

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Aplicación móvil para captura y centralización de
datos en la estrategia de rastreo de casos y contactos
de COVID-19 del Ministerio de Salud Pública y
Asistencia Social de Guatemala**

Trabajo de graduación presentado por José Pablo Viana Vidal
para optar al grado académico de Licenciado en Ingeniería en Ciencias
de la Computación y Tecnologías de la información

Guatemala
2022

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Aplicación móvil para captura y centralización de
datos en la estrategia de rastreo de casos y contactos
de COVID-19 del Ministerio de Salud Pública y
Asistencia Social de Guatemala**

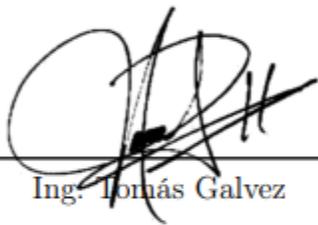
Trabajo de graduación presentado por José Pablo Viana Vidal
para optar al grado académico de Licenciado en Ingeniería en Ciencias
de la Computación y Tecnologías de la información

Guatemala
2022

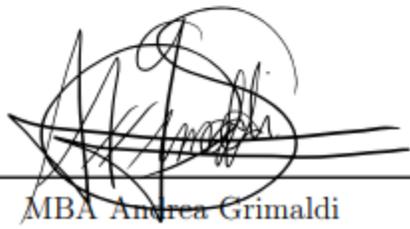
Vo.Bo.:

(f) 
Ing. Tomás Galvez

Tribunal Examinador:

(f) 
Ing. Tomás Galvez

(f) 
MSc. Douglas Barrios

(f) 
MBA Andrea Grimaldi

Fecha de aprobación: Guatemala, 10 de junio de 2022.

Primero, a mis padres y abuelos. Por ser el pilar y apoyo en mi vida, porque sin ellos no hubiera podido llevar a cabo esta investigación.

Segundo, a mis hermanos. Cuyo ejemplo de superación no ha sido nada más que motivación e inspiración para el desarrollo del proyecto.

Tercero, a mis amigos y pareja sentimental. Quienes a través de sus comentarios dieron ánimos para continuar con el esfuerzo, y nunca dudaron de mi capacidad para lograrlo.

Por último, me gustaría agradecerme a mí. Por la entereza de carácter para lograr superar los desafíos y siempre querer mantener una sonrisa.

Prefacio	II
Índice de figuras	VI
Resumen	VII
1. Introducción	1
2. Objetivos	3
2.1. Objetivo general	3
2.2. Objetivos específicos	3
3. Justificación	4
4. Marco teórico	6
4.1. ¿Qué es una pandemia?	6
4.2. Coronavirus	7
4.2.1. COVID-19	7
4.2.2. Síntomas	7
4.2.3. Transmisión	8
4.2.4. Demografía vulnerable al contagio	9
4.3. Pandemia de COVID-19 en Guatemala	10
4.4. Estrategia Nacional de Rastreo de Casos y Contactos	11
4.5. Aplicaciones móviles	12
4.5.1. ¿Qué es una aplicación móvil?	12
4.5.2. Aplicaciones móviles para COVID-19	13
4.5.3. Aplicaciones nativas	14
4.5.4. Aplicaciones multiplataforma	14
4.6. ¿Qué es un framework?	15
4.6.1. React Native	15
4.6.2. Almacenamiento asíncrono	19
4.7. Application Programming Interface	19
5. Metodología	21
5.1. Descripción del proyecto	21
5.2. Análisis de estado actual	23
5.3. Design Thinking	24

5.4. Decisión de herramientas	25
5.5. Programación del proyecto	26
5.5.1. Log-in y autenticación	26
5.5.2. Menú principal	27
5.5.3. Lectura de casos y contactos	28
5.5.4. Carga de información	31
5.5.5. Almacenamiento asincrónico	33
6. Resultados	34
7. Discusión de resultados	44
8. Conclusiones	46
9. Recomendaciones	47
10. Bibliografía	48
11. Anexos	52

Índice de figuras

1.	Casos positivos de COVID-19 en menores de 20 años en Guatemala marzo de 2022	9
2.	Casos positivos de COVID-19 por sexo en Guatemala marzo de 2022	10
3.	Renderizado en React Native	16
4.	Objeto JSON que contiene el estilo de un componente utilizando Flexbox	18
5.	llave/valor en memoria local de dispositivo móvil	19
6.	leer un valor en memoria local de dispositivo móvil	19
7.	Comunicación entre aplicación nativa y Go.Data	22
8.	Funcionalidades de la aplicación	23
9.	Prototipo a mano de la aplicación móvil	25
10.	Frameworks de desarrollo móvil a escala mundial (Liu, S., 2021)	26
11.	JSON de una respuesta satisfactoria del API con un log-in exitoso	27
12.	Menú principal de la aplicación.	28
13.	Captura parcial de JSON respuesta a lectura exitosa de caso	29
14.	JSON respuesta a lectura exitosa de contacto	30
15.	Información de ficha epidemiológica referente a caso	32
16.	Estructura del proyecto	35
17.	Splash screen de bienvenida con animación de logo Go.Data	36
18.	Reducer para manejo de LogIn	36
19.	Contexto de autenticación de la aplicación	37
20.	Ingreso por medio de usuario y contraseña	38
21.	Menú principal de la aplicación	38
22.	Vista general de todos los casos en sistema centralizado	39
23.	Vista detalle de caso en específico	39
24.	Vista detalle de ficha epidemiológica de caso en específico	40
25.	Vista de captación de datos para un nuevo caso	40
26.	Vista general de todos los contactos en sistema centralizado	41
27.	Vista detalle de contacto en específico	41
28.	Vista de captación de datos para un nuevo contacto	42
29.	Vista de captación de datos para un nuevo seguimiento	42
30.	Vista de carga exitosa	43
31.	Design Thinking fase de empatizar	52
32.	Design Thinking fase de empatizar	53
33.	Design Thinking fase de empatizar	54
34.	Design Thinking fase de empatizar	55
35.	Design Thinking fase de empatizar	56

36.	Design Thinking fase de empatizar	57
37.	Primer caso de uso	58
38.	Segundo caso de uso	59
39.	Tercer caso de uso	60
40.	Cuarto caso de uso	61

Ante la situación de alerta por pandemia de COVID-19 y el estado vulnerable en que se encuentra Guatemala, son imperativos los esfuerzos para la contención de la nueva enfermedad. Debido a la cantidad de infectados y la necesidad latente de una fuente de datos fehaciente, los sistemas de captación de información se vuelven una obligación. La presente investigación fue llevada a cabo para agilizar el proceso de captación de información por medio de la interoperabilidad de sistemas. Permitiendo a los colaboradores designados por el Ministerio de Salud Pública y Asistencia Social acceder y enviar datos a un sistema centralizado.

Se alimenta el banco de información de la estrategia de rastreo de casos y contactos de COVID-19 en Guatemala con nuevas ocurrencias de casos o contactos que ocurren a lo largo del territorio guatemalteco.

El proyecto se realizó implementando una aplicación móvil construida en React Native, la cual se comunica por medio de una API con una plataforma centralizadora de información. La aplicación realiza consultas sobre los casos y contactos de COVID-19 en el país y carga nueva información para alimentar el centro de datos. Esto hace posible obtener resultados mas precisos sobre la situación actual del país en temas referentes a la pandemia.

CAPÍTULO 1

Introducción

El coronavirus es una nueva pandemia para la humanidad. Se le informó a la Organización Mundial de la Salud (OMS) sobre una neumonía desconocida en Wuhan, China, el 31 de diciembre de 2019 (Kondylakis H et al., 2019). En febrero del 2020 la OMS nombró la enfermedad como COVID-19 (enfermedad por coronavirus 2019) y el 11 de marzo del mismo año la OMS declaró que el COVID-19 era una pandemia mundial. Taxonómicamente el nuevo coronavirus fue nombrado SARS-CoV-2. El día 13 de marzo se detecta y diagnostica el primer caso de COVID-19 en Guatemala y la primera muerte por dicha enfermedad el 15 del mismo mes (OPS, 2021).

Para la detección de casos de COVID-19 en Guatemala se realizan pruebas RT-PCR o antígeno a partir de muestras nasofaríngeas. Para la captación de información de las personas que se realizan dichas pruebas de COVID-19 el Ministerio de Salud Pública y Asistencia Social de Guatemala (MSPAS) generó la Ficha Epidemiológica para el COVID-19. En esta ficha se encuentra información personal del caso sospechoso como nombres, residencia, número telefónico e información clínica de interés, como la existencia de alguna enfermedad asociada como cáncer o asma. También capta la presencia de algún síntoma relacionado a la enfermedad. Además, información relevante a la prueba como tipo de prueba, fecha de toma de muestra, resultado y fecha de emisión de resultado.

Entre los esfuerzos para contener la nueva enfermedad, el MSPAS tuvo que ver la pandemia desde un punto epidemiológico para poder controlar los brotes emergentes de COVID-19. En los inicios de la pandemia, con el contexto de una vacuna inexistente, se realizaron los lineamientos para el rastreo de casos y contactos de COVID-19 (Ministerio de Salud Pública y Asistencia Social, 2020). El rastreo de casos y contactos es una estrategia utilizada para detectar nuevos casos en personas sospechosas de estar infectadas de manera temprana para poder interrumpir las cadenas de transmisión de SARS-CoV-2 (Smith AC et al., 2019). Debido al gran volumen de casos, los sistemas de captación de información son una necesidad. Para el manejo y creación de estos sistemas existe el Sistema de Información Gerencial de Salud (SIGSA) cuyo objetivo es proveer información relacionada con la salud de la población guatemalteca.

Como recomendación de la OMS, SIGSA adoptó Go.Data como sistema de información para realizar la estrategia de rastreo de casos y contactos COVID-19. Go.Data es una aplicación gratuita realizada por la OMS con el fin de realizar rastreo de casos y contactos para la detección y contención de brotes. Cuenta con características que lo hacen flexible a las necesidades específicas de cada país. La OMS se encarga de dar acompañamiento a la implementación en los países y son los encargados de lanzar actualizaciones continuamente. Además, cuenta con un API que facilita la interoperabilidad con otros sistemas (Triantafyllidis A et al., 2019). Buscando agilizar el proceso de recolección de datos, se pretende conectar una aplicación móvil al sistema de administración de información Go.Data. Dicho sistema actualmente permite a los usuarios conectarse a través de la web, mediante un usuario y contraseña previamente autorizados, e introducir información de manera manual sobre los casos y contactos identificados. Si bien es cierto que existen esfuerzos que habilitan su utilización en dispositivos móviles, estas alternativas se encuentran limitadas en capacidad y funcionalidades, no son personalizables a los objetivos del MSPAS y su desarrollo es de carácter general.

Con la implementación de la solución de esta investigación, se busca dotar a los rastreadores del Ministerio con una herramienta que facilite su labor y expanda el alcance de sus capacidades. Los rastreadores son capaces de consultar la data actual del ente centralizado, teniendo acceso a los casos y contactos registrados hasta el momento de realizar la consulta. Dicha información se presenta de manera ordenada, en listas y tablas, lo que permite que los usuarios tengan acceso organizado a los detalles que existen para cada caso y cada contacto. Además de poder consumir datos, también se dota de la capacidad para modificar o cargar nuevas entradas. En la aplicación, se detalla con preguntas abiertas y cerradas la información que cada rastreador debe llenar para cargar nuevos datos, o modificar los existentes. Teniendo en consideración la manera en que se desarrolla la estrategia nacional, se programa el sistema para que pueda leer la última información consumida en el teléfono y guardar los cambios que se vayan haciendo sin necesidad de una conexión activa a internet. Para la lectura, el rastreador deberá actualizar la base de datos local que existe en el dispositivo antes de perder la conexión a la red Wi-Fi. Esto le dará acceso a los casos y contactos actualizados hasta la última vez que se consumió la información, aunque pierda o se desconecte de internet. Para la carga, el rastreador podrá agregar o modificar casos y contactos de manera normal, tal y como lo haría con una conexión activa, siguiendo su trabajo de recolección de información sin la conexión a internet. Una vez que el individuo tenga acceso de nuevo a una red Wi-Fi, podrá realizar la carga de todas las modificaciones al sistema Go.Data. De esta manera, logrando centralizar toda la información captada en los dispositivos.

2.1. Objetivo general

Agilizar la captación de información por medio de una aplicación móvil permitiendo acceder y enviar datos para el rastreo de casos y contactos de COVID-19.

2.2. Objetivos específicos

- Comunicar la aplicación con una base de datos empleando una estructura JSON definida para el envío y recepción de mensajes a través de las funciones de un API.
- Garantizar la disponibilidad y acceso ininterrumpido a los datos ingresados con la implementación de un esquema de almacenamiento local actualizado automáticamente independiente de la conexión a internet.
- Permitir el ingreso y consulta de casos y contactos desde un dispositivo móvil Android sin distinción alguna al hecho que esté o no conectado a internet.

El número exponencialmente creciente de casos de COVID-19 ha llevado a una necesidad urgente de expandir actividades de salud pública con el fin de dilucidar la epidemiología del nuevo virus y determinar su impacto [2]. La velocidad en la que se desarrolla la pandemia plantea un desafío sin precedentes para la recolección de datos, lo que dificulta caracterizar el alcance completo de la enfermedad (Rosenthal, G., 2020). Durante cualquier brote o emergencia de salud, la captación y acceso oportuno a datos validados, su interpretación, junto a la toma de decisiones rápidas, son unos de los mayores desafíos que enfrentan muchos expertos. Un reto que crece en complejidad en entornos de escasos recursos. Esto es especialmente cierto para contextos donde se encuentran sistemas de vigilancia epidemiológica de bajo rendimiento, inexistentes, en desarrollo, o inexperimentados. Tales retrasos en el control de brotes durante una emergencia repercuten en acciones tardías de salud pública para mitigar la transmisión progresiva, y una detección de casos retrasada. En consecuencia, aumento de casos y mayor mortalidad de COVID-19 (WHO, 2019).

El objetivo de la estrategia de rastreo es dar seguimiento a personas en cuarentena, detectar nuevos casos, enviar a cuarentena a nuevos casos y velar por la buena evolución de estas mismas personas. Mientras una persona está bajo seguimiento se le puede enviar un kit médico, dar consultas médicas o referir a un centro de salud (Ministerio de Salud Pública y Asistencia Social, 2020). Para lograrlo, se implementó Go.Data como sistema de información para el monitoreo de la estrategia de rastreo de casos y contactos en Guatemala. Existen casos de éxito en la utilización de software de dispositivos móviles para el monitoreo y vigilancia de enfermedades. Por ejemplo, la aplicación WeTrace, que se encarga de la recolección de información básica referente a la salud de individuos a través de una aplicación, especialmente para COVID-19 (lanzi T., 2021). También, registros sobre aplicaciones que ayudan en la comunicación enfermera-paciente en la labor de vigilancia de enfermedades crónicas, como las cardíacas y renales. Por lo que la utilización de aplicaciones en dispositivos móviles es recurrente como sistema de vigilancia en temas de salud. En más de una oportunidad, cómo solución para el seguimiento, control y monitoreo de enfermedades. Depender de fuentes de datos digitales, como datos de teléfonos móviles y otros dispositivos digitales, es de especial valor en los brotes causados por patógenos recién descubiertos, para los que aún son escasos los datos oficiales y las previsiones fiables. Este trabajo de investigación se propone cómo una alternativa para agilizar la captación de datos de casos y contactos llevada a cabo por el MSPAS. El aporte del presente esfuerzo es una aplicación que busca facilitar la captación de información y toma en consideración requerimientos propios del Ministerio de Salud. Pensada desde su concepción para ajustarse a la

manera de trabajar de los rastreadores, quienes son los usuarios finales. Capaz de manejar la falta de conectividad existente en las áreas geográficas donde el personal de salud captura los datos. Dotando a los rastreadores con la capacidad de enviar dicha información para su centralización cuando se cuente con acceso a una red. Además, utiliza la estructura de datos para interoperabilidad definida para la correcta interacción con Go.Data. Sumado a este flujo de trabajo considerando el contexto, está la capacidad única de poder realizar seguimientos a casos y contactos desde el aparato móvil. Tiene documentación y es escalable. De esta manera, se busca apoyar al monitoreo de la evolución de la enfermedad en el territorio guatemalteco y ayudar al despliegue de los indicadores de rastreo en tiempo real. Se utilizará una aplicación híbrida como una herramienta para la captación de datos a través de dispositivos móviles y su posterior centralización en el sistema Go.Data.

El presente trabajo contempla el desarrollo e implementación de una aplicación móvil para la estrategia de rastreo de casos y contactos de COVID-19. Dicha aplicación debe contar con características específicas para cumplir con los requerimientos del entorno en el cual se piensa utilizar. Por lo tanto, es necesario desarrollar conocimientos y requerimientos que servirán como base de apoyo para la comprensión del proyecto y el proceso realizado.

4.1. ¿Qué es una pandemia?

Una pandemia es la propagación mundial de una nueva enfermedad (Vargas et al., 2018). La palabra pandemia viene del griego *pandemos* que significa todos, refiriéndose a que existe la creencia de que la población mundial es propensa a estar expuesta al virus. Se tiene un precedente en el año 2009 con el rápido contagio del virus de influenza H1N1 que causó confusión con el significado de la palabra "pandemia" y cómo reconocer cuando ocurre una. Usos del término por agencias de salud, científicos y los medios de comunicación tienden a parecer extraños. Algunos argumentan que un nivel explosivo de transmisibilidad es suficiente para declarar una pandemia, mientras otros mantienen que un nivel severo de infección también debe ser considerado (David M et al., 2009). Otros ejemplos de una pandemia son la peste bubónica en el siglo XIV, síndrome respiratorio agudo grave (SARS) en 2003 y HIV/SIDA.

Definiciones modernas para pandemia incluyen: extensamente epidémico (Stedman's medical dictionary, 2006), epidemia sobre una larga área de extensión y usualmente afectando a una gran proporción de la población (A dictionary of epidemiology, 1998) y distribuido o que ocurre ampliamente en una región, país, continente o globalmente, entre otras. Aunque muchas definiciones convergen en la idea intuitiva que una pandemia es una epidemia de grandes proporciones, dichos conceptos aún parecen vagos. Se sugiere que el concepto de pandemia, aplicado a eventos globales importantes que abarcan muchos siglos, incluye enfermedades de etiologías muy diferentes que exhiben una variedad de características epidemiológicas. Parece haber un solo denominador común invariable: la extensión geográfica generalizada. Sin embargo, existen características epidemiológicas comunes, por ejemplo, el movimiento y las altas tasas de ataque, mientras que otras características variables, como la no infecciosidad y la gravedad, parecen fuera de lugar (David M et al., 2009).

4.2. Coronavirus

Los coronavirus (CoV) son una larga familia de virus de ARN monocatenario, que pueden infectar a animales y también a humanos, causando enfermedades respiratorias, gastrointestinales, hepáticas y neurológicas (Weiss y Leibowitz, 2013). Como los virus ARN más grandes conocidos, los CoV se dividen en cuatro géneros: alfa-coronavirus, beta-coronavirus, gamma-coronavirus y delta-coronavirus (Yang y Leibowitz, 2015). Los nuevos coronavirus parecen surgir periódicamente en humanos, principalmente debido a la alta prevalencia y amplia distribución del coronavirus, la gran diversidad genética, la frecuente recombinación de sus genomas, y el aumento de las actividades de interacción humano-animal (Cui et al., 2019, Zhu et al., 2019).

4.2.1. COVID-19

El COVID-19 es una enfermedad infecciosa causada por el virus SARS-CoV-2, un tipo de coronavirus recientemente descubierto. La mayor parte de personas infectadas con este virus experimentarán una enfermedad respiratoria leve a moderada, y se recuperarán sin la necesidad de un tratamiento especial. Las personas mayores, o aquellas con problemas médicos preexistentes, como diabetes, enfermedades respiratorias crónicas o cáncer, tienen más probabilidades de desarrollar complicaciones serias. Cualquiera puede enfermarse de COVID-19 y desarrollar una enfermedad grave o morir, sin importar la edad.

La mejor manera de prevenir y disminuir la transmisión del virus, es mantenerse informado y conocer las maneras cómo se propaga. Es posible protegerse uno mismo y a los demás de la infección, permaneciendo al menos un metro aparte unos con otros, usando una mascarilla ajustada correctamente y lavándose las manos con frecuencia.

El virus se puede esparcir desde la boca o nariz de una persona infectada en pequeñas partículas de líquido cuando esta tose, estornuda, habla, canta o respira. Es importante poner en práctica buenas costumbres de respiración y etiqueta, por ejemplo, toser con el codo flexionado en la boca, y quedarse en casa en aislamiento hasta recuperarse en caso de presentar síntomas (World Health Organization, 2019).

4.2.2. Síntomas

El COVID-19 afecta de distintas maneras en función de cada persona. La mayoría que se contagia presenta síntomas de intensidad leve o moderada, y se recuperan sin necesidad de hospitalización. Los síntomas más habituales se reconocen como (World Health Organization, 2019):

- Fiebre
- Tos
- Cansancio
- Pérdida del gusto
- Pérdida de olfato

Los menos habituales como:

- Dolor de garganta
- Dolor de cabeza

- Molestias y dolores
- Diarrea
- Erupción cutánea o pérdida del color de los dedos de las manos o los pies
- Dolor en el pecho

Los primeros síntomas en aparecer son comúnmente fiebre, tos seca, taquipnea y dificultad para respirar (HUI, D et al., 2020). Aunque la diarrea estuvo presente en aproximadamente 20 a 25 % de los pacientes con infección por MERS-CoV o SARS-CoV, los síntomas intestinales rara vez se observan en pacientes con COVID-19. En otro estudio, también se informó confusión, dolor en el pecho, vómitos y náuseas como síntomas de COVID-19 (Chen, N et al., 2020) Mientras el 80 % de las personas se recuperan sin la necesidad de asistencia médica. Alrededor de 1 de cada 5 personas que contraen el COVID-19 presentan complicaciones y dificultad para respirar (World Health Organization, 2019).

4.2.3. Transmisión

Se sabe que la enfermedad es causada por el virus SARS-CoV-2, el cual se esparce entre las personas de diferentes maneras. (World Health Organization, 2019). El SARS-CoV-2 puede propagarse tanto por medios directos (gotas pequeñas y transmisión de persona a persona) como por contacto indirecto (objetos contaminados y contagio aéreo). El equipo de protección personal (EPP), un equipo especial que se usa para crear una barrera entre el individuo y los microbios, también podría ser la fuente de infecciones transmitidas por el aire (Y, Liu et al., 2020).

La evidencia actual sugiere que el virus se propaga principalmente entre personas que están en contacto cercano entre sí, generalmente dentro de 1 metro. Una persona puede infectarse cuando se inhalan aerosoles o gotas que contienen el virus o entran en contacto directo con los ojos, la nariz o la boca. El virus también se puede propagar en entornos interiores mal ventilados y/o concurridos, donde las personas tienden a pasar períodos de tiempo más prolongados. Esto se debe a que los aerosoles permanecen suspendidos en el aire o viajan más de 1 metro. El virus SARS-CoV-2 puede estar suspendido en el aire por hasta 3 horas (Van Doremalen, 2020). Por lo tanto, la ventilación de la habitación y la aplicación adecuada de desinfectante pueden restringir la propagación del virus en aerosol (Santarpia, 2020).

Las tasas de contagio informadas de un paciente con infección sintomática varían según la ubicación y la eficiencia de las medidas de control de la infección. Según un informe conjunto de la OMS y China, la tasa de infección secundaria por COVID-19 osciló entre el uno y el cinco por ciento entre decenas de miles de pacientes confirmados en China (McIntosh K et al., 2020). Aunque se ha descubierto ARN del SARS-CoV-2 en muestras de sangre y heces (Tang A et al., 2020) y se ha cultivado de las heces en algunos pacientes con COVID-19 (Wang W et al., 2020). Un informe conjunto OMS-China indicó que la vía de transmisión fecal-oral no parecía ser un factor importante en la propagación de la infección (McIntosh K et al., 2020). Con base en la evidencia mencionada anteriormente, la transmisión del virus podría ser más complicada que la observada en pandemias anteriores. El virus es muy contagioso. Además, un período de incubación relativamente largo, la presencia de pacientes asintomáticos y la diseminación viral continua después de la recuperación, subrayan la importancia de la cuarentena domiciliar a través del bloqueo de toda la sociedad (Wilder-Smith A et al., 2020).

4.2.4. Demografía vulnerable al contagio

Entre los grupos mayoritariamente afectados están las personas de edad avanzada, se debe a que la severidad de los síntomas incrementa con el tiempo de vida. Información desde China hasta Italia sugiere una tasa de fatalidad por caso de 2.3 % en pacientes con COVID-19, con más del 50 % de las defunciones ocurriendo en pacientes de 50 años o mayores (Yanez et al., 2020). En la serie más grande informada del norte de Italia, la letalidad en pacientes de 64 años o más fue del 36 % en comparación con el 15 % en pacientes más jóvenes (Grasselli et al., 2020). Individuos de 65 hasta 85 años tienen tasas de mortalidad por COVID-19 sorprendentemente más altas en comparación con las personas jóvenes. Los hombres tienen un mayor riesgo de muerte por COVID-19 que las mujeres. Las personas con una edad comprendida en el rango de 85 años o más se encuentran entre los más vulnerables.

A diferencia de los adultos, los niños con COVID-19 tienen síntomas más leves y mejores resultados clínicos (Rezaei N, 2020). Entre los pacientes con COVID-19 menores de 18 años, los niños menores de un año parecen tener el mayor riesgo de padecer la forma grave de la enfermedad (Dong Y et al., 2020). Aunque los primeros estudios mostraron que los niños con COVID-19 tenían menos probabilidades de desarrollar síntomas graves que otros grupos de edad, un nuevo estudio ha demostrado que los niños tienen la misma probabilidad de desarrollar COVID-19 que los adultos (Bi Q et al., 2020). Por tanto, la prevención y la búsqueda de un tratamiento adecuado para los niños es tan importante como para los adultos. En un estudio, se encuestó a 1391 niños con una edad media de 6,7 años y se les diagnosticó COVID-19 a 171. La Unidad de Cuidados Intensivos (UCI) y la ventilación mecánica fueron necesarias solo para tres niños, que tenían afecciones severas. Para el 8 de marzo de 2020, un niño con invaginación intestinal había muerto, 21 niños permanecían en condición estable en la sala y 149 casos fueron dados de alta (Lu X et al., 2020).

En Guatemala es posible observar un comportamiento que coincide con el descrito en la literatura. Teniendo un aumento de casos positivos en niños de un 1 año, en relación con los demás años de la infancia y niñez, posiblemente debido a la exacerbación de los síntomas en esta etapa de la infancia. Seguido, una curva creciente conforme avanza el eje del tiempo, hasta llegar al mayor número de casos, que se encuentra en personas de 20 años.

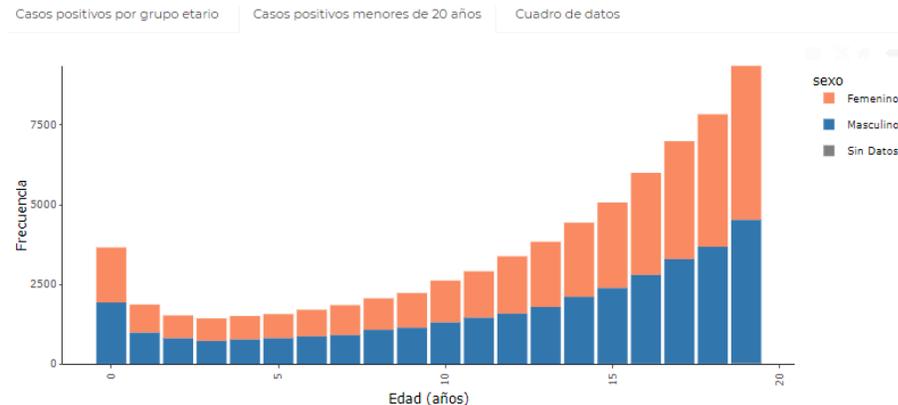


Figura 1: Casos positivos de COVID-19 en menores de 20 años en Guatemala marzo de 2022

Se puede observar también que se mantiene una proporción similar entre hombres y mujeres en los casos positivos de COVID-19 en Guatemala, teniendo una leve inclinación hacia la población

masculina, que también concuerda con la bibliografía consultada.

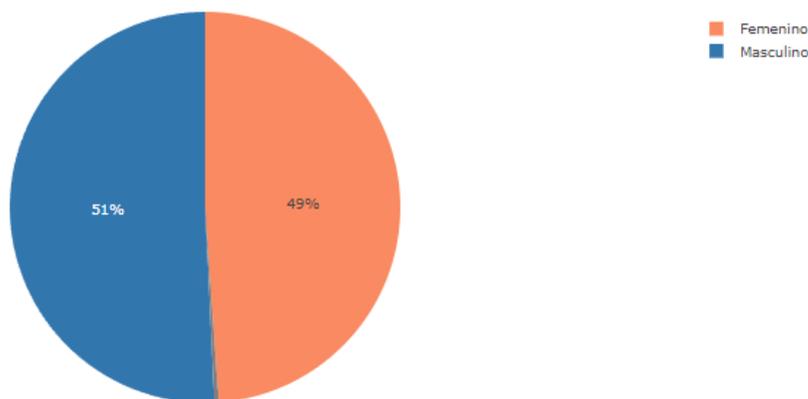


Figura 2: Casos positivos de COVID-19 por sexo en Guatemala marzo de 2022

4.3. Pandemia de COVID-19 en Guatemala

La pandemia COVID-19 causó directamente al menos 2 millones de muertes confirmadas en todo el mundo durante 2020. Durante los primeros meses de la pandemia, los impactos de mortalidad más fuertes ocurrieron en Inglaterra y España, con un exceso de mortalidad del 37 % y 38 %, respectivamente, entre mediados de febrero y mayo de 2020 (Sun P et al., 2020). Sin embargo, el impacto en las regiones menos desarrolladas, como América Latina, es mayor. Los sistemas de salud fragmentados, las amplias desigualdades sociales y una alta prevalencia de enfermedades crónicas contribuyen a agravar el número de víctimas en pandemias en países en vías de desarrollo (Glassman A et al., 2010). Estos factores pueden haber contribuido al elevado número de víctimas de COVID-19 en América Latina. Guatemala, informó su primer caso de COVID-19 el 13 de marzo de 2020. En respuesta, el gobierno guatemalteco implementó varias medidas de control. Restricciones de movilidad, órdenes para quedarse en casa, prohibición del consumo de alcohol y cierres nacionales, fueron algunos de los mecanismos de contención de la enfermedad empleados por el gobierno. Estas políticas fueron posteriormente revisadas en periodos de 15 días y se endurecieron o relajaron según la cantidad de casos de COVID-19 en el país (Martínez-Folgar K et al., 2021).

Estudiar el impacto de la pandemia al observar solo las muertes confirmadas por COVID-19 proporciona una imagen incompleta de la carga de la pandemia. Los recuentos de muertes confirmadas generalmente requieren confirmación de laboratorio de la infección por coronavirus (SARS-CoV-2), la cual se ve limitada en el territorio guatemalteco (Prensa Libre, 2020). La enfermedad puede exacerbar otras afecciones de salud, que causan muertes que pueden no clasificarse como causadas directamente por COVID-19. Los primeros datos de estudios observacionales, que posteriormente se han corroborado tanto en revisiones sistemáticas como en varios metaanálisis, demuestran que la hipertensión arterial, la diabetes mellitus tipo 2 (DM2), la obesidad y la enfermedad coronaria son las principales comorbilidades en la infección por COVID-19 (Pérez-Martínez, P et al., 2020). Las interrupciones en la prestación de servicios de salud pueden provocar un retraso en la atención de otras dolencias que conduzcan a un aumento de la mortalidad atribuible a otras causas (Rosenbaum L, 2020).

Cabe resaltar de entrada que, si bien ningún país del mundo está preparado para enfrentar la pandemia, Guatemala se presenta singularmente vulnerable por algunos rasgos sistémicos que caracterizan al país.

Primero, se registra una muy elevada incidencia de pobreza y pobreza extrema. Desigualdades insostenibles en niveles de ingreso y en oportunidades de acceder a servicios básicos, recursos y empleo remunerado. En ese sentido, la mayoría de la población tiene muy restringidas posibilidades para enfrentar las consecuencias de la pandemia. Segundo, se cuenta con un estado debilitado durante décadas en cuanto a su capacidad de gestión en casi todos los niveles. Muchas veces marcado por prácticas corruptas, y una creciente penetración del crimen organizado en las estructuras gubernamentales, a nivel departamental y nacional. Tercero, los indicadores denotan una falta de confianza por parte de la población en general en las autoridades electas; podemos mencionar el Índice de Percepción de la Corrupción (IPC). Indicador que da a Guatemala una puntuación de 25 puntos sobre 100, posicionándose en el lugar 149 de 180 países. Guatemala, ubicado por debajo del promedio mundial, no ha hecho reformas estructurales para combatir la corrupción de manera efectiva (FUNDESA, 2020). Esto fortalece una desconfianza alimentada en alto grado por la opacidad en que se desenvuelve la clase política del país. Eso, en momentos en que se precisa confianza entre Gobierno y ciudadanos para consensuar los pasos a seguir. Cuarto, según la Organización Panamericana de la Salud (OPS) Guatemala dispone de un sistema de salud particularmente debilitado: el gasto público en salud asciende a 2.2 % del Producto Interno Bruto (PIB), cuando el promedio del resto de Centroamérica asciende a 4.0 %. El país aparece en último lugar en América Latina y el Caribe en el número de camas hospitalarias por habitante (0.6 camas por cada 1000 habitantes). Casi 70 % de la población económicamente activa se desempeña en la economía informal, en general de baja productividad y que obliga a vivir bajo el modelo de “día a día”. Un elevado porcentaje de guatemaltecos vive hacinado en espacios de alta concentración demográfica en los principales centros urbanos. Donde el distanciamiento social es difícil, si no imposible, y en las áreas rurales el instructivo de lavarse las manos con frecuencia a veces no se cumple simplemente porque no se cuenta con acceso a agua potable.

Sumado a ese contexto ya complicado, en los últimos tiempos el país se ha enfrentado a fenómenos climáticos que han incidido sobre la agricultura. Lo que provoca que se amplíe cada vez más un “corredor seco” en el centro del país con elevado riesgo de sequía. Ello ha conducido, entre otros fenómenos, a graves problemas de seguridad alimentaria y desnutrición crónica, sobre todo de menores de edad. En efecto, según el Fondo de las Naciones Unidas para la Infancia (UNICEF), Guatemala es actualmente el sexto país del mundo con peores índices de malnutrición infantil. En algunas regiones rurales de Guatemala los índices de malnutrición infantil alcanzan 80 % (Rosenthal, G, 2020).

Guatemala se encuentra en una situación endeble. Aunado a todo lo anterior, la salud mental en las comunidades marginadas de Guatemala también es sujeto de interés para analizar el impacto del virus en el país (Rosenthal, G, 2020). Para muchas comunidades alrededor de la ciudad de Guatemala, las condiciones anteriores a la pandemia ya eran precarias. En estas denominadas ‘zonas rojas’ existen altos índices de extrema pobreza, crímenes violentos, con relación a pandillas y uso de sustancias. Prevalen de manera limitada las oportunidades de educación y empleo, así como altas las tasas de embarazo adolescente, delincuencia, violencia contra la mujer y doméstica. Para hacerle frente a tal crisis, el Poder Legislativo inició la adopción de una serie de leyes que dotan al Gobierno de la capacidad para poner en práctica medidas sanitarias, económicas, financieras y sociales (Branas, C., 2013). A octubre del año 2021, en Guatemala existen 6,579 casos activos estimados, 599,042 casos acumulados registrados, 577,526 casos recuperados estimados, 14,928 casos fallecidos registrados y un 2.4 % de letalidad de la enfermedad. La pandemia exige la toma de decisiones por parte de los distintos sectores del país para afrontar de manera exitosa los desafíos que esta situación presenta.

4.4. Estrategia Nacional de Rastreo de Casos y Contactos

El Ministerio de Salud Pública y Asistencia Social, en el contexto anteriormente descrito, pone en marcha la Estrategia Nacional de Rastreo de Casos y Contactos. Dicha estrategia pretende realizar

una labor de seguimiento y rastreo como esfuerzo para disminuir la transmisión de COVID-19 entre la población. El lanzamiento fue en primera instancia un plan piloto en el municipio de Guatemala. Este con la finalidad de detectar, evaluar y tomar acciones dirigidas a las personas que han sido expuestas al virus. Las acciones ejecutadas en momentos clave tienen como fin interrumpir la transmisión y minimizar el desarrollo de brotes en la población (Ministerio de Salud Pública y Asistencia Social, 2020).

Cuando se detecta un caso positivo, es necesario interrumpir la cadena de transmisión de COVID-19. Para esto se plantean pasos que buscan proteger al individuo y a sus contactos. Al mismo tiempo, intervenir el contagio y diseminación de la enfermedad en la población. Los pasos son:

- Detectar a las personas positivas de forma oportuna.
- Identificar a las personas con las que tuvo contacto el caso positivo.
- Darles seguimiento para aislarles lo antes posible y así evitar casos secundarios.

La estrategia integra dos aspectos: rastreo de contactos y seguimiento de casos a domicilio, apoyando así tanto la vigilancia como la atención primaria de salud (Organización Panamericana de la Salud, 2020). La meta es detectar nuevos casos de COVID-19 de manera oportuna y monitorearlos en sus domicilios; pero también identificar nuevos casos en los diferentes servicios del Ministerio de Salud Pública. Los datos recolectados se traducirán en información relevante y confidencial para la toma de decisiones y reorientación de acciones en la lucha contra el coronavirus (Ministerio de Salud Pública y Asistencia Social, 2020).

El plan piloto de la estrategia se realizó con la cooperación de la OPS/OMS, los Centros para el Control y la Prevención de Enfermedades y USAID Guatemala (Organización Panamericana de la Salud, 2020). Para el funcionamiento de los equipos de rastreadores en el plan piloto, la Dirección de Área Guatemala Central proporcionó coordinadores supervisores. Los Centros de Salud aportaron técnicos de laboratorio y auxiliares de enfermería para los campamentos de Bienestar Respiratorio de la Municipalidad de Guatemala. La comuna capitalina apoyó con infraestructura, ocho rastreadores y kit de medicamentos (Ministerio de Salud Pública y Asistencia Social, 2020). Ahora que la estrategia a crecido en alcance y se pretende abarcar no solo el área capitalina, si no la totalidad del territorio guatemalteco es necesario buscar alternativas para asegurar la captación de esta información.

Los usuarios finales del esfuerzo presentado en esta investigación son los rastreadores de casos y contactos de COVID-19. Los rastreadores son profesionales capacitados para dar seguimiento a los casos de COVID-19 y sus contactos, a fin de aislarles lo antes posible cuando desarrollan síntomas y así romper la cadena de transmisión. El trabajo que realiza el rastreador consiste en recolectar información sobre síntomas y estado de salud. El rastreo de casos y de contactos durante la pandemia es importante debido a que, con estas acciones se garantiza la atención a las personas con síntomas leves, se mantiene a las personas en vigilancia y si los casos se llegaran a complicar, se coordina el traslado a un centro asistencial (Ministerio de Salud Pública y Asistencia Social, 2020).

4.5. Aplicaciones móviles

4.5.1. ¿Qué es una aplicación móvil?

La definición de una aplicación móvil es un programa pensado desde su concepción para ser ejecutado en teléfonos inteligentes, tablets y otros dispositivos de carácter móvil; es decir, aparatos que pueden ser movidos de lugar y no están situados de manera estática. Este software permite al usuario que lo utiliza realizar actividades de carácter profesional, acceder a servicios, mantenerse informado, entre otro universo de posibilidades que aumenta al unísono con los avances tecnológicos.

Hace alrededor de dos décadas, la mayoría de los dispositivos móviles poseían capacidades limitadas. Existen precedentes de teléfonos inteligentes tempranos, pero estaban completamente enfocados en el correo electrónico o carecían de sofisticadas pantallas táctiles que pudieran usarse sin un lápiz óptico. Incluso menos tenían un navegador móvil decente capaz de mostrar algo más que un simple texto, enlaces y, en ocasiones, una imagen. Esto significaba que el público objetivo de estos aparatos, eran empresarios adictos al correo electrónico o entusiastas tecnológicos con la esperanza de que ese fuera el año del teléfono inteligente. Luego, Apple cambió todo con el lanzamiento del iPhone, y las expectativas para las experiencias móviles se restablecieron por completo (Charland, A et al., 2011).

La aplicación móvil es un segmento relativamente nuevo y de rápido desarrollo de la tecnología mundial de la información. Es fácil de usar, económica, descargable y ejecutable en la mayoría de los teléfonos móviles. Incluyendo teléfonos económicos y accesibles al público en general. Las aplicaciones tienen distintos usos dentro de las áreas de funcionamiento de un dispositivo móvil, como por ejemplo: llamadas, mensajería, navegación, chat, redes sociales, comunicación, audio, video, juegos, etc. Algunas están pre-instaladas en el teléfono y otras, usuarios pueden descargarlas desde Internet e instalarlas en el dispositivo móvil. Existe un gran mercado de aplicaciones móviles atendido por el número, en constante aumento, de desarrolladores, editores y proveedores de software móvil. Desde el punto de vista técnico, las aplicaciones móviles se pueden ejecutar en diferentes plataformas: iPhone, BlackBerry, Android, Symbian, Windows; también algunas máquinas virtuales como Java / J2ME, BREW, Flash Light, Silverlight. Según el área de aplicación, existen diferentes categorías de aplicación móvil (Islam, R et al., 2010):

- Comunicaciones: navegación por Internet, cliente de mensajería instantánea de correo electrónico, redes sociales.
- Juegos: rompecabezas/estrategia, cartas/casino, acción/aventuras
- Multimedia: visor de gráficos/imágenes, editor de presentaciones, reproductores de video y reproductores de audio
- Productividad: calendarios, calculadoras, agenda, bloc de notas/memo/procesadores de texto, hojas de cálculo.
- Viajes: guía de la ciudad, conversor de divisas, traductores, GPS/mapas, itinerarios/horarios, clima.
- Utilidades: administrador de perfiles, pantalla inactiva/protector de pantalla, libreta de direcciones, administrador de tareas, administrador de llamadas, manejador de archivos.

4.5.2. Aplicaciones móviles para COVID-19

Las aplicaciones móviles se usan satisfactoriamente en el tratamiento de enfermedades crónicas (Triantafyllidis A et al., 2019), pero la pandemia de COVID-19 ha empujado dichas soluciones. El desafío se afronta de cara, y existe la necesidad de reducir el riesgo de contaminación causada por el contacto cercano con un paciente (Smith AC et al., 2020). La tecnología móvil ha sido apalancada desde varios ángulos para controlar el contagio del SARS-CoV-2. Las aplicaciones móviles son accesibles, aceptables y de fácil adaptación, además tienen la habilidad de dar soporte a los esfuerzos por distanciamiento social.

Como tal, han sido ampliamente desarrolladas e implementadas durante la pandemia en un intento por ‘aplanar la curva’ del número en crecimiento de casos de COVID-19. Dotando de conocimiento

e información a civiles y autoridades mientras se trata de aliviar la presión a los sistemas de salud (Kondylakis H et al., 2020). Un análisis (Alanzi T., 2021) revela que varias aplicaciones han sido desarrolladas para diferentes funciones como seguimiento de contactos, construcción de conciencia, agenda de citas y consultas en línea. La mayoría de las aplicaciones están enfocadas en el seguimiento de contactos.

4.5.3. Aplicaciones nativas

Una aplicación móvil nativa es una aplicación de teléfono inteligente codificada en un lenguaje de programación específico, como Objective C para iOS o Java para sistemas operativos Android. Las aplicaciones móviles nativas brindan un rendimiento rápido y un alto grado de confiabilidad. También tienen acceso a varios controles de un teléfono, como su cámara y libreta de direcciones. Además, los usuarios pueden utilizar algunas aplicaciones sin conexión a Internet. Sin embargo, este tipo es costoso de desarrollar porque está vinculado a un tipo de sistema operativo. Lo que obliga a las empresas que crean aplicaciones a crear versiones duplicadas que funcionen en distintas plataformas.

Se identifican dos problemas con la línea de pensamiento de desarrollo nativo. Primero, crear una aplicación diferente para cada plataforma es muy costoso si no se planea utilizar un controlador en específico de alguna marca. En segundo lugar, el argumento de rendimiento de que las aplicaciones nativas son más rápidas puede aplicarse a juegos 3D o aplicaciones de procesamiento de imágenes, pero existe una penalización de rendimiento insignificante o imperceptible en una aplicación bien construida que utiliza tecnología de desarrollo multiplataforma (Charland, A et al., 2011).

4.5.4. Aplicaciones multiplataforma

Las aplicaciones móviles híbridas son aplicaciones que se instalan en un dispositivo, como cualquier otra aplicación. Lo que las diferencia es el hecho de que poseen elementos de aplicaciones nativas con elementos de aplicaciones web. Es decir, elementos desarrollados para una plataforma específica como iOS o Android, con sitios web que actúan como aplicaciones. Estos últimos no se instalan en un dispositivo, mejor dicho se acceden a través de Internet utilizando un navegador.

Las aplicaciones híbridas se implementan en un contenedor nativo que usa un objeto móvil para desplegar código web. Cuando se utiliza la aplicación, este objeto muestra contenido web gracias al uso de tecnologías como: CSS, JavaScript y HTML5. El contenido web puede mostrarse tan pronto como se abre la aplicación. Para acceder a las funciones de hardware de un dispositivo (acelerómetro, cámara, contactos, entre otras) es posible incluir elementos nativos de las interfaces de usuario de cada plataforma (iOS o Android). Se utiliza código nativo para acceder a las características específicas y así crear una experiencia de usuario aceptable. Las aplicaciones híbridas también pueden depender de plataformas que ofrecen API's de JavaScript si esas funcionalidades se llaman dentro de un contenedor web.

El acceso a características nativas a través de un API usando las herramientas de desarrollo basadas en web aporta una nueva dimensión al desarrollo multiplataforma. Las principales ventajas de este enfoque se encuentran principalmente en la unificación de los procesos de desarrollo para diferentes plataformas. Todo esto combinado con el tiempo ahorrado por los desarrolladores, afectando así directamente el costo total de desarrollo. Por otro lado, ninguno de estos middlewares proporciona un soporte completo y comprensivo para todos los sistemas operativos móviles. Las principales oportunidades que ofrece el desarrollo móvil multiplataforma se podrían resumir como (Amatya, S et al., 2014):

- Confiar en tecnologías web para fines de desarrollo.

- Menor tiempo de desarrollo, lo que reduce el costo del esfuerzo.
- Adecuado para prototipos rápidos y maquetas rápidas de alta fidelidad.
- Muy bueno con fines educativos para el desarrollo móvil.
- Combina la experiencia de usuario con un ciclo de desarrollo ágil y costos controlados.
- Las aplicaciones multiplataforma ofrecen una gran ventaja para los desarrolladores que desean actualizar su aplicación con frecuencia, ya que no es necesario volver a enviar la nueva versión si las modificaciones no han tocado el código nativo.
- La mayoría de las aplicaciones tienen una versión para iOS y una versión para Android. Por tanto, se desarrollan utilizando el lenguaje de programación correspondiente: Objective-C o Swift para iOS, Java para Android. Las aplicaciones híbridas permiten el uso de lenguajes de programación utilizados con frecuencia por los desarrolladores web (HTML, JavaScript y CSS) que pueden reutilizar sus conocimientos. Esto facilita la búsqueda de recursos para crear una aplicación híbrida.
- Reutilización del código del elemento web de la aplicación: el código se escribe una vez y se implementa en todas las plataformas móviles.

4.6. ¿Qué es un framework?

Es una maqueta o estructura conceptual de base tecnológica que permite al desarrollador organizar software, evitando posibles errores de programación gracias a la estructura previamente descrita por el framework. Se compone de un conglomerado de herramientas y módulos trabajando de manera conjunta que se pueden reutilizar y tener como punto de partida en varios proyectos.

Comúnmente, se utiliza un framework cuando se desea trabajar de una manera organizada, siguiendo las buenas prácticas de desarrollo, minimizando el riesgo de posibles errores, al mismo tiempo que se aumenta el nivel de productividad. Esto ahorra tiempo, esfuerzo y da dirección al inicio y durante todo el proceso de desarrollo. Pensando en el futuro del software, facilita la colaboración entre diferentes programadores que se encargarán de ampliar el alcance o entender el funcionamiento del software. Nos dota de la capacidad de utilizar herramientas de terceros que agregan capacidades o aplacan problemas específicos de la arquitectura del código. De esta manera, se evita la escritura de código repetitivo y permite enfocar esfuerzos en la funcionalidad y calidad de la aplicación.

4.6.1. React Native

En la conferencia React.js en 2015, Facebook presentó su nuevo marco de desarrollo React Native, uno que pensaron que revolucionaría la forma en que se crean las aplicaciones móviles. Cuando React Native fue lanzado, solo había soporte para iOS, pero desde entonces el soporte para Android se ha agregado y aún se está expandiendo. Facebook ha comenzado a ser más de código abierto y es el enfoque que han elegido para React Native. Aunque la fuente no es completamente abierta aún, Facebook intenta lograrlo y contempla que la comunidad contribuirá a mejorar el framework (Danielsson, W., 2016).

El propósito principal de React Native es simple; un desarrollador no debería requerir el conocimiento o necesitar gastar tiempo superfluo para crear una aplicación móvil, ya que al menos es necesario desarrollar dos aplicaciones para que sean compatibles con iOS y Android. Dado que las diferentes plataformas tienen diferentes apariencias, sensaciones y capacidades, no puede haber una aplicación que sea homogénea en todos los sistemas operativos. Sin embargo, dado que es la interfaz gráfica la que difiere, el desarrollo podría basarse en el mismo lenguaje pero tener los gráficos

renderizados de manera diferente, dependiendo de la plataforma destino y ser componentes nativos reales.

Facebook llama a este enfoque 'aprender una vez, escribir en cualquier lugar', que describe de qué se trata React Native. La tecnología de React Native se basa en React. Las ventajas de React se envían al framework y este se encarga de llevarlo a las aplicaciones nativas. En lugar de ejecutar React en el navegador y renderizar divs y textos, React Native se ejecuta en una instancia incrustada de JavaScriptCore (iOS) o V8 (Android) dentro de las aplicaciones y renderiza en componentes específicos de la plataforma de nivel superior (Danielsson, W., 2016). Los componentes de JavaScript se declaran mediante un conjunto de primitivas integradas compatibles con iOS o componentes de Android (Daniel W., 2015).

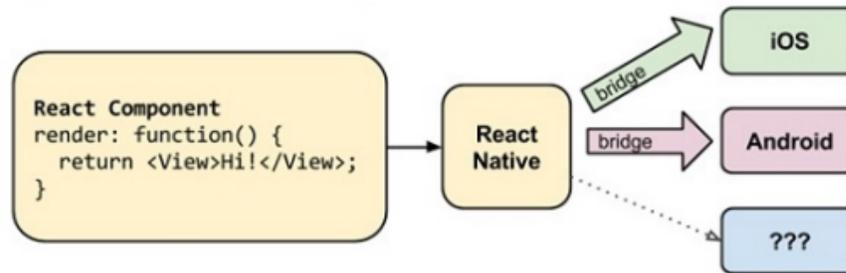


Figura 3: Renderizado en React Native

Ha habido muchos intentos de proporcionar a la comunidad de desarrollo un framework híbrido, pero ninguno tiene el conjunto completo de características que ofrece React Native. A diferencia de la mayoría de otros enfoques de desarrollo multiplataforma móvil, React Native no es una solución híbrida. A diferencia de Phonegap, Titanium e Ionic, React Native no se basa en el renderizado dentro de Web. React Native visualiza o intenta imitar HTML y CSS. El diseño se realiza en un hilo separado, lo que da como resultado que el hilo principal pueda realizar animaciones sin interrupciones. Desde una aplicación en React Native no se usan WebViews. Hace posible construir aplicaciones con una capacidad de respuesta nativa. Además, en lugar de enviar una base de código a todas las plataformas, una base de código compartida está escrita e implementada en las plataformas y solo algunas secciones como elementos gráficos y los componentes específicos de la plataforma se escriben por separado para cada plataforma de destino (React Native Documentation, 2021).

Como se indicó anteriormente, React Native es capaz de convertir sus componentes en componentes nativos reales. Vistas para Android o Vistas de UI para iOS. Esto es posible debido a la capa de abstracción conocida como el 'puente' que permite a React Native invocar las API de renderizado en Java para Android o Objective-C para iOS. Además, el framework revela la interfaz de JavaScript, lo que permite que la aplicación acceda a funciones específicas de la plataforma como la batería o la ubicación. Se han mencionado los diferentes subprocesos utilizados en React Native y en realidad hay tres principales hilos en los que se basa React Native: la cola de sombra, donde se maneja el diseño; el hilo principal, donde se realiza la representación de la interfaz de usuario; el hilo de JavaScript, donde las secuencias de comandos se están ejecutando. Estos hilos son responsables de manejar diferentes eventos a solicitud (Eisenman, B, 2015)

Existen varias características de React Native que lo hacen aún más atractivo para el desarrollo. En primer lugar, React Native tiene soporte para la ejecución asíncrona de operaciones entre el código JavaScript en la aplicación y la plataforma nativa, pero también permite subprocesos en los módulos nativos. Esto permite realizar muchas operaciones diferentes en segundo plano sin bloquear la interfaz de usuario. También permite al desarrollador depurar el código mientras ejecuta la aplicación, por ejemplo, utilizando las herramientas de desarrollo de Chrome. En segundo lugar,

React Native maneja la interacción de la pantalla implementando un sistema que maneja los 'toques' a pantalla en una vista compleja con características de alto nivel.

Dado que el reconocimiento de gestos es más avanzado en un dispositivo móvil que en la web, hay muchas acciones diferentes que pueden ser interpretadas por el tacto, como desplazarse, tocar y deslizar. Además, se puede realizar múltiples toques al mismo tiempo. Cuando los usuarios notan la diferencia entre una aplicación web y una nativa, uno de los puntos que más entorpecen la experiencia de usuario, es que cada toque en pantalla debe mostrar lo que sucederá cuando se lleve a cabo y el usuario debería poder cancelar la acción arrastrando el dedo. React Native ha resuelto este problema agregando una implementación abstracta `Touchable` y `TouchableHighlight` que entiende correctamente cuando se arrastra el dedo y otros elementos sin requerir mayor configuración (React Native Documentation, 2021).

Además, React Native usa JSX. Este es un marco de diseño para crear y dibujar componentes sin complicar la estructura o el código. En React, el estilo se especifica como un objeto y se agrega al elemento dentro la etiqueta HTML. Sin embargo, React Native utiliza un modelo de diseño de la web llamado Flexbox. Flexbox simplifica el proceso de construcción de diseños de interfaz de usuario comunes y su abstracción `StyleSheet` ofrece la posibilidad de declarar el diseño y estilo junto con el componente. El uso de Flexbox en la aplicación móvil para captura y centralización de datos en la estrategia de rastreo de casos y contactos de COVID-19 se puede ver en la Figura 4. Por último, React Native está inspirado en React, donde los componentes son los bloques de construcción. Un enfoque similar se aplica a React Native, ya que la aplicación está construida por módulos nativos y componentes de UI. React Native tiene soporte para las APIs de la plataforma común y los componentes nativos de la interfaz de usuario, pero también el soporte para crear módulos nativos y vistas si es necesario (Danielsson, W., 2016).

```

//Estilos de contactos
const styles = StyleSheet.create({
  notificationList: {
    marginTop: 20,
    padding: 10,
  },
  container2: {
    flex: 1,
    justifyContent: "center"
  },
  horizontal: {
    flexDirection: "row",
    justifyContent: "space-around",
    padding: 10
  },
  persona: {
    borderTopColor: 'black',
    borderBottomColor: 'black',
    borderBottomWidth: 1
  },
  mainView: {
    flexDirection: "row",
    backgroundColor: "white",
    borderRadius: 20,
    borderColor: "black",
    marginBottom: 10,
  },
  container: {
    flex: 1,
    backgroundColor: '#EBEBEB',
  },
  contacto: {
    margin: 20,
    borderRadius: 20,
    backgroundColor: 'black',
    justifyContent: 'center',
    textAlign: 'center',
    padding: 10
  },
  texto: {
    fontSize: 20,
    color: 'white',
    textAlign: 'center'
  },
});

```

Figura 4: Objeto JSON que contiene el estilo de un componente utilizando Flexbox

Para instalar React Native, es necesario instalar Homebrew y luego se usa para instalar Node.js. Al usar el administrador de paquetes de nodos (npm), que es un complemento de Node, el desarrollador puede instalar React Native con la línea de comando. Para ejecutar una aplicación de React Native en Android, Java JDK y Android SDK junto con Android SDK Build-Tools deben estar instalados en la computadora. Una vez instalado, se puede crear y ejecutar un nuevo proyecto con el framework de desarrollo utilizando el comando `react-native init "nombre-del-proyecto"` (React Native Documentation, 2021).

4.6.2. Almacenamiento asíncrono

El almacenamiento asíncrono es un sistema de almacenamiento basado en llave/valor. Se puede implementar fácilmente y está disponible globalmente para la aplicación. Este sistema de persistencia es simple y asíncrono, y también es una forma recomendada de almacenar datos.

```
try{
  userToken = data.id;
  await AsyncStorage.setItem('userToken', userToken);
} catch(e) {
  console.log(e);
}
```

Figura 5: llave/valor en memoria local de dispositivo móvil

```
async function fetchContactos() {
  setLoading(true);
  let userToken = await AsyncStorage.getItem('userToken');
  console.log(userToken);
}
```

Figura 6: leer un valor en memoria local de dispositivo móvil

4.7. Application Programming Interface

Un Application Programming Interface, desde ahora referido como API, provee una abstracción a un problema y especifica como los clientes deben interactuar con los componentes del software que implementan una solución a un problema. Los componentes están distribuidos por la librería del software permitiendo tener múltiples aplicaciones. El API permite reutilizar construcciones de bloques que permiten modular piezas de la funcionalidad que sean incorporadas en una aplicación de usuario final. Un API es una interfaz bien definida que provee un servicio a otras piezas del software. Su propósito es proveer una interfaz lógica a la funcionalidad de un componente mientras esconde detalles de su implementación (Reddy, M., 2011).

Un API permite que una aplicación se comuniquen con un servicio sin necesidad de conocer su implementación a detalle, solamente con conocimiento de los métodos de uso del API. Las API son una forma simplificada de proporcionar acceso a datos o secciones de infraestructura de una entidad sin el riesgo de proporcionar la lógica de negocio al que consume los servicios del API. Una API es útil cuando los usuarios necesitan acceder a datos en tiempo real o sus datos cambian o se actualizan con frecuencia.

Con el paso de los años, lo que es una API a menudo se ha descrito como cualquier tipo de interfaz de conectividad genérica para una aplicación. Más recientemente, sin embargo, la API moderna ha adquirido algunas características que las hacen extraordinariamente valiosas y útiles:

- Las API modernas se adhieren a los estándares (generalmente HTTP y REST), que son amigables para los desarrolladores, de fácil acceso y ampliamente comprensibles.
- Se tratan más como productos que como código. Están diseñados para el consumo de audiencias específicas (en nuestro caso, para desarrollo móvil), están documentadas y están versionados

de manera que los usuarios puedan tener ciertas expectativas de su mantenimiento y ciclo de vida.

- Debido a que están mucho más estandarizados, tienen una disciplina mucho más sólida para la seguridad y la gobernanza, además de ser monitoreados y administrados para el rendimiento y la escala.
- Como cualquier otra pieza de software producido, la API moderna tiene su propio ciclo de vida de desarrollo de software (SDLC) de diseño, prueba, construcción, administración y control de versiones. Además, las API modernas están bien documentadas para su consumo y control de versiones.

5.1. Descripción del proyecto

En el proyecto se utilizó la metodología ágil scrum, en la cual cada sprint tuvo una duración de 1 semana. Se usa la herramienta de projects habilitada para organizaciones en GitHub para llevar el control ágil del proyecto. Se realizó, al menos, una reunión con periodicidad mensual con colaboradores del MSPAS para presentar avances de la investigación, agregando reuniones cuando el desarrollo así lo requirió. En las reuniones con el MSPAS, se aspiró a obtener retroalimentación sobre la interfaz gráfica, capturar necesidades de la aplicación y validar que el desarrollo vaya acorde con los requerimientos del Ministerio. Se empleó design thinking para el diseño de las páginas de la aplicación. Apoyados en la elaboración de mock-ups en papel y lápiz para definir la navegación de la aplicación y posicionamiento general de los componentes. Escoger qué componentes visuales son compatibles con la estructura de los datos consumida desde el API; por ejemplo, listas para presentar en pantalla JSON. Esto con la intención de identificar insights o “hallazgos” necesarios para realizar con éxito el desarrollo y seguir un hilo conductor. Se someterá a pruebas de rendimiento, descargando y leyendo casos y contactos cuantificando el tiempo que se toma en guardarlos en memoria local del dispositivo y desplegándolos en pantalla. Esto para obtener una idea del comportamiento de la aplicación ante estas situaciones. Debido al contexto en donde será utilizada la plataforma se exponen dos requerimientos a cumplir para concluir de manera satisfactoria con el desarrollo. La comunicación entre la aplicación móvil y Go.Data descrita por un protocolo de interoperabilidad y el almacenamiento en memoria local de la información ingresada por un rastreador para su posterior carga a Go.Data, respectivamente.

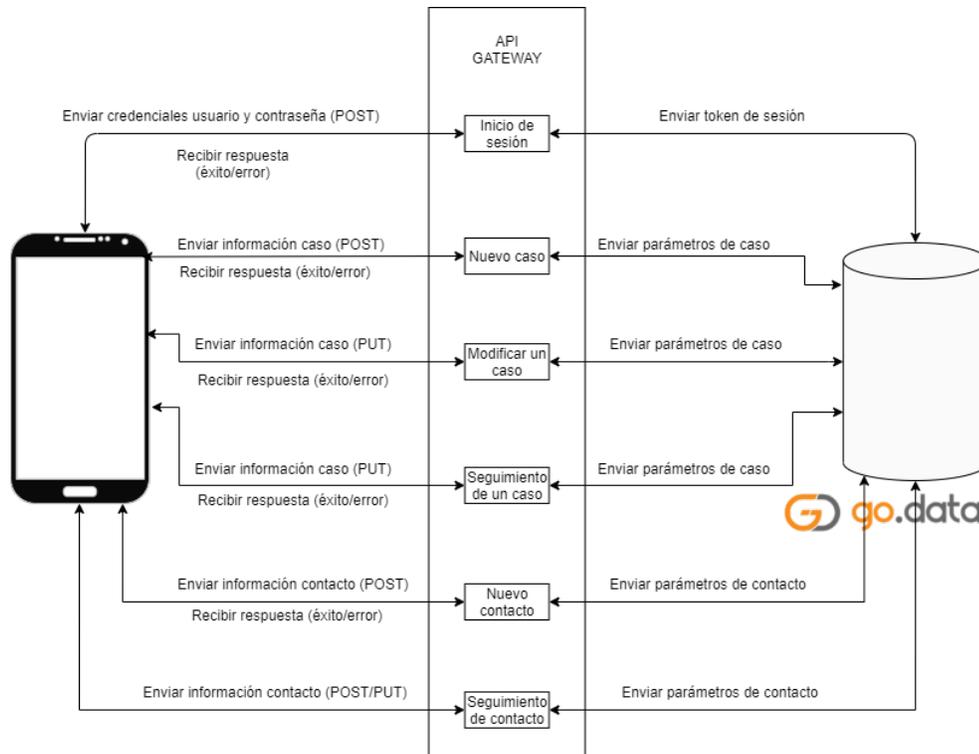


Figura 7: Comunicación entre aplicación nativa y Go.Data

Se debe tener un usuario en Go.Data con permiso de publicar casos y contactos. Esto se solicitó al administrador de Go.Data Guatemala al igual que el ID del brote de pruebas. La aplicación deberá ser capaz de iniciar sesión con credenciales previamente autorizadas para rastreadores mediante la comunicación con la API. Posteriormente, el sistema deberá dar la posibilidad de realizar un conjunto de operaciones sobre un caso o un contacto. La aplicación contará con cuatro apartados principales:

- Descarga: Este apartado será representado por un botón; los rastreadores deberán presionarlo antes de comenzar cualquier captura de datos. Esta información será enviada mediante comunicación con el API, la cual retornará un token de sesión. Este token de sesión se incluye entre los parámetros que construyen el url para la descarga de los casos y contactos de COVID-19 registrados en Go.Data. Información que se usará después en toda la aplicación y con la cual se harán consultas.
- Carga: representado por un botón, cumplirá la función de centralizar la información capturada en el dispositivo por un rastreador. En cualquier momento posterior a la creación o modificación de un caso o contacto, se podrá hacer uso de la función de carga. Dando como resultado de éxito la centralización de la data capturada.
- Casos: La visualización de los casos descargados. Se presentará de manera general e individual. En la vista general, se posibilitará el filtrado por medio de atributos. En este filtrado, cada uno de los casos será una entrada. La cual al presionar redirigirá a la vista individual del caso. Aquí, será posible utilizar la comunicación con el API para realizar un seguimiento a un caso o modificarlo.
- Contactos: La visualización de los contactos descargados. También en ámbito general e individual. Cada ítem corresponderá a una persona que tuvo exposición a un caso confirmado. A través de la comunicación con el sistema de información Go.Data será posible realizar un nuevo seguimiento a un caso desde la vista individual.

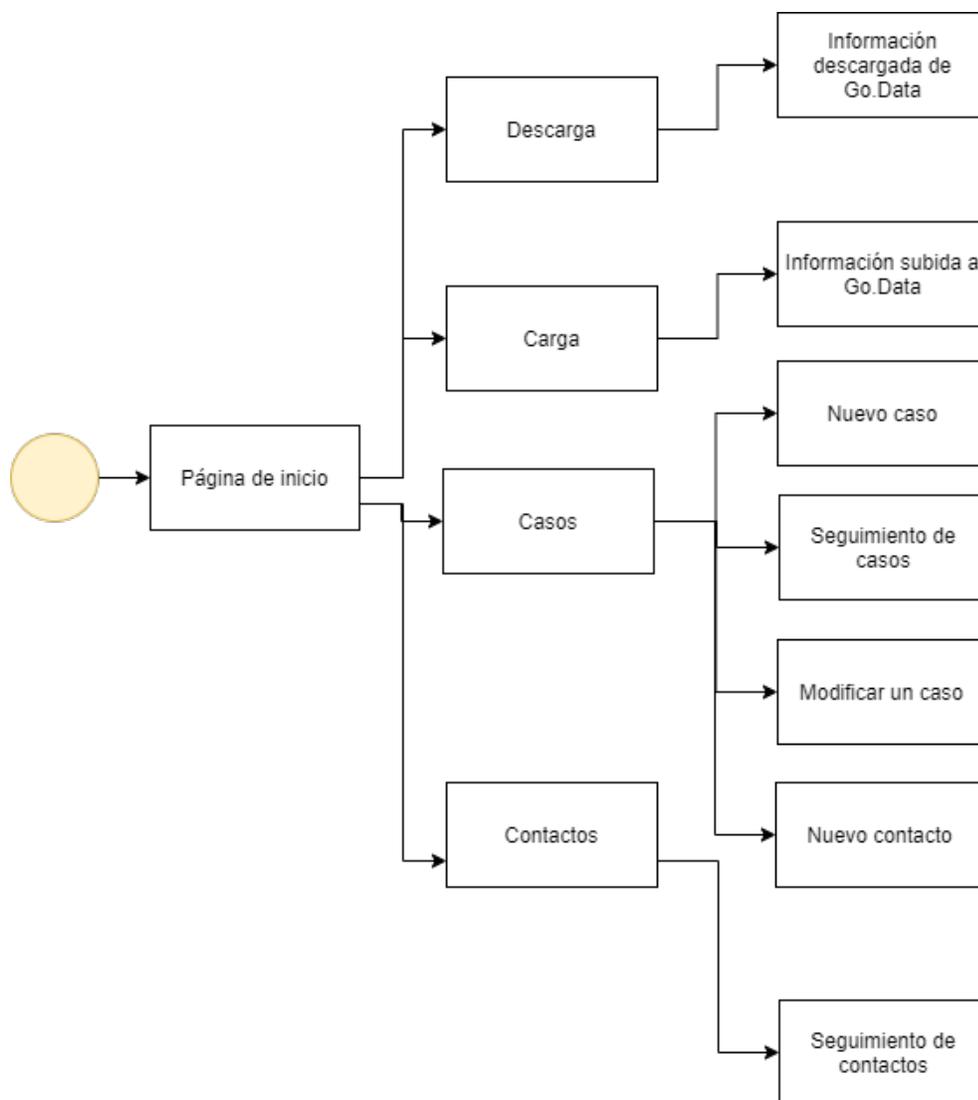


Figura 8: Funcionalidades de la aplicación

5.2. Análisis de estado actual

Se inicia el presente trabajo investigando sobre los procedimientos actuales con los cuales el MSPAS capta y centraliza los datos de casos y contactos de COVID-19. El plan piloto contempla el rastreo de casos y contactos, basándose en la implementación escalonada de la modalidad tipo call center; los rastreadores realizarán llamadas recolectando información para propósitos únicos de salud pública desde el número 2300-4400 que funcionará a partir del día jueves 13 de agosto de 2020, las llamadas se complementarán con visitas domiciliarias según las necesidades identificadas (Ministerio de Salud Pública y Asistencia Social, 2020). Esto para recolectar la información pertinente de cada caso, e identificar quiénes son las personas con las que un individuo positivo ha tenido contacto, a manera de rastrear casos y contactos de COVID-19. Esta información es digitada para su traslado al ente centralizador, la plataforma Go.Data. Es aquí donde toda la información debe almacenarse y desde donde es consultada para los distintos casos de uso del MSPAS, como por ejemplo el tablero nacional. Dentro de la plataforma, se hace una división de los casos y contactos por brote. Este

determina la información que será consumida y almacenada en Go.Data.

Como se explica anteriormente, para la aplicación móvil fue necesaria la utilización de un brote de prueba que contuviera una cantidad realista de casos y contactos, que permitiera reflejar el comportamiento de la aplicación. El identificador de dicho brote de prueba es f1fb1ec0-3107-407a-8a14-c4c0bf927f8e, el cual cuenta con 4,128 casos relacionado a salud pública.

5.3. Design Thinking

Se utilizó un ciclo de Design Thinking dividido en en 5 pasos:

- Empatizar: Reunir información de los usuarios.
- Definir: Transformar información de usuario en *insights*.
- Idear: Generar ideas basadas en los *insights*.
- Prototipar: Manifestar las ideas en posibles entregables.
- Evaluar: Poner a prueba los prototipos en presentaciones. Obtener el feedback sobre los puntos positivos y cambio a efectuar en la aplicación. Iterar.

Como primera parte del proceso para generar las ideas que dieron forma al entregable final fue necesario empaparse de la mayor cantidad de información posible sobre los usuarios finales. Se llevaron a cabo reuniones virtuales con colaboradores del Ministerio, con el fin último de conocer los procedimientos actuales y actores que intervendrán en el funcionamiento de la aplicación. Las reuniones se llevaron a cabo con epidemiólogos y doctores del ministerio, quienes fungieron como *stakeholders*. Con esto fue posible tener un grado de comprensión de las necesidades que tienen en el MSPAS y el alcance real del desarrollo. A manera de ordenar la información de los usuarios, se utilizó como ayuda un diagrama de empatía. Este consiste en escoger un actor, una acción, un impedimento y el sentir del actor frente al impedimento. Dicho diagrama representa las acciones de los actores en el proceso actual llevado a cabo por el MSPAS (Figura 31 - Figura 36).

A continuación, se realizó un proceso de depuración para abstraer la información vital obtenida. Pasando a la fase dos del proceso de *design thinking*. Se hizo una priorización de las necesidades, buscando plantear las soluciones dentro del alcance del proyecto. Los puntos identificados como cruciales de la abstracción de información sirvieron como guía a lo largo de todo el desarrollo. Para esta fase del proyecto, los diagramas de empatía se transformaron en casos de uso. Detallando los posibles escenarios para el software (Figura 37 - Figura 40) Una vez se identifican los requerimientos, se pasa a una fase de ideación, en donde esquematizamos mentalmente como será el flujo de navegación y los diferentes componentes que deben conformar la aplicación móvil.

Posteriormente, se procede a la fase de prototipado. Es aquí donde finalmente se plasmó con lápiz y papel, los diferentes componentes y vistas del desarrollo. Este prototipado se llevó a cabo teniendo en cuenta las estructuras de datos que se habían compartido e identificado en los pasos anteriores del proceso de Design Thinking. La estructura y los controles que se usarían para pasar de una página a otra, realizar una acción, presentar la información consumida y el ingreso de información fueron definidos en esta etapa del proceso. Por último, dicho prototipo fue puesto a prueba con el colaborados de Go.Data implicado en la solución que se desarrolló en esta investigación. Esto nos ayudó a determinar mejoras significativas, fallos a resolver y posibles carencias- Durante esta fase evolucionamos nuestra idea hasta convertirla en la solución que estábamos buscando.

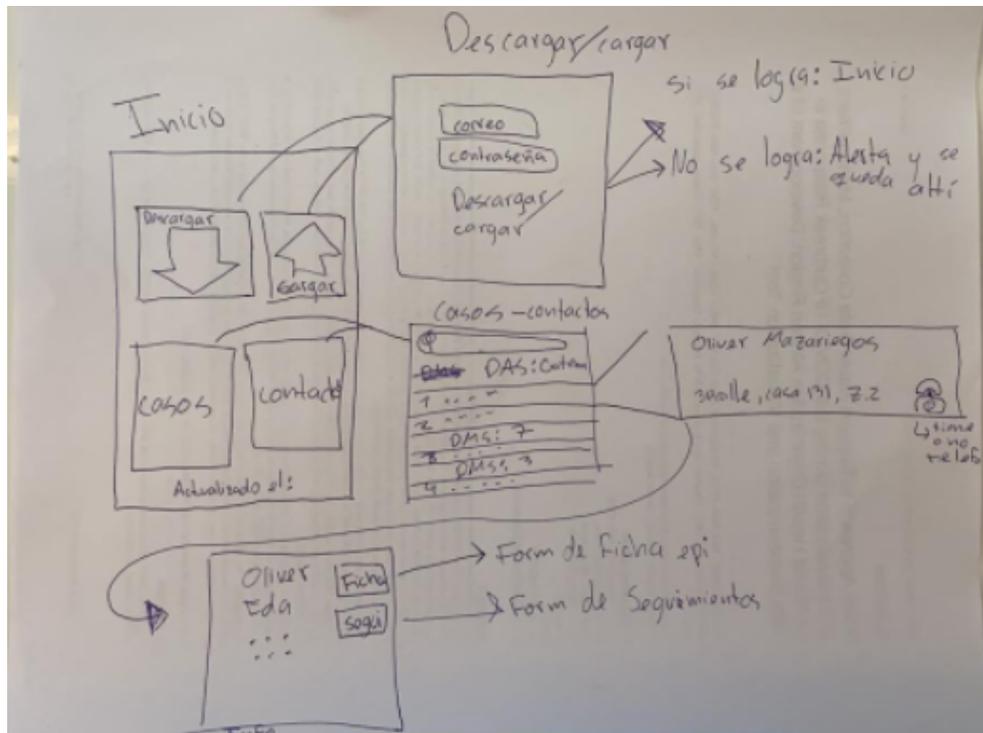


Figura 9: Prototipo a mano de la aplicación móvil

5.4. Decisión de herramientas

Se escogió utilizar un acercamiento de aplicación móvil híbrida o multiplataforma. esto debido a que nos ahorra esfuerzos y resulta ser menos costoso. Nos permite desarrollar software de manera ágil sin sacrificar calidad y aprovechando los conocimientos previos con los lenguajes de programación, acortando la curva de aprendizaje de una nueva tecnología y tiempos. Este acercamiento, nos permite cumplir con las necesidades del MSPAS, ya que la solución se podrá utilizar tanto en dispositivos Apple como Android. Permitiendo que la aplicación móvil sea escalable y habilitando la posibilidad de utilizarla en la solución de futuros casos de uso.

Se llevó a cabo una investigación bibliográfica para descubrir qué herramientas se podrían utilizar para programar la solución con un framework de desarrollo híbrido. Se encuentran múltiples opciones como lo son: Flutter, Cordova, Ionic, Xamarin, Unity, NativeScript, PhoneGap, Kotlin Multiplatform, Apache Flex, Kendo UI y Kivy. Sin embargo, se toma la decisión de utilizar React Native como entorno de desarrollo. Se debe a que, a diferencia de los anteriormente mencionados, React Native no se basa en el renderizado dentro de web. Si no, que tienen su propia estructura por medio de puentes, que permiten tener una base de código compartida entre muchas plataformas, y son únicamente pedazos nativos los que se escriben en el lenguaje de plataforma destino. Además, tiene su funcionamiento repartido en diferentes hilos de ejecución. Quiere decir que, mientras uno realiza las funciones principales de la aplicación, otro se encarga del diseño. Lo que posibilita hacer uso de funciones asíncronas y realizar animaciones de manera fluida y sin interrupciones. React Native posee una serie de características que concuerdan con los objetivos de desarrollo del presente esfuerzo y la hacen la mejor opción para programar la aplicación móvil.

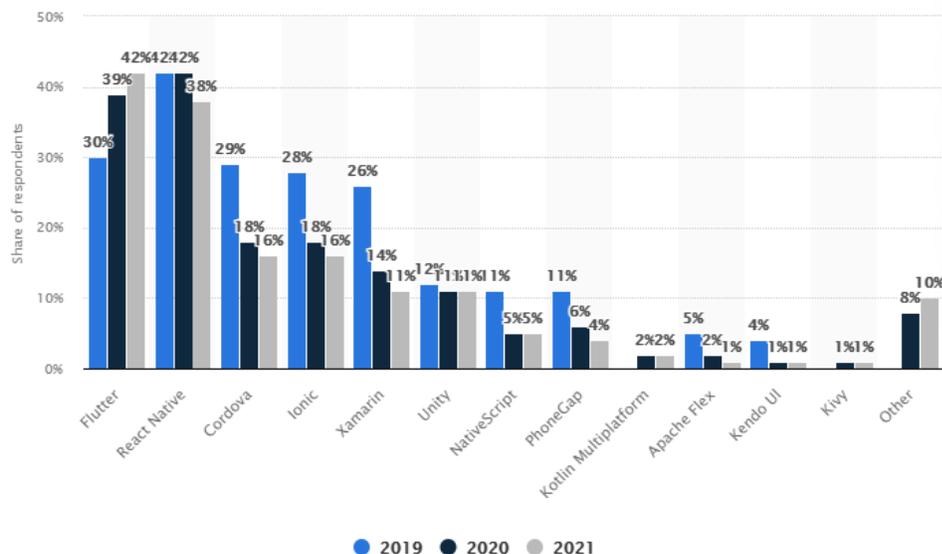


Figura 10: Frameworks de desarrollo móvil a escala mundial (Liu, S., 2021)

Podemos observar que React Native ha sido la decisión de más de 1/3 de los desarrolladores de aplicaciones móviles durante 2019 y 2020, siendo la más alta en estos años. Superada únicamente por Flutter en el año 2021. Esto nos da un indicador de que la comunidad y recursos accesibles sobre esta herramienta estarán en su mayoría documentados. Esta también es una razón de peso para la elección de este entorno de desarrollo.

5.5. Programación del proyecto

5.5.1. Log-in y autenticación

Se inicia el desarrollo colocando dos cajas de texto, en las cuales el usuario deberá colocar su usuario y contraseña. Estos dos datos se almacenan en variables para su uso posterior en la llamada al API. Esta información posteriormente es mandada a Go.Data para consultar si el usuario existe y cual es su nivel de acceso en la plataforma.

No se contempla el registro de usuarios, ya que el proceso de discernir quienes pueden entrar al sistema centralizado y modificarlo es un proceso que realiza de manera interna el Ministerio. Se utiliza una llamada asíncrona utilizando la librería axios en versión 0.21.1 o superior, haciendo uso de "GET", con los parámetros extraídos de las dos cajas de texto para la validación. Se crea un botón el cual dispara esta llamada asíncrona y maneja los errores en caso el API tenga un resultado no satisfactorio.

La autenticación nos sirve para que ninguna persona ajena al MSPAS pueda ingresar a ver la información sensible relacionada con la salud de ciudadanos guatemaltecos. Si el proceso resulta satisfactorio, se utiliza el almacenamiento asíncrono para guardar con carácter global el token de

sesión. Este token de sesión tiene un tiempo de expiración configurable en Go.Data de 15 minutos por inactividad. Por lo tanto, si después del tiempo estipulado se decide realizar alguna consulta o carga de datos, la aplicación automáticamente se redirige al inicio de sesión, solicitando la autenticación nuevamente.

```
{
  "id": "GJh8X01gXKsvNxm7NRAv6xwG80xqDwzYC6Pmtu0sEo8RS5p6XNuUoJNofh5vzozt",
  "ttl": 600,
  "created": "2021-10-29T09:23:18.896Z",
  "userId": "bdc20306-902c-4a44-87b4-80311dfc3b52",
  "createdAt": "2021-10-29T09:23:18.897Z",
  "createdBy": "system",
  "updatedAt": "2021-10-29T09:23:18.898Z",
  "updatedBy": "system",
  "deleted": false
}
```

Figura 11: JSON de una respuesta satisfactoria del API con un log-in exitoso

5.5.2. Menú principal

Se diseñó un menú principal con formato de mosaico, utilizando diferentes colores para diferenciar los distintos procesos posibles dentro de la aplicación móvil. Consta de 4 botones principales, que son la primera vista que tiene el usuario, una vez ingresa de manera satisfactoria sus credenciales de ingreso. Para la programación del mosaico de menú principal se utilizó el componente de botones nativo que nos proporciona React Native. Consta de 4 botones ordenados de dos en dos correspondientes a:

- Descarga
- Carga
- Casos
- Contactos



Figura 12: Menú principal de la aplicación.

Cada botón redirige hacia una nueva vista que contiene las especificaciones que cada una de las funcionalidades presentadas. Se utilizó la dependencia `@react-navigation/native` en una versión 6.0.1 o superior. Esta dependencia nos permite organizar la navegación de pantallas, y agregar la funcionalidad a los botones del menú principal para redirigir a sus respectivas vistas. Se realizó una función sobre cada uno de los botones para poder navegar por las páginas.

En esta sección se debieron realizar decisiones importantes, ya que existe más de una dependencia que maneja el routing de una aplicación de React Native. También, se encontró con el inconveniente que librerías auxiliares tenían conflictos de versionamiento. Por lo que se tuvo que actualizar la dependencia `react-navigation/stack` a una versión 6.0.6 o superior para solucionar el problema. Se utilizaron funciones descritas dentro de los eventos `onPress` de los componentes `TouchableOpacity`, que son componentes por defecto que vienen con la librería de `react-native`.

Estos componentes existen dentro de la función que representa toda la vista, `MainMenuScreen`, la cual recibe como parámetro la palabra reservada `navigation`. Esto para poder utilizar dentro de los componentes el ruteo hacia diferentes vistas. Dentro del retorno de la función, se definen específicamente los componentes que construyen la vista del menú principal. La cual consiste de: Un componente `Image`, correspondiente a la imagen de Go.Data y cuatro `TouchableOpacitys` que son las 4 funcionalidades de la aplicación. Cada uno de estos elementos tiene su propio estilo aplicado. Esto se define dentro de un objeto `StyleSheet`, donde se nombran los estilos. Estos estilos nombrados son después referenciados en el atributo `Style` en cada componente respectivamente.

5.5.3. Lectura de casos y contactos

En esta etapa del desarrollo se logró consumir información de la plataforma centralizadora de datos mediante su API. Fue necesaria la utilización de una llamada asíncronica utilizando la librería `axios` versión 0.21.1 o superior, haciendo uso de "GET". En esta etapa se creó una vista de React Native en la cual se genera una `FlatList` conteniendo todos los casos y contactos encontrados en Go.Data, respectivamente. Esta información es leída y renderizada en la vista, y es posible hacer scroll

para buscar el caso en particular de interés. Dentro de la FlatList se programa un TouchableOpacity. Esto para lograr que al presionar un caso o contacto en específico, se habilite la posibilidad de redireccionarlo a una nueva página que contenga el detalle del caso o contacto. Se posiciona un botón en la parte superior destinado a redirigir a una nueva vista en donde se agregará un nuevo caso o un nuevo contacto, dependiendo en que vista estemos. Cada entrada de la lista para los casos y contactos es un botón. Este botón redirecciona a una nueva vista que presenta información detallada del caso o contacto. Para los casos, dentro de esta vista detallada existe un botón más para acceder a la información de la ficha epidemiológica.

```
{
  "ID": "5ba5da13-f074-4fe4-bea4-6291e0a55cad",
  "Identificador": "RCC-2020-00005",
  "Fecha de notificación": "2020-10-11T00:00:00.000Z",
  "Fecha de notificación Aproximada": false,
  "Creado En": "2020-10-30T21:21:44.445Z",
  "Creado Por": "0201861f-a357-4906-bfcf-fe6228aa8826",
  "Actualizado En": "2020-10-30T21:21:44.445Z",
  "Actualizado Por": "0201861f-a357-4906-bfcf-fe6228aa8826",
  "Eliminado": false,
  "Eliminado En": null,
  "Created on": "API",
  "Primer Nombre": "EDINSON OBDULIO CORADO RUIZ",
  "Segundo Nombre": null,
  "Apellido": null,
  "Sexo": "Masculino",
  "Ocupación": "",
  "Edad": {
    "Años De Edad": 11,
    "Edad / Meses": null
  },
  "Fecha Nacimiento": null,
  "Clasificación": "No Es Un Caso (Descartado)",
  "Era un contacto": false,
  "Fecha de convertirse en caso": null,
  "Era un caso": null,
  "Fecha de convertirse en contacto": null,
  "Fecha de La Infección": null,
  "Fecha de inicio de síntomas": "2020-10-10T00:00:00.000Z",
  "Nivel De Riesgo": "",
  "Razón De Riesgo": null,
  "Condición del Paciente": "0",
  "Fecha de condición": "2020-10-11T00:00:00.000Z",
  "Documentos": null,
  "Tipo": "Caso",
  "Fechas": null,
  "SE NEGÓ A SER TRANSFERIDO A TRATAMIENTO": false,
  "Direcciones": [
    {
      "Tipo De Dirección": "Direccion Actual",
      "País": null,
      "Comunidad, aldea o zona": null,
      "Dirección Línea 1": "ALDEA BOLONCO",
      "Código Postal": null,
      "Ubicación": "DMS Fray Bartolome De Las Casas ",
      "Ubicación ID": "220161558",
      "Ubicación Identificadores": [
        "Fray "
      ],
      "Ubicación Ubicación principal": [
        "Guatemala",
        "Alta Verapaz",
        "DAS Alta Verapaz",
        "Fray Bartolomé de Las Casas",
        "DMS Fray Bartolome De Las Casas "
      ],
      "Geolocalización": null,
      "Geolocalización Precisa": false,
      "Dirección válida hasta la fecha": "2020-10-11T00:00:00.000Z",
      "Número De Teléfono": null,
      "Email address": null
    }
  ],
  "Entierro seguro realizado": false,
  "Fecha De Entierro": null,
  "Fecha de inicio de síntomas aproximada": null,
  "Lugar de entierro": null,
  "Lugar de entierro ID": null,
  "Lugar de entierro Identificadores": null,
  "Lugar de entierro Ubicación principal": null,
  "Nombre Lugar De Entierro": null,
  "Vacunas Recibidas": null,
  "Estado De Embarazo": "No aplica",
}
```

Figura 13: Captura parcial de JSON respuesta a lectura exitosa de caso

```

{
  "ID": "79308ac3-995b-47bc-abce-36d28d523bef",
  "Fecha de notificación": "2020-12-03T00:00:00.000Z",
  "La fecha del informe es aproximada": false,
  "Creado En": "2020-12-04T01:33:02.178Z",
  "Creado Por": "d2e66d66-304a-4391-b585-4980a5443ce8",
  "Actualizado En": "2020-12-04T01:35:25.671Z",
  "Actualizado Por": "b2585d7d-9e3b-479c-8b62-d71c0cb29472",
  "Eliminado": false,
  "Eliminado En": null,
  "Created on": "API",
  "Primer Nombre": "ContactoNombre",
  "Segundo Nombre": null,
  "Apellido": "ContactoApellido",
  "Sexo": "Femenino",
  "Ocupación": "",
  "Edad": {
    "Edad (años)": 20,
    "Edad (Meses)": 0
  },
  "Fecha Nacimiento": null,
  "Clasificación": null,
  "Era un contacto": null,
  "Fecha de convertirse en caso": null,
  "Era un caso": false,
  "Fecha de convertirse en contacto": null,
  "Fecha De La Infección": null,
  "Fecha de inicio de síntomas": null,
  "Nivel De Riesgo": "",
  "Razón": null,
  "Condición del Paciente": "",
  "Fecha de resultado": null,
  "Documentos": [],
  "Tipo": "Contacto",
  "Fechas": null,
  "SE NEGÓ A SER TRANSFERIDO A TRATAMIENTO": null,
  "Direcciones": [
    {
      "Tipo De Dirección": "Direccion Actual",
      "País": null,
      "Comunidad, aldea o zona": null,
      "Dirección Línea 1": null,
      "Código Postal": null,
      "Ubicación": "DAS Guatemala Central",
      "Ubicación ID": "278",
      "Ubicación Identificadores": [
        "So"
      ],
      "Ubicación Ubicación principal": [
        "Guatemala",
        "Guatemala",
        "DAS Guatemala Central"
      ],
      "Geolocalización": {
        "Latitud": 14.6349149,
        "Longitud": -90.5068824
      },
      "Geolocalización Precisa": false,
      "Dirección válida hasta la fecha": "2020-12-03T00:00:00.000Z",
      "Número De Teléfono": null,
      "Email address": null
    }
  ],
  "Entierro seguro realizado": false,
  "Fecha De Entierro": null,
  "Seguimiento": {
    "Inicio original del seguimiento": "2020-12-03T00:00:00.000Z",
    "Inicio de seguimiento": "2020-12-03T00:00:00.000Z",
    "Final Del Seguimiento": "2020-12-16T00:00:00.000Z",
    "Estado final del seguimiento": "Bajo Seguimiento"
  },
  "Seguimiento asignado al equipo": null,
  "Fecha de último contacto": "2020-12-02T00:00:00.000Z",
  "Vacunas Recibidas": [],
  "El Estado De Embarazo": "",
  "Questionnaire answers": null,
  "Relación": {
    "Personas Persona De Referencia": "00e59975-673b-4d8b-93a8-41e8ac0e911b",
    "Fecha Del Último Contacto": "2020-12-02T00:00:00.000Z",
    "Se Calcula La Fecha De Contacto?": false,
    "Nivel de riesgo": "3 - Alta",
    "Tipo De Exposición": "",
    "-": "",
    "Relación Detallada": null,
    "Cluster": null,
    "Comentario": null,
    "ID": "a93b2efc-1a86-4544-b2e1-934a6fe20f66",
    "Creado En": "2020-12-04T01:33:02.725Z",
    "Creado Por": "d2e66d66-304a-4391-b585-4980a5443ce8",
    "Actualizado En": "2020-12-04T01:33:02.725Z",
    "Actualizado Por": "d2e66d66-304a-4391-b585-4980a5443ce8",
    "Eliminado": false,
    "Eliminado En": null,
    "Created on": "API"
  }
}

```

Figura 14: JSON respuesta a lectura exitosa de contacto

5.5.4. Carga de información

Como siguiente punto en la programación, se codificó la funcionalidad de cargar un nuevo caso, un nuevo contacto o un seguimiento a un contacto al repositorio centralizador. Este proceso guarda similitud con los procesos anteriores. Se utiliza una llamada asincrónica con la librería axios en versión 0.21.1 o superior, pero el parámetro utilizado en esta llamada es "PUT". Se deben instalar las dependencias @react-native-picker/picker versión 2.1.0 o superior y @djetime/react-native-date-picker versión 1.1.2 o superior. Estas librerías nos servirán para habilitar los controles de DropDown y selector de fecha. Su función es mantener el ingreso de datos por parte de los rastreadores en preguntas con respuesta cerrada. Exceptuando aquellas que requieran el ingreso de información por parte del usuario, como por ejemplo, la dirección. Se posicionan alternando los controles de DropDown, selector de fecha y cuadro de texto para buscar captar toda la información necesaria para crear un nuevo caso, contacto o seguimiento.

Dentro del apartado de casos existe una sección que corresponde a las respuestas de un cuestionario; este cuestionario recibe el nombre de "ficha epidemiológica", cuyas preguntas deben estar presentes al crear un nuevo caso, para que este tenga un resultado de carga satisfactorio. Por lo que se debió realizar una labor de parseo de tipo de datos. También, existe un diccionario de variables, el cual, mapea una respuesta de texto a un número. Se realizó un esfuerzo para que al usuario final se le presentará la respuesta en formato de texto, y después internamente se interpretará la respuesta como su homólogo numérico. Sería este último el que se mandará como parámetro en el JSON de carga para la llamada PUT.

De manera muy similar a cómo se encuentra esta peculiaridad en los casos, existe una para los contactos. A cada contacto se le puede hacer un seguimiento, que consiste en la respuesta de un número finito de preguntas de respuesta cerrada. Se crea una vista accesible desde la página de contactos general, en la cuál es posible llenar esta encuesta y cargarla posteriormente a Go.Data.

```

"Respuestas Al Cuestionario": {
  "Case_whichForm": [
    {
      "value": "Ficha Epidemiológica 1"
    }
  ],
  "FE103no_de_ficha_de_notificacion": [
    {
      "value": "COVID17120112020100054"
    }
  ],
  "FE111grupo_cultural": [
    {
      "value": "1"
    }
  ],
  "FE112escolaridad": [
    {
      "value": "2"
    }
  ],
  "FE113enfermedades_asociadas": [
    {
      "value": []
    }
  ],
  "FE114síntomas": [
    {
      "value": [
        "3",
        "5",
        "6",
        "7",
        "8",
        "9",
        "11",
        "12"
      ]
    }
  ],
  "FE124tipo_de_vigilancia": [
    {
      "value": [
        "1"
      ]
    }
  ],
  "FE121se_tomo_una_muestra_respiratoria": [
    {
      "value": "1"
    }
  ],
  "FE12102fecha_y_hora_de_toma_de_la_muestra": [
    {
      "value": "2020-10-10T00:00:00.000Z"
    }
  ],
  "FE12101tipo_de_muestra": [
    {
      "value": [
        "2"
      ]
    }
  ]
}

```

Figura 15: Información de ficha epidemiológica referente a caso

5.5.5. Almacenamiento asincrónico

Para lograr la funcionalidad de descarga de datos cuando se tenga internet y carga de datos con los cambios realizados durante todo el día, fue necesario expandir el alcance del almacenamiento asincrónico. Se instaló la API nativa de @react-native-async-storage/async-storage versión 1.15.5 o superior. Una vez instalada esta librería, fue posible almacenar los resultados de los casos en la memoria local del dispositivo. Esto haciendo click sobre el botón "Descarga" del menú principal. Esto significa que el proceso se debe hacer cada vez que se desee actualizar la vista de casos y contactos. Los botones correspondientes a casos y contactos leen la información de la memoria local del dispositivo. Para el requerimiento de poder realizar cambios sin tener conexión a internet, y más adelante cargarlos a Go.Data cuando se cuente con internet, fue necesario la implementación de una clase stack. Se programó un stack que alberga todas las solicitudes "PUTz de carga en general de la aplicación. Al momento de presionar el botón "Carga" del menú principal, se realiza un pop de solicitudes, sacando una por una las request del stack y ejecutándolas si se tienen conexión a internet. De esta manera, actualizando y cargando la información a la plataforma centralizadora.

Para la validación del proyecto se realizó una reunión con stakeholders (colaboradores del MSPAS encargados de coordinar la estrategia de casos y contactos de COVID-19 en Guatemala) en la cual ellos probaron de primera mano la aplicación, y dieron visto bueno al desarrollo de la misma. Se tuvo una aceptación a la investigación realizada y cabe destacar que se cumplen los requerimientos identificados en la fase de design thinking. Los colaboradores del MSPAS mostraron interés en la forma en que se presentó la información, por medio de niveles. Primero teniendo una presentación general de los resultados, y al presionar los botones correspondientes, se accedía a un nivel más profundo de detalle. En la reunión está presente una persona que trabaja de cerca con el sistema centralizador Go.Data a nivel regional y se llega a la conclusión de que Guatemala está siendo pionera al poder hacer una carga de casos, contactos y seguimientos desde una aplicación móvil, ya que este comportamiento es único en la región.

La aplicación en su versión actual realiza las siguientes acciones:

- Ingreso a la aplicación por medio de usuario y contraseña previamente aceptados por los criterios del MSPAS.
- Manejo del tiempo de vida del token de sesión de un usuario.
- Ver información general y detallada sobre un caso registrado en el repositorio centralizador.
- Ver información general y detallada sobre un contacto registrado en el repositorio centralizador.
- Creación de un nuevo caso y captura de información con respuesta cerrada sobre ficha epidemiológica.
- Creación de un nuevo contacto y captura de información para la carga de un seguimiento.
- Almacenamiento en memoria local de información sobre casos y contactos.
- Stack de requests que abarca todos los cambios hechos sin conexión a los registros, y posteriormente los ejecuta uno por uno cuando hay conexión a internet.

Una vez terminado el desarrollo se cuenta con un total de 15 vistas en toda la aplicación y 13 *assets* compuestos en su totalidad por las imágenes e iconos utilizados, todos en formato PNG. Se siguieron las buenas prácticas descritas para react-native en cuanto a la estructura del proyecto. Apoyando la escalabilidad del proyecto.

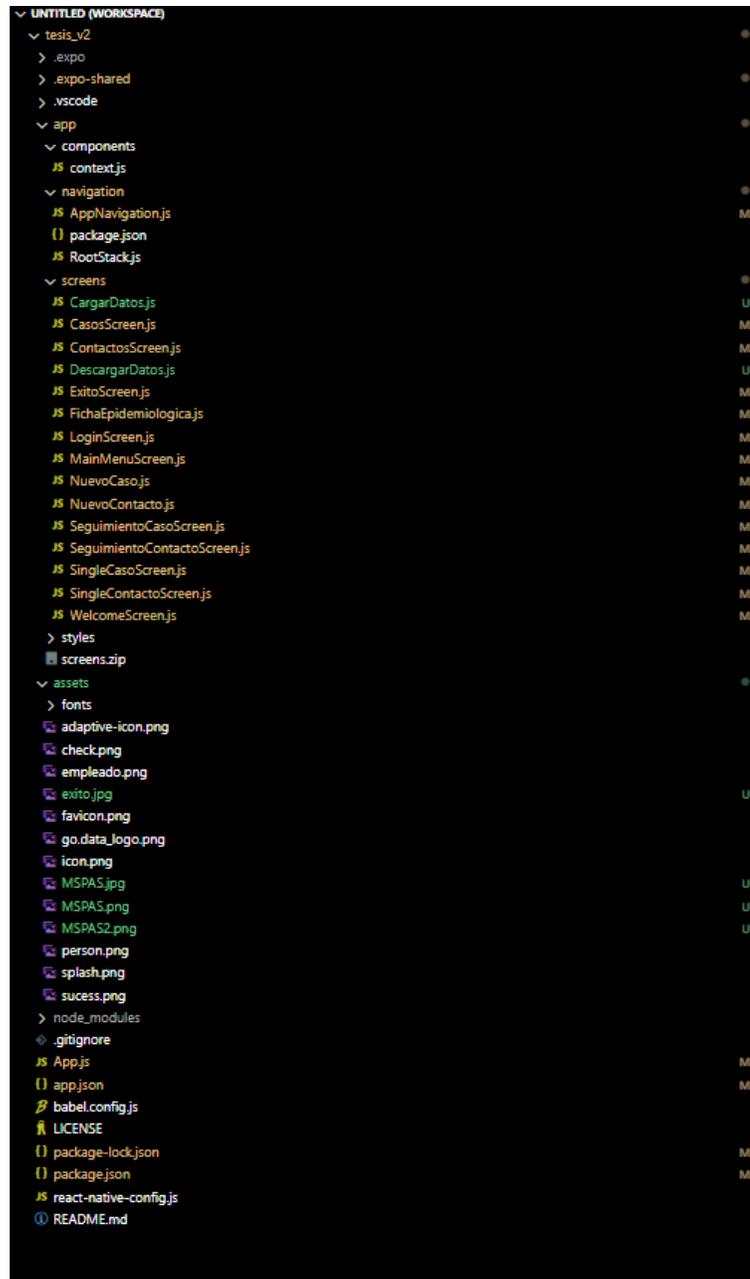


Figura 16: Estructura del proyecto

Al empezar la aplicación desde un dispositivo móvil, está detecta si se tiene una sesión activa o es necesario iniciar sesión. Esto lo logra mediante un token de sesión extraído desde la llamada del API. Si el token de sesión está activo, la aplicación se abre directamente en el menú principal. Si no está activo, se presenta una *splash screen* con el logo de Go.Data. Este logo tiene una animación que dura 5 segundos y después navega hasta la página de inicio de sesión.



Figura 17: Splash screen de bienvenida con animación de logo Go.Data

Esta característica se puede lograr mediante la utilización de una variable global llamada *userToken*. Para lograr esta funcionalidad fue necesario crear un *reducer* para la autenticación, que determina el estado de la variable *loginState*.

```
const loginReducer = (prevState, action) => {
  switch( action.type ) {
    case 'RETRIEVE_TOKEN':
      return {
        ...prevState,
        userToken: action.token,
        isLoading: false,
      }
    case 'LOGIN':
      return {
        ...prevState,
        username: action.id,
        userToken: action.token,
        isLoading: false,
      };
    case 'LOGOUT':
      return {
        ...prevState,
        userName: null,
        userToken: null,
        isLoading: false,
      }
  }
};
```

Figura 18: Reducer para manejo de LogIn

Dentro de react-native existe una funcionalidad nativa de crear un contexto de autenticación. Hacemos uso de esto para hacer una llamada asíncronica y obtener el token de sesión, para luego hacer un *dispatch* que envía el nombre de usuario y el token de sesión. La llamada asíncronica se realiza en la función asíncronica *LogIn*. Dentro del mismo contexto de autenticación, construimos el método para hacer *LogOut* de la aplicación.

```
const authContext = React.useMemo(() => ({
  signIn: async(userName, password) => {
    let userToken;
    userToken = null;
    try{
      await fetch('https://godataguatemala.mspas.gob.gt/api/users/login/', {
        method: 'POST',
        headers:{
          'Accept': '*/*',
          'Content-type': 'application/json'
        },
        body: JSON.stringify({
          'email': userName,
          'password': password
        })
      })
    }.then(res => res.json())
    .then(async data => {
      console.log(data);
      if(data != undefined){
        try {
          if( data.error.name == "Error" ){
            alert("¡No se puedo iniciar sesión! usuario o contraseña incorrectos");
          }
        } catch(e) {
          //console.log(e);
        }

        try{
          userToken = data.id;
          await AsyncStorage.setItem('userToken', userToken);
        } catch(e) {
          console.log(e);
        }
      }
    })
    console.log(userToken);
    dispatch({ type:'LOGIN', id: userName, token: userToken })
  } catch(e) {
    console.log(e);
  }
},
  signOut: async() => {
    try {
      await AsyncStorage.removeItem('userToken');
      console.log("cerre sesion");
    } catch (e) {
      console.log(e)
    }
    dispatch({ type:'LOGOUT' })
  },
}), []);
```

Figura 19: Contexto de autenticación de la aplicación

Utilizamos después llamadas en la función asíncronica por defecto de react-native *useEffect* para saber el valor actual de la variable *userToken* y actuar acorde. Si es null, redirigir a la página de *LogIn*, si es posible recuperar el token, redirige hacía el menú principal.

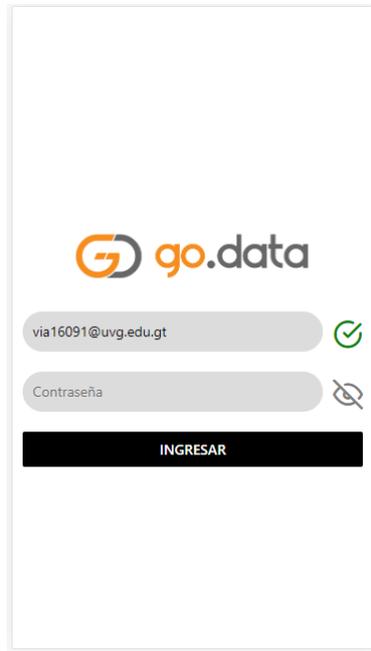


Figura 20: Ingreso por medio de usuario y contraseña



Figura 21: Menú principal de la aplicación

Una vez dentro del menú principal, es posible navegar hacía cualquiera de las 4 funcionalidades de la aplicación. Exploramos los resultados de la función de casos. Para lograr armar esta vista fue necesaria la utilización de una *FlatList* donde se reflejará la información recuperada del API. Es necesario parsear el JSON de respuesta para que sea acorde al tipo de dato aceptado por la *FlatList*. Dentro de cada elemento de la lista existirá un componente *TouchableOpacity* que redirigirá a una vista específica de cada caso. También es posible agregar un nuevo caso en esta vista.

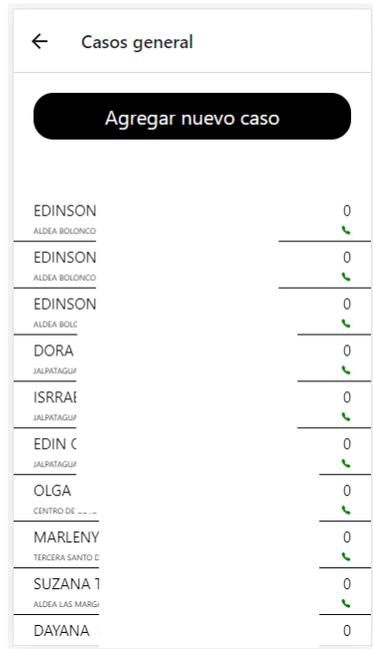


Figura 22: Vista general de todos los casos en sistema centralizado



Figura 23: Vista detalle de caso en específico

← Ficha Epidemiológica	
Campo	Valor
Forma	Ficha Epidemiológica 1
Ficha de notificación	COVID1712011202010...
Grupo cultural	1
Escolaridad	2
Enfermedades asociadas	
Síntomas	3567891112
¿muestra respiratoria?	1
Fecha y hora muestra	2020-10-10T00:00:00.0...
Tipo de muestra	2

Figura 24: Vista detalle de ficha epidemiológica de caso en específico

←
Nuevo caso

Nombres

Apellidos

Género

Masculino
▼

Edad

Fecha de notificación

1
2

1
2

2019
2020

Figura 25: Vista de captación de datos para un nuevo caso

En la vista de los casos en general, se utilizan los mismo componentes que para la vista de casos. Utilizamos una *FlatList* donde se reflejará la información recuperada del API. Es necesario parsear el JSON de respuesta para que sea acorde al tipo de dato aceptado por la *FlatList*. Dentro de cada elemento de la lista existirá un componente *TouchableOpacity* que redirigirá a una vista específica de cada contacto. También es posible agregar un nuevo contacto en esta vista.



Figura 26: Vista general de todos los contactos en sistema centralizado



Figura 27: Vista detalle de contacto en específico

← Nuevo contacto

Nombres

Escriba aquí nombre(s)

Apellidos

Escriba aquí apellido(s)

Género

Masculino

Edad

Escriba aquí edad

Fecha de notificación

1 1 2019
2 2 2020
3 3 2021

Nivel de riesgo

Figura 28: Vista de captación de datos para un nuevo contacto

← Seguimiento de un caso

Estado de seguimiento

Activo

¿Se logro seguimiento?

Si

Fecha si se logro

1 1 2019
2 2 2020
3 3 2021

Fecha no se logro

1 11 2019
2 12 2020
3 1 2021

¿Por qué?

No respondió la llamada

Figura 29: Vista de captación de datos para un nuevo seguimiento

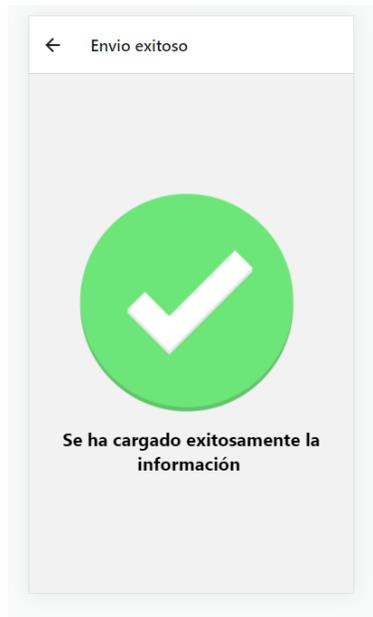


Figura 30: Vista de carga exitosa

Se logró implementar todas las funcionalidades anteriormente descritas de las cuáles cabe destacar la comunicación por medio de un API con Go.Data, la carga exitosa de nueva información al sistema centralizado y el manejo de un stack de requests.

Posterior a la finalización del proyecto se presentó por medio de Microsoft Teams al equipo que dio seguimiento durante todo el desarrollo de la aplicación. Al tener la autorización, se procedió a compartir un archivo con extensión .APK que se entrega al MSPAS como finalización de la investigación. Se obtuvo una retroalimentación positiva, indicando que la aplicación era sencilla de utilizar y cumple a cabalidad su propósito. Se realizan pruebas, llevando al menos 50 intentos por cada una de las funcionalidades. Se logra hacer una consulta a todos los casos generales en un tiempo promedio de 3 minutos y 25 segundos. Se logra hacer una consulta de casos en un tiempo promedio de 1 minuto 13 segundos. Para la carga de datos, depende de la longitud del stack de solicitudes, pero en promedio, una solicitud de carga tarda 33 segundos.

Discusión de resultados

El propósito esperado de la aplicación es que sea utilizada por los rastreadores del MSPAS para facilitar su labor de captación y seguimiento de datos. Dándoles la posibilidad, sin importar el área geográfica ni conexión a internet, de recolectar datos que servirán para alimentar a un sistema centralizado. Dicha información servirá para tener resultados más fehacientes de la situación actual del país en temas de la pandemia de COVID-19. Esta aplicación ayuda a facilitar la tarea de los colaboradores rastreadores del Ministerio, así como agiliza la captación de datos al ser en un dispositivo móvil, de fácil uso para cualquier individuo.

De la retroalimentación y entrega de investigación en formato solicitado, se logran extraer comentarios positivos con respecto al esfuerzo realizado. Comentarios de los stakeholders nos dan a entender que la aplicación cumple con las expectativas iniciales del Ministerio y tiene potencial para escalar y convertirse en una herramienta de uso regular a lo interno.

Se mostró bastante interés especialmente en el apartado de cargar nuevos casos, contactos y seguimientos. Esta tarea es complicada con los procedimientos actuales que se tienen, por lo que la facilidad de poder llevarla a cabo en una aplicación, desde el teléfono móvil, es de especial interés para los actores y usuarios del software. Otro punto a destacar es el comentario realizado por la colaboradora regional que trabaja con Go.Data. Dicho comentario hace alusión al esfuerzo pionero por parte del Ministerio y de esta iniciativa, ya que específicamente la funcionalidad de carga es única en la región, y se cataloga como innovadora.

Para el proyecto se buscó un objetivo principal de generar una aplicación que facilitará la captación de datos y tuviera el requerimiento de mostrar información pertinente sobre los casos y contactos registrados en el territorio en el sistema Go.Data. Lo cual se logró mediante la investigación recién presentada. Habilitando la posibilidad de:

- Login en la aplicación.
- Manejo del tiempo de vida del token de sesión de un usuario.
- Información general de un caso.
- Información general de un contacto.
- Creación de nuevo caso.
- Creación de un nuevo contacto.

- Creación de un nuevo seguimiento.
- Posibilidad de trabajar sin conexión.

Al cumplir satisfactoriamente con los puntos anteriormente mencionados, se logra interactuar desde un dispositivo externo con la entidad centralizadora para el envío y recepción de datos. Permite a los rastreadores del MSPAS acceder y enviar datos a Go.Data para la estrategia de rastreo de casos y contactos. Continuando con los objetivos específicos, se logra implementar una comunicación mediante la estructura de datos JSON para definir el protocolo de envío y recepción a través de un API. Se garantiza la disponibilidad de los datos ingresados en un esquema de almacenamiento local, actualizado a petición del usuario final. Se permite el ingreso y consulta de casos y contactos sin importar si el dispositivo cuenta con conexión a internet o no, si previamente se realizó la descarga respectiva de información.

Durante el proceso de desarrollo se encontraron diferentes contratiempos con el entorno de desarrollo, React Native, ya que existían discrepancias con el versionamiento de distintos paquetes y se debió buscar la correcta sinergia entre estos. Debido a las particularidades de los requerimientos, errores en el almacenamiento de solicitudes e inicio de sesión debieron ser solucionados mediante la evaluación propia del código, ya que una solución como tal no se encontraban en foros populares de consulta.

A pesar de la entrega de un APK para la utilización en dispositivos Android, el MSPAS decide implementar en Guatemala una aplicación promovida por el mismo sistema centralizados Go.Data. Esta decisión se toma para estar alineados con los países de la región y estandarizar el uso de la aplicación Kawoq a lo largo del territorio guatemalteco. Un factor de peso en la decisión es que se trabaja con Go.Data desde hace tiempo y ellos mismos ofrecen brindar soporte por cualquier eventualidad con la aplicación; situación que al ser este un proyecto de graduación, queda descartada la posibilidad de brindar soporte a lo largo del tiempo de uso.

- Por medio de una revisión bibliográfica se determinó que la utilización de React Native como entorno de desarrollo es la más adecuada para la creación de la aplicación móvil.
- Reuniones con stakeholders por medio de Microsoft Teams prueban la aceptación de la aplicación para la agilización y captación de datos para la estrategia de rastreo de asos y contactos de COVID-19.
- El tiempo promedio en completar una consulta de casos general es 3 minutos y 25 segundos.
- El tiempo promedio en completar una carga de información al sistema centralizado es de 33 segundos.
- La implementación correcta de las funcionalidades permite la interacción entre un dispositivo móvil y el sistema centralizador de datos Go.Data

Recomendaciones

Una vez entregada la aplicación y comprobado su funcionamiento, se identifican posibles puntos de mejora. Como recomendación para ampliar el alcance de la aplicación realizada, es importante el contacto en intimidad con la labor de los rastreadores del MSPAS. Ya que este contacto solo se tuvo de manera indirecta. La interacción directa con los rastreadores podría revelar nuevas oportunidades de mejora para la aplicación, y agilizar aún más el proceso de captación de datos. La aplicación está pensada únicamente para la interacción que tienen los rastreadores con las personas. Sin embargo, ellos desempeñan otras labores más que la interacción misma con el caso o contacto. Por lo que existen más posibilidades de apoyarlos, o mejor dicho, apoyar a mejorar el accionar de los elementos públicos ante una pandemia. Una de estas podrían ser con sistemas que les permitan llevar un historial de los lugares visitados con base en la geografía de Guatemala. Que les permita tener una base de datos que indique cuando falte entrevistar a alguna persona en específico. Facilitar las rutas de los pilotos mediante la recopilación de la posición geográfica del área donde viven las personas.

Es posible agregar funcionalidades a la aplicación para que el desarrollo de esta investigación sea un módulo de la misma, y así cubrir otras necesidades o casos de uso que se logren identificar en el trabajo diario de los colaboradores del Ministerio. Verificar siempre que las dependencias instaladas sean compatibles con los módulos esenciales del proyecto, para evitar discrepancias en el versionamiento. Se recomienda agregar un filtro a la hora de buscar casos y contactos, que facilite encontrar alguno en específico que se desea manipular. Si se desea, es posible implementar preguntas para cubrir otras enfermedades. Es decir, agregar un nuevo set de preguntas y hacer una división sobre el brote.

1. *Aerosol and Surface Stability of SARS-CoV-2 as Compared with SARS-CoV-1* N. Engl. J. Med. (2020)
2. Alanzi T. (2021). *A Review of Mobile Applications Available in the App and Google Play Stores Used During the COVID-19 Outbreak*. *Journal of multidisciplinary healthcare*, 14, 45–57. <https://doi.org/10.2147/JMDH.S285014>
3. Amatya, S., Kurti, A. (2014). *Cross-Platform Mobile Development: Challenges and Opportunities*. *Advances in Intelligent Systems and Computing*, 219–229. doi:10.1007/978-3-319-01466-1_21
4. Bi Q, Wu Y, Mei S, Ye C, Zou X, Zhang Z, et al. *Epidemiology and Transmission of COVID-19 in Shenzhen China: Analysis of 391 cases and 1,286 of their close contacts*. medRxiv. 2020.
5. Branas, C. C., DiNardo, A. R., Puac-Polanco, V. D., Harvey, M. J., Vassy, J. L., Bream, K. (2013). *An exploration of violence, mental health and substance abuse in post-conflict Guatemala*. *Health*, 5(5), 825–833.
6. Charland, A., Leroux, B. (2011). *Mobile application development: web vs. native*. *Communications of the ACM*, 54(5), 49-53.
7. Cui et al., 2019 J. Cui, F. Li, Z. Shi *Origin and evolution of pathogenic coronaviruses* *Nat Rev Microbiol*, 17 (2019), pp. 181-192
8. Daniel W., von Weitershausen, P. *React Native for Android: How we built the first cross-platform React Native app*. (2015) <https://code.facebook.com/posts/first-cross-platform-react-native-app/>. Accessed: 2021-10-24. /1189117404435352 / react - native - for - android - how - we - built - the -
9. Danielsson, W. (2016). *React Native application development*. *Linköpings universitet*, Swedia, 10(4).
10. David M. Morens, Gregory K. Folkers, Anthony S. Fauci, *What Is a Pandemic*, *The Journal of Infectious Diseases*, Volume 200, Issue 7, 1 October 2009, Pages 1018–1021, <https://doi.org/10.1086/644537>
11. *Detection of SARS-CoV-2 in Different Types of Clinical Specimens*. *Jama*. (2020)
12. Drosten et al., 2020 C. Drosten, S. Günther, W. Preiser *Identification of a novel coronavirus in patients with severe acute respiratory syndrome* *N Engl J Med*, 348 (2020), pp. 1967-1976

13. Eisenman, B. *Learning React Native*. O'Reilly Media, Inc. (2015). ISBN: 9781491929049
14. *Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study* *The Lancet.*, 395 (10223) (2020), pp. 507-513
15. FUNDESA. *Percepción de Corrupción. Fundación para el Desarrollo de Guatemala*. Acceso November 16, 2021, de <https://www.fundesa.org.gt/indices-y-evaluaciones-de-pais/indices-internacionales/percepcion-de-corrupcion>.
16. Glassman A, Gaziano TA, Buendia CPB, de Aguiar FCG. *Confronting the chronic disease burden in Latin America and the Caribbean*. *Health Aff (Millwood)*. 2010;29(12):2142–2148.
17. Grasselli G, Zangrillo A, Zanella A, Antonelli M, Cabrini L, Castelli A, Cereda D, Coluccello A, Foti G, Fumagalli R, Iotti G, Latronico N, Lorini L, Merler S, Natalini G, Piatti A, Ranieri MV, Scandroglio AM, Storti E, Cecconi M, Pesenti A. *COVID-19 Lombardy ICU Network. Baseline characteristics and outcomes of 1591 patients infected with SARS-CoV-2 admitted to ICUs of the Lombardy region, Italy*. *JAMA*. 2020;323(16):1574–81. <https://doi.org/10.1001/jama.2020.5394>.
18. Hui DS, I Azhar E, Madani TA, Ntoumi F, Kock R, Dar O, et al. *The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health—The latest 2019 novel coronavirus outbreak in Wuhan, China*. *International Journal of Infectious Diseases*. 2020;91:264-6.
19. Islam, R., Islam, R., Mazumder, T. (2010). *Mobile application and its global impact*. *International Journal of Engineering Technology (IJEST)*, 10(6), 72-78.
20. Kevin Martinez-Folgar, Diego Alburez-Gutierrez, Alejandra Paniagua-Avila, Manuel Ramirez-Zea, and Usama Bilal, *2021: Excess Mortality During the COVID-19 Pandemic in Guatemala* *American Journal of Public Health* 111, 1839_1846, <https://doi.org/10.2105/AJPH.2021.306452>
21. Kondylakis H, Katehakis DG, Kouroubali A, Logothetidis F, Triantafyllidis A, Kalamaras I, Votis K, Tzovaras D. *COVID-19 Mobile Apps: A Systematic Review of the Literature*. *J Med Internet Res* 2020;22(12):e23170
22. Last JM. *A dictionary of epidemiology*. 2nd ed, 1988 Oxford Oxford University Press p.94
23. Liu Y, Ning Z, Chen Y, Guo M, Liu Y, Gali NK, et al. *Aerodynamic characteristics and RNA concentration of SARS-CoV-2 aerosol in Wuhan hospitals during COVID-19 outbreak*. bioRxiv. 2020.
24. Liu, S. (2021, September 16). *Cross-platform mobile frameworks used by global developers 2021*. Statista. Retrieved October 29, 2021, from <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>. Accessed: 2021-10-28
25. McIntosh K, Hirsch MS, Bloom A. *Coronavirus disease 2019 (COVID-19)*. UpToDate Hirsch MS, Bloom A (Eds) Accessed Mar. 2020;5.
26. Ministerio de Salud Pública y Asistencia Social. (2020, August 12). *Estrategia nacional de rastreo de casos y contactos COVID-19 con plan piloto en municipio de Guatemala*. <https://www.Mspas.Gob.Gt/>. Retrieved October 15, 2021, from <https://www.mspas.gob.gt/noticias/noticias-ultimas/5-noticias-mspas/1009-estrategia-nacional-de-rastreo-de-casos-y-contactos-covid-19-con-plan-piloto-en-municipio-de-guatemala.html>
27. Ministerio de Salud Pública y Asistencia Social. (2020b, August 14). *¿Qué es un rastreador de casos y contactos COVID-19?* <https://www.Mspas.Gob.Gt/>. Retrieved October 15, 2021, from <https://www.mspas.gob.gt/noticias/noticias-ultimas/5-noticias-mspas/1012-%C2%BFqu%C3%A9-es-un-rastreador-de-casos-y-contactos-covid-19.html>
28. N. Rezaei, *COVID-19 affects healthy pediatricians more than pediatric patients* *Infection Control Hospital Epidemiology* (2020).

29. *Organización Panamericana de la Salud*. (2020, August 12). La OPS/OMS participa en el lanzamiento de Estrategia de rastreo de casos y contactos COVID-19 del Ministerio de Salud. <https://www.paho.org/>. Retrieved October 15, 2021, from <https://www.paho.org/es/noticias/12-8-2020-opsoms-participa-lanzamiento-estrategia-rastreo-casos-contactos-covid-19>
30. P. Pérez-Martínez, F.J. Carrasco Sánchez, J. Carretero Gómez, R. Gómez-Huelgas, *Resolviendo una de las piezas del puzle: COVID-19 y diabetes tipo 2*, Revista Clínica Española, Volume 220, Issue 8, 2020, Pages 507-510, ISSN 0014-2565, <https://doi.org/10.1016/j.rce.2020.05.003>. (<https://www.sciencedirect.com/science/article/pii/S0014256520301351>)
31. Prensa Libre. *Coronavirus: as cases go up there is still little evidence [in Spanish]*. Available at: <https://www.prensalibre.com/guatemala/comunitario/coronavirus-mientras-casos-suben-aun-hay-pocas-pruebas>. Accessed June 19, 2020
32. *React Native Documentation*. <https://facebook.github.io/react-native/>. Accessed: 2021-10-28.
33. Reddy, M. (2011). *API design for C++*. Burlington, MA: Morgan Kaufmann Publishers/Elsevier
34. Rosenbaum L. *The untold toll—the pandemic’s effects on patients without COVID-19*. N Engl J Med. 2020;382(24):2368–2371.
35. Rosenthal, G. (2020). *El impacto de Covid-19 en Guatemala*. *Journal of Economic Literature*, 17(51), 147–160.
36. Santarpia JL, Rivera DN, Herrera V, Morwitzer MJ, Creager H, Santarpia GW, et al. *Transmission Potential of SARS-CoV-2 in Viral Shedding Observed at the University of Nebraska Medical Center*. medRxiv. 2020.
37. Smith AC, Thomas E, Snoswell CL, Haydon H, Mehrotra A, Clemensen J, et al. *Telehealth for global emergencies: implications for coronavirus disease 2019 (COVID-19)*. J Telemed Telecare 2020 Jun;26(5):309-313
38. *Stedman’s medical dictionary*. 28th ed, 2006 Philadelphia Lippincott, Williams Wilkins
39. Sun P, Lu X, Xu C, Sun W, Pan B. *Understanding of COVID-19 based on current evidence*. J Med Virol. 2020;92(6):548–551.
40. Tang A, Tong Z, Wang H, Dai Y, Li K, Liu J, et al. *Detection of Novel Coronavirus by RT-PCR in Stool Specimen from Asymptomatic Child, China*. Emerging infectious diseases. W. Wang, Y. Xu, R. Gao, R. Lu, K. Han, G. Wu, et al.2020;26(6).
41. Tang Y, Schmitz JE, Persing DH, Stratton CW. *Laboratory diagnosis of COVID-19: current issues and challenges*. J Clin Microbiol. 2020;58(6):e00512–e00520.
42. Triantafyllidis A, Kondylakis H, Votis K, Tzovaras D, Maglaveras N, Rahimi K. *Features, outcomes, and challenges in mobile health interventions for patients living with chronic diseases: A review of systematic reviews*. Int J Med Inform 2019 Dec;132:103984.
43. *University of Maryland Pathogenic Microbiology Web page*. Available at: <http://www.life.umd.edu/classroom/bsci424/HostParasiteInteractions/HostParasiteSummary.htm>. Accesado 27 de octubre 2021
44. Weiss and Leibowitz, 2011 S.R. Weiss, J.L. Leibowitz *Coronavirus pathogenesis Adv Virus Res*, 81(2011), pp. 85-164
45. Wilder-Smith A, Freedman D. *Isolation, quarantine, social distancing and community containment: pivotal role for old-style public health measures in the novel coronavirus (2019-nCoV) outbreak*. Journal of travel medicine. 2020;27(2):taaa020.

46. World Health Organization. (2019). *Coronavirus disease (COVID-19)*. www.Who.Int. Retrieved October 14, 2021, from https://www.who.int/health-topics/coronavirustab=tab_1
47. X. Lu, L. Zhang, H. Du, J. Zhang, Y.Y. Li, J. Qu, et al. *SARS-CoV-2 Infection in Children* N. Engl. J. Med. (2020)
48. Y. Dong, X. Mo, Y. Hu, X. Qi, F. Jiang, Z. Jiang, et al. *Epidemiological characteristics of 2143 pediatric patients with 2019 coronavirus disease in China* *Pediatrics* (2020)
49. Yanez, N.D., Weiss, N.S., Romand, JA. et al. *COVID-19 mortality risk for older men and women*. *BMC Public Health* 20, 1742 (2020). <https://doi.org/10.1186/s12889-020-09826-8>
50. Yang and Leibowitz, 2015 D. Yang, J.L. Leibowitz *The structure and functions of coronavirus genomic 3' and 5' ends* *Virus Res*, 206 (2015), pp. 120-133
51. Zaki et al., 2012 A.M. Zaki, S.V. Boheemen, T.M. Bestebroer, A.D.M.E. Osterhaus, R.A.M. Fouchier *Isolation of a novel coronavirus from a man with pneumonia in Saudi Arabia*, N Engl J Med, 367 (2012), pp. 1814-1820



Figura 31: Design Thinking fase de empatizar



Figura 32: Design Thinking fase de empatizar

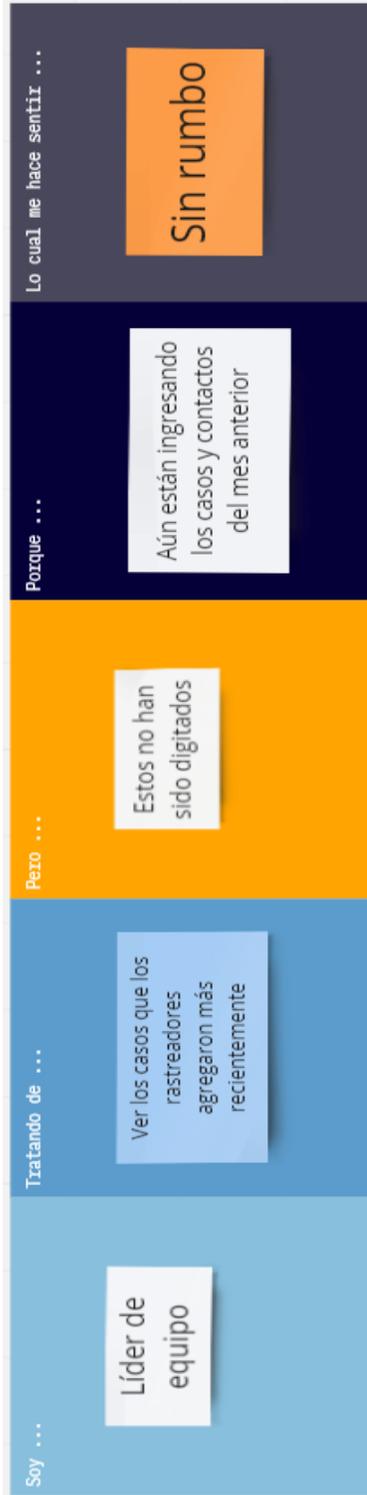


Figura 33: Design Thinking fase de empatizar

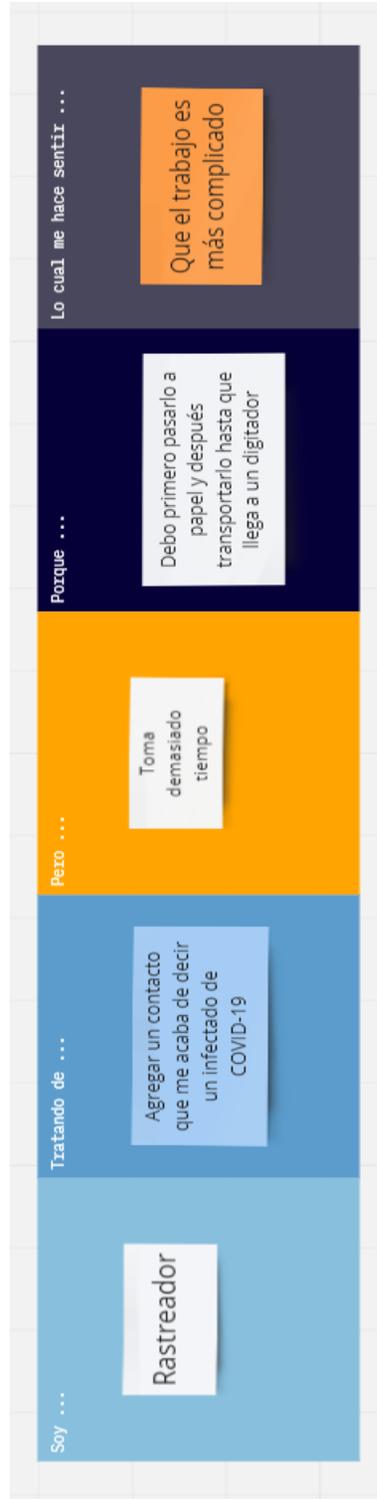


Figura 34: Design Thinking fase de empatizar

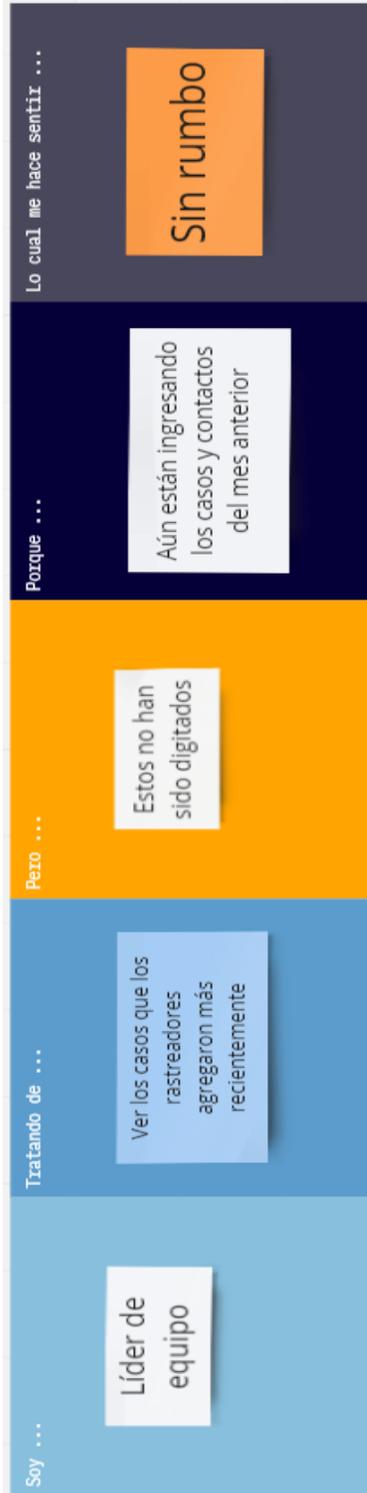


Figura 35: Design Thinking fase de empatizar

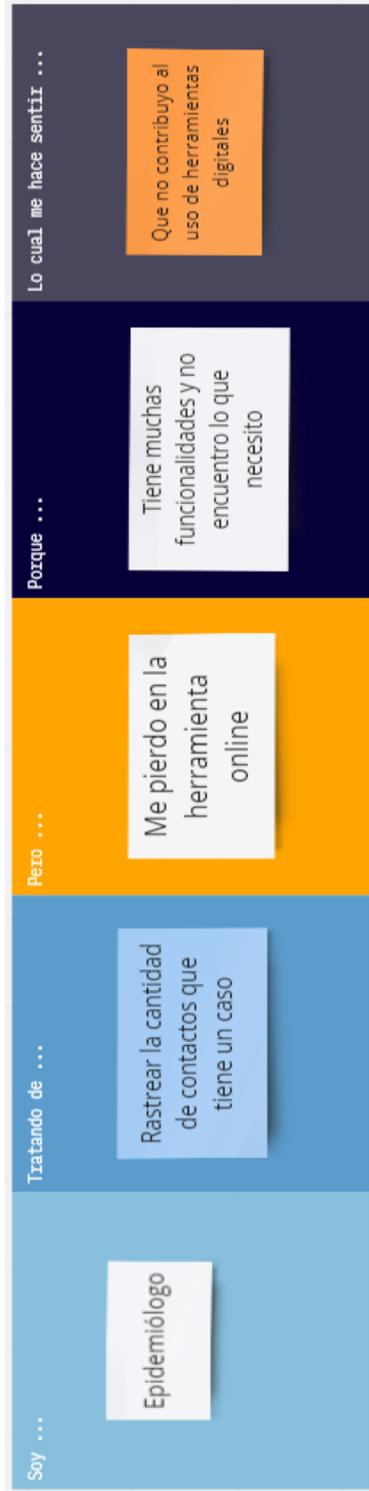


Figura 36: Design Thinking fase de empatizar

Diagrama de caso de uso

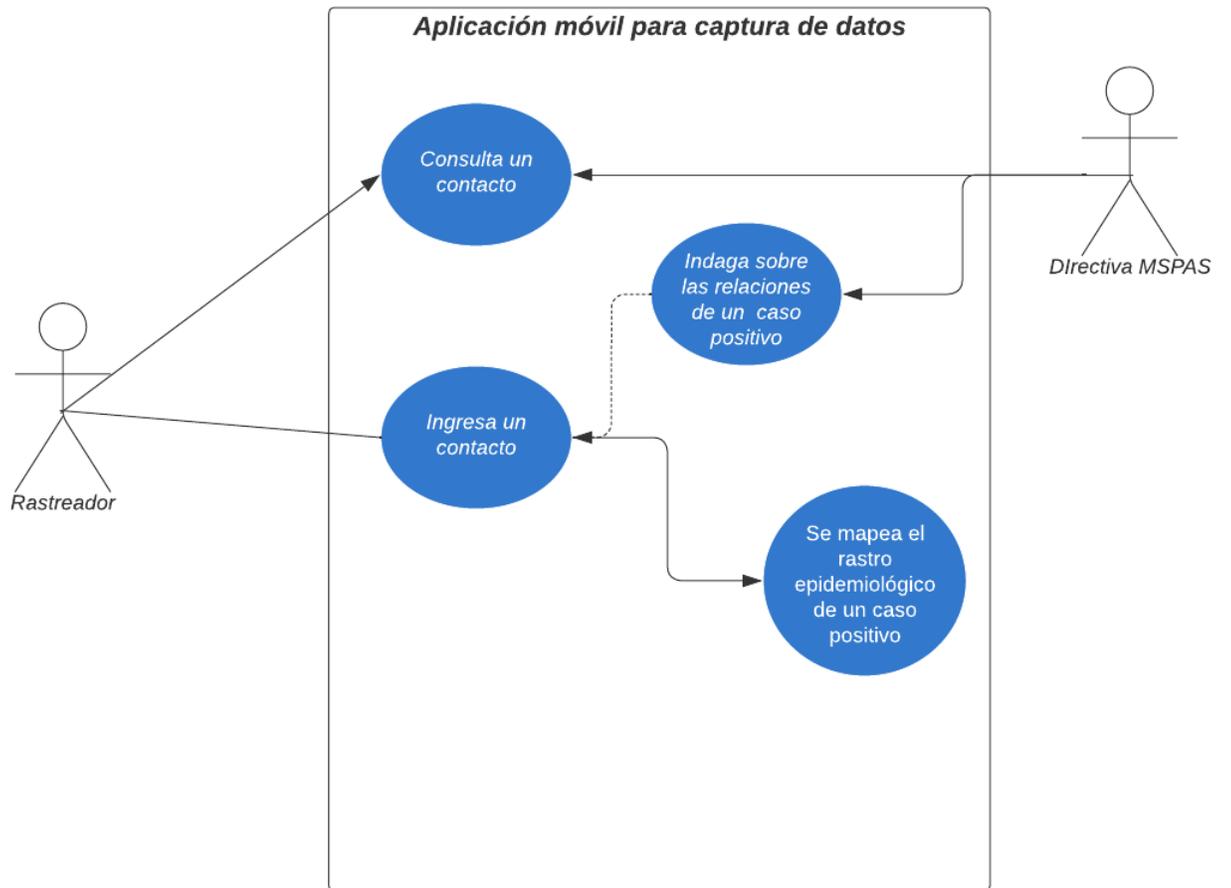


Figura 37: Primer caso de uso

Diagrama de caso de uso

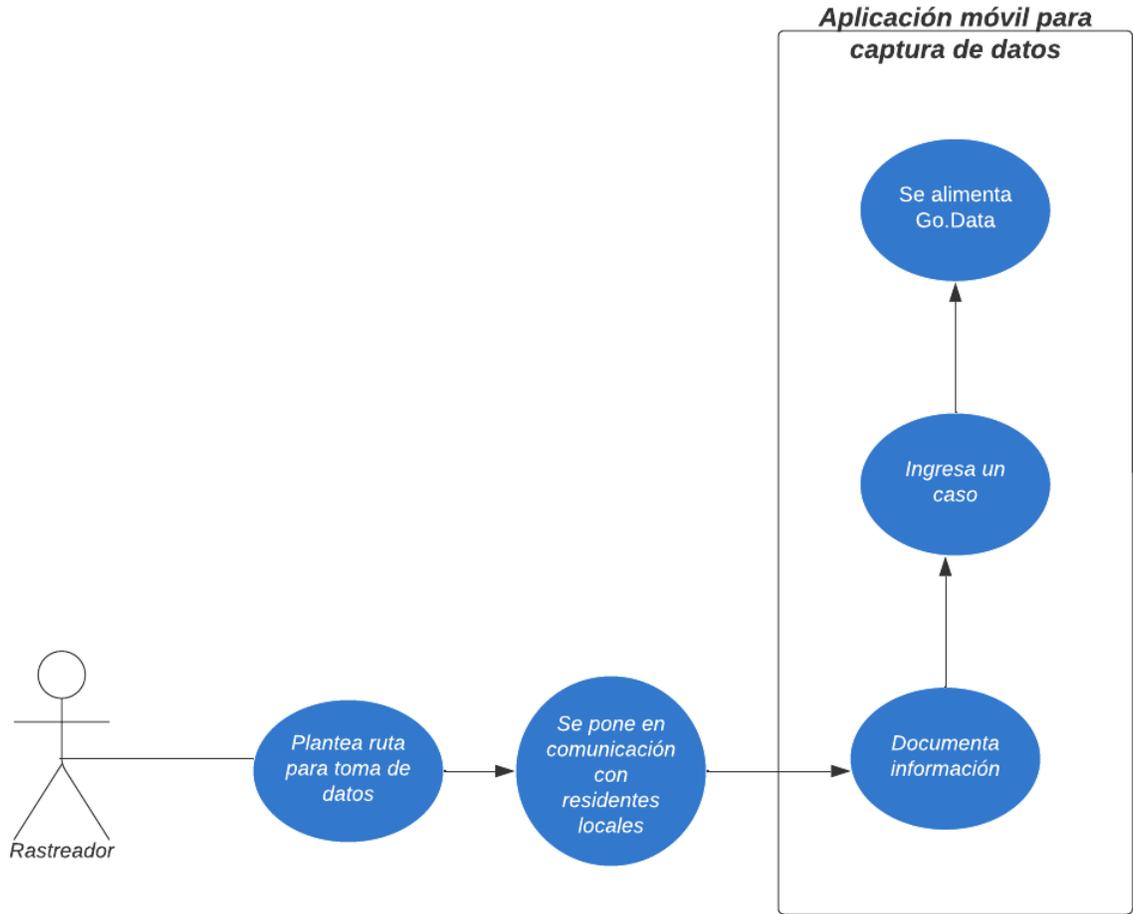


Figura 38: Segundo caso de uso

Diagrama de caso de uso

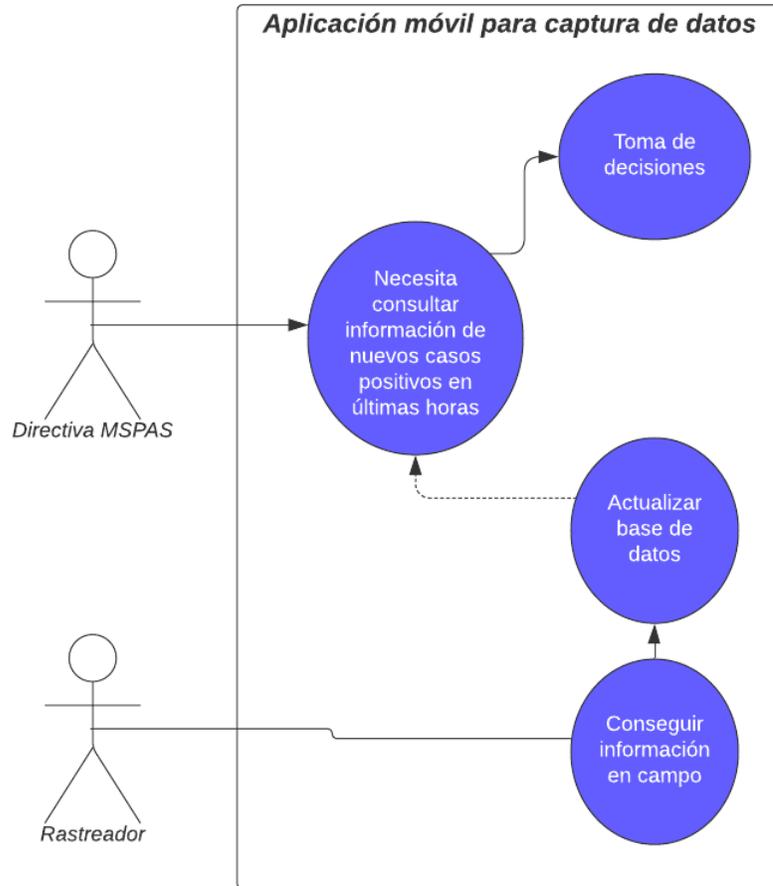


Figura 39: Tercer caso de uso

Diagrama de caso de uso

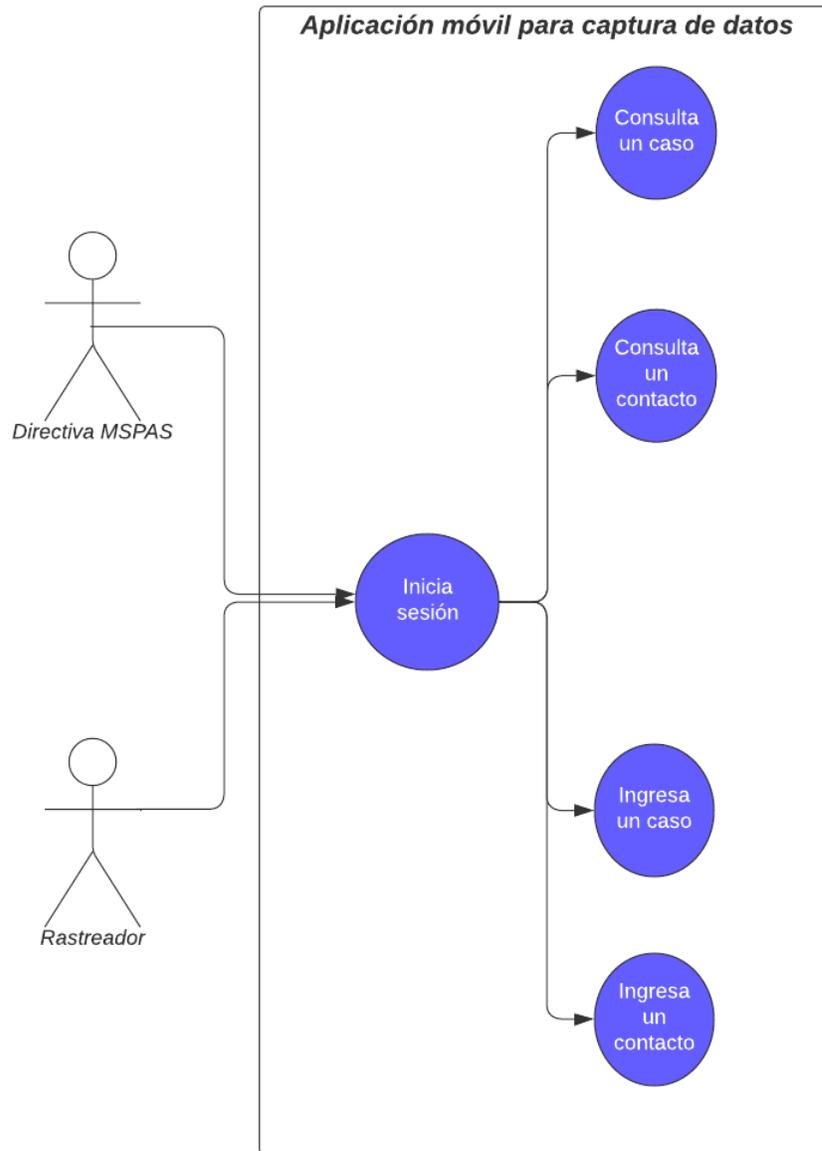


Figura 40: Cuarto caso de uso