

Documentación Técnica

Aplicaciones Móviles

Sistema descentralizado para detectar zonas de riesgo y
contacto con personas confirmadas con COVID-19 protegiendo
la privacidad de los participantes

CONACyT-313572-Appacovid-DT-AM Versión 1.0

Centro de Investigación y de Estudios Avanzados del IPN
Departamento de Computación

1 de diciembre de 2020

Datos del proyecto		
Organización:	Cinvestav	Departamento de Computación
Proyecto:	CONACyT-313572	Sistema descentralizado para detectar zonas de riesgo y contacto con personas confirmadas con COVID-19 protegiendo la privacidad de los participantes
Sistema:	Applacovid	Aplicaciones Móviles

Documento		
Clave	Nombre	Versión
DT-AM	Documentación Técnica	Versión 1.0

Documentos Relacionados		
Clave	Versión	Nombre
Protocolo	1.0	Propuesta Técnica
DT-Android	1.0	Aplicación móvil Android
DT-iOS	1.0	Aplicación móvil iOS
DT-SIG	1.0	Sistema de Información Geográfica
DT-PW	1.0	Página Web

Firmas		
Responsable Técnico	Colaborador	Colaborador
Dr. Francisco Rodríguez Henríquez Profesor-Investigador Cinvestav	Dra. Brisbane Ovilla Martínez Profesor-Visitante Cinvestav	Dr. Cuauhtémoc Mancillas López Profesor-Investigador Cinvestav

Índice general

1. Introducción	1
1.1. Objetivo del documento	1
1.2. Descripción general del sistema	1
1.3. Estructura del documento	2
2. Glosario	3
2.1. Glosario de términos	3
2.2. Definición de abreviaturas	4
3. Estructura del sistema	5
3.1. Estructura general de Applacovid	5
3.2. Subsistema: Aplicaciones móviles	7
3.2.1. Protocolo de comunicación entre dispositivos	7
3.2.2. Reporte de usuario con COVID-19	7
3.2.3. Notificación de riesgo de contacto	8
3.2.4. Creación de Identificadores Efímeros	8
3.2.5. Cálculo del riesgo de infección	8
3.3. Diseño del concepto	9
3.4. Diseño de interfaces	12
4. Requerimientos del sistema	17
4.1. Requerimientos funcionales	18
4.2. CU-01 Descargar aplicación	20
4.2.1. Objetivo	20
4.2.2. Atributos	20
4.2.3. Trayectorias del Caso de Uso	20
4.3. CU-02 Instalar Applacovid	23
4.3.1. Objetivo	23
4.3.2. Atributos	23
4.3.3. Trayectorias del Caso de Uso	23
4.4. CU-03 Iniciar rastreo	31

4.4.1. Objetivo	31
4.4.2. Atributos	31
4.4.3. Trayectorias del Caso de Uso	31
4.4.4. Puntos de extensión	31
4.5. CU-04 Revisar contactos	33
4.5.1. Objetivo	33
4.5.2. Atributos	33
4.5.3. Trayectorias del Caso de Uso	33
4.6. CU-05 Revisar notificaciones	35
4.6.1. Objetivo	35
4.6.2. Atributos	35
4.6.3. Trayectorias del Caso de Uso	36
4.7. CU-06 Ingresar prueba positiva	39
4.7.1. Objetivo	39
4.7.2. Atributos	39
4.7.3. Trayectorias del Caso de Uso	39
4.8. CU-07 Compartir código QR	45
4.8.1. Objetivo	45
4.8.2. Atributos	45
4.8.3. Trayectorias del Caso de Uso	45
4.9. CU-08 Notificar posible infección	49
4.9.1. Objetivo	49
4.9.2. Atributos	49
4.9.3. Trayectorias del Caso de Uso	49
4.10. Requerimientos no-funcionales	53
5. Modelo lógico de información	55
5.1. Modelo lógico del dispositivo	55
5.2. Modelo lógico del servidor	57
6. Especificación de la plataforma	61
6.1. Arquitectura del sistema	61
6.1.1. Cliente	62
6.1.2. Servidor	63
6.2. Arquitectura de los servicios web	64
6.3. Seguridad	65
6.4. Entorno de desarrollo	66
6.4.1. Bibliotecas de funciones de terceros	67
6.4.2. Código fuente del proyecto	67
7. Servicios Web	69
7.1. GET /v1/	69
7.2. POST /v1/exposed/	70
7.3. GET /v1/exposed/{batchReleaseTime}	71
7.4. Modelos	72
7.4.1. ExposeeRequest	72
7.4.2. ExposeeAuthData	72
7.4.3. ExposedOverview	72

7.4.4. Exposee	73
--------------------------	----

Índice de figuras

3.1. Subsistemas que componen a Applacovid.	6
3.2. Detección de contactos vía Bluetooth.	7
3.3. Logotipo y colores de Applacovid.	9
3.4. Ícono de aplicación en el dispositivo móvil.	9
3.5. Personajes.	10
3.6. Personajes siguiendo recomendaciones de salud.	10
3.7. Personajes explicando funciones de la aplicación.	11
3.8. Regreso a la nueva normalidad de forma segura 1.	11
3.9. Regreso a la nueva normalidad de forma segura 2.	12
3.10. Íconos y botones de Applacovid.	12
3.11. Pantalla inicial de Applacovid en dispositivos móviles.	13
3.12. Diseño para iOS (izquierda) y Android (derecha).	14
3.13. Formato de pantalla, iOS (izquierda) y Android (derecha).	15
3.14. Botones e íconos, iOS (izquierda) y Android (derecha).	15
3.15. Mensajes del sistema operativo, iOS (izquierda) y Android (derecha).	16
4.1. Casos de uso.	18
4.2. Página web de Applacovid.	21
4.3. Descargar Applacovid para Android.	21
4.4. Descargar Applacovid para iOS.	22
4.5. Formulario para descarga de Applacovid.	22
4.6. Iniciar instalación.	24
4.7. Instalando aplicación.	24
4.8. Finaliza instalación.	25
4.9. Objetivo.	25
4.10. Protegiendo tu privacidad.	26
4.11. ¿Cómo funciona?	26
4.12. Activar Bluetooth.	27
4.13. Ignorar optimización de batería.	28
4.14. Autorizar ignorar optimización de batería.	28
4.15. ¿Qué hace Applacovid?	29

4.16. Aplicación instalada.	29
4.17. Autorizar uso de GPS.	30
4.18. Inicio de Applacovid.	32
4.19. Opción “Contactos”.	34
4.20. Rastreo apagado.	34
4.21. Menú principal con rastreo apagado.	35
4.22. Opción “Notificaciones”.	36
4.23. <i>Push notifications</i> con pantalla bloqueada.	37
4.24. Barra de notificaciones del dispositivo.	37
4.25. Detener las cadenas de contagio.	40
4.26. Aviso importante.	40
4.27. Introduce código COVID-19.	41
4.28. Muchas gracias.	41
4.29. Rastreo terminado.	42
4.30. Te deseamos pronta mejoría.	43
4.31. Prueba positiva.	43
4.32. Auto aislamiento personal.	44
4.33. Código QR.	46
4.34. Compartir código QR.	47
4.35. Código QR en imagen.	47
4.36. Ingresar a Applacovid.	50
4.37. Menú principal con notificación de posible contagio.	50
4.38. Posible contagio.	51
4.39. ¿Qué debo hacer?	52
 5.1. Base de datos del dispositivo.	55
5.2. Base de datos del servidor.	57
 6.1. Arquitectura del subsistema.	61
6.2. Modelo-Vista-Controlador de Swift.	62
6.3. Arquitectura de los servicios REST.	65
6.4. Estructura del código del proyecto.	68

CAPÍTULO 1

Introducción

El presente documento es parte de los entregables del proyecto **CONACyT** con número de registro **313572** titulado “Sistema descentralizado para detectar zonas de riesgo y contacto con personas confirmadas con COVID-19 protegiendo la privacidad de los participantes”.

1.1. Objetivo del documento

Applacovid es una aplicación móvil para dispositivos móviles desarrollado para el proyecto **CONACyT-313572**. El presente documento es el **manual técnico** del proyecto y tiene como finalidad presentar las decisiones de análisis y diseño finales para implementar el funcionamiento del sistema. En este documento se presentan los términos utilizados, la lógica de negocio, los casos de uso, las interfaces de usuario, los mensajes y los requerimientos no funcionales que fueron revisados y aprobados por parte de los proponentes del proyecto.

1.2. Descripción general del sistema

Applacovid¹ es una aplicación móvil para celulares iOS y Android de la gama *smartphone*, desarrollada para ayudar a los usuarios a detectar contactos con contagio COVID-19.

A partir de su descarga, por parte de los usuarios en sus dispositivos móviles, Applacovid genera credenciales seudónimas, las cuales son renovadas periódicamente. Utilizando la tecnología de proximidad Bluetooth, estas credenciales seudónimas son intercambiadas constantemente entre todos los usuarios que estén en contacto cercano al usuario y que, por supuesto, tengan instalada la aplicación en sus dispositivos móviles. Por ejemplo, si dos usuarios (con Applacovid instalada) coinciden en la salida/entrada de una estación del metro o del metrobus, las aplicaciones intercambian los seudónimos de manera multitudinaria. Los seudónimos, al ser recibidos, son almacenados por la aplicación para ser clasificados con respecto

¹En adelante se usará indistintamente aplicación y/o Applacovid para referirse al sistema.

a sus metadatos asociados, siendo dos de los metadatos más importantes: la fecha en el instante de la coincidencia; y la proximidad del celular que hizo el envío.

Por otro lado, aquellos pacientes que recién han sido confirmados como casos activos de COVID-19 pueden informarlo oportunamente y de manera voluntaria, a través de una comunicación anónima entre Applacovid y el servidor del sistema². Esta acción permite al servidor hacer público a todos los usuarios, con Applacovid instalada, los seudónimos que pertenecen a personas que han sido confirmados como portadores de COVID-19. Esta información preserva la privacidad de las personas enfermas, puesto que el servidor desconoce las identidades reales de los dueños de tales seudónimos e incluso desconoce con qué seudónimos ha tenido interacción cada usuario ya que esto jamás es transmitido. Finalmente, el ciclo se cierra cuando Applacovid, en cada celular instalado, actualiza su base de datos con los registros de seudónimos más recientes agregados por el servidor del sistema. Applacovid verifica entonces si alguno o algunos de los seudónimos publicados en la última actualización aparecen en la base de datos local, y si fueron recibidos en los últimos días o semanas. De ser así, implica que ese usuario estuvo en contacto cercano con al menos una persona de quien ahora se sabe que es portadora del virus.

Applacovid cifra la información almacenada en el dispositivo y conforme a las políticas de control establecidas, permite que esta información se elimine una vez haya pasado su periodo de uso.

Por otro lado aquellos usuarios que den positivo a la prueba COVID-19 se les proporciona un código único e intransferible, por parte de la Secretaría de Salud. Este código puede ser utilizado para que el usuario se declare así mismo como contagiado ante el servidor del sistema. Este mecanismo evita que personas malintencionadas propaguen información falsa sobre contagios no existentes. De esta manera, sólo los usuarios verdaderamente contagiados tienen permiso de compartir información sobre su contagio con el servidor del sistema.

1.3. Estructura del documento

El presente documento se encuentra estructurado de la siguiente manera:

- Capítulo 2, presenta el glosario de términos y abreviaciones.
- Capítulo 3, describe la estructura a nivel de subsistemas de Applacovid, el protocolo de comunicación entre dispositivos, así como las consideraciones teóricas para el desarrollo del sistema.
- Capítulo 4, describe los requerimientos funcionales del sistema a manera de casos de uso; y los requerimientos no-funcionales.
- Capítulo 5, describe los modelos lógicos de información implementados en los dispositivos móviles y el *backend*.
- Capítulo 6, describe la arquitectura del sistema.
- Capítulo 7, presenta los servicios web implementados y utilizados por las aplicaciones móviles.

²También denominado *backend* (BE)

CAPÍTULO 2

Glosario

2.1. Glosario de términos

APK Paquete Android (*Android Package*). Es la forma principal en la que se distribuyen e instalan las aplicaciones Android. Cuando se descarga una aplicación Android de Google Play, lo que se está descargando e instalando en segundo plano es un archivo APK.

Appacovid Es el nombre otorgado al “Sistema descentralizado para detectar zonas de riesgo y contacto con personas confirmadas con COVID-19 protegiendo la privacidad de los participantes”.

App Store También conocida como tienda de aplicaciones Mac (*Mac App Store*) es una plataforma de distribución digital de aplicaciones MacOS, creada y mantenida por Apple Inc.

Código QR Código de respuesta rápida (por sus siglas en inglés *Quick Response code*) es la evolución del código de barras. El código QR permite almacenar información en una matriz de puntos o en un código de barras bidimensional. La matriz se lee en el dispositivo móvil por un lector de QR y de forma inmediata ejecuta la acción configurada, por ejemplo: llevarnos a una aplicación en internet, mostrar un mapa de localización; mostrar un correo electrónico; mostrar una página web o un perfil en una red social.

eFID Identificadores efímeros los cuales se utilizan en el *broadcast* que transmite el dispositivo vía Bluetooth.

Framework En el desarrollo de software, es un conjunto estandarizado de conceptos, prácticas y criterios enfocado en resolver un tipo de problemática particular. Convirtiéndose en una referencia, para enfrentar y resolver nuevos problemas de índole similar.

Google Play Es un servicio de distribución digital operado y desarrollado por Google. Ésta es la tienda oficial de aplicaciones Android las cuales son desarrolladas con el estuche de desarrollo de software de Android (SDK por sus siglas en inglés).

Kd Es la llave secreta para cada día, generada por el algoritmo DP3T.

UDID Identificador de Dispositivo Único (*Unique Device Identifier*). Es un elemento de los dispositivos Apple. Es un número de 40 dígitos que identifica de manera única a cada dispositivo Apple. Esto permite autenticar e instalar aplicaciones solo aprobadas por Apple.

UML Lenguaje de modelado unificado (UML por sus siglas en inglés). Es un lenguaje de modelado de propósito general en el campo de la ingeniería de Software. Su objetivo es proporcionar una forma estándar de visualizar el diseño de un sistema.

URL Localizador Uniforme de Recursos (*Uniform Resource Locator*).

2.2. Definición de abreviaturas

Las abreviaciones y claves utilizadas en este documento tienen la finalidad de identificar los elementos presentados. Las claves utilizadas son generalmente seguidas de un número (XX), el cual es utilizado para enumerar los elementos.

CU-XX Caso de uso, se utiliza para identificar los casos de uso. XX significa el número de caso de uso, por ejemplo: CU-01.

CAPÍTULO 3

Estructura del sistema

En este capítulo se describe la estructura general de Applacovid a nivel de subsistemas; los conceptos teóricos y conceptuales del protocolo de comunicación entre celulares, vía Bluetooth; los protocolos de seguridad implementados para resguardar la identidad de los usuarios; la lógica de negocio destinada a guiar el flujo de los procesos; y las decisiones finales de diseño para de desarrollar e implementar Applacovid.

3.1. Estructura general de Applacovid

Applacovid se divide en cuatro subsistemas principales, como se muestra en la Figura 3.1, los cuales interactúan entre sí para complementar la información proporcionada a los usuarios en las aplicaciones móviles y permitir una administración sencilla de los portales web.

- **Página Web:** Es la página web de Applacovid desde la cual se pueden descargar las aplicaciones móviles para iOS y Android. Además, los usuarios podrán encontrar información útil para la protección y cuidado ante COVID-19.
- **Sistema de Información Geográfico Web:** Este sistema geográfico proporciona información actualizada sobre los contagios de COVID-19 en el país y por municipio. La página web proporciona un acceso directo para la consulta de los mapas.
- **Consola de Administración:** Este sistema permite a una autoridad de salud registrar a un usuario como contagiado ante Applacovid, por medio de su código QR. También permite al administrador del sistema actualizar las noticias y preguntas frecuentes de la página web.
- **Aplicaciones móviles:** Son las aplicaciones móviles para iOS y Android de Applacovid. Es un subsistema independiente de los anteriores ya que está diseñado para responder la alta demanda de las aplicaciones móviles.

En este documento se describe la arquitectura y las decisiones de diseño finales del subsistema "aplicaciones móviles", la descripción detallada de los subsistemas restantes se puede consultar en los documentos correspondientes a cada uno (mencionados en la tabla "Documentos relacionados" del presente documento).



Figura 3.1: Subsistemas que componen a Applacovid.

3.2. Subsistema: Aplicaciones móviles

Applacovid consiste de tres modos de funcionamiento principal: Protocolo de comunicación entre dispositivos, reporte de usuario con COVID-19 y notificación de riesgo de contacto. A continuación se describe cada uno de ellos.

3.2.1. Protocolo de comunicación entre dispositivos

Cuando Applacovid se instala en el dispositivo móvil se genera una llave de seguridad llamada *Kd*. Esta llave se utiliza para generar un identificador distinto cada 15 minutos titulado "identificador efímero" (*efID*). Este identificador efímero es generado a partir de procesos criptográficos que permiten ocultar la identidad del usuario. Una vez que la aplicación comienza a generar los identificadores efímeros, estos se transmiten hacia otros dispositivos móviles cercanos que tengan Applacovid instalada. Los *efID* se transmiten vía Bluetooth entre los dispositivos y se envían junto con los saludos del protocolo Bluetooth. Este proceso se ilustra en la Figura 5.1.

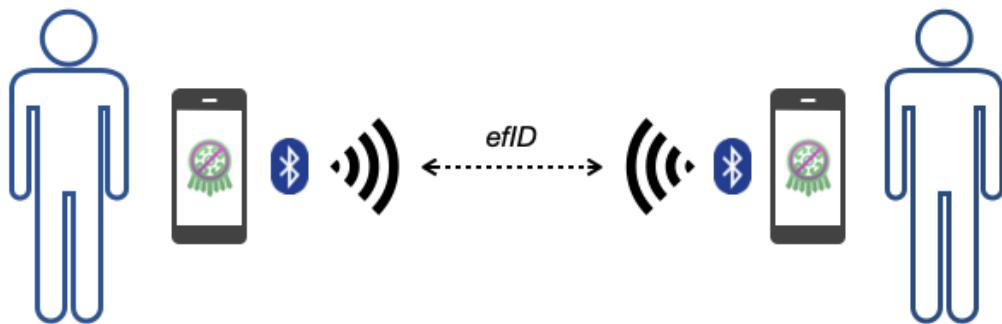


Figura 3.2: Detección de contactos vía Bluetooth.

Cuando un dispositivo recibe un saludo, vía Bluetooth, y en éste viaja el *efID*, se almacena localmente el ID recibido, el *timestamp* y la duración del encuentro. En las siguientes subsecciones se detallan los fundamentos teóricos y conceptuales aquí descritos.

3.2.2. Reporte de usuario con COVID-19

Cuando el usuario ha sido notificado positivo con COVID-19, puede reportarse de manera anónima ante Applacovid. Para esto, el usuario debe ingresar un código de seguridad, proporcionado por alguna autoridad de salud. Este código es único e intransferible. Los datos que viajan al servidor de Applacovid no contienen información personal del usuario, pero permite que otros usuarios, con la aplicación instalada, puedan ser alertados de un posible contagio, en caso de haber tenido contacto con el contagiado. El sistema garantiza la integridad y anonimato de los datos transmitidos y de la identidad real del contagiado COVID-19, respectivamente.

Para mayor detalle técnico sobre la forma en que la información viaja al servidor, se puede consultar el documento "Especificación de la plataforma" (mencionados en la tabla "Documentos relacionados" del presente documento).

3.2.3. Notificación de riesgo de contacto

Cada **dos horas** Applacovid solicita una actualización de información al servidor. El servidor responde a estas solicitudes enviando la lista de identificadores efímeros, enviados por los usuarios que se reportaron con COVID-19.

Una vez recibida esta lista, cada uno de los dispositivos deriva los identificadores de los usuarios contagiados y los compara con los que tiene en su lista local. En caso de coincidencia, y si el nivel de riesgo es alto, se emite una notificación al usuario.

3.2.4. Creación de Identificadores Efímeros

Partiendo de una llave secreta K_0 , generada aleatoriamente, se genera una llave secreta para cada día d como sigue:

$$K_d = H(K_{d-1}),$$

donde H denota una función picadillo criptográfica segura.

La seguridad y anonimato del usuario es prioritario en esta aplicación, por lo tanto el identificador cambia cada **24 horas**, dando lugar a los identificadores efímeros (*efID*), los cuales serán utilizados en el *broadcast* (o saludos) que transmite el dispositivo, vía Bluetooth.

Cada día se utiliza la correspondiente llave secreta K_d , para generar todos los *efID* que se usarán durante ese día de la siguiente forma:

$$\textit{efID}_1, \textit{efID}_2, \dots, \textit{efID}_n = \textit{PRG}(\textit{PRF}(K_d, \text{"sal"})),$$

donde *PRF* es una función pseudo aleatoria usando el código de autenticación de mensajes (MAC), mientras que *PRG* es un generador pseudo aleatorio implementado por un cifrador por bloques en modo contador. La cadena "sal" es un parámetro del sistema que todos los usuarios conocen.

3.2.5. Cálculo del riesgo de infección

Cuando un usuario se reporta positivo con COVID-19 ante Applacovid, los dispositivos que tuvieron contacto con él tienen almacenado los *efID*, la distancia y la duración del encuentro. Con base en esta información, y tomando en cuenta la opinión de los expertos del sistema de salud, se genera un algoritmo para determinar si el usuario ha sido contagiado o no. El algoritmo es el siguiente:

1. Contacto lejano y de poca duración, es decir, dos personas sólo se cruzaron caminando sin tener una interacción directa.
2. Contacto lejano y de larga duración, dos personas dentro del rango de comunicación Bluetooth pero que no tienen interacción directa.
3. Contacto cercano de larga duración, podría representar a dos personas teniendo una conversación. O bien dos personas que caminaron durante cierto tiempo manteniendo proximidad, por ejemplo, un transbordo en el metro.

Recordando que hay medidas como la sana distancia, que establece un rango de metro y medio entre personas para reducir la probabilidad de contagio, y con base a la información disponible en los dispositivos, se puede ponderar un umbral para lanzar o no una notificación de posible contagio.

3.3. Diseño del concepto

Se realizaron diversos bosquejos para seleccionar el diseño final del logotipo de la aplicación, la paleta de colores y los elementos que aparecerían en las pantallas de la aplicación móvil. El diseño final del logotipo de Applacovid y la paleta de colores, se muestran en la Figura 3.3. En la teoría del color, el morado representa calma y tranquilidad, debido a esto se eligió como color principal del producto.



Figura 3.3: Logotipo y colores de Applacovid.

La Figura 3.4 muestra cómo luce el logotipo de Applacovid como ícono de aplicación en el dispositivo móvil.

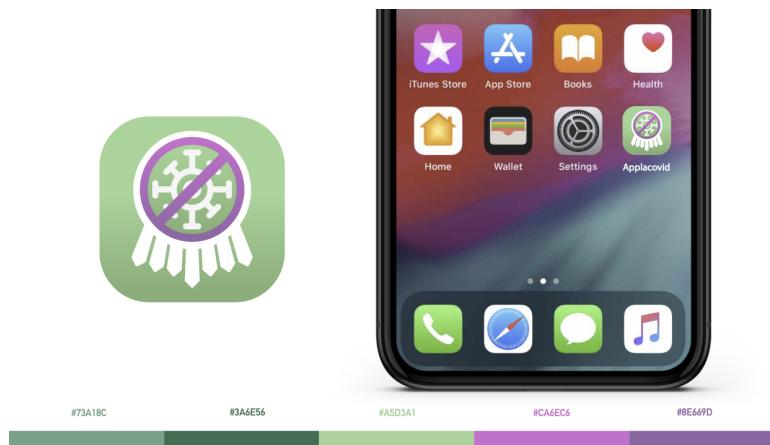


Figura 3.4: Ícono de aplicación en el dispositivo móvil.

Existen diversos personajes que acompañan al usuario durante el uso de Applacovid, estos personajes se muestran en la Figura 3.5.

Una solicitud de diseño fue que los personajes simularan interactuar en el exterior, siguiendo las recomendaciones de salud, como son: el uso de cubre bocas y mantener sana distancia. En la Figura 3.6 se muestran diversas imágenes con los personajes en el exterior siguiendo estas recomendaciones de salud.



Figura 3.5: Personajes.

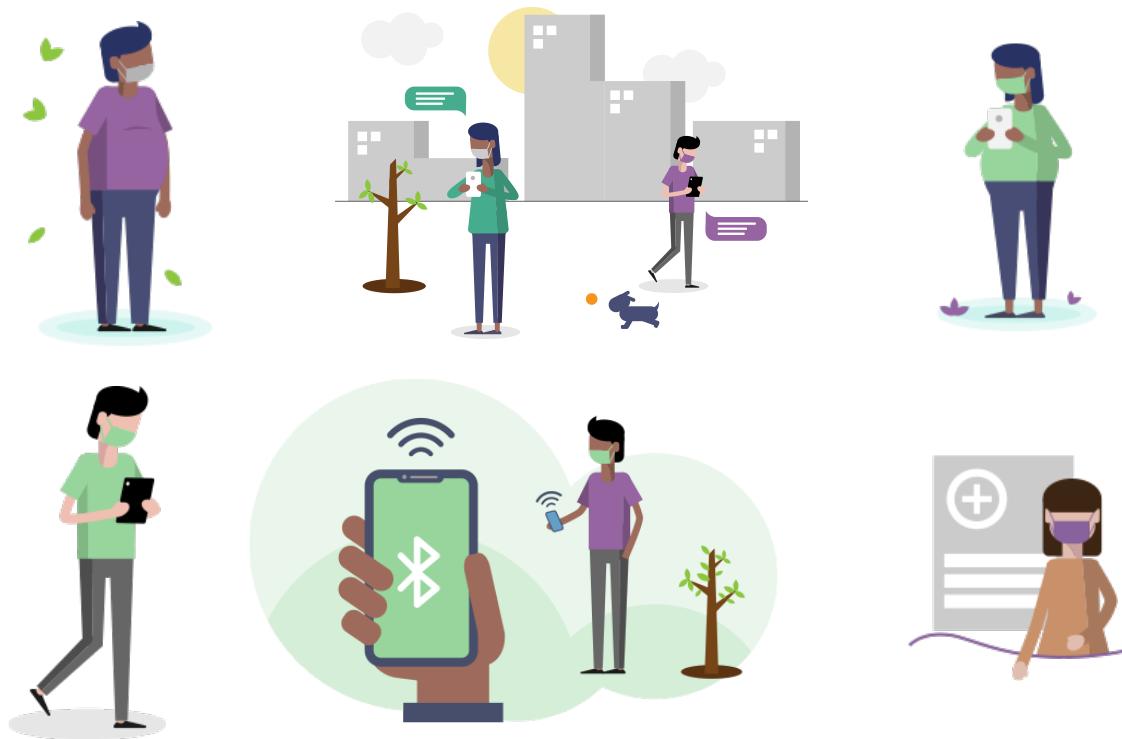


Figura 3.6: Personajes siguiendo recomendaciones de salud.

Algunos personajes nos ayudan a explicar diversas funciones de la aplicación tales como: reportarse ante Applacovid en caso de resultar positivo en la prueba COVID-19, la aplicación te notifica en caso de detectar un contacto de riesgo, la aplicación te avisa cuánto tiempo has estado expuesto y la aplicación mantiene segura privacidad. Esto se muestra en la Figura 3.7.

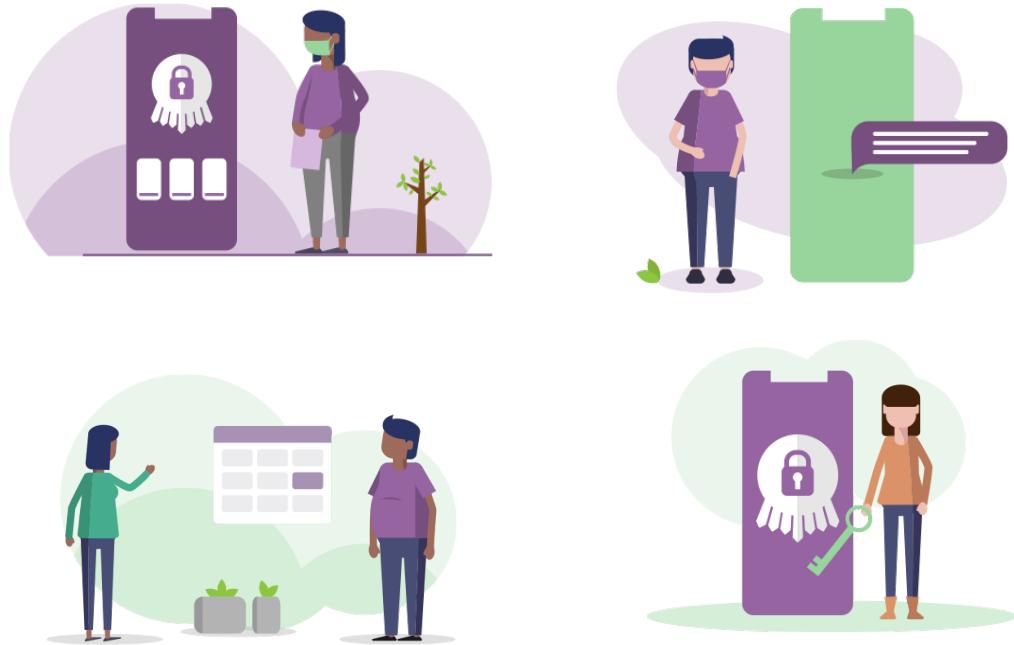


Figura 3.7: Personajes explicando funciones de la aplicación.

El objetivo principal de Applacovid es ayudar a las personas a regresar a la nueva normalidad de forma segura y tomando en cuenta las recomendaciones de salud. Este mensaje es transmitido por algunos de nuestros personajes como se muestra en las Figuras 3.8 y 3.9.



Figura 3.8: Regreso a la nueva normalidad de forma segura 1.

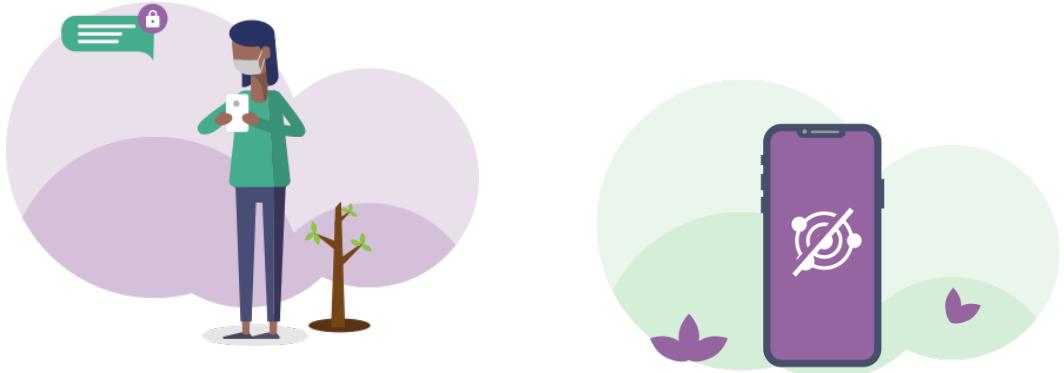


Figura 3.9: Regreso a la nueva normalidad de forma segura 2.

Algunos íconos y botones diseñados para las aplicaciones móviles de Applacovid se muestran en la Figura 3.10.



Figura 3.10: Íconos y botones de Applacovid.

Finalmente, la pantalla inicial de Applacovid en un dispositivo móvil se muestra en la Figura 3.11.

3.4. Diseño de interfaces

El diseño de interfaces para las aplicaciones móviles sigue los patrones de diseño establecidos para aplicaciones iOS y Android. Algunas restricciones de diseño son las siguientes:

- El idioma de la aplicación es español por defecto, sin posibilidad de cambiar el idioma.
- La aplicación debe guiar al usuario en todo el proceso de instalación.
- Los colores utilizados no deben estresar al usuario.

No existen diferencias entre el diseño de interfaces para iOS y Android, las únicas diferencias notables son en los componentes propios del sistema operativo, como son: botones, íconos, mensajes del sistema operativo y notificaciones del sistema operativo.

Las Figuras 3.12 y 3.13 muestran las diferencias en el formato de pantalla, el cual dependerá del dispositivo que se utilice. Algunos dispositivos presentan una pantalla mucho más alargada que otros, o mucho más ancha que otros. La Figura 3.14 muestra las diferencias notables entre los dispositivos iOS y Android, siendo estos únicamente a nivel de botones e íconos. La Figura 3.15 muestra la forma



Figura 3.11: Pantalla inicial de Applacovid en dispositivos móviles.

de desplegar los mensajes en un sistema iOS y un Android. El diseño de estos mensajes no se puede estandarizar, debido a que son parte del sistema operativo y éste es el encargado de crearlos.

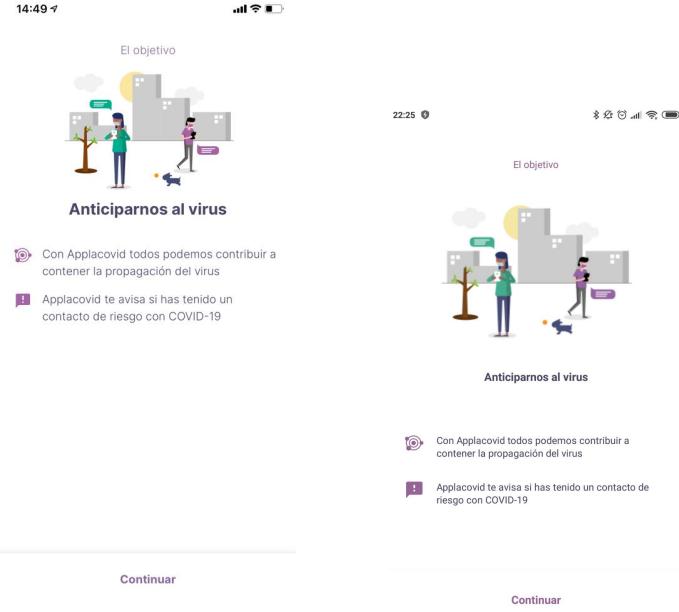


Figura 3.12: Diseño para iOS (izquierda) y Android (derecha).

Las aplicaciones solo utilizan el módulo de Bluetooth para comunicarse con los dispositivos a su alrededor. Sin embargo, debido a la arquitectura del módulo Bluetooth en dispositivos Android, el GPS debe ser activado para poder utilizar el Bluetooth, no es posible desvincular ambos módulos ya que estos módulos son parte del sistema operativo, lo cual hace imposible su modificación.

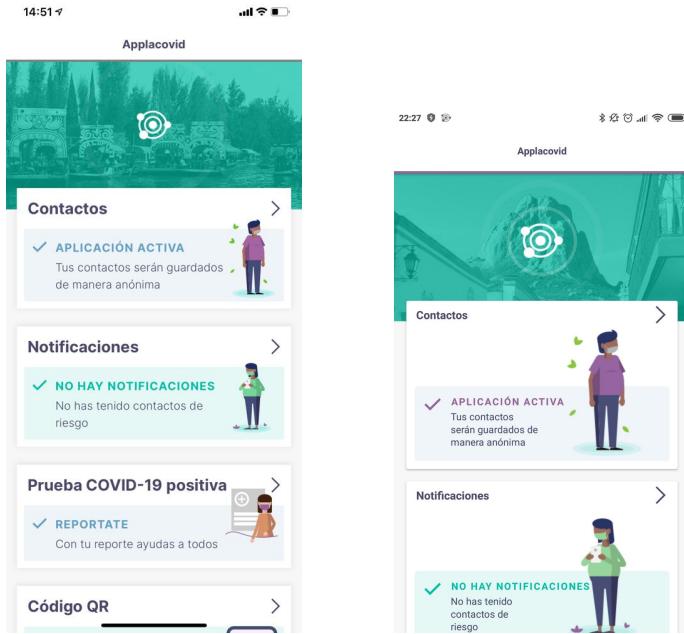


Figura 3.13: Formato de pantalla, iOS (izquierda) y Android (derecha).

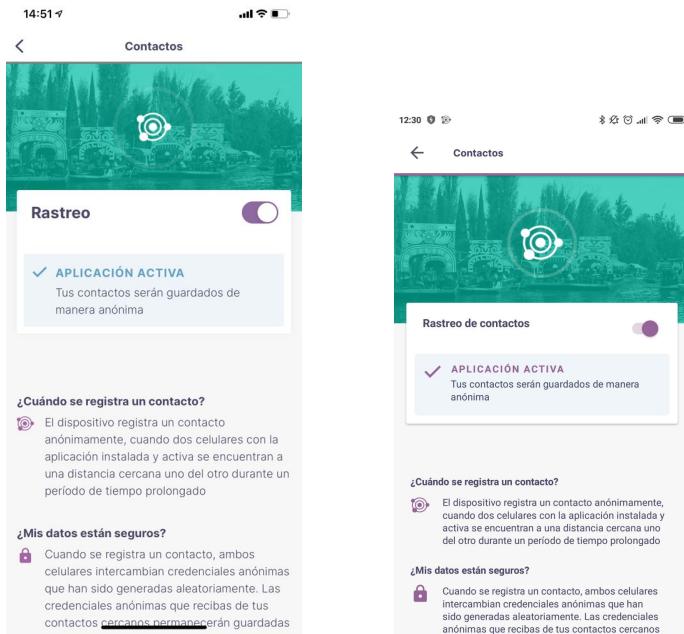


Figura 3.14: Botones e íconos, iOS (izquierda) y Android (derecha).

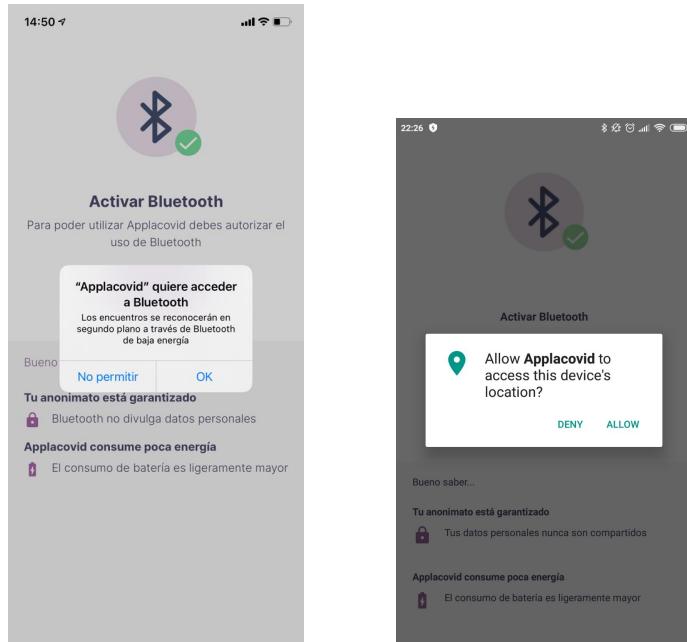


Figura 3.15: Mensajes del sistema operativo, iOS (izquierda) y Android (derecha).

CAPÍTULO 4

Requerimientos del sistema

En este capítulo se describen los requerimientos funcionales y no-funcionales de Applacovid. Los requerimientos funcionales se describen por medio de casos de uso, los cuales tienen como objetivo, mostrar paso a paso la funcionalidad de la aplicación y la interacción de ésta con el usuario final.

Los casos de uso están conformados por diferentes elementos, los cuales nos ayudan a describir de mejor manera la información que fluye a través de los mismos y las condiciones en las que el sistema debe encontrarse para ejecutar correctamente el caso de uso. El número de elementos en cada caso de uso puede variar, dependiendo del propósito del mismo, por ejemplo, hay algunos casos que solo muestran información y no necesitan datos de entrada por parte del usuario o de algún actuador, estos elementos son:

Elementos de un caso de uso

- **Objetivo:** Breve descripción del propósito del caso de uso.
- **Entradas:** Datos de entrada requeridos para la ejecución del caso de uso.
- **Salidas:** Datos de salida que genera el sistema después de la ejecución del caso de uso.
- **Precondiciones:** Condiciones iniciales que debe cumplir el sistema para ejecutar el caso de uso.
- **Postcondiciones:** Condiciones en las que estará el sistema después de ejecutar el caso de uso.
- **Trayectorias:** Secuencia de pasos que ejecutará el caso de uso para obtener una salida. Las trayectorias pueden ser: principal u óptima; y trayectorias alternas.

4.1. Requerimientos funcionales

Los requerimientos funcionales son aquellos que describen lo que debe hacer el sistema, cómo debe lucir, la información de entrada que debe recibir, cómo debe procesar dicha información y cuál es el resultado que se espera después de dicho procesamiento.

Los casos de uso son parte del lenguaje UML y son una forma de visualizar los requerimientos funcionales del sistema. Los casos de uso nos ayudarán a describir el comportamiento de la aplicación en dispositivos móviles iOS y Android, éstos se muestran en la Figura 4.1.

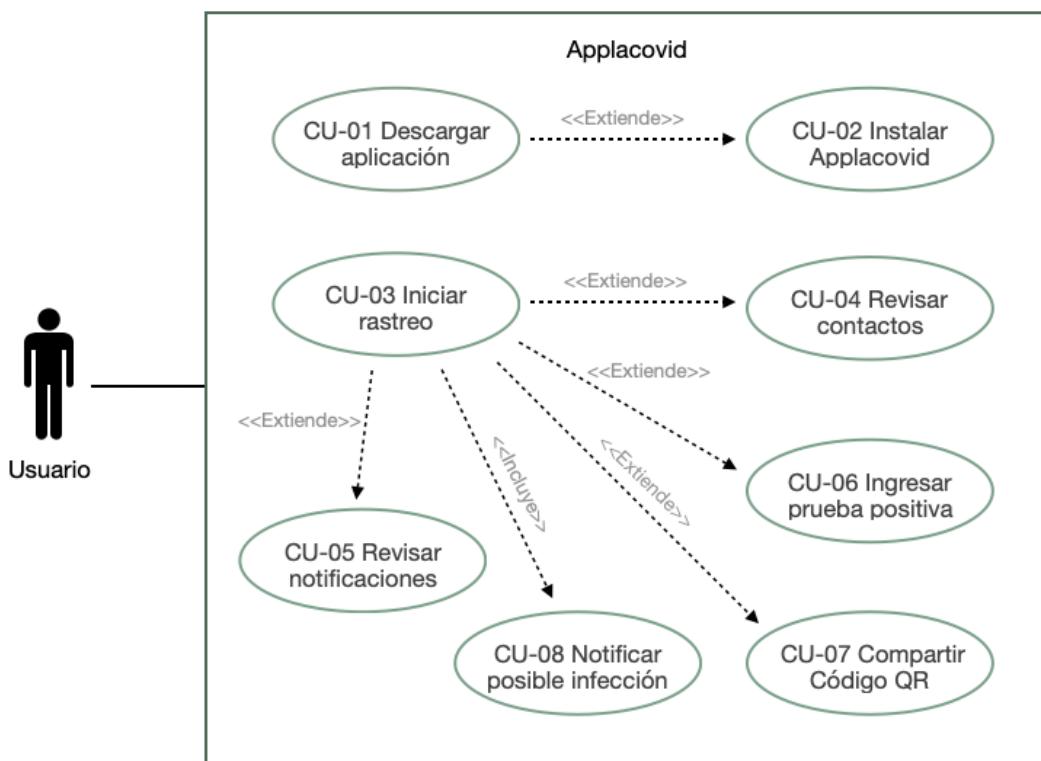


Figura 4.1: Casos de uso.

- CU-01 Descargar aplicación: Muestra el proceso de descarga de la aplicación.
- CU-02 Instalar Applacovid: Describe el proceso para instalar la aplicación.
- CU-03 Iniciar rastreo: Describe el proceso para iniciar el rastreo de contactos.
- CU-04 Revisar contactos: Permite al usuario revisar el estado de la aplicación.
- CU-05 Revisar notificaciones: Permite al usuario revisar si ha tenido contactos cercanos con COVID-19.
- CU-06 Ingresar prueba positiva: Permite al usuario registrarse de manera anónima como contagiado con COVID-19 ante Applacovid.

- CU-07 Compartir código QR: Permite al usuario generar el código QR para compartirlo con una autoridad de salud.
- CU-08 Notificar posible infección: Notifica al usuario cuando se ha detectado un contacto de riesgo con COVID-19.

Para facilitar la lectura del documento, solo se muestran las interfaces gráficas para dispositivos Android, como se explicó en el Capítulo 3 Sección 3.4, las diferencias notables entre el diseño de interfaces iOS y Android solo se encuentra a nivel de íconos, botones y mensajes del sistema operativo.

4.2. CU-01 Descargar aplicación

4.2.1. Objetivo

Describir los pasos a seguir para descargar Applacovid en el dispositivo móvil del usuario. La aplicación está disponible para dispositivos iOS y Android.

4.2.2. Atributos

Caso de Uso:	CU-01 Descargar aplicación
Entradas:	<ul style="list-style-type: none">Página web pikal.cs.cinvestav.mx de Applacovid.Para el caso de un dispositivo iOS, se requiere el nombre de usuario, el UDID del teléfono móvil y el correo electrónico del usuario.
Salidas:	<ul style="list-style-type: none">Para dispositivos Android se obtendrá un archivo APK.Para dispositivos iOS se obtendrá un correo electrónico con la URL de descarga de la aplicación.
Precondiciones:	<ul style="list-style-type: none">Los dispositivos Android deben tener un sistema operativo versión 6 Marshmallow o superior.Los dispositivos iOS deben tener un sistema operativo versión 13 o superior.

4.2.3. Trayectorias del Caso de Uso

Trayectoria principal

- 1 Ingresa a la página web pikal.cs.cinvestav.mx de Applacovid para descargar la aplicación (ver pantalla 4.2).
- 2 Selecciona los pasos a seguir dependiendo del tipo de dispositivo que posea:
 - Para el caso de dispositivos Android selecciona el botón “Descarga” en la pantalla 4.3. [Trayectoria A]
 - Para el caso de dispositivos iOS, selecciona el botón “Formulario” en la pantalla 4.4. [Trayectoria B]

- - - - *Fin del caso de uso.*

Trayectoria alternativa A:

Condición: *El dispositivo es Android.*

- A-1 Proporciona el archivo APK de Applacovid.
 - A-2 Autoriza la instalación de Applacovid en su dispositivo móvil.
- - - - *Fin de trayectoria.*

Trayectoria alternativa B:

Condición: *El dispositivo es iOS.*



Figura 4.2: Página web de Applacovid.

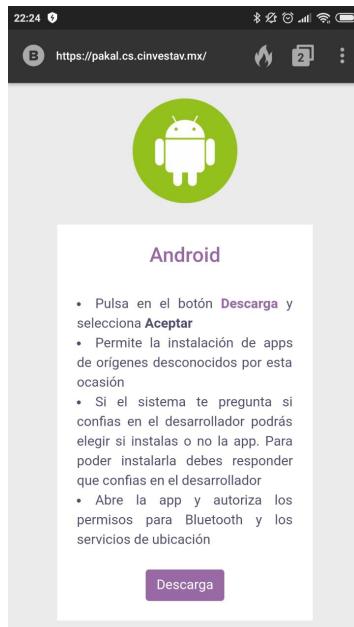


Figura 4.3: Descargar Applacovid para Android.

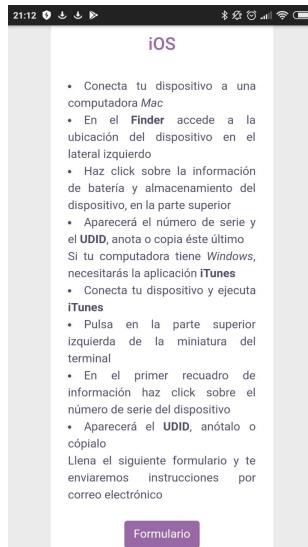


Figura 4.4: Descargar Applacovid para iOS.

- B-1** Muestra el formulario para dispositivos iOS, como se muestra en la pantalla 4.5.
- B-2** Ingresa su nombre de usuario, el **UDID** del teléfono móvil y su correo electrónico.
- B-3** Envía un correo electrónico con la **URL** de descarga de la aplicación.
- B-4** Autoriza la instalación de Applacovid en su dispositivo móvil.

- - - - *Fin de trayectoria.*

Formulario de Registro

Datos del Dispositivo

Usuario:

UDID:

Correo:

Enviar

Figura 4.5: Formulario para descarga de Applacovid.

4.3. CU-02 Instalar Applacovid

4.3.1. Objetivo

Guia al usuario en la instalación de Applacovid en su dispositivo móvil Android o iOS.

4.3.2. Atributos

Caso de Uso:	CU-02 Instalar Applacovid
Entradas:	<ul style="list-style-type: none">Para dispositivos Android es necesario tener el archivo APK.Para dispositivos iOS el correo electrónico con la URL de descarga de la aplicación.
Salidas:	<ul style="list-style-type: none">Applacovid funcionando en el dispositivo móvil.
Precondiciones:	<ul style="list-style-type: none">Para dispositivos Android tener encendido el Bluetooth y GPS al momento de la instalación.Para dispositivos iOS tener encendido el Bluetooth al momento de la instalación.
Postcondiciones:	<ul style="list-style-type: none">El Bluetooth permanecerá encendido en todo momento, para el correcto funcionamiento de la aplicaciónPara dispositivos Android permanecerá encendido el GPS.El consumo de energía aumentará ligeramente durante el uso de Applacovid en los dispositivos móviles.

4.3.3. Trayectorias del Caso de Uso

Trayectoria principal

- 1 Abre la aplicación.
- 2 Muestra la pantalla [4.6](#) para iniciar la instalación.
- 3 Selecciona el botón “Instalar” en la pantalla [4.6](#).
- 4 Inicia la instalación de la aplicación mostrando la pantalla [4.7](#).
- 5 Despues de finalizar la instalación muestra la pantalla [4.8](#).
- 6 Selecciona el botón “Abrir” en la pantalla [4.8](#).
- 7 Muestra el objetivo de la aplicación en la pantalla [4.9](#).
- 8 Revisa la información y da clic en el botón “Continuar” en la pantalla [4.9](#).
- 9 Muestra la pantalla [4.10](#) en la que se informa al usuario sobre la forma de recolectar y proteger la información de sus contactos.
- 10 Da clic en el botón “Continuar” de la pantalla [4.10](#).
- 11 Muestra la pantalla [4.11](#) con información acerca de como funciona Applacovid.
- 12 Da clic en el botón “Continuar” de la pantalla [4.11](#).
- 13 Muestra la pantalla [4.12](#) solicitando al usuario permisos para utilizar el Bluetooth del dispositivo.
- 14 Autoriza el uso del Bluetooth dando clic en el botón “Autorizar” de la pantalla [4.12](#).

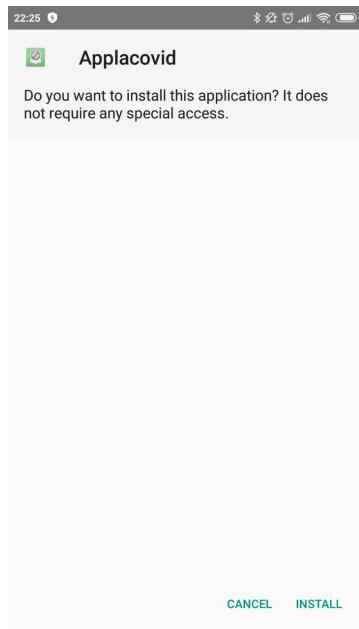


Figura 4.6: Iniciar instalación.

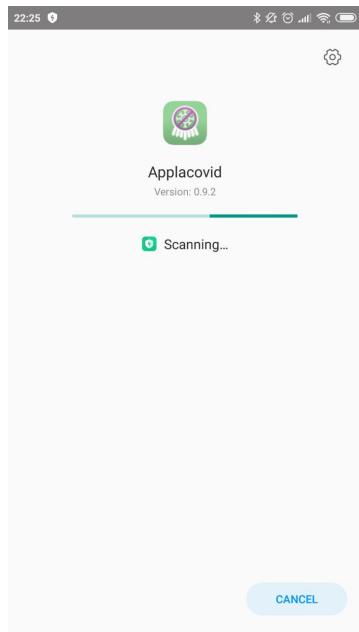


Figura 4.7: Instalando aplicación.

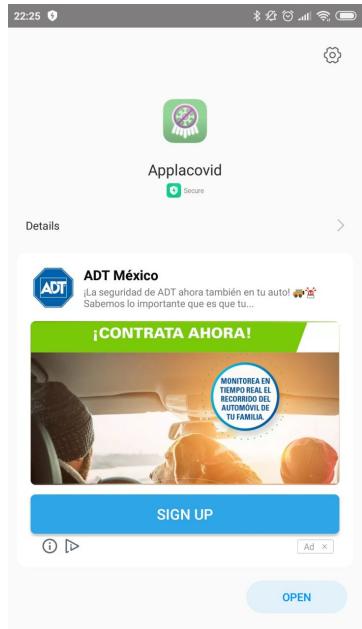


Figura 4.8: Finaliza instalación.



Figura 4.9: Objetivo.



Figura 4.10: Protegiendo tu privacidad.



Figura 4.11: ¿Cómo funciona?

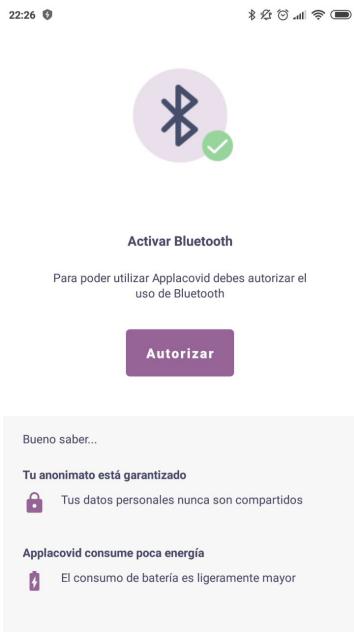


Figura 4.12: Activar Bluetooth.

- Para el caso de dispositivos Android continua en la [Trayectoria A].
- 15 ⓘ Muestra la pantalla 4.13 para informar al usuario que el consumo de la batería será ligeramente mayor y por lo tanto ignorar su optimización.
 - 16 ⓘ Da clic en el botón “Autorizar” de la pantalla 4.13.
 - 17 ⓘ Muestra la pantalla 4.14 solicitando la autorización para ignorar la optimización de batería.
 - 18 ⓘ Da clic en el botón “Sí” de la pantalla 4.14.
 - 19 ⓘ Accede al Bluetooth, GPS y apaga la optimización de batería (en caso de estar habilitada), mientras tanto muestra la pantalla 4.15 con información adicional de Applacovid.
 - 20 ⓘ Da clic en el botón “Continuar” de la pantalla 4.15.
 - 21 ⓘ Muestra la pantalla 4.16 para informar al usuario que la aplicación ha sido instalada correctamente.
 - 22 ⓘ Da clic en el botón “Comenzar” de la pantalla 4.16.
 - 23 ⓘ Realiza las siguientes acciones:
 - Crea la base de datos local en el dispositivo.
 - Genera la *Kdpor* día y la almacena en la base de datos local.
 - Generar los *efID* (sobre escribiendo el anterior) cada 15 minutos.
 - Comienza a enviar saludos cada minuto vía Bluetooth, con la finalidad de encontrar dispositivos a su alrededor. Al enviar un saludo se envía el *efID* actual.

- - - - *Fin del caso de uso.*

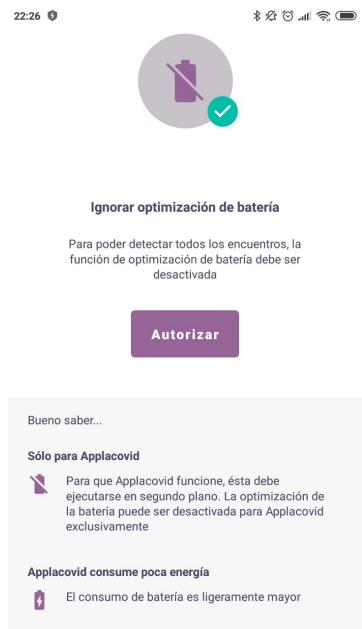


Figura 4.13: Ignorar optimización de batería.

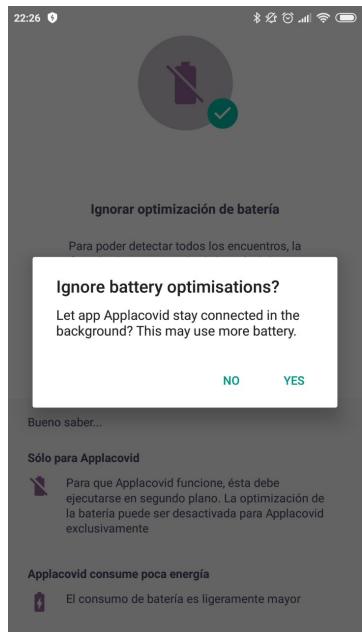


Figura 4.14: Autorizar ignorar optimización de batería.

22:26 ⓘ ⚡ ☰ ☰ 🔍

¿Qué hace Applacovid?



Notificación de un contacto riesgoso

! Applacovid te avisa si has estado en contacto cercano con una persona contagiada de COVID-19

Continuar

Figura 4.15: ¿Qué hace Applacovid?

22:26 ⓘ ⚡ ☰ ☰ 🔍

La aplicación está instalada

Gracias por ayudar a romper las cadenas de contagio

Comenzar

Figura 4.16: Aplicación instalada.

Trayectoria alternativa A:

Condición: Autorizar uso de GPS.

- A-1  Muestra la pantalla 4.17 solicitando la autorización para el uso del GPS.

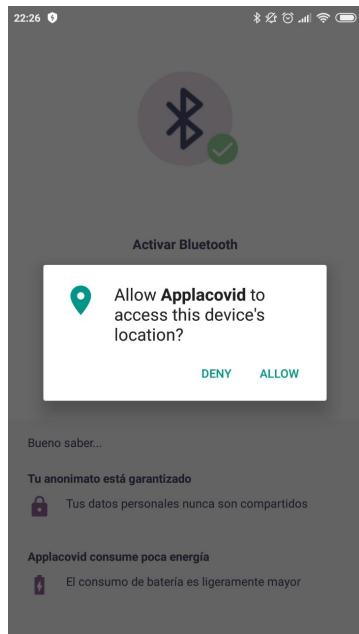


Figura 4.17: Autorizar uso de GPS.

- A-2  Autoriza el uso del GPS dando clic en el botón “Permitir” en la pantalla 4.13.

- A-3 Continúa en la trayectoria principal.

- - - - Fin de trayectoria.

4.4. CU-03 Iniciar rastreo

4.4.1. Objetivo

Una vez instalada la aplicación el dispositivo comenzará a enviar saludos vía Bluetooth, con la finalidad de descubrir dispositivos cercanos y revisar si el usuario ha tenido un contacto de riesgo con COVID-19. El usuario puede navegar en las opciones de menú que Applacovid posee para mantenerse informado de las notificaciones, así como revisar la información que ésta proporciona acerca de COVID-19.

4.4.2. Atributos

Caso de Uso:	CU-03 Iniciar rastreo
Entradas:	<ul style="list-style-type: none">Tener instalada Applacovid en el dispositivo móvil.
Salidas:	<ul style="list-style-type: none">En caso de que el dispositivo descubra algún contacto de riesgo con COVID-19, la aplicación despliega una notificación.
Precondiciones:	<ul style="list-style-type: none">Para dispositivos Android tener encendido el Bluetooth y GPS en todo momento.Para dispositivos iOS tener encendido el Bluetooth en todo momento.
Postcondiciones:	<ul style="list-style-type: none">El consumo de energía aumentará ligeramente durante el uso de Applacovid.

4.4.3. Trayectorias del Caso de Uso

Trayectoria principal

- 1  Muestra la pantalla 4.18 con el menú principal de Applacovid.
- 2  Navega en el menú principal 4.18 seleccionando alguna de las siguientes opciones:
 - **Contactos:** En esta opción el usuario puede revisar que la aplicación esté funcionando correctamente.
 - **Notificaciones:** Muestra si ha habido algún contacto de riesgo con COVID-19.
 - **Prueba COVID-19 positiva.** Permite al usuario registrarse anónimamente en la aplicación como contagiado con COVID-19.
 - **Código QR.** Puede generar su código QR para compartirlo con una autoridad de salud.

- - - - *Fin del caso de uso.*

4.4.4. Puntos de extensión

Selecciona la opción “Contactos”.

Origen: Paso 2.

Extiende a: CU-04 Revisar contactos.

Selecciona la opción “Notificaciones”.

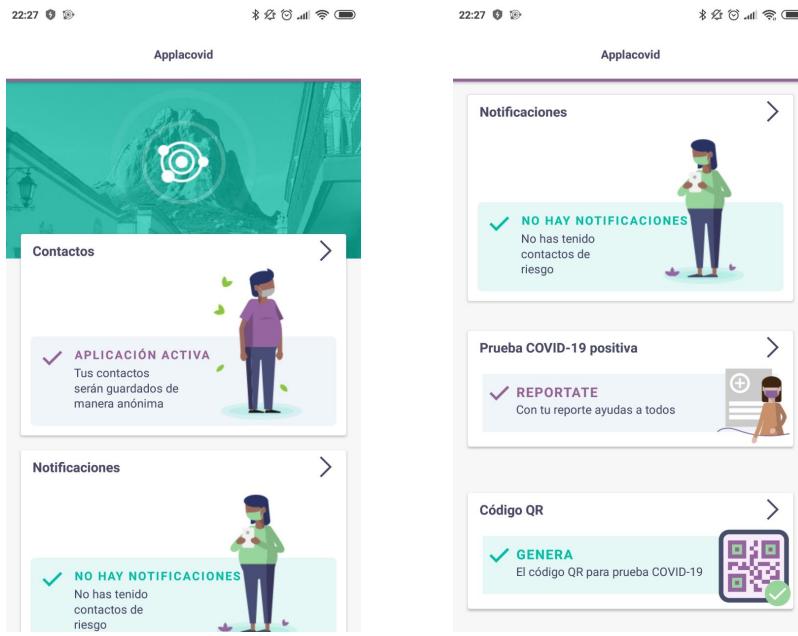


Figura 4.18: Inicio de Applacovid.

Origen: Paso 2.

Extiende a: CU-05 Revisar notificaciones.

Selecciona la opción “Prueba COVID-19 positiva”.

Origen: Paso 2.

Extiende a: CU-06 Ingresar prueba positiva.

Selecciona la opción “Código QR”.

Origen: Paso 2.

Extiende a: CU-07 Compartir código QR.

4.5. CU-04 Revisar contactos

4.5.1. Objetivo

El usuario puede revisar que la aplicación esté funcionando correctamente o puede apagar el rastreo en cualquier momento desde esta opción.

4.5.2. Atributos

Caso de Uso:	CU-04 Revisar contactos
Entradas:	<ul style="list-style-type: none">Mientras la aplicación esté activa, puede recibir saludos de otros dispositivos vía Bluetooth.
Salidas:	<ul style="list-style-type: none">Mientras la aplicación esté activa, puede enviar saludos a otros dispositivos.La aplicación se comunica con el <i>backend</i> cada dos horas, para recibir actualizaciones sobre contactos de riesgo.
Precondiciones:	<ul style="list-style-type: none">Para dispositivos Android tener encendido el Bluetooth y GPS en todo momento.Para dispositivos iOS tener encendido el Bluetooth en todo momento.
Postcondiciones:	<ul style="list-style-type: none">El consumo de energía aumentará ligeramente durante el uso de Applacovid.

4.5.3. Trayectorias del Caso de Uso

Trayectoria principal

- 1 Selecciona la opción “Contactos” en el menú principal 4.18.
 - 2 Muestra la pantalla 4.19 con información sobre el rastreo de contactos. [Trayectoria A]
 - 3 Presiona el botón en la esquina superior izquierda de la aplicación para regresar al menú principal 4.18.
- - - - *Fin del caso de uso.*

Trayectoria alternativa A:

Condición: Apagar rastreo.

- A-1 Presiona el botón para apagar el rastreo.
 - A-2 Muestra la pantalla 4.20 notificando al usuario que apagó el rastreo y que la aplicación dejará de funcionar correctamente.
 - A-3 Presiona el botón para regresar al menú principal 4.18.
 - A-4 Muestra el menú principal 4.21 indicando que la aplicación no está funcionando correctamente.
- - - - *Fin de trayectoria.*

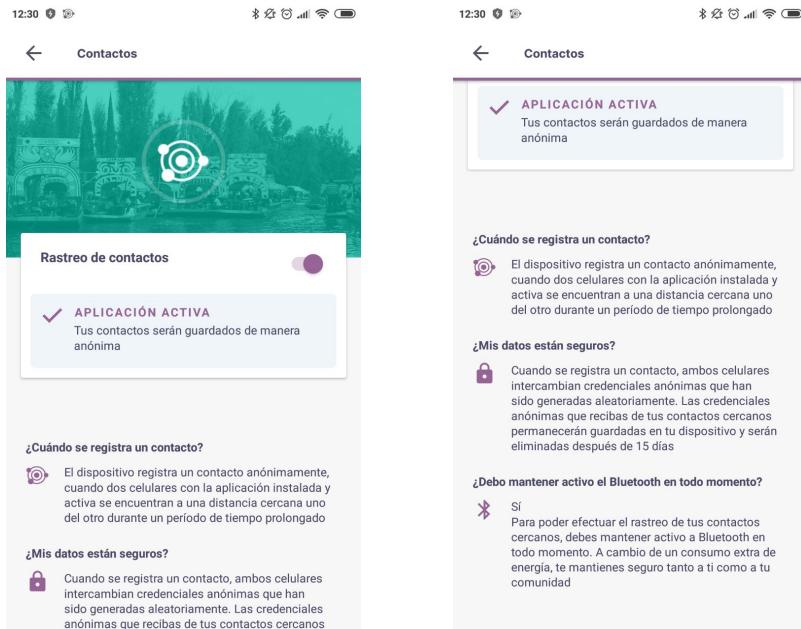


Figura 4.19: Opción “Contactos” .

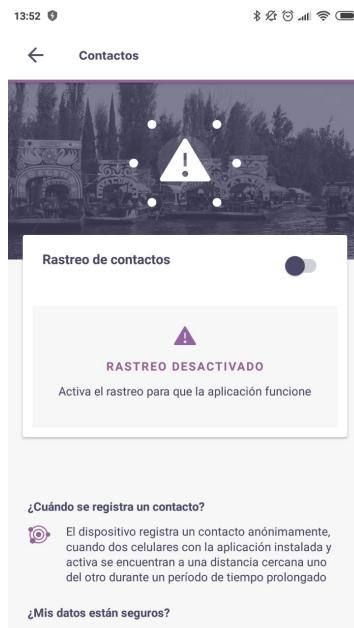


Figura 4.20: Rastreo apagado.

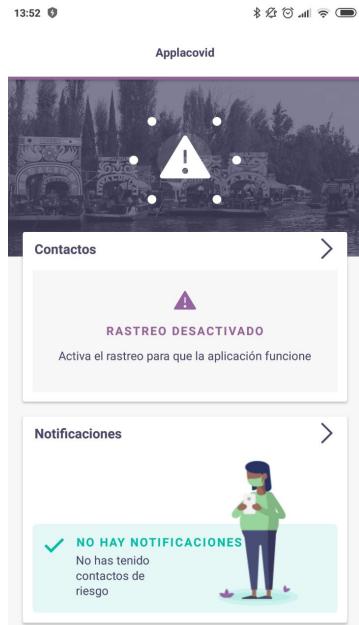


Figura 4.21: Menú principal con rastreo apagado.

4.6. CU-05 Revisar notificaciones

4.6.1. Objetivo

Por medio de esta opción el usuario puede revisar, en cualquier momento, si la aplicación ha detectado algún contacto de riesgo con COVID-19. Esto permite al usuario tomar acciones para evitar contagiar a su familia, amigos y gente a su alrededor. En esta opción también puede encontrar algunas recomendaciones vigentes de protección ante el COVID-19.

4.6.2. Atributos

Caso de Uso:	CU-05 Revisar notificaciones
Entradas:	<ul style="list-style-type: none"> Mientras la aplicación esté activa, puede recibir saludos de otros dispositivos vía Bluetooth.
Salidas:	<ul style="list-style-type: none"> Mientras la aplicación esté activa, puede enviar saludos a otros dispositivos. La aplicación se comunica con el <i>backend</i> cada dos horas, para recibir actualizaciones sobre contactos de riesgo.
Precondiciones:	<ul style="list-style-type: none"> Para dispositivos Android tener encendido el Bluetooth y GPS en todo momento. Para dispositivos iOS tener encendido el Bluetooth en todo momento.

Caso de Uso:	CU-05 Revisar notificaciones
Postcondiciones:	<ul style="list-style-type: none"> El consumo de energía aumentará ligeramente durante el uso de Applacovid.

4.6.3. Trayectorias del Caso de Uso

Trayectoria principal

- 1 ♂ Selecciona la opción “Notificaciones” en el menú principal 4.18.
- 2 ○ Se comunica con el *backend* cada dos horas para descargar la lista de *Kd* correspondiente a los contagios COVID-19 registrados. [Trayectoria A]
- 3 ○ En caso de no encontrar contactos de riesgo, muestra la pantalla 4.22 para informar al usuario que no ha tenido contactos de riesgo COVID-19. [Trayectoria B]

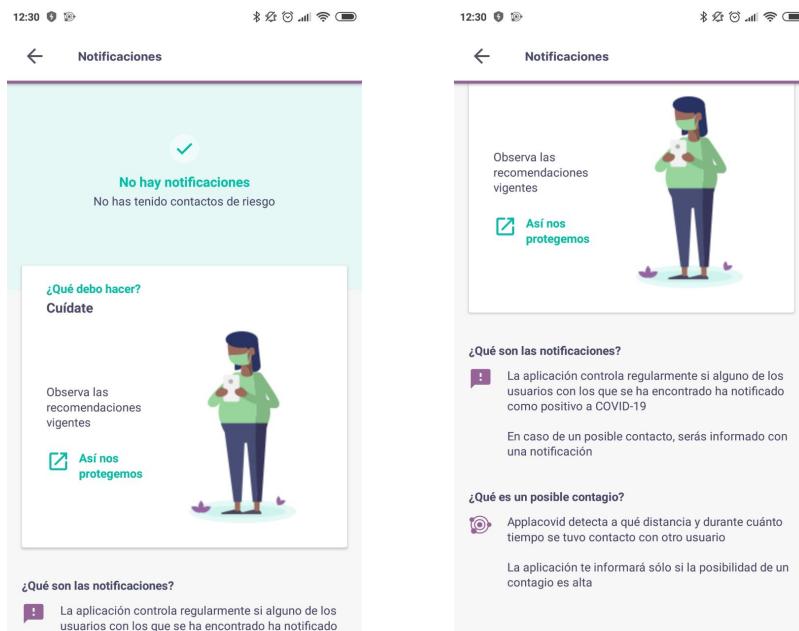


Figura 4.22: Opción “Notificaciones” .

- 4 ♂ Presiona el botón ← en la esquina superior izquierda de la aplicación para regresar al menú principal 4.18.

- - - - *Fin del caso de uso.*

Trayectoria alternativa A:

Condición: Notificaciones de Applacovid.

- A-1 ○** Applacovid muestra notificaciones cada 2 días, 6 días y 7 días. Las notificaciones se despliegan de distinta manera como se muestra en las pantallas 4.23 y 4.24. De esta manera el usuario se asegura de que Applacovid sigue trabajando correctamente.



Figura 4.23: *Push notifications* con pantalla bloqueada.



Figura 4.24: Barra de notificaciones del dispositivo.

- - - - *Fin de trayectoria.*

Trayectoria alternativa B:

Condición: Visitar el sitio web.

B-1  Presiona el ícono  “Así nos protegemos” para revisar algunas instrucciones de cuidado ante el COVID-19.

B-2  Sale de la Applacovid y abre el navegador predeterminado del dispositivo para conectarse a la página web de Applacovid, mostrada en la pantalla [4.2](#).

- - - - *Fin de trayectoria.*

4.7. CU-06 Ingresar prueba positiva

4.7.1. Objetivo

En caso de que el usuario de positivo a la prueba COVID-19, suministrada por una autoridad de salud, puede reportarse anónimamente en la aplicación por medio de esta opción. Con esta acción ayuda a detener las cadenas de contagio y permite que la comunidad de usuarios Applacovid pueda rastrear contactos contagiados. La información del usuario jamás es leída ni compartida por la aplicación, lo único que se envía al *backend*, son los identificadores anónimos (*efID*) generados por Applacovid.

4.7.2. Atributos

Caso de Uso:	CU-06 Ingresar prueba positiva
Entradas:	<ul style="list-style-type: none">Código COVID-19 de doce dígitos, previamente suministrado por una autoridad de salud.
Salidas:	<ul style="list-style-type: none">Envía al <i>backend</i> la <i>Kd</i> del día, el <i>timestamp</i>, y el código COVID-19, a través del endpoint <code>POST /v1/exposed/</code>
Precondiciones:	<ul style="list-style-type: none">Para dispositivos Android tener encendido el Bluetooth y GPS en todo momento.Para dispositivos iOS tener encendido el Bluetooth en todo momento.El dispositivo debe estar conectado a una red de internet o tener acceso a la red del dispositivo.
Postcondiciones:	<ul style="list-style-type: none">Applacovid deja de utilizar el Bluetooth del dispositivo.Para dispositivos Android, se detiene el uso del GPS.

4.7.3. Trayectorias del Caso de Uso

Trayectoria principal

- 1 Selecciona la opción “Prueba COVID-19 positiva” en el menú principal 4.18.
- 2 Muestra la pantalla 4.25 para recordar al usuario que su reporte es anónimo.
- 3 Presiona el botón “Introducir el código covid” en la pantalla 4.25.
- 4 Muestra la pantalla 4.26 informando al usuario que a pesar de ser un reporte anónimo es posible que alguna persona recuerde haber tenido contacto con el usuario.
- 5 Presiona el botón “Entendido” en la pantalla 4.26.
- 6 Muestra la pantalla 4.27 con el formulario para ingresar el código COVID-19.
- 7 Ingresa el código COVID-19 suministrado por una autoridad de salud y presiona el botón “Enviar” .
- 8 Envía al *backend* la *Kd* del día, el *timestamp*, y el código COVID-19 para registrarlos en la base de datos, a través del endpoint `POST /v1/exposed/`. Los dispositivos que realicen una petición al servidor en las siguientes dos horas recibirán el *Kd* del usuario contagiado.
- 9 Muestra la pantalla 4.28 para agradecer la contribución del usuario para detener las cadenas de contagio.

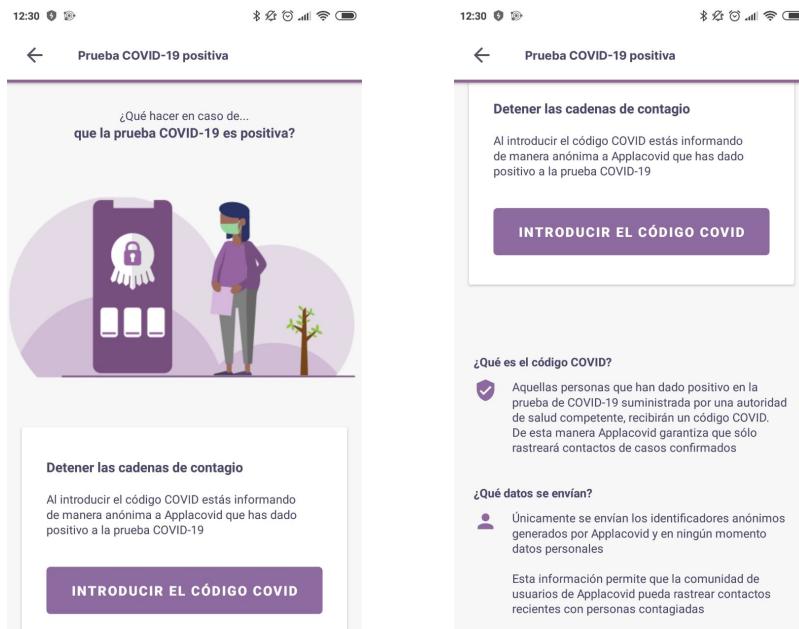


Figura 4.25: Detener las cadenas de contagio.



Figura 4.26: Aviso importante.

18:08 ④ 🔋

Cancelar

Introduce el código Covid

Introduce el código Covid recibido



Enviar

1	2	3	-
4	5	6	—
7	8	9	✖
,	0	.	→

Figura 4.27: Introduce código COVID-19.

16:14 ④ 🔋



¡Muchas gracias!

Los contactos cercanos que tuviste en los últimos días serán informados de manera inmediata preservando tu anonimato

Has hecho una muy valiosa contribución hacia el control y reducción de cadenas de contagio

Continuar

Figura 4.28: Muchas gracias.

- 10  Presiona el botón “Continuar” en la pantalla 4.28.

11  Muestra la pantalla 4.29 para informar al usuario que el rastreo de Applacovid termina después de hacer un reporte anónimo.



Figura 4.29: Rastreo terminado.

- 12  Presiona el botón “Continuar” en la pantalla 4.29.

13  Muestra la pantalla 4.30 para desecharle pronta mejoría y alentarlo a seguir las instrucciones recibidas por la autoridades de salud.

14  Presiona el botón “Cerrar” en la pantalla 4.30.

15  Muestra la pantalla 4.31 con el menú final de Applacovid. En este momento el rastreo de ha terminado y la aplicación detiene:

16 El envío de saludos, vía Bluetooth, a dispositivos cercanos.

17 Para dispositivos Android, detiene el uso del GPS.

18 Applacovid deja de producir la llave Kd por día y los $efID$ cada 15 minutos.

19  Revisa la pantalla 4.31 para verificar que el rastreo ha terminado. [Trayectoria A]

- - - - - Fin del caso de uso.

Trayectoria alternativa A:

Condición: Selecciona la opción “Notificaciones”

- A-1**  Selecciona la opción “Notificaciones” en el menú final de Applacovid 4.31.

A-2  Muestra la pantalla 4.32 para conocer las medidas necesarias para realizar un auto aislamiento personal. [Trayectoria B]

- - - - *Fin de trayectoria.*



Figura 4.30: Te deseamos pronta mejoría.



Figura 4.31: Prueba positiva.



Figura 4.32: Auto aislamiento personal.

Trayectoria alternativa B:

Condición: Visitar el sitio web.

- B-1** Presiona el ícono “Instrucciones para el auto aislamiento” para revisar algunas medidas de salud ante el COVID-19.
- B-2** Sale de la Applacovid y abre el navegador predeterminado del dispositivo para conectarse a la página web de Applacovid, mostrada en la pantalla 4.2.
- - - - *Fin de trayectoria.*

4.8. CU-07 Compartir código QR

4.8.1. Objetivo

En caso de que el usuario resulte positivo en su prueba COVID-19, puede aceptar que una autoridad de salud lo registre como contagiado ante Applacovid. Para ello debe compartir su código QR, generado por la aplicación, el cual contiene la semilla con la que se generaron los mensajes aleatorios los últimos 5 días.

Con el QR la autoridad de salud puede reportar ante Applacovid un caso de contagio más. El QR no comparte información personal del usuario ni del dispositivo.

4.8.2. Atributos

Caso de Uso:	CU-07 Compartir código QR
Entradas:	<ul style="list-style-type: none">Mientras la aplicación esté activa, puede recibir saludos de otros dispositivos, vía Bluetooth.
Salidas:	<ul style="list-style-type: none">Una imagen con el código QR, la cual contiene: la Kd del día, el UDID de la aplicación y el <i>timestamp</i> asociado a la Kd.Mientras la aplicación esté activa, puede enviar saludos a otros dispositivos.La aplicación se comunica con el <i>backend</i> cada dos horas, para recibir actualizaciones sobre contactos de riesgo.
Precondiciones:	<ul style="list-style-type: none">Para dispositivos Android tener encendido el Bluetooth y GPS en todo momento.Para dispositivos iOS tener encendido el Bluetooth en todo momento.
Postcondiciones:	<ul style="list-style-type: none">El consumo de energía aumentará ligeramente durante el uso de Applacovid.

4.8.3. Trayectorias del Caso de Uso

Trayectoria principal

- 1 Selecciona la opción “Código QR” en el menú principal 4.18.
- 2 Muestra la pantalla 4.33 con el código QR listo para compartir.
- 3 Presiona el botón “Compartir” en la pantalla 4.33.
- 4 Muestra la pantalla 4.34 con el código QR listo para compartir en cualquiera de sus aplicaciones de redes sociales.
- 5 Selecciona una opción en la pantalla 4.34.
- 6 Genera una imagen con el código QR, como se muestra en la figura 4.35, la cual se enviará a la aplicación seleccionada.

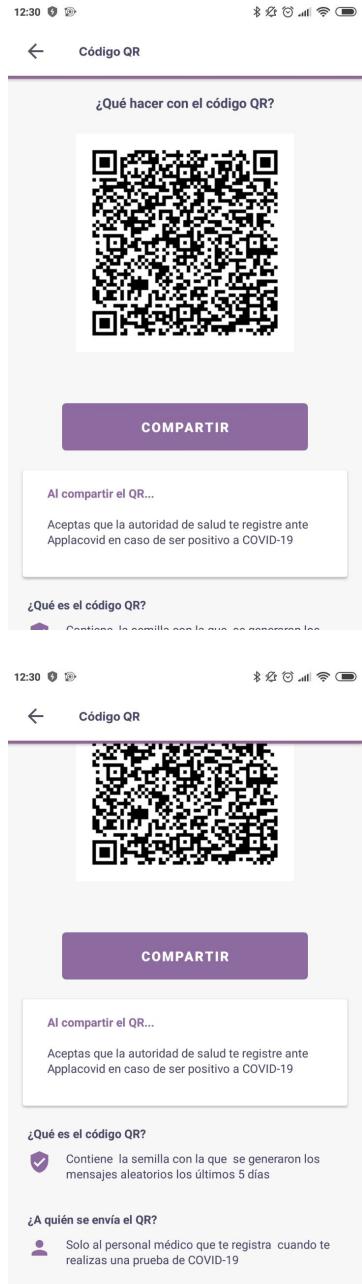


Figura 4.33: Código QR.



Figura 4.34: Compartir código QR.



Figura 4.35: Código QR en imagen.

7 ☰ Presiona el botón  en la esquina superior izquierda de la aplicación para regresar al menú principal [4.18](#).

- - - - *Fin del caso de uso.*

4.9. CU-08 Notificar posible infección

4.9.1. Objetivo

Cuando Applacovid detecta que el usuario ha estado en contacto cercano con alguna persona portadora con COVID-19, ejecuta este caso de uso para notificar al usuario de dicho evento.

Por medio de este caso de uso el usuario podrá revisar el tiempo que ha pasado desde el contacto con la (o las) personas contagiadas.

4.9.2. Atributos

Caso de Uso:	CU-08 Notificar posible infección
Entradas:	<ul style="list-style-type: none">Mientras la aplicación esté activa, puede recibir saludos de otros dispositivos vía Bluetooth.La aplicación se comunica con el <i>backend</i> cada dos horas, para recibir actualizaciones sobre contactos de riesgo, a través del endpoint <code>GET /v1/exposed/batchReleaseTime</code>.
Salidas:	<ul style="list-style-type: none">Mientras la aplicación esté activa, puede enviar saludos a otros dispositivos.En caso de detectar un contacto de riesgo Applacovid muestra un <i>push notification</i> al usuario.
Precondiciones:	<ul style="list-style-type: none">Para dispositivos Android tener encendido el Bluetooth y GPS en todo momento.Para dispositivos iOS tener encendido el Bluetooth en todo momento.
Postcondiciones:	<ul style="list-style-type: none">El consumo de energía aumentará ligeramente durante el uso de Applacovid.

4.9.3. Trayectorias del Caso de Uso

Trayectoria principal

- 1 Detecta un contacto de riesgo al revisar la lista de *Kd* obtenida del *backend*, a través del endpoint `GET /v1/exposed/batchReleaseTime`.
- 2 Envía una *push notification* como se muestra en la pantalla 4.23, para indicar al usuario que ha ocurrido un evento en Applacovid.
- 3 Ingresa a la aplicación presionando el ícono de Applacovid en la pantalla del dispositivo, como se muestra en la pantalla 4.36
- 4 Muestra la pantalla 4.37, con un mensaje sobre el posible contagio en el menú principal.
- 5 Selecciona la opción “Notificaciones” en el menú principal 4.37. [Trayectoria B]
- 6 Muestra la pantalla 4.38 para notificar al usuario que se ha detectado un contacto de riesgo con COVID-19.



Figura 4.36: Ingresar a Applacovid.



Figura 4.37: Menú principal con notificación de posible contagio.



Figura 4.38: Posible contagio.

- 7 ⚡ Presiona el botón “Continuar” para seguir las instrucciones de protección.
 - 8 🟡 Muestra la pantalla 4.39 con algunas instrucciones para protegerse. Puede visitar el sitio web de la aplicación para encontrar más medidas de seguridad.
 - 9 ⚡ Presiona el botón ⏪ para regresar al menú principal 4.37. [Trayectoria A]
 - 10 🟡 Muestra el menú principal 4.37.
- - - - *Fin del caso de uso.*

Trayectoria alternativa A:

Condición: Visitar el sitio web.

- A-1** ⚡ Presiona el botón “Visitar ahora” de la pantalla 4.39 para visitar la página web de Applacovid.
 - A-2** 🟡 Sale de la Applacovid y abre el navegador predeterminado del dispositivo para conectarse a la página web de Applacovid, mostrada en la pantalla 4.2.
- - - - *Fin de trayectoria.*

Trayectoria alternativa B:

Condición: Visitar el sitio web.

- B-1** ⚡ Presiona el enlace “Applacovid” de la pantalla 4.37 para visitar la página web de Applacovid.
 - B-2** 🟡 Sale de la Applacovid y abre el navegador predeterminado del dispositivo para conectarse a la página web de Applacovid, mostrada en la pantalla 4.2.
- - - - *Fin de trayectoria.*

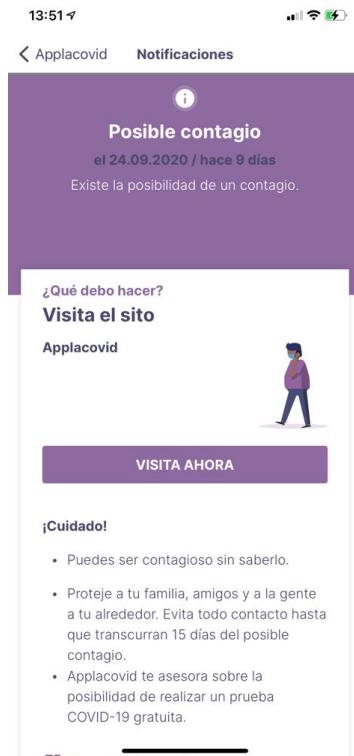


Figura 4.39: ¿Qué debo hacer?
52

4.10. Requerimientos no-funcionales

Los requerimientos no-funcionales describen las características cualitativas del sistema. A comparación de los requerimientos funcionales, éstos son mostrados de forma narrativa, ya que el diseño de interfaces, las restricciones del sistema, el flujo de negocio y la arquitectura del sistema proporcionan la implementación de los mismos.

Los requerimientos no-funcionales detectados para Applacovid son:

- Seguridad: Los protocolos de comunicación vía Bluetooth garantizan que los datos personales del usuario nunca son transmitidos a otros celulares o al servidor.
- Uso masivo: Las aplicaciones móviles y el *backend* implementan servicios REST para comunicarse y transmitir información. Los servicios web REST permiten expandir el número de servicios y la información que se transmite sin necesidad de un archivo de configuración.
- Escalabilidad: Nginx es un balanceador de carga que permite expandir el número de servicios atendiendo las peticiones del cliente.
- Confidencialidad: El canal de comunicación entre las aplicaciones móviles y el *backend* está cifrado sobre un canal HTTPS.

CAPÍTULO 5

Modelo lógico de información

Este capítulo describe los modelos entidad-relación diseñados e implementados para las aplicaciones móviles y el *backend*.

5.1. Modelo lógico del dispositivo

Del lado de las aplicaciones móviles se presentan las tablas mostradas en la Figura 5.1. La base de datos utilizada en los dispositivos móviles es **SQLite**. Los campos marcados con * son las llaves primarias de la entidad.

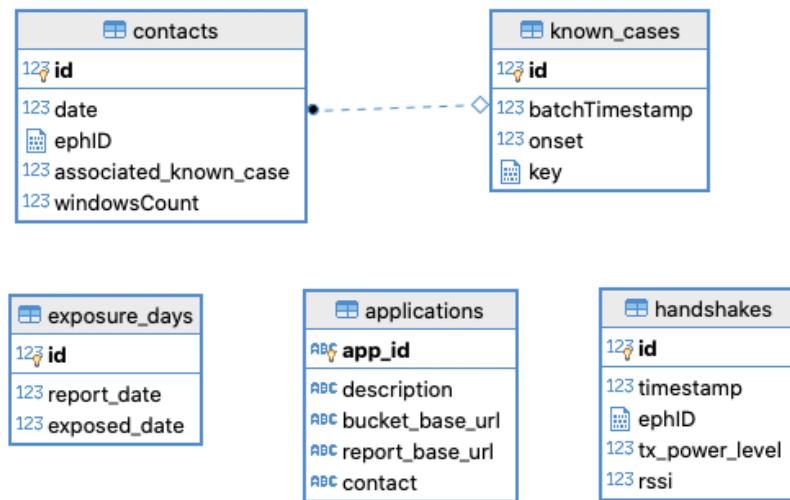


Figura 5.1: Base de datos del dispositivo.

contacts

Esta tabla almacena los contactos con contagio encontrados por el dispositivo al procesar la tabla `know_cases`.

Campo	Tipo de dato	Descripción	Null
<code>id *</code>	Integer	Es el identificador del registro.	No
<code>date</code>	Integer	Es la fecha asociada a la <i>Kd</i> .	No
<code>ephID</code>	BLOB	Es el identificador efímero en que se encontró la coincidencia de contagio.	No
<code>associated_know_case</code>	Integer	Es la llave foránea a la tabla <code>know_cases</code>	Sí
<code>windowsCount</code>	Integer	Es el número de ocurrencias encontradas con el mismo identificador efímero.	No

know_cases

Esta tabla almacena los casos conocidos con COVID-19, después de realizar la consulta al *backend*.

Campo	Tipo de dato	Descripción	Null
<code>id *</code>	Integer	Es el identificador del registro.	No
<code>batchTimestamp</code>	Integer	Es la fecha y hora en que se consultó al <i>backend</i> y en la cual venía el contacto de riesgo.	No
<code>onset</code>	Integer	Es la fecha asociada a la <i>Kd</i> .	No
<code>key</code>	BLOB	Es la <i>Kd</i> del contacto con contagio.	No

exposure_days

Esta tabla almacena la fecha en que el usuario se reportó como contagiado con COVID-19 y la fecha asociada a la *Kd*. Con esta información se puede determinar el número de días que el usuario lleva contagiado.

Campo	Tipo de dato	Descripción	Null
<code>id *</code>	Integer	Es el identificador del registro.	No
<code>report_date</code>	Integer	La fecha en que se reporta como contagiado.	Sí
<code>exposed_date</code>	Integer	La fecha asociada a la <i>Kd</i> .	Sí

applications

Esta tabla almacena información que utiliza la aplicación para comunicarse con el *backend*.

Campo	Tipo de dato	Descripción	Null
app_id *	Text	Es el identificador de la aplicación, comúnmente el UDID para dispositivos iOS y UUID para dispositivos Android.	No
description	Text	Es la descripción del registro.	Sí
bucket_base_url	Text	Es la URL a la que se enviará el reporte de contagio.	No
report_base_url	Text	Es la URL a consultar cada dos horas para actualizar la tabla <code>know_cases</code> .	No
contact	Text		Sí

handshakes

Esta tabla almacena todos los saludos recolectados por el dispositivo.

Campo	Tipo de dato	Descripción	Null
id *	Integer	Es el identificador del registro.	No
timestamp	Integer	Es la fecha en que se recibe el saludo.	No
ephID	BLOB	Es el identificador efímero recibido en el saludo.	No
tx_power_level	Real	Es la potencia de la señal Bluetooth.	Sí
rssi	Real	Es la intensidad de la señal .	Sí

5.2. Modelo lógico del servidor

Del lado del servidor existen las tablas mostradas en la Figura 5.2. La base de datos utilizada del lado del servidor es **PostgreSQL**. Los campos marcados con * son las llaves primarias de la entidad.

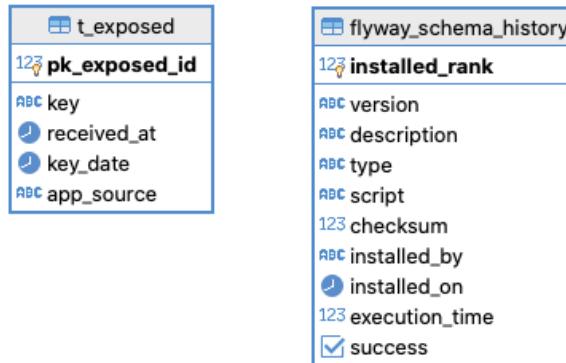


Figura 5.2: Base de datos del servidor.

t_exposed

t_exposed almacena todas las *Kd* reportadas por los dispositivos como contagiados con COVID-19.

Campo	Tipo de dato	Descripción	Null	Ejemplo
pk_exposed_id *	int	Es el identificador del registro.	No	35
key	text	Es la <i>Kd</i> del usuario reportado como contagiado con COVID-19.	No	T9ky/P2FTCQyQ ENVfwjkjfBoMR2A behix0Hm8NzjSRo =
received_at	timestamp	Es la fecha en la que se registró la información en la base de datos. Formato: YYYY-MM-DD HH:MM:SS	No	2020-11-14 16:14:41.048364-06
key_date	timestamp	Es la fecha de la <i>Kd</i> . Formato: YYYY-MM-DD HH:MM:SS	No	2020-11-06 18:00:00-06
app_source	varchar	Es el nombre de la aplicación que envía la información. En caso de las aplicaciones móviles es "Applacovid", cualquier otro caso es "AdminConsole".	No	Applacovid

flyway_schema_history

flyway_schema_history es una tabla que nos ayuda a llevar un control sobre quién y cuándo se realizó una modificación en las tablas. Esta tabla también nos ayudará a no sobre escribir los datos existentes en la base de datos, en caso de requerir un cambio en producción.

Campo	Tipo de dato	Descripción	Null	Ejemplo
installed_rank *	int	Es el identificador del registro.	No	1
version	varchar	Es la versión del script de la base de datos.	No	0.1
description	varchar	Una breve descripción del script de base de datos.	Sí	init the database
type	varchar	El tipo de lenguaje utilizado en el script.	No	SQL
script	varchar	Nombre del script.	No	V0_1__init.sql
checksum	int	El <i>checksum</i> del archivo, para verificar que se trata de una versión diferente del script.	Sí	V0_1__init.sql
installed_by	varchar	Indica el nombre del usuario que ejecuto el script.	No	postgres
installed_on	timestamp	Indica la fecha en la que ejecutó el script de base de datos.	No	2020-10-24 13:13:58.9588
execution_time	int	Indica el tiempo que tomó la ejecución del script de base de datos en milisegundos.	No	82
success	bool	Indica si la ejecución de script de base de datos fue exitosa o no.	No	t (true)

CAPÍTULO 6

Especificación de la plataforma

En este capítulo se detalla la arquitectura del subsistema “Aplicaciones móviles”. Se describen las tecnologías y patrones de diseño utilizados para modelar, implementar y administrar la arquitectura de Applacovid.

6.1. Arquitectura del sistema

En la Figura 6.1 se muestra la arquitectura **cliente-servidor** utilizada en el subsistema “Aplicaciones móviles”.

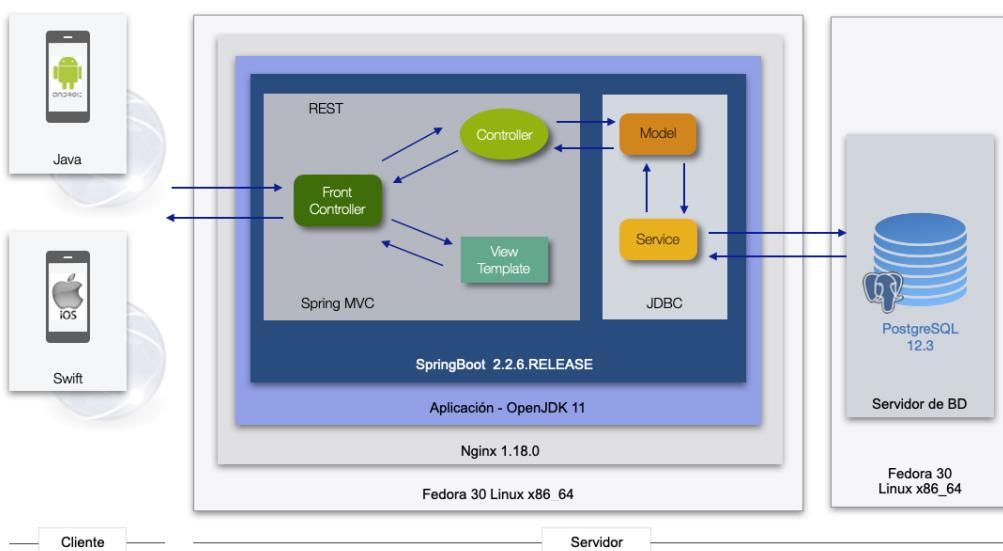


Figura 6.1: Arquitectura del subsistema.

Esta arquitectura es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios (servidores), y los demandantes (clientes). Básicamente, un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Este tipo de arquitectura es ampliamente utilizada en aplicaciones basadas en web, ya que el cliente y el servidor pueden actuar como una sola entidad y/o actuar como entidades separadas, realizando actividades o tareas independientes. En las siguientes secciones se describen las tecnologías utilizadas en esta arquitectura.

6.1.1. Cliente

Compuesto por las aplicaciones móviles iOS y Android, cada una desarrollada en su lenguaje de programación.

- **Swift:** Es un lenguaje de programación intuitivo y poderoso para dispositivos Apple, tales como: macOS, iOS, watchOS, tvOS, etc. El código Swift es seguro por diseño y el código que genera se ejecuta velozmente. Swift utiliza un patrón de diseño **Modelo-Vista-Controlador**, como se muestra en la Figura 6.2, lo que permite organizar y administrar correctamente el código. La parte más importante en este modelo es el *UIViewController*, el cual es un objeto que representa cada una de las pantallas de la aplicación. Éste se encarga de la gestión de las vistas; de la entrada del usuario con “taps” o gestos sobre la interfaz de usuario; y de la comunicación con el modelo y la actualización de las vistas.

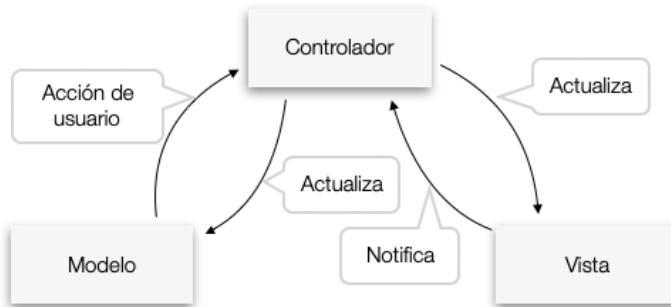


Figura 6.2: Modelo-Vista-Controlador de Swift.

Aplicación es soportado en versiones iOS 13 o superior.

- **Java:** Del lado de los dispositivos Android se encuentra el entorno de ejecución basado en Java. Una de las principales desventajas de desarrollo en estos dispositivos es su “Fragmentación”, es decir la gran variedad de dispositivos Android que existen en el mercado con múltiples fabricantes, diferentes versiones de la plataforma y software específico, esto hace que frecuentemente la aplicación móvil no funcione bien en todos los dispositivos. Los componentes principales de una aplicación Android son:

- Clases: Contiene las Actividades, Servicios y Receptores de notificaciones.
- Recursos: Es el contenido estático de la aplicación, es decir, no define funcionalidad alguna de la aplicación.

- Manifiesto: Es la identificación de la aplicación, en este archivo se encuentra el nombre completo de la aplicación, los componentes que utiliza, los permisos que necesita para trabajar y la versión mínima del sistema operativo Android para funcionar.

Applacovid es soportado en versiones Android 6 *Marshmallow* o superior

6.1.2. Servidor

El servidor de Applacovid está compuesto por dos módulos principales. El servidor de aplicaciones y el servidor de base de datos.

- **Spring MVC**¹: El *framework* de *Spring MVC* proporciona componentes que pueden usarse para desarrollar aplicaciones web flexibles y poco acopladas. El patrón de diseño MVC separa los diferentes aspectos de la aplicación: lógica de datos, lógica de negocio y lógica de la interfaz de usuario, al tiempo que proporciona un bajo acoplamiento entre estos elementos. En la Figura 6.1 se muestra este patrón de diseño en alto nivel, sus principales componentes son:

- El **modelo** encapsula los datos de la aplicación.
- La **vista** es responsable de interpretar los datos del modelo y en general, de generar el HTML que se enviará al navegador/aplicación del cliente, quien será el encargado de interpretar la respuesta.
- El **controlador** es responsable de procesar las peticiones del cliente; construir un modelo apropiado; y pasar el modelo a la vista para que ésta lo interprete. Además, es el encargado de atender las peticiones HTTP realizadas por el cliente, mismas que deben realizarse en convención REST.

- **JDBC**²: Es una Interfaz de Programación de Aplicaciones (API por sus siglas en inglés) para Java, la cual define los métodos para acceder a la base de datos por parte del cliente. Provee métodos para consultar, actualizar, insertar y borrar los datos en la base de datos. La principal característica de la API es que está orienta a base de datos relacionales. En la Figura 6.1 se observan dos componentes principales en JDBC:

- El **modelo** encapsula los datos de la aplicación, ya sea los que se envían en la petición o los recuperados de la base de datos.
- El **servicio** contiene la implementación personalizada de los métodos para acceder a la base de datos, éste es el encargado de comunicarse con la base de datos para consultar, almacenar, actualizar o borrar los datos enviados en el modelo.

- **Spring Boot**³: Es un *framework* para construir rápidamente aplicaciones *stand-alone* basadas en JAVA, fáciles de ejecutar. Spring Boot se encarga de facilitar todas las bibliotecas de funciones de terceros para que el desarrollador se enfoque en implementar la lógica del negocio, sin preocuparse por la infraestructura de la aplicación. Casi todas las aplicaciones desarrolladas con Spring Boot requieren una configuración mínima inicial. Una de las principales ventajas de las aplicaciones desarrolladas usando Spring Boot, es que no requiere desplegar un archivo WAR en el servidor de aplicaciones, ya que provee diversos servidores embebidos, lo que hace sencillo el desarrollo y el despliegue de la aplicación en producción. Applacovid utiliza la versión de Spring Boot 2.2.6.RELEASE.

¹<https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>

²<https://www.javatpoint.com/spring-JdbcTemplate-tutorial>

³<https://spring.io/projects/spring-boot>

- **OpenJDK 11**⁴: El servidor tiene instalada la versión 11 del OpenJDK. Ésta es una versión es libre y de código abierto de la plataforma Java Standard Edition (Java SE). La implementación esta licenciada bajo la Licencia Pública General GNU (GNU GPL).
- **Nginx**⁵: Es un servidor web que puede ser utilizado también como un proxy inverso, balanceador de carga, un proxy para protocolos de correo electrónico (IMAP/POP3) y como cache de HTTP. Nginx utiliza poca memoria y ofrece alta concurrencia. En lugar de crear nuevos procesos por cada petición web, Nginx usa un enfoque asíncrono basado en eventos, en el cual las peticiones son manejadas en un hilo por separado, lo que permite que se ejecuten concurrentemente sin bloquear otros trabajos. Nginx es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada. Applacovid utiliza Nginx como servidor de aplicaciones, contenedor de aplicaciones Java y como balanceador de carga para los subsistemas que conforman Applacovid. La versión que se utiliza es la 1.18.0.
- **PostgreSQL**⁶: Es un sistema manejador de base de datos (RDBMS por sus siglas en inglés) extensible y de código abierto. Algunas de sus principales características son:
 - Alta concurrencia: Mediante un sistema denominado MVCC (Acceso Concurrente Multiversión por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
 - Amplia variedad de tipos nativos: Se pueden crear tipos de datos personalizados, los cuales pueden ser completamente indexables gracias a la infraestructura GiST de PostgreSQL.

La versión de PostgreSQL utilizada para Applacovid es la 12.3.

- **Linux**: Todos los componentes descritos anteriormente se ejecutan en una máquina Linux x86_64 con la distribución Fedora 30.

6.2. Arquitectura de los servicios web

Applacovid implementa servicios REST para establecer la comunicación entre las aplicaciones móviles iOSy Android con el *backend*, como se muestra en la Figura 6.1.

REST significa *REpresentational State Transfer* y son básicamente servicios web basados en la arquitectura REST. El objetivo principal de los servicios REST es reutilizar los métodos que ya presenta el protocolo HTTP, siendo los más comunes:

- **GET**: Proporciona acceso de sólo lectura a un recurso en el servidor.
- **POST**: Es usado para crear un recurso nuevo en el servidor.
- **DELETE**: Usado para eliminar un recurso.
- **UPDATE**: Usado para actualizar un recurso ya existente o crear un recurso nuevo.

Los servicios REST giran entorno a un recurso. Un recurso puede ser cualquier cosa y éste puede ser accedido por la interfaz común usando los métodos estándares de HTTP. Los recursos pueden tener diversas representaciones como XML, HTML y JSON, siendo JSON el más popular, y pueden ser accedidos mediante el *Uniform Resource Identifier* (URI), por ejemplo:

⁴<https://openjdk.java.net/projects/jdk/11/>

⁵<https://www.nginx.com/welcome-to-nginx/>

⁶<https://www.postgresql.org/>

<https://pikal.cs.cinvestav.mx/v1/exposed>

<https://pikal.cs.cinvestav.mx/admin/console/newRecord?key=123>

En el primer ejemplo la ruta de la URI es `v1/exposed`, mientras que en el segundo ejemplo la ruta es `admin/console/newRecord`, además de presentar una consulta adicional con `newRecord?key=123`. Ambos estilos de URIs son considerados aceptables en un formato JSON.

Los servicios web REST son ligeros, altamente escalables, fáciles de mantener y son comúnmente utilizados al momento de crear APIs para aplicaciones basadas en web. Es importante resaltar que REST es un enfoque arquitectónico, no un protocolo.

En la Figura 6.3 se muestra cómo se realiza la comunicación entre el *Service Provider* (servidor) y el *Service Consumer* (cliente, y en nuestro caso las aplicaciones móviles de Applacovid).

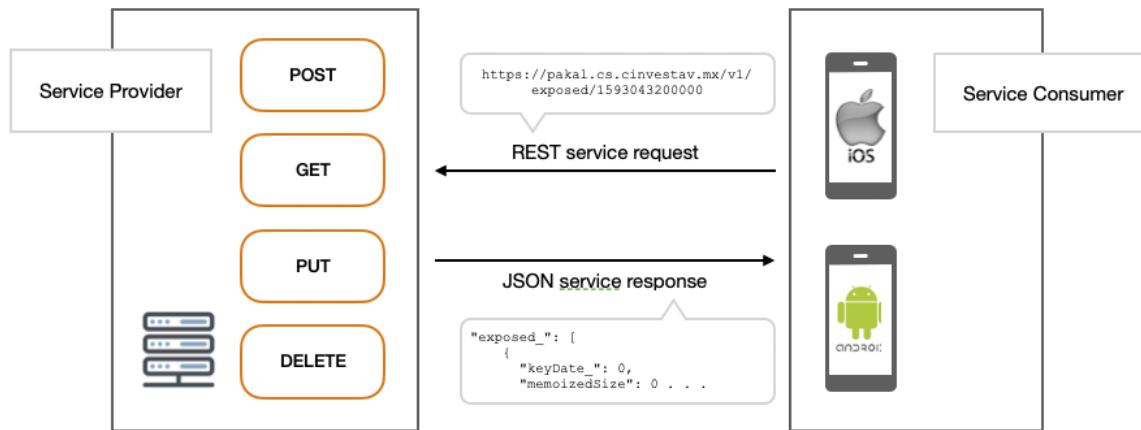


Figura 6.3: Arquitectura de los servicios REST.

Cuando la aplicación móvil solicita un recurso al servidor, ésta envía la URI del recurso en el método HTTP. El *Service Provider* busca y/o procesa el recurso y se lo envía de regreso a la aplicación móvil. Una vez obtenido el recurso, la aplicación móvil accede y modifica el recurso.

Los servicios REST no definen el formato del mensaje a intercambiar, esto lo define los requerimientos de la aplicación.

6.3. Seguridad

Los servicios web REST trabajan con las rutas de las URLs del protocolo HTTP, por lo tanto, es importante resguardar los servicios REST de la misma manera como se hace en un sitio web normal. Los siguientes puntos son los mecanismos de seguridad implementados en la comunicación entre cliente-servidor de Applacovid, y recomendados en la literatura como mejores prácticas en el diseño de servicios web REST:

- Validación: Validar todos los datos de entrada que llegan al servidor. Esto puede prevenir cualquier ataque de inyección SQL o NoSQL.

- Sesión basada en autenticación: Autenticar cualquier petición realizada a los métodos de los servicios web.
- Evitar mostrar datos sensibles en la URL de la petición. En lugar de usar parámetros en la URL, utilizar el método POST para enviar datos sensibles al servidor.
- Restringir la ejecución de métodos: Restringir la eliminación de información en los métodos GET, POST y DELETE. GET no debería borrar información bajo cualquier circunstancia.
- Validar datos de entrada: Revisar que todos los datos de entrada estén bien formados antes de ejecutar cualquier transacción.
- Enviar mensajes de error genéricos: Los servicios web REST deberían usar los códigos y mensajes de error establecidos en el protocolo HTTP, como: 403 - Acceso prohibido, etc.

Además de estas recomendaciones Applacovid garantiza la confidencialidad del canal de comunicación entre el cliente y servidor, implementando un protocolo SSL sobre HTTP, es decir HTTPS. Todas las peticiones que los clientes realicen al servidor deberán viajar por el protocolo HTTPS.

<https://pikal.cs.cinvestav.mx/>

6.4. Entorno de desarrollo

En esta sección se describe el entorno de desarrollo del *backend*.

- **IntelliJ IDEA, Community Edition**⁷: Es un ambiente de desarrollo integrado (IDE por sus siglas en inglés), escrito en Java y para desarrollos de software de computadora. Proporciona diversos servicios y herramientas a los programadores para el desarrollo de software. Los IDEs al menos incluyen un editor de código fuente, herramientas para automatizar la compilación del código y herramientas de *debug*.
- **Java SE 11**⁸: Es una colección de APIs del lenguaje de programación Java propiedad de Oracle.
- **Maven**⁹: Es una herramienta para automatizar la compilación del código fuente, principalmente de proyectos Java. Además, es un gestor de dependencias (bibliotecas de funciones). Mediante un archivo XML se describe la forma en que el código fuente se debe compilar, enlazar con otros componentes y sus dependencias con directorios o recursos externos.
- **GIT**¹⁰: Es un software de control de versiones. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.
- **Bash script**¹¹: Es un procesador de comandos. Applacovid lo utiliza principalmente para automatizar tareas, así como desplegar componentes en el servidor (*backend*).

⁷<https://www.jetbrains.com/idea/>

⁸<https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

⁹<https://maven.apache.org/>

¹⁰<http://rogerdudler.github.io/git-guide/>

¹¹<https://www.gnu.org/software/bash/>

- **Docker**¹²: Es un conjunto de servicios vistos como plataforma (PaaS, por sus siglas en inglés) que utilizan la virtualización a nivel de sistema operativo para proporcionar software en paquetes, llamados contenedores. Applacovid sólo utiliza esta herramienta en el ambiente de desarrollo para virtualizar PostgreSQL.
- **PostgreSQL container**: Es un contenedor Docker, el cual no necesita instalar el manejador de base de datos PostgreSQL nativamente.
- **Spring Boot**¹³: Ayuda a construir rápidamente aplicaciones *stand-alone* basadas en JAVA, las cuales son fáciles de ejecutar, incluso en un contenedor de Docker.

Para mayor información de las tecnologías y/o *frameworks* utilizados, favor de referirse a las páginas web oficiales de cada tecnología.

6.4.1. Bibliotecas de funciones de terceros

Algunas bibliotecas de funciones de terceros y código fuente utilizado en la construcción de Applacovid son los siguientes. Es importante señalar que estos proyectos son de libres y de código abierto.

- **BouncyCastle**: Es una biblioteca de funciones especializada en algoritmos criptográficos. La especificación de esta biblioteca de funciones puede encontrarse en el siguiente enlace:

<https://www.bouncycastle.org/>

- **DP3T**: Rastreo de proximidad descentralizado preservando la privacidad (DP3T por sus siglas en inglés *Decentralised Privacy-Preserving Proximity Tracing*). Es un protocolo abierto para rastrear la proximidad de COVID-19 utilizando la funcionalidad de Bluetooth *Low Energy*. El protocolo asegura que los datos personales y su procesamiento se realiza enteramente en el dispositivo del usuario. El proyecto está licenciado bajo los términos de la licencia MPL 2¹⁴. La especificación del protocolo puede encontrarse en el siguiente enlace:

<https://github.com/DP-3T/dp3t-sdk-backend>

6.4.2. Código fuente del proyecto

El código fuente del proyecto se encuentra disponible en el siguiente repositorio GIT:

<https://github.com/PanchoRH/Applacovid/tree/main/Backend/dp3t-sdk-backend>

La estructura del código del proyecto se muestra en la Figura 6.4. En caso de requerir acceso a la plataforma GitHub, para la revisión del código, se debe solicitar acceso y autorización a los responsables del proyecto.

¹²<https://www.docker.com/>

¹³<https://spring.io/projects/spring-boot>

¹⁴https://en.wikipedia.org/wiki/Mozilla_Public_License

main		
	Applacovid / Backend / dp3t-sdk-backend /	
Angelina Reyes	Se agrega el backend y la consola de administración	7f4c014 1 minute ago
..		History
EC_key	Se agrega el backend y la consola de administración	1 minute ago
documentation	Se agrega el backend y la consola de administración	1 minute ago
dpppt-backend-sdk	Se agrega el backend y la consola de administración	1 minute ago
postgresql	Se agrega el backend y la consola de administración	1 minute ago
server-configuration	Se agrega el backend y la consola de administración	1 minute ago
ws-sdk	Se agrega el backend y la consola de administración	1 minute ago
CONTRIBUTING.md	Se agrega el backend y la consola de administración	1 minute ago
GenerateKeyPair.java	Se agrega el backend y la consola de administración	1 minute ago
GenerateKeyPairEC.java	Se agrega el backend y la consola de administración	1 minute ago
LICENSE	Se agrega el backend y la consola de administración	1 minute ago
Makefile	Se agrega el backend y la consola de administración	1 minute ago
README.md	Se agrega el backend y la consola de administración	1 minute ago
configure.sh	Se agrega el backend y la consola de administración	1 minute ago
discovery.json	Se agrega el backend y la consola de administración	1 minute ago

Figura 6.4: Estructura del código del proyecto.

CAPÍTULO 7

Servicios Web

En este capítulo se presentan los diferentes servicios web REST desarrollados e implementados durante la ejecución de los casos de uso de Applacovid. En la Sección [Modelos](#) se proporcionan ejemplos a los campos de las peticiones, para facilitar la lectura de los mismos. Para cada uno de los servicios web, se listan todas las posibles respuestas, sus códigos de estatus y la razón por la que podrían ocurrir. Los campos marcados con * son obligatorios.

La implementación de los servicios se encuentran en la siguiente página web:

<https://pikal.cs.cinvestav.mx/>

7.1. GET /v1/

Este servicio es un saludo. En caso de que el servicio esté activo regresa un “hello” de respuesta.

Parámetros

No presenta parámetros

Cuerpo de la respuesta

CURL

```
1 curl -X GET "http://pikal.cs.cinvestav.mx/v1/" -H "accept: application/json"
```

Request URL

```
1 http://pikal.cs.cinvestav.mx/v1/
```

Respuestas

Código	Descripción
200	Servidor activo y funcionando correctamente.

Endpoint

<https://pikal.cs.cinvestav.mx/v1/>

7.2. POST /v1/exposed/

Este servicio registra en la base de datos un contacto reportado con COVID-19. El cliente que realiza la petición debe enviar la *Kd* en la petición.

Parámetros

Nombre	Descripción
User Agent *	App Identifier (PackageName/BundleIdentifier) + App-Version + OS (Android/iOS) + OS-Version. Ejemplo: ch.ubique.android.starsdk;1.0;iOS;13.3

Cuerpo de la petición*

El objeto `ExposeeRequest` contiene la *Kd*, la fecha de inicio de la *Kd* y el código COVID-19 proporcionado por la autoridad de salud para verificar el resultado de la prueba. Es obligatorio enviar el objeto en el cuerpo de la petición.

```

1 {
2   "fake": 0,
3   "key": "string",
4   "keyDate": 0,
5   "authData": {
6     "value": "string"
7   }
8 }
```

Respuestas

Código	Descripción
200	La <i>Kd</i> ha sido almacenada correctamente en la base de datos.
400	Codificación base 64 inválida en la petición.
403	Falló la autenticación.

Endpoint

<https://pikal.cs.cinvestav.mx/v1/exposed/>

7.3. GET /v1/exposed/{batchReleaseTime}

Este servicio regresa una lista de Kd registradas como contagiadas dentro de un periodo de tiempo específico. Este periodo de tiempo es de dos horas, es decir, el cliente realiza una petición cada dos horas al *backend* enviando el último periodo en que realizó la consulta. De esta manera se asegura que sólo se envían las Kd que no ha revisado.

Parámetros

Nombre	Descripción
batchReleaseTime * integer (\$long) (path)	Es el <i>batch release date</i> de las Kd contagiadas, expresado en milisegundos desde Unix Epoch (1970-01-01). Debe ser un múltiplo de $2 * 60 * 60 * 1000$. Ejemplo: 1593043200000

Cuerpo de la respuesta

```
1 {  
2     "bitField0_": 0,  
3     "batchReleaseTime_": 0,  
4     "exposed_": [  
5         {  
6             "keyDate_": 0,  
7             "memoizedSize": 0,  
8             "memoizedHashCode": 0  
9         }  
10    ],  
11    "memoizedSize": 0,  
12    "memoizedHashCode": 0  
13 }
```

Respuestas

Código	Descripción
200	Regresa el objeto ExposedOverview , en formato protobuf ¹ , con todas las Kd contagiadas que fueron publicadas dentro del batchReleaseTime. Ver la Sección Cuerpo de la respuesta para un mejor ejemplo.
400	No pudo encontrar el batchReleaseTime.

¹protobuf es un *media type* del protocolo HTML: application/x-protobuf.

Endpoint

`https://pakal.cs.cinvestav.mx/v1/exposed/{batchReleaseTime}`

7.4. Modelos

Todos los modelos utilizados por los *endpoints* se describen a continuación. Para cada campo se proporciona un ejemplo que da un mejor panorama de lo que el *backend* espera recibir en una petición. Los campos marcados con * son obligatorios.

7.4.1. ExposeeRequest

Campo	Tipo de dato	Descripción	Ejemplo
<code>key *</code>	string	Es la <i>Kd</i> usada para generar los <i>efID</i> , codificada en base64.	QUJDREVGR0hJSktMTU5PUFFSU1RVVIdYWVpBQkNERUY=
<code>keyDate *</code>	string	La fecha de inicio de la <i>Kd</i> . Formato: yyyy-MM-dd	2020-10-31
<code>authData *</code>	ExposeeAuthData	Datos para autenticar el código COVID-19 proporcionado por una autoridad de salud.	123 852 469 894

7.4.2. ExposeeAuthData

Campo	Tipo de dato	Descripción	Ejemplo
<code>value</code>	string	Código COVID-19 proporcionado por una autoridad de salud (en base64)	123 852 469 894

7.4.3. ExposedOverview

Campo	Tipo de dato	Descripción	Ejemplo
<code>exposed</code>	Exposee[]	Una lista de todas las <i>Kd</i> confirmadas con COVID-19.	

7.4.4. Exposee

Campo	Tipo de dato	Descripción	Ejemplo
key *	string	La <i>Kd</i> de un caso confirmado con COVID-19, como string en base64. La <i>Kd</i> consiste de 32 bytes.	QUJDREVGR0hJSktMTU5PUFFSU1RVVIdYWVpbBQkNERUY=
onset *	string	La fecha de inicio de la <i>Kd</i> . Formato: yyyy-MM-dd.	2020-04-06