

UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN



DESARROLLO DE UNA APLICACIÓN MÓVIL, PARA EL PROCESO DE VISITAS A CLÍNICAS DE LAS COMPAÑÍAS ASEGURADORAS.

Trabajo Especial de Grado presentado ante la ilustre

Universidad Central de Venezuela por

Br. Juan Andrés Vásquez Ferrer.

Para optar al título de Licenciado en Computación

Tutor:

Prof. Franky Uzcátegui

Caracas, Julio 2018

UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

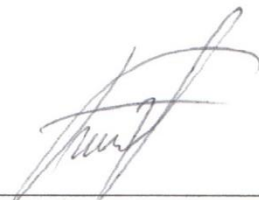
ESCUELA DE COMPUTACIÓN

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "**Desarrollo de una aplicación móvil, para el proceso de Visitas a Clínicas de las Compañías Aseguradoras.**" y presentado por el Br. **Juan Andrés Vásquez Ferrer** titular de la Cédula de Identidad **V-21.192.841**, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

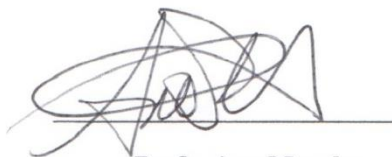
Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 28 de MAYO de 2018, a las 11,30 AM horas, para que el autor lo defendiera en forma pública, lo que este hizo en AULA POSTGRADO de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondiera a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 18 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 28 de MAYO de 2018.



Prof. Franky Uzcátegui

(Tutor)



Profa. Ana Morales

(Jurado)



Prof. Franklin Sandoval

(Jurado)

DEDICATORIA

A mis padres, por su apoyo incondicional, sin ustedes esto no sería posible.

A mis hermanos, por ser mi fuente de inspiración y brindarme los mejores momentos.

A mis amigos, por su compañía y apoyo en este camino.

A Dios, por cuidarme, guiarme y permitirme alcanzar esta meta.

Juan Andrés Vásquez Ferrer.

AGRADECIMIENTOS

Agradezco principalmente a mis padres, por educarme, guiarme y apoyarme a lo largo de mi vida. Gracias por enseñarme el valor del trabajo, que con esfuerzo y dedicación se puede lograr cualquier cosa. Gracias por todo.

A mi hermana, por motivarme y ayudarme siempre que lo necesite. Por ser un gran ejemplo de estudio y superación en todas mis etapas académicas. Gracias por tu amistad.

A todos mis amigos, gracias por todo su apoyo y compañía a lo largo de este largo camino. A mis amigos de la infancia y que aun comparten este trayecto conmigo Reinaldo y Miguel, gracias por su apoyo incondicional, espero que muy pronto logren esta meta también. A mis amigos que conocí durante este camino, espero sean muy exitosos en su carrera y logren todas las metas que se propongan. A todos mis amigos quiero que sepan que pueden seguir contando conmigo para lo que necesiten.

Quiero agradecer a Nebraska Meléndez, por todo tu apoyo a lo largo de este difícil camino, sin ti, este logro no sería realidad. Gracias por todos los buenos momentos, espero que podamos seguir cumpliendo todas nuestras metas de manera satisfactoria.

A mis profesores, gracias por brindarme todas las herramientas necesarias, que me permitieron hoy por hoy, ser el profesional que soy. Indudablemente su trabajo es uno de los más importantes en este país. Agradecimientos especiales a mi tutor Franky Uzcategui, por toda la orientación y experiencia brindada en este proceso de elaboración de mi trabajo especial de grado. Gracias por ser un guía en este difícil camino.

Agradezco a la Universidad Central de Venezuela, por ser mi segundo hogar, por brindarme experiencias que nunca olvidare, espero representar y ser un profesional a la altura de esta grande institución.

Por ultimo agradezco a Dios, gracias por guiarme en toda mi vida, por ser esa luz que siempre me orienta en cada decisión y camino que tomo.

A todos mis más sinceros agradecimientos.

UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

DESARROLLO DE UNA APLICACIÓN MÓVIL, PARA EL PROCESO DE
VISITAS A CLÍNICAS DE LAS COMPAÑÍAS ASEGURADORAS.

Autor: Juan Andrés Vásquez Ferrer.

Tutor: Prof. Franky Uzcategui Polo.

Fecha: Caracas, Mayo 2018.

RESUMEN

Las compañías de seguro ofrecen a sus asegurados del ramo de Salud el beneficio de Carta Aval, que tiene por finalidad presupuestar el costo de una intervención quirúrgica. Al momento de ingresar a la Clínica donde se realizará la operación, la compañía de seguro debe asegurarse que los servicios recibidos por el asegurado, sean los acordados en la Carta Aval y registrar las variaciones del presupuesto. Esta revisión se realiza a través del proceso de Visita a la Clínica. Este trabajo de investigación describe el desarrollo de una aplicación móvil que permite que el Visitador asignado, pueda realizar el recaudo de la información de los pacientes de manera automatizada, a través de un dispositivo móvil, sincronizando dicha información con el sistema web central, para agilizar el proceso de recolectar los datos necesarios. Para el desarrollo de esta solución se utilizaron diversas herramientas entre ellas el framework Ionic en su versión 3, el framework minimalista Flask, y la base de datos móvil SQLite, además se realizó una adaptación a la metodología Scrum ya que cumple con las necesidades de dicha investigación.

Palabras clave: Proceso, Compañía de Seguros, Salud, Aplicación Móvil, Visitador Clínico, Scrum.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	11
CAPÍTULO 1	13
PLANTEAMIENTO DEL PROBLEMA.....	13
1.1 Planteamiento del problema	13
1.2 Objetivos.....	15
1.2.1 Objetivo general	15
1.2.2 Objetivos específicos.....	15
1.3 Justificación	15
1.4 Alcance	15
CAPÍTULO 2	16
MARCO CONCEPTUAL	16
2.1 Sistema de Información	16
2.2 Aplicación informática	16
2.3 Aplicación móvil	17
2.4 Arquitectura Cliente-Servidor	17
2.5 Base de datos	18
2.5.1 Base de datos relacional	18
2.5.2 Base de datos no relacional	19
2.6 Base de datos móvil.....	19
2.6.1 SQLite	19
2.6.2 Berkeley DB.....	20
2.6.3 Couchbase Lite.....	20
2.7 Servicios web.....	21
2.7.1 REST.....	21
2.7.2 SOAP.....	22
2.8 Sistema operativo móvil	23
2.8.1 IOS	23
2.8.2 Android.....	23
2.8.3 Windows Phone.....	23
2.9 Aplicación móvil nativa.....	23

2.9.1	Java.....	24
2.9.2	Objective-C	24
2.9.3	Swift	24
2.9.4	C Sharp.....	24
2.10	Aplicación móvil híbrida.....	25
2.10.1	HTML 5.....	25
2.10.2	CSS.....	25
2.10.3	Javascript.....	25
2.10.4	Frameworks	26
2.10.5	Angular.....	26
2.10.6	iONIC.....	26
2.10.7	React.....	27
2.10.8	React Native	27
2.11	Aseguradora.....	27
2.11.1	Asegurado.....	27
2.11.2	Póliza de seguro	28
2.11.3	Póliza de seguro de salud	28
2.11.4	Proceso de supervisión de clínicas	28
CAPÍTULO 3		29
MARCO METODOLÓGICO		29
3.1	SCRUM.....	29
3.1.1	Proceso	29
3.1.2	Planificación de la iteración	30
3.1.3	Ejecución de la iteración	31
3.1.4	Inspección y adaptación	31
3.2	Adaptación de la metodología Scrum.....	32
CAPÍTULO 4		33
MARCO APLICATIVO		33
4.1	Descripción general de la solución.....	33
4.2	Aplicación de la metodología SCRUM.....	35
4.2.1	Lista de Objetivos.....	35
4.2.2	Lista de actividades de la iteración (Pila de Sprint).....	36
4.3	Requerimientos del sistema.....	38

4.3.1	Requerimientos funcionales	38
4.3.2	Requerimientos no funcionales	38
4.4	Descripción del flujo asociado a la solución	38
4.5	Análisis del modelo de datos y definición.....	40
4.5.1	Listado de tablas en la aplicación.....	40
4.6	Servicios web.....	41
5.2	Descripción de los módulos del sistema y sus interfaces	42
5.2.1	Descripción de los módulos	42
5.2.2	Interfaces del aplicativo móvil	43
5.3	Fase de pruebas.....	54
5.3.1	Pruebas funcionales.....	54
5.3.2	Pruebas de aceptación	56
CONCLUSIONES Y RECOMENDACIONES		63
BIBLIOGRAFÍA		65

ÍNDICE DE FIGURAS

Figura 1. <i>Metodología SCRUM</i> (https://proyectosagiles.org/que-es-scrum/).....	30
Figura 2. <i>Arquitectura propuesta. Fuente: El autor.</i>	34
Figura 3. <i>Diagrama solución propuesta. Fuente: El autor.</i>	39
Figura 4. <i>Modelo de datos de la aplicación. Fuente: El Autor.</i>	40
Figura 5. <i>Vista de inicio de sesión. Fuente: Captura de pantalla.</i>	43
Figura 6. <i>Vista Visitas clínicas pendientes. Fuente: Captura de pantalla.</i>	44
Figura 7. <i>Vista Registro de variación de presupuesto. Fuente: Captura de pantalla.</i> .	45
Figura 8. <i>Vista de Encuesta realizada al paciente. Fuente: Captura de pantalla.</i>	46
Figura 9. <i>Vista Lista de opciones de respuesta Fuente: Captura de pantalla.</i>	47
Figura 10. <i>Vista de Menú de funcionalidades. Fuente: Captura de pantalla.</i>	48
Figura 11. <i>Vista de Funcionalidad de localización Fuente: Captura de pantalla.</i>	49
Figura 12. <i>Vista Detalle de la ubicación. Fuente: Captura de pantalla.</i>	50
Figura 13. <i>Vista de Historial de Visitas clínicas. Fuente: Captura de pantalla.</i>	51
Figura 14. <i>Vista de Detalle de una encuesta. Fuente: Captura de pantalla.</i>	52
Figura 15. <i>Vista Notificación de sincronización manual. Fuente: Captura de pantalla.</i>	53
Figura 16. <i>Resultados de la pregunta N° 1. Fuente: Pruebas de aceptación con usuarios finales.</i>	58
Figura 17. <i>Resultados de la pregunta N° 2. Fuente: Pruebas de aceptación con usuarios finales.</i>	58
Figura 18. <i>Resultados de la pregunta N° 3. Fuente: Pruebas de aceptación con usuarios finales.</i>	59
Figura 19. <i>Resultados de la pregunta N° 4. Fuente: Pruebas de aceptación con usuarios finales.</i>	59
Figura 20. <i>Resultados de la pregunta N° 5. Fuente: Pruebas de aceptación con usuarios finales.</i>	60
Figura 21. <i>Resultados de la pregunta N° 6. Fuente: Pruebas de aceptación con usuarios finales.</i>	60
Figura 22. <i>Resultados de la pregunta N° 7. Fuente: Pruebas de aceptación con usuarios finales</i>	61
Figura 23. <i>Resultados de la pregunta N° 8. Fuente: Pruebas de aceptación con usuarios finales.</i>	61
Figura 24. <i>Resultados de la pregunta N° 9. Fuente: Pruebas de aceptación con usuarios finales.</i>	62

ÍNDICE DE TABLAS

Tabla 1. <i>Pila del producto. Fuente: El Autor.</i>	35
Tabla 2. <i>Pila del producto (Continuación). Fuente: El Autor.</i>	36
Tabla 3. <i>Instrumento evaluación de módulos del sistema. Fuente: El autor.</i>	54
Tabla 4. <i>Matriz de ciclos de pruebas. Fuente: El autor.</i>	55
Tabla 5. <i>Encuesta de aceptación. Fuente: El Autor</i>	57

INTRODUCCIÓN

El uso de la tecnología se ha convertido en una parte fundamental de nuestra vida cotidiana. Áreas educativas, comerciales y organizacionales aplican tecnologías en sus procesos para obtener mejores resultados y mayores beneficios. Esta última es una de las que más se han visto impulsadas por el desarrollo tecnológico, han automatizado procesos que eran realizados manualmente, han minimizado costos, han evolucionado y han mejorado la calidad de los servicios. Actualmente muchas empresas tienen la posibilidad de aplicar la tecnología a los distintos servicios que puedan ofrecer y a los procesos internos involucrados en la prestación estos servicios, con la intención de ahorrar en gastos y además, lograr que sus clientes y trabajadores tengan la mayor satisfacción posible.

Teniendo en cuenta la necesidad que tienen las empresas para automatizar sus procesos, el presente Trabajo Especial de Grado se centró en agilizar la gestión de un proceso conocido como Visita Clínica perteneciente a un sistema de Control de Visitas Clínicas, el cual es llevado a cabo por una compañía aseguradora en la cobertura de su seguro de hospitalización o de salud. Este proceso tiene como objetivo validar el monto total de gastos a cubrir por parte de la compañía aseguradora ante el tratamiento realizado por el asegurado, además de verificar la calidad de atención de sus clientes que se encuentran en las clínicas.

Esta gestión realizada por parte del ente asegurador genera varios inconvenientes y retrasos, los cuales quieren ser solventados y mejorados por la misma, a través del uso de tecnologías móviles. En tal sentido este trabajo consiste en la investigación de herramientas y mejores prácticas para solventar el problema en cuestión.

Para profundizar en el desarrollo de la aplicación móvil anteriormente descrita, el presente documento de Trabajo Especial de Grado se encuentra dividido en cuatro capítulos que se detallaran a continuación:

Planteamiento del problema (Capítulo 1): En este capítulo se expone la problemática que se quiere solventar a través del presente Trabajo Especial de Grado y la solución que se desarrolla para solventarla. De igual forma se presenta el objetivo general, objetivos específicos, justificación y alcance de este trabajo.

Marco Conceptual (Capítulo 2): En este capítulo se presentan conceptos relacionados con los Sistemas de Información, Bases de datos móviles, Arquitectura Cliente – Servidor, Aplicaciones móviles y las principales tecnologías utilizadas para el desarrollo de la solución.

Marco Metodológico (Capítulo 3): En este capítulo se presenta la Metodología de Desarrollo SCRUM, la cual fue utilizada para guiar el desarrollo de la Aplicación Web sobre la cual se centró el Trabajo Especial de Grado.

Marco Aplicativo (Capítulo 4): en este capítulo se presenta la Metodología de Desarrollo SCRUM adaptada a las necesidades de la Aplicación Móvil desarrollada, detallando las iteraciones llevadas a cabo.

Por último se presentan conclusiones, recomendaciones y referencias bibliográficas del presente Trabajo Especial de Grado.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

1.1 Planteamiento del problema

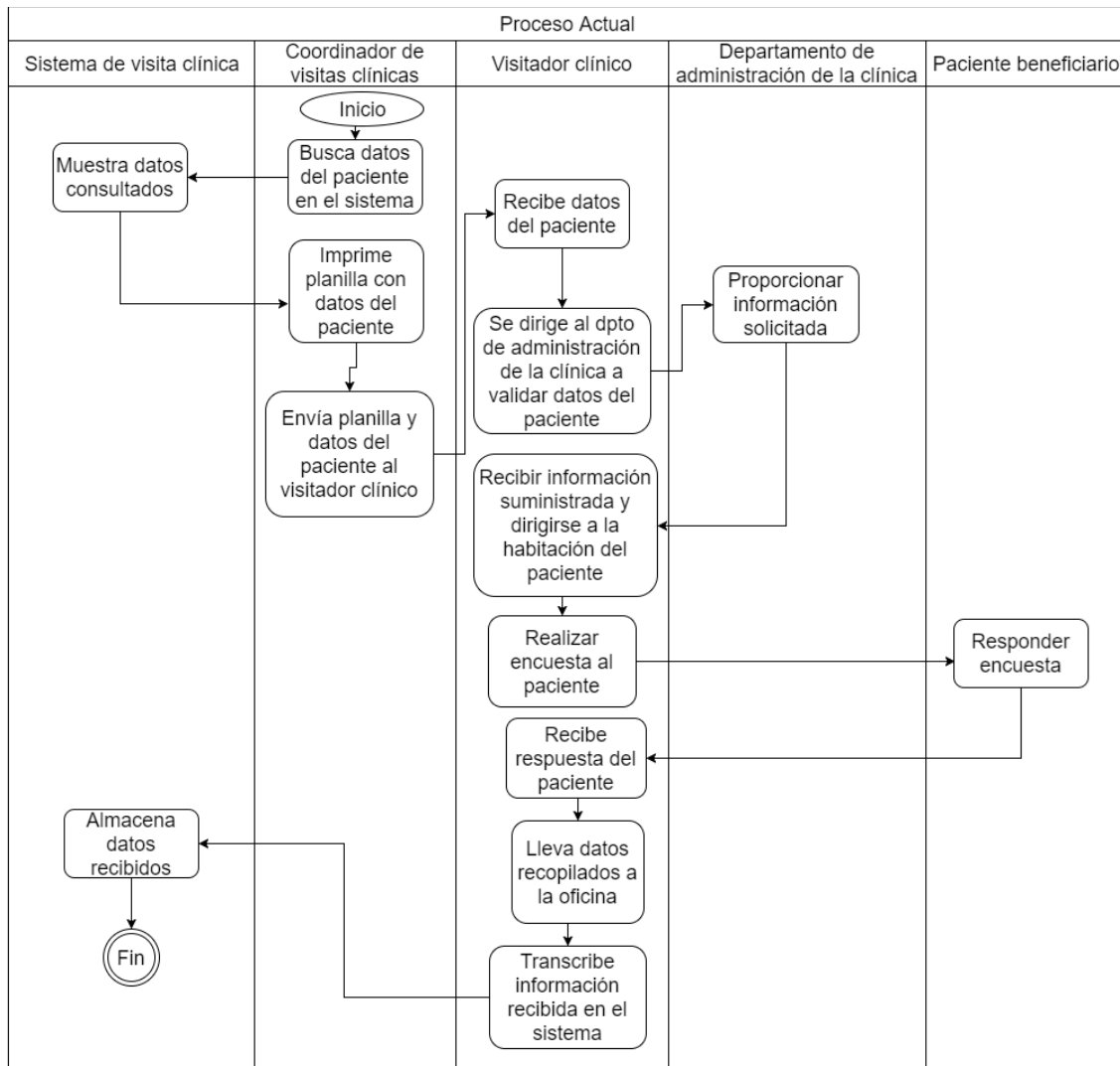
El proceso de supervisión de visitas clínicas es un proceso mediante el cual, un empleado de una compañía de seguros, hace una visita técnica a la clínica en la cual residen o atienden a sus clientes, recaudando información sobre la estadía de estos en las clínicas. Con el objetivo de hacer una evaluación o análisis de dicha información recopilada que permita validar los datos y así ofrecer un mejor servicio.

Una persona encargada de la supervisión, tiene que realizar el recaudo de información a sus clientes del servicio recibido en una clínica. Esta información se recauda de manera física y posteriormente es cargada en un sistema para analizar dichos resultados. Al realizar el recaudo manualmente existe una alta probabilidad que los datos obtenidos contengan errores, lo cual es un gran problema ya que los mismos arrojaran resultados erróneos los cuales no son útiles para la organización.

Otro problema asociado a dicho proceso, es el tiempo invertido en el mismo, ya que una vez recaudado los datos de los clientes, tienen que ser llevados a una oficina para así cargarlos en un sistema.

También hay que considerar que actualmente no se cuenta con una herramienta capaz de verificar los datos de los clientes en las clínicas con el objetivo de evitar fraudes en el uso de las pólizas de seguro.

En el siguiente diagrama se describe el proceso de visitas clínicas actual:



Ante dicho problema, se plantea una solución empleando tecnologías móviles, donde un supervisor de clínica pueda realizar el recaudo de información de los clientes de manera automatizada a través de un dispositivo móvil, sincronizando dicha información con el sistema web ya existente para agilizar el proceso de obtención de resultados.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar una aplicación móvil, para el proceso de visitas clínicas de las compañías aseguradoras.

1.2.2 Objetivos específicos

- Analizar requerimientos del sistema.
- Diseñar prototipo y vistas del aplicativo móvil.
- Desarrollar funcionalidades del aplicativo móvil.
- Desarrollar una capa de web services que soporte las funcionalidades del aplicativo móvil.
- Realizar pruebas funcionales, no funcionales y de calidad.

1.3 Justificación

Una aplicación móvil es una herramienta eficaz y eficiente para automatizar el proceso de visitas de clínicas, ya que un supervisor de clínicas debe trasladarse a distintas localidades donde se encuentran sus clientes.

Con una herramienta de este tipo el supervisor podrá recaudar de una manera óptima los datos, reduciendo drásticamente los tiempos del proceso actual y obteniendo mayor precisión en la obtención de resultados.

1.4 Alcance

El alcance de este proyecto se basa en la realización de un aplicativo móvil, un sistema web y el diseño e implementación del API Rest para la comunicación de estos sistemas, esto permitirá la automatización del proceso de supervisión de clínicas. Sin embargo este proyecto se encuentra abierto a futuras integraciones con otros sistemas relacionados al proceso de supervisión de clínicas.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1 Sistema de Información

Un sistema de información se puede definir técnicamente como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización [1].

Por definición es un conjunto de elementos que interactúan entre sí con un fin común; que permite que la información esté disponible para satisfacer las necesidades en una organización, un sistema de información no siempre requiere contar con recurso computacional aunque la disposición del mismo facilita el manejo e interpretación de la información por los usuarios.

Los elementos que interactúan entre sí son: el equipo computacional (cuando este disponible), el recurso humano, los datos o información fuente, programas ejecutados por las computadoras, las telecomunicaciones y los procedimientos de políticas y reglas de operación [2].

2.2 Aplicación informática

En informática, una aplicación es un software diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas. Esto lo diferencia principalmente de otros tipos de programas, como los sistemas operativos (que hacen funcionar la computadora), las utilidades (que realizan tareas de mantenimiento o de uso general), y las herramientas de desarrollo de software (para crear programas informáticos) [3].

2.3 Aplicación móvil

Una aplicación móvil, es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles y que permite al usuario efectuar una tarea concreta de cualquier tipo: profesional, de ocio, educativa, de acceso a servicios, etc. Facilitando las gestiones o actividades a desarrollar [4].

Una App es una aplicación de software que se instala en dispositivos móviles o tabletas para ayudar al usuario en una labor concreta, ya sea de carácter profesional o de ocio y entretenimiento [5].

2.4 Arquitectura Cliente-Servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se dividen entre dos partes. Una parte son los proveedores de recursos o servicios, conocidos como servidores, y la otra parte son los demandantes de estos recursos o servicios, conocidos como clientes.

En esta arquitectura los recursos y servicios se centralizan en el servidor al cual los clientes se encuentran conectados. Para que los clientes puedan acceder a estos recursos o servicios, deben realizar peticiones, las cuales serán recibidas y procesadas por el servidor. La comunicación entre ambas partes se lleva a cabo a través de algún protocolo de comunicación, el cual suele ser el protocolo HTTP [6].

Arquitectura Cliente servidor. Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes. Ayuda a comprender las bases sobre las que están contruidos los algoritmos distribuidos [7].

2.5 Base de datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos [8].

2.5.1 Base de datos relacional

Una base de datos relacional es una base de datos que se trata como un conjunto de tablas y se manipula de acuerdo con el modelo de datos relacional. Contiene un conjunto de objetos que se utilizan para almacenar y gestionar los datos, así como para acceder a los mismos. Las tablas, vistas, índices, funciones, activadores y paquetes son ejemplos de estos objetos.

Una base de datos relacional particionada es una base de datos relacional cuyos datos se gestionan repartidos en múltiples particiones (también denominadas nodos). Esta separación de los datos entre particiones es transparente para los usuarios de la mayoría de sentencias de SQL. Sin embargo, algunas sentencias DLL (lenguaje de definición de datos) tienen en cuenta la información de las particiones (por ejemplo, `CREATE DATABASE PARTITION GROUP`). (DLL, lenguaje de definición de datos, es el subconjunto de sentencias de SQL que se utilizan para describir las relaciones de los datos de una base de datos.)

Una base de datos federada es una base de datos relacional cuyos datos están almacenados en varias fuentes de datos (tales como bases de datos relacionales separadas). Los datos son tratados como si pertenecieran a una sola gran base de datos y se pueden acceder mediante las consultas SQL normales. Los cambios en los datos se pueden dirigir explícitamente hacia la fuente datos apropiada [48].

2.5.2 Base de datos no relacional

NoSQL es un término que describe las bases de datos no relacionales de alto desempeño. Las bases de datos NoSQL utilizan varios modelos de datos, incluidos los de documentos, gráficos, claves-valores y columnas. Las bases de datos NoSQL son famosas por la facilidad de desarrollo, el desempeño escalable, la alta disponibilidad y la resiliencia.

Las bases de datos no relacionales (NoSQL) no suelen contener un esquema. Se suele utilizar una clave de partición para recuperar valores, conjuntos de columnas o documentos JSON o XML semiestructurados, así como otros documentos que contengan atributos de elementos relacionados.

Habitualmente, las bases de datos NoSQL intercambian algunas de las propiedades ACID de los sistemas de administración de bases de datos relacionales (RDBSM) tradicionales por un modelo de datos más flexible que se escala de forma horizontal. Estas características convierten las bases de datos NoSQL en una elección excelente en las situaciones en las que los RDBMS detectan desafíos en la arquitectura a la hora de superar una combinación de cuellos de botella de desempeño, escalabilidad, complejidad operativa y el aumento de los costos de administración y soporte [49].

2.6 Base de datos móvil

Es una base de datos que está alojada en un teléfono móvil. Para posteriormente sincronizar los datos con una base de datos remota [9].

2.6.1 SQLite

Es la base de datos seleccionada por Research in Motion, Apple y Google para servir de base de datos local en sus aplicaciones desplegadas en los sistemas operativos para dispositivos móviles Blackberry OS, Blackberry PlayBook OS, iOS y Android. Actualmente es la base de datos más utilizada del mercado.

Debe su popularidad a que es de código abierto, consume muy pocos recursos y no necesita servicios instalados para su funcionamiento.

A diferencia de los sistemas de gestión de bases de datos de servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones [10].

2.6.2 Berkeley DB

Berkeley DB es una familia de librerías de código abierto que le permiten administrar sus datos en la forma relacional tradicional (usando SQL) o como pares clave-valor (donde las claves y los valores son matrices de bytes) o como objetos Java o como documentos XML. Independientemente del API que elija, Berkeley DB promete indexación, almacenamiento en caché, transacciones, acceso de único escritor / múltiples lectores, búsqueda de texto completo (para la interfaz SQL), recuperación automática de fallas, replicación, compresión y encriptación. También admite bases de datos en memoria y en disco.

Otra característica intrigante de Berkeley DB es que la API SQL que proporciona es totalmente compatible con la de SQLite. Como resultado, puede hacer que sus aplicaciones SQLite utilicen Berkeley DB como el motor de almacenamiento subyacente sin tener que volver a escribirlas. Todo lo que necesita hacer es vincularlos a una versión adecuada de la biblioteca Berkeley DB. Se ha informado que esta combinación de SQLite y Berkeley DB tiene un mejor rendimiento en el caso de aplicaciones de escritura intensiva y muy concurrente.

Berkeley DB proporciona enlaces API para varios lenguajes de programación, incluidos C ++ y Java, y se ha compilado, ejecutado y probado en varios sistemas operativos, incluidos Android e iOS [50].

2.6.3 Couchbase Lite

Couchbase Lite es una base de datos orientada a documentos, los datos se almacenan como documentos JSON. Cada documento puede tener uno o más archivos adjuntos, que son datos binarios esencialmente no interpretados que se almacenan y cargan por separado del documento en sí. Couchbase Lite admite índices persistentes (llamados vistas) y usa map-reduce para administrarlos y consultarlos. Couchbase Lite, junto con Couchbase Sync Gateway y Couchbase Server, son los tres componentes de la solución

de base de datos móvil NoSQL JSON que ofrece Couchbase llamada Couchbase Mobile. Una de las características clave de Couchbase Mobile es la sincronización integrada (a través de Couchbase Sync Gateway) entre bases de datos locales (Couchbase Lite) y bases de datos en la nube (Couchbase Server), que libera al desarrollador de la carga de escribir su propio código de sincronización. Para respaldar esta característica, Couchbase Lite viene con un mecanismo de resolución de conflictos bastante similar al usado por Git. Couchbase Lite proporciona una API nativa para Android e iOS, así como complementos para dos herramientas multiplataforma ampliamente utilizadas para el desarrollo móvil: PhoneGap y Xamarin [51].

2.7 Servicios web

Un web service o servicio web es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet [11].

También se define como un método de comunicación entre dos aparatos electrónicos en una red. Es una colección de protocolos abiertos y estándares usados para intercambiar datos entre aplicaciones o sistemas. Las aplicaciones escritas en varios lenguajes de programación que funcionan en plataformas diferentes pueden utilizar web services para intercambiar información a través de una red. La interoperabilidad, por ejemplo entre Java y Python o Windows y Linux se debe al uso de estándares abiertos [12].

2.7.1 REST

La Transferencia de Estado Representacional (en inglés Representational State Transfer) o REST es un estilo de arquitectura software para sistemas hipertexto distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo [13].

REST, REpresentational State Transfer, es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP. REST nos permite crear servicios y

aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC.

REST se definió en el 2000 por Roy Fielding, coautor principal también de la especificación HTTP. Podríamos considerar REST como un framework para construir aplicaciones web respetando HTTP.

Por lo tanto REST es el tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a Internet [14].

2.7.2 SOAP

SOAP (originalmente las siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML [15].

Simple Object Access Protocol (SOAP) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

SOAP puede formar la capa base de una "pila de protocolo de web service", ofreciendo un framework de mensajería básica en la cual los web services se puedan construir. Este protocolo basado en XML consiste de tres partes: un sobre (envelope), el cual define qué hay en el mensaje y cómo procesarlo, un conjunto de reglas de codificación para expresar instancias de tipos de datos y una convención para representar llamadas a procedimientos y respuestas. El protocolo SOAP tiene tres características principales:

Extensibilidad: seguridad y WS-routing son extensiones aplicadas en el desarrollo.

Neutralidad: SOAP puede ser utilizado sobre cualquier protocolo de transporte como SMTP, TCP o JMS, como así también sobre HTTP.

Independencia: SOAP permite cualquier modelo de programación [16].

2.8 Sistema operativo móvil

Un sistema operativo móvil o SO móvil es un conjunto de programas de bajo nivel que permite la abstracción de las peculiaridades del hardware específico del teléfono móvil y provee servicios a las aplicaciones móviles, que se ejecutan sobre él. Al igual que los PCs que utilizan Windows, Linux o Mac OS, los dispositivos móviles tienen sus sistemas operativos como Android, IOS, Windows Phone o BlackBerry OS, entre otros. Los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos [17].

2.8.1 IOS

iOS es el sistema operativo diseñado por Apple para sus productos, iPhone, iPad, iPod Touch, y Apple TV , otros dispositivos como el iPod Nano y el iWatch utilizan otro sistema más básico y dirigido a una función más específica basado en iOS porque incorpora algunos de sus gestos e iconos y además se pueden sincronizar con teléfonos o Tablets. Presentado en 2007 junto con el primer teléfono de la compañía dedicada en sus inicios a revolucionar el mundo de los ordenadores de mesa y portátiles [18].

2.8.2 Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma [19].

2.8.3 Windows Phone

Windows Phone es un sistema operativo móvil desarrollado por Microsoft como sucesor de Windows Mobile. A diferencia de su predecesor está enfocado en el mercado de consumo en lugar del mercado empresarial [20].

2.9 Aplicación móvil nativa

Una aplicación nativa es una app que ha sido desarrollada específicamente para cada plataforma, utilizando el mismo lenguaje que usa el smartphone o tableta desde donde va a usarse. Es decir, una app nativa para Android se desarrolla en lenguaje Java, mientras que para iOS se emplea el lenguaje Objective-C o Swift [21].

Las aplicaciones móviles nativas son aquellas desarrolladas y optimizadas específicamente para el sistema operativo determinado y la plataforma de desarrollo del fabricante, iOS, Android o Windows Phone, adaptando a cada uno el lenguaje con el que se desarrolla: lenguaje Objective-C /SWIFT para iOS, Java para Android, y .Net para Windows Phone.

Este tipo de aplicaciones se adapta al 100% con las funcionalidades y características del dispositivo obteniendo así una mejor experiencia de uso. Sin embargo, el desarrollo de una aplicación nativa comporta un mayor coste, puesto que si se desea realizar una aplicación multiplataforma se ha de realizar una nueva versión para cada sistema operativo, multiplicando así los costes de desarrollo [22].

2.9.1 Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes [23].

2.9.2 Objective-C

Objective-C. Es un lenguaje de programación orientado a objetos y basado en mensajes, altamente dinámico. Este lenguaje consiste en un pequeño número de incrementos a ANSI C, y está caracterizado por su aplazamiento a muchas decisiones hasta que se cumple el tiempo de corrida del programa creado, soportando envíos dinámicos, escritura y carga dinámica [24].

2.9.3 Swift

Swift es un intuitivo lenguaje de programación creado por Apple que permite diseñar apps para iOS, Mac, el Apple TV y el Apple Watch. Está pensado para dar a los desarrolladores más libertad que nunca [25].

2.9.4 C Sharp

C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se

ejecutan en .NET Framework. Puede utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y muchas tareas más [26].

2.10 Aplicación móvil híbrida

Las aplicaciones híbridas son aplicaciones móviles diseñadas en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un framework que permite adaptar la vista web a cualquier vista de un dispositivo móvil. En otras palabras, no son más que una aplicación construida para ser utilizada o implementada en distintos sistemas operativos móviles, tales como, iOS, Android o Windows Phone, evitándose la tarea de crear una aplicación para cada sistema operativo. De esta manera, una aplicación híbrida puede ser adaptada a múltiples plataformas móviles sin crear nuevos códigos, pero ajustándose a algunos cambios operacionales para cada uno de ellos [27].

2.10.1 HTML 5

HTML5 (HyperText Markup Language) es la quinta revisión del lenguaje de marcado estándar que se emplea para la web. Es uno de los lenguajes de marcado más usados en todo el mundo y la razón es bastante obvia: gracias a HTML5 podemos crear la estructura de una página web. Texto, imágenes y material multimedia pueden mostrarse correctamente gracias a HTML5 [28].

2.10.2 CSS

Corresponde a la expresión inglesa Cascading StyleSheets, que puede traducirse como “Hojas de estilo en cascada”. El concepto se utiliza en el ámbito de la informática para referirse a un lenguaje empleado en el diseño gráfico. El lenguaje CSS permite presentar, de manera estructurada, un documento que fue escrito en un lenguaje de marcado. Se usa especialmente en el diseño visual de un sitio web cuando las páginas se hallan escritas en XML o HTML [29].

2.10.3 Javascript

JavaScript es un robusto lenguaje de programación que puede ser aplicado a un documento HTML y usado para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, co-fundador del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla [30].

2.10.4 Frameworks

Es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto [31].

2.10.5 Angular

Es un Framework que facilita la creación de aplicaciones web. Angular combina plantillas declarativas, inyección de dependencias herramientas de extremo a extremo y las mejores prácticas para ser aplicadas en el desarrollo FrontEnd [32].

Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o ser recuperados de recursos JSON estáticos o dinámicos [33].

2.10.6 iONIC

Ionic es una herramienta, gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS, el cual se basa en el framework AngularJS y Angular en sus versiones más recientes [34].

Ionic es una herramienta, gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con ANGULAR. Ionic está construido para ser rápido gracias a la mínima manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware.

Ionic utiliza Angular con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con Angular.

Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android. Lo interesante, como sabéis, es que desarrollas una vez, y compilas para varios sistemas operativos

Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc[35].

2.10.7 React

Es una biblioteca Javascript de código abierto para crear interfaces de usuario con el objetivo de animar al desarrollo de aplicaciones en una sola página. Es mantenido por Facebook, Instagram y una comunidad de desarrolladores independientes y compañías [36].

2.10.8 React Native

React-Native es una plataforma desarrollada por Facebook, open source para el desarrollo de aplicaciones móviles cross-platform completamente nativas utilizando React [37].

2.11 Aseguradora

Una aseguradora, también designada en algunas partes como compañía de seguros, puede ser una persona física o una empresa que se ocupa de la aseguración de riesgos a terceros, es decir, su principal actividad consiste en resguardar a aquellos bienes materiales de determinados riesgos a los que están expuestos [43].

2.11.1 Asegurado

El asegurado es el titular del interés, objeto del contrato de seguro, pudiendo ser una persona física o una persona jurídica, aunque en los seguros de personas, normalmente, tiene que ser una persona física [44].

2.11.2 Póliza de seguro

Una póliza de seguro es un documento que le da validez al contrato de seguro realizado y firmado en conformidad tanto por el asegurado como por el asegurador, en el cual se especifican las normas, los derechos y las obligaciones de las partes involucradas [45].

2.11.3 Póliza de seguro de salud

Su póliza de seguro de salud es un acuerdo entre usted y su compañía de seguros. La póliza hace una lista de un paquete de beneficios médicos tales como pruebas, medicamentos y servicios de tratamiento. La compañía de seguro se compromete a cubrir el costo de ciertos beneficios que están anotados en su póliza [46].

2.11.4 Proceso de supervisión de clínicas

Es un proceso mediante el cual, un empleado de la compañía de seguros, hace una visita técnica a la clínica en la cual residen o atienden a sus clientes, recaudando información sobre la estadía de estos en las clínicas. Con el objetivo de hacer una evaluación o análisis de dicha información recopilada que permita validar los datos y así ofrecer un mejor servicio [47].

CAPÍTULO 3

MARCO METODOLÓGICO

3.1 SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales [42].

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

Ver en detalle cuales son los beneficios de Scrum, sus fundamentos y sus requisitos.

3.1.1 Proceso

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

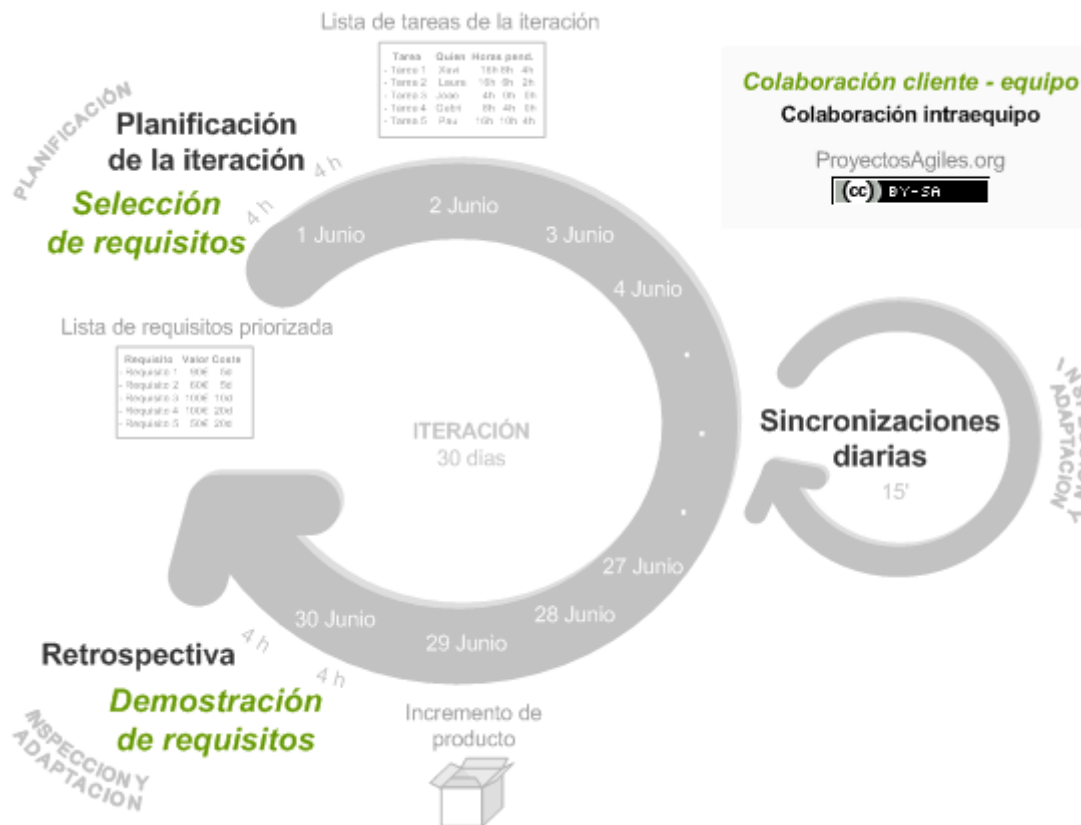


Figura 1. Metodología SCRUM (<https://proyectosagiles.org/que-es-scrum/>)

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas.

Las actividades que se llevan a cabo en Scrum son las siguientes:

3.1.2 Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

- Selección de requisitos (4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
- Planificación de la iteración (4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido.

La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas.

3.1.3 Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 minutos máximo), normalmente delante de un tablero físico o pizarra (Scrum Taskboard). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el Facilitador (Scrum Master) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

Durante la iteración, el cliente junto con el equipo refinan la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o planifican los objetivos del proyecto para maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

3.1.4 Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

- Demostración (4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las

adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.

- Retrospectiva (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados [42].

3.2 Adaptación de la metodología Scrum

El desarrollo de este trabajo se realizó bajo una adaptación de la metodología Scrum, ya que está orientada a la productividad, posee gran capacidad de adaptación, permite desarrollar soluciones con estándares de calidad, y gracias a la comunicación permanente con el cliente se pueden realizar los ajustes necesarios en cada etapa del proyecto para garantizar que el producto final cumpla los requerimientos establecidos. Para el desarrollo de la aplicación, se realizarán cuatro iteraciones (Sprints), donde cada una tendrá una duración de cuatro semanas aproximadamente.

El proceso contará con reuniones semanales para detallar los avances de cada sprint.

Siguiendo los lineamientos de la metodología se definieron tres roles básicos:

Product Owner: son los beneficiarios y las personas interesadas en el desarrollo de la solución, en este caso, las empresas aseguradoras beneficiadas por este sistema.

Team: Personas encargadas del desarrollo de la aplicación, en este caso el Bachiller Juan Andres Vásquez Ferrer.

Scrum Master: Persona encargada de liderar el equipo de desarrollo, este rol le corresponde al profesor Franky Uzcategui, tutor del TEG.

CAPÍTULO 4

MARCO APLICATIVO

En este capítulo se presenta una descripción general acerca de la solución planteada para agilizar el proceso de Visitas Clínicas de las compañías aseguradoras, la arquitectura implementada para el desarrollo de la solución, el conjunto de tecnologías utilizadas para implementar la solución y la forma en que se aplicó la metodología de desarrollo ágil SCRUM para guiar el desarrollo de esta solución. De igual manera, se detallan los requerimientos a cumplir, los perfiles de usuarios que interactúan con el sistema así como las funcionalidades que abarcan los módulos desarrollados.

4.1 Descripción general de la solución

Con la intención de agilizar el proceso de Visitas Clínicas de las compañías aseguradoras, se planteó desarrollar para el presente Trabajo Especial de Grado, una aplicación móvil, que permitiese a los visitantes clínicos de la compañía de seguros, hacer el recaudo y levantamiento de información requerida en el proceso de visitas clínicas, de una manera más fácil y automatizada. A través de la aplicación móvil se podrán consultar todas las visitas clínicas que tiene asignada un visitador, realizar una encuesta a un paciente asegurado asignado, verificar los datos de la carta aval de la póliza de seguro, registrar algún cambio en el presupuesto cubierto por dicha carta aval, obtener la ubicación e información de las clínicas donde se encuentra cada paciente asegurado asignado. Además, la aplicación permite sincronizar de forma automática todos los datos recopilados con el sistema de visitas clínicas.

Para el desarrollo de esta aplicación, se planteó la siguiente arquitectura:

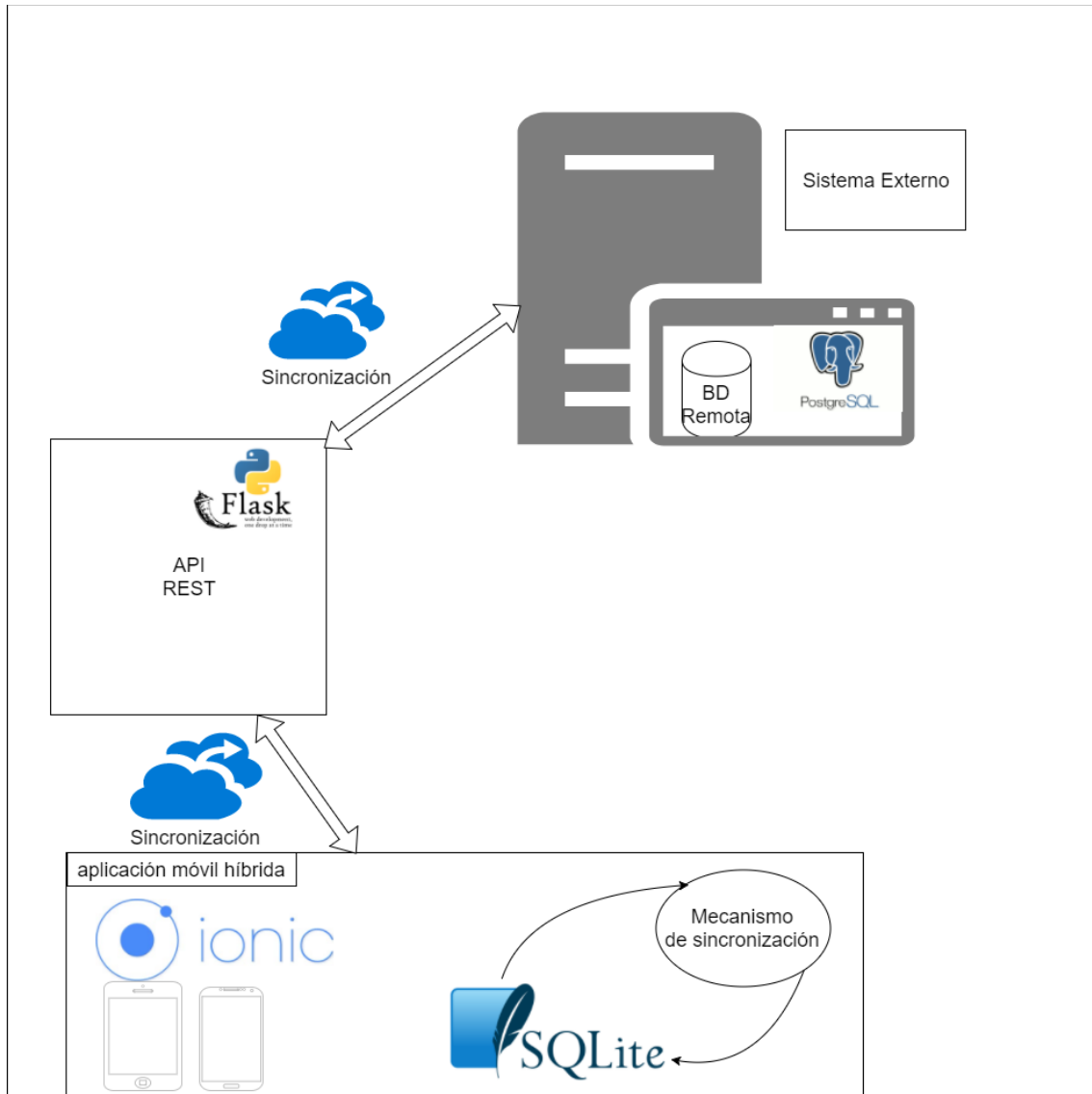


Figura 2 *Arquitectura propuesta. Fuente: El autor.*

Para el desarrollo del cliente móvil, se utilizará el framework IONIC en su versión 3, este framework nos permite hacer una aplicación móvil híbrida, que se ejecute tanto en sistemas operativos Android como iOS, sin la necesidad de realizar múltiples desarrollos para cada sistema. La aplicación contará con una base de datos móvil SQLite para almacenar todos los datos recogidos localmente, y posteriormente mediante una rutina de sincronización enviar todos los datos almacenados cuando exista una conexión a internet disponible.

Para el desarrollo de la capa de web services, se utilizará el micro framework Flask escrito en el lenguaje de programación Python. Flask es un framework minimalista que nos da la posibilidad de crear un servidor web, que nos servirá para desarrollar un API Rest que permita a nuestro aplicativo móvil conectarse a una base de datos relacional postgresSQL perteneciente a un sistema externo, con el objetivo de enviar los datos

recopilados a dicho sistema. Además el API también proveerá una interfaz para que futuras aplicaciones puedan consumir sus servicios.

4.2 Aplicación de la metodología SCRUM

Los requerimientos acordados con el cliente (la compañía aseguradora) se clasifican de la siguiente forma:

4.2.1 Lista de Objetivos

No	Sprint	Actividad	Duración
1	Analizar requerimientos del sistema.	<ul style="list-style-type: none"> Definición de los requerimientos del negocio. 	2 Semanas
		<ul style="list-style-type: none"> Instalación de las herramientas de desarrollo necesarias. 	1 Semana
		<ul style="list-style-type: none"> Configuración de los ambientes de desarrollo. 	1 Semana
2	Diseñar prototipo y vistas del aplicativo móvil.	<ul style="list-style-type: none"> Creación de prototipos de la aplicación móvil. 	1 Semana
		<ul style="list-style-type: none"> Maquetado de las vistas. 	2 Semanas
		<ul style="list-style-type: none"> Elaboración del modelo de datos de la aplicación. 	1 Semana
3	Desarrollar funcionalidades del aplicativo móvil.	<ul style="list-style-type: none"> Desarrollo de la funcionalidad de autenticación. Desarrollo de un servicio web que permita la obtención de datos asociados a un visitador. Desarrollo de un servicio web que permita la carga de datos asociados a un visitador. 	2 Semanas

Tabla 1. Pila del producto. Fuente: El Autor.

No	Sprint	Actividad	Duración
		<ul style="list-style-type: none"> ● Desarrollo de la funcionalidad listar todas las visitas clínicas pendientes de un visitador. ● Desarrollo de la funcionalidad realizar encuesta a paciente asegurado asignado. ● Desarrollo de la funcionalidad registrar variaciones de presupuesto de la póliza de seguro. ● Desarrollo de la funcionalidad ver historial de visitas realizadas. ● Desarrollo de la funcionalidad ver ubicación del paciente asegurado en un mapa. ● Desarrollo de la funcionalidad ver datos de la clínica donde se encuentra un paciente asignado. ● Desarrollo de la funcionalidad sincronización de datos automática y manual. 	2 Semanas
4	Realizar pruebas funcionales, no funcionales y de calidad al sistema.	<ul style="list-style-type: none"> ● Realización de pruebas no funcionales. ● Realización de pruebas de calidad 	4 Semanas

Tabla 2. Pila del producto (Continuación). Fuente: El Autor.

4.2.2 Lista de actividades de la iteración (Pila de Sprint)

- **Sprint 1:** Se inició con el análisis y definición de los requerimientos del negocio, una vez definido esto, se seleccionaron las herramientas tecnologías necesarias para el desarrollo de la aplicación y posteriormente se procedió a la instalación, las cuales fueron Python Flask 3.6.5, Ionic 3 y SQLite. Finalmente se configuró todo el ambiente de desarrollo para su correcto funcionamiento.

- **Sprint 2:** En este Sprint se realizó el diseño y creación de prototipos de la aplicación móvil. También se realizó el maquetado de las vistas de la aplicación utilizando las guías de diseño proporcionadas por el framework Ionic 3 para la creación de interfaces de usuario usables. Luego de esto, se creó el modelo de datos utilizado por la aplicación.

- **Sprint 3:** En el siguiente Sprint se desarrollaron las funcionalidades de la aplicación, tanto del lado de backend como del lado de front end. Este proceso se dividió en dos (2) etapas:
 - **Creación de web services:** En el cual se desarrollaron todos los servicios necesarios para que la aplicación tuviese un correcto funcionamiento. Los servicios desarrollados fueron:
 - Autenticación de usuarios: este servicio permite identificar a un visitador clínico en la aplicación.
 - Sincronización de usuario: este servicio permite sincronizar los datos recopilados por un visitador clínico con un sistema central
 - Visitas clínicas: este servicio permite obtener todas las visitas clínicas asignadas a un visitador.
 - Validación de presupuesto: este servicio registra el cambio de un presupuesto de una póliza de seguros específica.

 - **Funcionalidades de la aplicación móvil:** Se desarrollaron las funcionalidades de autenticación, listar todas las visitas clínicas disponibles, realizar una encuesta a un paciente asegurado, registrar alguna modificación en el presupuesto de la póliza de seguro, ver el historial de visitas clínicas, ver la ubicación de un paciente asegurado así como los datos de la clínica donde se encuentra, iniciar el proceso de sincronización de los datos recopilados de manera manual. También se desarrolló la funcionalidad de inicio del proceso de sincronización de manera automática.

- **Sprint 4:** Se realizaron las pruebas funcionales necesarias para validar el correcto funcionamiento de la aplicación móvil y de los servicios web. Además de la realización de pruebas no funcionales como lo son disponibilidad, usabilidad, rendimiento, seguridad, accesibilidad, y de calidad al sistema desarrollado.

4.3 Requerimientos del sistema

De acuerdo a los requerimientos, se definen los siguientes requerimientos

4.3.1 Requerimientos funcionales

- Autenticación de usuarios.
- Consulta de visitas clínicas asignadas.
- Realización de encuesta a un paciente asegurado asignado.
- Registro de cambio de cobertura en póliza de seguro.
- Consulta del histórico de visitas clínicas.
- Consulta de ubicación de los pacientes asignados así como la información de la clínica donde se encuentra.
- Sincronización de los datos recopilados.

4.3.2 Requerimientos no funcionales

- Validar las entradas de los datos tanto en el aplicativo móvil como en los servicios web.
- Usar tecnologías multiplataforma que permitan ejecutar la aplicación en diferentes sistemas operativos móviles.
- Interfaces de usuario usables e intuitivas.

4.4 Descripción del flujo asociado a la solución

Primeramente se encuentra el visitador clínico, que es el encargado de realizar las visitas clínicas a los distintos pacientes asegurados asignados. El visitador clínico sincroniza los datos de la aplicación con los sistemas de visitas clínicas. Una vez realizado esto, el visitador puede ver una lista de todas las visitas que tiene asignadas. El visitador puede acceder a información de utilidad, como la ubicación de las clínicas a las cuales se tiene que dirigir así como también a los datos de dichas clínicas, en caso de que los necesite. Una vez el visitador se dirige a la clínica, el mismo procede a dirigirse al departamento administrativo para validar los datos de la carta aval, si existe alguna modificación en el presupuesto cubierto por la póliza de seguro, el visitador procede a

registrarlo a través de la aplicación. Una vez verificado los datos de la carta aval, el visitador se dirige a donde se encuentra el paciente asegurado o en su defecto un familiar o representante, para realizar una encuesta al mismo. El visitador observa todas las preguntas que necesita realizar a través del dispositivo móvil y registra las respuestas obtenidas. Por último el visitador tiene la posibilidad de sincronizar estos datos de manera manual con el sistema de visitas clínicas, seleccionando la opción de sincronización. O simplemente puede esperar que el aplicativo sincronice los datos obtenidos de manera automática, cuando detecte una conexión a internet activa.

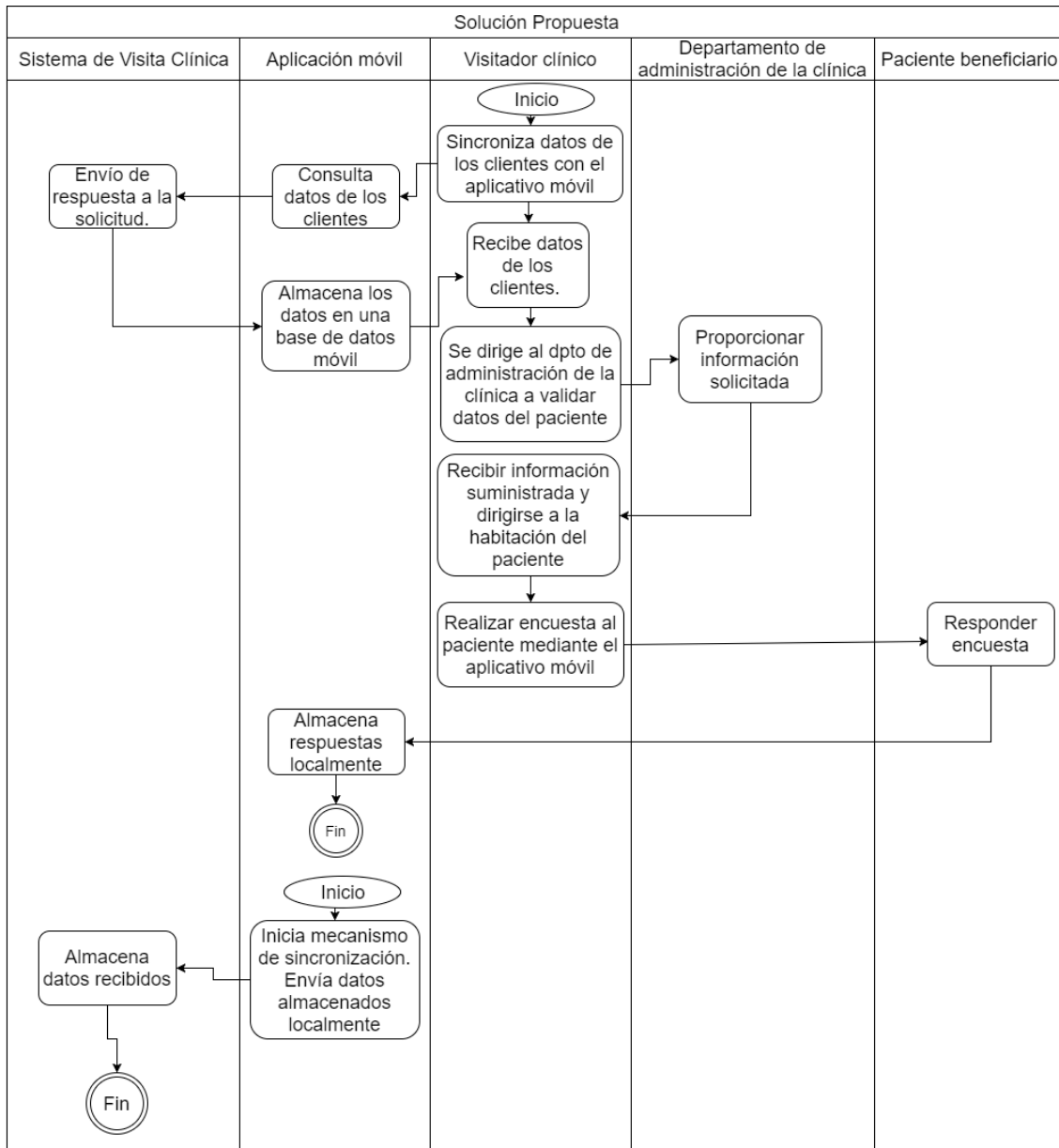


Figura 3. Diagrama solución propuesta. Fuente: El autor.

4.5 Análisis del modelo de datos y definición

Para el almacenamiento de los datos requeridos para el funcionamiento de la aplicación móvil se crearon 7 tablas. Para poder almacenar y consultar dichos datos de forma local en el dispositivo.

4.5.1 Listado de tablas en la aplicación

- **Clinic:** Tabla utilizada para almacenar los datos correspondientes a las clínicas afiliadas a la compañía aseguradora.
- **Clinical Visitor:** Tabla que almacena la información de un visitador clínico.
- **Patient:** Tabla que almacena la información de un paciente asegurado.
- **Quiz:** Utilizada para almacenar la información correspondiente a una encuesta.
- **Question:** Tabla que almacena la información correspondiente a una pregunta.
- **Option:** Tabla que almacena la información correspondiente a una opción para contestar una pregunta.

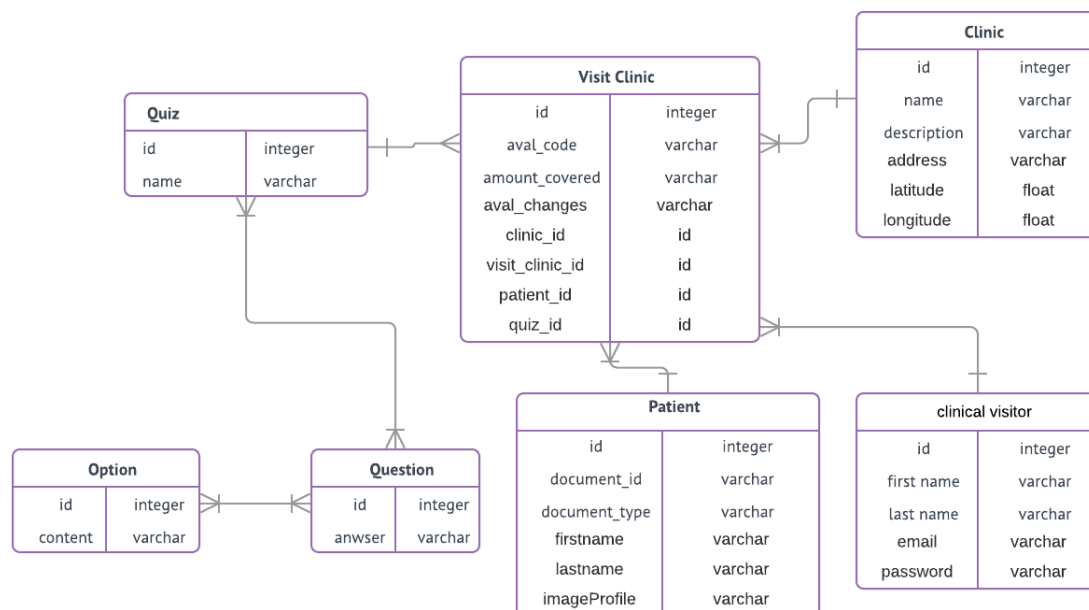


Figura 4. Modelo de datos de la aplicación. Fuente: El Autor.

4.6 Servicios web

La aplicación móvil se comunica con un sistema de visitas clínicas central, a través de servicios web, los cuales fueron desarrollados bajo el estándar REST. A continuación se describen dichos servicios implementados para el correcto funcionamiento de la aplicación:

1 **Nombre:** Login.

1.1 **Método HTTP:** POST

1.1.1 **Descripción:** Servicio que permite al usuario identificarse iniciando una sesión en la aplicación.

1.1.2 **Entrada:** Nombre de usuario y contraseña.

1.1.3 **Salida:** “Access Token” necesario para utilizar los otros servicios del sistema.

2 **Nombre:** User.

2.1 **Método HTTP:** GET

2.1.1 **Descripción:** Servicio utilizado para consultar la información del visitador clínico.

2.1.2 **Entrada:** “Access Token”

2.1.3 **Salida:** datos correspondientes al visitador clínico.

3 **Nombre:** User/Sincronization.

3.1 **Método HTTP:** POST

3.1.1 **Descripción:** Servicio utilizado para sincronizar los datos almacenados en la base de datos local del dispositivo con el sistema central.

3.1.2 **Entrada:** “Access Token”, datos almacenados a sincronizar.

3.1.3 **Salida:** Notificación de éxito o error.

3.2 **Método HTTP:** GET

3.1.1 **Descripción:** Servicio utilizado para sincronizar los datos almacenados en el sistema central, con la base de datos local del dispositivo.

3.1.2 **Entrada:** “Access Token”.

3.1.3 **Salida:** Visitas clínicas pendientes, correspondientes al visitador clínico.

4 **Nombre:** User/history.

4.1 **Método HTTP:** GET

4.1.1 **Descripción:** Servicio utilizado para obtener todo el historial de visitas clínicas de un visitador clínico.

4.1.2 **Entrada:** “Access Token”.

4.1.3 **Salida:** Historial de visitas clínicas.

5 **Nombre:** Logout.

5.1 **Método HTTP:** DELETE

5.1.1 **Descripción:** Servicio invalida el “Access token” cerrando la sesión del usuario.

5.1.2 **Entrada:** “Access token”

5.1.3 **Salida:** Notificación de éxito o error.

5.2 Descripción de los módulos del sistema y sus interfaces

5.2.1 Descripción de los módulos

La aplicación móvil se encuentra dividida en dos (2) módulos descritos a continuación:

- **Módulo de visita clínica:** Este módulo está dedicado al visitador clínico. A través de este módulo el visitador cuenta con una serie de herramientas que le facilitaran sus tareas cotidianas. El visitador puede ver todas las visitas clínicas pendientes asignadas así como el historial de todas las visitas clínicas realizadas. Puede acceder a la ubicación de los pacientes e información de las clínicas donde estos se encuentran. También tiene la posibilidad de registrar un cambio de presupuesto en el sistema a través de este módulo. El visitador también podrá realizar las encuestas de calidad a los pacientes a través del aplicativo.
- **Módulo de sincronización:** Este módulo se encarga de sincronizar los datos almacenados localmente en el dispositivo con el sistema de visitas clínicas central y viceversa. Este proceso es disparado automáticamente cuando el dispositivo detecta una conexión a internet activa. Dicho proceso también puede ser disparado manualmente mediante una opción que se encuentra en el menú principal.

5.2.2 Interfaces del aplicativo móvil

- **Vista correspondiente al inicio de sesión:** En esta vista el usuario introduce su nombre de usuario y contraseña para realizar la autenticación.



Figura 5. Vista de inicio de sesión. Fuente: Captura de pantalla.

- **Vista correspondiente a las visitas clínicas pendientes de un visitador:** Esta es la vista inicial, una vez el visitador inicia sesión, en ella se puede ver un listado de todas las visitas clínicas que el visitador tiene que realizar en su jornada laboral.

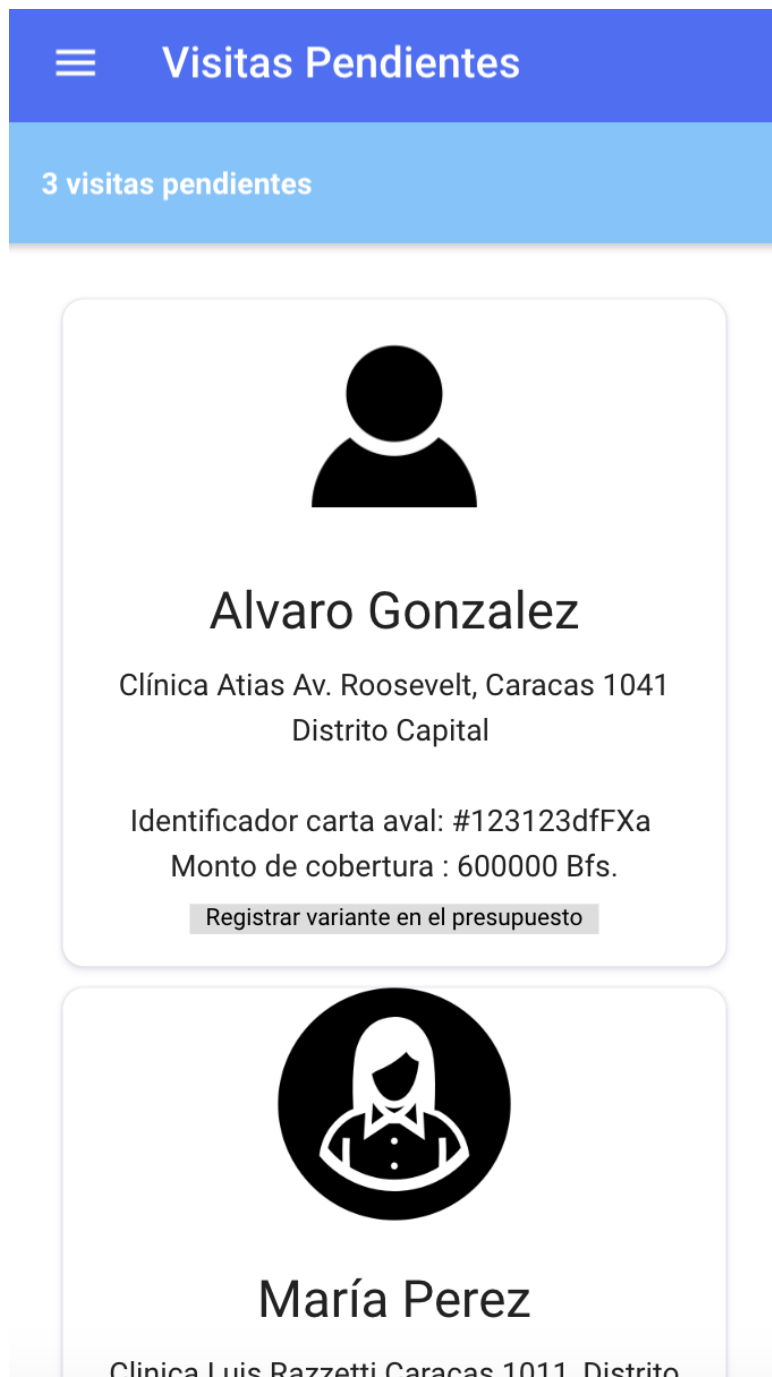


Figura 6. Vista *Visitas clínicas pendientes*. Fuente: *Captura de pantalla*.

- **Vista correspondiente al registro de variación del presupuesto:** En esta vista un visitador clínico puede registrar alguna variación con respecto al presupuesto que cubre la carta aval con el presupuesto reflejado en la clínica.

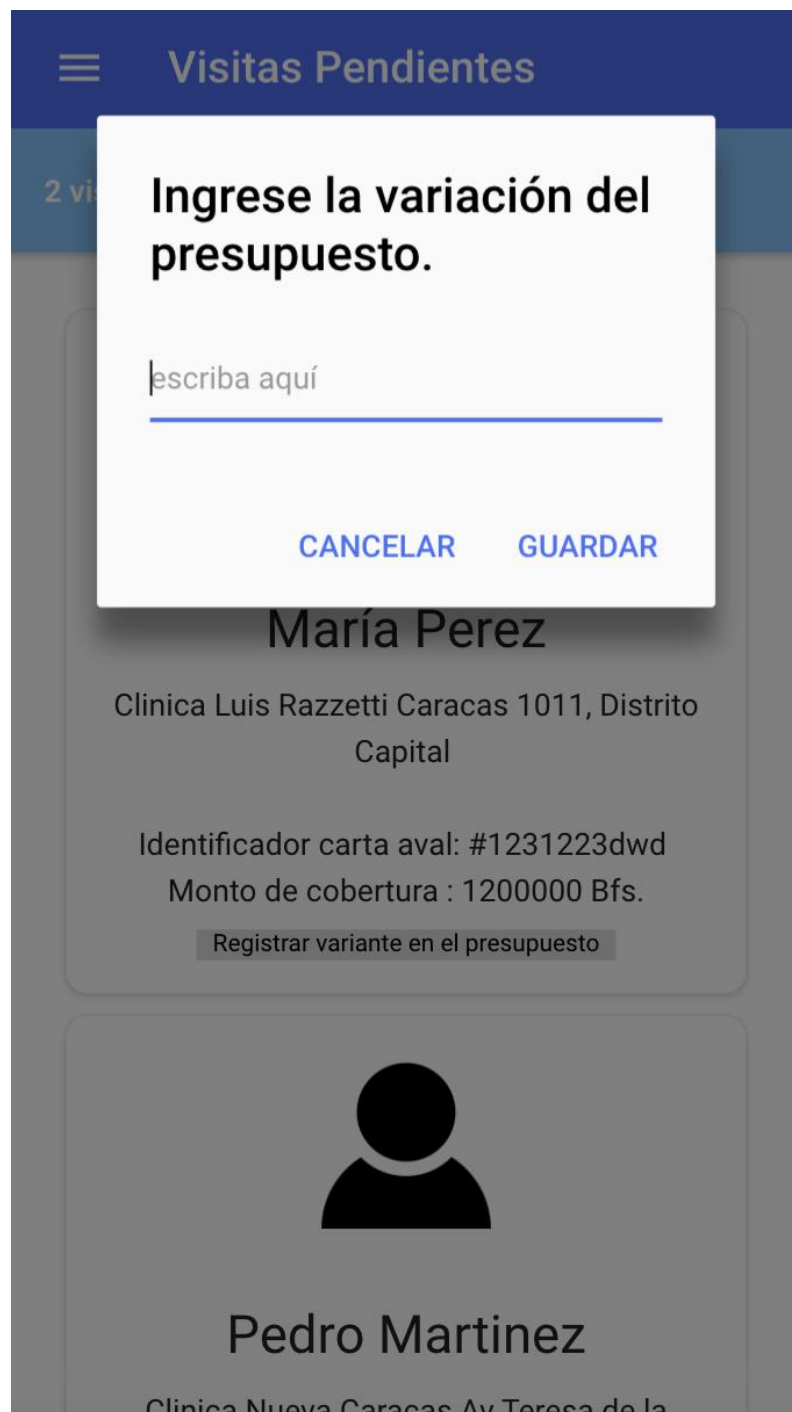


Figura 7. Vista Registro de variación de presupuesto. Fuente: Captura de pantalla.

- **Vista correspondiente a la realización de encuesta a un paciente asegurado:**
En esta vista un visitador clínico puede realizar la encuesta a través del dispositivo, registrando todas las respuestas necesarias proporcionadas por el paciente o representante legal.

← Alvaro Gonzalez

Preguntas

¿Qué tal fue la atención en la clínica?
Opciones ▼

¿Qué le pareció la comida de la clínica?
Opciones ▼

¿Le proporcionaron los medicamentos que necesito?
Opciones ▼

¿Volvería a esta clínica?
Opciones ▼

Figura 8. Vista de Encuesta realizada al paciente. Fuente: Captura de pantalla.

- **Vista correspondiente al listado de opciones disponibles:** En esta vista se puede ver el listado de opciones disponibles para la pregunta en cuestión.

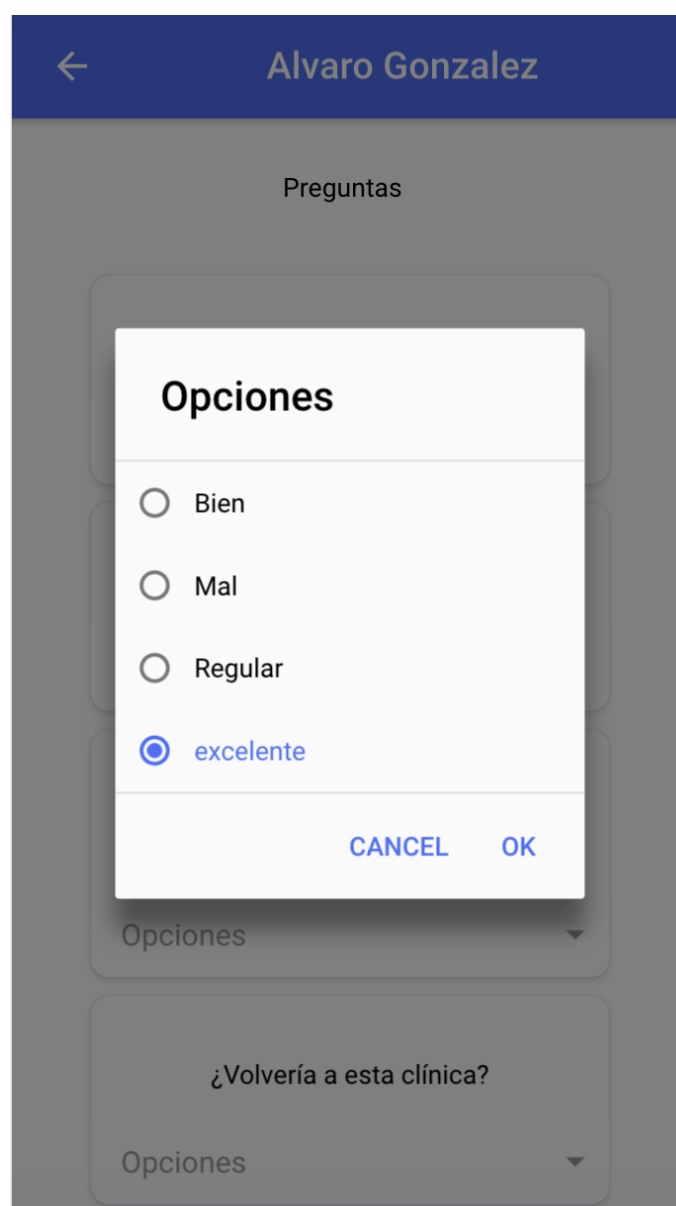


Figura 9. Vista Lista de opciones de respuesta Fuente: Captura de pantalla.

- **Vista correspondiente al menú de funcionalidades:** En esta vista se pueden ver y acceder a funcionalidades extra de la aplicación.

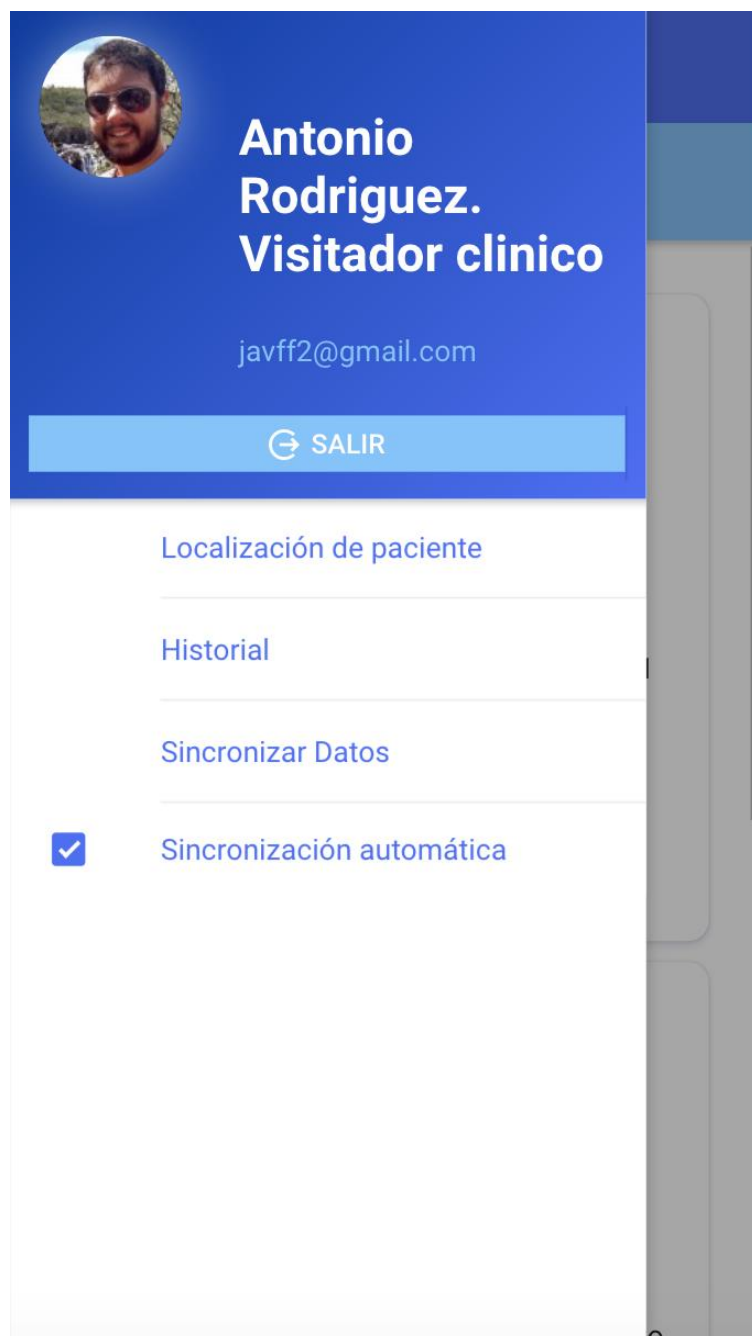


Figura 10. Vista de Menú de funcionalidades. Fuente: Captura de pantalla.

- **Vista correspondiente a la funcionalidad de localización:** En esta vista se puede ver el listado de pacientes con visitas pendientes para buscar la información de la clínica en la cual se encuentra ubicado.

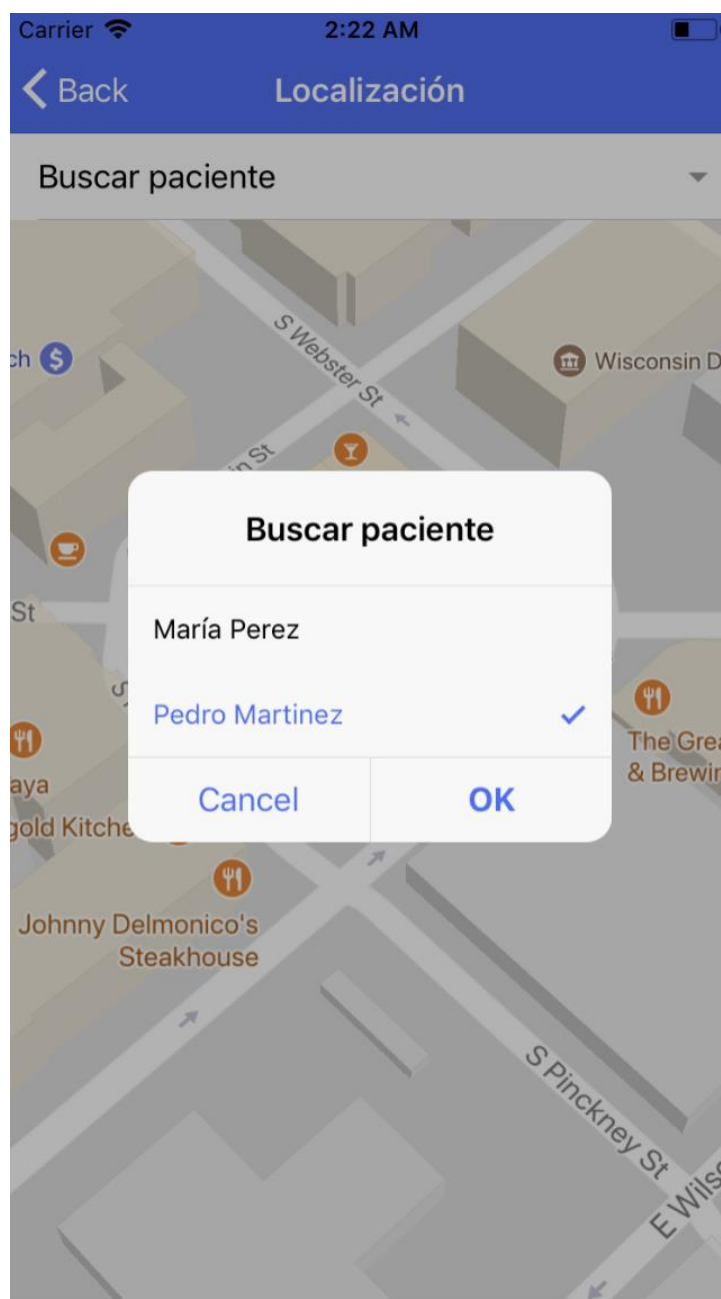


Figura 11 Vista de Funcionalidad de localización Fuente: Captura de pantalla.

- **Vista correspondiente al detalle de ubicación:** En esta vista se puede ver la información correspondiente de la búsqueda realizada.



Figura 12. Vista Detalle de la ubicación. Fuente: Captura de pantalla.

- **Vista correspondiente a la funcionalidad de historial:** En esta vista se puede ver el listado de visitas clínicas realizadas.

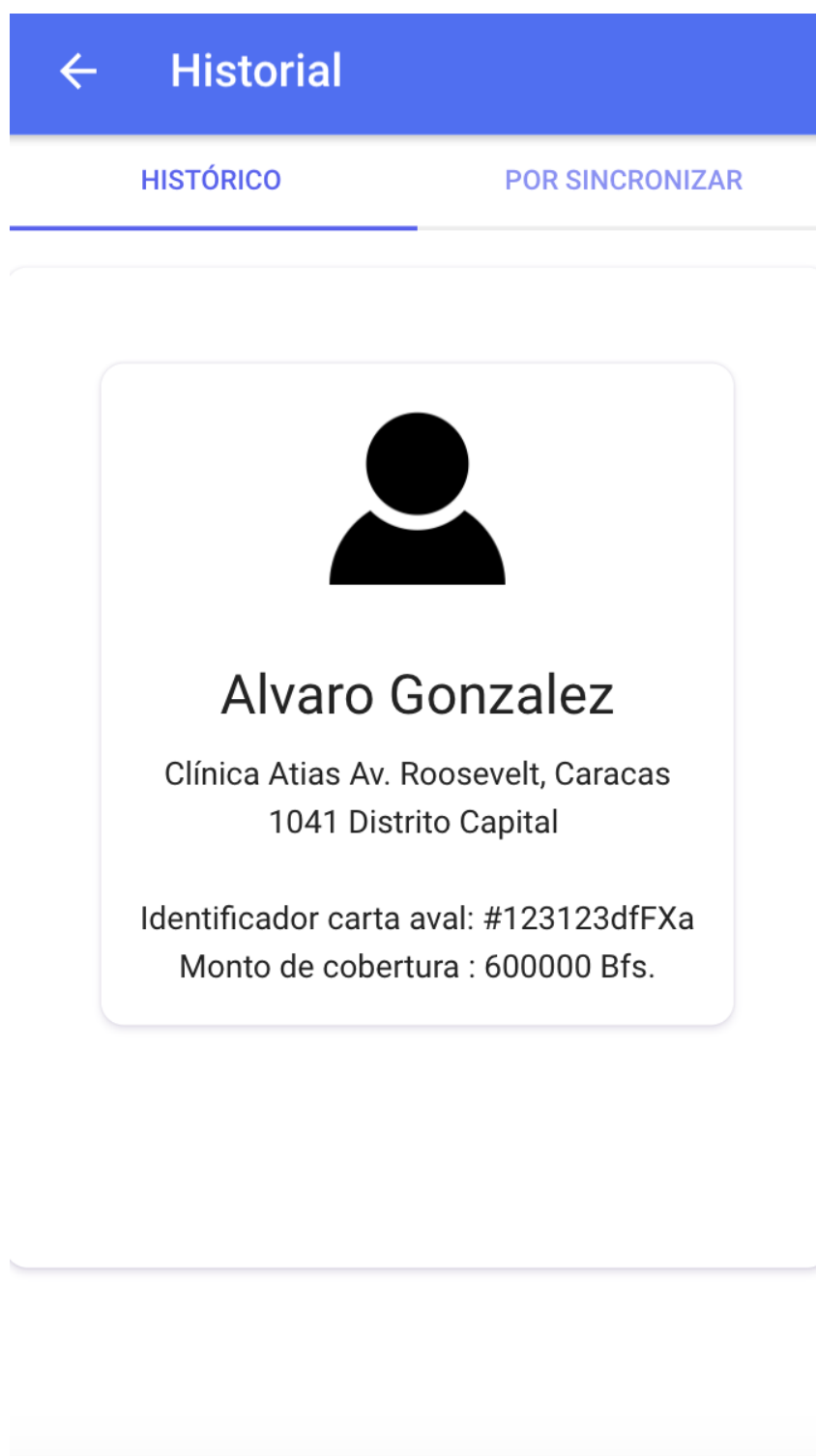


Figura 13. Vista de Historial de Visitas clínicas. Fuente: Captura de pantalla.

- **Vista correspondiente de detalle:** En esta vista se puede ver el detalle de los datos recopilados en una visita clínica ya realizada.

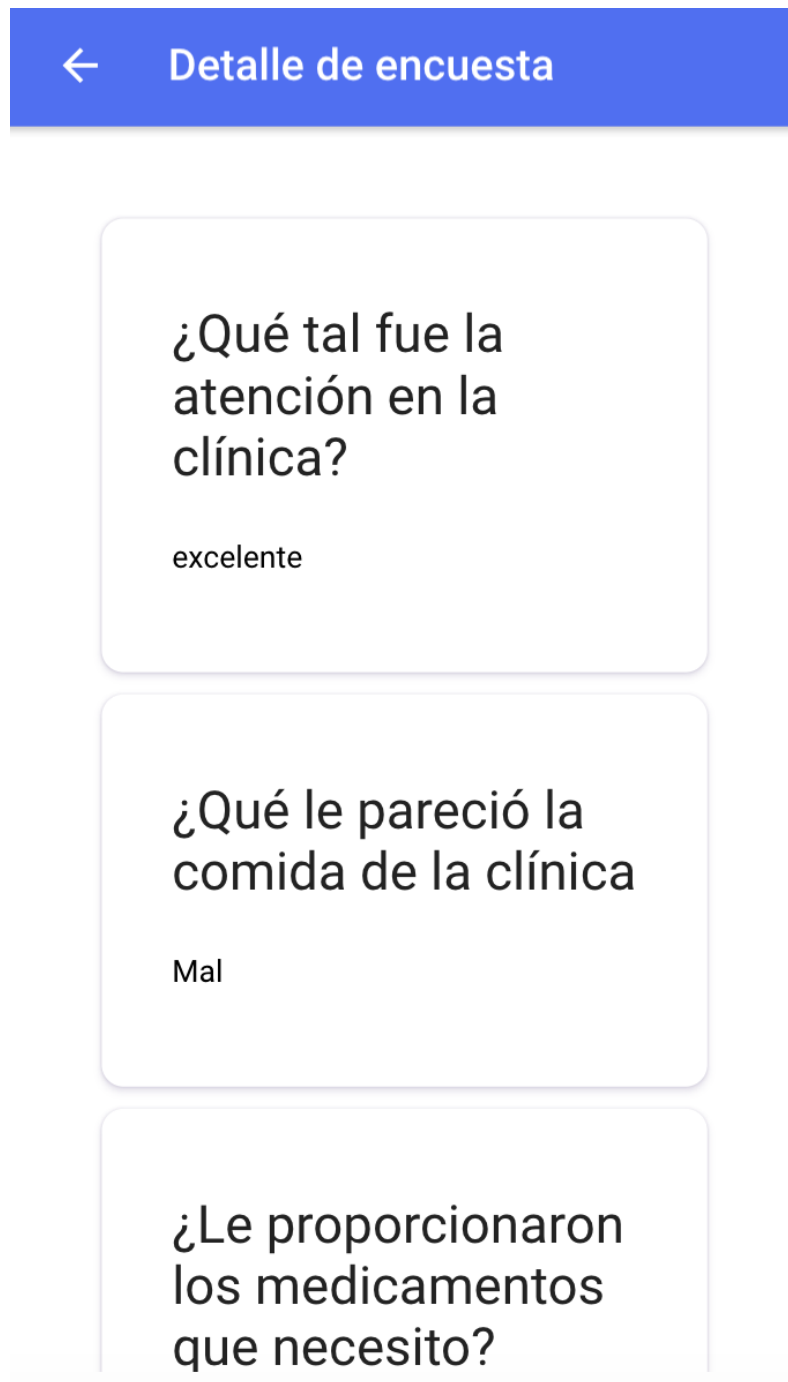


Figura 14. Vista de Detalle de una encuesta. Fuente: Captura de pantalla.

- **Vista correspondiente al proceso de sincronización:** En esta vista se puede disparar el proceso de sincronización de datos con el sistema central de forma manual.

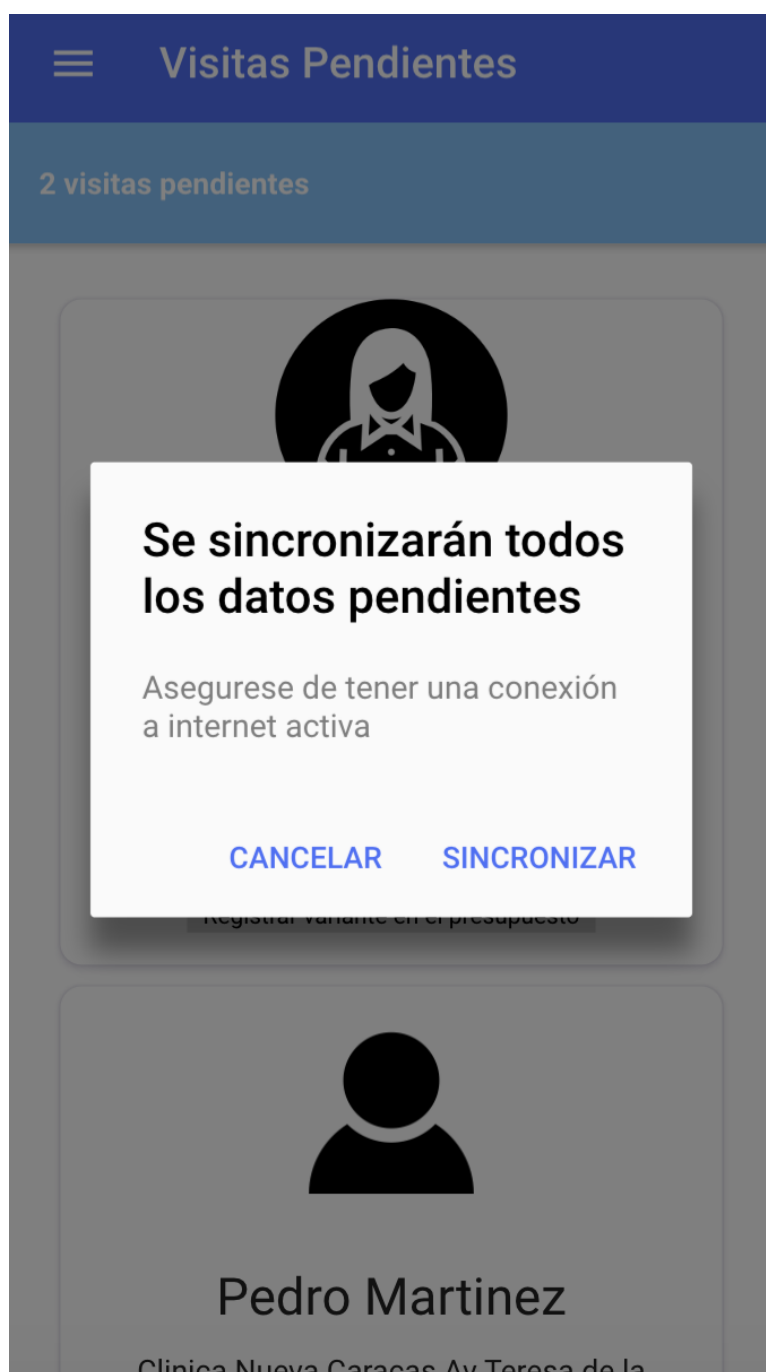


Figura 15. Vista Notificación de sincronización manual. Fuente: Captura de pantalla.

5.3 Fase de pruebas

Las pruebas realizadas para verificar el correcto funcionamiento del sistema y la aceptación de los usuarios por el mismo se dividieron en dos (2) partes a continuación se explica detalladamente las mismas.

5.3.1 Pruebas funcionales

Para las siguientes pruebas funcionales se realizó una tabla en donde se listan las funcionalidades requeridas que se quieren evaluar en el módulo junto con el estatus de aprobación (para cada funcionalidad) por parte del usuario final del sistema. Estas pruebas fueron realizadas de forma cíclica hasta que finalmente se cumplieran las funcionalidades requeridas para el mismo. En estas pruebas participaron dos (2) visitantes clínicos, un (1) supervisor de visitas clínicas y un analista de sistemas.

Módulo	Funcionalidades probadas	Resultados
Visitas clínicas	<ul style="list-style-type: none"> • Autenticación de usuarios. • Consulta de visitas clínicas asignadas. • Realización de encuesta a un paciente asegurado asignado. • Registro de cambio de cobertura en póliza de seguro. • Consulta del histórico de visitas clínicas. • Consulta de ubicación de los pacientes asignados así como la información de la clínica donde se encuentra. • Visualizar ubicación de los pacientes asignados a través de un mapa. • Registrar variaciones de presupuesto de la póliza de seguro. 	
Sincronización	<ul style="list-style-type: none"> • Sincronización manual de los datos recopilados. • Sincronización automática de los datos recopilados. 	

Tabla 3. Instrumento evaluación de módulos del sistema. Fuente: El autor.

Las pruebas presentadas en esta tabla se fueron realizando mediante ciclos hasta lograr la completa aprobación de los módulos.

	Ciclo de prueba		
Usuario	1	2	3
1	<ul style="list-style-type: none"> ○ 0 Módulos aprobados. ○ 2 Módulos no aprobados 	<ul style="list-style-type: none"> ○ 1 Módulo aprobado. ○ 1 Módulo no aprobado 	<ul style="list-style-type: none"> ○ 2 Módulos aprobados.
2	<ul style="list-style-type: none"> ○ 0 Módulos aprobados. ○ 2 Módulos no aprobados 	<ul style="list-style-type: none"> ○ 1 Módulo aprobado. ○ 1 Módulo no aprobado 	<ul style="list-style-type: none"> ○ 2 Módulos aprobados.
3	<ul style="list-style-type: none"> ○ 0 Módulos aprobados. ○ 2 Módulos no aprobados 	<ul style="list-style-type: none"> ○ 1 Módulo aprobado. ○ 1 Módulo no aprobado 	<ul style="list-style-type: none"> ○ 2 Módulos aprobados.
4	<ul style="list-style-type: none"> ○ 0 Módulos aprobados. ○ 2 Módulos no aprobados 	<ul style="list-style-type: none"> ○ 1 Módulo aprobado. ○ 1 Módulo no aprobado 	<ul style="list-style-type: none"> ○ 2 Módulos aprobados.

Tabla 4. Matriz de ciclos de pruebas. Fuente: El autor.

○ **Ciclo 1:**

Usuario 1: De los 2 módulos probados, 2 terminaron en un estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:

- No se podía visualizar la ubicación de los pacientes asignados a través de un mapa.
- Error al llenar la encuesta del paciente asignado.
- No se visualizaba el historial de visitas clínicas.
- Error en la actualización de presupuesto de la póliza de seguro.

Usuario 2: De los 2 módulos probados, 2 terminaron en un estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:

- Error en la actualización de presupuesto de la póliza de seguro
- Error al llenar la encuesta del paciente asignado.
- No se visualizaba el historial de visitas clínicas.

Usuario 3: De los 2 módulos probados, 2 terminaron en un estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:

- No se visualizaba el historial de visitas clínicas.
- Error al llenar la encuesta del paciente asignado.
- No se visualizaba el historial de visitas clínicas.
- Fallaba la sincronización automática de los datos recopilados

Usuario 4: De los 2 módulos probados, 2 terminaron en un estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:

- Error al llenar la encuesta del paciente asignado.
- Fallaba la sincronización automática de los datos recopilados.
- No se visualizaba el historial de visitas clínicas.

○ **Ciclo 2:**

Usuario 1: De los 2 módulos probados, 1 termino en un estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:

- No se podía visualizar la ubicación de los pacientes asignados a través de un mapa.
- Error al llenar la encuesta del paciente asignado.

Usuario 2: De los 2 módulos probados, 1 termino en un estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:

- Error en la actualización de presupuesto de la póliza de seguro
- Error al llenar la encuesta del paciente asignado.

Usuario 3: De los 2 módulos probados, 1 termino en un estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:

- No se visualizaba el historial de visitas clínicas.
- Error al llenar la encuesta del paciente asignado.

Usuario 4: De los 2 módulos probados, 1 termino en un estatus de no aprobación. Las funcionalidades que fallaron fueron las siguientes:

- Error al llenar la encuesta del paciente asignado.

○ **Ciclo 3:** En este ciclo todos los usuarios aprobaron todas las funcionalidades de los dos módulos.

5.3.2 Pruebas de aceptación

Para comprobar el nivel de aceptación del sistema, se realizó una encuesta basada en el modelo de usabilidad definido por Nielsen, J. (2003). Dicha encuesta fue realizada a un grupo de diez (10) personas, con los siguientes perfiles:

- Dos (2) visitantes clínicos.
- Un (1) analista de sistema.
- Un (1) supervisor de visitas clínicas.
- Un (1) diseñador grafico.
- Tres (3) desarrolladores web.
- Un (1) desarrollador Mobile.
- Una (1) persona del área administrativa.

De esta manera se pudo obtener una respuesta de parte de los usuarios para mejorar las interfaces del sistema.

No	Pregunta	Totalmente de acuerdo	De acuerdo	En desacuerdo	Totalmente en desacuerdo
1	¿La aplicación es intuitiva?				
2	¿Considera usted que los colores crean un buen contraste?				
3	¿El diseño y estilo de la aplicación es consistente a lo largo de sus interfaces?				
4	¿El lenguaje usado en la aplicación es claro?				
5	¿La aplicación ofrece mensajes acordes a las operaciones realizadas?				
6	¿Es fácil navegar a través de la aplicación?				
7	¿La aplicación es usable?				
8	¿Las acciones realizadas son fáciles de memorizar?				
9	¿La aplicación muestra mensajes de errores explícitos de acuerdo al error ocurrido?				

Tabla 5. Encuesta de aceptación. Fuente: El Autor

Para el desarrollo de esta encuesta se deben seleccionar valores que van desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo". Los resultados de las encuestas aplicadas, para evaluar el nivel de aceptación, se pueden apreciar en las siguientes Figuras.

- Pregunta 1

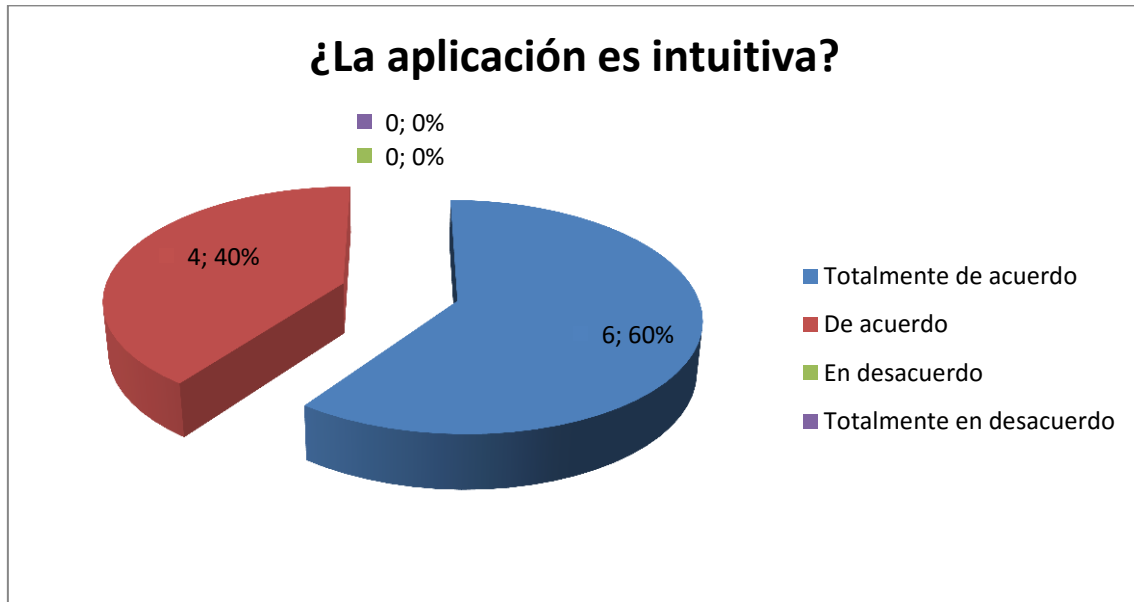


Figura 16 Resultados de la pregunta N° 1. Fuente: Pruebas de aceptación con usuarios finales.

- Pregunta 2

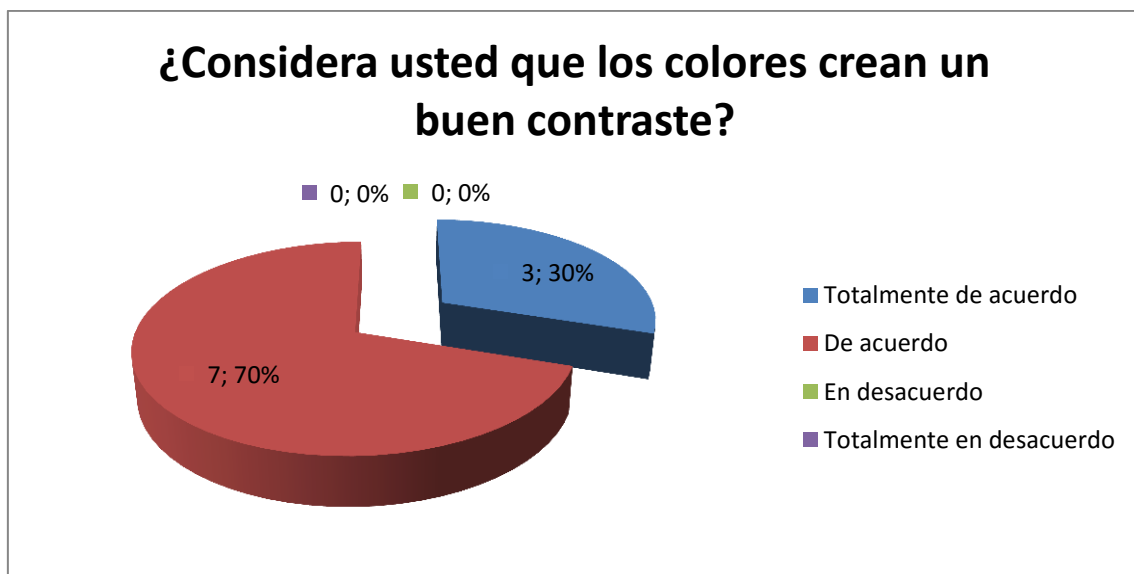


Figura 17. Resultados de la pregunta N° 2. Fuente: Pruebas de aceptación con usuarios finales

- Pregunta 3

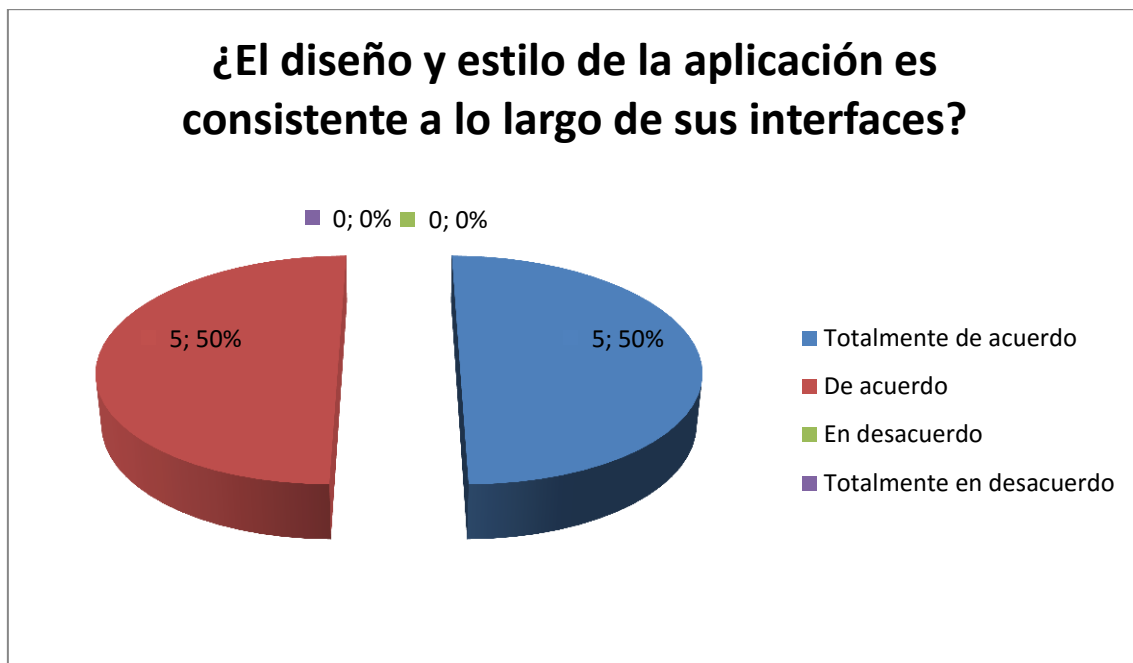


Figura 18. Resultados de la pregunta N° 3. Fuente: Pruebas de aceptación con usuarios finales.

- Pregunta 4

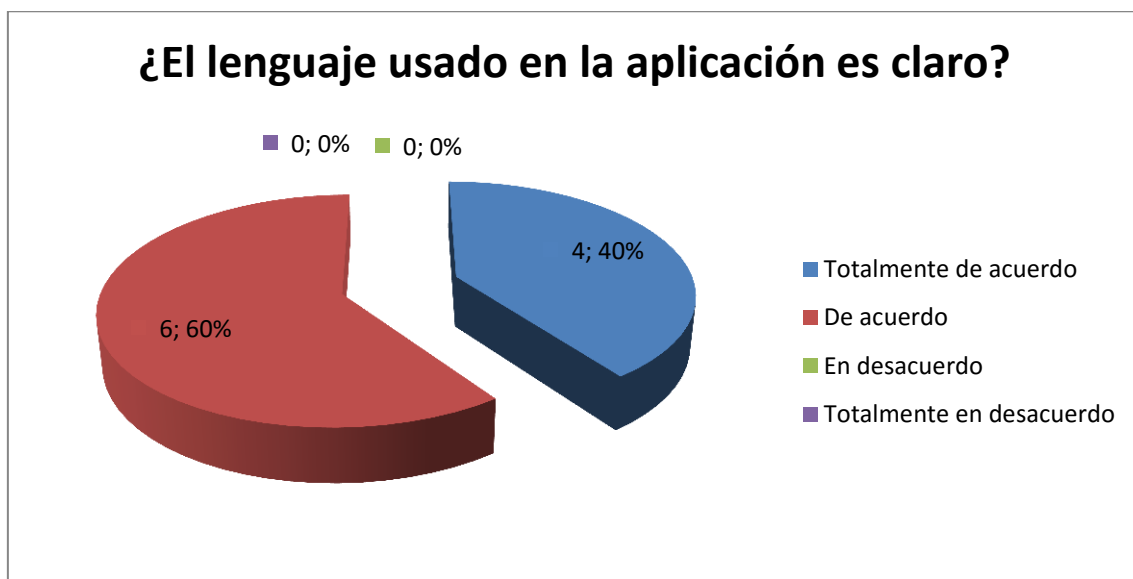


Figura 19. Resultados de la pregunta N° 4. Fuente: Pruebas de aceptación con usuarios finales.

- Pregunta 5

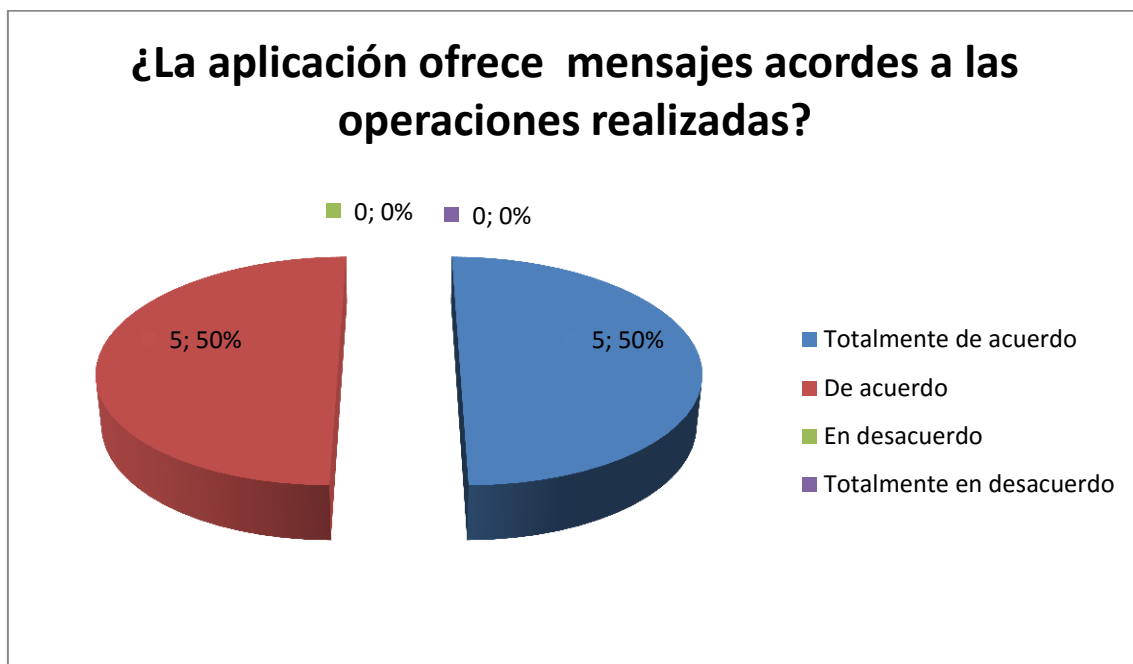


Figura 20. Resultados de la pregunta N° 5. Fuente: Pruebas de aceptación con usuarios finales.

- Pregunta 6

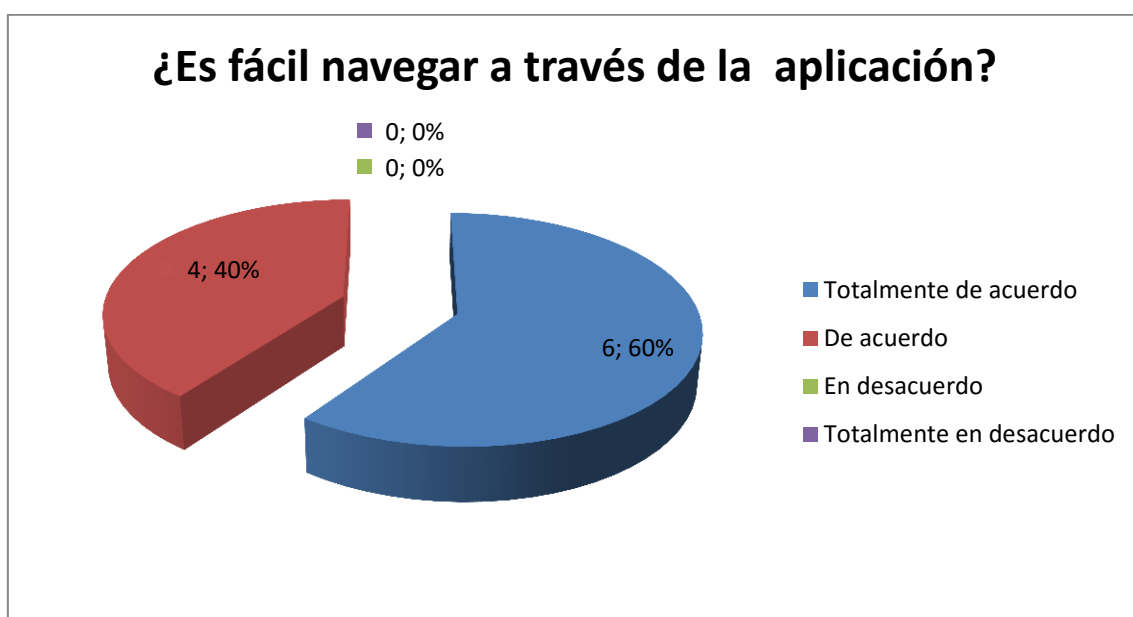


Figura 21. Resultados de la pregunta N° 6. Fuente: Pruebas de aceptación con usuarios finales.

- Pregunta 7.

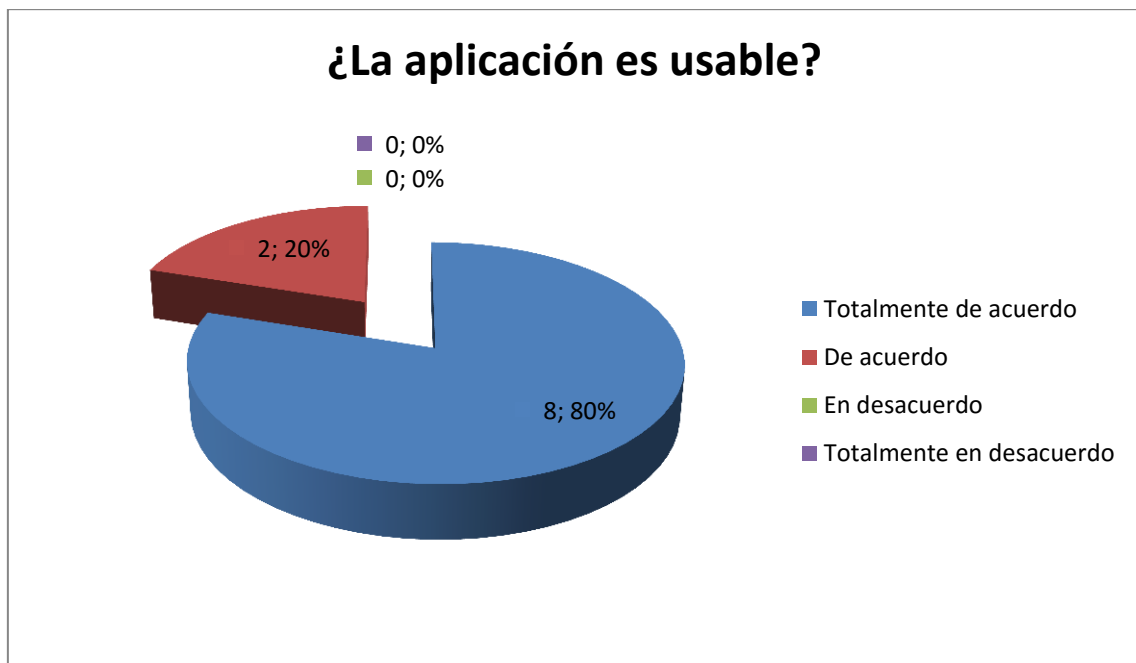


Figura 22. Resultados de la pregunta N° 7. Fuente: Pruebas de aceptación con usuarios finales

- Pregunta 8

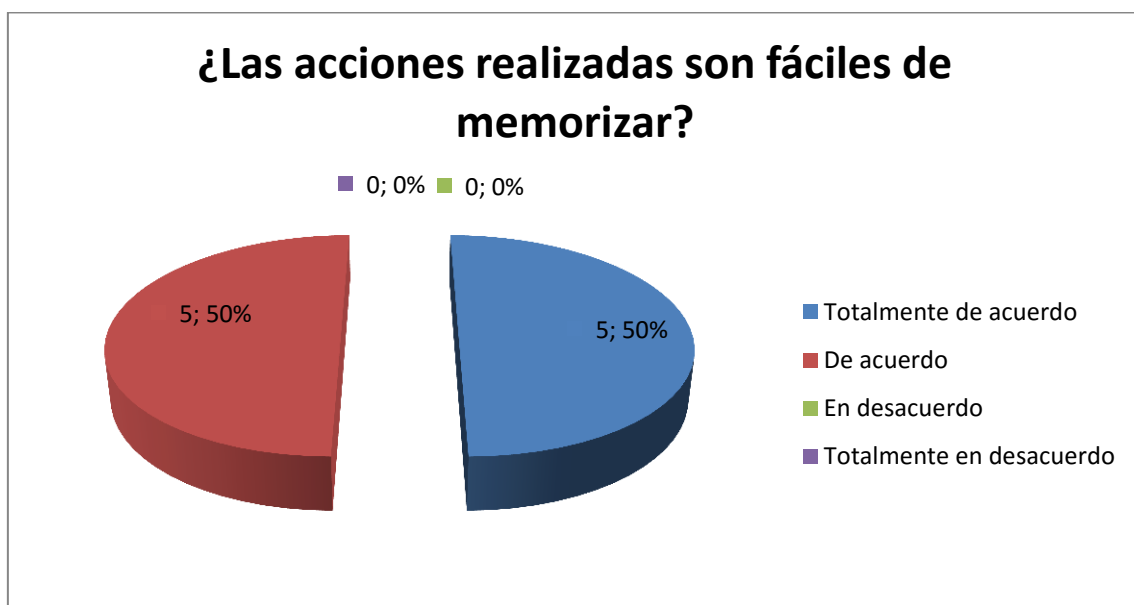


Figura 23. Resultados de la pregunta N° 8. Fuente: Pruebas de aceptación con usuarios finales.

- Pregunta 9

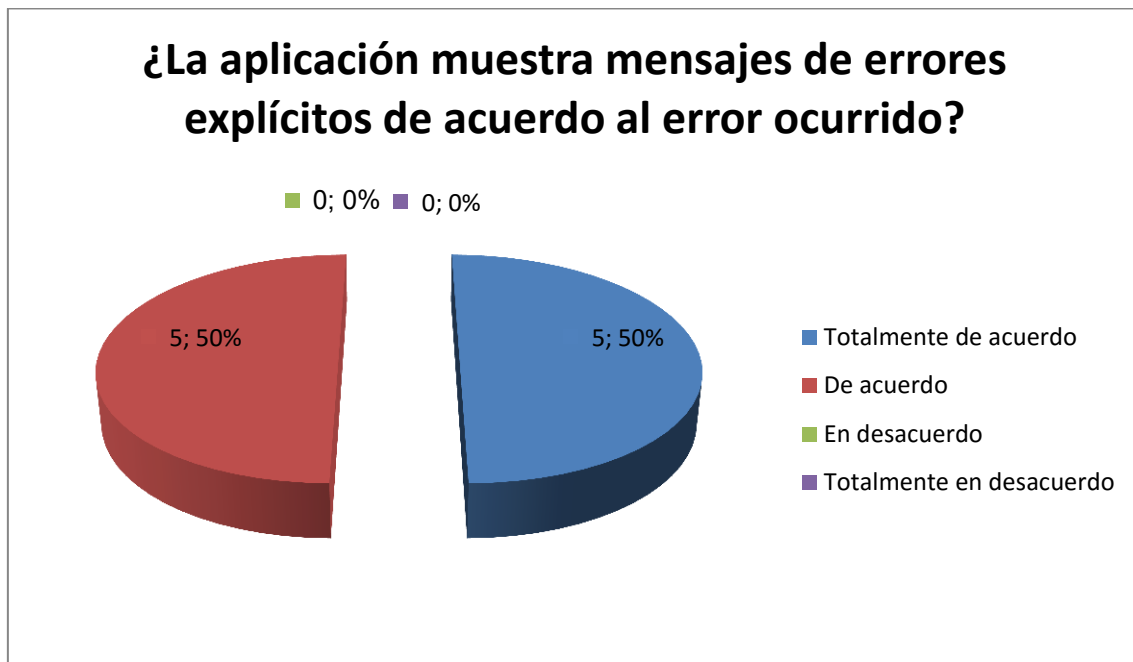


Figura 24. Resultados de la pregunta N° 9. Fuente: Pruebas de aceptación con usuarios finales.

Con base en los resultados anteriores se puede concluir que la aplicación es bastante intuitiva, utiliza colores que crean buen contraste y haciendo la aplicación consistente entre sus interfaces, logrando así una gran usabilidad. Además se puede apreciar que el lenguaje utilizado en la aplicación tanto como los mensajes de respuesta de la misma es de agrado para los usuarios aunque en parte del lenguaje de la aplicación hay ciertos tecnicismos los cuales le dan un nivel de complejidad a los mismos. Aun así por ser intuitiva, es fácil recordar las acciones realizadas.

CONCLUSIONES Y RECOMENDACIONES

A través del presente Trabajo Especial de Grado se desarrolló una aplicación móvil para la gestión de visitas clínicas de una compañía de seguros. Con el objetivo de brindar una herramienta para agilizar este proceso. Para el cumplimiento de los objetivos y requerimientos fue de vital importancia el análisis y entendimiento del proceso de visitas clínicas existente, así como del sistema encargado de llevarlo. Luego de ser planteada una solución para la agilización de este proceso, fue importante evaluar el conjunto de tecnologías a utilizar para implementar dicha solución. Para este desarrollo se utilizaron tecnologías móviles de punta, como el framework IONIC en su versión tres (3), este nos permitió desarrollar una aplicación móvil híbrida que funciona perfectamente tanto en dispositivos con sistemas operativos Android como iOS. Se utilizó el framework Python Flask que permitió la creación de una capa de web services para conectar la aplicación móvil con el sistema de visitas clínicas existentes. También se hizo uso de una base de datos móvil bajo la tecnología de SQLite, esto permitió que la aplicación trabaje perfectamente en lugares donde no se encuentre una conexión a internet disponible, sin afectar el rendimiento ni las funcionalidades de la aplicación. Fue igualmente importante utilizar una metodología de desarrollo para guiar este proceso. La solución desarrollada fue conducida bajo una adaptación de la metodología SCRUM, en la cual el propietario del producto estuvo representado por el Prof. Franky Uzcátegui y el equipo de SCRUM fue representado por el Br. Juan Vásquez. Así mismo fue de vital importancia la participación de usuarios involucrados al proceso, durante la etapa de desarrollo de la solución, la verificación de pruebas tanto funcionales como no funcionales, ya que todo esto permitió comprender mejor las necesidades y así realizar el desarrollo de una solución que cumpliera con lo deseado.

Se debe tomar en cuenta que todo sistema en un periodo de tiempo cercano, se verá en la necesidad de adaptarse a nuevos requerimientos del negocio, por esta razón se deben tomar en cuenta las siguientes consideraciones:

- Mantener actualizadas las tecnologías móviles utilizadas, en este caso el framework IONIC, esto con el objetivo de lograr compatibilidad con nuevos dispositivos y sistemas operativos que salgan al mercado en un futuro.
- Desarrollar un sistema que permita analizar los datos recopilados por el aplicativo móvil, con el objetivo de obtener información relevante que permita a

la alta directiva de las compañías aseguradoras a tomar decisiones con respecto al negocio.

- Agregar nuevas funcionalidades a la aplicación con el objetivo de expandir las herramientas que tienen los visitantes clínicos para llevar su labor.

BIBLIOGRAFÍA

[1] Laudon, K., & Laudon, J. (2012). Sistemas de información gerencial. Naucalpan de Juárez: Pearson Educación de México.

[2] Universidad Central de Venezuela, Sistemas de Información, (2012). Recuperado de: <https://www.scribd.com/doc/123464875/Introduccion-a-BPMS-ppt>

[3] Aplicación informática. (2018). Recuperada de: https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica

[4] Santiago, Raul et al. (2015). Mobile learning: nuevas realidades en el aula. Grupo Océano

[5] Métodos para el desarrollo de aplicaciones móviles - PDF. (2018). Recuperado de: <http://docplayer.es/3103691-Metodos-para-el-desarrollo-de-aplicaciones-moviles.html>

[6] Cliente-servidor. (2018). Recuperado de: <https://es.wikipedia.org/wiki/Cliente-servidor>

[7] Arquitectura Cliente Servidor - EcuRed. (2018). Recuperado de: https://www.ecured.cu/Arquitectura_Cliente_Servidor

[8] Wikipedia (s.f). Base de datos. Artículo en línea. Disponible en: http://es.wikipedia.org/wiki/Base_de_datos.

[9] Wikipedia (s.f). Base de datos movil. Artículo en línea. Recuperado de: https://es.wikipedia.org/wiki/Base_de_datos_movil

[9] Bases de datos en dispositivos móviles (2018). Recuperado de: http://jmaw.blogspot.com/2012/07/bases-de-datos-en-dispositivos-moviles_23.html

[10] SQLite with Mobile Development. (2018). Recuperado de: <https://www.experts-exchange.com/questions/29005528/SQLite-with-Mobile-Development.html>

[11] ¿Qué es y para qué sirve un web service? (2018). Recuperado de: <http://culturacion.com/que-es-y-para-que-sirve-un-web-service/>

[12] Introducción a los Web Services. (2018). Recuperado de: <https://diego.com.es/introduccion-a-los-web-services>

[13] REST. (2018). Recuperado de: https://www.ibm.com/support/knowledgecenter/es/SSMKHH_10.0.0/com.ibm.etools.mft.doc/bi12017_.htm

[14] Conceptos sobre APIs REST. (2018). Recuperado de: <http://asiermarques.com/2013/conceptos-sobre-apis-rest/>

- [15] W3C Recommendation (27 April 2007). SOAP Version 1.2 Part 0: Primer (Second Edition)
- [16] SOAP Enciclopedia Proyecto AjpdSoft. (2018). Recuperado de: <http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=1134>
- [17] Sistema operativo móvil. (2018). Recuperado de: https://es.wikipedia.org/wiki/Sistema_operativo_m%C3%B3vil
- [18] Definista. (2018). ¿Qué es iOS? - Su Definición, Concepto y Significado. Recuperado de: <http://conceptodefinition.de/ios/>
- [19] Android - EcuRed. (2018). Recuperado de: <https://www.ecured.cu/android>
- [20] Windows Phone - EcuRed. (2018). Recuperado de: https://www.ecured.cu/Windows_Phone
- [21] ¿Qué es una app nativa? ¿Qué ventajas tiene? (2018). Recuperado de: <http://elixir.es/que-es-una-app-nativa/>
- [22] Aplicación móvil híbrida (2018). Recuperado de: <https://www.solbyte.com/blog/2014/07/21/tipos-de-aplicaciones-moviles-nativas-webs-hibridas/biblio>
- [23] ¿Qué es Java y para qué es necesario? (2018). Recuperado de: https://www.java.com/es/download/faq/whatis_java.xml
- [24] Objective-C - EcuRed. (2018). Recuperado de: <https://www.ecured.cu/Objective-C>
- [25] Swift - Apple (ES). (2018). Recuperado de: <https://www.apple.com/es/swift/>
- [26] Introducción al lenguaje C# y .NET Framework. (2018). Recuperado de: [https://msdn.microsoft.com/es-es/library/z1zx9t92\(VS.80\).aspx](https://msdn.microsoft.com/es-es/library/z1zx9t92(VS.80).aspx)
- [27] APLICACIONES HÍBRIDAS: ¿QUÉ SON Y CÓMO USARLAS? (2018). Recuperado de: <https://www.nextu.com/blog/aplicaciones-hibridas-que-son-y-como-usarlas/>
- [28] Definición, u. (2018). Definición, usos y ventajas del lenguaje HTML5. Recuperado de: <http://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-html5/>
- [29] Definición de CSS — Definicion.de. (2018). Recuperado de: <https://definicion.de/css/>
- [30] Fundamentos de JavaScript. (2018). Recuperado de: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- [31] Framework. (2018). Recuperado de: <https://es.wikipedia.org/wiki/Framework>

- [32] Vázquez, G. (2018). ¿Qué es Angular? Cómo aprender Angular desde Cero - Código OnClick. Recuperado de: <https://codigoonclick.com/que-es-angular/>
- [34] Rojas, C. (2018). ¿Que es Ionic Cloud Services ? Recuperado de: <https://blog.ng-classroom.com/blog/tips/ionic-cloud-services/>
- [35] Qué es y cómo empezar con Ionic Framework | PhoneGap Spain. (2018). Recuperado de: <http://www.phonegapspain.com/que-es-y-como-empezar-con-ionic-framework/>
- [36] React. (2018). Recuperado de: <https://es.wikipedia.org/wiki/React>
- [37] React-Native está cambiando la forma en la que desarrollamos aplicaciones móviles. (2018). Recuperado de: <https://medium.com/monoku/react-native-est%C3%A1-cambiando-la-forma-en-la-que-desarrollamos-aplicaciones-m%C3%B3viles-de7bbb742555>
- [38] Qué es SCRUM. (2018). Recuperado de: <https://proyectosagiles.org/que-es-scrum/>
- [39] Definición de Aseguradora. (2018). Recuperado de: <https://www.definicionabc.com/derecho/aseguradora.php>
- [40] Asegurado. (2018). Recuperado de: <https://es.wikipedia.org/wiki/Asegurado>
- [41] Seguro, S. (2018). Significado de Póliza de seguro. Recuperado de: <https://www.significados.com/poliza-de-seguro/>
- [42] Staff, f. (2018). Seguro médico: entendiendo lo que éste cubre - familydoctor.org. Recuperado de: <https://es.familydoctor.org/seguro-medico-entendiendo-lo-que-este-cubre/>
- [43] Seguro de salud (2018). Recuperado de: <https://www.geosalud.com/seguromedico/segurosalud/procesos.html>
- [44] Base de datos relacional en línea en: https://www.ibm.com/support/knowledgecenter/es/SSEPGG_8.2.0/com.ibm.db2.udb.doc/admin/c0004099.htm
- [45] Bases de datos no relacionales | Bases de datos de gráficos | AWS. (2018). Recuperado de: <https://aws.amazon.com/es/nosql/>
- [46] Couchbase-lite Five popular databases for mobile. (2018). Recuperado de: <https://www.developereconomics.com/five-popular-databases-for-mobile>
- [47] Berkeley DB Five popular databases for mobile. (2018). Recuperado de: <https://www.developereconomics.com/five-popular-databases-for-mobile>
- [48] Nielsen, J. (2003). Usability 101: Introduction to Usability. Recuperado de: <http://www.useit.com/alertbox/20030825.html>