



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

APLICACIÓN MÓVIL PARA EL
SEGUIMIENTO DE PLANES DE CUIDADO
DE ENFERMERÍA

TRABAJO DE TESIS
PARA OBTENER EL GRADO DE
INGENIERO EN REDES

PRESENTA
JESÚS ALFREDO CAMPOS CARRILLO

DIRECTORA DE TESIS
M.T.I. MELISSA BLANQUETO ESTRADA

ASESORES

DR. JAIME SILVERIO ORTEGÓN AGUILAR

M.C.E ESMERALDA FUENTES FERNÁNDEZ

M.T.I. VLADIMIR VENIAMIN CABAÑAS VICTORIA

M.S.I. LAURA YÉSICA DÁVALOS CASTILLA





UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

TRABAJO DE TESIS TITULADO

APLICACIÓN MÓVIL PARA EL SEGUIMIENTO DE
PLANES DE CUIDADO DE ENFERMERÍA

ELABORADO POR

JESÚS ALFREDO CAMPOS CARRILLO

BAJO SUPERVISIÓN DEL COMITÉ DE ASESORÍA Y APROBADO COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE
INGENIERO EN REDES

COMITÉ DE TESIS

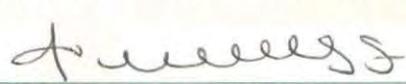
DIRECTORA:


M.T.I. MELISSA BLANQUETO ESTRADA

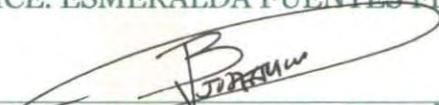
ASESOR:


DR. JAIME SILVERIO ORTEGÓN AGUILAR

ASESORA:


MCE. ESMERALDA FUENTES FERNÁNDEZ

ASESOR:


MTI. VLADIMIR VENJAMÍN CABAÑAS VICTORIA

ASESORA:


MSI. LAURA YÉSICA DÁVALOS CASTILLÓN



Resumen

A través de este trabajo se describe el proceso de desarrollo de una aplicación para el cuidado de pacientes de enfermería que propone un estándar para la evaluación y seguimiento a pacientes bajo la norma del catálogo de Planes de Cuidado de Enfermería (PLACE) emitido por la Secretaría de Salud del cual se sintetizaron algunos planes de cuidado para incluir en un formato que la aplicación móvil permitiera mostrar al usuario.

La aplicación fue resultado de un trabajo en conjunto por parte de la División de Ciencias de la Salud (DCS) y la División de Ciencias e Ingeniería (DCI), que permitiera su utilización en centros médicos hospitalarios con conexión a Internet, pero también caracterizándose por su capacidad de trabajar en escenarios donde no se posee una cobertura telefónica o con acceso a Internet por lo cual se considera como una aplicación online/offline.

En la fase de pruebas participaron un grupo de estudiantes de enfermería, una experta en el área de enfermería de la DCS y personal del Hospital General de Chetumal, logrando un nivel de aceptación bueno gracias a su interfaz intuitiva y sencilla para el usuario, ya que se usan interfaces con las que los usuarios de enfermería están familiarizados.

Agradecimientos

Esta tesis no sería posible sin el apoyo de todas las personas que me han acompañado desde el inicio de mi formación universitaria, entre los que se encuentran profesores, compañeros y amigos. Gracias por tantos momentos, experiencias y situaciones que me han permitido culminar esta etapa de mi vida.

Dedicatoria

No se llega hasta este punto sin la compañía y amor que encuentras en tu familia. Son el principal motor e inspiración para realizar este trabajo, que los haga sentir orgullosos para devolverles un poco de lo que he recibido para llegar hasta donde estoy. En especial a mi querida abuela Griselda, a quien le debo todo lo que he logrado y a quien estaré eternamente agradecido por sus enseñanzas y el amor que me brindó desde que era un niño.

Le agradezco a Dios por haber puesto en mi camino a personas maravillosas, como mis padres, tíos y familiares que estuvieron a mi lado.

Contenido

Resumen.....	i
Agradecimientos.....	ii
Dedicatoria	iii
Índice de figuras	vi
Índice de tablas	vii
Capítulo 1 Introducción.....	8
1.1 Antecedentes	8
Aplicaciones e-Salud	9
1.4 Objetivos particulares.....	11
1.4 Alcance	11
Capítulo 2 Marco Teórico	13
2.1 La tecnología móvil	13
2.1.1 Desarrollo de aplicaciones híbridas	13
2.2 Metodología para el desarrollo de aplicaciones móviles.....	18
2.2.1 Metodología XP (Extreme Programming)	18
Capítulo 3 DESARROLLO.....	19
3.1 Fase de planificación.....	19
3.1.1 Descripción de los interesados	19
3.1.2 Historias de usuario	19
3.1.4. Restricciones.....	20
3.2 Fase de diseño.....	21
3.2.1. Planes de enfermería sintetizados.....	21
3.2.2. Diagrama arquitectónico.....	23
3.2.3. Diagrama de navegación.....	24

3.2.4. Diseño de las interfaces gráficas	25
3.3 Fase de Codificación	31
3.3.1. Desarrollo del módulo de sincronización de información.	31
3.3.2. Desarrollo de las interfaces de usuario	31
3.4 Fase de Pruebas	34
3.4.1. Elaboración del plan de pruebas	34
3.4.2. Ejecución de las pruebas con usuarios.....	35
3.4.3. Análisis de los resultados	36
3.5 Documentación	39
3.5.1. Manual del usuario	39
Capítulo 4 CONCLUSIONES.....	52
REFERENCIAS.....	53
ANEXOS	55
Anexo 1. Configuración del entorno de desarrollo.....	55

Índice de figuras

Figura 1. Metodología para el desarrollo de la app. Elaboración propia.	18
Figura 2. Esquema de la arquitectura de la aplicación. Elaboración propia.	23
Figura 3. Flujo de navegación de la app PLACE	24
Figura 4. Pagina para agregar un nuevo paciente	25
Figura 5. Diagnóstico.....	26
Figura 6. Inicio de Sesión	27
Figura 7. Lista de pacientes.....	28
Figura 8. Lista de planes	29
Figura 9. Página de resultados	30
Figura 13. Vista de registro del usuario.	39
Figura 14. Correo de confirmación de cuenta	40
Figura 15. Página de Login	41
Figura 16. Página de Inicio	42
Figura 17. Página para agregar un nuevo paciente	43
Figura 18. Página de inicio con un nuevo paciente agregado.....	44
Figura 19. Menú por paciente	45
Figura 20. Menú de planes disponibles	46
Figura 21. Página para realizar evaluación de paciente.	48
Figura 22. Actividades disponibles.	49
Figura 23. Valor asignado a cada escala.....	50
Figura 24. Historial de evaluaciones por usuario.	51

Índice de tablas

Tabla 1. Ejemplos de historias de usuario	19
---	----

Capítulo 1 Introducción

1.1 Antecedentes

El desarrollo de aplicaciones móviles es el conjunto de procesos y procedimientos involucrados en la escritura de software para pequeños dispositivos inalámbricos de cómputo, como teléfonos inteligentes o tabletas.

El desarrollo de aplicaciones móviles es similar al desarrollo de aplicaciones web, y tiene sus raíces en el desarrollo de software más tradicional. Una diferencia fundamental, sin embargo, es que las aplicaciones móviles (apps) a menudo se escriben específicamente para aprovechar las características únicas que ofrece un dispositivo móvil en particular. Por ejemplo, una aplicación para juegos podría escribirse para aprovechar el acelerómetro del iPhone (Rouse, 2017).

Con la aparición del iPhone en el evento MacWorld del 2007, surgen las aplicaciones móviles tal y como las conocemos actualmente. Estas primeras aplicaciones, inicialmente lanzadas para el teléfono inteligente iPhone fueron desarrolladas bajo el lenguaje Objective-C Swift por el propio equipo de desarrollo de Apple y no se admitían aplicaciones desarrolladas por terceros, por lo cual la única alternativa para desarrollar tus propias “apps” para este dispositivo era creando un sitio web de tal manera que se adaptara a este dispositivo para posteriormente ser consultado a través del navegador Safari para IOS que venía instalado de fábrica con el teléfono.

Este hecho significó un gran avance en el desarrollo móvil en dos aspectos, el primero era la propuesta de eliminar completamente el enfoque del hardware y pensar exclusivamente en el software en la etapa de desarrollo, ya que el iPhone carecía de botones y presentaba una interfaz uniforme para todos los modelos del dispositivo lo que le quitaba complejidad al desarrollo de software, ahora los programadores ya no tendrían que preocuparse por la incompatibilidad en el hardware de los dispositivos móviles y enfocar gran parte de su trabajo exclusivamente en el software; el segundo aspecto fue que debido a la negativa de Steve Jobs por liberar un SDK para el iPhone disponible para cualquiera, se produjo un gran interés en la versión 5 de HTML, cuya versión era completamente nueva en esa época, por lo que

terminó impulsando el uso de esta nueva versión que hoy en día es la base para cualquier aplicación nativa, además, muchas personas empezaron a pensar en la idea de visualizar sitios web desde un móvil compartiendo el mismo código sin importar desde donde se está consultando, un dispositivo móvil o una PC, debido a que el iPhone a través de su navegador seguía el mismo estándar para acceder a sitios Web a diferencia de la mayoría de los dispositivos de la época que utilizaban tecnologías obsoletas como WAP para navegar por internet, esto debido a las grandes capacidades del iPhone en comparación con los teléfonos de aquella época.

Aplicaciones e-Salud

Actualmente en el mercado existen las siguientes aplicaciones móviles enfocadas al área de enfermería, describimos algunas a continuación:

Enfermería Med IV

Una de las aplicaciones con las que cuenta de más prestigio es Enfermería Med IV (Google, 2020), aplicación hecha por Nieves Robledo Casado y Pablo Robledo Casado, con base en la descripción disponible en su ficha de Play Store, nos permite consultar información destacando la correcta manera de inyectar vacunas o métodos de extracción de sangre, además de ofrecer una calculadora para saber la cantidad a inyectar de una sustancia(suero) o comparaciones de índice de masa corporal (IMC). La aplicación cuenta actualmente con más de 100 mil instalaciones y aproximadamente 1700 reseñas con una valoración de 4.4 de 5 puntos como máximo.

Enfermería Blog

Otra app interesante de enfermería muy popular es Enfermería Blog centrada en compartir historias y experiencia entre enfermeros y estudiantes, permite consultar información como la historia de la enfermería, los diferentes grados que existen en esta rama, así como guías y procedimientos.

Esta aplicación tiene más de 100, 000 descargas y posee un porcentaje muy aceptable de aprobación, con una valoración de 4 puntos en la escala de 1 a 5 (Google, 2020).

Las apps de enfermería no se limitan a un pequeño grupo de potenciales usuarios y dependiendo de su nivel de funcionalidad se puede llegar tener muchas descargas, por lo

que se debe de presentar un proyecto lo suficientemente robusto, tanto en análisis como en desarrollo para que la app pueda llegar a un público muy amplio y ser descargada por muchos usuarios en diversos centros hospitalarios.

1.2 Justificación

El desarrollo tecnológico e informático tiene una influencia en los sistemas de salud de los países desarrollados; con el uso de tecnología como soporte a la enfermería (Reza & Saneipour, 2010). Por lo tanto, la enfermería está obligada a actualizarse en sus conocimientos y metodologías de trabajo para continuar su función de proveedora de cuidados.

En este sentido, el proyecto propuesto fue el desarrollo de una aplicación móvil para la utilización y el seguimiento de un Plan de Cuidados de Enfermería que forme parte del Catálogo Nacional en México, asimismo al crear una arquitectura móvil se logró un mecanismo para que estos planes puedan actualizarse y difundirse de forma ágil entre los profesionales de la enfermería y también sirva como un entrenamiento para la simulación de casos en enfermería.

El aprendizaje a través de dispositivos móviles se ha integrado a la enseñanza de diversas disciplinas. En este caso se podrán simular escenarios para capacitar a estudiantes de la Licenciatura en Enfermería de la Universidad de Quintana Roo, considerando un enfoque humanístico, científico y tecnológico. También se aborda la problemática de salud pública, que busca conformar un sistema integral de salud de alta calidad para toda la población; en ese sentido, la aplicación móvil sería una posibilidad de actualización de los profesionales de la enfermería quienes son responsables de la atención de pacientes en ambientes reales y deben seguir procedimientos estándar de atención a pacientes. A través de la aplicación móvil será posible descargar Planes de Cuidado de Enfermería actualizados, sintetizando el contenido de un plan para facilitar su aplicación en el seguimiento de los pacientes. En este proyecto convergen dos disciplinas: las Ciencias de la Salud, que buscan mejorar la calidad de la atención a través del uso de Planes de Cuidado de Enfermería, y la Ingeniería, para dar respuesta a una problemática que involucra el uso de tecnología para desarrollo de software.

1.3 Objetivo general

Desarrollar una aplicación móvil para facilitar la utilización y seguimiento de un plan del Catálogo Nacional de Planes de Cuidados de Enfermería que permita mejorar la capacitación profesional del personal de salud en el uso de una herramienta metodológica para el cuidado integral de un paciente.

1.4 Objetivos particulares

- Analizar los requerimientos técnicos y funcionales para el desarrollo de una aplicación móvil que gestione un plan del Catálogo Nacional de Planes de Cuidados de Enfermería por parte del personal de salud.
- Diseñar una arquitectura de aplicación móvil que facilite la incorporación de cualquiera de los planes incluidos en el Catálogo Nacional de Planes de Cuidado de Enfermería.
- Diseñar la estructura de la base de datos local y remota de la aplicación móvil.
- Construir las interfaces gráficas de la aplicación móvil.
- Realizar pruebas de la funcionalidad de las interfaces gráficas de la aplicación móvil.
- Evaluar la usabilidad de la aplicación móvil.

1.4 Alcance

El proyecto tuvo como objetivo crear una aplicación móvil que guíe a los usuarios a evaluar a sus pacientes y posteriormente dar un seguimiento a esas evaluaciones con base en el catálogo nacional de planes médicos emitido por la Secretaría de Salud, estos planes estarán disponibles para el usuario a través de la app, pero para poder visualizarlos y descargarlos el usuario deberá contar con una conexión a internet. Solo se podrán utilizar planes de cuidado de enfermería que previamente se hayan descargado.

Al actualizar un plan de cuidados en la base de datos alojada en la nube, se replicarán los cambios automáticamente en todos los dispositivos que hayan descargado este plan sin que

el usuario tenga que volver a seleccionar el plan en la aplicación, es decir, que permite un funcionamiento en segundo plano.

La aplicación debe ser intuitiva, con funcionamiento accesible al usuario, y que contenga información para solucionar dudas en cuanto al uso de la app.

Alternativo a la app, se debe crear y configurar una base de datos en la nube a la cual se conectarán todos los dispositivos que tengan la app instalada para descargar los planes que el usuario elija. Por el momento la aplicación se distribuirá de forma directa al compartir la aplicación por mensajería e instalarla, debido a que se pretende lanzar en una plataforma de distribución cuando se haya formado una base de usuarios que retroalimenten esta primera versión.

Capítulo 2 Marco Teórico

2.1 La tecnología móvil

2.1.1 Desarrollo de aplicaciones híbridas

Una aplicación híbrida es una aplicación que se desarrolla bajo el conjunto de herramientas del estándar web HTML, CSS Y JavaScript y otros marcos de trabajo (frameworks) que no presentan diferencias en cuanto al aspecto o diseño en comparación con las aplicaciones nativas (aplicación que se desarrolla en el lenguaje de programación nativo del sistema operativo del dispositivo móvil) y además puede acceder a varios componentes de un teléfono como son:

- Cámara
- GPS
- Bluetooth
- Wifi

Estas apps se pueden ofrecer en las tiendas de aplicaciones como Play Store o App Store y se instalan en los dispositivos móviles de la misma forma que las aplicaciones nativas.

Para construir una aplicación móvil híbrida, los desarrolladores escriben el núcleo de la aplicación como una aplicación móvil HTML5 y luego colocar un empaquetador (wrapper) de dispositivo nativo alrededor. El empaquetador nativo actúa como intermediario y traduce las instrucciones a una forma que el dispositivo móvil entiende. El núcleo HTML5 de la aplicación móvil híbrida funciona dentro del contenedor nativo y utiliza el motor del navegador del dispositivo móvil (no el propio navegador) para cargar el HTML5 y procesar el código JavaScript a nivel local. El núcleo HTML 5 de la aplicación puede ser reutilizado para otras plataformas móviles. Solo el empaquetador nativo debe ser escrito para cada dispositivo móvil y sistema operativo. Los desarrolladores pueden reducir el tiempo de desarrollo generando empaquetadores con software de proveedores como PhoneGap y Apache Cordova.

IONIC

Ionic es un framework que nos permite desarrollar aplicaciones híbridas siguiendo los estándares de desarrollo Web (HTML, CSS Y JavaScript)

En palabras de sus creadores, el núcleo de Ionic es:

“Ionic facilita el desarrollo de aplicaciones con alto nivel de rendimiento y Aplicaciones web progresivas (PWAs por sus siglas en inglés) que en cualquier plataforma o dispositivo” (Ionic , 2019).

Un tema que le da popularidad a este framework es que al utilizar tecnologías web facilita la curva de aprendizaje y es más fácil de aprender en comparación con los lenguajes nativos de cada Plataforma, Java para Android y Objective-C para IOS. De igual forma Ionic es atractivo para los desarrolladores web ya que pueden aprovechar el conocimiento que tienen y aplicarlo al momento de estar programando ya que la base del desarrollo es la gama de tecnologías HTML-CSS-JS en lugar de aprender un nuevo lenguaje

Una aplicación hecha en Ionic se puede descargar desde las tiendas oficiales de cada plataforma (App Store y Play Store) al igual que las aplicaciones nativas y el usuario final de la aplicación no notará la diferencia al momento de estar utilizando una aplicación híbrida o nativa ya que una app hecha con Ionic no presenta desventajas visuales de sus elementos además de que Ionic puede alcanzar un buen nivel de rendimiento en la mayoría de los casos al igual que las apps hechas con un lenguaje nativo.

Ionic proporciona etiquetas que se incrustan en el código de archivos HTML tales como `<p>` o `<h1>`. Para la versión 3 utiliza sistema de rutas para controlar el flujo entre cada vista, permitiendo compartir objetos de una vista a otra. Estos objetos se utilizan para crear la estructura de la vista que se presenta dentro de una página de la aplicación, estas etiquetas son componentes visuales que solo se pueden utilizar en Ionic y pueden combinarse junto a las etiquetas HTML tradicionales. Al utilizar etiquetas de Ionic en los documentos HTML se cargan una serie de estilos que se incluyen al instalar la plantilla que ocupa la aplicación, por lo cual basta con incluir algo tan simple como: `<ion-range> </ion-range>` para que Ionic incorpore el componente a la vista.

Ionic nos proporciona las interfaces necesarias para mostrar y estructurar los documentos con elementos con diseño visual incorporado para el usuario, sin embargo, en muchos casos esto no es suficiente para crear una herramienta funcional, ya que también se requiere que un usuario pueda usar otras características del teléfono como tomar una fotografía o saber la ubicación actual. Para esto Ionic incorpora dentro de su framework al framework Cordova, que se encarga de crear todo el canal de comunicación con el hardware del teléfono dentro del mismo entorno de trabajo.

Cordova

Cordova es un framework que proporciona una serie de bibliotecas de código abierto que permiten la comunicación entre esta interfaz común y las distintas funciones nativas del dispositivo” (Indigitall, 2018).

Cordova se puede incorporar dentro de Ionic para interactuar con el hardware del dispositivo y acceder a elementos tales como: cámara, bluetooth, WiFi o GPS. Cordova nos proporciona código para que podamos programar el funcionamiento de cada uno de estos elementos a través de plugins que se instalan desde la consola de Visual Studio Code utilizando el manejador de paquetes de NodeJS(node). Por ejemplo, se podría indicar que el WiFi del teléfono se encienda en el teléfono cuando se carga la página de inicio de la aplicación, y debido a que esta información ya viene empaquetada al momento de instalar el plugin no se necesita interactuar con código nativo de la plataforma(Android o IOS), además este mismo plugin se puede utilizar para cualquier SO móvil disponible por lo que no es necesario instalar el plugin dependiendo de las plataformas para la que se esté desarrollando.

Angular JS

Angular es un framework que se incorpora junto con Ionic y Cordova en un mismo entorno de programación y que basa su lenguaje de programación en TypeScript, el cual es un supercomponente de JavaScript. Si se utiliza HTML para construir las vistas, entonces TypeScript es el lenguaje en el que se incorpora toda la lógica de desarrollo de la aplicación, tal como manejar eventos al hacer clic a un botón o para controlar el flujo de la aplicación entre ventanas. Por si solo TypeScript no ofrece todas estas características, pero Angular si,

ya que incorpora módulos y librerías para lograr un intercambio de información entre la vista y el controlador de un componente o controlar el flujo de comunicación en las llamadas HTTP hacia una API externa de un servidor alojado en IIS o Apache. Toda página que se crea en Ionic asimilará el esquema por componentes que se utiliza en Angular, es decir, Angular agrupa la vista generada en HTML, la página de diseños en CSS y la lógica de negocio codificado en TypeScript dentro de una misma carpeta o a lo cual se refiere como un componente.

Una de las definiciones más acertadas de Angular es la siguiente:

“Angular es un framework de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitarnos el desarrollo de aplicaciones web SPA y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y optima.

Otro propósito que tiene Angular es la separación completa entre el front-end y el back-end en una aplicación web.” (Robles, 2017)

Ionic utiliza el esquema desarrollado por Angular para organizar las vistas y controladores en carpetas. Se creará un componente por cada página que se genere, y este componente a su vez se integra por un documento HTML, un documento CSS y un archivo TypeScript que terminan agrupándose para generar una vista.

De igual cuenta, Angular le proporciona a Ionic los módulos necesarios para cargar todo el contenido del desarrollo cuando se abre la aplicación y determinar las acciones del usuario como, por ejemplo, si se encuentra con una sesión activa en la aplicación o la página que se debe mostrar cuando se inicializa la aplicación.

PouchDB

PouchDB es una base de datos de código abierto inspirada en Apache CouchDB diseñada para trabajar dentro del navegador. PouchDB fue creada para ayudar a los programadores a desarrollar aplicaciones que funcionen tanto en línea como desconectada. (PouchDB, 2019)

Es un motor de base de datos que nos permite almacenar la información que se genera en la aplicación y se almacena de manera interna en la memoria del dispositivo con un esquema orientado a documentos en formato JSON, similar a CouchDB.

La principal ventaja de utilizar PouchDB en comparación con otras bases de datos como SQLite (el lenguaje SQL equivalente para los móviles) es que incorpora un conjunto de herramientas para facilitar la conexión con CouchDB. Por medio de la API que incorpora PouchDB se puede implementar de manera rápida y fácil el módulo necesario para descargar los documentos de CouchDB e incorporarlos al teléfono sin tener que utilizar JavaScript para controlar las conexiones a través de HTTP.

Se puede pensar en PouchDB como una versión de CouchDB para trabajar en dispositivos móviles, pero hay que recalcar que terminan siendo tecnologías diferentes que se adaptan para trabajar bajo el mismo enfoque, guardar la información en formatos JSON y sincronizar los cambios que se producen en CouchDB para guardarlos en la memoria del teléfono gracias a PouchDB.

CouchDB

CouchDB es una base de datos NoSQL que nos permite almacenar información y registros en documentos JSON para que puedan ser utilizados a través de JavaScript, o en el caso de nuestra a través de TypeScript

De acuerdo con Leo Pryzbylsk y Warner Onstine CouchDB es una base de datos orientada a documentos donde cada propiedad de un documento es almacenada en la relación campo-valor.

Una de las características más potentes de CouchDB es la replicación de información que se encuentran almacenados en cualquiera de sus bases de datos hacia los dispositivos desde donde se está consultando, de una manera fácil, pero a la vez robusta y que ofrece la posibilidad de replicar toda la información en un sentido bidireccional, es decir, desde CouchDB hacia el dispositivo y desde el dispositivo hacia CouchDB.

2.2 Metodología para el desarrollo de aplicaciones móviles

2.2.1 Metodología XP (Extreme Programming)

La metodología empleada en el desarrollo del prototipo de la app es la metodología ágil para el desarrollo de software denominada Programación Extrema (XP) que busca un desarrollo dinámico e incremental de las funcionales de una aplicación, con una comunicación y retroalimentación constante con el usuario.

Una de las principales ventajas al utilizar esta metodología es potenciar las relaciones entre el equipo de desarrollo mediante trabajo en conjunto, reuniones continuas y garantizando un ambiente sano entre todas las partes involucradas en el proyecto, siendo utilizada cuando los requisitos del proyecto pueden ir cambiando debido a diversas razones. (Calvo, 2018)

Para esta aplicación se realizaron diversas entrevistas a una enfermera que fungió como experta para obtener las historias de usuario, priorizar el desarrollo incremental y posteriormente realizar las pruebas preliminares de la app. La Figura 1 presenta la metodología que se siguió para este proyecto.



Figura 1. Metodología para el desarrollo de la app. Elaboración propia.

Las historias de usuario que se desarrollaron fueron: descargar plan de manera remota, agregar paciente, evaluar paciente, seleccionar plan a evaluar, historial de evaluación, lista de pacientes. Estas historias de usuario se fueron priorizando para desarrollar de forma incremental la aplicación móvil.

Capítulo 3 DESARROLLO

3.1 Fase de planificación

3.1.1 Descripción de los interesados

Los interesados en esta app pueden ser profesores y estudiantes de la Licenciatura en Enfermería, así como profesionistas de Enfermería con conocimientos en las taxonomías de enfermería NANDA, NIC y NOC.

Para esta tesis se realizaron entrevistas con una profesora que imparte clases en la División de Ciencias de la Salud de la UQROO, Unidad Académica Chetumal, quien fungió como usuario experto, debido a que también se desempeña profesionalmente en el Hospital General de Chetumal.

3.1.2 Historias de usuario

Las historias de usuario se redactaron a partir de las entrevistas con el usuario experto, quien describió cómo debería ser la interacción y a partir de esas descripciones obtener las funcionalidades de la aplicación. Ver Tabla 1.

Tabla 1. Ejemplos de historias de usuario

Historia de usuario	Descripción y validación
<i>Agregar paciente</i>	<p>--Descripción: El usuario tendrá disponible una página para que registre cada paciente al que le dará seguimiento a través de su plan de cuidados previamente descargado.</p> <p>--Validación: El usuario podrá ver todos los pacientes que haya agregado en la página principal de la aplicación.</p>
<i>Evaluar paciente</i>	<p>--Descripción: El usuario seleccionará el indicador y resultado de un plan e indicará una escala del estado del paciente para guardar la evaluación.</p> <p>--Validación: No se podrá registrar una evaluación si no se ha seleccionado un valor para indicador, resultado y escala.</p>

<p><i>Historial de evaluación</i></p>	<p>--Descripción: En esta sección el usuario podrá ver la evaluación más reciente de un paciente ordenado por indicador y resultado.</p> <p>--Validación: No debe de haber evaluaciones en donde se repita el indicador y el resultado al mismo tiempo. Deberá de aparecer la última evaluación registrada para este criterio.</p>
---------------------------------------	--

3.1.4. Restricciones

- La aplicación debe ejecutarse en dispositivos móviles Android, a partir de la versión 6.0.
- La replicación de datos sólo ocurrirá desde la nube hacia los dispositivos.
- Sintetizar los planes de cuidados con el formato estándar (JSON) proveniente del catálogo de planes de cuidado de enfermería (PLACE).

3.2 Fase de diseño

3.2.1. Planes de enfermería sintetizados.

La primera actividad consistió en sintetizar la información proveniente de los planes de enfermería que se encuentran en el documento PDF emitido por la Secretaría de Salud. Para esto, se recurrió al apoyo de una experta en el área de Enfermería, seleccionó la información más importante de cada plan que debía incluirse en la aplicación, de esta forma se acordó incluir información, tal como: nombre, descripción, etiquetas, resultados e indicadores. La información que se almacena en CouchDB es una colección de documentos en formato JSON, por lo cual cada plan emitido se representaría en un documento, que contiene elementos tales como cadenas de caracteres, colecciones de elementos y a la vez objetos.

En la siguiente página se muestra un segmento del *Plan de Atención al Paciente Hospitalizado*.

```
{ "_id": "ddf323794140b406e168dc5c84003376",
  "_rev": "19-a2c2ac079a2caf223d436b6057182682",
  "key": "plan",
  "Nombre": "Unicos",
  "version": "0.1",
  "Descripcion": "Se refiere al ingreso de una persona a una empresa institución de
salud en calidad",
  "Problema": {
    "Dominio": "11 Seguridad / protección",
    "Clase": "2 lesión física",
    "Etiqueta": "Protección ineficaz",
    "Factores": "Proceso infeccioso",
    "Caracteristicas": "Cifras de signos vitales",
    "Resultado": [
      "Mantener los signos vitales en parámetros normales",
      "Respuesta favorable a la medicación" ],
    "Indicador": {
      "0": [
        "Temperatura Corporal",
        "Frecuencia del pulso apical",
        "Frecuencia cardiaca apical",
        "Frecuencia del pulso radial",
        "Frecuencia respiratoria",
        "Presión arterial sistólica",
        "Presión arterial diastólica" ],
      "1": [
        "Temperatura Corporal",
        "Frecuencia del pulso apical",
        "Frecuencia cardiaca apical",
        "Frecuencia del pulso radial",
        "Frecuencia respiratoria",
        "Presión arterial sistólica",
        "Presión arterial diastólica" ] },
    "Escala": {
      "0": [
        "Desviación grave del rango normal",
        "Desviación sustancial del rango normal",
        "Desviación moderada del rango normal",
        "Desviación leve del rango normal",
        "Sin desviación del rango normal" ] },
    "Actividades": [
      "Monitorización de signos vitales",
      "Vigilancia de seguridad",
      "Administración de medicación" ] }
}
```

3.2.2. Diagrama arquitectónico

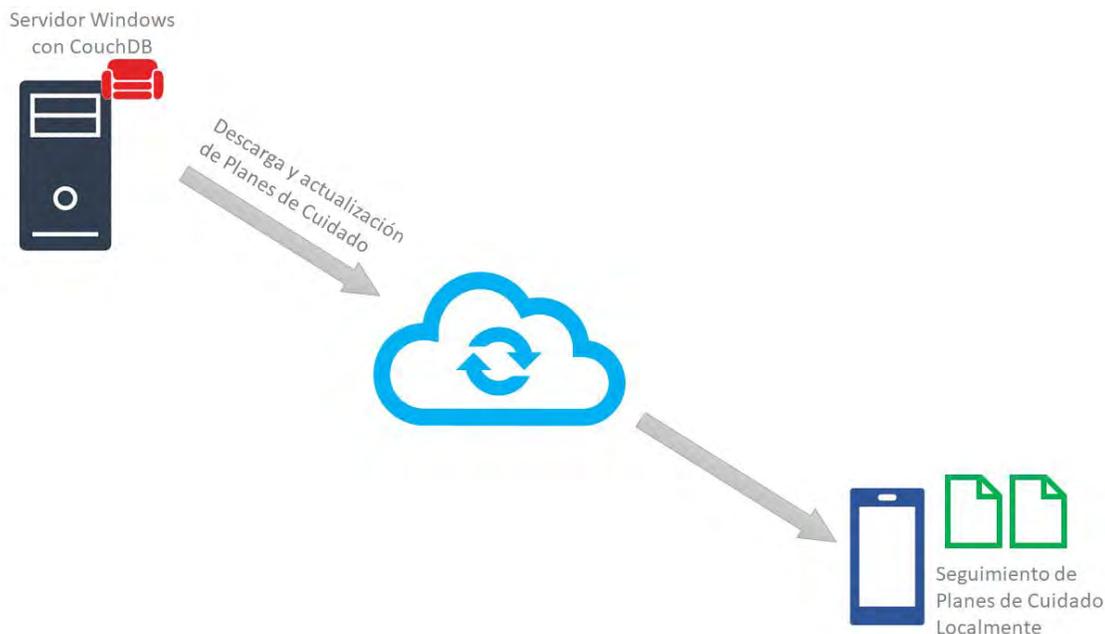


Figura 2. Esquema de la arquitectura de la aplicación. Elaboración propia.

La arquitectura de la aplicación consiste en una base de datos alojada en la nube, accesible a cualquier usuario de la aplicación a través de Internet, la funcionalidad de sincronización viene integrada dentro del código de la app, que cuenta con una sección definida para replicar todos los planes de enfermería existentes en CouchDB y llevarlos hacia el teléfono en donde quedarán almacenados en PouchDB, concluido este proceso, todos los planes estarán disponibles para el usuario, incluso cuando no cuente con una conexión a internet.

Por último, cuando un documento sufra una actualización o se haya agregado un nuevo objeto en CouchDB, la aplicación sincronizará los cambios hacia el teléfono, siendo este proceso transparente para el usuario.

3.2.3. Diagrama de navegación

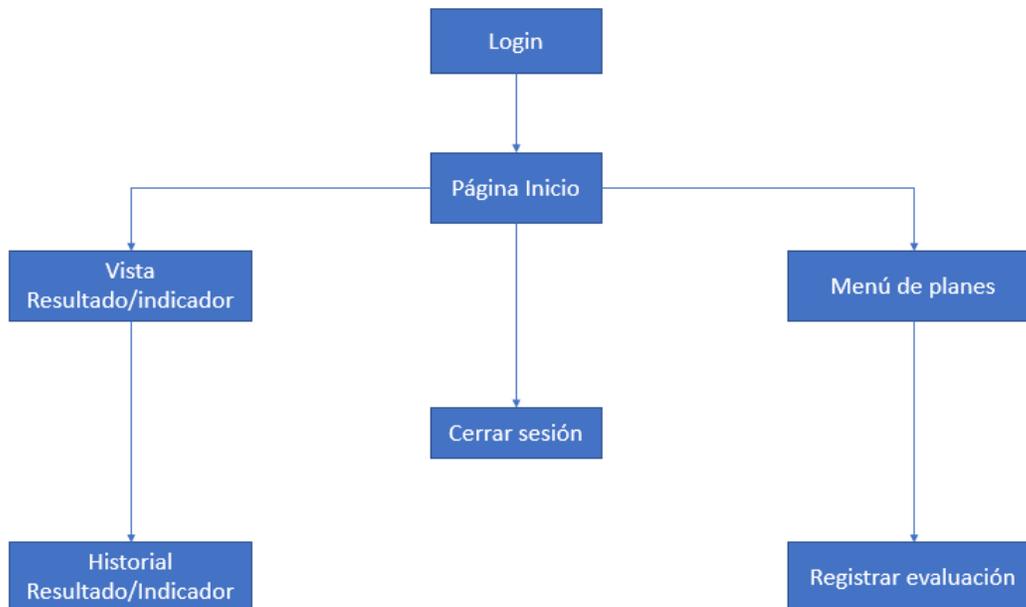


Figura 3. Flujo de navegación de la app **PLACE**.

3.2.4. Diseño de las interfaces gráficas

A continuación, se presentan los bocetos preliminares de la aplicación de acuerdo con el mapa de navegación en la Figura 3.



Figura 4. Pagina para agregar un nuevo paciente

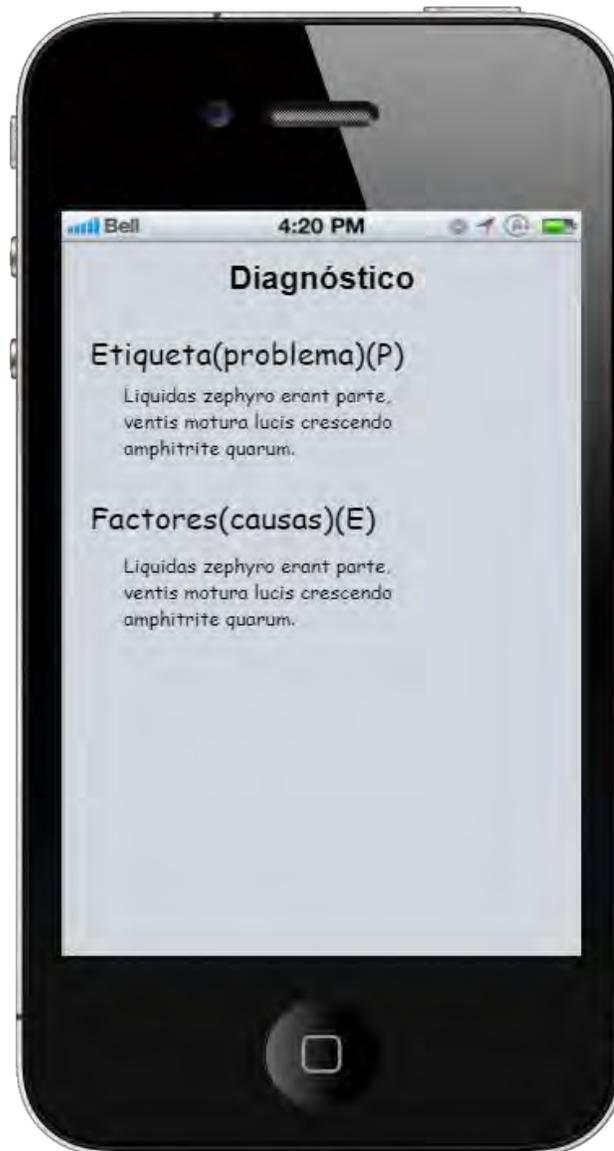


Figura 5. Diagnóstico



Figura 6. Inicio de Sesión



Figura 7. Lista de pacientes



Figura 8. Lista de planes



Figura 9. Página de resultados

3.3 Fase de Codificación

3.3.1. Desarrollo del módulo de sincronización de información.

Al cargar el proveedor de PouchDB de la aplicación, se crean dos instancias de PouchDB, la primera que se genera es una base de datos de manera local, a la cual se le pasarán todos los planes de enfermería provenientes de la nube.

A la segunda base de datos de PouchDB se le pasa la url del servidor donde se encuentra CouchDB, y gracias a la compatibilidad que existe entre estas dos herramientas, se accede a la información de CouchDB utilizando el plugin de PouchDB, esto quiere decir que PouchDB cumple con dos importantes funciones, proporcionarnos la base de datos para registrar las evaluaciones que se realicen y al mismo tiempo proporcionar las consultas de CouchDB necesarias para acceder a la información. Para que el proceso de sincronización funcione correctamente se tiene que conservar el *identificador* de cada documento al momento de que se guarda en PouchDB, ya que a través del Id PouchDB puede detectar alguna actualización que surge en CouchDB. Tras ocurrir una actualización en CouchDB, PouchDB detecta cual es el documento que sufrió un cambio, si el *id* de ese documento existe en la memoria interna, entonces se trata de una actualización, en caso contrario se trata de un nuevo documento que no existe en el dispositivo, por lo que se hace una copia y se guarda de manera local junto con los documentos ya existentes.

```
If plan.Id == null
    postNewPlan(plan)
Else
    updatePlan(plan)
```

3.3.2. Desarrollo de las interfaces de usuario

De forma paralela al desarrollo del módulo de la base de datos, las vistas de la aplicación comenzaron con su etapa de codificación se trabajó en las vistas de la aplicación, comenzando por las secciones que son independientes de la nube y que funcionaban de

manera correcta con la información que se genera de manera local o a través del proveedor de Autenticación (Firebase), para esto se consideraron las vistas de Login, Registro, Home y Agregar Paciente ya que estas secciones no necesitan interactuar con la lógica que funciona para sincronizar la información en la nube, gracias a esto se pudieron optimizar los tiempos de desarrollo.

La estructura de la mayoría de las vistas se basó iterando sobre un array de ítems provenientes del controlador en nuestros archivos HTML del módulo, esto es posible gracias al binding de datos que nos proporciona Angular en el cual se puede acceder a la información de un dato o variable de nuestro componente directamente desde la vista, y además utilizando la directiva “NgFor” de Angular para añadir funcionalidades a la vista, provocando que sean dinámicas y tengan funcionalidad más allá de solo presentar información”.

A continuación se muestra un ejemplo de la implementación directa de PouchDB para recuperar información en tiempo real proveniente de CouchDB, nota que la siguiente parte `{ live: true, since: 'now', include_docs: true }` es la configuración clave para detectar cualquier cambio proveniente del servidor

```

getTodosDocs(email) {
  return new Promise(resolve => {
    this.db.allDocs({
      include_docs: true
    }).then((result) => {
      this.dataLocal = [];

      let docs = result.rows.map((row) => {
        if (row.doc.key == 'paciente' && row.doc.usuario == email) {
          this.dataLocal.push(row.doc);
        }
      });

      resolve(this.dataLocal);

      this.db.changes({ live: true, since: 'now', include_docs: true }).on('change',
        (change) => {
          this.handleChange(change);
        });

    }).catch((error) => {
      console.log(error);
    });
  });
}

```

3.4 Fase de Pruebas

3.4.1. Elaboración del plan de pruebas

Planes que se cargaron a CouchDB

Plan de atención básica al paciente hospitalizado.

El plan que se utilizó en la etapa de pruebas corresponde al primer plan proveniente del Catálogo Nacional de Planes de Cuidados de Enfermería, el nombre del plan que se sintetizó fue: “Atención básica al paciente hospitalizado”, para esto se hizo un esquema de todo el contenido que originalmente proviene de un archivo de texto, esto con el objetivo de establecer los esquemas de clave-valor que contendrá nuestro plan en formato JSON, posteriormente se transcribió la información en la consola de CouchDB que se ejecuta en nuestro servidor.

Un plan estará disponible para todos los usuarios de la app PLACE cuando se incorpore al servidor. Al estar centralizada la información, basta con actualizar un único plan para que los cambios se reflejen en todos los dispositivos conectados, cumpliendo con uno de los propósitos del proyecto.

Para verificar que la información del plan se haya cargado correctamente se procedió a obtener la información a través de la aplicación, pero imprimiendo en consola el resultado de la petición a CouchDB, con esto nos enfocamos en concretar que la llamada sea exitosa antes de presentar los datos en la vista.

Encuesta de Usabilidad

La escala SUS es una herramienta que nos permite determinar el nivel de usabilidad de un sistema, objeto, sistema, dispositivo o aplicación, siguiendo un estándar desarrollado en 1986 por John Brooke, en el cual se toma un rango de 5 opciones partiendo desde “Fuertemente aceptable” hasta “Completamente en desacuerdo”.

Entre las grandes ventajas de utilizar SUS encontramos las siguientes (Devin, 2017):

- Es relativamente simple de usar para los participantes.
- Las preguntas suelen ser las mismas para cualquier sistema, aunque pueden presentar pequeñas variaciones.
- Con pequeñas muestras se pueden obtener grandes resultados.

Este sistema de usabilidad se puede es muy amplio y se puede utilizar en distintas áreas de software, tales como: sistemas, objetos, dispositivos, aplicaciones.

Su nivel de confiabilidad es muy seguro sin importar si se trata de un pequeño proyecto o uno de gran escala, por lo que es bastante usado hoy en día.

Las preguntas que se utilizaron para medir la escala de usabilidad fueron una adaptación de la prueba original en idioma inglés.

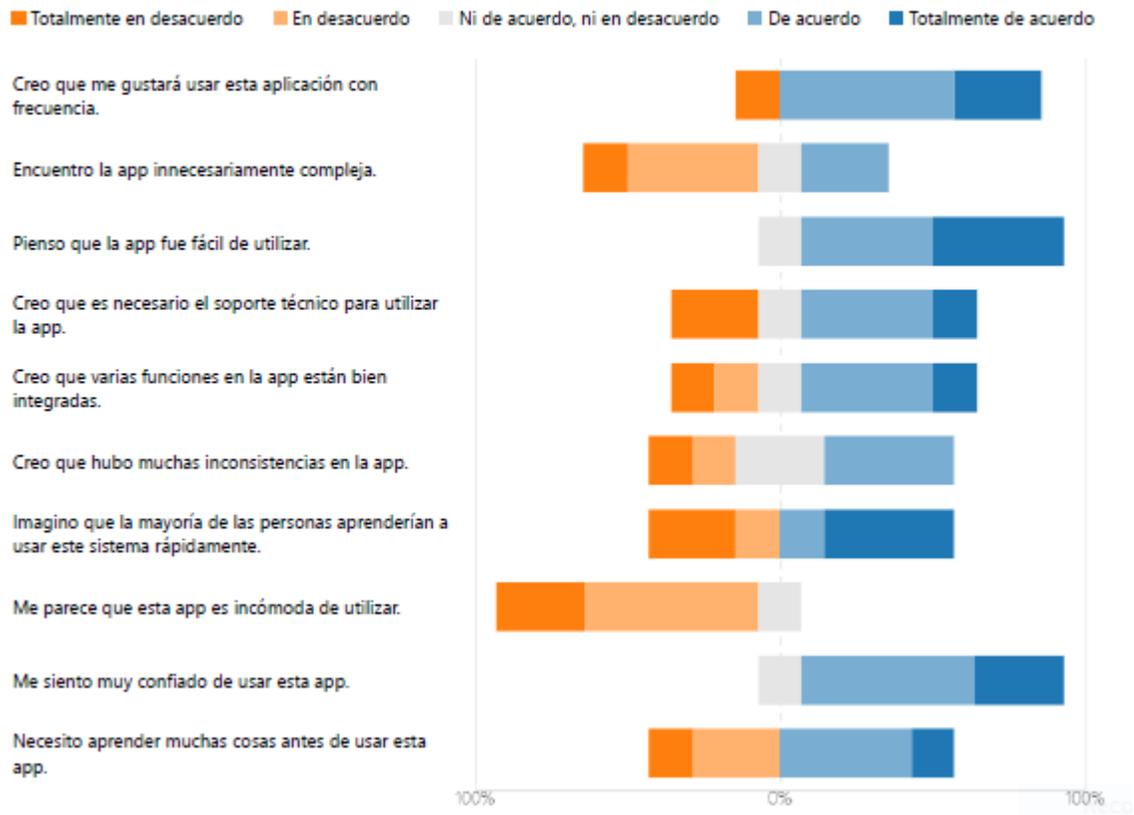
3.4.2. Ejecución de las pruebas con usuarios

En las pruebas de usabilidad participaron 7 estudiantes del 6o semestre de la Licenciatura en Enfermería de la División de Ciencias de la Salud en la Universidad de Quintana Roo y 4 enfermeras que laboran en el Hospital General de Chetumal. Esta prueba consistía en instalar la aplicación en sus dispositivos móviles, registrarse como usuarios de la aplicación, descargar uno de los PLACE disponibles, agregar pacientes, evaluar un paciente y mostrar el historial de evaluación. Posteriormente se les aplicó el cuestionario de usabilidad SUS (Brooke, SUS: A Quick and Dirty Usability Scale, 1996), el cuestionario se compone de cinco afirmaciones positivas y cinco afirmaciones negativas que se presenta de forma alterna. Cada afirmación presenta una escala Likert del 1 al 5, donde 1 corresponde a *Totalmente en desacuerdo* y 5 *Totalmente de acuerdo*. Después se calcula la puntuación de usabilidad propuesta en (Brooke, SUS: A Quick and Dirty Usability Scale, 1996) y se comparó con la escala presentada en (Bangor, Kortum, & Miller, 2009) donde es posible obtener una puntuación que sitúe a la aplicación en los siguientes rangos: lo peor imaginable (12.5), horrible (20.3), pobre (35.7), aceptable (50.9), buena (71.4), excelente (85.5), lo mejor imaginable (90.9).

3.4.3. Análisis de los resultados

Resultados de encuesta de usabilidad (SUS) a estudiantes de la Licenciatura en Enfermería

1. Para cada sentencia, elige la opción que más refleje tu opinión sobre la aplicación móvil:



En la segunda parte del cuestionario de usabilidad los usuarios expresaron su opinión a través de una pregunta abierta.

Comentario anónimo 1:

“En lugar de usar a diario las diferentes fuentes, me ayuda con más facilidad al momento de usar otras apps.”

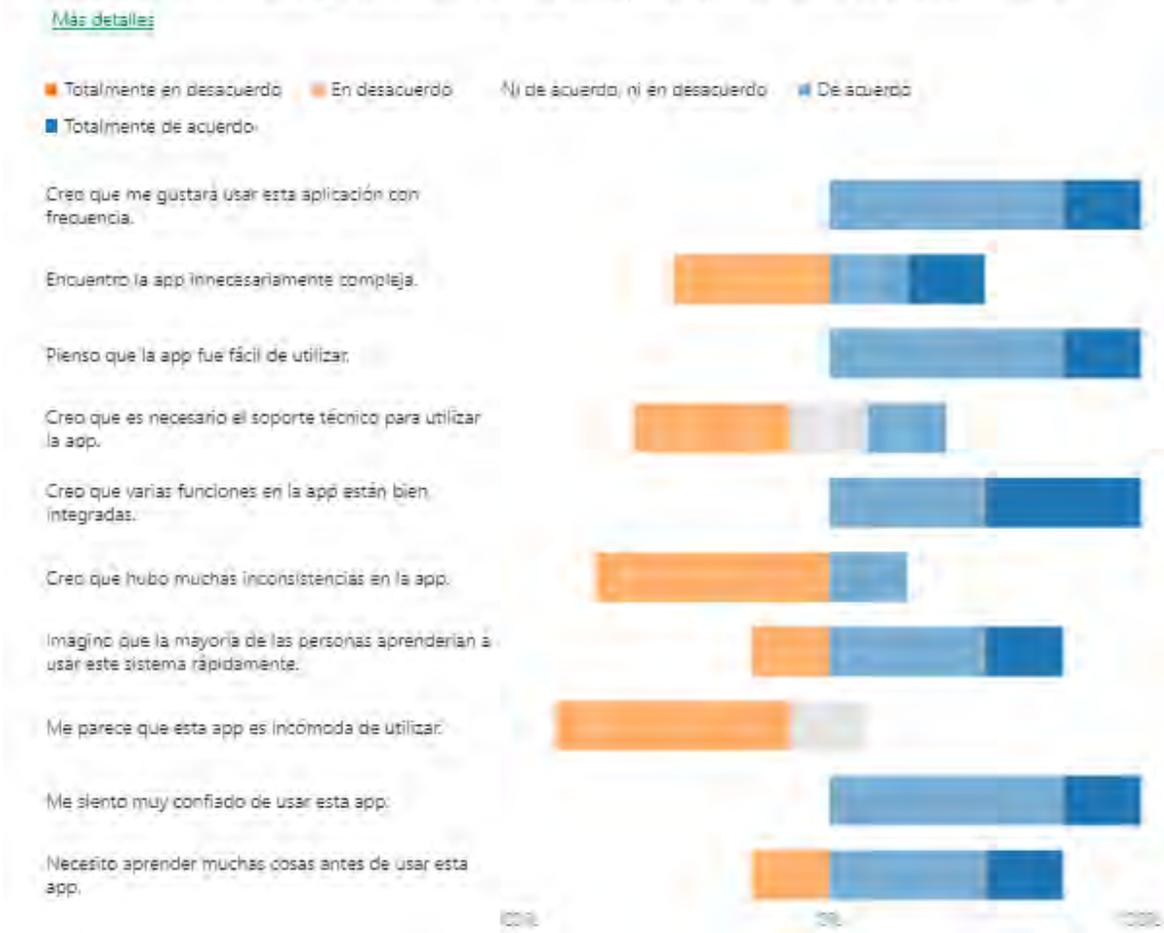
Comentario anónimo 2:

“Es necesario mejorarla en la red para que pueda funcionar de la mejor manera y así usarla en el hospital, también sería bueno agregarle otros datos en la información del paciente por ejemplo el número de cama, su diagnóstico.”

La puntuación media del cuestionario SUS que se obtuvo fue de 55. De los usuarios encuestados 6 dieron una puntuación de **BIEN** y 1 usuario **EXCELENTE**. Para valorar las puntuaciones se comparó con la escala en Bangor (2008), lo que significa que la usabilidad de la aplicación es buena para el grupo de estudiantes de la Licenciatura en Enfermería.

Resultados de encuesta de usabilidad a enfermeros del Hospital General

1. Para cada sentencia, elige la opción que más refleje tu opinión sobre la aplicación móvil:



En la segunda parte del cuestionario de usabilidad los usuarios expresaron su opinión a través de una pregunta abierta.

Comentario anónimo 1:

“Muy útil para nuestro trabajo.”

La puntuación media del cuestionario SUS que se obtuvo fue de 62.5. De los usuarios encuestados 4 dieron una puntuación de **BIEN**. Para valorar las puntuaciones se comparó con la escala en Bangor (2008), lo que significa que la usabilidad de la aplicación es buena para el grupo de estudiantes enfermeras del Hospital General de Chetumal.

3.5 Documentación

3.5.1. Manual del usuario

Manual de Usuario

1. Registrarse en la aplicación

Para poder entrar a la aplicación se necesita completar el registro para cuando se trata de un nuevo usuario, este registro incluye los siguientes datos:

- Email
- Teléfono
- Contraseña

Una vez que se hayan llenado todos los campos, se activará el botón de registrarse para que se complete el registro, tras ocurrir lo anterior se enviará un correo al usuario para que se verifique el email que ingresó, por lo que es importante registrar un correo al cual se tenga completo acceso

The screenshot shows a registration form with the following fields and values:

- EMAIL**: sergiodg105@hotmail.com
- TELEFONO**: 9831764180
- CONTRASEÑA**: (two instances)

A green button labeled **REGISTRARSE** is located at the bottom of the form.

Figura 10. Vista de registro del usuario.

Si el email que ingresaste no se encuentra verificado, no se podrá iniciar sesión en la aplicación.

2. Verificación de cuenta

Tras registrarse en la aplicación, se debe verificar el email que se registró, es necesario acceder a la cuenta y buscar el correo enviado por PLACE, es posible que el mensaje se encuentre en la carpeta de spam, tras abrir el correo se tiene que hacer clic en el enlace que aparece, tras completar esto, la cuenta quedará como verificada y se podrá iniciar sesión en la aplicación.

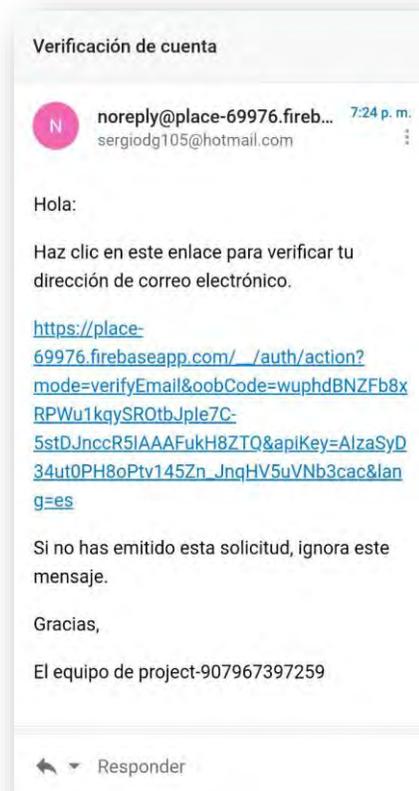


Figura 11. Correo de confirmación de cuenta

3. Inicio de sesión

Para ingresar en la aplicación se tiene que proporcionar el email y la contraseña el registro y que además se haya verificado el email de la cuenta. Si las credenciales son correctas, se permitirá el acceso a la aplicación y se cargará la página de inicio.

The image shows a mobile application login screen. At the top, the title 'Iniciar Sesión' is centered. Below it, there are two input fields: the first is labeled 'Email' and the second is labeled 'Contraseña'. Under the password field, there is a blue link that says 'Olvidaste Tu Contraseña?'. Below that is another blue link that says 'Registrarse'. At the bottom of the screen, there is a large, rounded green button with the text 'INGRESAR' in white capital letters.

Figura 12. Página de Login

4. *Página de Inicio.*

En esta sección se muestra la lista de pacientes que se han agregado, desde aquí se puede acceder a la vista para agregar un nuevo paciente, haciendo clic en el botón con el signo de más ubicado en la parte superior derecha.

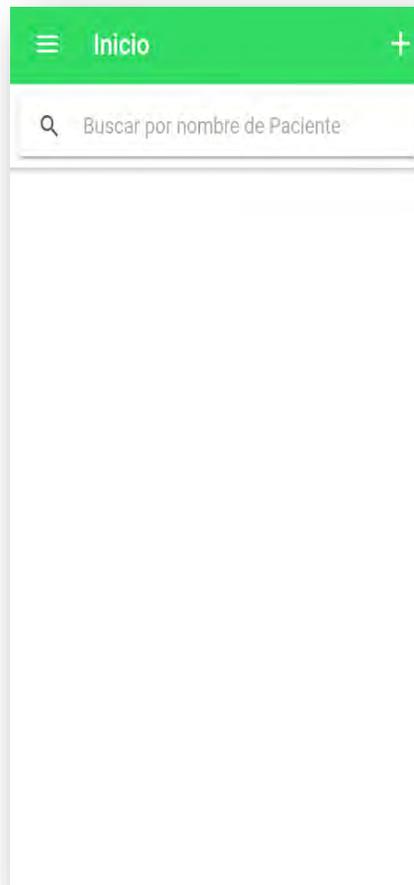


Figura 13. Página de Inicio

5. *Agregar nuevo paciente*

Para registrar de manera correcta a un paciente, el usuario debe de escribir en los campos que se indican información tal como el nombre, apellido, dirección estos tres últimos son datos obligatorios, tras esto, se desbloqueará el botón de Agregar paciente. Los campos de teléfono y email son opcionales y por lo tanto no se requieren para agregar a un nuevo paciente.

The screenshot shows a mobile application screen titled "Agregar Paciente". The interface consists of a green header bar with a white back arrow on the left and the text "Agregar Paciente" in white. Below the header, there are five white input fields with green labels: "Nombre(s)", "Apellido(s)", "Direccion", "Telefono", and "Email". At the bottom of the form, there is a green button with the text "AGREGAR PACIENTE" in white. The background of the screen is white.

Figura 14. Página para agregar un nuevo paciente

Cuando se agrega un nuevo paciente, la aplicación dirigirá al usuario a la página de inicio en donde estará disponible el nuevo paciente agregado.



Figura 15. Página de inicio con un nuevo paciente agregado

Menú que se ofrece de acuerdo a cada paciente para acceder a los distintos componentes de la aplicación, tales como: Sección de evaluación, editar información, eliminar al paciente o consultar su historial de evaluaciones.



Figura 16. Menú por paciente

6. *Seleccionar plan a evaluar*

En esta sección se encuentran todos los planes de enfermería que existen. Esta información se muestra resumida, mostrando únicamente el nombre del plan y la descripción.

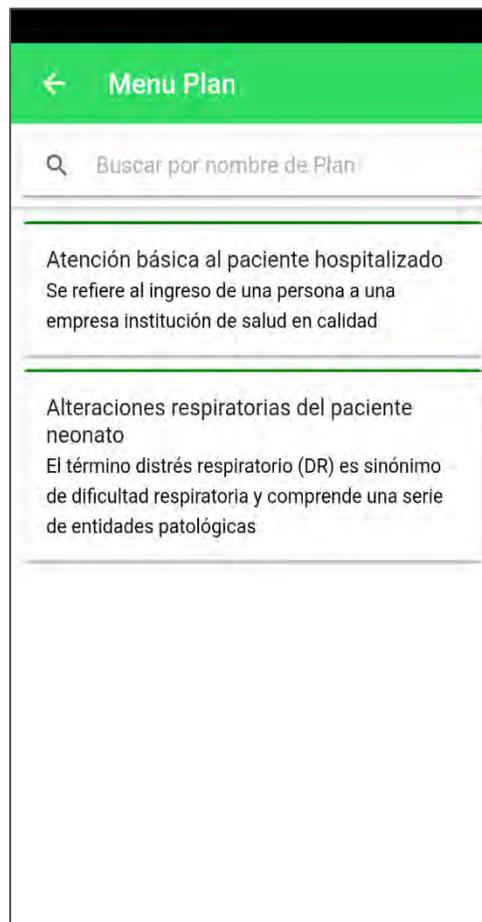


Figura 17. Menú de planes disponibles

Para acceder a toda la información del plan y agregar una nueva evaluación se deberá pulsar el plan con el que desea trabajar y tras lo cual se le llevará a la página de Evaluaciones.

Se dispone de un control de búsqueda para filtrar la información ya sea a través del nombre o descripción del plan desde el cuadro de búsqueda en la parte superior de la página.

7. Registrar evaluación

En esta sección, el usuario puede registrar las evaluaciones para el paciente que seleccionó desde la página de inicio. Para complementar una evaluación se debe seleccionar el texto que dice “Resultado”, esto desplegará una lista de todos los resultados que contiene el plan que se seleccionó en la página de “Planes”, cuando esto se haya completado, entonces se pondrán disponibles los indicadores correspondientes para el resultado seleccionado y por último se encuentra un rango para indicar la escala de la evaluación, con valores de 1 a 5, siendo el valor de 1 el que se registrará en la evaluación en caso de que no se haya seleccionado alguna otra escala.

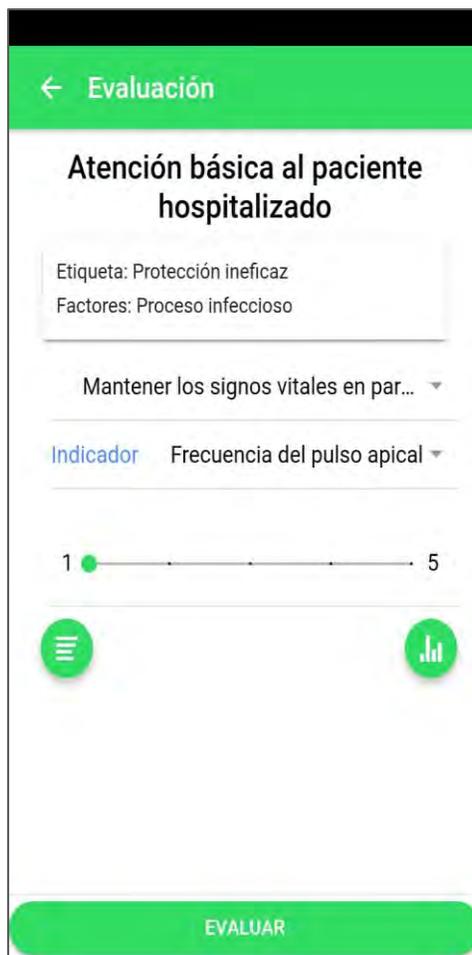


Figura 18. Página para realizar evaluación de paciente.

En la página de evaluación se proporciona ayuda para saber cuáles actividades corresponden al plan actual y lo que significa el valor asignado a cada escala.



Figura 19. Actividades disponibles.

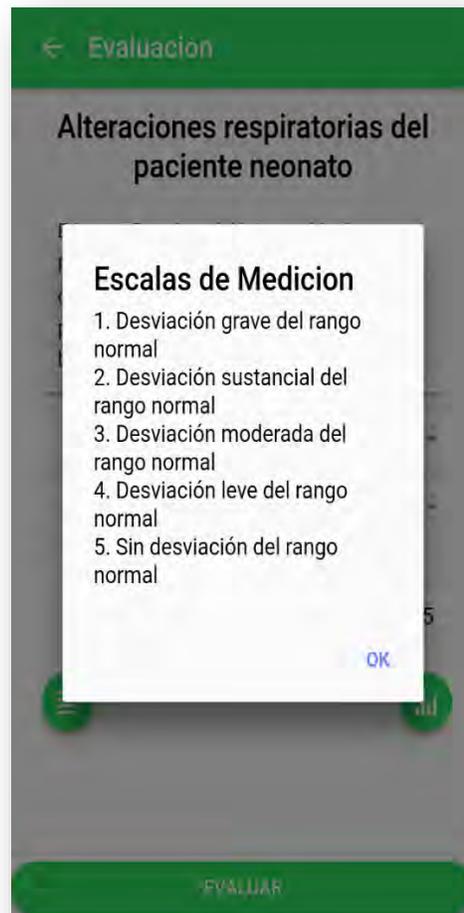


Figura 20. Valor asignado a cada escala.

8. *Historial de evaluaciones*

Se podrá acceder a esta sección desde la página de Inicio. Aquí se encuentra una lista resumida de las evaluaciones que se han hecho para el paciente que seleccionaste en la página principal.

La lista que se presenta incluye la última evaluación registrada agrupando la información por resultado e indicador, por lo tanto, si el usuario ha hecho dos evaluaciones para el resultado “R” con el indicador “I” entonces solo se mostrará la evaluación más reciente.



Figura 21. Historial de evaluaciones por usuario.

Capítulo 4 CONCLUSIONES

El desarrollo y evaluación de aplicaciones que apoyen las actividades profesionales de enfermería pueden aprovechar las metodologías de atención en enfermería aprobadas por Comisiones del Sector Salud en México a través de una plataforma tecnológica que facilite su uso y difusión.

La distribución de las aplicaciones móviles va en aumento entre la población que tiene a su alcance servicios de información, por lo que esta aplicación móvil se enfocó a mejorar el acceso a los planes de cuidado en Enfermería como una estrategia para su utilización entre los estudiantes y profesionistas de la enfermería.

La creación de software no es tarea fácil, se requiere de tiempo, paciencia y mucho empeño por parte del equipo de desarrollo (que para este caso sólo fue un programador) que pueda plasmar las ideas que surgen de una planeación y convertirlas en un proyecto funcional. Las tecnologías que se utilizaron eran desconocidas, se trabajó por primera vez con algunas de ellas por lo cual le agregó complejidad y riesgo al proyecto.

Esta fue la primera vez durante toda la formación académica que se trabajó en coordinación con otra disciplina no relacionada a la ingeniería. Por lo tanto, la problemática planteada surgió del ámbito de la enfermería y eso implicó conocer el entorno de esta profesión, su vocabulario y la forma de trabajar, entre otras características.

La aplicación contó con los requerimientos que se solicitaron al inicio del proyecto, se pudo compartir la aplicación con el grupo focal de estudiantes de enfermería de la División de Ciencias de la Salud, instalando la aplicación en sus celulares y posteriormente obtuvimos un rango de usabilidad de bueno a aceptable, de acuerdo con la encuesta SUS.

Gracias a los objetivos que se consiguieron y a las recomendaciones que se obtuvieron, es posible realizar una segunda etapa de este proyecto, incorporando nuevas características y haciendo ajustes a algunas tareas. Tomando como base el código realizado en esta tesis.

REFERENCIAS

- Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding and Adjective Rating Scale. *Journal of Usability Studies*, 4(3), 114-123.
- Brooke, J. (1996). SUS: A Quick and Dirty Usability Scale. En J. P. W., T. B., W. B. A. , & M. I. L., *Usability Evaluation in Industry*. London: Taylor & Francis.
- Brooke, J. (2013). SUS: A Retrospective. *Journal of Usability Studies*, 8(2), 29-40.
Recuperado el Septiembre de 2019
- Calvo, D. (7 de Abril de 2018). *Diego Calvo*. Obtenido de <http://www.diegocalvo.es/>
- Devin, F. (25 de Febrero de 2017). *UXpañol*. Obtenido de <https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>
- Indigitall. (5 de Mayo de 2018). *Indigitall.com*. Obtenido de <https://indigitall.com/crear-primera-app-cordova-framework/>: <https://indigitall.com/crear-primera-app-cordova-framework/>
- Ionic . (11 de octubre de 2019). *Ionic Framework*. Obtenido de <https://ionicframework.com/>
- PouchDB. (12 de Noviembre de 2019). *PouchDB*. Obtenido de <https://pouchdb.com/>
- Pryzbylsk, L. (4 de Julio de 2012). *InfoQ*. Obtenido de <https://www.infoq.com/articles/warner-couchdb/>
- Red Hat, Inc. (30 de mayo de 2018). *redhat*. Obtenido de <https://www.redhat.com/en/topics/cloud-computing/what-is-public-cloud#>
- Reza, M., & Saneipour, J. (2010). Self-* e-nursing: a new idea in nursing. *SIGSOFT Softw. Eng.*, 1-4. doi:10.1145/1734103.1734117
- Robles, V. (08 de Mayo de 2017). *VictorRoblesWeb*. Obtenido de <https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/>
- Rouse, M. (Julio de 2017). *TechTarget*. Obtenido de <https://searchdatacenter.techtarget.com/es/definicion/Desarrollo-de-aplicaciones-moviles>

Secretaria de Salud del Estado de Quintana Roo. (s.f.). Recuperado el 21 de Marzo de 2019, de <https://qroo.gob.mx/sesa/mision-y-vision>

Secretaria de Salud del Estado de Quintana Roo. (Marzo de 2010). Recuperado el 25 de Marzo de 2019, de <https://salud.qroo.gob.mx/portal/descargas/manual1.pdf>

Thomas, N. (18 de Abril de 2018). *usabilitygeek.com*. Obtenido de <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>

ANEXOS

Anexo 1. Configuración del entorno de desarrollo

Para el desarrollo de la aplicación, se necesitó de un equipo de trabajo, que fuera óptimo en cuanto a recursos, ya que el desarrollo móvil puede llegar a ser un poco lento debido a los recursos que necesarios al momento de lanzar la aplicación de la computadora hacia el dispositivo si el equipo en la que se está trabajando no es la indicada para este tipo de labores, pero no basta con solo contar con un buen equipo para realizar las tareas que se requieren, se necesita contar con todo el software necesario que te permita escribir las líneas de códigos de cada módulo de la aplicación, por lo cual se procedió a instalar los programas necesarios en el entorno de trabajo la lista de software instalado se tomó de acuerdo a las especificaciones que requiere Ionic. A continuación, se mencionan los programas utilizados en este proyecto:

Visual Studio Code

Visual Studio Code es el editor de código desde donde se crea cualquier componente o vista que se utilizará en la aplicación. Desde aquí se escriben las líneas de código en HTML para crear la estructura de código cada página o implementar la lógica necesaria para decirle al componente de cada módulo que nos lleve a una nueva página tras hacer click a un botón.

Visual Studio Code se encuentra disponible para los principales sistemas operativos como Windows, Linux o MacOS. Visual Studio Code incorpora una terminal desde la cual se pueden ejecutar cualquiera de las terminales o consolas disponibles en la computadora, por ejemplo, CMD o PowerShell, aunque se recomienda el uso de Git Bash. dentro de visual Studio Code se puede introducir cualquier comando para trabajar con Ionic, Angular o NodeJS, desde aquí se le puede indicar a Ionic que genere una nueva página, que se instale una nueva librería o paquete o que se genere un nuevo servicio para comunicarnos con el api de CouchDB.

CouchDB

La instalación de CouchDB es similar a otros programas en Windows, como primer paso se tiene que descargar el archivo de instalación a través de su página oficial:

<https://couchdb.apache.org/>

Cuando se haya completado la descarga, se tiene que abrir el archivo de instalación.

Una vez que se haya terminado la instalación de CouchDB, no se instalará un nuevo programa de Windows, en su lugar, se accede al portal de CouchDB a través del puerto 5984, escribiendo lo siguiente en el navegador: localhost:5984, tras esto, se nos abrirá la interfaz de administración, por cuestiones de seguridad, se tiene que crear un nuevo usuario y proporcionar una contraseña para la nueva cuenta que se creará.

Por último, se tiene que editar el archivo inicial de configuración, cambiando lo siguiente:

Esto se hace para permitir las conexiones a la base de datos desde cualquier dispositivo, si no se realiza esta configuración no se podrá acceder externamente a los planes de enfermería y estos solo estarán disponibles para el servidor.

NodeJS

Aunque se utiliza una consola para ejecutar el comando para instalar un paquete o un plugin es realmente el manejador de paquetes (npm) de NodeJS quien se encarga instalar las dependencias (al descargar el proyecto de un repositorio), librerías y plugins que se necesitan para hacer más útil y funcional la aplicación, de lo contrario, las apps en Ionic serían solamente visuales y no serían atractivas para los desarrolladores, en el repositorio de Node se encuentran almacenados todos los plugins que posteriormente se insertan en el repositorio gracias a NPM

Ionic

La instalación de Ionic se realiza mediante NPM, por lo que es necesario contar con NodeJS previamente instalado en la computadora, posterior a esto basta con introducir el comando “npm install – g Ionic 3.9.0” desde la terminal de visual studio code, seguidamente se empezarán a descargar todos los paquetes y dependencias que utiliza Ionic, por último, se puede verificar la correcta instalación tecleando de nueva cuenta en la terminal el comando

“ionic -v”, tras lo cual se muestra la versión que se ha instalado o incluso basta con revisar el archivo “package.json” que muestra el conjunto de plugins, herramientas y paquetes que se han instalado especificando la versión de cada uno.