomprensible para algunas personas. Cuando su mundo se está cayendo a pedazos, necesitan más consuelo y ayuda de sus herramientas, no

menos. Asegúrese de pensar en la redacción de los mensajes de error y en la presentación visual de los errores (Sauble, 2015).

2.2.3.1.4. No me dejes comenzar algo que no puedo terminar

Las aplicaciones web y móvil se construyen para que puedan cumplir un proceso, por lo tanto, cuando se empieza a usar un módulo de la aplicación, tiene que permitirnos lograr el objetivo desea. Supongamos que empiezo una tarea que consta de ocho pasos. En el paso siete, si me dices que no puedo terminar porque estoy fuera de línea, me sentiré frustrado. Si me dices que no puedo guardar mi progreso y Tendré que volver a intentarlo más tarde, estaré furioso o simplemente que mi informaciòn se perdió debido al cambio de red no creo que vuelva a utilizar la aplicaciòn (Sauble. 2015).

Como desarrolladores de una aplicaciòn tenemos que adelantarnos a estos problemas, y los procesos que va tener nuestro aplicativo móvil tienen que cumplirse aun si me quede sin internet. No podemos hacer que nuestros usuarios sufran estos inconvenientes.

2.2.3.1.5. Una aplicación nunca debe contradecirse. Si existe un conflicto, sea honesto al respecto.

Uno de los mayores retos en offline first, es la resolución de conflictos en nuestra aplicación cuando se sincroniza la información offline a online o cuando dos personas cambian la misma información al mismo tiempo en offline. Sauble D. (2015) nos da un ejemplo diciendo: “Un escenario de cerebro dividido ocurre cuando dos personas cambian los mismos datos al mismo tiempo, en diferentes dispositivos. Cuando los dispositivos intentan reconciliar sus datos automáticamente, no pueden”. En este caso nuestro aplicativo nos debe dar la opción de decidir qué cambió conservar y qué no.

En una aplicación mal diseñada, pueden suceder varias cosas. La aplicación puede bloquear uno de los cambios sin preguntar, lo que genera confusión. Puede mostrar ambos cambios uno al lado del otro sin explicación, lo que también es confuso. Sauble D. (2015) menciona: “Siempre obtenga información de las personas cuando un conflicto no se puede resolver automáticamente. Si debe mostrar datos inconsistentes, siempre brinde una explicación.”

2.2.3.1.6. Los estados vacíos deberían decirme qué hacer a continuación de una manera amigable

Si nuestra aplicación está fuera de línea y no tiene nada de contenido para mostrar nos tiene que indicar un mensaje amigable sobre lo que está pasando. Los usuarios pueden hacer poco o nada con estos estados, lo cual es una mala experiencia.

Como desarrolladores siempre se debe pensar en el usuario final en este caso podemos decirle al usuario qué hacer a continuación. Puede ser una buena experiencia: proporcionar un mensaje inteligente, una imagen, una animación o un video que la gente normalmente no vería. Pueden contener datos de muestra, con los que las personas pueden interactuar para ver cómo funcionará normalmente la pantalla. Incluso pueden mantener a la gente entretenida con un juego, acertijo u otras distracciones (Sauble, 2015).

Los estados vacíos son muy fáciles de ignorar. Cuando los recursos de desarrollo son ajustados, están entre las primeras cosas para cortar, determine con qué frecuencia las personas las verán en relación con las otras pantallas y luego aplique los recursos de manera apropiada. Durante el uso fuera de línea, es más probable que ocurran estados vacíos, ahí es donde tenemos que actuar de manera inteligente como desarrolladores.

2.2.3.1.7. No me hagas recordar lo que estaba haciendo la última vez. Recuérdalo por mí

Si cierro mi aplicación, se debe conservar mi estado actual. Cuando vuelvo a abrir la aplicación, quiero ver lo que estaba haciendo por última vez o me resulta sencillo llegar allí. Cuanto más complicada es la aplicación, más importante es esto.

Muchas aplicaciones olvidan el contexto cuando se cambian de online a offline. Se dan permiso para restablecer ciertas partes de la IU, lo cual no es amigable y fácilmente se puede evitar. Al igual que con la mayoría de estos otros principios, no comprometa los atributos de una buena experiencia solo porque una aplicación esté fuera de línea (Sauble, 2015).

2.2.3.2. Protocolos de sincronización de offline a online

2.2.3.2.1. CouchDB Replication Protocol

Para entender esto tenemos que saber primero sobre CouchDB el cual es una base de datos NoSQL de código abierto basado en estándares comunes para facilitar la accesibilidad,

sincronización y compatibilidad web con diversos dispositivos. Apache CouchDb nos permite acceder a nuestros datos donde lo necesitemos, todo esto gracias a "Couch Replication Protocol" que se implementan en variedad de proyectos y productos. Su almacén de datos es de forma segura porque podemos realizarlo en nuestro propio servidor o en cualquier proveedor Cloud. El protocolo de replicación de Couch (CouchDB Replication Protocol) permite que nuestros datos fluyan sin problemas entre los clústeres del servidor a teléfonos móviles y navegadores, gracias a esto podemos disfrutar de una experiencia fuera de línea.

El protocolo de replicación CouchDB es un protocolo para sincronizar documentos JSON entre 2 pares a través de HTTP, mediante el uso de la API de REST pública de CouchDB y se basa en el modelo de datos de Apache CouchDB MVCC.

La estrategia de CouchDB para la resolución de conflictos, cuando se sincroniza datos de offline a online, se realiza con el protocolo de replicación couch de una manera deterministically-chosen winner, lo que significa que el dato que prevalecerá en los conflictos es escogido deterministicamente gracias al algoritmo determinístico. También cuenta con un árbol de revisión de documentos en conflicto, lo que permite la selección de un documento ganador por el desarrollador o usuario final.

A continuación, mostramos algunas implementaciones:

- PouchDB
- Cloudant Mobile Sync (iOS, Android)
- Couchbase Mobile (iOS, Android, Java, .NET, Xamarin, PhoneGap)

2.2.3.2.2. Realm Sync Engine

Realm Sync Engine es el motor de sincronización de de Realm, el cual es un sistema de base de datos NoSQL. Es similar a SQLite, pero en si no tiene que ver nada con SQLite. EnRealm definimos clases, luego estas clases definen su esquema y Realm almacena las instancias de estas clases como objetos.

El motor de sincronización de realm soluciona los conflictos mediante dos estrategias. La primera es que la última escritura gana ya sea cuando actualizamos o eliminamos objetos, la segunda es anexando documentos cuando se insertan en una lista.

A continuación, se muestran algunas implementaciones:

- Reino de JavaScript (React Native, Node.js)
- Realm Objective - C (iOS)
- Realm Swift (iOS)
- Reino de Java (Android)
- Reino Xamarin (.NET)

2.2.3.2.3. Firebase Realtime Database

Almacena y sincroniza datos con la base de datos NoSQL alojada en la nube. La sincronización de los datos es con todos los clientes en tiempo real, gracias a realtime database tenemos disponibilidad constante a los datos aun cuando la app no tiene conexión.

Firebase Realtime Database se alojada en la nube. La forma de almacenar los datos es en formato JSON. La forma cómo solucionar los conflictos de sincronización son: la última escritura gana cuando se usa lo métodos set () o update(), se anexa elementos cuando se una el método push() y por último mediante las estrategias suministradas por el desarrollador (developer-supplied strategy) cuando se usa el método transaction() .

2.2.4. Firebase

Firebase es una plataforma de desarrollo de aplicaciones web y móviles que proporciona un conjunto de herramientas orientadas a la creación de aplicaciones de alta calidad, al crecimiento de los usuarios y a ganar más dinero. Personalmente describo la plataforma como una suite de diferentes aplicaciones que nos harán más fácil el desarrollo de nuestra aplicación (Miguh R, 2017).

2.2.4.1. Historia

En 2011, antes que se le conociera como firebase, este era una startup conocida como Envolve. En aquel entonces, proporcionaba a los desarrolladores una API que permitía integrar funcionalidad de chat en línea, dentro de su sitio web.

Lo que es interesante es que las personas usaron Envolve para pasar datos de aplicaciones que eran más que mensajes de chat. Los desarrolladores utilizaron Envolve para sincronizar los datos de las aplicaciones, como el estado de un juego, en tiempo real entre sus usuarios.

Gracias a esto los fundadores de Envolve, James Tamplin y Andrew Lee, empezaron a separar el sistema de chat y la arquitectura en tiempo real. En abril de 2012, Firebase se creó como una compañía independiente que proporcionaba Backend-as-a-Service con funcionalidad en tiempo real.

En el 2014 fue comprada por Google, y desde entonces se convirtió rápidamente en el gigante multifuncional de una plataforma móvil y web que es hoy en día.

2.2.4.2. Servicios de firebase

Los servicios de firebase se dividen en dos grupos como se muestra en la tabla 1.

Tabla 1.
Servicios de firebase

Servicios para desarrollar y realizar pruebas de la aplicación	**Servicios para hacer crecer y comprometer a la audiencia**
● Realtime Database ● Auth ● Test Lab ● Crashlytics ● Cloud Functions ● Firestore ● Cloud Storage ● Performance Monitoring ● Crash Reporting ● Hosting	● Firebase Analytics ● Invita ● Mensajería en la nube ● Predicciones ● AdMob ● Enlaces dinámicos ● Adwords ● Configuración remota ● Indexación de aplicaciones

2.2.4.3. Realtime Database

Almacena y sincroniza datos con la base de datos NoSQL alojada en la nube. La sincronizaciòn de los datos es con todos los clientes en tiempo real, gracias a realtime database tenemos disponibilidad constante a los datos aun cuando la app no tiene conexión.

Realtime database guardar todos los datos que requiera una aplicación. Se integra muy bien con React, Vue.js, NativeScript y otros; su patrón reactivo permite actualizar los datos en los componentes automáticamente.

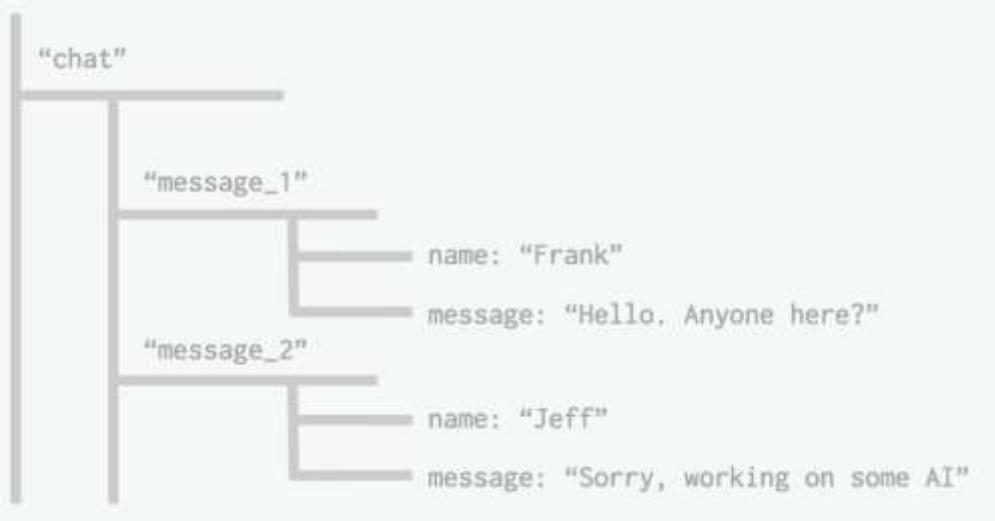


Figura 5. Objeto JSON de realtime database
Fuente: Rajat S. (2017) Introducción a firebase.

Mediante una sola API, proporciona a la aplicación el valor actual de los datos como las actualizaciones de esos datos. Gracias a la sincronización en tiempo real que tiene, facilita a los usuarios el acceso a la información desde cualquier dispositivo, ya sea web o móvil sin la necesidad de refrescar la página o ejecutar un método. Esto a la vez permite a los usuarios realizar trabajos colaborativos entre sí (Rajat S. 2017).

Otro beneficio increíble de Realtime Database es que viene con SDK para dispositivos móviles y web, lo que le permite crear aplicaciones sin la necesidad de servidores. Cuando sus usuarios se desconectan, los SDK de la base de datos en tiempo real utilizan el caché local en el dispositivo para servir y almacenar los cambios. Cuando el dispositivo se conecta, los datos locales se sincronizan automáticamente.

Figura 6. Simulación de la sincronización en tiempo real te realtime firebase
Fuente: Rajat S. (2017) Introducción a firebase.

2.2.4.4. Firestore

Figura 7. Diagrama de funcionamiento de la arquitectura de firestore
Fuente: Rajat S. (2017) Introducción a firebase.

Cloud Firestore es una base de datos de documentos NoSQL que le permite almacenar, sincronizar y consultar datos de una forma simple en las aplicaciones móviles y web, a una escala global. Aunque esto puede sonar como algo similar a Realtime Database, Firestore trae muchas cosas nuevas a la plataforma que lo convierten en algo completamente diferente al anterior.

2.2.4.4.1. Mejora de la consulta y la estructura de datos

Mientras Realtime Database almacena datos en forma de un árbol JSON gigante, Cloud Firestore adopta un enfoque mucho más estructurado. Firestore mantiene sus datos dentro de objetos llamados documentos. Estos documentos consisten en pares clave-valor y pueden

contener cualquier tipo de datos, desde cadenas hasta datos binarios e incluso objetos que se parecen a árboles JSON (Firestore lo llama mapas). Estos documentos a su vez, se agrupan en colecciones (Rajat S. 2017).

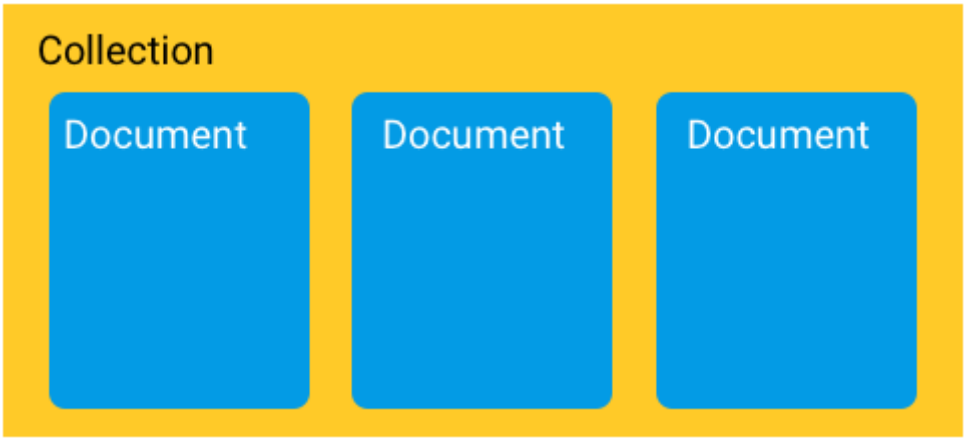


Figura 8. Diseño de Firestore una colección con 3 documentos
Fuente: Rajat S. (2017) Introducción a firebase.

Firestore trabaja con colecciones que contienen documentos, y estos también pueden contener sub-colecciones. Estas sub-colecciones a su vez pueden contener nuevos documentos que apuntan a otras sub-colecciones, y así sucesivamente, como se puede observar en la figura 9.

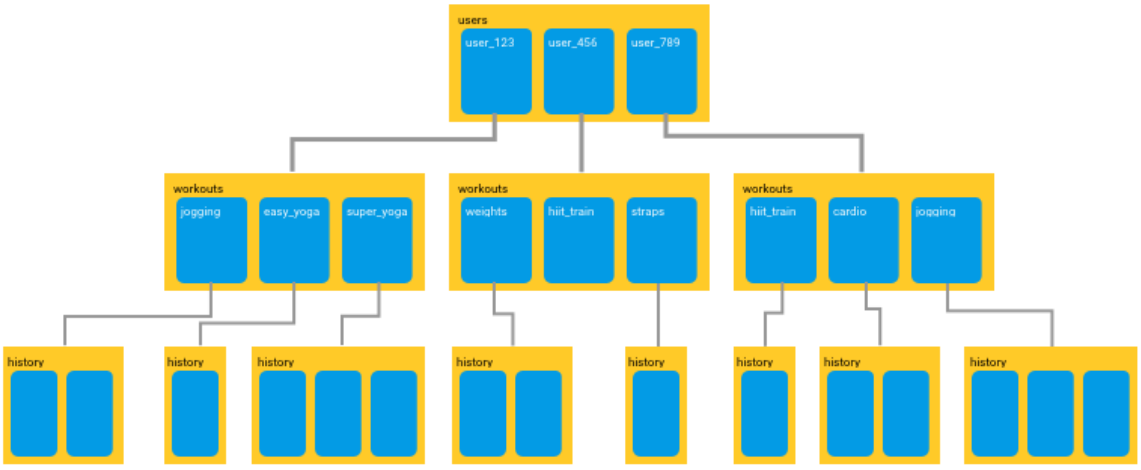


Figura 9. Diseño de Firestore mostrando un árbol de colecciones y documentos
Fuente: Rajat S. (2017) Introducción a firebase.

Las consultas en Firestore son superficiales, porque se puede recuperar cualquier documento que se desee, sin tener que recuperar todos los datos que se vinculan con el dato que deseamos.

2.2.4.4.2. Consultas en Firestore

Las consultas inician desde las colecciones, por ejemplo, se tiene una colección de ciudades, ahora para poder recuperar el listado de ciudades se tiene que almacenar primeramente la conección a la base de datos en una variable.

Figura 10. Variable citiesRef almacenando la colección de ciudades.
Fuente: Adaptación propia.

Ahora, si se requiere encontrar una lista de ciudades capitales, se debe escribir una consulta como esta:

Figura 11. Variable query almacenando la colección de ciudades capitales
Fuente: propia

2.2.4.5. Diferencias entre Realtime Database y Firestore

Para poder elegir entre las dos soluciones de base de datos se tiene que tener en cuenta lo siguiente:

El modelo de datos, como sabemos Realtime Database y Firestore son base de datos NoSQL.

Tabla 2.
Diferencias en la forma de almacenar los datos entre realtime database y firestore

Realtime Database	Cloud Firestore
Su forma de almacenar de datos es en un árbol JSON	La forma de almacenar sus datos son en documentos organizados en colecciones
Los datos simples se almacenan muy fácil	Los datos simples se almacenan muy fácil en documentos que son similares a los JSON
Los datos complejos y jerárquicos son más difíciles de organizarlos a escala	Los datos complejos y jerárquicos son más fáciles de organizar a escala, mediante las subcolecciones dentro de los documentos.

La compatibilidad sin conexión y tiempo real, esto es muy importante pero afortunadamente ambos tienen SDK en tiempo real centrados en dispositivos móviles y admiten el almacenamiento de datos locales para las apps que funciona sin conexión.

Las consultas, es otro factor a tomar en cuenta al momento de elegir una base de datos.

Tabla 3.
Diferencias en la forma de realizar las consultas entre realtime database y firestore

Realtime Database	Cloud Firestore
• Las consultas son directas con funciones de ordenamiento y filtrado limitado. • Solo permite ordenar y filtrar según una propiedad. • Las consultas son profundas de predeterminada las cuales siempre dan como resultado el subárbol completo.	• Las consultas son indexadas con ordenamiento y filtrado compuesto. • Permite encadenar filtros y combinar filtrado con ordenamiento según una propiedad en una misma consulta. • Las consultas son superficiales para subcolecciones: Puedes consultar subcolecciones dentro de un documento en lugar de una colección entera o incluso un documento entero. • Las consultas se indexan de forma predeterminada: El rendimiento de las consultas es proporcional al tamaño del conjunto de resultados, no del conjunto de datos

Tabla 4.
Otras diferencias entre realtime database y firestore

Realtime Database	Cloud Firestore
• Operaciones básicas de escritura y transacción • El escalamiento necesita fragmentación • Reglas en cascada que se deben validar por separado • Se cobra solo por ancho de banda y almacenamiento, pero con una tarifa mayor.	• Operaciones atómicas de escritura y transacción • El escalamiento será automático • Seguridad más sencilla y potente para los SDK para dispositivos móviles, la Web y servidores • Se cobra principalmente por operaciones ejecutadas en la base de datos (lecturas, escrituras y eliminaciones) y, con una tarifa menor, por ancho de banda y almacenamiento. • Cloud Firestore admite límites de gasto diarios para proyectos de Google App Engine, a fin de garantizar que no excedas los costos esperados.

2.2.4.6. Autenticación

Figura 12. Tipos de autenticación de firebase.
Fuente: Adaptación propia.

La autenticación de firebase proporciona servicios de back-end, SDK fáciles de usar y bibliotecas de UI listas para usar en la autenticación de los usuarios en una aplicación.

Normalmente, construir un sistema de autenticación toma meses e incluso después de eso, se tendría que mantener un equipo dedicado para mantener las actualizaciones y mejoras en el sistema. Pero cuando usamos firebase, la configurar de autenticación al sistema es en menos de 10 líneas de código que manejan todo, incluidas operaciones complejas como la fusión de cuentas.

Los métodos de autenticación de usuarios, en una aplicación son de las siguientes formas:

- Contraseña de Email
- Números de teléfono
- Google
- Facebook
- Gorjeo
- & ¡Más!

El uso de la autenticación firebase hace que la creación de sistemas de autenticación segura sea más fácil, al mismo tiempo que mejora la experiencia de inicio de sesión e integración para los usuarios finales. La autenticación de direbase fue creada por las mismas personas que crearon el inicio de sesión de Google, Smart Lock y Chrome Password Manager (Rajat S. 2017).

2.2.4.7. Firebase Storage

Firebase Storage es una solución independiente para cargar contenido generado por usuarios como imágenes y videos desde un dispositivo iOS y Android, así como también desde la Web. Está diseñado específicamente para escalar sus aplicaciones, brindar seguridad y garantizar la resistencia de la red. La forma en cómo gestiona el contenido es en simples carpetas/archivos para estructurar sus datos.

2.2.5. Vue.js

Vue es un marco progresivo para crear interfaces de usuario. A diferencia de otros marcos monolíticos, Vue está diseñado desde cero para ser adoptable de forma incremental. La

biblioteca central está enfocada solo en la capa de vista, y es fácil de captar e integrar con otras bibliotecas o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar aplicaciones sofisticadas de una sola página cuando se usa en combinación con herramientas modernas y bibliotecas de soporte.

2.2.5.1. Ciclo de vida de una instancia en Vue

Toda aplicación de Vue comienza con una nueva instancia con la que funciona como se puede observar en el siguiente fragmento de código:

```
var vm = new Vue ({
  // options
})
```

Aunque no está estrictamente asociado con el patrón MVVM, el diseño de Vue se inspiró en parte en él. Como convención, a menudo usamos la variable vm (abreviatura de ViewModel) para referirnos a nuestra instancia de Vue. La figura 13, muestra el diagrama del ciclo de vida de una instancia.

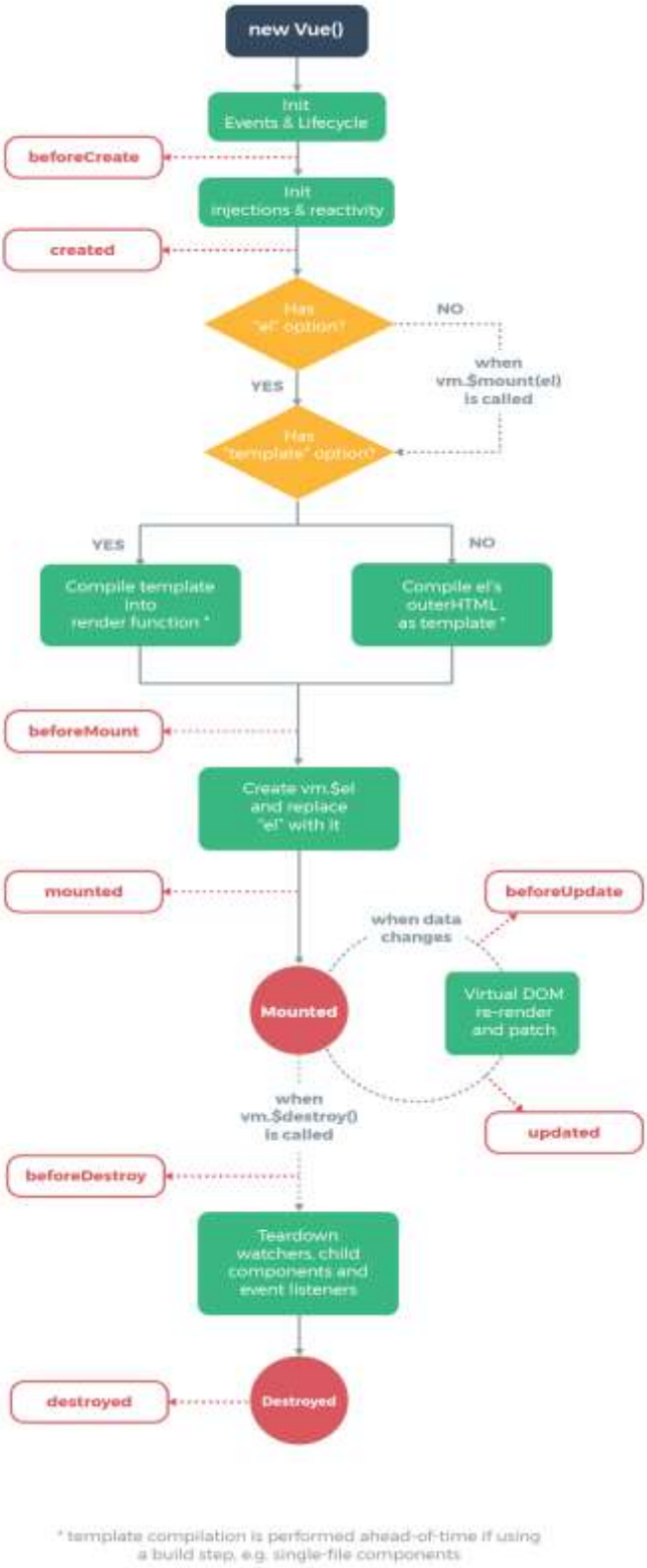


Figura 13. Ciclo de vida de una instancia vue.
Fuente: Vuejs.org, 2019

2.2.6. NativeScript

NativeScript es un framework creado por Telerik para crear aplicaciones multiplataforma nativas para iOS y Android sin vistas web (aplicaciones móviles interpretadas). Permite utilizar Angular, TypeScript o JavaScript moderno para obtener una interfaz de usuario y rendimiento verdaderamente nativos, al tiempo que reutiliza las habilidades y el código de nuestros proyectos web. (Telerik, 2017)

NativeScript proporciona un módulo multiplataforma que permite obtener aplicaciones nativas desde código JavaScript, dicho módulo expone las capacidades del dispositivo y de la plataforma subyacente, de una manera consistente y permite accederlas desde el código JavaScript (Walker & Anderson, 2017).

Además de JavaScript, se apoya en archivos XML para la definición del interfaz y un subconjunto de archivos CSS para personalización, abstrayéndose de los componentes nativos reales. Cuando la aplicación es compilada, parte del código multiplataforma es traducido a código nativo, mientras que el resto es interpretado en tiempo de ejecución (Delía, 2017).

Por el momento, NativeScript permite generar aplicaciones para Android e iOS, y se prevé dar soporte a Windows Phone. NativeScript renderiza controles nativos, sin utilizar WebViews, logrando una performance y experiencia de usuario similar al de las aplicaciones nativas. Esto nos permite poder utilizar el glue-code que se encarga de juntar las piezas y dejar el procesamiento pesado y la interfaz a las APIs nativas del dispositivo. Para ello, Nativescript utiliza el motor V8, en el cual nos permite incluso ejecutar código nativo utilizando objetos predefinidos (Anderson, 2016).

2.2.6.1. ¿Por qué NativeScript es importante?

NativeScript ofrece un marco de implementación en todas partes que se escribe una sola vez. En mi experiencia de trabajar con otros marcos, hay muchos códigos "shim" que deben escribirse. Este código actúa como una pieza de madera que se usa para enmarcar una puerta. En la mayoría de los casos, la puerta encajaría perfectamente, pero a veces necesitamos agregar un poco aquí y un poco allí para que encaje y funcione correctamente.

Al escribir el código, es posible que deba agregar un poco de código de UI para hacer que un botón se muestre correctamente en la versión de Android de la aplicación. Del mismo modo, es posible que deba escribir un código de UI adicional para que el cuadro desplegable se vea bien en la aplicación de iOS (Nick B, 2016).

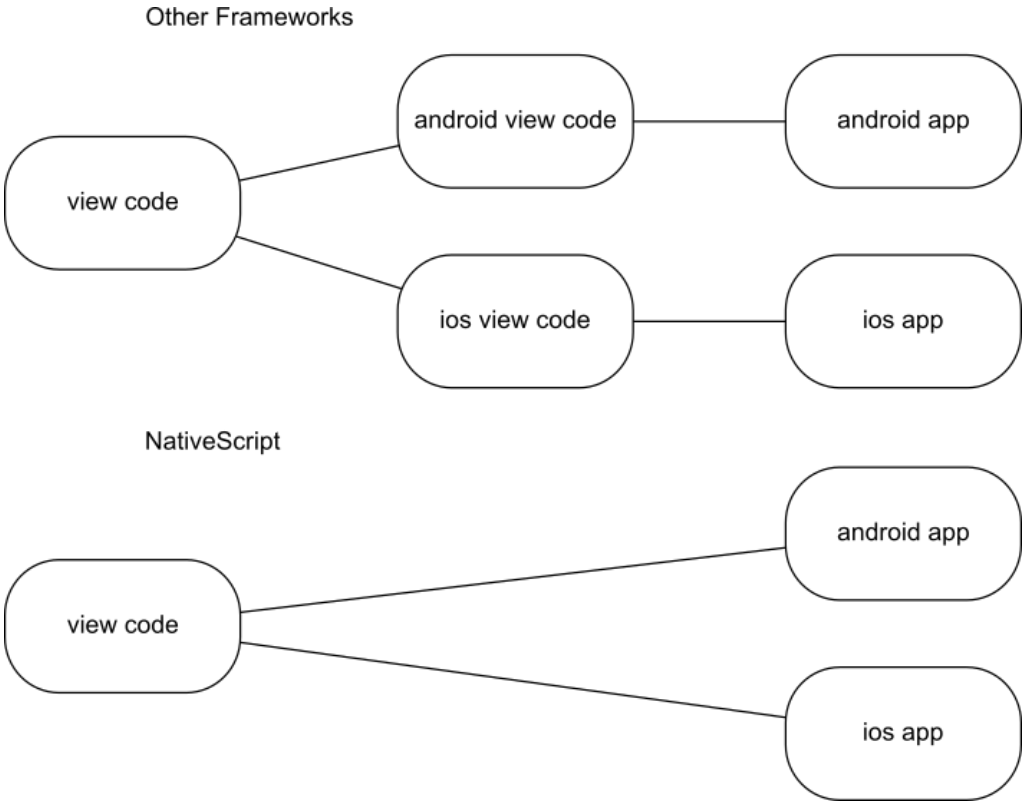


Figura 14. NativeScript: escritura una vez, despliegue en todas partes.
Fuente: Nick B, 2016.

La figura 14 ilustra el ejemplo del código shim. Otros marcos pueden requerir que se escriba un código específico de Android o iOS para su aplicación. El código de vista de NativeScript (y el código que no es de vista) se escribe una vez y se traduce en una experiencia nativa en cualquier plataforma en la que esté apuntando y ejecutando su aplicación, lo que significa que no necesita escribir código shim. Aunque no tiene que escribir código shim en aplicaciones NativeScript, aún puede escribir una aplicación totalmente personalizada utilizando características y hardware nativos.

2.2.6.2. Cómo funciona NativeScript

Figura 15. Funcionamiento de NativeScript
Fuente: nativescript.org, (2019)

Escribir aplicaciones móviles nativas utilizando JavaScript, XML y CSS no es algo que se escuche comúnmente; en cambio, escuchas sobre la escritura de aplicaciones móviles nativas en Objective C, Swift o Java. NativeScript hace posible escribir aplicaciones móviles nativas con varios componentes: el tiempo de ejecución de NativeScript; módulos centrales; Máquinas virtuales de JavaScript; su código de aplicación; y la interfaz de línea de comandos de NativeScript (CLI) (Nick B, 2016). La Figura 15 muestra cómo estos componentes trabajan juntos para crear proyectos nativos de Android e iOS, que se integran en aplicaciones nativas que se ejecutan en un dispositivo móvil.

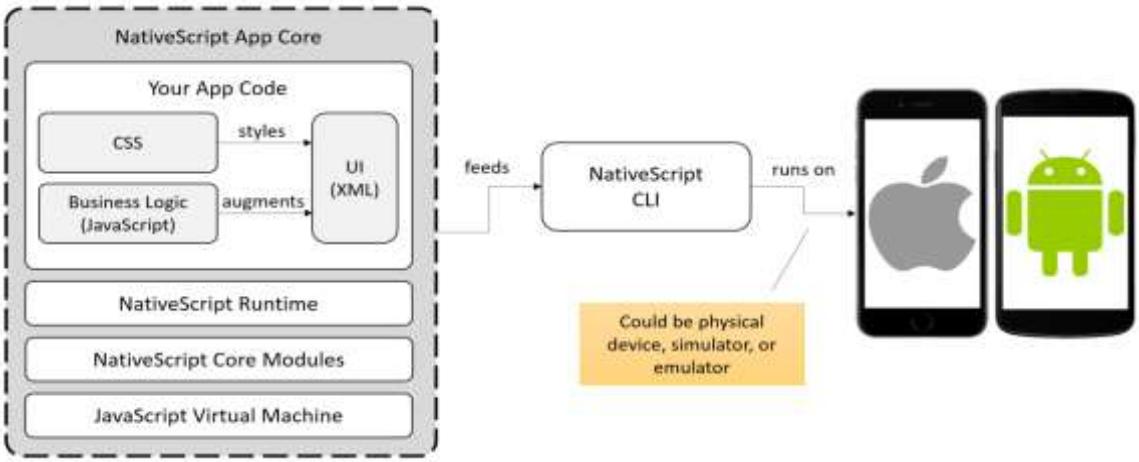


Figura 16. Cómo los componentes de NativeScript y el código de su aplicación trabajan juntos para crear y ejecutar aplicaciones nativas de iOS y Android.
Fuente: Nick B, (2016)

Por sí mismo, el código de su aplicación no tiene todo lo que necesita para ejecutarse en un dispositivo móvil. También necesita la ayuda de tres componentes adicionales: el tiempo de ejecución de NativeScript; módulos centrales; y una máquina virtual de JavaScript. El código de su aplicación y estos tres componentes forman el núcleo de su aplicación NativeScript.

Una vez que haya desarrollado el código de su aplicación, el núcleo de la aplicación se alimenta de la interfaz de línea de comandos (CLI) de NativeScript. La CLI es responsable de crear proyectos nativos de Android e iOS y fusionar el núcleo de la aplicación NativeScript. La CLI invoca los kits de desarrollo de software (SDK) nativos para compilar y compilar los proyectos de aplicaciones nativas (que incluyen el núcleo NativeScript de su aplicación) en una aplicación nativa. La aplicación compilada es implementada por la CLI y se ejecuta en un dispositivo físico, simulador o emulador.

2.2.6.3. Nativescript-vue

Simplemente, es un complemento que permite el uso de Vue.js con NativeScript. No es una bifurcación de Vue ni tampoco una bifurcación de NativeScript. Se basa en las capacidades de esos marcos para proporcionarle los medios para crear aplicaciones verdaderamente nativas y multiplataforma con esta nueva y brillante herramienta para la que todos están entusiasmados (Rob Lauer, 2018).

Dado que Vue.js no (directamente) interactúa con el DOM del navegador (similar a React y Angular), el código que podría haber pensado que solo estaba destinado a la web funciona bien en una aplicación NativeScript (excepto la sintaxis de la plantilla). En la siguiente figura se puede observar el funcionamiento de nativescript con vue y otros frameworks.

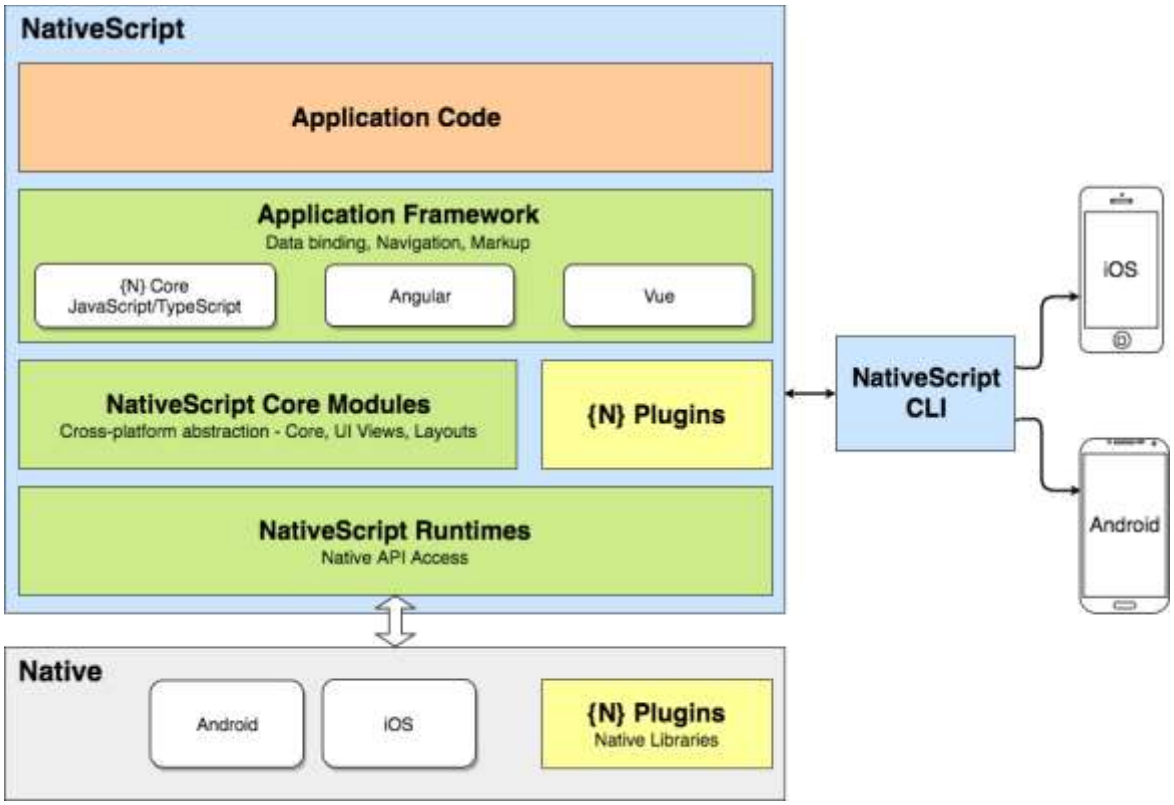


Figura 17. Funcionamiento de nativescript-vue al ejecutar la aplicación en iOS y Android.
Fuente: docs.nativescript.org, (2019)

Básicamente, los tiempos de ejecución de NativeScript convierten la IU en IU nativa para el dispositivo y ejecuta los códigos JavaScript, como NodeJS, lo que significa que puede cargar tantos paquetes NPM como necesite siempre que no estén diseñados para la interfaz web, básicamente bibliotecas que ¡La necesidad de acceder al DOM a través del navegador web no puede funcionar con NativeScript! (Sandoche A, 2018)

2.2.7. Geolocalización

La geolocalización es una herramienta que permite obtener las coordenadas geográficas de un elemento en cualquier momento. Este proceso se puede realizar mediante el uso de un dispositivo GPS o mediante una conexión a internet. Los dispositivos GPS son elementos que utilizan la red de satélites especializados en este campo, permitiendo obtener la posición de un elemento en cualquier lugar del mundo. Esta red está compuesta por 27 satélites (24 en funcionamiento y tres extras por si falla alguno de los anteriores), situados todos ellos a una distancia de 12.645 millas (unos 19.300 Km.) (Trimble Navigation, 1996)

A continuaciòn damos el proceso que siguen los dispositivos de GPS para calcular la localizaciòn, Trimble Navigation (1996) lo define de la siguiente manera: “El dispositivo emite una señal indicando la hora y la información de la localización donde se encuentra. Esta señal es recibida por los satélites GPS, y comparada por tres de ellos, para calcular la posición del elemento”.

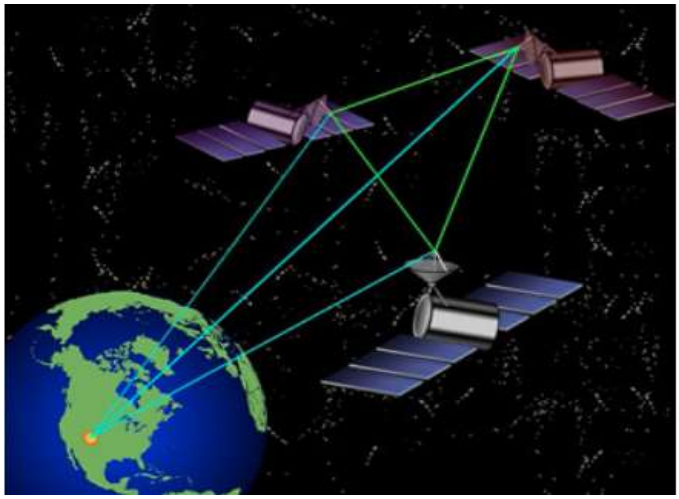


Figura 18. Triangulaciòn de los satélites GPS
Fuente: Sanchez, 2012

En la actualidad los satélites GPS tienen un sistema muy preciso que nos permiten calcular sin error la hora exacta en la que nos encontramos. De la misma forma, los dispositivos cuentan con sistemas de monitorización de los satélites para poder obtener con total precisión su posición (Sanchez, 2012).

2.2.7.1. Geolocalización en aplicaciones móviles

Actualmente, los teléfonos móviles, de gama alta, y unos cuantos, de gama media, traen integrados receptores GPS que mediante la red de satélites que rodea al planeta, puede ubicarnos en cualquier punto del planeta (Santiago, 2013).

Hoy en día existen una amplia gama de aplicaciones para dispositivos móviles que hacen uso de la geolocalización para resolver problemas cotidianos en la sociedad, tales como: proporcionar rutas de conducción, obtener estado de tráfico, localizar de objetos perdidos e incluso brindar servicios de transporte y mensajería privados

2.2.7.2. API (Application Programming Interfaces)

Una API es un conjunto de funciones o procedimientos utilizados por los programas informáticos para acceder los servicios del sistema operativo, bibliotecas de software, u otros sistemas. (BBVAOpen4U, 2016)

Figura 19. ¿Cómo funciona una API?
Fuente: (BBVAOpen4U, 2016)

La figura 18, muestra el funcionamiento de una API en la cual la aplicación/cliente realiza una petición web mediante HTTP o HTTPS la cual procesará la información enviada, realizará la acción necesaria y devolverá una respuesta en un formato deseado como JSON, XML o texto plano.

2.2.8. Google Console Developer

Es una plataforma web proporcionada por Google para los desarrolladores que deseen integrar las APIs de Google a sus proyectos. Los programadores deben crear sus proyectos en esta plataforma y habilitar las APIs deseadas para obtener un API KEY que es una cadena de texto necesaria para el uso de cualquier API proporcionada por Google. De esta manera mediante el API KEY Google puede medir el tráfico de una aplicación que hace uso de una o más APIs, así como también limitar o bloquear ciertas irregularidades en el uso de estas.

2.2.9. Google Maps API

Las Google Maps API permiten a los desarrolladores crear aplicaciones web, móviles y de escritorio capaces de integrar las funcionalidades de Google Maps mediante el uso de SDKs o peticiones HTTP. Cabe recalcar que no todas las APIs de Google son gratuitas,

algunas de estas presentan limitaciones (Ejemplos: números de peticiones por día) o funciones que solo están disponibles en versiones de pago.

2.2.10. Google Place API Web Service

Esta Api nos permite obtener información de la misma base de datos que usan Google Maps. La API presenta más de 100 millones de negocios y puntos de interés que se actualizan regularmente mediante listas verificadas por el propietario y contribuciones moderadas por el usuario.

Para obtener los datos se debe realizar una petición HTTP en la cual se debe pasar como parámetros la ubicación del usuario, tipos de establecimientos o punto de interés, un radio de búsqueda y lo más importante el API KEY obtenido desde la consola de desarrolladores de google.

2.2.11. Mapbox

2.2.12. Metodología de Desarrollo Ágil

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término "ágil" aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía "ágil". (Canós, José H.; Letelier, Patricio; Penadés, Carmen, 2003, pág. 2).

Esta metodología se basa en principios para el desarrollo de software ágil, en la cual se valora: "Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar: Individuos e interacciones sobre procesos y herramientas, Software funcionando

sobre documentación extensiva, Colaboración con el cliente sobre negociación contractual y Respuesta ante el cambio sobre seguir un plan." (Cunningham, 2001).

La metodologías ágil, como ya se mencionó se deriva de la lista de principios que se encuentran en el "Manifiesto Ágil" (Cunningham, 2001), y están basados en un desarrollo iterativo que se centra más en capturar mejor los requisitos cambiantes y la gestión de los riesgos, rompiendo el proyecto en iteraciones de diferente longitud, cada una de ellas generando un producto completo y entregable; e incremental donde un producto se construye bloque a bloque durante todo el ciclo de vida de desarrollo del producto, las iteraciones individuales deben producir alguna característica completamente funcional o mejorada (Szalvay, 2004).

2.2.12.1. Manifiesto Ágil

Beck K y Colaboradores (como se citó en Sullon A, 2014), fueron los que definieron los principios sobre los que se basan los métodos alternativos, de desarrollo ágil de software en cuatro postulados o valores, lo que ha quedado denominado como manifiesto ágil:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

De la misma forma juntamente con estos valores se redactaron los siguientes principios que derivan de estos valores:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiar la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.

2.2.12.2. Marco de Trabajo SCRUM

Schwaber K y Sutherland J (2010), autores de Scrum, definen que Scrum se basa en la teoría de control empírico del proceso, emplea un enfoque iterativo incremental para optimizar la previsibilidad y controlar los riesgos.

Resco y Castañeda (2016) menciona que: “Scrum es una metodología ágil en la cual se llevan a cabo una serie de prácticas iterativas, cuyo objetivo es que el grupo de desarrolladores trabajen unidos, contribuyendo con sus habilidades individuales para la obtención de un software de buena calidad”. Una de las principales características de Scrum es la entrega de un producto utilizable y funcional al final de cada iteraciòn; de esta manera, se puede ir observando los avances de las historias de usuario y al mismo tiempo realizar observaciones o modificaciones al requerimiento inicial.

En un equipo de Scrum existen tres diferentes roles: el propietario del producto, el Scrum master y los miembros del equipo (Resco y Castañeda, 2016).

Tabla 5.
Roles de un equipo en SCRUM

Rol	Descripciòn
Product owner	Es el encargado de que las necesidades del cliente y el usuario final sean entendidas por el equipo. Esto lo realiza creando, perfeccionando y comunicando los requisitos.
Scrum Master	Su principal función es la implementaciòn de las características de SCRUM, al mismo tiempo se encarga de velar los inconvenientes que puede ocurrir en el equipo creando soluciones anticipadas a ello.
Team	El equipo es el encargado de ejecutar las historias de Usuario, son altamente colaborativos, también se autoorganizan. Cada miembro del equipo toma sus propias decisiones y tiene la autoridad total sobre cómo va realizar su trabajo.

En SCRUM la base para el triunfo está en las distintas reuniones que se pueda tener, las reuniones ayudan a la planeación de cada actividad a desarrollar, en donde las experiencias adquiridas por los desarrolladores son muy relevantes, y por estos motivos la reocpilaciòn y retroalimentaciòn de datos es de suma importancia (Resco y Castañeda, 2016).

Tabla 6.
Reuniones dentro del equipo de SCRUM

Reuniòn	Descripciòn
Daily Scrum	Son las reuniones diarias, que se dan antes de iniciar las actividades del equipo. En esta reunión los participantes comparten de manera breve lo siguiente: - Las actividades completadas desde el ùltimo Daily scrum - Las actividades que espera completar - Las dificultades que tuvo durante el desarrollo
Sprint Planning Meeting	Marca el principio de un Sprint, Generalmente tiene dos partes: La primera parte tiene como objetivo comprometer al equipo a cumplir un conjunto de metas para el sprint. La segunda parte se identifican las tareas a realizar de acuerdo al orden de prioridad de cada historia de usuario acordado.
Sprint Review Meeting	En esta etapa del sprint, el equipo muestra el trabajo realizado durante el sprint. Se muestra cada historia de usuario completa y las que faltan completar. En estas reuniones es donde el propietario compara el trabajo realizado con las historias de usuario para revisar las características solicitadas por el cliente; al mismo tiempo

	evalúa si hay la necesidad de hacer cambios o agregar nuevas funcionalidades.
Retrospective	Se realiza al final de cada sprint, donde cada miembro del equipo analiza sobre cómo le fue durante el sprint, qué cosas aprendió y qué dificultades tuvo. Todo esto ayudarà a realizar mejoras del siguiente sprint

Scrum se fundamenta en los Sprint, los cuales son ciclos de desarrollo (iteración). en Scrum no está bien establecida los tiempos del sprint. Dos semanas de duración son las más frecuentes, aunque también puede existir y es común sprint de 1 y 3 semanas de duraciòn

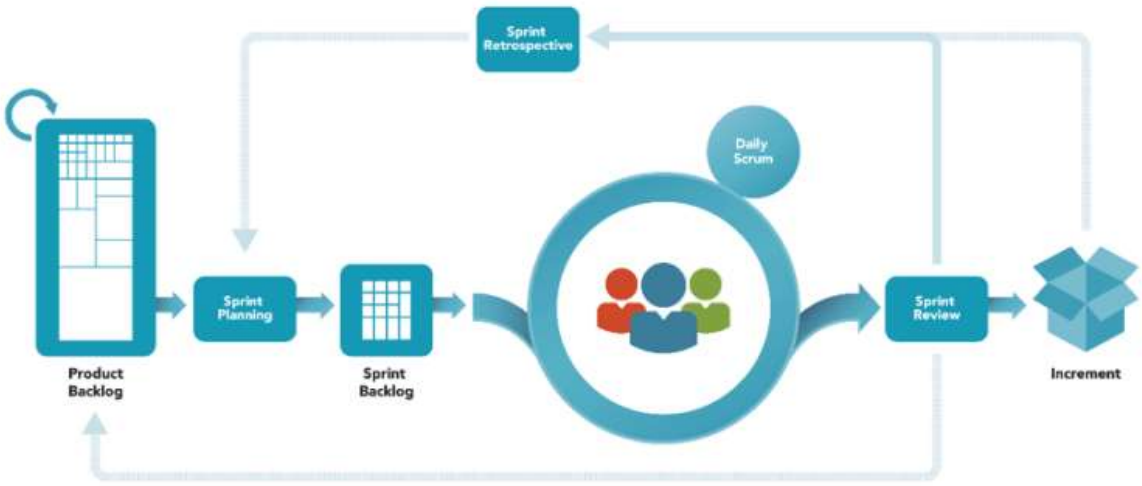


Figura 20. Visiòn General de Scrum
Fuente: Scrumguides.org, (2018)

CAPÍTULO III. MATERIALES Y MÉTODOS

3.1. Descripción del lugar de ejecución

Puno está situada en la sierra sureste del país, Por el norte limita con la regiòn de Madre de Dios, por el sur con la regiòn de Tacna, por el este con la república de Bolivia y por el oeste con las regiones de Moquegua, Cusco y Arequipa. Su territorio tiene una extensión de 71,999 km2, en donde también se incluye el lago titicaca con una extensiòn de 4,996 km2 y 39 mk2 de territorio insular. El nevado de Viscachani es considerado el punto más alto de la regiòn que está a 6000 msnm y se encuentra ubicado en distrito de Sina, provincia de San Antonio de Putina. (Vera, 2010)

Vera (2015) menciona que: “la región Puno debido a su ubicación estratégica (eje Cusco – Puno - La Paz), su ancestral cultura, la presencia de culturas Pre Incas, Incas y vestigios

del Virreinato; aunado a innumerables atractivos de carácter natural (lago Titicaca, lagunas, ríos, ceja de selva, flora, fauna, etc.), ruinas arqueológicas, templos coloniales y su rico y variado folclore (es conocido como la 'Capital del Folclore Peruano'). El desarrollo de sus potencialidades turísticas, aunado a una agresiva promoción turística, le permitiría continuar siendo un destino importante en cuanto al turismo receptivo en el país".

3.2. Materiales e insumos

3.2.1. Principales Tecnologías de Desarrollo

Las principales herramientas de desarrollo que se usaron durante el desarrollo son: El framework Nativescript con su complemento Nativescript-Vue basado en el lenguaje de programación Javascript. La base de datos es gestionada por firebase.

3.2.2. Tecnologías de Seguimiento

Para la gestión del desarrollo del aplicativo móvil se utilizó la herramienta IceScrum. Para la comunicación del equipo se uso Slack, se trabajo, con herramientas de control de versiones, conjuntamente con gestores de repositorios en red (Git y GitLab), con tal de garantizar la disponibilidad del código y facilitar la recuperación de fallos.

3.2.3. Recursos Personales

- El proyecto estuvo conformado por un equipo de 3 personas con dedicación exclusiva de 10 horas semanales durante todo el periodo.
- Un asesor del proyecto con el que se tuvo reuniones de 2 horas semanales.
- Un patrocinador del proyecto, la persona que valido nuestros requerimientos.

3.2.4. Recursos Materiales

Tabla 7.
Distribuciòn de los recursos Materiales necesarios para el proyecto

Recurso	Tipo	Finalidad
Ordenador portátil Toshiba de 6 GB de RAM con distribución Ubuntu 18.10.0 LTS	Herramienta de Desarrollo	Equipo para desarrollar el aplicativo móvil
Ordenador de escritorio DELL de 8 GB de RAM con Distribución ubuntu 18.10.0 LTS	Herramienta de Desarrollo	Equipo para desarrollo del aplicativo móvil
Un monitor	Herramienta de Visualización	Equipo para desarrollo del aplicativo móvil
Smartphone HUAWEI Y7	Herramienta de desarrollo	Para testear el aplicativo android
Iphone	Herramienta de desarrollo	Para testear el aplicativo iOS
Visual Studio Code	Herramienta de desarrollo	Editor de texto para codificar el aplicativo
Google Drive	Herramienta de desarrollo	Para la documentación del proyecto
Gmail	Herramienta de Comunicación	Para la comunicación con los actores del proyecto - Dr. Investigación, Asesor y Patrocinador.
Git	Herramienta de desarrollo	Para el control de versiones del repositorio de código fuente
Servicios de Internet	Herramienta de desarrollo	Para la navegación y búsqueda de información
Cursos en EDTeam	Herramienta de aprendizaje	Para obtener conocimiento sobre las tecnologías de desarrollo a usar en el proyecto
Libro de Development offline First Web	Herramienta de aprendizaje	Para obtener conocimiento de la teoría offline first
Microsoft Project 2016	Herramienta de Gestión	Para la planificación del proyecto

3.3. Tipo de Investigación

La presente investigación es de tipo aplicada y tecnológica ya que se desarrolló un producto, utilizando las tecnologías actuales para lograr los objetivos planteados que solucionen la necesidad en un sector de la población. (CIANCIACTIVA & CONCYTEC, 2017), en las bases integradas de proyectos de investigación básica y aplicada, define una investigación aplicada de la siguiente manera: “Está dirigida a determinar, a través del

conocimiento científico, los medios (metodologías, protocolos y/o tecnologías) por los cuales se puede cubrir una necesidad reconocida y específica".

3.4. Metodología

La metodología Ágil, fue aplicada para desarrollar nuestra aplicación mediante el marco de trabajo de Scrum, ya que es un proceso en el que se aplican de manera regular, un conjunto de buenas prácticas para elaborar colaborativamente en equipo, y obtener el mejor resultado posible de un proyecto junto con el lenguaje Unificado de Modelado (UML), con el objetivo de esquematizar la interacción de los usuarios con la aplicación móvil y web.

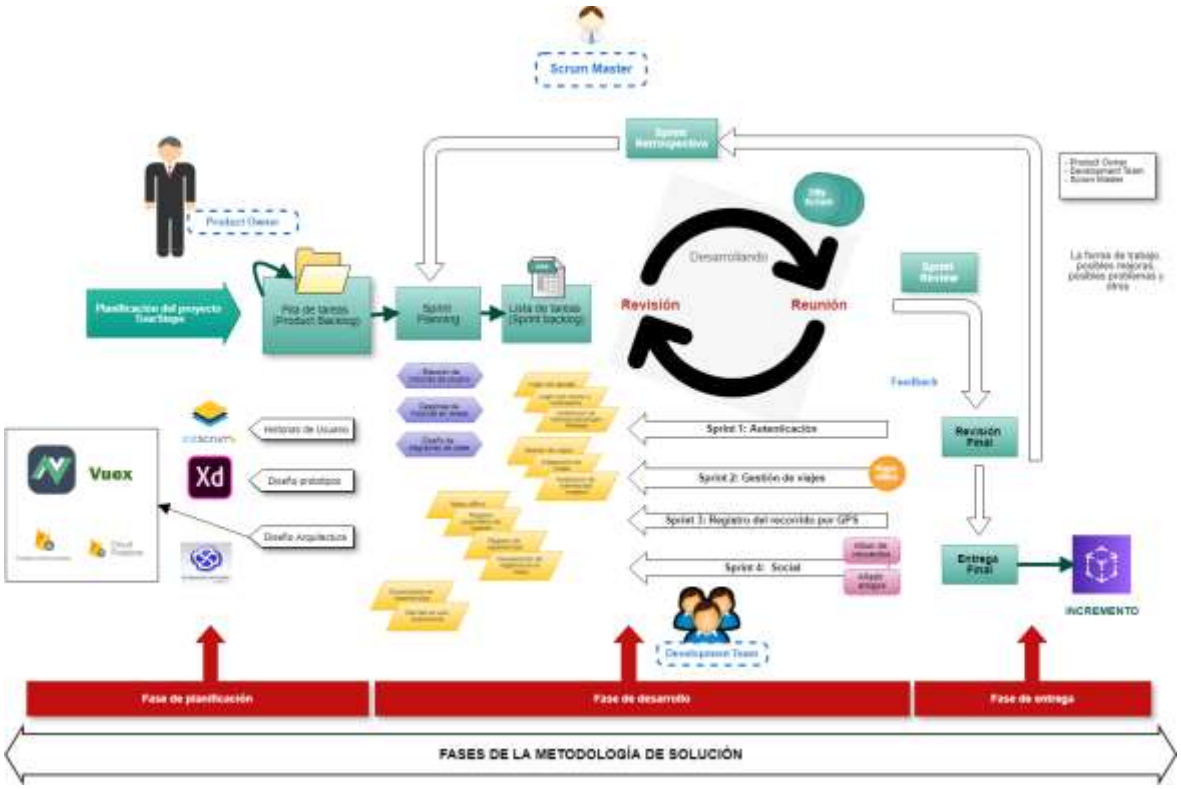


Figura 21. Muestra las 3 fases de la metodología scrum y las actividades a realizar en cada fase.
Fuente: Adaptación propia

Gracias al marco de trabajo de scrum el desarrollo de nuestro proyecto se realizó en tres fases.

3.4.1. Fases del marco de desarrollo Scrum

Este modelo cuenta con algunas fases no complejas, las cuales se resumen a continuación:

- **Planificación:** Aquí se realizó una idea del producto a desarrollar, los requisitos principales que se buscan en el proyecto y ciertas especificaciones. Principalmente se elaboró la Pila de tareas (Product Backlog)

- **Desarrolló:** La fase de desarrollo estuvo separado en 6 iteraciones iniciando desde la lista de tareas (Sprint Backlog), una iteraciòn tuvo como objetivo generar un incremento (Producto Terminado).

- **Entrega:** En esta parte se realizó para revisión final y entrega del incremento. Como estamos con scrum cada iteración también tuvo su fase de entrega del incremento (sprint review).

3.4.2. Fase de planificación

En esta etapa inicial del proyecto nos planteamos el nombre de la aplicación, los requisitos fundamentales (funcionales y no funcionales). Para gestionar de una manera más óptima scrum se utilizó la herramienta IceScrum donde empezamos a construir nuestro Product Backlog. En esta herramienta nuestras historias de usuario van pasar por un Flujo de Historias, que se compone de 8 estados.

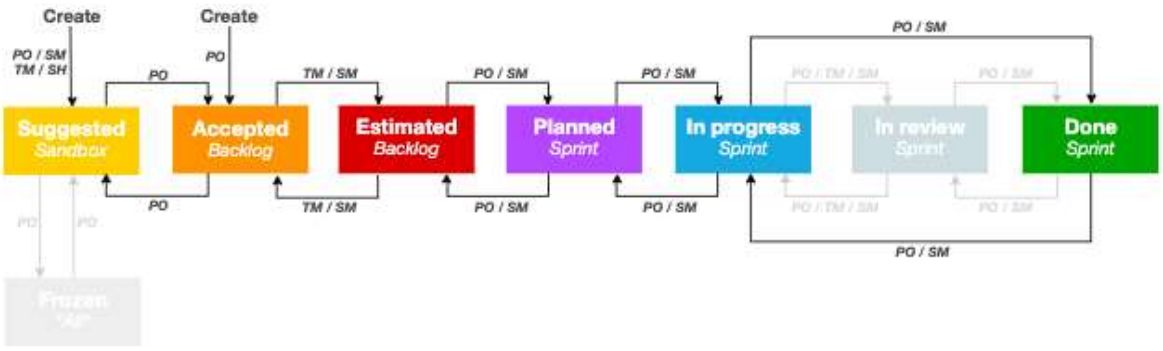


Figura 22. Flujo de estados de una historia de usuario en IceScrum
Fuente: icescrum.com, (2019)

En la figura 21, se puede observar los 8 estados donde inicia de las historias con estado sugerido en el backlog y termina en historias terminadas dentro de un sprint.

IceScrum además nos permite separar las historias por caracteristicas. Por eso nuestro Product Backlog va estar separado por características (Features) de nuestra aplicación, como se puede mostrar en la figura 22:

Figura 23. Tablero de características en IceScrum
Fuente: propia

3.4.2.1. Product Backlog

Una vez definida las características, aunque no siempre son necesarias, se inició a registrar todos los requerimientos posibles, en iceScrum cualquier persona que tenga acceso al proyecto puede registrar los requerimientos. Esta es la primera etapa de la construcción de nuestro Product Backlog. La figura 23, nos muestra el listado de requerimientos o historias de usuario registrados en iceScrum, con un estado sugerido (suggested), esto significa que no necesariamente todo lo que se menciona aquí fueron desarrollados.

Figura 24. Lista de requerimiento sugeridos en IceScrum
Fuente: propia

Una vez que se definió los requerimientos posibles el dueño del producto (Product Owner) acepta las historias y los publica para que sean estimadas por los miembros del equipo. Eso es lo que indica iceScrum, pero como se sabe dentro de un Equipo Scrum, no necesariamente el Product Owner, puede ser alguien que conozca el alcance total del

producto, es por eso que en nuestro caso las historias de usuario fueron aceptadas por todo el Equipo Scrum como se puede observar en la siguiente figura.

Figura 25. Product Backlog en IceScrum
Fuente: propia

La figura 24, muestra nuestro primero Product Backlog ya concluido, cabe recordar que no necesariamente se realiza todo lo que indica, puesto que en el proceso pueden aumentarse nuevas historias, pueden cambiar o simplemente quitarse. Para visualizar de manera más detallada las historias de usuario ver el ANEXO A.

3.4.2.2. Arquitectura

La aplicación TourStep se desarrolló con el framework Nativescript, específicamente usamos Nativescript y Vue.js. Vue.js es un marco ligero para crear interfaces de usuario atractivas. NativeScript potencia las aplicaciones móviles multiplataforma (realmente nativas), utilizando las habilidades web que ya conoces. La instalación del framework es un proceso simple que se detalla en el ANEXO B.

Como uno de nuestros requerimientos no funcionales, es una aplicación que funcione de manera offline, luego de realizar investigaciones optamos por firebase para que gestione nuestro almacén de datos y la sincronización cuando nuestra aplicación cambie de offline a online, puedes ver más información en el apartado Protocolos de sincronización de offline-online y también el apartado Firebase.

Además de eso gracias a que desarrollamos con Vue.js utilizamos Vuex, este es una biblioteca de patrones de administración de estado para aplicaciones Vue.js. Sirve como un

almacén centralizado para todos los componentes de una aplicación, con reglas que garantizan que el estado solo se puede mutar de manera predecible.

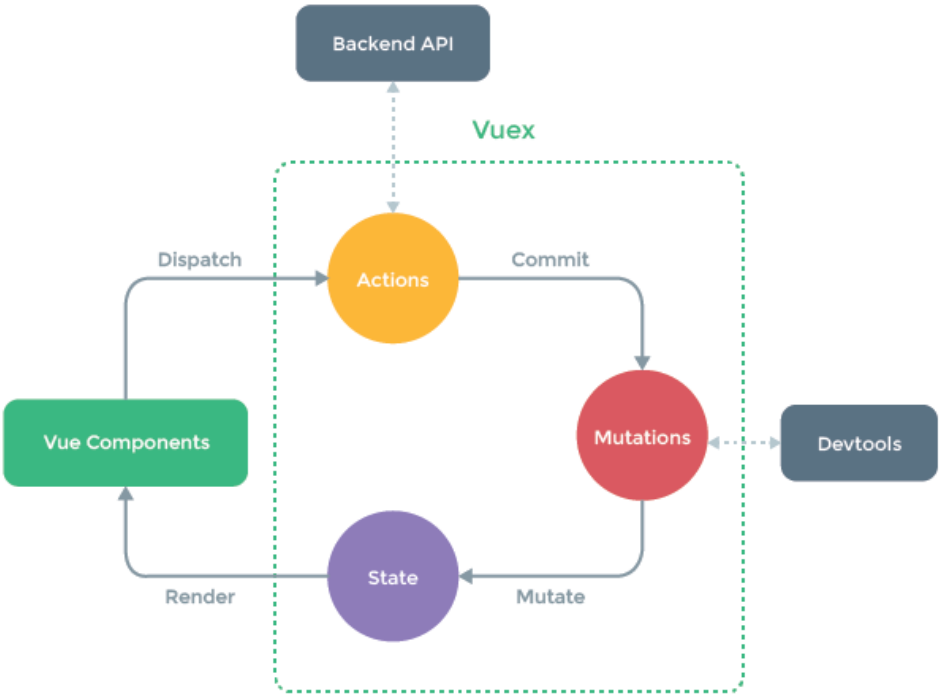


Figura 26, *Ciclo* de vida de un componente en Vuex
Fuente: vuex.org, (2019)

En conclusión, las combinaciones de estas tres librerías nos permiten tener una arquitectura para desarrollar aplicaciones móviles multiplataformas (Android y iOS), como si estuviéramos desarrollando aplicaciones web gracias a nativescript-vue, y firebase que le da un funcionamiento offline y gestiona nuestra base de datos, en la figura 26 se puede observar la arquitectura de la aplicación TourSteps.

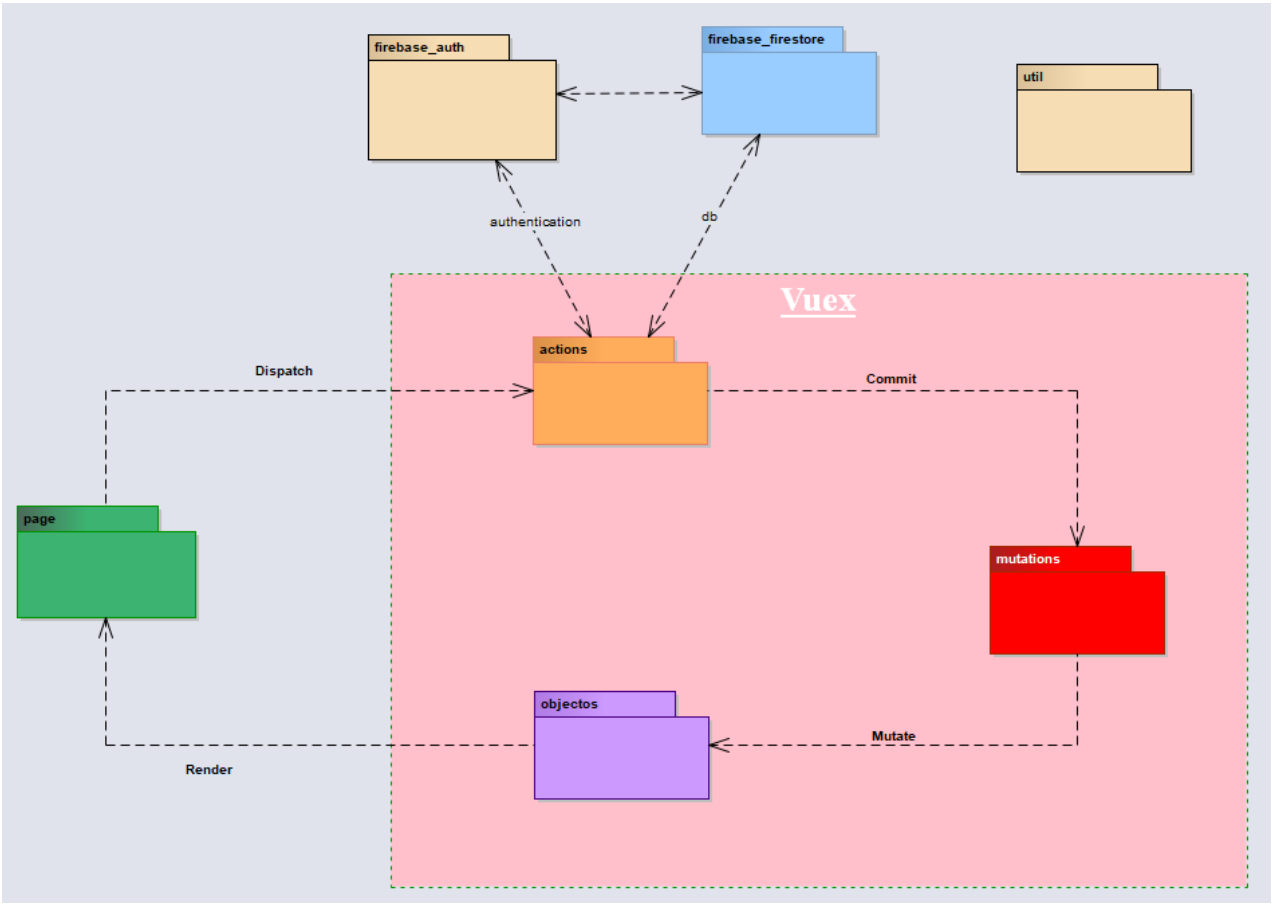


Figura 27. Arquitectura de la aplicación TourSteps
Fuente: propia

Para ver detalladamente nuestra arquitectura y la estructura de carpetas de nuestra app TourSteps ver el ANEXO C.

3.4.3. Fase de Desarrollo

La aplicación TourSteps ha evolucionado durante su proceso de desarrollo, en cada sprint del proyecto se ha implementado o se decidió quitar funcionalidades a la aplicación, con el fin de obtener una aplicación móvil de calidad.

La fase de desarrollo se llevó a cabo en 4 sprint o iteraciones de cada dos semanas.

3.4.3.1. Sprint #1

3.4.3.1.1. Sprint Planning

En el Sprint #1 se realizó el primer Sprint Planning, donde el Product Owner explicó al equipo de desarrolló que la prioridad es que un usuario puede autenticarse con google o gmail y el desarrolló inicie sobre una arquitectura que soporte funcionamiento offline.

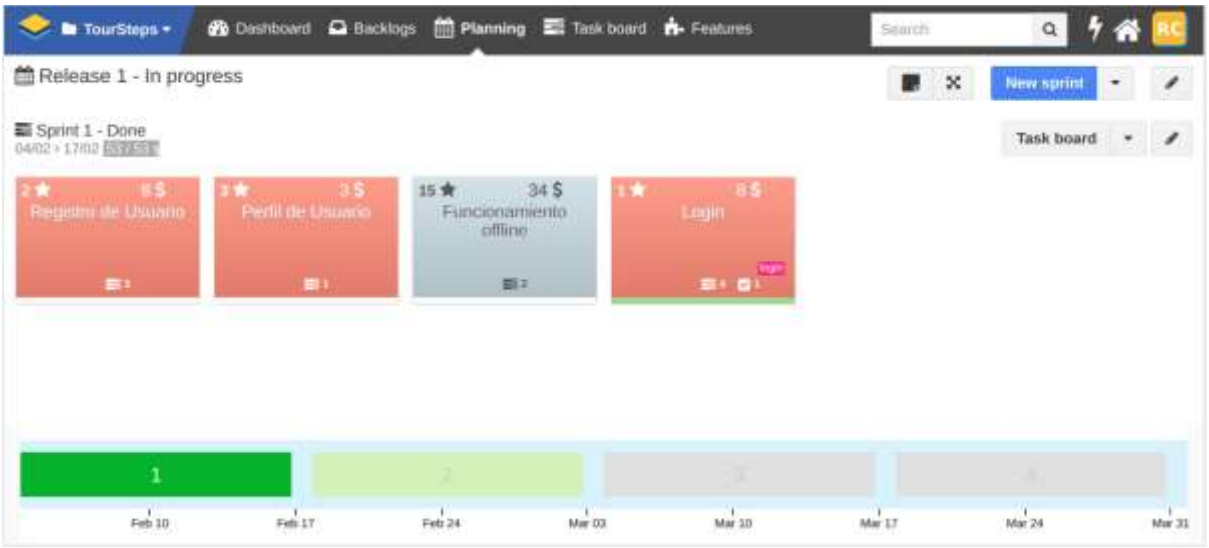


Figura 28. Lista de historias de usuario elegidas para el sprint #1 en IceScrum
Fuente: propia

- **Sprint backlog**

Primero el equipo de desarrollo (Development Team) mira nuestra lista de historias, que está actualizada y ordenada por prioridades y decide qué porción creen que pueden completar en unas dos semanas atendiendo a la meta para esa iteración. En la segunda parte, una vez decidido el trabajo a realizar, el Development Team se encarga de analizar y desmenuzar en tareas cada uno de las historias que han escogido, lo suficiente para empezar a trabajar, diseñan los prototipos y diagramas de UML requeridos.

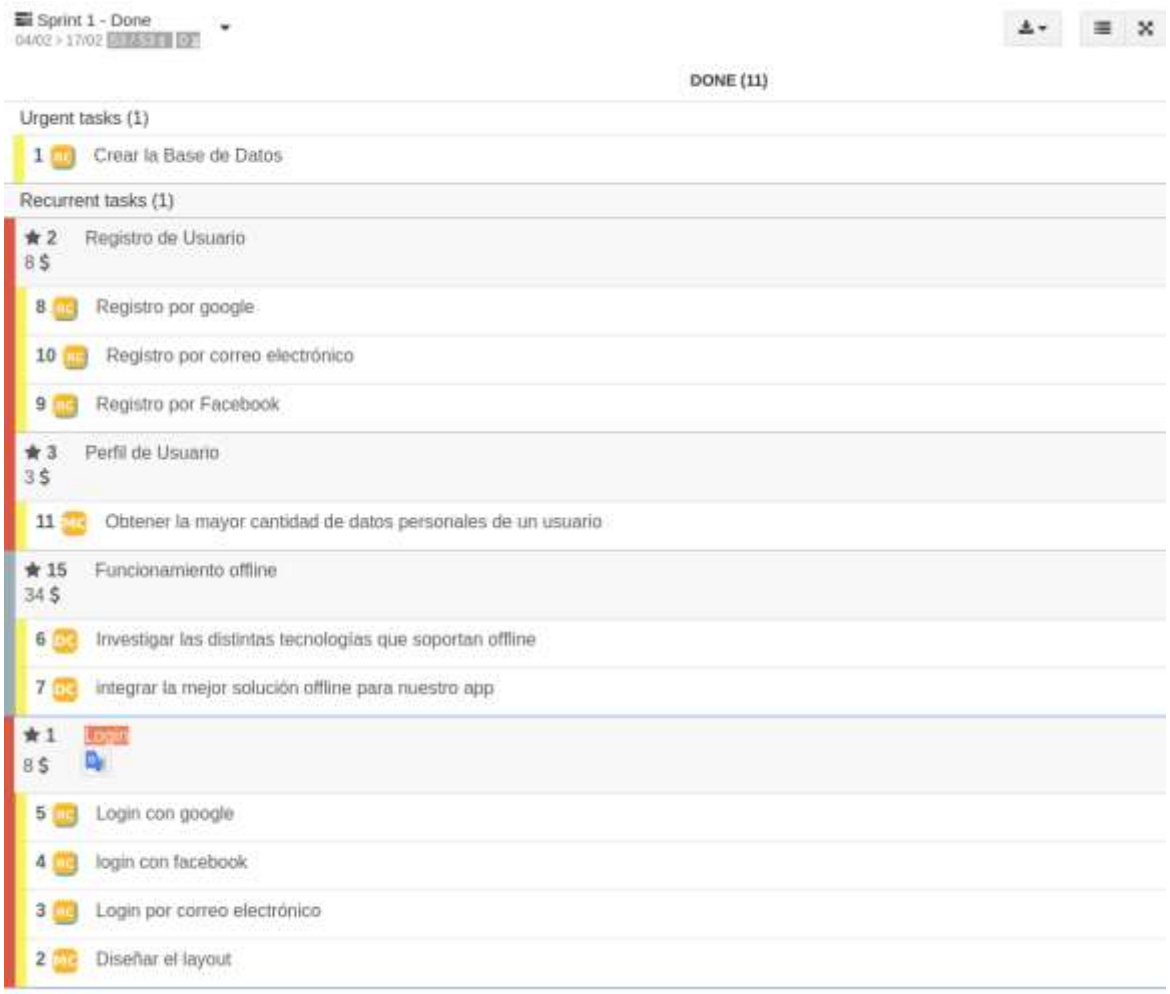


Figura 29. Sprint Backlog del sprint #1 en
Fuente: propia

La figura 28 muestra todas las tareas, junto con los requerimientos escogidos que completan la lista de tareas o Sprint Backlog.

- **Historias de usuario**

Ver el ANEXO A.1

- **Diseño de los prototipos de autenticación**

Los prototipos fueron diseñados con la herramienta Adobe XD, esto con el fin de tener una idea de cómo queremos que sea el proceso de autenticación a nuestra aplicación.

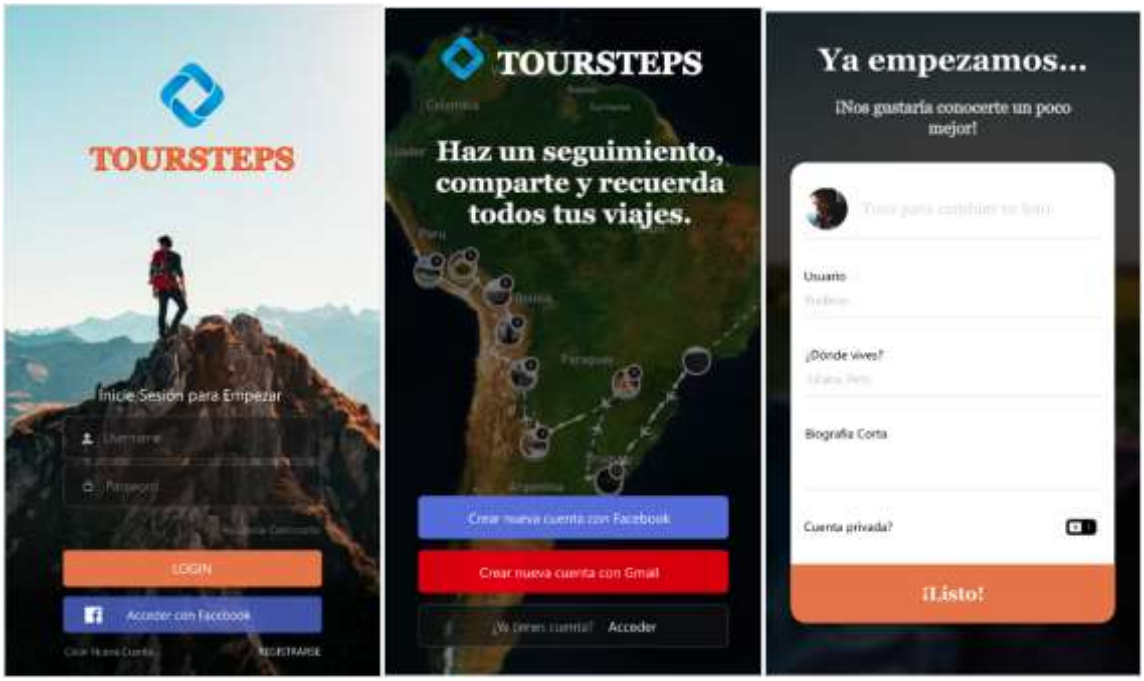


Figura 30, Prototipos del proceso de autenticación
Fuente: propia

- **Diagramas de UML**

Ver el ANEXO C, donde se muestra a detalle los diagramas UML para este sprint.

3.4.3.1.2. Desarrollo de sprint

- **Instalación y configuración de la librería nativescript-plugin-firebase**

Esta librería permite iniciar sesión en nuestra aplicación de las siguientes formas: anónima, correo electrónico y contraseña y Gmail. La instalación y configuración de esta librería es un proceso simple el cual se detalla en el ANEXO D.

- **Autenticación con google**

Una vez configurado firebase, escribimos las líneas de código para la autenticación con google, el siguiente fragmento de código nos muestra el método JavaScript encargado de realizar la autenticación.

```
loginGoogle() {
  firebase
   .login({
    type: firebase.LoginType.GOOGLE
```

```
    })
    .then(data => {
      const personCollection = this.$store.state.db.collection("person").doc(data.uid);
      personCollection.get().then(doc => {
        if (doc.exists) {
          this.$store.commit('UPDATE_DATA_USER', doc.data())
        } else {
          this.createPerson(data)
        }
      });
      loader.hide();
      setTimeout(() => {
        this.$navigateTo(Home);
      }, 1000);
      return Promise.resolve(JSON.stringify(data));
    })
    .catch(error => {
      console.error(error);
      return Promise.reject(error);
    });
},
```

- **Autenticación con correo y contraseña**

La autenticación con usuario y contraseña es más complejo, porque nosotros mismo tenemos que construir nuestro formulario de registro de usuario y logeo, el siguiente fragmento de código nos muestra el método registrarUsuario().

```
registerUsuario() {
  if (this.user.password != this.user.confirmPassword) {
    loader.hide();
    this.alert("Your passwords do not match.");
    return;
  }
  if (this.user.password.length < 6) {
    loader.hide();
    this.alert("Your password must at least 6 characters.");
```

```
    return;
  }
  var validator = require("email-validator");
 if (!validator.validate(this.user.email)) {
   loader.hide();
   this.alert("Please enter a valid email address.");
   return;
  }

  userService
    .register(this.user)
    .then(() => {
      loader.hide();
      this.alert("Your account was successfully created.");
      this.isLoggingIn = true;
    })
    .catch(err => {
      loader.hide();
      this.alert(err);
    });
},
```

Una vez registrado el usuario tiene que loguearse ingresando su correo electrónico y contraseña con la que se registró anteriormente.

3.4.3.1.3. Sprint review

Al final del sprint #1 se obtiene el primer incremento que cumple con la meta marcada de la aplicación TourStep la cual permite la autenticación y registro del usuario por google y correo electrónico.

Al final del Sprint, el Product Owner organizó una reunión que se llama Sprint Review, donde invitó a Stakeholders y al equipo de desarrollo para mostrar el Incremento terminado; los stakeholders dieron los feedback al Product Owner sobre el Incremento.

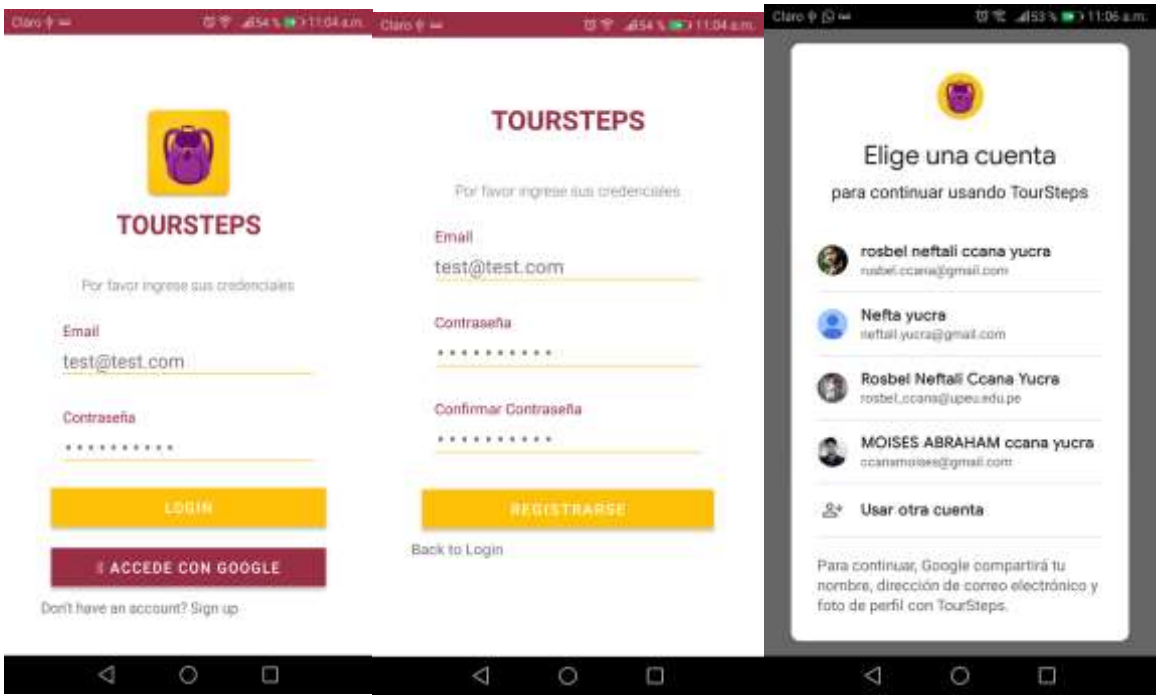


Figura 31. Producto final del sprint #1
Fuente: Adaptación propia

3.4.3.2. Sprint #2

3.4.3.2.1. Sprint planning

En el Sprint #2 se realizó nuevamente el Sprint Planning, donde el Product Owner explicó al equipo de desarrolló que la prioridad es que los usuarios autenticados pueden gestionar sus viajes futuros, pasados o actuales. Si los viajes son actuales entonces la ubicación del usuario tiene registrarse automáticamente y mostrarle en un mapa integrado en la aplicación.

Figura 32. Lista de historias de usuario elegidas para el sprint #2 en IceScrum
Fuente: propia

- **Sprint backlog**

Primero el equipo de desarrollo (Development Team) nuevamente mira nuestra lista de historias, que está actualizada y ordenada por prioridades y decide qué porción creen que pueden completar en unas dos semanas atendiendo a la meta para esa iteración. En la segunda parte, una vez decidido el trabajo a realizar, el Development Team se encarga de analizar y desmenuzar en tareas cada uno de las historias que han escogido, lo suficiente para empezar a trabajar, diseña nuevamente los prototipos y diagramas de UML requeridos.

Al final nuevamente tenemos un Sprint Backlog como se puede mostrar en la figura 32.

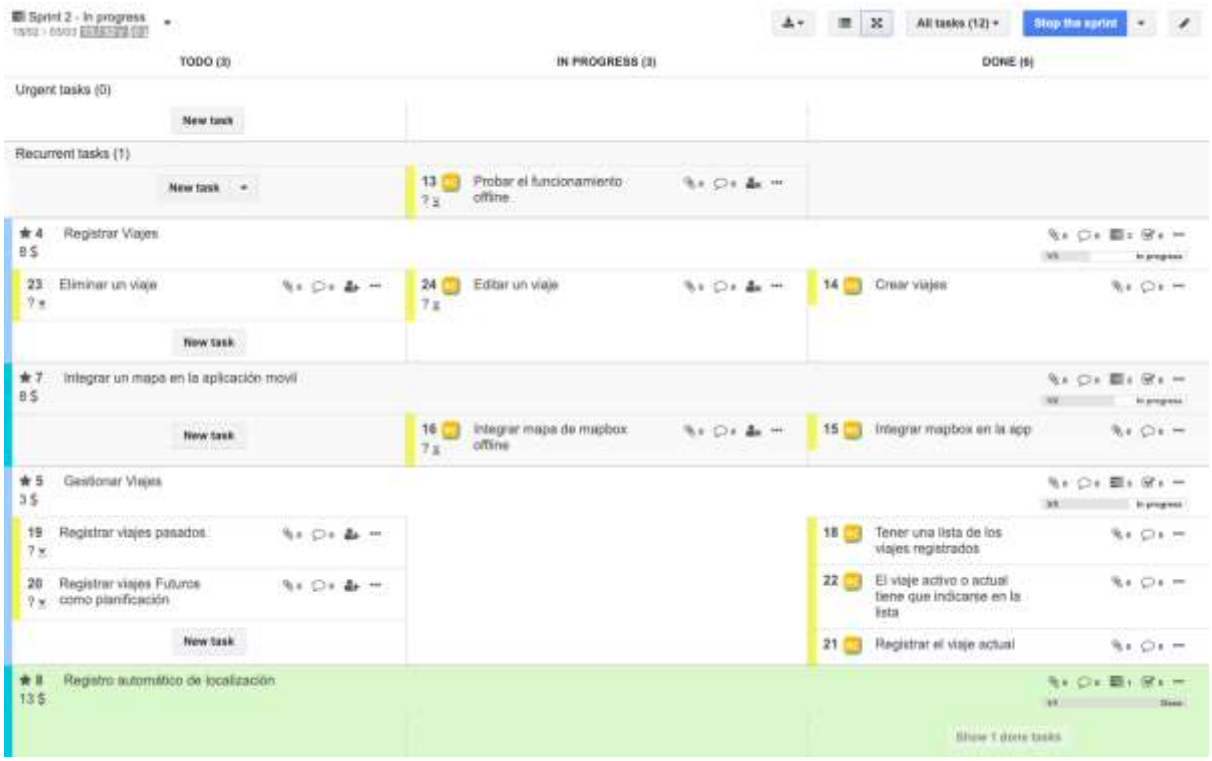


Figura 33. Sprint Backlog del sprint #2 en IceScrum (fuente propia)

- **Historias de usuario**

Ver el ANEXO A.2

- **Diseño de prototipos**

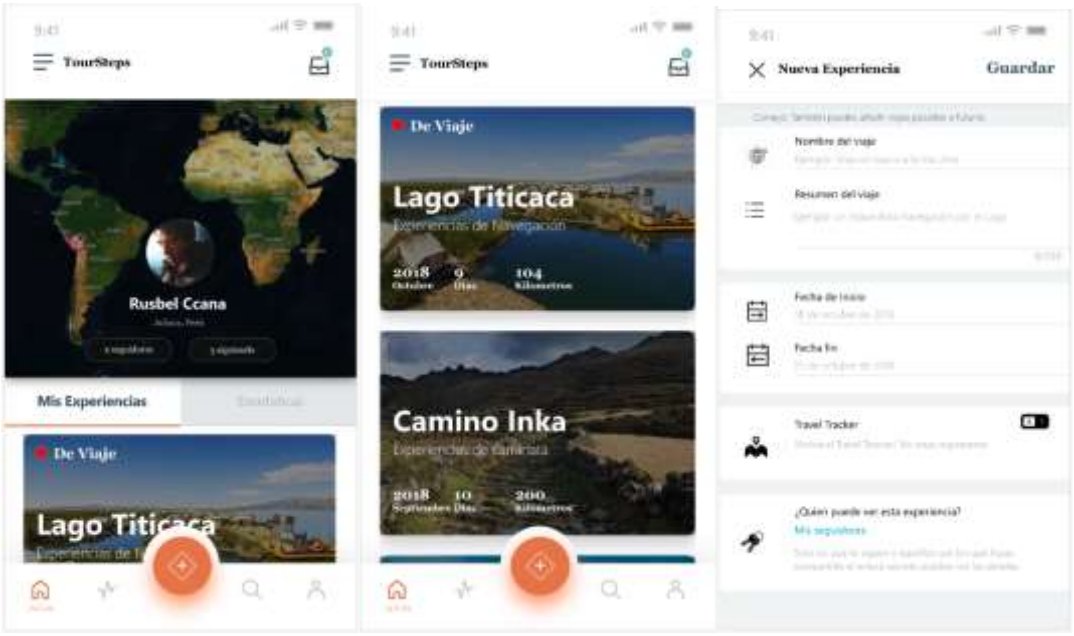


Figura 34. Prototipos del proceso de gestión de viajes
Fuente: propia

- **Diagramas de UML**

Ver el ANEXO C, donde se muestra a detalle los diagramas UML para este sprint.

3.4.3.2.2. Desarrolló de sprint

- **Instalación y configuración de la librería nativescript mapbox**

Esta librería permite mostrar una vista de mapa en aplicaciones Nativescript. La instalación y configuración de esta librería es un proceso complejo el cual se detalla en el ANEXO E.

- **Registro de viajes recuperando la ubicación actual del usuario**

La recuperación de la ubicación actual de un usuario para el registro de un viaje se puede realizar de varias maneras ya sea haciendo uso de la API Geolocation proporcionada por Nativescript, por la API de google map o a través de módulos nativos (librerías creadas con lenguajes de programación Java o Swift) que pueden ser usados con código JavaScript. En nuestro caso usamos nativescript-mapbox.

El registro de un viaje se realiza por los usuarios logueados, las siguientes imágenes muestran fragmentos de código para el registro de viaje con la ubicación actual del usuario.

```
steps-master ▸ app ▸ pages ▸ Trip.vue ▸ {} "Trip.vue" ▸ script
75      const map = new Mapbox()
76      map.getUserLocation().then( d => {
77          console.log("Current user location: " +  d.location.lat + ", " + d.location.lng);
78          console.log("Current user speed: " +  d);
79          this.location = d;
80        }
```

Figura 35. Método para obtener la ubicación actual de un usuario con la librería mapbox
Fuente: propia

En la figura 34, se observa cómo podemos obtener la ubicación del usuario con la librería nativescript-mapbox, el método getUserLocation() de mapbox nos permite tener la ubicación y guardarlo en la variable this.location. Con la ubicación ya obtenida del usuario se registra un viaje. Como resultado tenemos todas las tareas del sprint backlog terminadas.

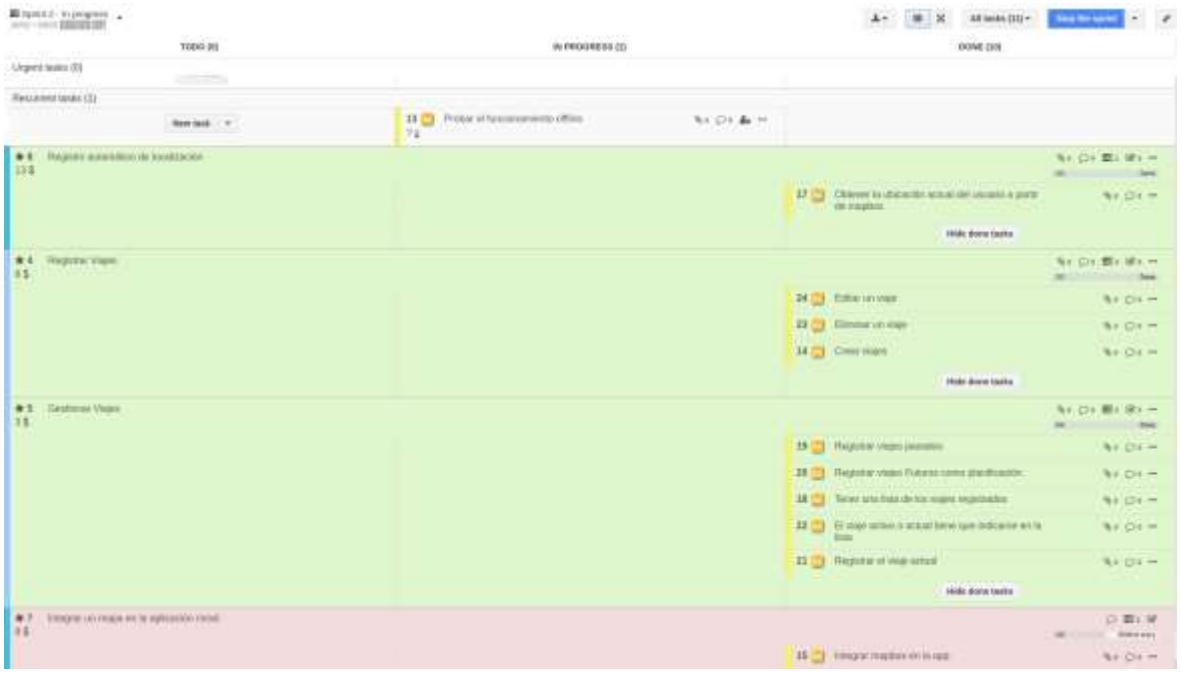


Figura 36, Lista de tareas del sprint backlog terminadas
Fuente: propia

3.4.3.2.3. Sprint review

Al final del sprint #2 se obtiene el incremento que cumple con la meta marcada de la aplicación TourStep la cual gestionar un viaje registrandolo como actual, pasado o futuro y la integración de un mapa de manera offline.

Al final del Sprint, el Product Owner organizó una nueva reunión Sprint Review, donde invitó a Stakeholders y al equipo de desarrollo para mostrar el incremento terminado; los stakeholders dieron los feedback al Product Owner sobre el Incremento.

En este sprint quedó pendiente el funcionamiento del mapa de manera offline, porque solo se logró integrar el mapo online.

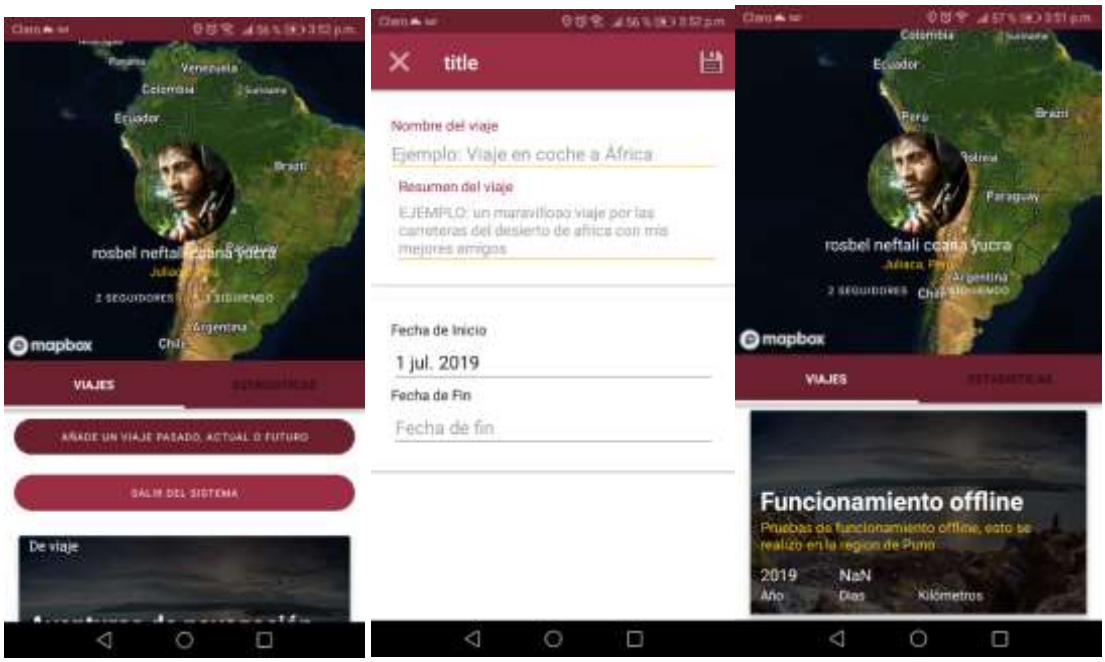


Figura 37. Producto final del sprint #2
Fuente: propia

3.4.3.3. Sprint #3

3.4.3.3.1. Sprint planning

En el Sprint #3 se realizó nuevamente el primer Sprint Planning, donde el Product Owner explicó al equipo de desarrolló que la prioridad es que se registre las coordenadas por donde recorrió un usuario de manera automática por medio del GPS del dispositivo móvil y tener un mapa offline donde se visualice las experiencias registradas de un viaje.

Figura 38. Lista de historias de usuario elegidas para el sprint #3 en IceScrum
Fuente: propia

- **Sprint backlog**

Bajo la meta propuesta para este incremente nuevamente el equipo de desarrollo (Development Team) mira la lista de historias, que está actualizada y ordenada por prioridades y decide qué porción creen que pueden completar en unas dos semanas. En la segunda parte, una vez decidido el trabajo a realizar, el Development Team se encarga de analizar y desmenuzar en tareas cada uno de las historias que han escogido, lo suficiente para empezar a trabajar.

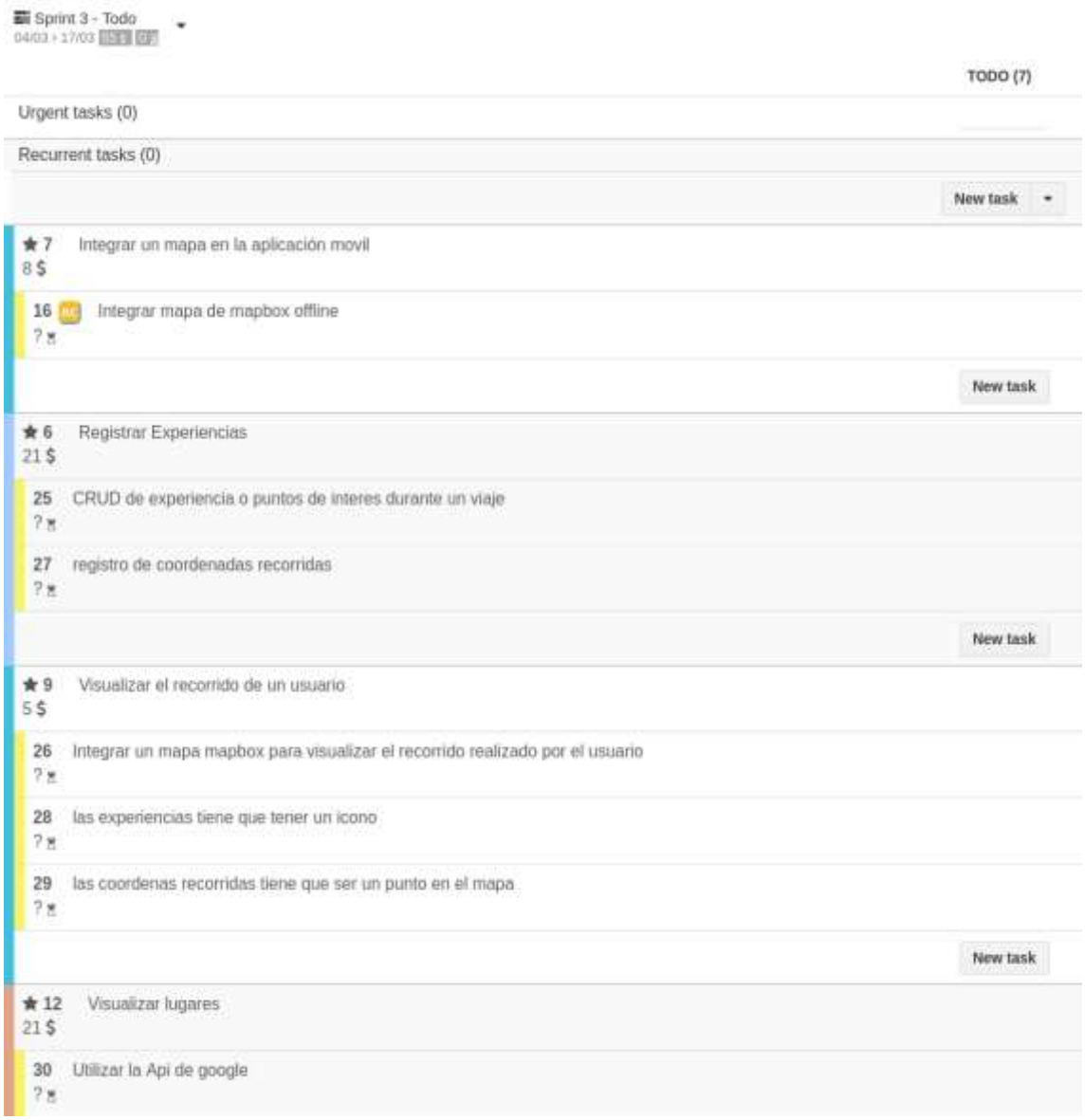


Figura 39. Sprint Backlog del sprint #3 en IceScrum
Fuente: propia

En este sprint también se incluyó la historia que no se pudo completar en el anterior sprint como es el caso de la integración de un mapa offline, se diseñan los prototipos y los diagramas de UML requeridos.

- **Historias de usuario**

Ver el ANEXO A.3

- **Diseño de prototipos**

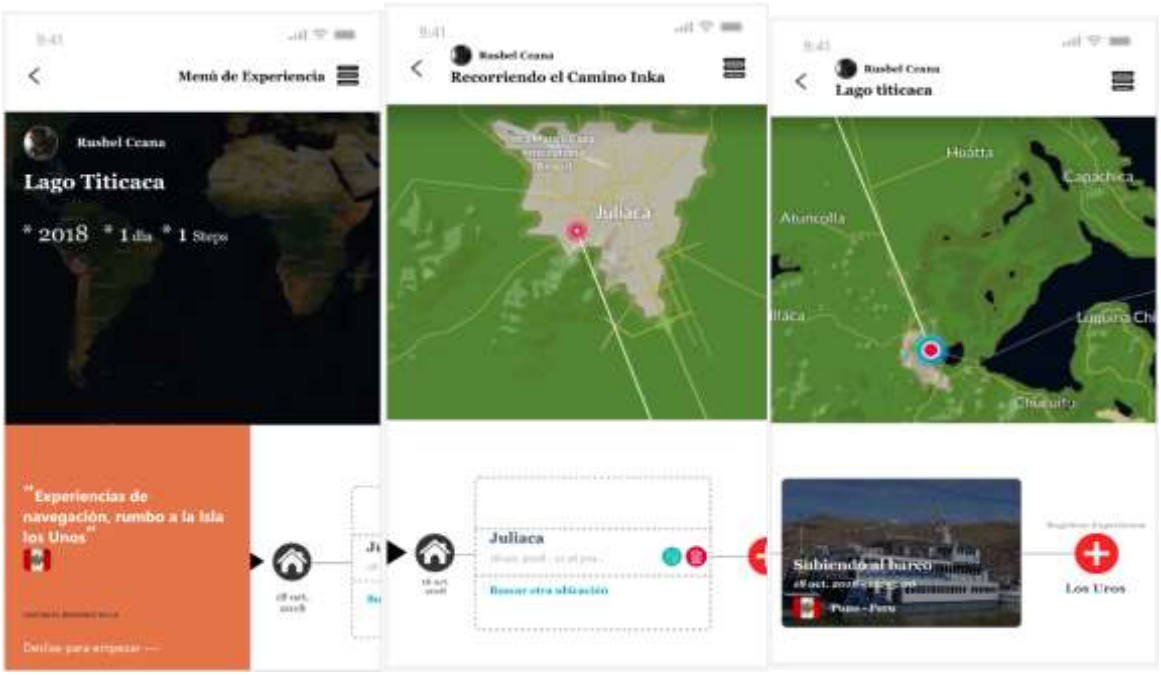


Figura 40. Prototipos del proceso de registro de coordenadas y experiencias en viajes
Fuente: propia

- **Diseño de UML**

Ver el ANEXO C.3, donde se muestra a detalle los diagramas UML para este sprint.

3.4.3.3.2. Desarrolló del sprint

- **Configuración de mapa offline**

La integración de un mapa offline fue crucial para el cumplimiento del incremento en este sprint. Como ya se integró anteriormente un mapa con la librería nativescript-mapbox, solo se realizó las configuraciones necesarias para que funcione, ver anexo E.1.

- **Registro automático de lugares recorridos en un viaje activo**

Para el registro la aplicación TourSteps hace uso de la APi de Google Maps Geocode por medio de una petición web a la siguiente URL:

```
https://maps.googleapis.com/maps/api/geocode/output?parametros
```

En la URL anterior se debe pasar los siguientes parámetros:

- output: formato de la respuesta puede ser xml o json
- latlng: la latitud y longitud desde donde se realiza la búsqueda
- key: la clave de la API correspondiente a la aplicación (Ver ANEXO F).

El siguiente fragmento de código muestra el método JavaScript encargado de recuperar esta lista de sitios directamente de la API de Google Maps geocode y el registro correspondiente del lugar.

```
getLocation(map){
    this.$store.dispatch('tripGet')
    let mapbox = new Mapbox()
    mapbox.getUserLocation().then( userLocation=> {
     console.log("Current user location: " +  JSON.parse(JSON.stringify(userLocation.location.lat)));
     console.log("Current user location: " +  JSON.parse(JSON.stringify(userLocation.location.lng)));
     console.log("Current user speed: " +  userLocation.speed);
     this.location = userLocation.location

     map.setCenter({
      lat: userLocation.location.lat, // mandatory
      lng: userLocation.location.lng, // mandatory
      animated: true // default true
     })

     let puntoA = Math.pow(userLocation.location.lat, 2) - 2*(userLocation.location.lat*userLocation.location.lng) +
Math.pow(userLocation.location.lng, 2)
     console.log("el valor del punto A", puntoA);

     let Puntos=[]

axios.get(`https://maps.googleapis.com/maps/api/geocode/json?latlng=${userLocation.location.lat},${userLocatio
n.lng}&key=AIzaSyAfFTpwJLzH2vNEgk9qwUAbqWtE8xRSr0o`)
      .then(response => {
```

```
            // Respuesta de los lugares cercanos a la ubicación del usuario
//
          })
          .catch(error => {
            console.log('error', error)
          })
        })
    },
```

El método geoLocation recibe el parámetro map, este parámetro JavaScript es un objeto con la ubicación actual del usuario (latitud y longitud).

Una vez que se obtiene la respuesta se realiza el registro del lugar, para esto primeramente se busca el lugar más cercano a la ubicación del usuario mediante la distancia entre dos puntos y el punto más cercano se toma para guardar en la base de datos, como se muestra en el siguiente fragmento de código.

```
 let puntoB
// calculando las distancias entre dos puntos
 response.data.results.map(p=>{
   let punto = p.geometry.location
   puntoB = Math.pow(punto.lat, 2) -2*(punto.lat*punto.lng) + Math.pow(punto.lng, 2)
   Puntos.push({distancia:puntoA+puntoB, data:p})
 })
  const filter1='administrative_area_level_1'
  const filter2='administrative_area_level_2'
  const filter3='administrative_area_level_3'
  setTimeout(() => {
    let puntos_filtrados= Puntos.filter(m=>this.filterType(m.data.types, filter3)===filter3 || this.filterType(m.data.types,
filter2)===filter2 || this.filterType(m.data.types, filter1)===filter1)
              puntos_filtrados.sort((a,b)=>{
                return a.distancia-b.distancia
              })
             let new_point = {
              name:puntos_filtrados[0].data.formatted_address,
              date:new Date(),
              //start_time: new Date().toLocaleTimeString(),
              location:userLocation.location,
              //photos:[],
              state:true,
              //history:''
              }
```

```
    let trip_temp = this.trip.trips.find(x=>x.state===true)

    if (trip_temp) {
      // Metodo que registra el nuevo lugar.
      this.$store.dispatch('point_autoPost', {trip_id: trip_temp.id, step:new_point})
    }
  }, 1000);
```

- **Registro y visualización de las experiencias en el mapa**

Anteriormente vimos cómo se registra los lugares por donde recorre el usuario sin la necesidad que este realice alguna acción en la aplicación, en esta parte se desarrolló el registro de experiencias que el usuario tiene durante su viaje, y la visualización de este en nuestro mapa ya integrado.

Nuestro sprint término como todas las tareas programadas en el sprint backlog.

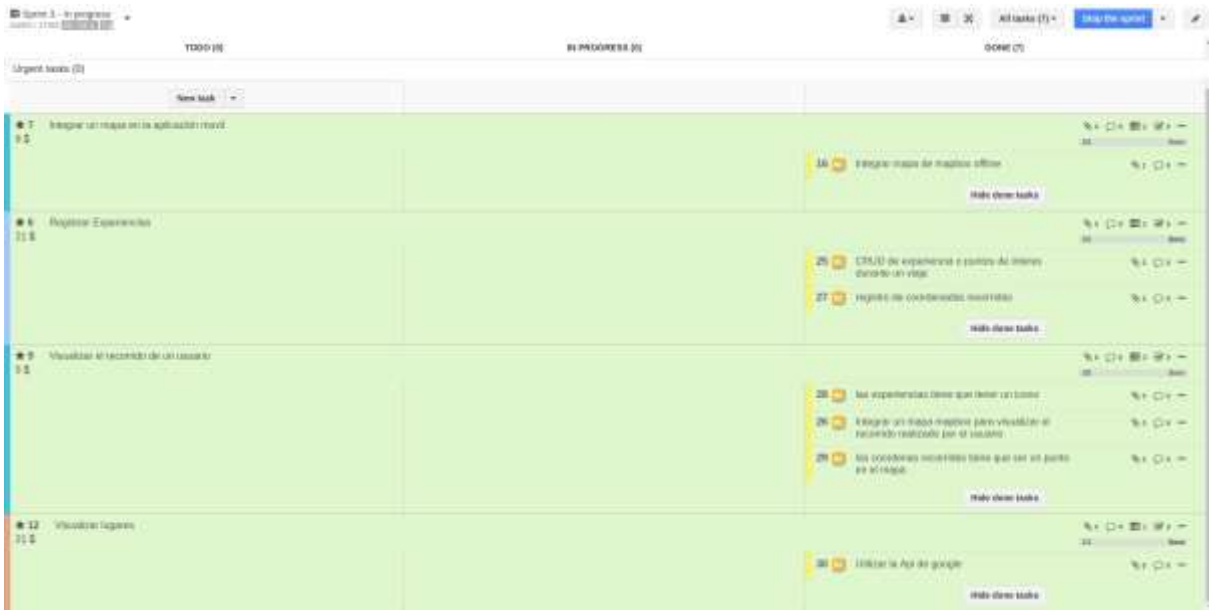


Figura 41. Tablero del sprint backlog con todas las tareas terminadas
Fuente: propia

3.4.3.3.3. Sprint review

Al final del sprint #3 se obtiene el incremento que cumple con la meta marcada de la aplicación TourStep la cual es registrar automáticamente los lugares por donde recorre un usuario y las experiencias de dicho recorrido, por ultimo mostrar todos estos registros en un mapa donde se visualiza la ruta recorrida con las experiencias marcadas.

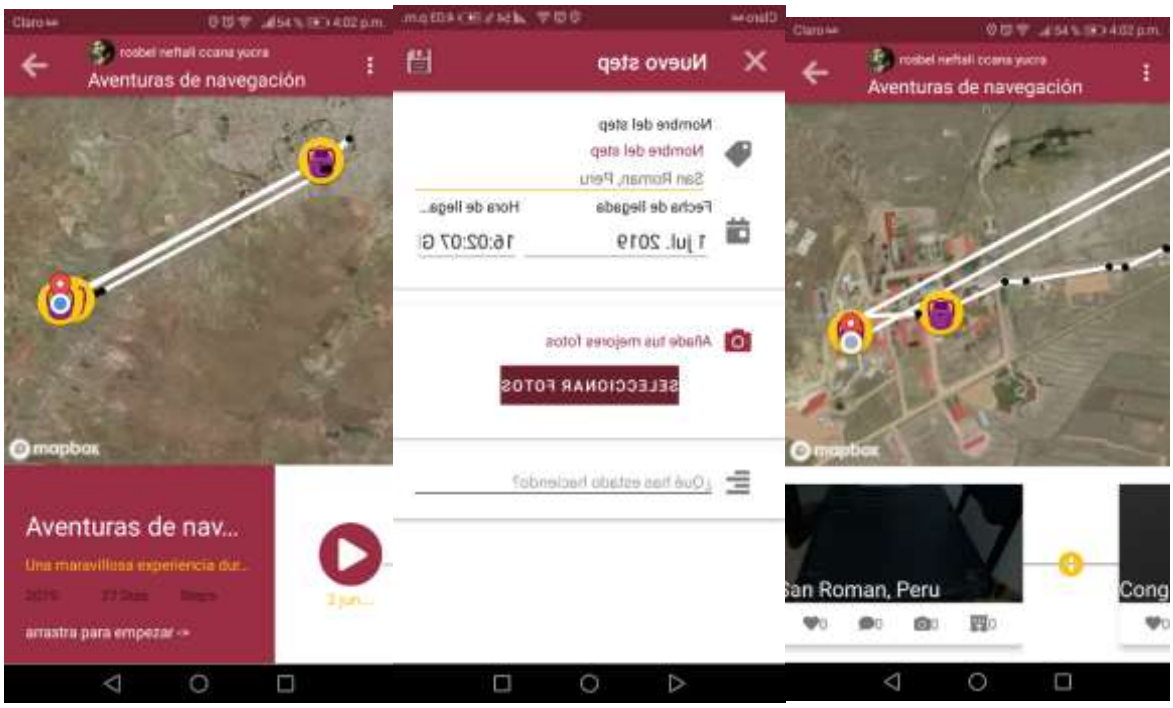


Figura 42. Producto final del sprint #3
Fuente: propia

En Product Owner organizó una nueva reunión Sprint Review, donde invitó a Stakeholders y al equipo de desarrollo para mostrar el incremento terminado; los stakeholders dieron los feedback al Product Owner sobre el Incremento.

3.4.3.4. Sprint #4

3.4.3.4.1. Sprint planning

En el Sprint #4 se realizó nuevamente el primer Sprint Planning, donde el Product Owner explicó al equipo de desarrolló que la prioridad es realizar comentarios a las experiencias registradas en un viaje y tener un álbum de los viajes.

Figura 43. Lista de historias de usuario elegidas para el sprint #3 en IceScrum
Fuente: propia

- **Sprint backlog**

Bajo la meta propuesta para este incremente nuevamente el Development Team mira la lista de historias, que está actualizada y ordenada por prioridades y decide qué porción creen que pueden completar en unas dos semanas. En la segunda parte, una vez decidido el trabajo a realizar, el Development Team se encarga de analizar y desmenuzar en tareas cada uno de las historias que han escogido, lo suficiente para empezar a trabajar.

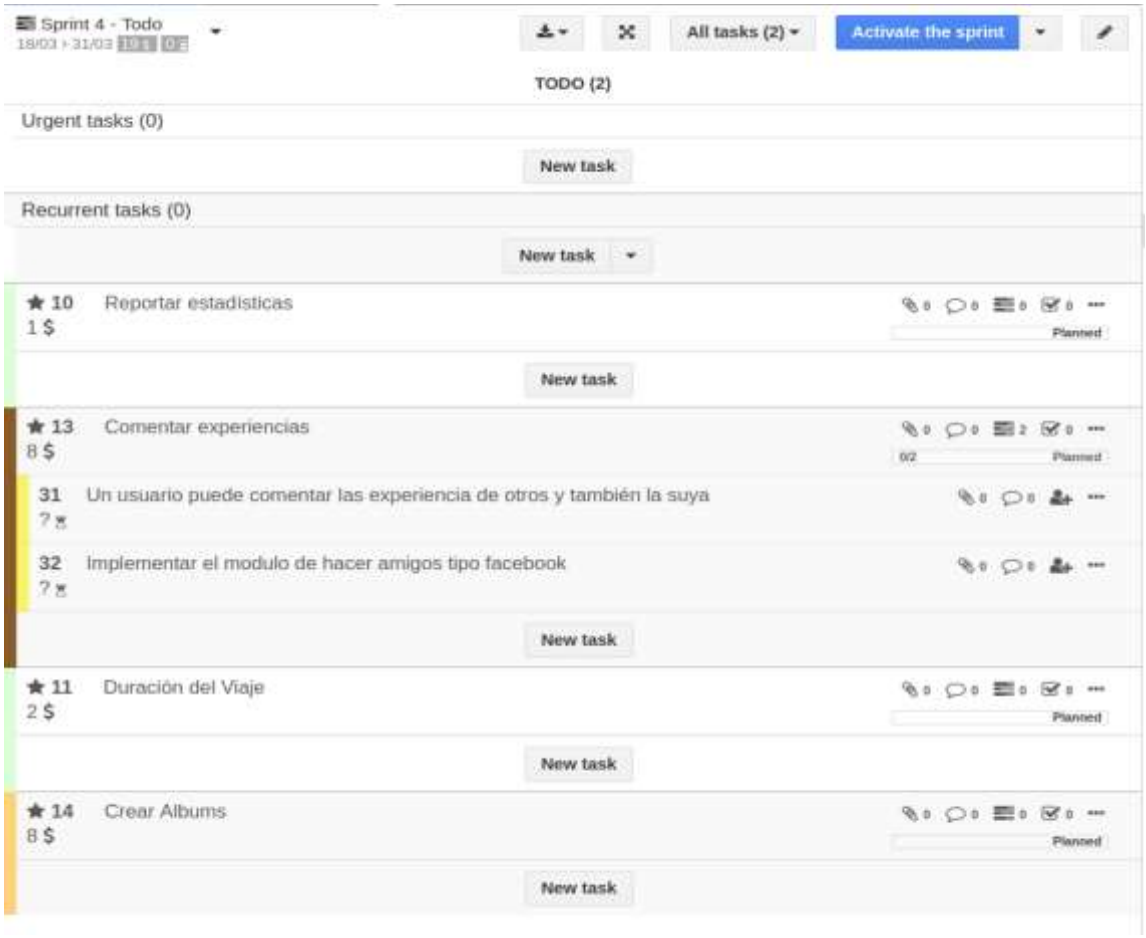


Figura 44. Sprint Backlog del sprint #4 en IceScrum
Fuente: propia

Una vez definida las tareas en el sprint backlog se diseñaron los prototipos y los diagramas de UML requeridos.

- **Historias de usuario**

Ver el ANEXO A.4

- **Diseño de prototipos**

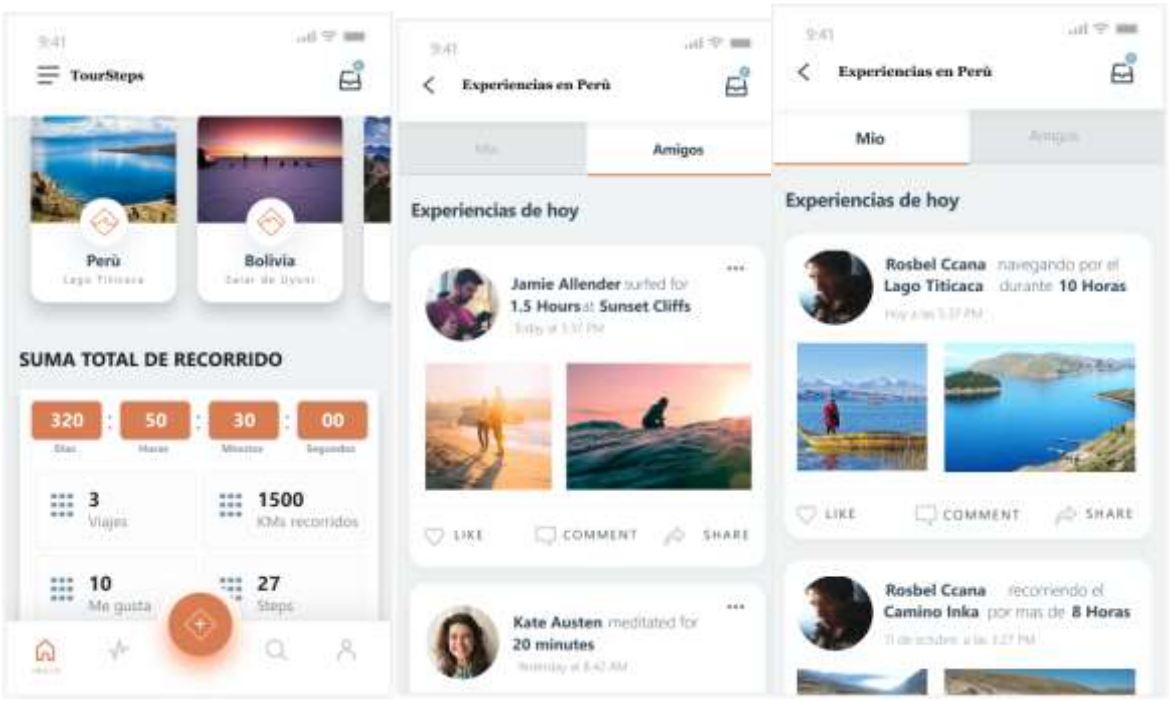


Figura 45. Prototipos del proceso de comentarios en las experiencias registradas
Fuente: propia

- **Diseño de diagrama UML**

Ver el ANEXO C.4, donde se muestra a detalle los diagramas UML para este sprint.

3.4.3.4.2. Desarrolló del sprint

- **Realizar comentarios en las experiencias**

Cómo estamos utilizando firestore para la gestión de nuestros datos, también tiene la característica de cambios en tiempo real. Esto nos facilitó las cosas, para ver la configuración de firebase ver el ANEXO D.

- **Gestión de álbum como recuerdo de un viaje**

En esta parte trabajos un poco duro, pues se decidió que el álbum de viajes que se genera con toda la información de un viaje, cuando este termine debe estar un servidor web. Para esto el equipo de desarrolló optó, por Vue.js, los motivos, porque nuestra aplicación se desarrolló con este framework y el método de desarrolló era casi el mismo.

Como resultado se tiene un sprint backlog con todas sus tareas terminadas, las experiencias registradas ahora pueden ser comentadas por el mismo usuario u otro y lo más

importante, el usuario puede descargar un álbum de recuerdos con toda la información de su viaje.

3.4.3.4.3. Sprint review

Al final del sprint #4 se obtiene el incremento que cumple con la meta final de la aplicación TourStep.

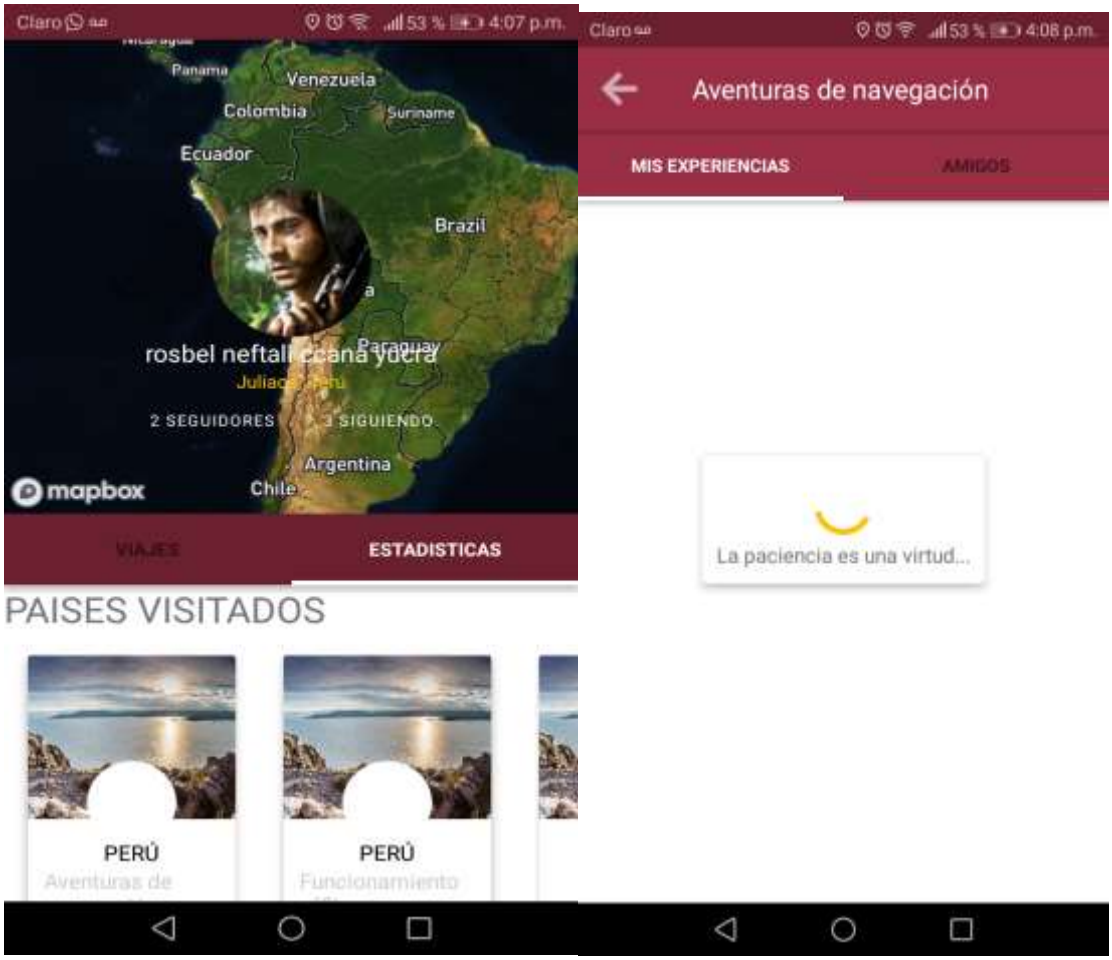


Figura 46. Producto final del sprint #4
Fuente: propia

En Product Owner organizó una nueva reunión Sprint Review, donde invitó a Stakeholders y al equipo de desarrollo para mostrar el incremento terminado; los stakeholders dieron los feedback al Product Owner sobre el Incremento.

En este caso nos pidieron mejorar la gestión de álbums.

3.4.4. Fase de entrega

En esta parte nuestra aplicación TourSteps está lista para su salida a producción y publicación en el Google Play (tienda oficial para aplicaciones Android). Pero primeramente se pasa a realizar las pruebas de funcionamiento offline puesto que nuestro desarrollo está basado en offline-first.

Cabe mencionar que nuestro producto se está entregando con dos requerimientos pendientes: La Creación de álbum de recuerdo y añadir amigos para ver sus viajes y seguirles.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

En este capítulo se describe los resultados que se obtuvo durante el desarrollo de nuestro proyecto, analizaremos si se llegaron a cumplir los objetivos propuestos a un principio.

4.1. Resultados

Nuestro objetivo fue desarrollar la aplicación móvil TourSteps, gracias a que nativescript está dentro del desarrollo de aplicaciones interpretadas, una vez concluida la etapa de desarrollo, tenemos la opción de compilar nuestra primera versión de la aplicación para los sistemas operativos iOS y Android, sin la necesidad de hacer configuraciones o cambios complicados. Nuestra aplicación móvil es multiplataforma con apariencia nativa y un rendimiento de alta calidad para dispositivos de gama media y alta.

En el apartado de Fase de planificación se describe a detalle todo lo realizado para cumpli el objetivo especifico 1, como resultado de esto se obtuvo un product backlog que se gestiona en icescrum, nuestros prototipos diseñados en adobe XD y los diagramas de clases.

En cuanto al objetivo específico 2, se tiene como resultado una aplicación móvil multiplataforma, con soporte offline, mapas integrados con Mapbox. La implementación de mapas con mapbox fue todo un éxito, gracias a que mapbox está estrictamente orientado a desarrolladores y es bien personalizable; además de eso, nos proporciona métodos muy simples para implementar, nuestra aplicación funciona con mapas offline que se descargan por regiones y esto ayudó a que la experiencia de nuestros usuarios sea satisfactoria.

Las pruebas de funcionamiento offline de la aplicación se realizaron satisfactoriamente, porque firebase almacena los cambios de dato en el caché local del dispositivo. Y cuando se conecta a internet se sincroniza de manera automática.

En la figura 47 se puede observar la implementación del mapa, esto se probó en un dispositivo Huawei Y7 con Android 8.0 Oreo.

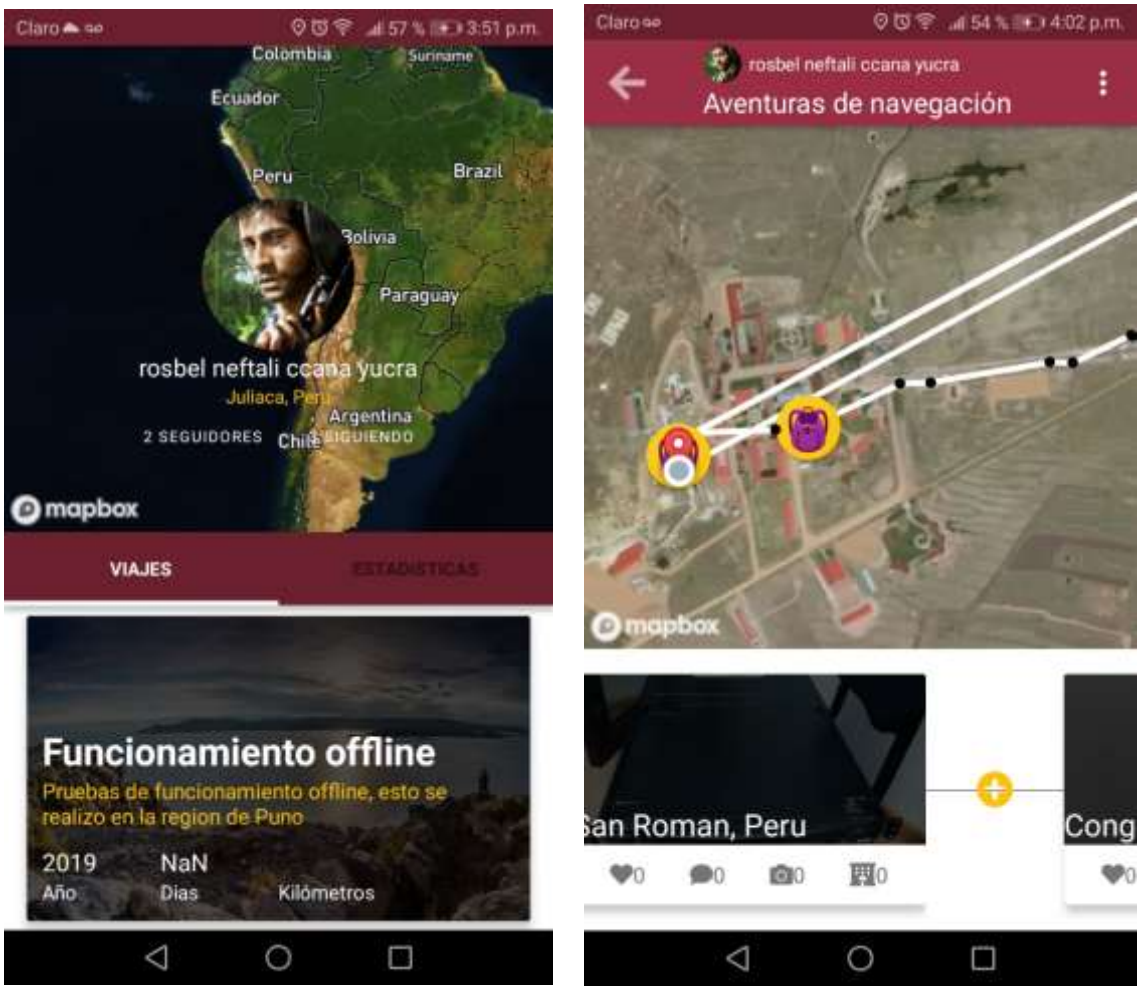


Figura 47. Aplicación TourSteps, funcionando del mapa offline en un dispositivo Huawei Y7
Fuente: propia

Por último, podemos decir que la librería nativescript-vue es un excelente framework de desarrollo móvil para aquellos desarrolladores que estamos más orientados a la web. Durante el proceso comprobamos que solo existen 3 diferencias grandes al momento de desarrollar aplicación web y aplicaciones móvil.

La primera diferencia es al momento de usar vue, para desarrolló móvil se usa el paquete nativescript-vue en lugar de la biblioteca estándar de Vue.js

Figura 48. Diferencia #1 del desarrolló móvil y web con vue.
Fuente: Adaptada de nativescript.org/vue

La segunda diferencia es al momento de crear la instancia de Vue, como nativescript no manipula el DOM ya no es necesario pasar el parámetro div.

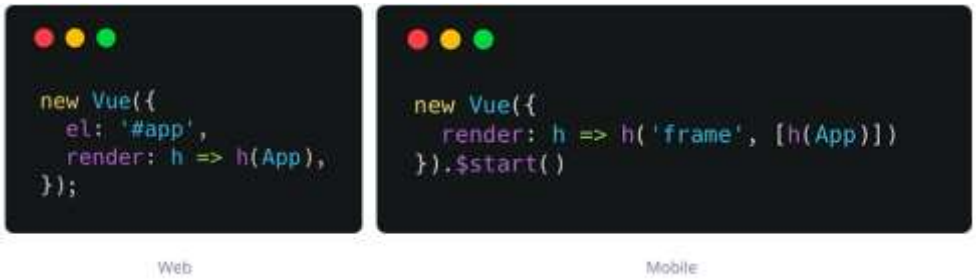


Figura 49. Diferencia #2 del desarrolló móvil y web con vue.
Fuente: Adaptada de nativescript.org/vue

La tercera y última diferencia es al momento de escribir código en la plantilla, en web se usa html semántico mientras que en móvil se usa los módulos de nativescript con XML.

```
<template>
   <v-card color="tile" tile class="pa-4">
    <h1>Student Home</h1>
    <p align="left">Here, you'll find a list of grades per student.</p>
   </v-card>
</template>
```

```
<template>
  <StackLayout ref="initialContainer" class="initialContainer">
    <Label text="Elocute" class="initial-label">
   <StackLayout @tap="welcome" class="initial-button">
   <Label text="welcome" class="initial-button-label">
   </StackLayout>
  </StackLayout>
</template>
```

Figura 50. Diferencia #3 del desarrolló móvil y web con vue.
Fuente: Adaptada de nativescript.org/vue

4.2. Discusiones

Uno de los aportes importantes de la investigación es el enfoque de desarrollo offline-first que se aplica, normalmente como desarrolladores siempre pensamos que nuestros clientes o usuarios finales, usan nuestros productos en las mismas condiciones que nosotros con acceso a internet 4G o más altos. Sin embargo, en este caso decidimos desarrollar la

aplicación priorizando la disponibilidad de datos sin importar la condición de la red en la que se encuentran nuestros usuarios.

El principal problema que se genera cuando desarrollamos con un enfoque offline-first son los conflictos que existen cuando se sincroniza los datos, para solucionar este problema optamos usar firebase, ya la forma solucionar los conflictos es conservando el último cambio detectado.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Terminado el desarrollo de la aplicación móvil TourSteps que permite el registro automático de rutas y lugares visitados durante las experiencias turísticas, basado en el paradigma offline first, se llegó a las siguientes conclusiones

En cuanto al objetivo general de desarrollar una aplicación móvil para el registro automático de rutas y lugares visitados durante las experiencias turísticas, basado en el paradigma offline first, se siguió el modelo de desarrollo ágil en el marco scrum, la aplicación TourSteps fue evolucionando durante el proceso de desarrollo que consta de 3 fases: planificación, desarrollo y entrega. Gracias a esto se obtuvo sin lugar a dudas una aplicación móvil multiplataforma que supera su propuesta inicial mediante una navegación intuitiva, interfaz de usuario simple, gran performance y funcionalidades adicionales que no se plantearon en el momento de la concepción de la idea inicial del proyecto.

Respecto a nuestro primero objetivo específico, se concluye que la herramienta de gestión IceScrum es muy óptima para la construcción del Product Backlog; y también para gestionar los Sprint Backlog de cada iteración o sprint.

Respecto al segundo objetivo específico se concluye, que el uso de la librería nativescript-mapbox fue de gran ayuda por estar enfocado para el uso de los desarrolladores y por sus métodos que son simples de implementar.

Respecto al tercer objetivo específico se concluye, que las pruebas de funcionamiento offline-first fueron satisfactorias ya que nuestro aplicativo tiene integrado mapas offline que se descargan de acuerdo a la región donde nos encontremos y la resolución de conflictos de sincronización está bien gestionada por firebase.

Por último, se tiene una arquitectura de desarrollo de aplicación móviles, para desarrolladores con conocimiento en sistemas web. Al mismo tiempo con soporte offline.

5.2. Recomendaciones

- Se recomienda realizar evaluaciones a la arquitectura propuesta mediante diferentes escenarios de los atributos de calidad: performance, disponibilidad, seguridad y otros.

- También se propone realizar un estudio comparativo de los framework de desarrollo de aplicaciones interpretadas, ya que la elección de nativescript-vue para este proyecto fue estrictamente porque nuestro equipo de desarrollo trabaja con Vue.js
- Se propone construir un método de elección para saber que protocolo de sincronización usar cuando se desarrollan aplicaciones offline-first, en nuestro caso solo analizamos los requerimientos.

REFERENCIAS

Hurtado I. (2016) Hábitos de uso de las aplicaciones móviles de viaje en los Millennuals adultos residentes en Lima Metropolitana (Tesis de pregrado). Universidad San Ignacio de Loyola. Lima.

DITENDRIA. (2018). Informe Mobile en España y en Mundo. Recuperado de https://mktefa.ditrendia.es/hubfs/Ditrendia-Informe%20Mobile%202018.pdf

Robles J. (2015) Desarrollo de una aplicación para equipos android, basada en la geolocalización para obtener información de atractivos turísticos en la ciudad de Tulcán (Tesis de Maestría). Pontificia Universidad Católica de Ecuador. Ecuador

Becerra, E., Silva, M., & Rocha, A. (2012, January 9). El turismo en la sociedad de la información, Un abordaje conceptual sobre el...: EBSCOhost. Obtenido el June 2, 2015, de http://web.a.ebscohost.com/ehost/detail/detail?sid=426af8c0-4d18-4764- bcb8-e447cb6d3134%40sessionmgr4004&vid=0&hid=4107&bdata=Jmxhbmc9ZX Mmc2l0ZT1laG9zdC1saXZl#db=fua&AN=89439689

Feyerke (2014). Offline First: faster, more fun, and more robust (web) apps. De https://fronteers.nl/congres/2014/sessions/alex-feyerke-offline-first-faster-more-fun-and-more-robust-web-apps

Sanchez E. (2012) Integración de Foursquare y Geolocalización en una Aplicación Móvil para la Creación de Rutas Turísticas, Universidad Politécnica de Valencia.

Spandana, Prithvi & K a, Shreyas & Kumar M, Anand. (2015). AN EFFECTIVE AND RELIABLE TRACKING SYSTEM FOR MONITORING FIELD EMPLOYEE USING GPS-ENABLED MOBILE DEVICES. International Journal of Advanced Research in Information and Communication Engineering 2321-8762. 3. 33-36.

Vanhala J. (2017). Implementing an Offline First Web Application

Gill O. (2018). Using React Native for mobile software development

Apple Vs Android - A comparative study (2017) [online]. 1st March 2017. URL: https://android.jlelse.eu/apple-vs-android-a-comparative-study-2017 -c5799a0a1683. Accessed: 19th septiembre 2018

Lisandro Delía, Nicolás Galdamez, Pablo Thomas, Leonardo Corbalan Patricia Pesado (2016). Análisis Experimental de desarrollo de Aplicaciones Móviles Multiplataforma

Thomas p. y Lisandra D. (2018). Tendencias en el desarrollo de aplicaciones para dispositivos móviles.

Fuller, N. (2009,). Turismo y Cultura. Entre el entusiasmo y el recelo | norma fuller - Academia.edu.

Kurniawan y Pranoto (2018). Sistema de información de destino para la ciudad de Bandung utilizando los servicios basados en la ubicación (LBS) en Android.

Risco G. y Castañeda P. (2016). Aplicación Móvil para la promoción y publicidad del Turismo en la ciudad de Huancayo.

Broad J. (). Creación de una aplicación React Native sin conexión a la red

Canós, José H.; Letelier, Patricio; Penadés, Carmen. (2003). Metodologías ágiles en el desarrollo de software. Recuperado el 21 de octubre de 2018, de Grupo ISSI - Ingeniería de Software y Sistemas de Información - Universidad Politécnica de Valencia, España: http:issi.dsic.upv.es/archives/f-1069167248521/actas.pdf

Cunningham, W. (2001). Manifesto for Agile Software Development. Recuperado el 21 de Octubre de 2018, de Manifesto for Agile Software Development: http://agilemanifesto.org/iso/es/

Szalvay V. (2004) An introduction to agile software development, Danube Technologies.

Beck K y Colaboradores (2001). Manifiesto por el Desarrollo Ágil de Software. http://agilemanifesto.org/iso/es/. Consultado el 06 de octubre de 2011.

Schwaber K, Sutherland J. (2011). The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. 17 p.

COCEMFECYL http://www.cocemfecyl.es/index.php/discapacidad-y-tu/66-actividades-de-la-vida-diaria-avd

https://www.polarsteps.com/about Mark Twain 2018

Cómo funciona el sistema GPS, en cinco pasos lógicos. Trimble Navigation. 1996.

Chandra, Dr A. M. Higher Surveying. s.l.: New Age International, 2002.

Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.Tipos de dispositivos móviles. http://leo.ugr.es/J2ME/INTRO/intro_4.htm. Acceso 20 de julio del 2018

https://www.ijcsi.org/papers/IJCSI-9-1-2-237-242.pdf

https://www.researchgate.net/publication/323339293_Destination_Information_System_for_Bandung_City_Using_Location-Based_Services_LBS_on_Android

Dar formatos

Telerik. (2017). Welcome to NativeScript. Recuperado el 18 de 02 de 2018, de NativeScript: https://docs.nativescript.org/

ANEXOS

Anexo A: Análisis de requerimientos e Historias de usuario

Como ya se mencionó anteriormente, nuestro product backlog contiene la lista de requerimientos para nuestra aplicación.

Tabla A.1
Lista de requerimientos de la aplicación TourSteps

Nombre de la Aplicación: TourSteps					
Nro	**REQUERIMIENTOS FUNCIONALES**	**Complejidad**	**Prioridad**	**Actores**	**Tipo**
REQT-001	Registro y autentificación al aplicativo mediante google y facebook	Media	Media	Turista	Funcional
REQT-002	Gestionar el perfil del usuario registrado, obteniendo la mayor información posible de su cuenta de google y facebook	Alta	Media	App	Funcional
REQT-003	Registro de Viaje (Viajes que el turista está realizando, realizarà o realizó)	Media	Alta	Turista	Funcional
REQT-004	Seguimiento y Captura de experiencias durante el viaje (Toma de Fotos, Comentario de un lugar y otros), Puede hacerlo es turista o tambien la aplicacion dar sugerencias segun la ubicacion del turista	Alta	Alta	Turista	Funcional
REQT-005	Registro automático de la localizaciòn del usuario durante la experiencia turística, creando una ruta recorrida	Alta	Alta	App	Funcional
REQT-006	Vizualizaciòn de la ruta recorrida y los registros de paradas (Captura de momentos) en un Mapa (Esto tiene que ser en tiempo real)	Alta	Alta	App	Funcional
REQT-007	Reporte de estadisticas generales de las experiencias del turista autentificado turista	Baja	Media	Turista	Funcional
REQT-008	Vizualizaciòn de lugares registrados en google maps u otro, para que el turista pueda tener alguna referencia	Alta	Baja	App	Funcional
REQT-009	Vizualizar el listado de experiencias por pais (Turisto autentificado y de sus amigos)	Media	Baja	App	Funcional
REQT-010	Una experiencia puede tener comentarios, likes y ser compartido en facebook	Alta	Baja	Turista	Funcional
REQT-011	Crear un album de cada experiencia cuando esta finalice.	Media	Media	App	Funcional
REQT-012	La aplicacion movil tiene que funcionar de manera offline y online	Alta	Alta	App	No funcional
REQT-013	La aplicacion tiene que registrar como borrador un punto de interes en el que estuvo el usuario, esta puede ser aprobada o rechazada por el usuario	Alta	Alta	App	No funcional
REQT-014	La aplicacion tiene que registrar automaticamente el recorrido que esta realizando el usuario en segundo	Alta	Alta	App	No funcional

	plano				

Como se sabe en scrum esto representa el alcance que va tener nuestro producto. En este apartado se muestra los requerimientos en historias de usuario.

Identificador (ID) de la historia	Enunciado de la historia				Criterios de aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	# ES	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
REQT-001	Como turista	Necesito que los usuarios puedan autentificarse con su cuenta de facebook o google	Con la finalidad de obtener de manera rapida sus datos personales y algunas caracteristicas.	1	Autentificación con facebook	Cuando el usuario decida autentificarse con su cuenta de facebook	Cuando se inicia la aplicacion movil	A continuacion del inicio de seccion con su cuenta de facebook, se muestrará el perfil personal del usuario, con los siguientes datos: Photo, Nombre, Usuario, Pais, Edad y otros.
				2	Autentificación con google	Cuando el usuario decida autentificarse con su cuenta de google	Cuando se inicia la aplicacion movil	A continuacion del inicio de seccion con su cuenta de google, se muestrará el perfil personal del usuario, con los siguientes datos: Photo, Nombre, Usuario, Pais, Edad y otros.
REQT-003	Como turista	Necesito ver un listado de mis viajes turisticos	Con la finalidad de darles seguimiento	1	Viaje activo	Cuando el usuario esta en pleno viaje	Cuando se inicia seccion en la app	Seguidamente se observa como primero de la lista, el viaje activo
				2	Viajes finalizadas	Cuando el usuario ya finalizo el viaje	Cuando se inicia seccion en la app	Seguidamente se observa la lista de viajes despues de la que esta activa.
REQT-003	Como turista	Necesito crear un viaje	Con la finalidad de registrar todo el recorrido turistico y los lugares visitados	1	Viaje que va iniciar.	En caso de que vamos a iniciar nuestra viaje	Cuando damos click en el boton agregar viaje	A continuación procedemos a guardar; se mostrará el viaje creado con un punto de partida y las opciones para darle un seguimiento a nuestra experiencia.
				2	Viaje Futuro.	En caso que estamos planificando un viaje	Cuando damos click en el boton agregar viaje	A continuacion procedemos a guardar; se mostrará el viaje creado en el listado de experiencias con un identificador que indique que recien va inicar.
				3	Viaje pasada.	En caso que estamos registrando un viaje que ya paso	Cuando damos click en el boton agregar viaje	A continuacion procedemos a guardar; se mostrará el viaje creada en donde tendremos la opcion de registrar todas las experiencias de un viaje pasado.
REQT-004	Como un turista	Necesito registrar mis experiencias durante mi viaje	Con la finalidad de tener un recuerdo cronologico del viaje	1	Registro de lugar visitado sugerida por la aplicación	En caso que no estemos usando el aplicacion	Cuando haya un punto de interes y se encuentre informacion de ellos (FOTO)	Acontinuacion se tiene registrado una experiencia pero con un estado de borrador
				2	Registro de lugar visitado decidada por el turista	En caso que estemos usando el aplicativo	Cuando decidamos registrar el punto de interes	A continuacion se tiene registrado una experiencia con un estado publicado
				3	Registro de lugar visitado de un viaje pasado	En caso que el viaje sea pasada y se quiere registrar una experiencia	Cuando decidamos registrar el punto de interes	A continuacion se tiene registrado una experiencia con un estado publicado
				4	Registro de lugares visitados pasada en un viaje activo	En caso tengamos un viaje activa y la aplicación no a sugerido el punto de interes , ni el turista pudo registrarlo	Cuando decidamos registrar el punto de interes	A continuacion se tiene registrado una experiencia con un estado publicado

Figura A.1. Historias de usuario de la aplicación TourSteps

ANEXO A.1: HISTORIAS DE USUARIO DEL SPRINT #1

Identificador (ID) de la historia	Enunciado de la historia				Criterios de aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	# ES	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
REQT-001	Como turista	Necesito que los usuarios puedan autentificarse con su cuenta de facebook o google	Con la finalidad de obtener de manera rapida sus datos personales y algunas caracteristicas.	1	Autentificación con facebook	Cuando el usuario decida autentificarse con su cuenta de facebook	Cuando se inicia la aplicacion movil	A continuacion del inicio de seccion con su cuenta de facebook, se muestrará el perfil personal del usuario, con los siguientes datos: Photo, Nombre, Usuario, Pais, Edad y otros.
				2	Autentificación con google	Cuando el usuario decida autentificarse con su cuenta de google	Cuando se inicia la aplicacion movil	A continuacion del inicio de seccion con su cuenta de google, se muestrará el perfil personal del usuario, con los siguientes datos: Photo, Nombre, Usuario, Pais, Edad y otros.

ANEXO A.2: HISTORIAS DE USUARIO DEL SPRINT #2

Identificador (ID) de la historia	Enunciado de la historia				Criterios de aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	# ES	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
REQT-003	Como turista	Necesito ver un listado de mis viajes turisticos	Con la finalidad de darles seguimiento	1	Viaje activo	Cuando el usuario esta en pleno viaje	Cuando se inicia seccion en la app	Seguidamente se observa como primero de la lista, el viaje activo
				2	Viajes finalizadas	Cuando el usuario ya finalizo el viaje	Cuando se inicia seccion en la app	Seguidamente se observa la lista de viajes despues de la que esta activa.
REQT-003	Como turista	Necesito crear un viaje	Con la finalidad de registrar todo el recorrido turistico y los lugares visitados	1	Viaje que va iniciar.	En caso de que vamos a iniciar nuestra viaje	Cuando damos click en el boton agregar viaje	A continuación procedemos a guardar; se mostrará el viaje creado con un punto de partida y las opciones para darle un seguimiento a nuestra experiencia.
				2	Viaje Futuro.	En caso que estamos planificando un viaje	Cuando damos click en el boton agregar viaje	A continuacion procedemos a guardar; se mostrará el viaje creado en el listado de experiencias con un identificador que indique que recien va inicar.
				3	Viaje pasada.	En caso que estamos registrando un viaje que ya paso	Cuando damos click en el boton agregar viaje	A continaucion procedemos a guardar; se mostrará el viaje creada en donde tendremos la opcion de registrar todas las experiencias de un viaje pasado.

ANEXO A.3: HISTORIAS DE USUARIO DEL SPRINT #3

Identificador (ID) de la historia	Enunciado de la historia				Criterios de aceptación			
	Rol	Característica / Funcionalidad	Razón / Resultado	# ES	Criterio de aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
REQT-004	Como un turista	Necesito registrar mis experiencias durante mi viaje	Con la finalidad de tener un recuerdo cronologico del viaje	1	Registro de lugar visitado sugerida por la aplicación	En caso que no estemos usando el aplicacion	Cuando haya un punto de interes y se encuentre informacion de ellos (FOTO)	Acontinuacion se tiene registrado una experiencia pero con un estado de borrador
				2	Registro de lugar visitado decidada por el turista	En caso que estemos usando el aplicativo	Cuando decidamos registrar el punto de interes	A continuacion se tiene registrado una experiencia con un estado publicado
				3	Registro de lugar visitado de un viaje pasado	En caso que el viaje sea pasada y se quiere registrar una experiencia	Cuando decidamos registrar el punto de interes	A continuacion se tiene registrado una experiencia con un estado publicado
				4	Registro de lugares visitados pasada en un viaje activo	En caso tengamos un viaje activa y la aplicación no a sugerido el punto de interes , ni el turista pudo registrarlo	Cuando decidamos registrar el punto de interes	A continuacion se tiene registrado una experiencia con un estado publicado

Anexo B. **Intalación de Nativescript-vue**

La instalación de nativescript-vue es muy sencilla y casi toda la información se encuentra en su página oficial, recomiendo que vayan directamente ahí. En este apartado mostraremos más que una instalación las configuraciones que realizamos para trabajar nuestra App TourSteps.

Primeramente, instalamos nativescript-vue para trabajar localmente, ejecutamos el siguiente comando en nuestra terminal.

```
$ npm install -g nativescript
```

Una vez concluida la instalación, asegúrese de verificar que su instalación haya sido exitosa, ejecutando el comando tns doctor, recuerde que debe tener configurado su java y android en su computador.

```
$ tns doctor
```

Ahora iniciamos a crear la aplicación TourSteps con las siguientes líneas de código en nuestra terminal, para esto vamos usar la plantilla VUE-CLI.

```
$ npm install -g @vue/cli @vue/cli-init
$ vue init nativescript-vue/vue-cli-template TourSteps
```

Cuando se crea la aplicación vue-cli nos muestro opciones de configuración para nuestro proyecto de desarrolló como se puede observar en la figura B.1

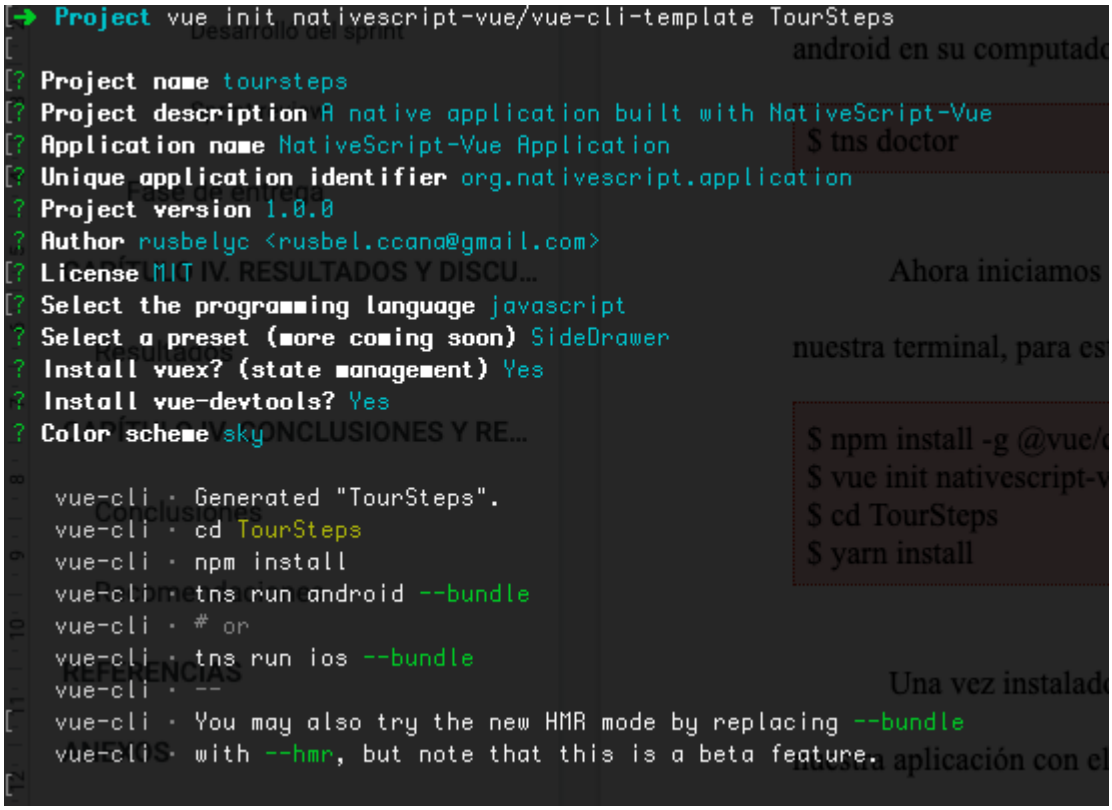


Figura B.1. Formulario de configuración de nativescript-vue/vue-cli-template
Fuente: Propia

Configurado la aplicación ahora ingresamos a nuestra carpeta base del proyecto.

```
$ cd TourSteps
```

Una vez ubicados dentro de la carpeta base, ejecutamos nuestra aplicación con el siguiente comando dependiendo si es para android o ios.

```
$ tns run android --bundle
$ # or
$ tns run ios --bundle
```

En esta parte se tiene que tener un dispositivo ya sea físico o en un emulador, donde se pueda visualizar la aplicación. La primera vez que ejecutemos nuestra app demora el proceso de compilación, así que calma ve a tomar un café y seguro cuando vuelva todo estará listo.

Una vez compilada nuestra aplicación veamos la estructura de nuestro proyecto.

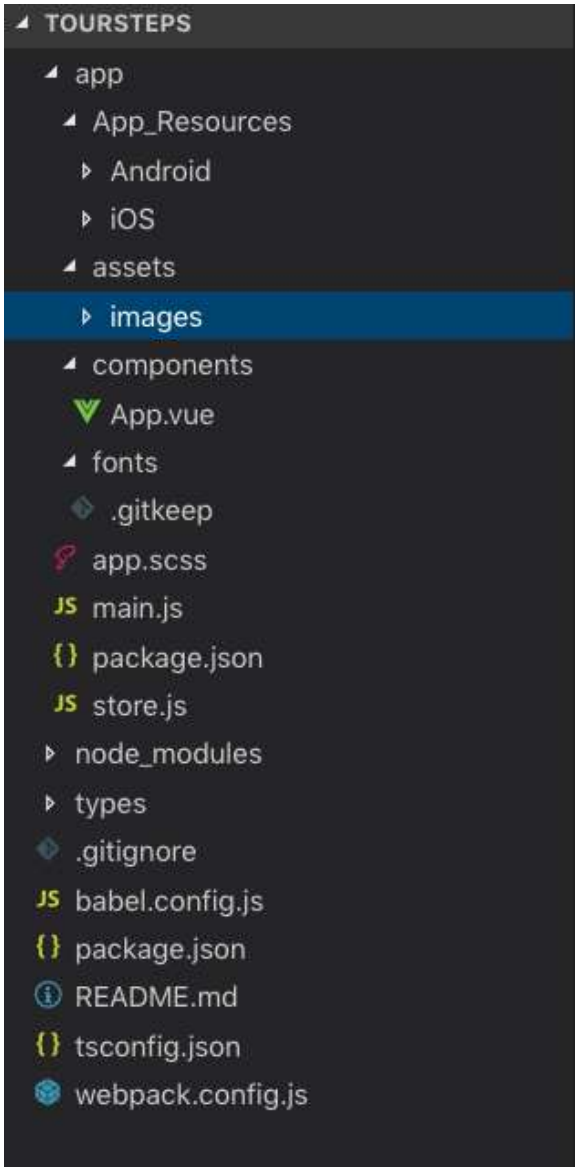


Figura B.2. Estructura de carpetas inicial de la aplicación TourSteps
Fuente: Propia

Como se puede observar en la figura B.2 el proyecto TourSteps tiene la subcarpeta app, node_modules y los archivos de configuración. Lo que nos interesa en esta parte es la carpeta app, porque dentro de ella estructuramos el funcionamiento de vuex para nuestra app. Lo primero es crear una carpeta ‘store’ que contiene la carpeta ‘modules’ y la carpeta ‘page’.

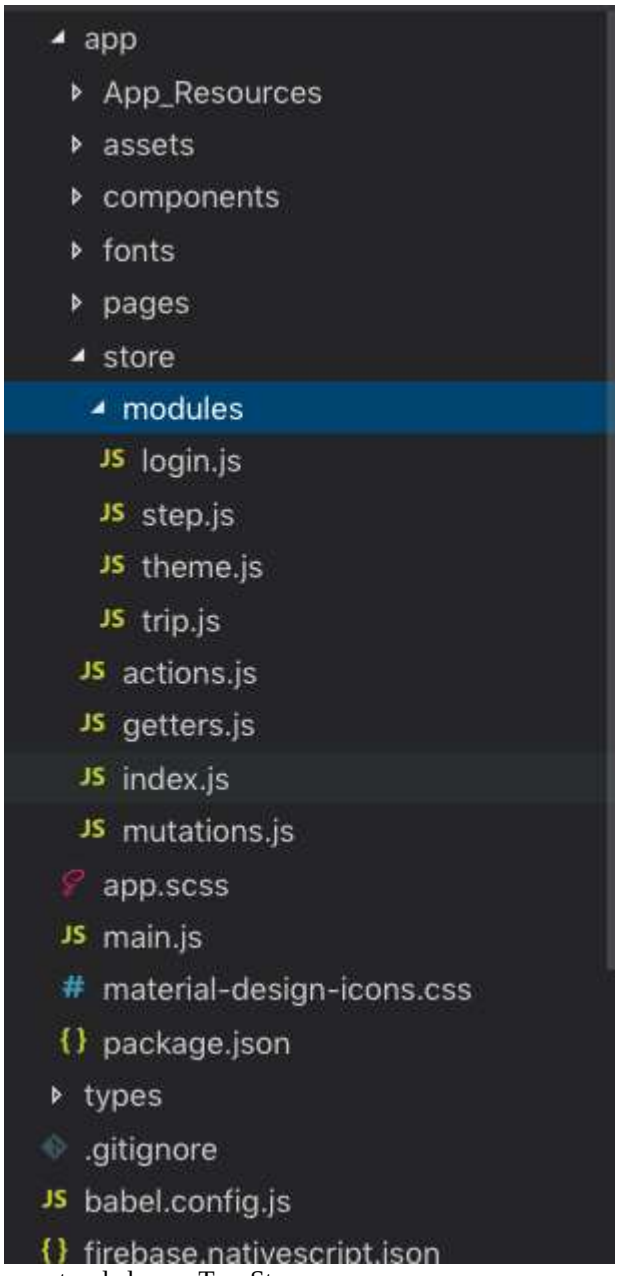


Figura B.3. Estructura de carpetas de la app TourSteps

Fuente: Propia

Visite nuestro repositorio de gitlab https://gitlab.com/tesis-ny/steps para ver el código completo de la aplicación.

Anexo C: Diagramas UML

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Recordemos que un modelo es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

TourSteps.
Type: Package
Status: Proposed. Version . Phase 1.0.
Package: Model
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {0EFEE25A-B15E-4409-8E69-DCF35028DFC0}

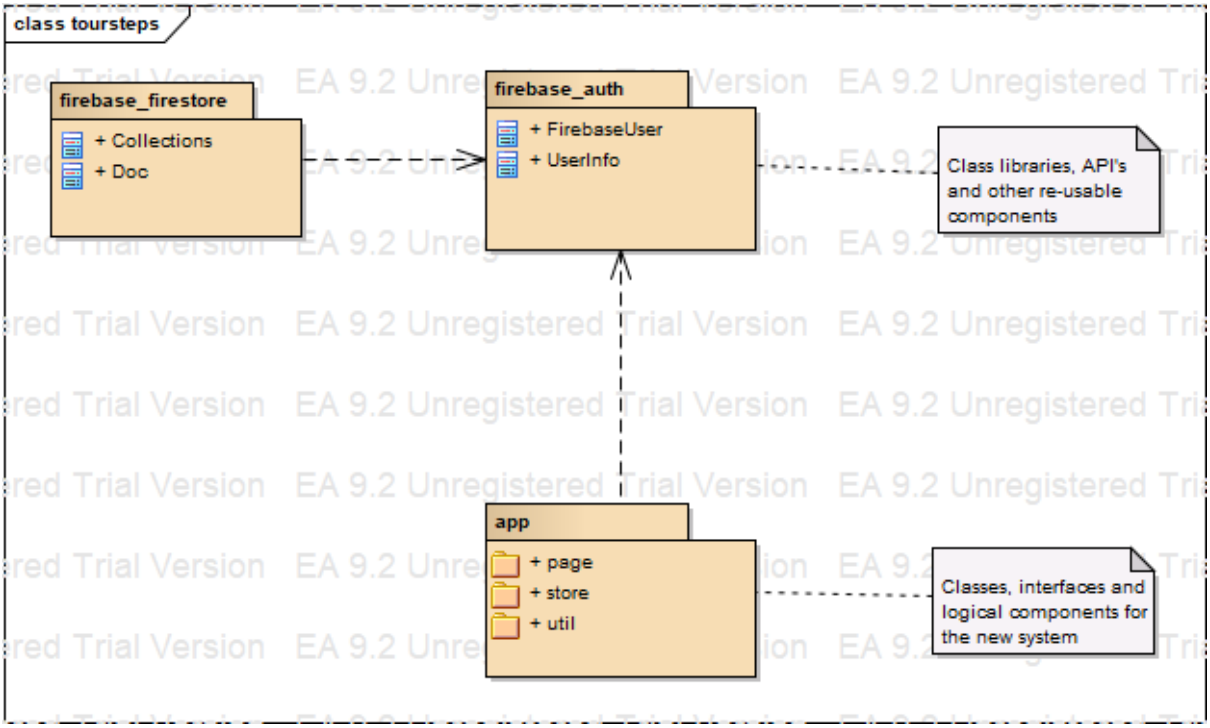


Figura C.1. Diagrama de paquetes

firebase_firestore
Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: toursteps
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {B3DBF9EC-8F60-4fcf-81B6-D53E7212E53F}

firebase_firestore - *(Class diagram)*

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {64CF1933-714E-41c5-B7D0-AF1D816A73AA}

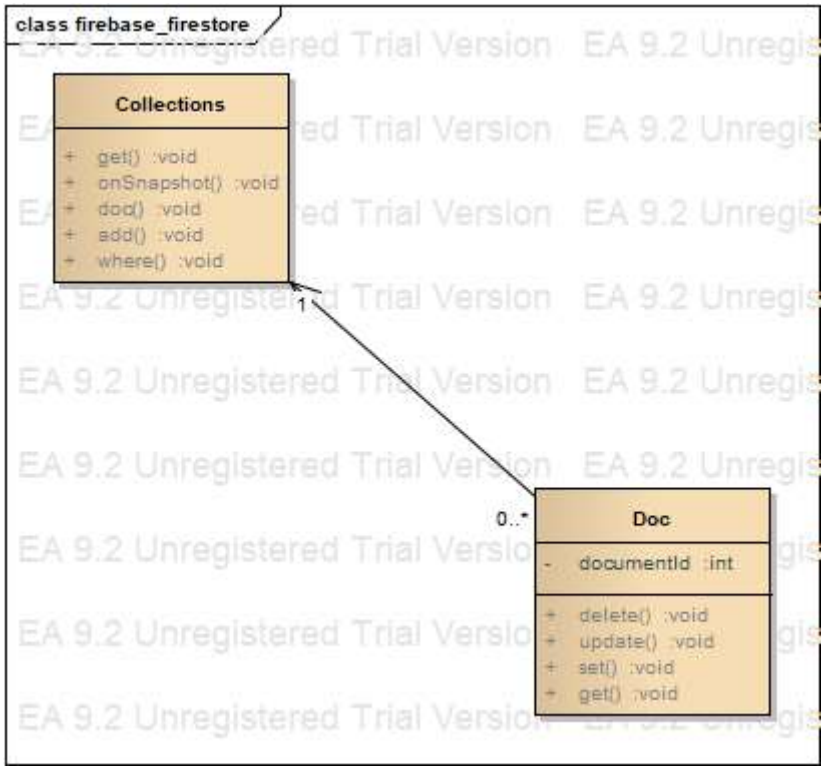


Figura C.2. Diagrama de clases del paquete firebase_firestore

firebase_auth

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: toursteps
Detail: *Created on 19/11/2005. Last modified on 25/06/2019*
GUID: {A89060A6-0021-405a-9FCF-A7C953C44CDB}

firebase_auth - (Class diagram)

Created By: Rusbelyc *on* 20/11/2005
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {6E49A0EF-0DF4-466b-ADC9-74BE50674EF5}

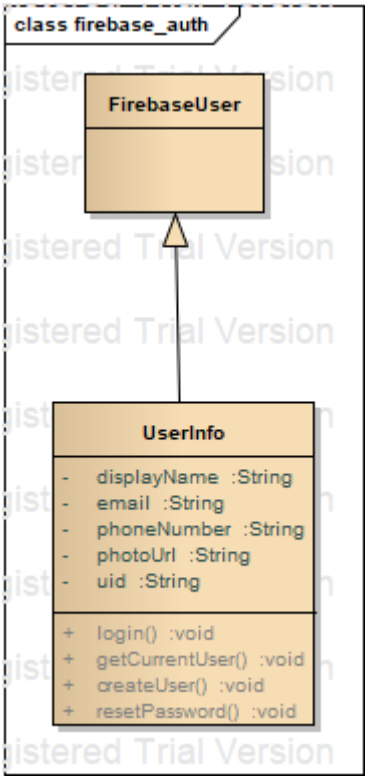


Figura C.3. Diagrama de clases del paquete firebase_auth

app

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: toursteps
Detail: *Created on 19/11/2005. Last modified on 25/06/2019*
GUID: {08C78E11-15F2-4b7c-8533-8E9309104267}

arquitectura

Created By: Rusbelyc *on* 20/11/2005
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {797CCF7F-5DFE-4a19-9140-921BFE741069}

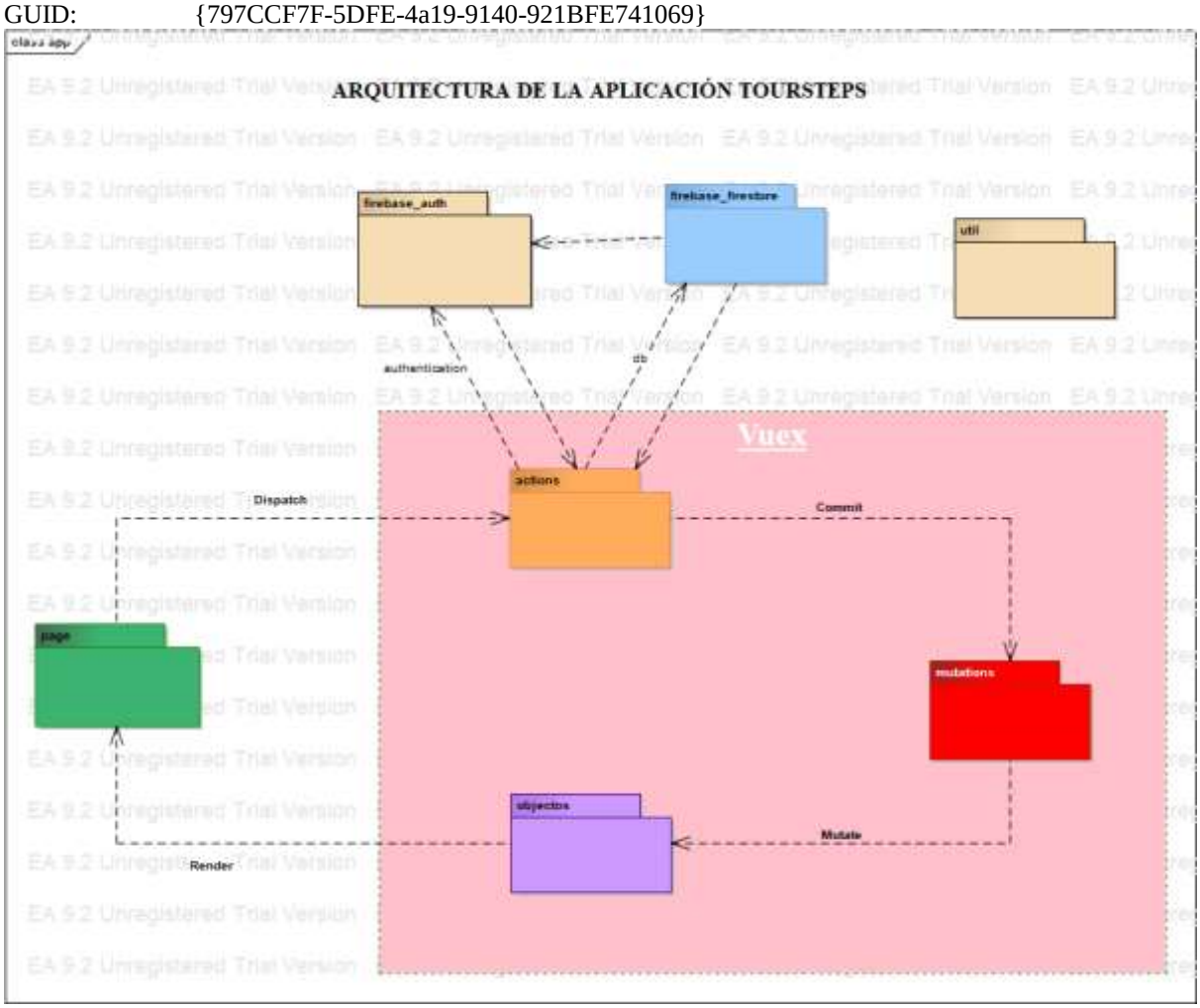


Figura C.4. Arquitectura de la aplicación

page

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: app
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {7A67E1D3-996D-40f1-8998-BDDC0D49F1EC}

page - (Package diagram)

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {2887BE85-828D-40ac-B6A8-B28BFA08B5E5}

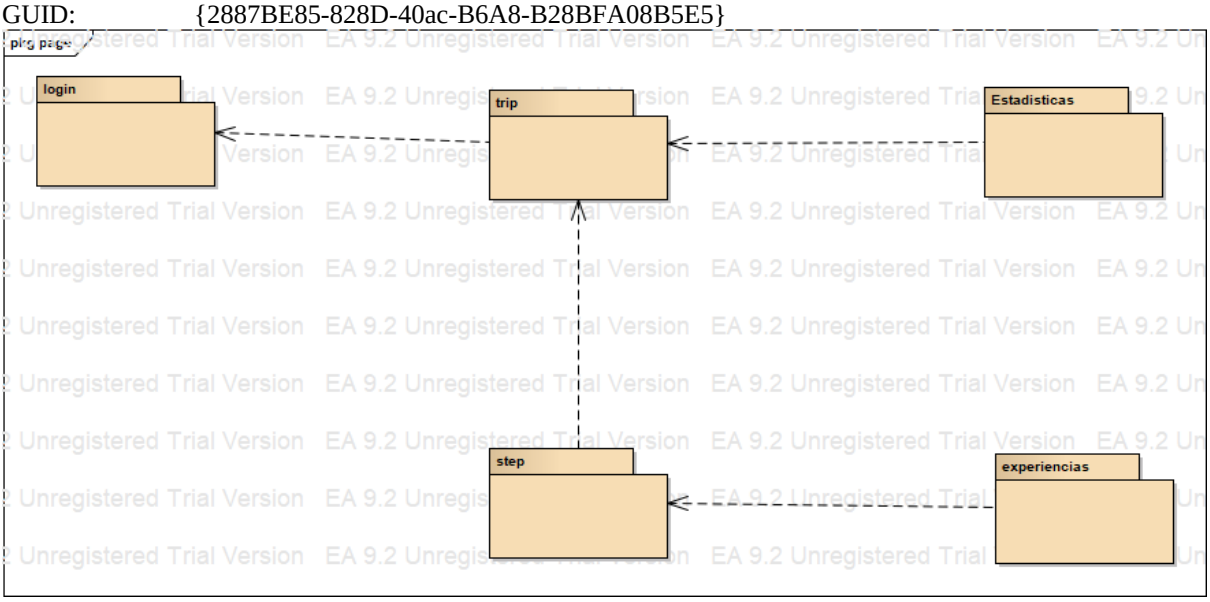


Figura C.5. Diagrama de paquetes

Estadísticas

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: page
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {ACE8F962-1408-4bca-A906-6BD62F66821B}

Estadísticas - (User Interface diagram)

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {EF5A2537-417B-4273-9DD3-339745BABF52}

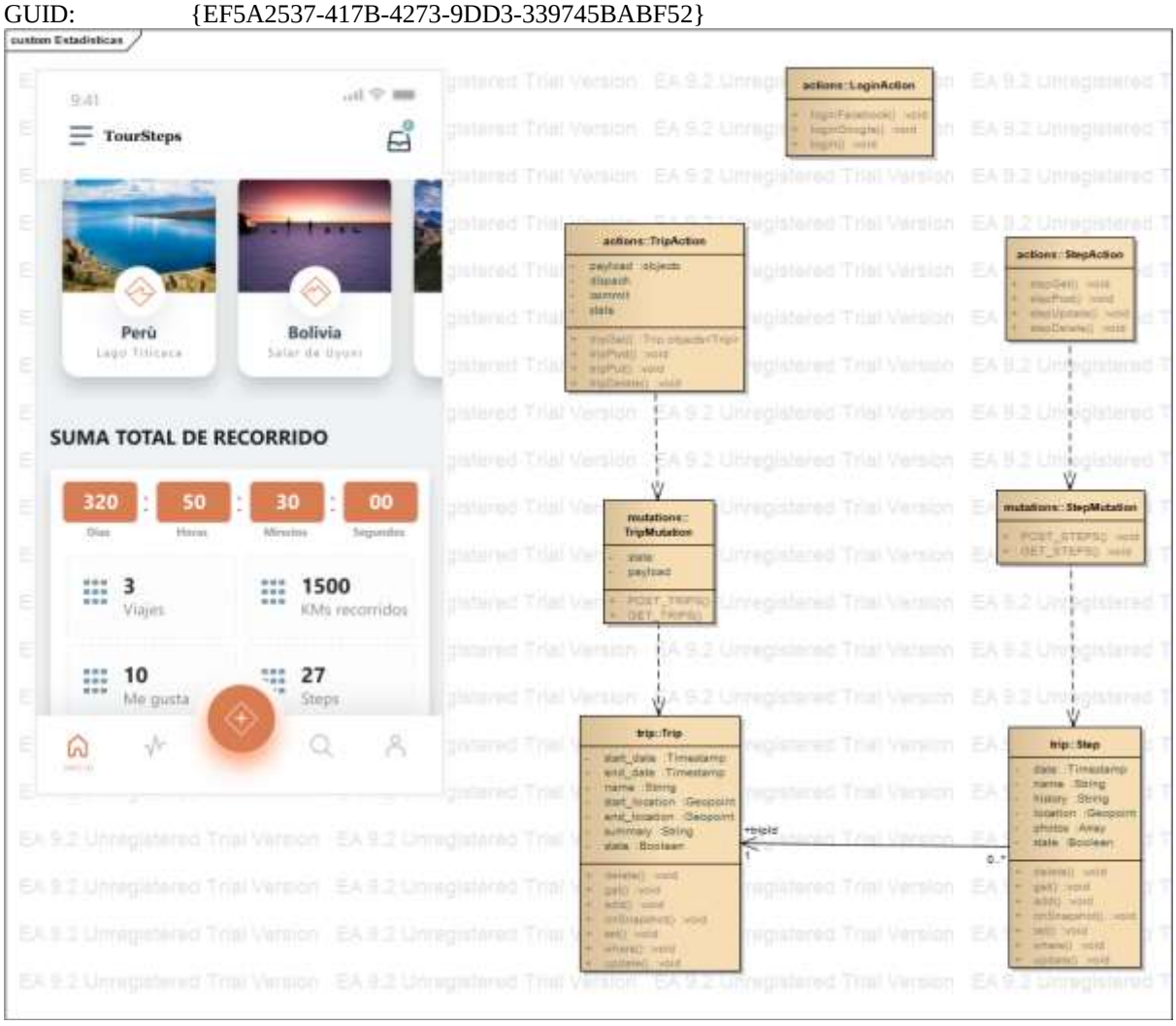


Figura C.6. Diagrama personalizado de la página estadística

Experiencias

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: page
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {04B0A080-369A-4d47-9E72-672D0B62A8C5}

experiencias - *(User Interface diagram)*

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {B07190F0-B397-4cd0-84FE-D3EA0A7B3AD3}

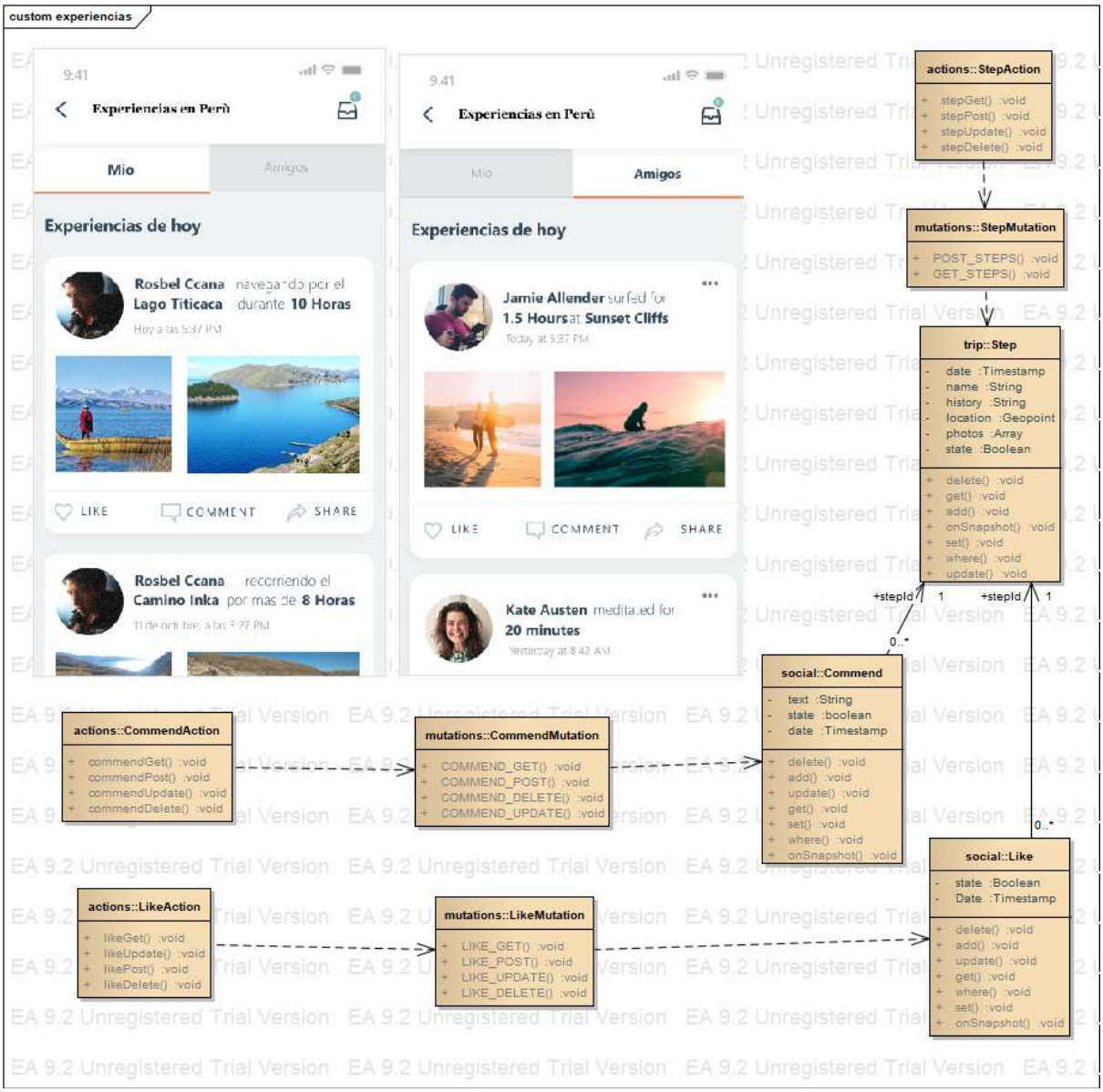


Figura C.7. Diagrama personalizado de la página experiencia

Login

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: page
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {8DE07276-3121-440c-9600-3194E594AF49}

login - *(User Interface diagram)*

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {D029A66E-2738-490e-8782-1F7991AB29AE}

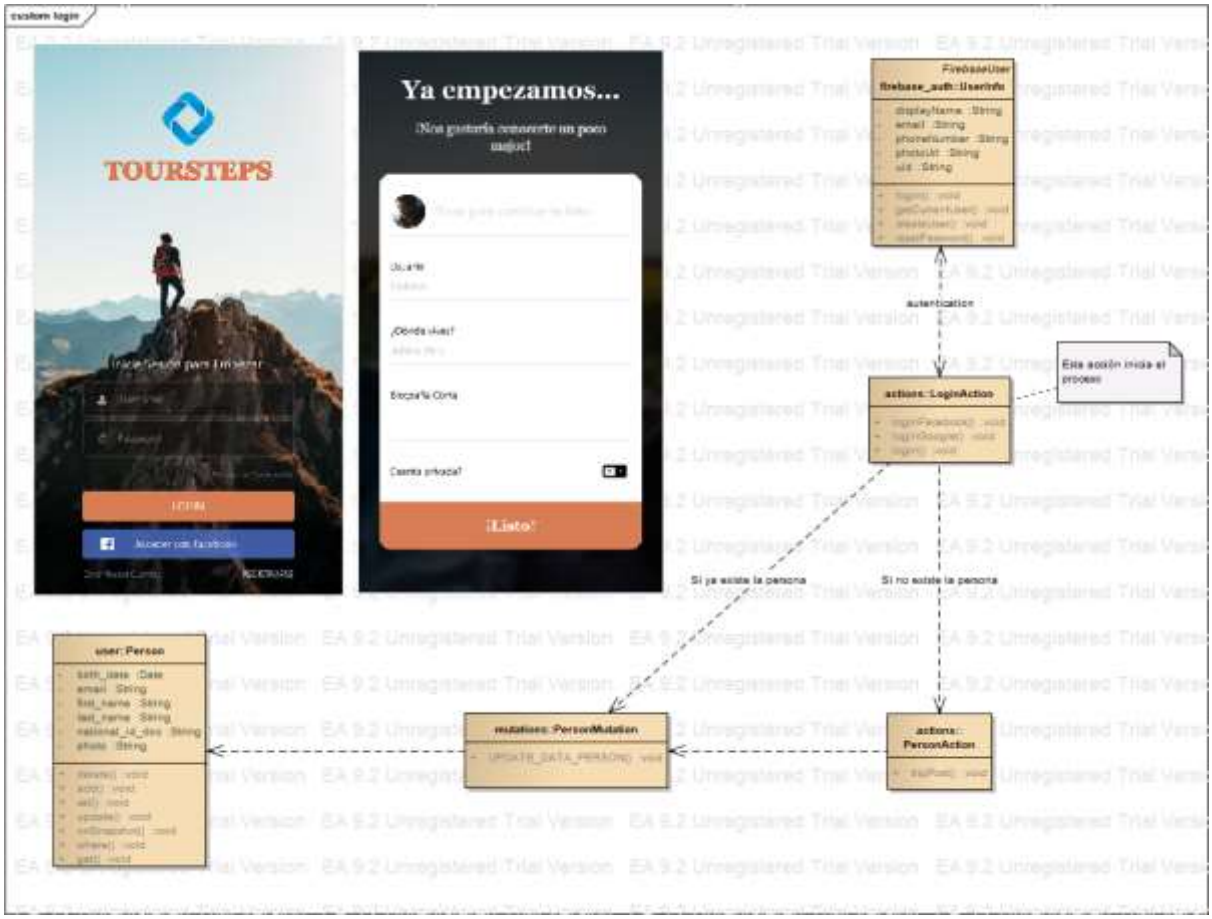


Figura C.8. Diagrama personalizado de la página login

Register

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: page
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {5BCF2696-1485-4990-8422-C17682E6229A}

register - *(User Interface diagram)*

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {E7BDFFD2-6BA8-4932-865B-93A78C75B7DD}

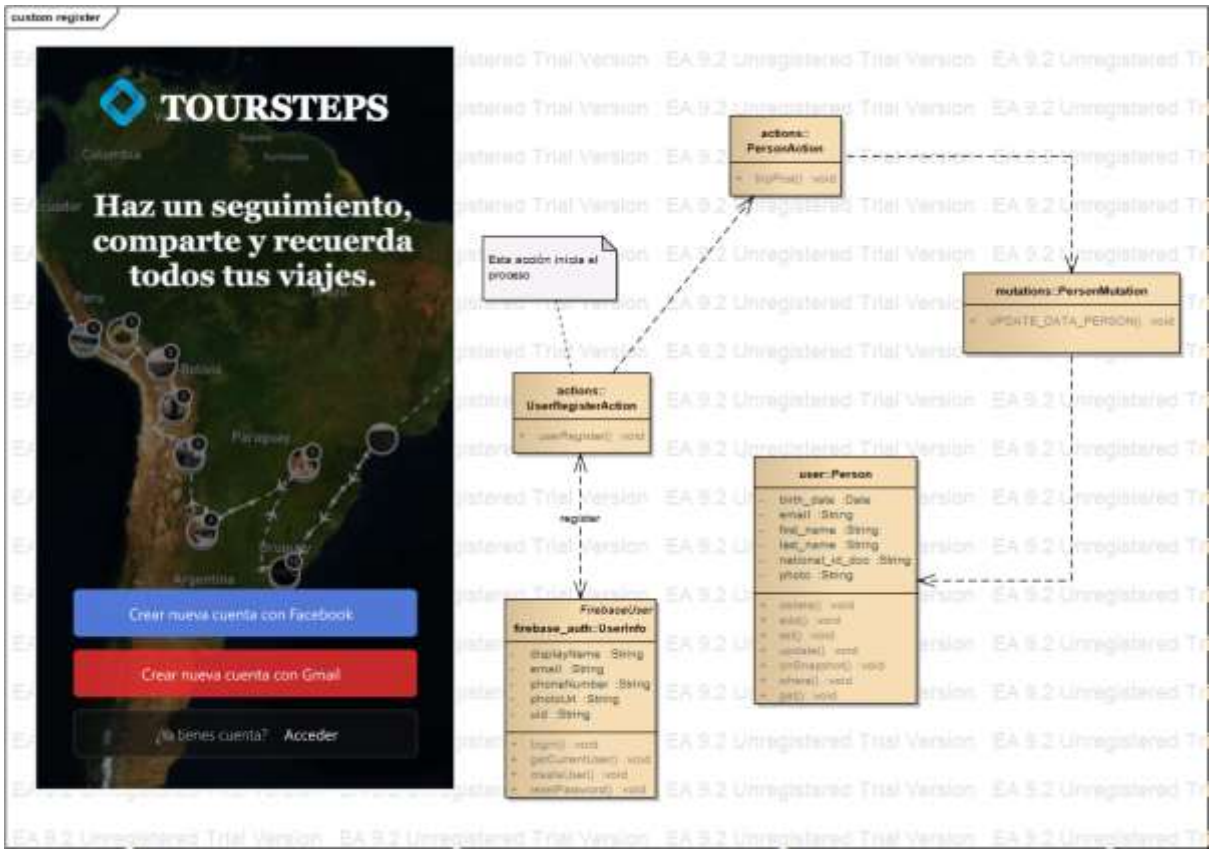


Figura C.9. Diagrama personalizado de la página experiencias

step

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: page
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {367B84BB-0099-42e1-8856-B0CD037B5E5B}

step_page - *(User Interface diagram)*

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {CF69E390-8FF6-4bd1-A229-CD2BF29B0274}

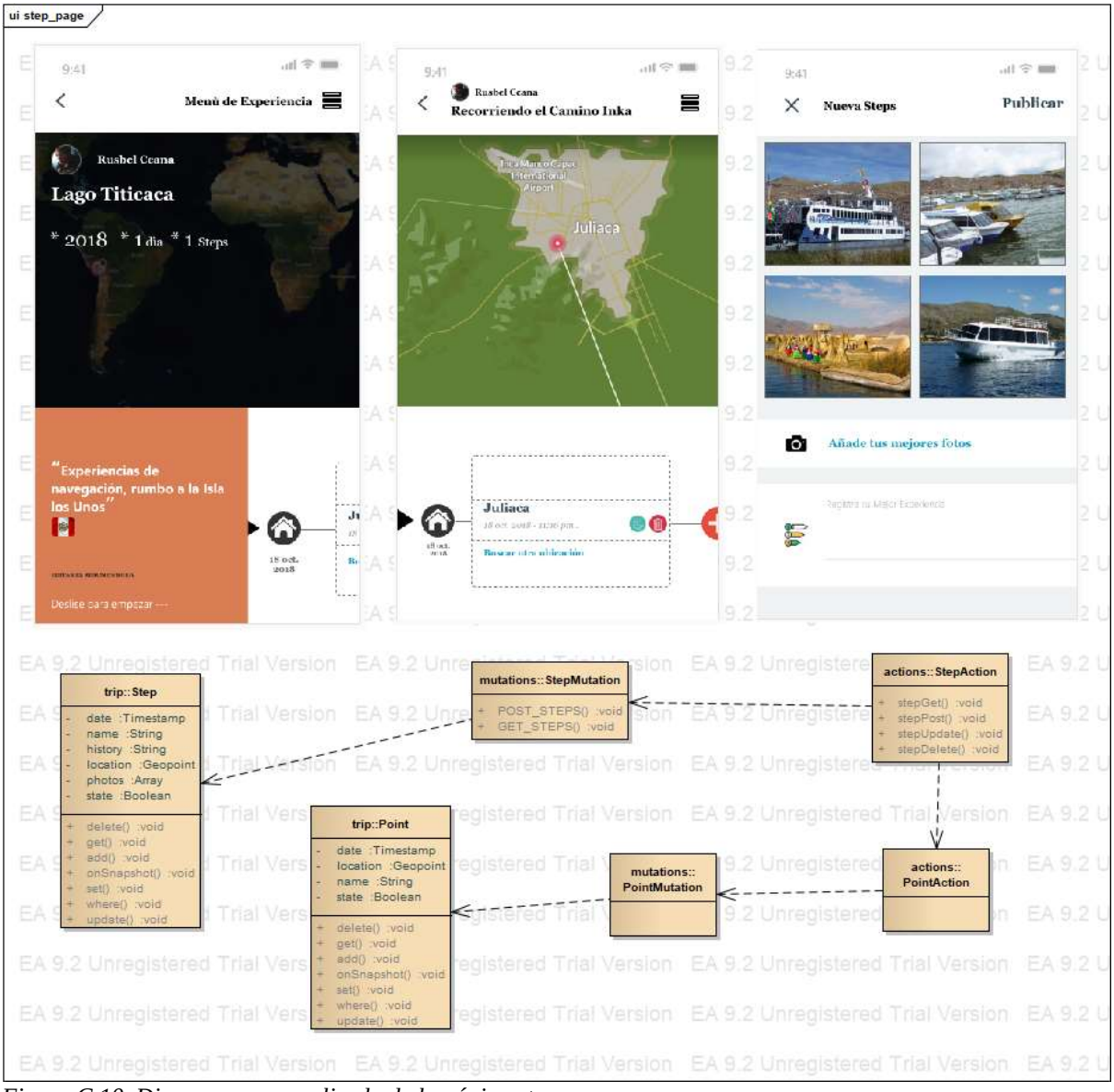


Figura C.10. Diagrama personalizado de la página step

trip
Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: page
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {937CA210-AC77-4577-96E8-AE95FBD03A96}

trip_page - *(Custom diagram)*
Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {E4975B68-88F6-4338-8B1E-EC2177F7D817}

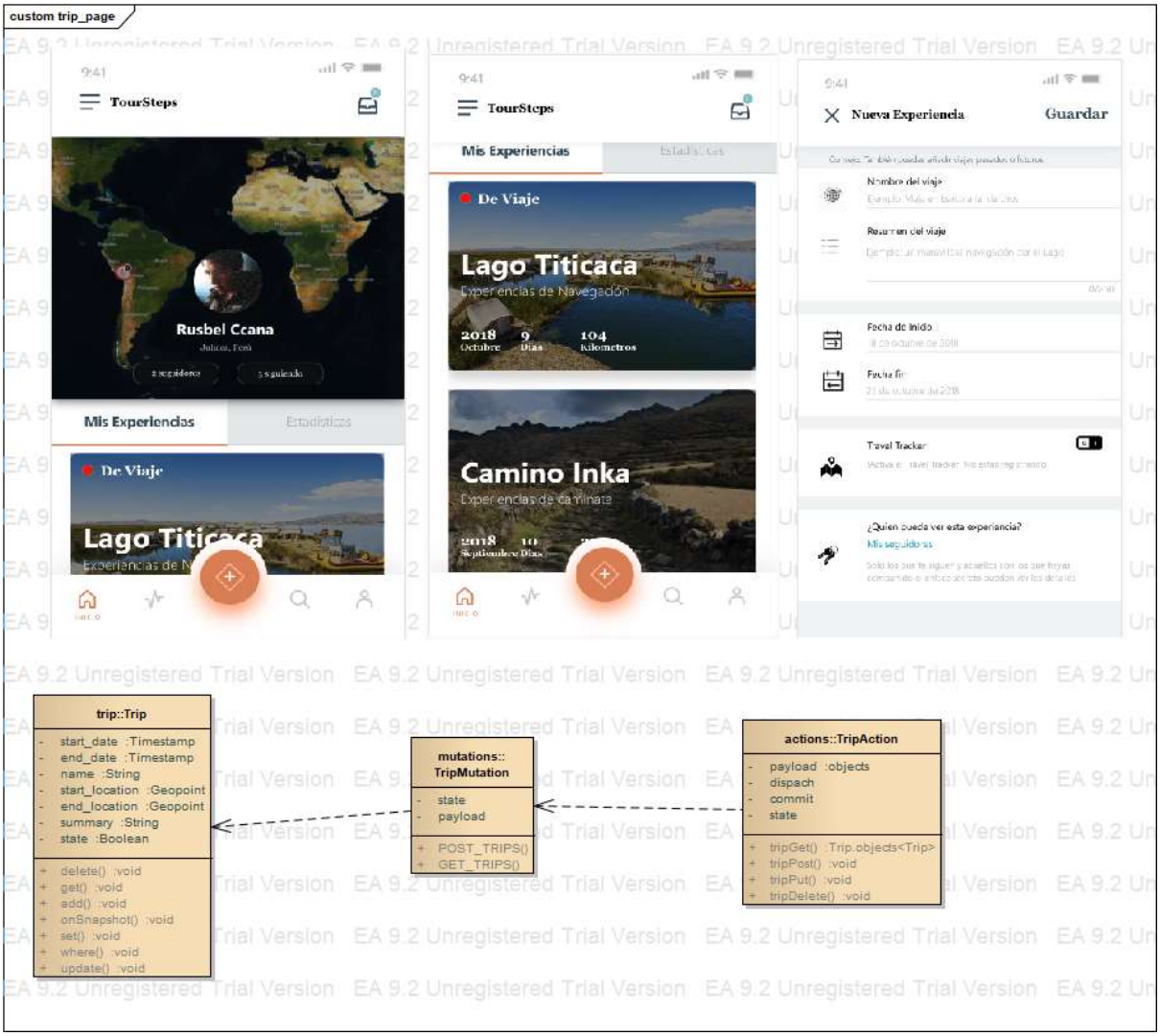


Figura C.11. Diagrama personalizado de la página trip

Store

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: app
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {B4F65556-E674-474e-A827-618D77A84627}

store - (Package diagram)

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {3471AE48-8E7E-4d68-BE51-4BEDE7FF5054}

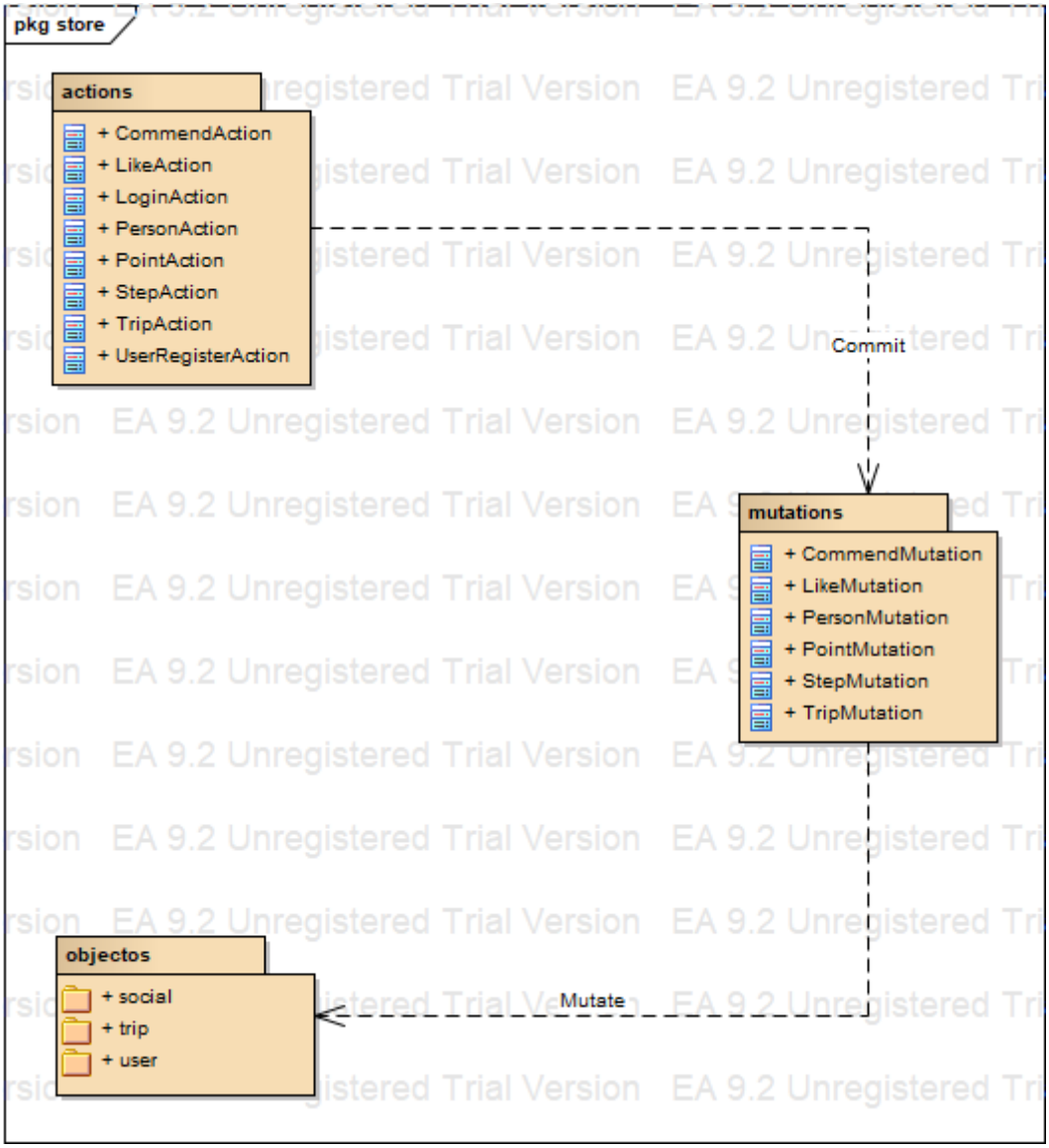


Figura C.12. Diagrama de paquetes del store

Actions

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: store
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {C7309EEB-36BE-4248-B78E-8736B467F3BB}

autentication_actions - *(Class diagram)*

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {CA470F9D-736C-4e67-8500-D1CE8F724DCE}

Figure: 13

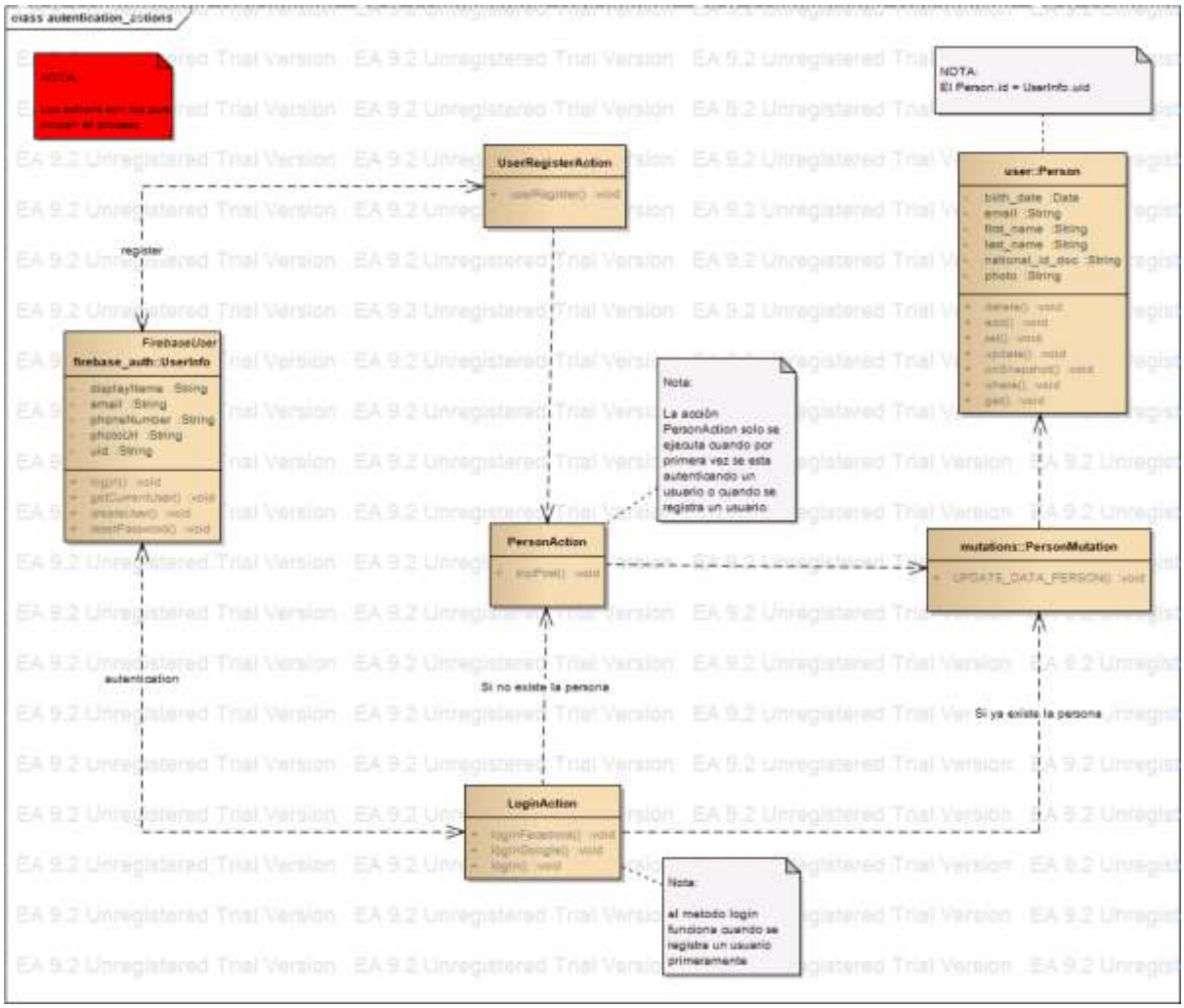


Figura C.13. Diagrama de clases de autenticación actions

social_actions - *(Class diagram)*
Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {4CBF5360-B504-4cee-8994-A1619F43F294}

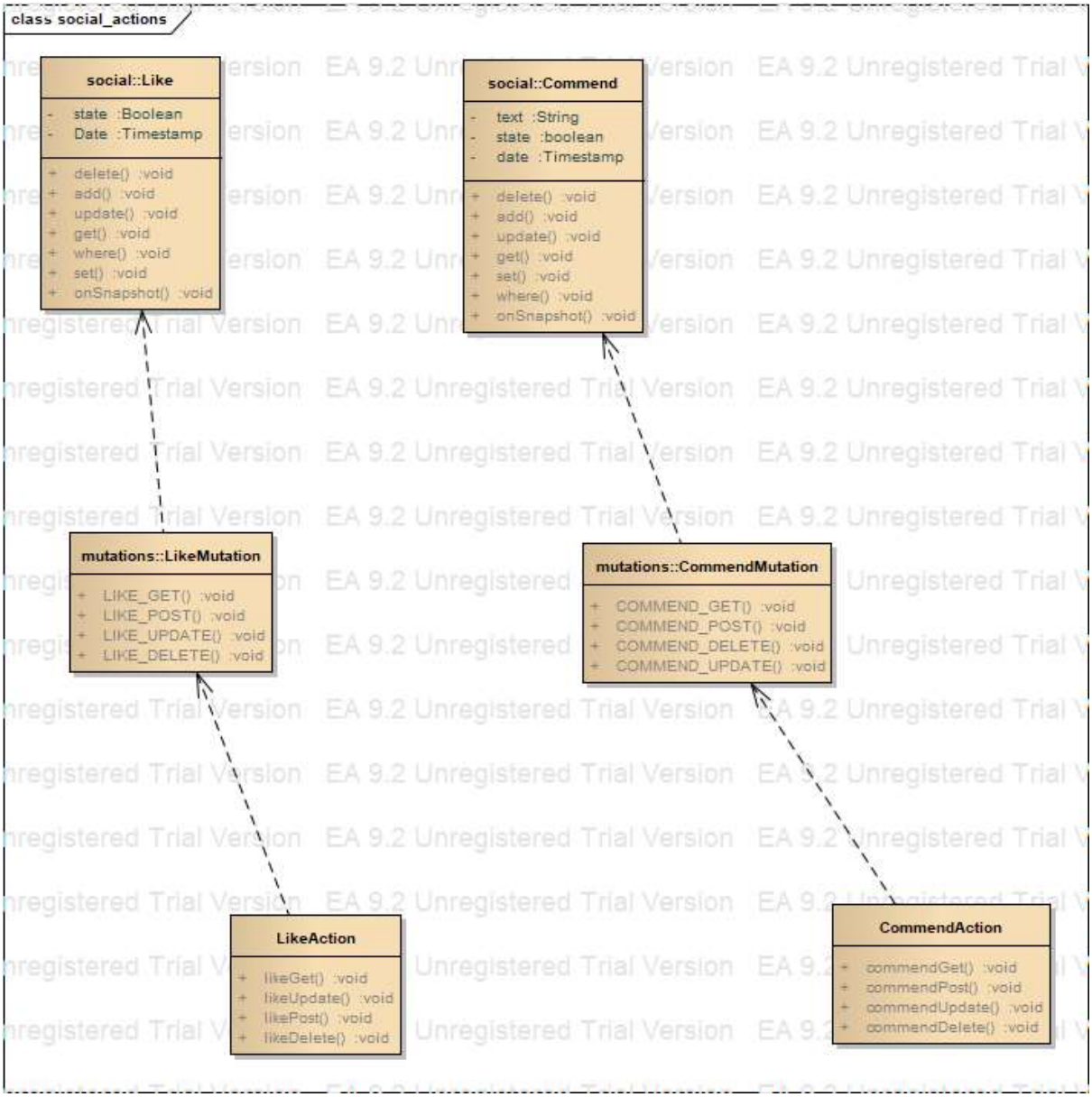


Figura C.14. Diagrama de clases de social actions

trip_actions - (Class diagram)
Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {378617F5-46C5-41b1-BEA7-D9351109A325}

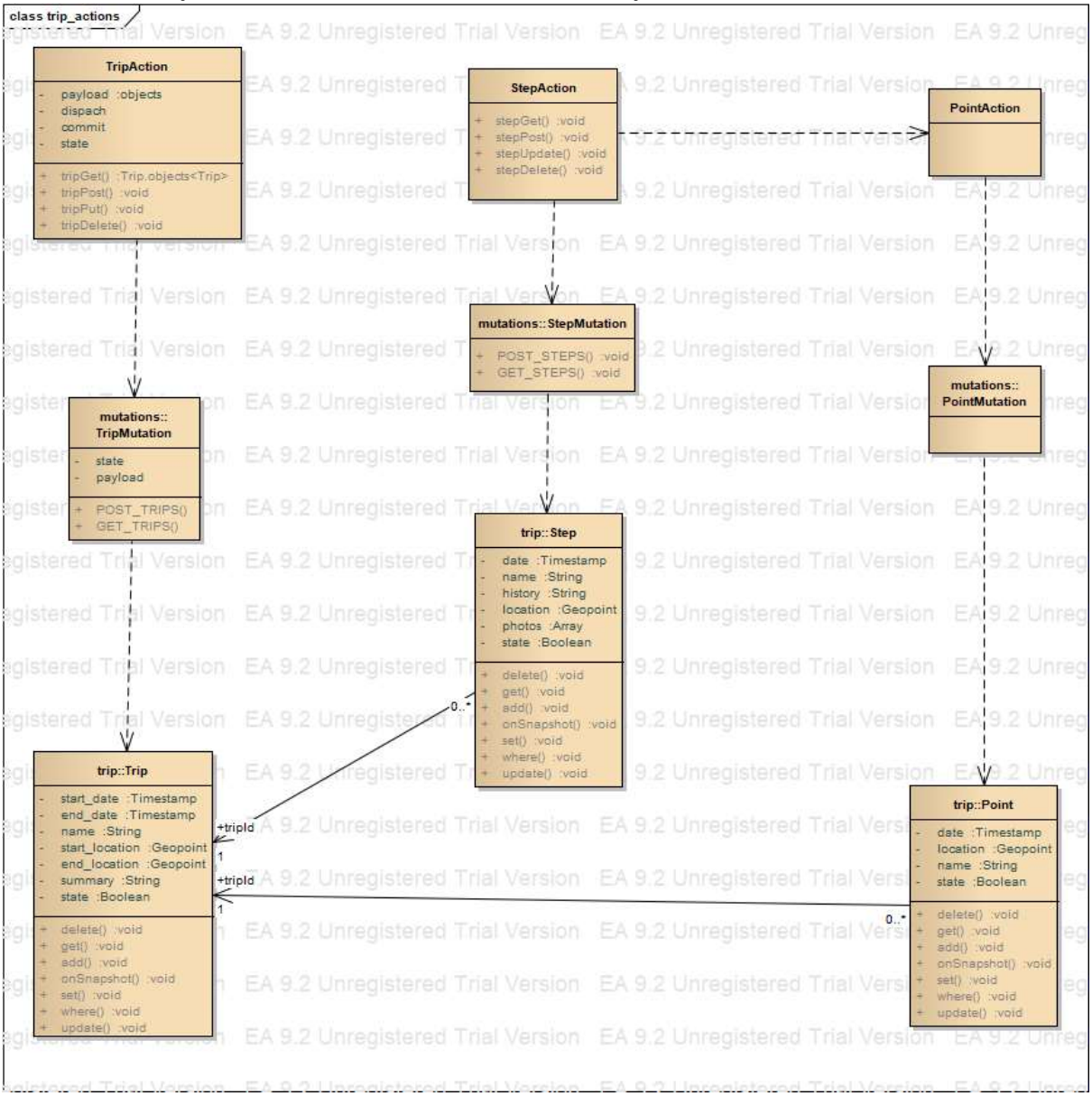


Figura C.15. Diagrama de clases de trip actions

Objetos

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: store
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {10202FA9-849E-4f78-B1F0-D9481196496D}

objetos - *(Package diagram)*
Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {B8473CCD-CFCE-4b7e-B509-34A755D8B163}

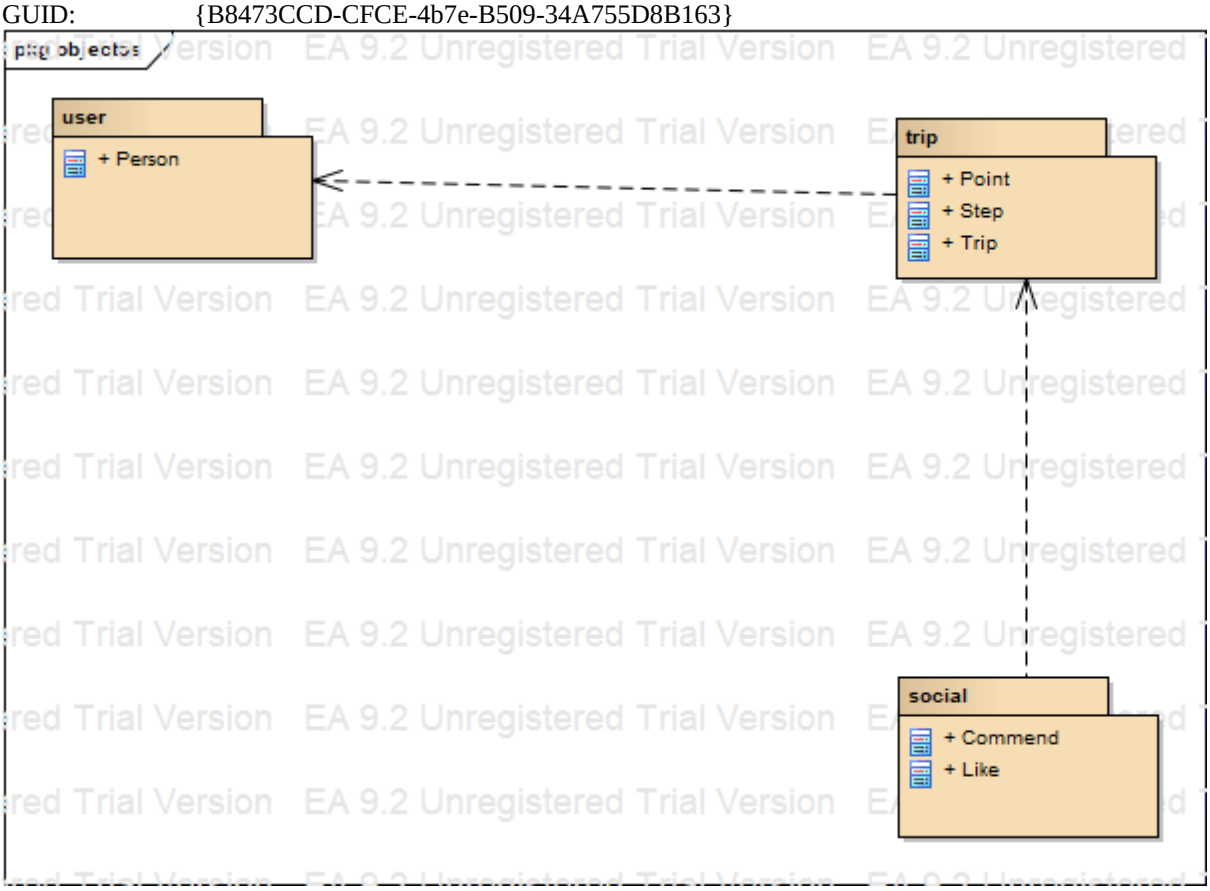


Figura C.16. Diagrama de paquetes de objects

Social

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: objectos
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {0D8604D3-E30E-4f00-8D9D-F76DED3478A9}

object_social - (Class diagram)

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {31B77E4D-F160-409c-AABB-A29425A39308}

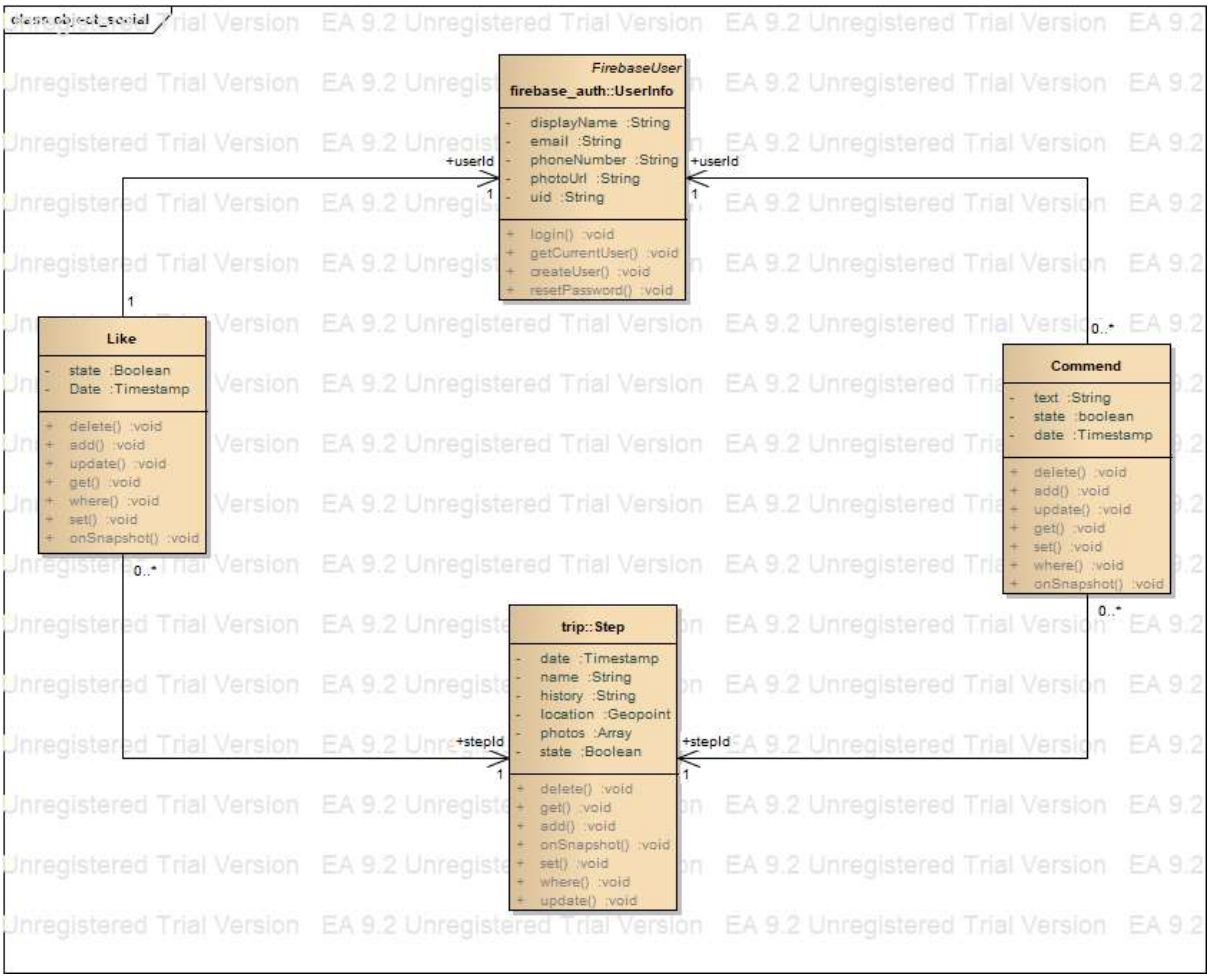


Figura C.17. Diagrama de clases de objects_social

Trip

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: objectos
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {2B621DC1-C92C-4655-9974-FE964E676333}

object_trip - *(Class diagram)*

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 26/06/2019
Version: 1.0. *Locked:* False
GUID: {4BB6F7DF-69E8-4541-BDFF-E911B297D6C4}

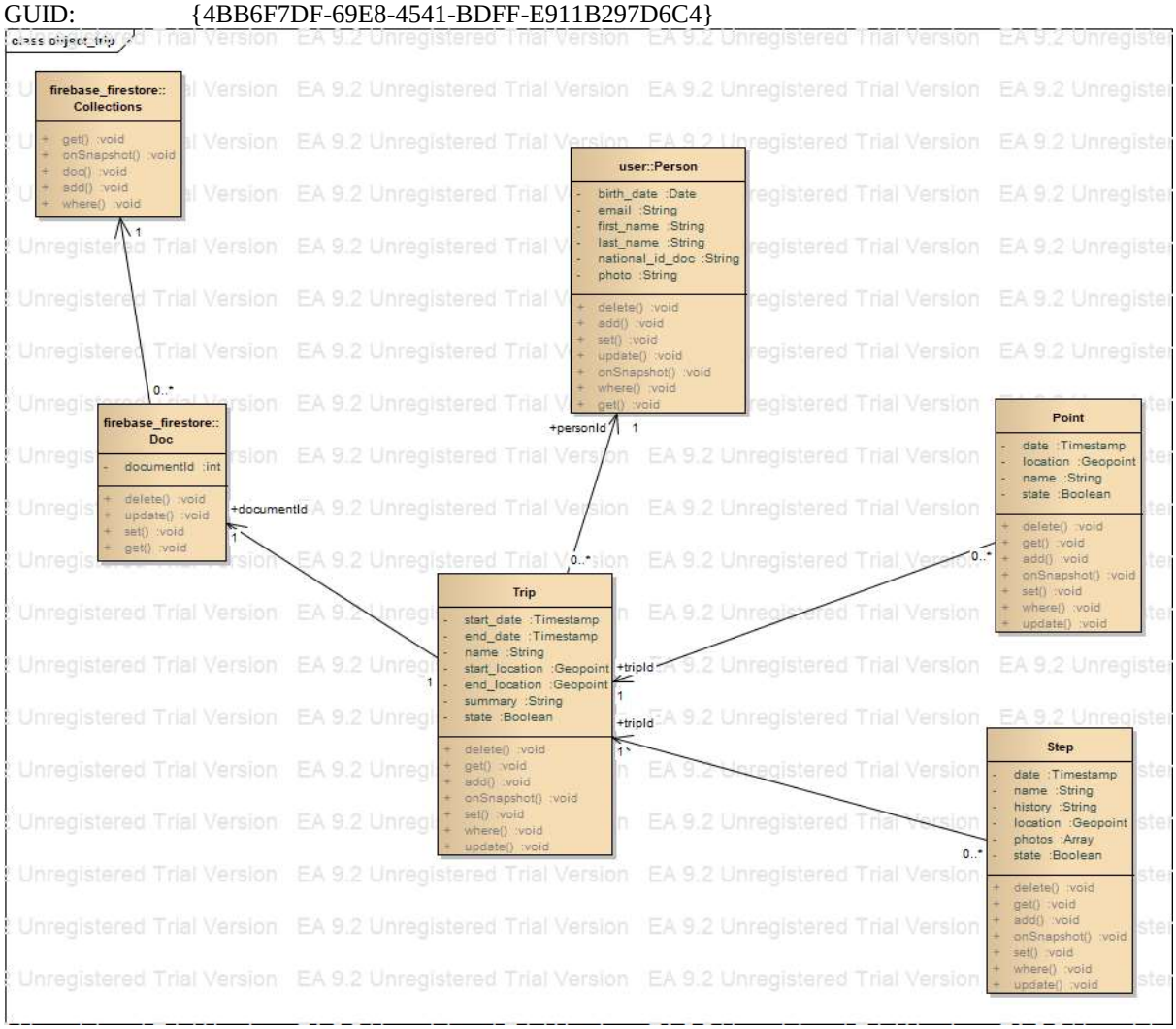


Figura C.18. Diagrama de clases de objects_trip

User

Type: Package
Status: Proposed. Version 1.0. Phase 1.0.
Package: objectos
Detail: *Created on 25/06/2019. Last modified on 25/06/2019*
GUID: {19CF2127-E75C-4bc5-8273-268E6EE5DECF}

user - (Class diagram)

Created By: Rusbelyc *on* 25/06/2019
Last Modified: 25/06/2019
Version: 1.0. *Locked:* False
GUID: {C5114747-492C-411a-95E7-EC0514744516}

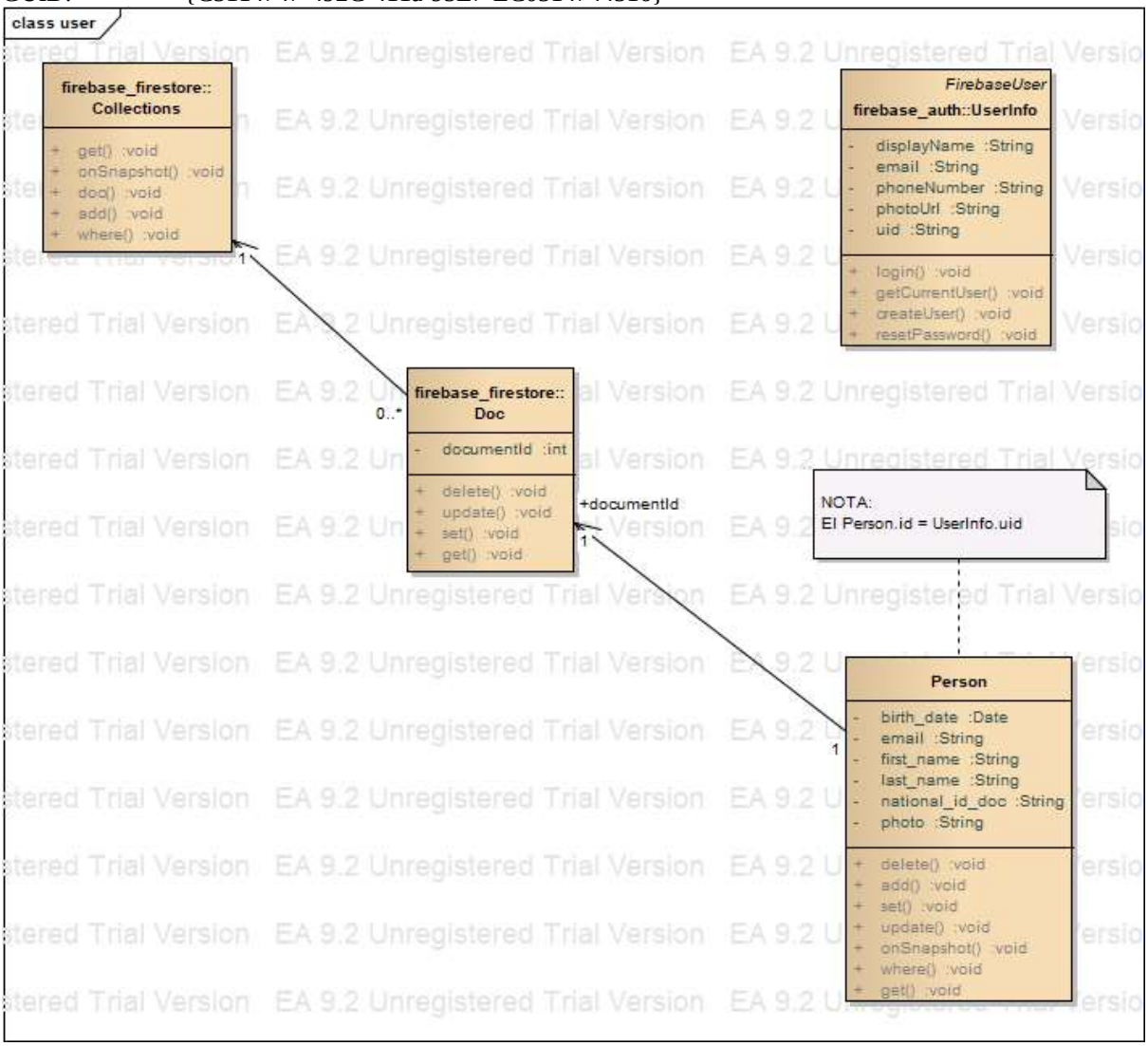


Figura C.19. Diagrama de clases de objects_user

Si usted desea visualizar a más detalle nuestro diagrama UML, revise en el siguiente link

Anexo D. Instalación y Configuración de Firebase

Prerrequisitos

Primeramente dirijase a https://console.firebase.google.com/ y registrese para obtener una cuenta gratuita. Seguidamente le pedirá crear un proyecto, siga todo los pasos que le indica.

Abra su proyecto Firebase en la consola de Google y haga clic en 'añadir aplicación' para agregar una aplicación iOS y/o Android. Siga los pasos que le indica (asegúrese de que el id de nativescript.id sea el mismo que el de package.json) y al final usted descargara uno de los siguientes archivos dependiendo el caso en el que se encuentre:

- iOS: GoogleService-Info.plist, esto tienes que añadirlo a tu proyecto NativeScript en app/App_Resources/iOS/GoogleService-Info.plist
- Android: google-services.json, esto tienes que agregar a tu proyecto NativeScript en app/App_Resources/Android/google-services.json

Instalación

Para la instalación ejecute el siguiente comando en su terminal.

```
$ tns plugin add nativescript-plugin-firebase
```

Esto iniciará un script de instalación que lo guiará en la instalación de componentes adicionales. Revisa los enlaces de documentos arriba para ver qué es qué. Siempre puedes cambiar tus elecciones más tarde.

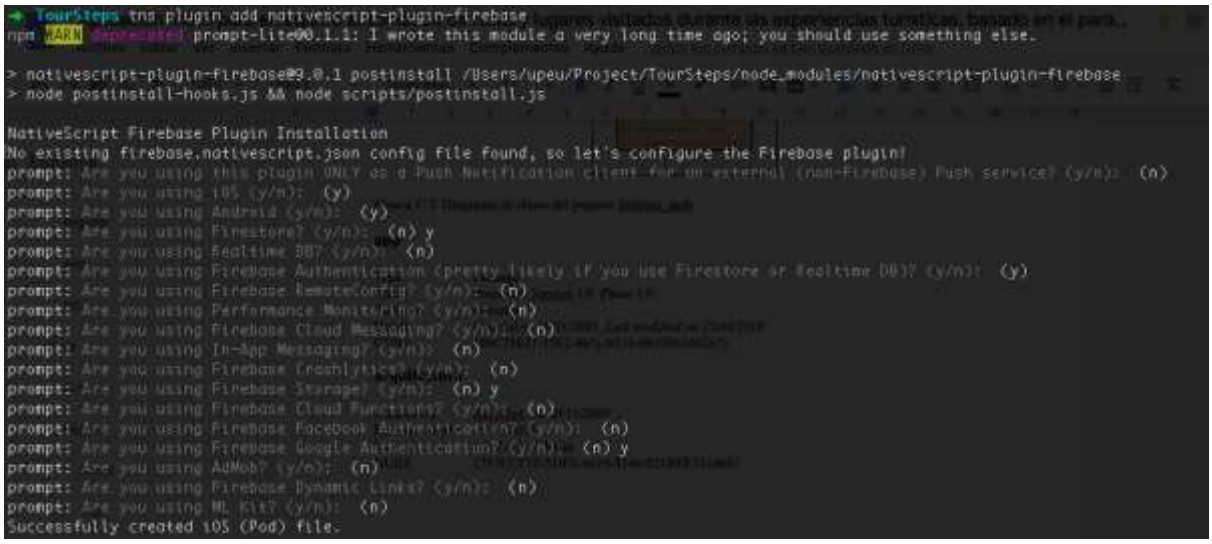


Figura D.1. Script de instalación de nativescript-plugin-firebase

Configuración

Las configuraciones realizadas anteriormente se guardan en el archivo firebase. nativescript.json, pero antes tienes que haber indicado que deseas guardar las configuraciones.

Primeramente, usted tiene que trabajar con NativeScript-Vue versión 2.0 porque se alinea perfectamente con este complemento, a parte de esto no hay más configuraciones necesarias.

Si está trabajando en un entorno iOS, nativescript-firebase se instala a través de Cocoapods y en algunos casos es necesario tenerlo actualizado, para esto ejecute “pod repo update” en la terminal.

Siempre recomiendo ir a la documentación oficial donde usted puede ver los últimos cambios, https://github.com/EddyVerbruggen/nativescript-plugin-firebase.

Habilitar la autenticación de firebase

Puede utilizar la API nativa o la API web. Es sólo una cuestión de antecedentes personales o preferencias.

Funciones

- Obtener el usuario actual

```
firebase.getCurrentUser()
    .then(user => console.log("User uid: " + user.uid))
```

```
    .catch(error => console.log("Trouble in paradise: " + error));
```

- Login por correo electrónico

```
const emailAddress = "someone@domain.com";
 firebase.fetchSignInMethodsForEmail(emailAddress).then((methods: Array<string>) => {
  console.log(`Sign-in methods for ${emailAddress}: ${JSON.stringify(methods)}`);
 });
```

- Login por correo y contraseña

```
firebase.login(
    {
     type: firebase.LoginType.PASSWORD,
     passwordOptions: {
       email: 'useraccount@provider.com',
       password: 'theirpassword'
     }
    })
    .then(result => JSON.stringify(result))
    .catch(error => console.log(error));
```

- Login por Facebook, ver la documentación oficial para más detalles

```
firebase.login({
  type: firebase.LoginType.FACEBOOK,
  // Optional
  facebookOptions: {
    // defaults to ['public_profile', 'email']
    scopes: ['public_profile', 'email'] // note: this property was renamed from "scope" in 8.4.0
  }
 }).then(
    function (result) {
      JSON.stringify(result);
    },
    function (errorMessage) {
      console.log(errorMessage);
    }
 );
```

- Login por Google

```
firebase.login({
  type: firebase.LoginType.GOOGLE,
  // Optional
  googleOptions: {
    hostedDomain: "mygsuitedomain.com",
```

```
    // NOTE: no need to add 'profile' nor 'email', because they are always provided
    // NOTE 2: requesting scopes means you may access those properties, but they are not automatically fetched by the plugin
    scopes: ['https://www.googleapis.com/auth/user.birthday.read']
  }
}).then(
    function (result) {
      JSON.stringify(result);
    },
    function (errorMessage) {
      console.log(errorMessage);
    }
);
```

Para más información revise la documentación oficial en https://github.com/EddyVerbruggen/nativescript-plugin-firebase/blob/master/docs/AUTHENTICATION.md

Habilitar Firestore

Durante la instalación del complemento, nos pregunta si deseamos usar firestore o no. En caso de que nuestra respuesta fue no, tenemos que ir al archivo firebase.nativescript.json en el directorio raíz del proyecto, para editar y añadir: "firestore": true. Por último para ver los cambios ejecute el siguiente comando en su terminal.

```
$ rm -rf platforms && rm -rf node_modules && npm i
```

APIs de firestore

Para poder usar firestore primeramente tenemos que inicializar firebase en el archivo index.js dentro del directorio store de nuestra aplicación.

```
import firebase from "nativescript-plugin-firebase"
 firebase.init({
 persist: true,
 storageBucket:'gs://steps-rusbel.appspot.com',
 onAuthStateChanged: function(data) {
     console.log("el valor del usuario ", data);
 }
})
```

Una vez inicializado firebase, almacenamos firestore en nuestro state de Vuex.Store, esto con la finalidad de tener acceso en todas las páginas del proyecto, en el siguiente fragmento de código estamos almacenando el valor de firestore en el objeto state.db.

```
export default new Vuex.Store({
   state: {
       db:firebase.firestore,
        auth2:'',
        user: '',
    },
});
```

- **collection**

Las colecciones son la raíz de cualquier interacción de firestore, estas almacenan datos en forma de documentos.

```
const collectionTrips= rootState.db.collection("trip");
```

- **collection.get(options?)**

```
const collectionTrips= rootState.db.collection("trip");

collectionTrips.get().then(querySnapshot => {
  querySnapshot.forEach(doc => {
     console.log(`${doc.id} => ${JSON.stringify(doc.data())}`);
  });
});
```

- **collection().onSnapshot()**

Este api, permite escuchar los cambios en una determinada colección, para esto puede registrase una función de devolución de llamada que se invoque cada vez que se cambian los datos.

```
const citiesCollection = firebase.firestore().collection("cities");

const citiesCollection = firebase.firestore().collection("cities");

const unsubscribe = citiesCollection.onSnapshot((snapshot: firestore.QuerySnapshot) => {
  snapshot.forEach(city => console.log(city.data()));
});

// then after a while, to detach the listener:
unsubscribe();

const docRef: firestore.DocumentReference =
firebase.firestore().collection("cities").doc("SF");
docRef.onSnapshot(
    {includeMetadataChanges: true},
     (doc: firestore.DocumentSnapshot) => {
        const source = doc.metadata.fromCache ? "local cache" : "server";
        console.log("Data came from " + source);
        console.log("Has pending writes? " + doc.metadata.hasPendingWrites);
    },
    (error: Error) => {
      console.error(error);
    }
  );
```

Para más información revise la documentación oficial en https://github.com/EddyVerbruggen/nativescript-plugin-firebase/blob/master/docs/FIRESTORE.md

Anexo E. Instalación y Configuración de mapbox

Prerrequisitos

Token de acceso a la API de Mapbox, Para esto tienes que registrarte en mapbox. Una vez que estemos registrados, tenemos que ir a Cuenta> Aplicaciones> Nuevo token. El 'Símbolo secreto predeterminado' es lo que necesitaremos.

Instalación

Primero tenemos que ubicarnos en el directorio raíz de nuestra aplicación y ejecutar el siguiente comando en la terminal.

```
$ tns plugin add nativescript-mapbox

//luego de esto se debe incluir el componente mapbox en la carpeta raiz

Vue.registerElement("Mapbox", () => require("nativescript-mapbox").MapboxView);
```

Una vez instalado el plugin no hay necesidad de realizar más configuraciones. Ahora para poder usar simplemente tenemos que importar nativescript-mapbox en la página que deseamos mostrar y luego escribir el componente. En nuestro caso lo estamos haciendo en el home; el siguiente fragmento de código muestra la implementación.

```
<template>
 <Page class="page" :actionBarHidden="true" @loaded="appLoaded">
  <StackLayout>
      <AbsoluteLayout>
       <FlexboxLayout flexDirection="column" width="100%" height="55%"
justifyContent="center" alignItems="center">
          <Mapbox
           :accessToken="accessToken"
           mapStyle="satellite"
           :latitude="location.lat"
           :longitude="location.lng"
           zoomLevel="n"
           showUserLocation="true"
           disableZoom="true"
           disableRotation="true"
           @mapReady="onMapReady($event)">
          </Mapbox>
        </FlexboxLayout>
      </AbsoluteLayout>
  </StackLayout>
```

```
 </Page>
</template>

<script>
 import { Mapbox } from 'nativescript-mapbox'
 export default {
   data: () => ({
     user_id:'',
     location:'',
     msg: 'Hello World!',
     accessToken:
'pk.eyJ1IjoicnVzYmVsIiwiYSI6ImNqdWJwdGM4OTBnMG80M28wOXdtYjZ2cG8ifQ.16wj2ZJOzFkDRMisf8WMKw',
   }),
 };
</script>
```

En el fragmento de código lo que hay que resaltar es el accessToken, esto se genera primeramente en los prerrequisitos.

Métodos en Mapbox

Solo mencionaremos los más importante y necesarios que fueron para nosotros, si desea usted puede ir a ver en su documentación oficial.

- **addMarkers**

```
import { MapboxMarker } from "nativescript-mapbox";

 const FirstMarker={
   id: 2, '
   lat: 52.3602160,
   lng: 4.8891680,
   title: 'One-line title here',
    subtitle: 'Infamous subtitle!',
```

```
  // icon: 'res://cool_marker',
  icon: 'http(s)://website/coolimage.png',
  iconPath: 'res/markers/home_marker.png',
  selected: true, /
  onTap: marker => console.log("Marker tapped with title: '" + marker.title + "'"),
  onCalloutTap: marker => alert("Marker callout tapped with title: '" + marker.title + "'")
};

mapbox.addMarkers([
  FirstMarker,
  {
    // aqui se puede registrar mas marcadores..
  }
])
```

- **Actualizar marcadores**

```
FirstMarker.update({
  lat: 52.3622160,
  lng: 4.8911680,
  title: 'One-line title here (UPDATE)',
  subtitle: 'Updated subtitle',
  selected: true, // this will trigger the callout upon update
  onTap: (marker: MapboxMarker) => console.log(`UPDATED Marker tapped with title: ${marker.title}`),
  onCalloutTap: (marker: MapboxMarker) => alert(`UPDATED Marker callout tapped with title: ${marker.title}`)
})
```

- **setCenter**

```
mapbox.setCenter(
    {
      lat: 52.3602160, // mandatory
      lng: 4.8891680, // mandatory
      animated: false // default true
    }
)
```

- **getUserLocation**

Si la ubicación del usuario se muestra en el mapa, puede obtener sus coordenadas y velocidad.

```
mapbox.getUserLocation().then(
    function(userLocation) {
      console.log("Current user location: " +  userLocation.location.lat + ", " +
userLocation.location.lng);
      console.log("Current user speed: " +  userLocation.speed);
    }
 )
```

- **addPolyline**

```
mapbox.addPolyline({
    id: 1,
    color: '#336699', // Set the color of the line (default black)
    width: 7, // Set the width of the line (default 5)
    opacity: 0.6, //Transparency / alpha, ranging 0-1. Default fully opaque (1).
    points: [
        {
           'lat': 52.3833160, // mandatory
           'lng': 4.8991780 // mandatory
        },
        {
           'lat': 52.3834160,
           'lng': 4.8991880
        },
        {
           'lat': 52.3835160,
           'lng': 4.8991980
        }
    ]
 });
```

Offline Maps

Para situaciones en las que desea que el usuario cargue previamente ciertas regiones, puede utilizar estos métodos para crear y eliminar regiones sin conexión.

Descargar una región para funcionamiento offline

Describir la region que estamos descargando

```
mapbox.downloadOfflineRegion(
```

```
  {
    accessToken: accessToken, // required for Android in case no map has been shown yet
    name: "Amsterdam", // this name can be used to delete the region later
    style: mapbox.MapStyle.OUTDOORS,
    minZoom: 9,
    maxZoom: 11,
    bounds: {
      north: 52.4820,
      east: 5.1087,
      south: 52.2581,
      west: 4.6816
    },
    // this function is called many times during a download, so
    // use it to show an awesome progress bar!
    onProgress: function (progress) {
      console.log("Download progress: " + JSON.stringify(progress));
    }
  }
).then(
  function() {
    console.log("Offline region downloaded");
  },
  function(error) {
    console.log("Download error: " + error);
  }
);
```

Para más información revise la documentación oficial en https://github.com/EddyVerbruggen/nativescript-mapbox

Anexo F. Mapic

VARIABLE FÁTICA - Hecho	DIMENSIONES - área y/o variable	INDICADORES
1. Registro automático de experiencias del recorrido turístico "Los recorridos que realizamos como turistas no se registran con facilidad, y al final de los viajes no tenemos recuerdos de lo que realizamos día a día"	1.1. Persona "Turista"	1.1.1. Procedencia
		1.1.2. Género
		1.1.3. Edad
	1.2. Paradas del Turista "Los destinos del destino que se registran durante el recorrido"	1.2.1. Parada turística
		1.2.2. Valoración
	1.3. Tiempo	1.3.1. Horas
		1.3.2. Días
		1.3.3. Meses
	1.4. Actividad	1.4.1. Toma de Fotos
		1.4.2. Comentarios
		1.4.3. Valoración
VARIABLE TEMÁTICA - Teoría	**EJES TEMÁTICOS - ¿Qué aspectos comprende la teoría?**	**SUB EJES TEMÁTICOS**
2. Offline First "Paradigma de Desarrollo Offline First"	2.1. Desarrollo Offline First	2.1.1. Paradigmas de Desarrollo Offline First
		2.1.2. Principios de diseño Offline First
		2.1.3. Diseño del Comportamiento Offline First
		2.1.4. Sincronización de Datos en Online
	2.2. Geolocalización	2.2.1. Geolocalización por GPS
	2.3. Aplicativo Móvil	2.3.1. React-Native
		2.3.2. Redux, Redux-React
		2.4.3. Redux-Offline
		2.5.4. React-Navigations

VARIABLE PROPOSITIVA - Solución	EJES PROPOSITIVOS - ¿Qué aspectos comprende mi respuesta? (Metodología: Scrum)	SUB EJES PROPOSITIVOS - Actividades
3. Aplicativo Móvil "Desarrollar un App que funcione offline y online para Gestión de rutas turísticas"	Análisis y Planificación	Diagramas de clases de diseño UML
		Historias de Usuario y Designación de tareas
	Codificación	Reuniones de 2 veces a la semana
		Codificación
	Pruebas y retrospectivas	Revisión del Sprint
		Despliegue y retrospectiva