

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DESARROLLO DE UN SISTEMA DE ANUNCIOS
DIGITALIZADOS A PARTIR DE UNA BALIZA BLE, EL
PROTOCOLO EDDYSTONE Y UN APLICATIVO
MULTIPLAFORMA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

EFRÉN EDUARDO SANDOVAL AVILEZ

efren.sandoval@epn.edu.ec

DIRECTOR: HERNÁN DAVID ORDÓÑEZ CALERO

hernan.ordonez@epn.edu.ec

CODIRECTORA: DIANA YACCHIREMA VARGAS

diana.yacchirema@epn.edu.ec

Quito, julio 2022

DECLARACIÓN

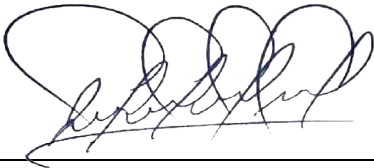
Yo, EFRÉN EDUARDO SANDOVAL AVILEZ, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

EFRÉN EDUARDO SANDOVAL AVILEZ

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Efrén Eduardo Sandoval Avilez, bajo nuestra supervisión.



HERNÁN DAVID ORDÓÑEZ CALERO
DIRECTOR

DIANA YACCHIREMA VARGAS
CODIRECTORA

AGRADECIMIENTO

Mi más profundo y sentido agradecimiento es para mi querida esposa María Magdalena, cuyo presencia y apoyo tanto físico como moral me ha permitido concluir este trabajo y estar próximo a graduarme y titularme.

A Dalila Mera y María Ofelia Páez, mis compañeras pero sobre todo amigas en el Banco Central del Ecuador sin cuyo empuje pero sobre todo paciencia no lo hubiese logrado.

A mi alma mater la Escuela Politécnica Nacional y a todos y cada uno de sus profesores y profesoras cuyo legado lo llevo impreso en el alma.

Finalmente quiero agradecer a mi director MSc. Hernán Ordoñez por su tiempo, dedicación, consejos pero sobre todo por su paciencia.

DEDICATORIA

Este trabajo lo dedico a mis amadas hijas Julieta y Renata quienes fueron en todo momento mi fuente de motivación para continuar adelante especialmente en los momentos más difíciles en los que estuve por botar la toalla.

TABLA DE CONTENIDO

1 INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DEL PROBLEMA	1
1.2 OBJETIVO GENERAL	1
1.3 OBJETIVOS ESPECÍFICOS	1
1.4 ALCANCE	2
1.5 MARCO TEÓRICO	2
2 METODOLOGÍA	6
2.1 ESPECIFICACIÓN DE REQUERIMIENTOS	6
2.2 DEFINICIÓN DE ROLES	7
2.3 HISTORIAS DE USUARIO	8
2.4 ESTIMACIÓN DEL PRODUCT BACKLOG	11
2.5 ARQUITECTURA DEL SISTEMA	12
2.6 IMPLEMENTACIÓN	13
2.6.1 SPRINT 0	13
2.6.1.1 SPRINT PLANNING	15
2.6.1.2 IMPLEMENTACIÓN	15
2.6.1.3 SPRINT REVIEW	20
2.6.2 SPRINT 1	21
2.6.2.1 SPRINT PLANNING	21
2.6.2.2 IMPLEMENTACIÓN	21
2.6.2.3 SPRINT REVIEW	24
2.6.3 SPRINT 2	24
2.6.3.1 SPRINT PLANNING	24
2.6.3.2 IMPLEMENTACIÓN	25
2.6.3.3 SPRINT REVIEW	28

2.6.4 SPRINT 3	29
2.6.4.1 SPRINT PLANNING	29
2.6.4.2 IMPLEMENTACIÓN	29
2.6.4.3 SPRINT REVIEW	33
2.6.5 SPRINT 4	33
2.6.5.1 SPRINT PLANNING	34
2.6.5.2 IMPLEMENTACIÓN	34
2.6.5.3 SPRINT REVIEW	38
2.6.6 SPRINT 5	38
2.6.6.1 SPRINT PLANNING	39
2.6.6.2 IMPLEMENTACIÓN	40
2.6.6.3 SPRINT REVIEW	43
2.6.7 SPRINT 6	44
2.6.7.1 SPRINT PLANNING	44
2.6.7.2 IMPLEMENTACIÓN	44
2.6.7.3 SPRINT REVIEW	47
3 RESULTADOS Y DISCUSIÓN	48
3.1 PRUEBAS DE FUNCIONALIDAD	48
3.1.1 CASO DE PRUEBA PARA COMERCIOS	48
3.1.2 CASO DE PRUEBA PARA CLIENTES	51
3.2 PRUEBAS DE USABILIDAD	53
3.2.1 EVALUACIÓN DEL APLICATIVO EL PREGONERO COMERCIOS	54
3.2.2 EVALUACIÓN DEL APLICATIVO EL PREGONERO CLIENTES	55
3.3 DISCUSIÓN	57
4 CONCLUSIONES Y RECOMENDACIONES	58
4.1 CONCLUSIONES	58
4.2 RECOMENDACIONES	58
REFERENCIAS BIBLIOGRÁFICAS	60

LISTA DE FIGURAS

Figura 1 – Arquitectura del Sistema de Anuncios Digitales	13
Figura 2 – Interfaz de Usuario principal del IDE VSC	16
Figura 3 – Muestra del sitio web de Flutter	16
Figura 4 – Ruta del sistema de archivos donde Flutter está alojado	17
Figura 5 – Variable de entorno PATH apuntando a Flutter	17
Figura 6 – Resultado de la ejecución del comando <i>flutter –version</i>	17
Figura 7 – Complemento Flutter instalado	17
Figura 8 – Complemento Dart instalado	18
Figura 9 – SDK Android	18
Figura 10 – Variable de entorno PATH apuntando al SDK Android	18
Figura 11 – Pantalla de bienvenida de Raspberry Pi OS	18
Figura 12 – Estructura del directorio del proyecto <i>el_pregonero_comercios</i>	19
Figura 13 – Estructura del directorio del proyecto <i>el_pregonero_clientes</i>	20
Figura 14 – Actualizando los paquetes del sistema operativo Raspberry Pi	21
Figura 15 – Instalación de dependencias	22
Figura 16 – Descarga del paquete BlueZ	22
Figura 17 – Instalación del paquete BlueZ	22
Figura 18 – Resultado de la ejecución del comando <i>hciconfig</i>	22
Figura 19 – Contenido del archivo <i>beacon.sh</i>	23
Figura 20 – Resultado de la ejecución del script <i>beacon.sh</i>	23
Figura 21 – Comando adherido al archivo <i>.bashrc</i>	23
Figura 22 – Después del reinicio la baliza sigue difundiendo el mensaje	24
Figura 23 – Vista del archivo <i>pubspec.yaml</i>	25
Figura 24 – Pantalla <i>Anuncio</i> vacía	26
Figura 25 – <i>Image picker</i> con ícono de una cámara como placeholder	26
Figura 26 – <i>Image picker</i> con una imagen capturada con la cámara	27

Figura 27 – <i>Image picker</i> con una imagen recuperada desde la galería	27
Figura 28 – <i>Image picker</i> junto con widget de inserción de texto	28
Figura 29 – Probando el ingreso de texto	28
Figura 30 – Campo de texto <i>Precio de venta al público</i>	30
Figura 31 – Campo de texto <i>Porcentaje de Descuento</i>	30
Figura 32 – Teclado numérico al ingresar al campo pvp o al campo %d	31
Figura 33 – Widget <i>Anuncio</i>	31
Figura 34 – Pantalla <i>Lista de Anuncios</i>	32
Figura 35 – Cambios realizados a un <i>Anuncio</i>	32
Figura 36 – Eliminar un <i>Anuncio</i> de la pantalla <i>Anuncios</i>	33
Figura 37 – Pantalla <i>Promociones</i>	35
Figura 38 – Notificación de la aplicación	35
Figura 39 – Reacción al toque sobre la notificación	36
Figura 40 – Pantalla <i>Promoción</i> y el flujo asociado	36
Figura 41 – Widget de marcado	37
Figura 42 – Pantalla <i>Favoritos</i>	37
Figura 43 – Eliminar un <i>Anuncio</i> de la pantalla <i>Favoritos</i>	38
Figura 44 – Pantalla <i>Registro de Usuario Nevo</i>	40
Figura 45 – Solicitud de registro	40
Figura 46 – Registro de nuevo usuario exitoso	41
Figura 47 – Pantalla <i>Inicio de Sesión</i>	41
Figura 48 – Solicitud de inicio de sesión	42
Figura 49 – Navegación hacia la pantalla <i>Registro de Usuario Nuevo</i>	42
Figura 50 – Navegación hacia la pantalla <i>Anuncios</i>	43
Figura 51 – Pantalla <i>Perfil y Configuraciones</i> vacía	45
Figura 52 – Pantalla <i>Perfil y Configuraciones</i> con sus elementos	45
Figura 53 – Cierre de sesión	46
Figura 54 – Widget <i>Perfil y Configuraciones</i>	46

Figura 55 – Pantalla principal con el widget *Perfil y Configuraciones* 47

LISTA DE TABLAS

Tabla 1 – Roles SCRUM y sus responsables	7
Tabla 2 – Historias de usuario y sus criterios de aceptación	8
Tabla 3 – Priorización y estimación de la duración de la implementación de cada historia de usuario	11
Tabla 4 – Distribución de las historias de usuario en cada sprint	12
Tabla 5 – Herramientas del entorno de desarrollo	13
Tabla 6 – Tareas correspondientes al Sprint 0	15
Tabla 7 – Cumplimiento de los criterios de aceptación – Sprint 0	20
Tabla 8 – Tareas correspondientes al Sprint 1	21
Tabla 9 – Cumplimiento de los criterios de aceptación – Sprint 1	24
Tabla 10 – Tareas correspondientes al Sprint 2	24
Tabla 11 – Cumplimiento de los criterios de aceptación – Sprint 2	28
Tabla 12 – Tareas correspondientes al Sprint 3	29
Tabla 13 – Cumplimiento de los criterios de aceptación – Sprint 3	33
Tabla 14 – Tareas correspondientes al Sprint 4	34
Tabla 15 – Cumplimiento de los criterios de aceptación – Sprint 4	38
Tabla 16 – Tareas correspondientes al Sprint 5	39
Tabla 17 – Cumplimiento de los criterios de aceptación – Sprint 5	43
Tabla 18 – Tareas correspondientes al Sprint 6	44
Tabla 19 – Cumplimiento de los criterios de aceptación – Sprint 6	47
Tabla 20 – Ficha caso de prueba para comercios	48
Tabla 21 – Ficha resultados esperados	49
Tabla 22 – Resultados obtenidos después de ejecutado el caso de prueba	50
Tabla 23 – Ficha caso de prueba para clientes	51
Tabla 24 – Ficha resultados esperados	52
Tabla 25 – Resultados obtenidos después de ejecutado el caso de prueba	52
Tabla 26 – La 10 preguntas cuestionario SUS	54

Tabla 27 – Puntajes obtenidos al aplicar el cuestionario SUS	54
Tabla 28 – Puntajes normalizados y resultado total	55
Tabla 29 – Puntajes obtenidos al aplicar el cuestionario SUS	56
Tabla 30 – Puntajes normalizados y resultado total	56

RESUMEN

La industria del papel al deforestar extensas áreas de bosques para obtener la materia prima (pulpa) con la que se confecciona el papel, es una de las industrias que mayormente aportan al cambio climático. La falta de árboles provoca que el dióxido de carbono que se encuentra en la atmosfera no se reabsorbido como parte del proceso de fotosíntesis. La producción de papel también es una de las mayores consumidoras de energía la cual proviene de fuentes no renovables (Environmental Paper Network, 2018).

El presente trabajo de titulación tiene por objeto presentar una alternativa digital a la publicidad impresa. Esta alternativa consiste en la implementación de un sistema de marketing de proximidad (Jeon, She, Soonsawad, & Ng, 2018), una idea que en otros países se la viene trabajando. El sistema consiste de dos aplicaciones multiplataforma (con enfoque móvil) construidas con Flutter, una baliza BLE (implementada sobre Raspberry Pi) y Firebase como backend del sistema. La alternativa propuesta espera convertirse en una solución al problema que plantea la deforestación y alto consumo energético de la industria del papel.

Palabras clave: Papel, Baliza, BLE, Flutter, Raspberry Pi, Firebase

ABSTRACT

The paper industry, by deforesting large areas of forests to obtain the raw material (pulp) used to make paper, is one of the industries that contributes most to climate change. The lack of trees means that carbon dioxide in the atmosphere is not reabsorbed as part of the photosynthesis process. Paper production is also one of the largest consumers of energy which comes from non-renewable sources (Environmental Paper Network, 2018).

The present degree work aims to present a digital alternative to print advertising. This alternative consists of the implementation of a proximity marketing system (Jeon, She, Soonsawad, & Ng, 2018), an idea that is being worked on in other countries. The system consists of two cross-platform applications (with mobile focus) built with Flutter, a BLE beacon (implemented on Raspberry Pi) and Firebase as the backend of the system. The proposed alternative hopes to become a solution to the problem posed by deforestation and high energy consumption of the paper industry.

Keywords: Paper, beacon, BLE, Flutter, Raspberry Pi, Firebase

1. INTRODUCCIÓN

1.1 Planteamiento del Problema

Los folletos, volantes, flyers, panfletos, pasquines y demás formatos cuyo soporte tradicional ha sido el papel se han utilizado durante ya un largo tiempo, para entregar información a las personas acerca de promociones (para el caso de productos o servicios) o emitir comunicados respecto de cualquier tópico que sea de interés público (Environmental Paper Network, 2018).

La deforestación de los bosques tropicales tiene en la producción del papel a una de sus principales causas. Se estima que la demanda de este producto crecerá a largo de la segunda mitad de este siglo. El 40% de la tala industrial de árboles se dedica a la elaboración de papel y se lo proyecta un crecimiento de hasta el 50% en el futuro cercano. En los últimos 20 años el uso de productos elaborados a partir o con base en el papel ha crecido desde 92 millones de toneladas a 208 millones de toneladas, lo cual representa un crecimiento del 126%. La industria de la pulpa y el papel se encuentra en el cuarto lugar entre los mayores consumidores de energía y en consecuencia, esta industria es una de las mayores emisoras de gases de efecto invernadero (Green America, 2017).

La información impresa por su naturaleza es inflexible, estática y no inmediata. Esto hace que si el entorno de pronto cambia entonces la información presentada carecerá de valor y será de poca utilidad para el receptor de esta. Si el establecimiento o comercio tiene la necesidad inmediata de comunicar a sus clientes fijos o potenciales acerca de alguna promoción que busque aprovechar alguna circunstancia que les es favorable, con el método tradicional eso sería imposible ya que el proceso necesario para obtener la publicidad impresa y repartirla toma varios días (Papyrus Ltda, 2019).

En este sentido, para dar respuesta al problema planteado en el presente proyecto de titulación se propone la construcción de un sistema de software cuyos componentes son: un aplicativo multiplataforma, el backend correspondiente, un dispositivo IoT conocido como baliza BLE (BLE beacon) y finalmente conceptos e ideas tomadas de lo que Google ha bautizado como Physical Web (Google LLC., s.f.).

La naturaleza flexible, inmediata e inmaterial de los sistemas informáticos, en particular aquellos que involucran aplicativos web, móviles y/o multiplataforma, como el propuesto en este plan, los convierten en el medio adecuado para transmitir información promocional y de cualquier otra naturaleza al usuario o cliente. Ya que la información es desmaterializada no es necesario el uso de impresos lo que puede provocar una reducción en la demanda de papel y con ella reducir la emisión de gases de efecto invernadero. La flexibilidad del sistema le otorga al establecimiento o comercio que publica la información la capacidad de adaptar sus ofertas o promociones a la dinámica del entorno de forma inmediata y tratar de maximizar la venta de sus productos y/o servicios.

1.2 Objetivo General

Desarrollar un sistema de anuncios digitalizados a partir de una baliza BLE, el protocolo Eddystone y un aplicativo multiplataforma.

1.3 Objetivos Específicos

- Diseñar la arquitectura del sistema que está conformado por el dispositivo baliza BLE, el aplicativo multiplataforma (que conforma el frontend) y Firebase (como backend).
- Configurar un Raspberry Pi de tal manera que este dispositivo pase a ser el mecanismo emisor de los mensajes (balizas BLE) del sistema propuesto.

- Crear los mecanismos necesarios y adecuados para que el smartphone sea capaz de captar la señal emitida por la baliza y reaccione emitiendo una notificación hacia el usuario.
- Hacer uso de SCRUM para desarrollar la aplicación multiplataforma. según su perfil podrá publicar información para compartirla.
- Implementar una base de datos que permita gestionar la información.
- Realizar pruebas de funcionalidad y usabilidad.

1.4 Alcance

Una aplicación multiplataforma (con enfoque móvil) construida con el framework Flutter y el lenguaje Dart. Firebase (el backend como servicio proporcionado por Google) es la plataforma que soporta el backend del sistema. Finalmente, se configura el protocolo Eddystone en una Raspberry Pie, en su versión 3 B+, lo cual transforma al dispositivo en una baliza BLE.

El sistema descrito se despliega en el minimarket Muyu ubicado en el sector de La Armenia, en el Valle de los Chillos, en el camino hacia el camposanto Colinas de Paz.

La información de las promociones es editada y publicada a través del módulo del aplicativo móvil dispuesto para el evento, el cual le proporciona una interfaz al usuario designado por los propietarios del establecimiento para llevar a cabo las mencionadas tareas. Este usuario tendrá los roles de editor y publicador.

La publicidad respecto de las promociones es mostrada al usuario (cliente o cliente potencial) a través del módulo del aplicativo móvil designado para el evento. La interacción del usuario con la interfaz de este módulo de la aplicación está limitada a la navegación entre las promociones disponibles, razón por la cual el usuario no necesita realizar un inicio de sesión.

La URL que corresponde a las ofertas y promociones publicadas por el establecimiento es difundida por la baliza BLE. El efecto de la exposición del smartphone a la señal emitida por la baliza es la emisión de una notificación en el smartphone que alertará al usuario respecto de ofertas y promociones que se encuentran disponibles en el minimarket Muyu.

1.5 Marco Teórico

Bluetooth Low Energy (BLE) fue introducido en el 2010, el mismo es diseñado y mantenido por el SIG (Special Interest Group) para hacer un uso más restrictivo (a diferencia de su versión clásica) de la energía necesaria para su operación y con ello permitir que los dispositivos electrónicos creados para IoT tengan una alternativa forma de comunicación. Bluetooth (ahora Bluetooth Classic) es bien conocido, difundido y popular en el mundo de la comunicación inalámbrica y es justamente este hecho del que se quiere aprovechar BLE para posicionarse en el mercado IoT. BLE transmite datos a través de 40 canales (3 para publicidad y 37 para datos) en la banda no licenciada de los 2,4GHz (banda ISM), cada canal tiene una separación de 2MHz y soporta varias topologías de red. Los dispositivos BLE son además utilizados como dispositivos de posicionamiento interno o bajo techo para lo cual tiene las siguientes características: anunciación lo que permite detectar la presencia del dispositivo, direccionamiento a través de RSSI y finalmente medición de distancias (Bluetooth SIG).

La tecnología de las balizas BLE surgió en el año 2014. Para el año 2015 varias compañías se dedicaron a implementar esta tecnología obteniendo como resultado una multitud de opciones comerciales que se encuentran disponibles en el mercado, mientras que otras iniciaron un amplio despliegue de estos dispositivos en sus entornos. Las balizas BLE son dispositivos electrónicos que difunden pequeños paquetes de datos

a tiempos regulares, anunciando su presencia a otros dispositivos que se encuentran escaneando las señales Bluetooth próximas. Las balizas difunden el perfil de acceso genérico (GAP por sus siglas en inglés) lo cual implica que estos dispositivos se anuncian, pero no aceptan conexiones entrantes, entonces la comunicación es de una sola vía desde la baliza hacia el smartphone. La información difundida por la baliza es codificada dentro de los campos del paquete de datos, campos que tienen sus propios tipos, el límite para el tamaño del paquete de datos es de 31 bytes. Ya que la tecnología BLE es abierta, no existe una obligación de usar un formato específico para la codificación de la información dentro del paquete de datos, de hecho, existen varios esquemas de codificación (formatos de marco) que son utilizados para este fin (Akpinar, 2021).

- Eddystone es un formato de marco desarrollado por Google, este define la estructura del mensaje que es difundida por la baliza. Eddystone define varios tipos de marco entre los cuales se encuentran (Google LLC, 2016):
- UID, es el identificador único de la baliza, está compuesto de 16 bytes, los 10 primeros hacen referencia al espacio de nombres mientras que los 6 bytes restantes corresponden a una instancia del espacio de nombres.
- URL, en este marco se codifica de forma comprimida la URL que la baliza se encuentra difundiendo. La compresión se da debido al espacio limitado del paquete de datos (para anunciar) es limitado.
- TLM, a través de este marco se difunde métricas respecto de la baliza, tales como capacidad de la batería, temperatura y la cuenta de los paquetes transmitidos.
- EID, este marco surgió como respuesta a la necesidad de mejorar la seguridad de las balizas. Proporciona privacidad de tal manera que la información transmitida solo pueda ser observada por los usuarios con los permisos necesarios.

En el presente trabajo se hace uso de este marco y su formato para construir el mensaje que se difunde a través de la baliza.

Una de las aplicaciones que tiene esta tecnología es el marketing de proximidad. Esta forma de marketing permite llevar una personalización masiva y en tiempo real de las promociones que tiene disponible un comercio. Las empresas usan el marketing de proximidad con el fin de mejorar sus servicios y la experiencia del cliente. El marketing de proximidad tiene la capacidad de, en tiempo real, adaptar la oferta marketing y personalizarla según donde se encuentre el (posible) cliente. El beneficio es también para los clientes ya que estos pueden percibir que su tiempo de compras es bien aprovechado además de producirles un ahorro de dinero al aprovechar las promociones que el comercio tenga disponibles en el momento que el cliente pasa por el establecimiento (Navalkrushna, Balasaheb, & Swapnesh, 2016).

Además de la baliza BLE este sistema cuenta con un frontend, el cual es un aplicativo multiplataforma, y un backend soportado por Firebase. Para la construcción del aplicativo se hará uso de Flutter, el cual es un framework open source desarrollado por Google como medio para construir aplicativos móviles, para el escritorio y además para la web con una sola base de código y compilados nativamente (código de máquina para ARM, Intel, así como también para JavaScript). Dart, lenguaje de programación OPP desarrollado también por Google, es el lenguaje en el que está escrito Flutter (<https://flutter.dev/>). Firebase es la propuesta de Backend como Servicio (BaaS), ofertada por Google. Firebase posee varios servicios que pueden ser utilizados en conjunto para construir un backend robusto que soporte las actividades de los usuarios llevadas a cabo en la aplicación y para diferentes casos de uso, además ofrece una capa de uso gratuita lo cual permite que se pueda probar nuevas ideas de negocio más

allá de la teoría, recoger datos y realmente validar las hipótesis que se puedan generar en torno a la propuesta de negocio emergente (Google LLC, s.f.).

Con base en el problema planteado, el trabajo de titulación se concentrará en el cómo aprovechar las características que ofrece la tecnología de balizas Bluetooth de Bajo Consumo Energético (BLE beacons) para utilizarlo en el desarrollo de un sistema de software que difunda anuncios informativos y/o publicitarios. El sistema utiliza la baliza BLE para difundir la URL de donde el aplicativo (instalado en el smartphone del usuario) recupera la información de las ofertas que el establecimiento o comercio ha publicado. El usuario es alertado de la presencia de promociones por medio de una notificación, la misma que es lanzada por el smartphone del usuario una vez que este entra en la zona de alcance de la señal Bluetooth, emitida por la baliza. Una vez que el usuario acciona la notificación, el aplicativo entra en acción y le presenta el catálogo de ofertas y promociones que se encuentra disponible. La tecnología Bluetooth ha venido formando parte de la electrónica de consumo mucho antes de la aparición de los smartphones y su nueva versión BLE forma parte de la pila de comunicaciones de los nuevos dispositivos electrónicos de uso común y por lo tanto cuentan con la capacidad para recibir la señal emitida por las balizas sin importar su marca y/o modelo.

La ubicuidad de la tecnología BLE hace de esta una base firme para la construcción del sistema de software propuesto. La generación de la notificación en el smartphone se aprovechará de esta característica al hacer uso del soporte BLE brindado por el sistema operativo Android. Esta ubicuidad también se evidencia en los diversos dispositivos de hardware que forman parte de lo que se conoce como Internet de las Cosas (IoT por sus siglas en inglés). Las balizas BLE, un tipo de dispositivo IoT, que tiene entre sus funciones la difusión ininterrumpida y a intervalos regulares de tiempo de mensajes de longitud corta, será utilizado para difundir la URL del aplicativo multiplataforma que le presentará al usuario la información que ha sido publicada. En conclusión, el uso generalizado de la tecnología BLE facilitará el desarrollo del sistema de software y su futura adopción por parte de los usuarios (bin Aftab, 2017).

La existencia de al menos dos protocolos de comunicaciones para balizas BLE, cuyo desarrollo y mantenimiento se encuentra apoyado por Google para el caso de Eddystone y Apple para el caso de iBeacon, facilita la tarea de implementación del sistema. Para el proyecto planteado se ha escogido al protocolo Eddystone debido a sus particulares características entre las cuales se destacan las siguientes (Bhargava, 2018):

- Es compatible tanto con Android como con iOS y otras plataformas.
- Considera de manera más amplia a la especificación de BLE lo que significa que se tiene más oportunidades para aprovechar todas las posibilidades que ofrece esta tecnología.
- Es publicado por Google como open source.
- Tiene características de seguridad incorporadas.

En conclusión, los tópicos que se abordarán durante el desarrollo de este proyecto de titulación son BLE, balizas BLE, el protocolo Eddystone y desarrollo de aplicaciones iOS.

El marco de trabajo Scrum para la gestión del Ciclo de Vida del Desarrollo de Software (SDLC por sus siglas en inglés), posee las características ágiles que facilitarán la conducción del proyecto de titulación planteado. Estas características son (Schwaber & Sutherland, 2020):

- Actividades limitadas temporalmente, debido a que los Sprints involucran lapsos cortos de tiempo, entonces se puede obtener una retroalimentación rápida respecto de las características del software implementadas en dicho Sprint.

- Se puede priorizar las diferentes características del software bajo desarrollo. Así y de esta manera se puede comenzar desarrollando las funciones que más valor entregue el aplicativo a sus usuarios.
- Es un proceso colaborativo y auto organizado. Brinda la libertad adecuada al desarrolladora o desarrollador para que este implemente la característica o conjunto de características asignadas.
- La toma de decisiones se concentra en el equipo de desarrollo.

Al hacer uso de Google Scholar para hacer una búsqueda con base en los términos: “ble beacons” y scrum, se puede hallar varios papers que indican o dan entender que el marco de trabajo Scrum se viene usando para llevar cabo el desarrollo de proyectos de software que involucren o contengan elementos de IoT como lo son las balizas BLE. Por lo tanto y en particular resulta válido hacer uso de Scrum para conducir el proyecto propuesto considerando además que el mismo no involucra, en lo absoluto, la construcción o ensamblaje de elementos electrónicos. Como ejemplo de lo dicho uno de los papers hallados junta todos estos conceptos y obtiene un sistema de software que se aprovecha de las posibilidades que brinda las balizas BLE (Bravo, 2015).

En conclusión, la naturaleza de Scrum hace de este marco de trabajo la vía adecuada para conseguir que la expectativa planteada para este trabajo de titulación pase a convertirse una realidad.

El objetivo del presente trabajo de titulación es entregar una alternativa viable, digital y moderna al papel como medio de transmisión de información, que además disponga de más prestaciones. Con ello contribuir a la disminución de la deforestación causada por la tala indiscriminada de árboles a fin de convertirlos en papel.

Como consecuencia del desarrollo de la solución propuesta al problema planteado, se obtendrá, como resultado, un sistema de anuncios digitalizado y automatizado que será un canal alternativo al tradicional para difundir información que puede resultarle de interés al usuario que la recibe. El uso masivo de un sistema como el planteado coadyuvará a reducir el consumo del papel y junto con ello disminuir los efectos negativos que su elaboración trae consigo.

Las condiciones actuales en las cuales se desenvuelve la sociedad hacen posible que la propuesta del desarrollo un sistema de anuncios digitalizados difundidos mediante balizas BLE, tenga cabida. Si a lo mencionado se le suma el hecho de que esta tecnología podría estar desplazando al papel como medio para presentar información, lo cual implica una disminución de su consumo y por ende de la deforestación de los bosques, entonces el desarrollo de este proyecto de titulación se encuentra plenamente justificado.

2. METODOLOGÍA

Scrum es el vehículo que se viene utilizando en la industria del software para tomar las ideas y transformarlas en productos y/o servicios de software funcionales que le entreguen valor al usuario. El marco Scrum viene siendo usado durante ya muchos años en diversos proyectos de desarrollo de software con éxito, de lo dicho se puede colegir que la aplicación del mencionado marco de trabajo en el proyecto de titulación propuesto le dotará a este de una sólida base a partir de la cual estructurar el conjunto de actividades que transformarán a la idea en un producto útil y en consecuencia de mucho valor.

2.1 Especificación de Requerimientos

Un sistema como el propuesto no existe en el mercado ecuatoriano entonces y con el fin de guiar la construcción del sistema de anuncios digitalizados (también conocido como marketing de proximidad), los requerimientos han sido tomados de tres fuentes, a saber:

- Papers que indican las características que deben poseer este tipo de sistemas.
- Publicaciones en línea de empresas que se dedican a la construcción de este tipo de sistemas.
- Información proporcionada por el propietario del establecimiento donde se pondrá a operar el sistema en su versión de prueba de concepto.

Dado los antecedentes previamente expuestos, el sistema debe cumplir con los siguientes requerimientos:

- Debe ser capaz de transmitir señales inalámbricas mediante el uso de tecnología Bluetooth.
- El dispositivo Bluetooth (la baliza) que transmite la señal no permite conexiones hacia sí mismo, entonces la comunicación se da en una sola vía desde el dispositivo hacia cualquier smartphone que se encuentre dentro de la zona de cobertura de la señal.
- Una aplicación móvil, El Pregonero Clientes, instalada en el smartphone del usuario (cliente o potencial cliente del establecimiento) que cuente con las siguientes capacidades:
 - Interpretar el mensaje emitido por la baliza.
 - Recuperar la parte de la URL del mensaje.
 - Hacer el respectivo requerimiento al servidor apuntado por la URL.
 - Mostrarle al usuario la lista de anuncios de las promociones que ha publicado el establecimiento comercial.
- Al usuario responsable de publicar los anuncios de las promociones por parte del establecimiento comercial se le dota con una aplicación móvil, El Pregonero Comercios, que le permite:
 - Registrarse con un nombre de usuario y contraseña.
 - Adherir anuncios de promociones a la lista de Anuncios:
 - Cargar imágenes o tomar fotografías del producto.
 - Capturar la descripción del anuncio.
 - Registrar el precio de venta al público (pvp) del producto.

- Registrar el porcentaje de descuento (%d) que se quiere aplicar al producto.
- Publicar el anuncio de la promoción.
- Mostrar la lista de anuncios.
- Eliminar anuncios de la lista de anuncios.

La modularización es la manera en la que los enunciados de los requerimientos se transforman en el sistema propuesto. A continuación, se presenta los módulos en los que se ha dividido el sistema con el fin de organizar, facilitar y controlar el avance de la construcción su construcción. Los módulos solo hacen referencia a la parte software del sistema mas no a la parte del hardware (la baliza BLE). El sistema se descompone en dos grupos de módulos, a saber:

- En cuanto se refiere al aplicativo móvil para el usuario cliente, este se compone de los siguientes módulos:
 - Módulo de visualización de los anuncios, a través del cual el cliente puede observar las promociones que se encuentran vigentes y por las cuales puede optar al momento de acceder al establecimiento que las ha publicado.
 - Módulo para la gestión de las comunicaciones, que se encarga de gestionar la comunicación entre la baliza, el aplicativo y el backend.
- En cuanto se refiere al aplicativo móvil para el usuario del comercio, este se compone de los siguientes módulos:
 - Módulo inicio de sesión o registro de usuario nuevo, a través del cual se realiza el ingreso a las opciones de la aplicación.
 - Módulo registro de anuncios, le dota al usuario de las herramientas necesarias para crear un anuncio respecto del producto en promoción.
 - Módulo visualización y edición de los anuncios, muestra el listado de anuncios que se encuentran vigentes. Este módulo cuenta también con mecanismos para realizar las actividades de eliminación o edición de las publicaciones.

En cuanto requerimientos no funcionales se tiene los siguientes:

- El sistema solo le debe permitir el ingreso, edición o eliminación de anuncios a los usuarios que hayan iniciado sesión en el aplicativo El Pregonero Comercios.
- El sistema le debe permitir al usuario cambiar a la aplicación a modo oscuro.
- Se debe adjuntar el nombre de usuario al anuncio creado por el usuario.
- Asignar un esquema de colores adecuado para la aplicación.
- Las aplicaciones El Pregonero Comercios y El Pregonero Clientes deben ser mobile first, es decir que su diseño debe considerar que el principal entorno de ejecución son los smartphones.
- La interfaz de usuario debe simple y cumplir con criterios de usabilidad.

2.2 Definición de Roles

En la siguiente tabla se presenta los roles SCRUM, que son utilizados para organizar el trabajo requerido para construir el sistema:

Tabla 1 – Roles SCRUM y sus responsables

Rol	Responsable
Desarrollador	Efrén Sandoval
Product Owner	MSc. Hernán Ordoñez
Scrum Master	MSc. Hernán Ordoñez
Stackholder	Edgar Carlosama - Propietario Micromercado Muyu

2.3 Historias de Usuario

La funcionalidad requerida del sistema se especifica por medio de historias de usuario. En la siguiente tabla se presentan todas las historias de usuario según uno de los formatos más conocidos: “Como (tipo de usuario), yo (quiero | puedo | necesito) (objetivo) (tal que | de tal manera que) (razón)” (Rehkopf). Además, y por cada historia de usuario, se presentan los criterios de aceptación lo que permite definir cuando una historia se ha completado.

Tabla 2 – Historias de usuario y sus criterios de aceptación

Código	Historia de Usuario	Criterio de Aceptación
T01	Como usuario de la aplicación dedicada a los establecimientos comerciales, quiero que mis clientes se enteren de las promociones que estoy ofertando tal que pueda conseguir más ventas.	La baliza BLE sea capaz de: <ul style="list-style-type: none"> Emitir mensajes a través de Bluetooth. Anunciar (difundir) la URL del catálogo de promociones que el establecimiento ha publicado.
T02	Como usuario de la aplicación dedicada a los establecimientos comerciales, quiero que al menos una imagen del producto en promoción forme parte del anuncio de tal manera que los clientes puedan apreciarlo.	La aplicación debe ser capaz de: <ul style="list-style-type: none"> Mostrar la pantalla <i>Anuncio</i> que contiene el formulario con los campos del anuncio. Permitir al usuario tomar una fotografía del producto con la cámara del smartphone. Permitir al usuario navegar por la galería de imágenes del smartphone y escoger una de ellas.
T03	Como usuario de la aplicación dedicada a los establecimientos comerciales, quiero poner en el anuncio una reseña de la promoción de tal manera que los clientes pueden familiarizarse con el mismo.	La aplicación debe ser capaz de: <ul style="list-style-type: none"> Capturar el texto ingresado por el usuario (reseña del anuncio) y validarlo.

T04	Como usuario de la aplicación dedicada a los establecimientos comerciales, puedo fijar el descuento del producto como un porcentaje del <i>Precio de venta al público (pvp)</i> .	<p>La aplicación debe ser capaz de:</p> <ul style="list-style-type: none"> • Mostrar el teclado numérico. • Capturar el valor de <i>Precio de venta al público (pvp)</i> del producto ingresado por el usuario y validarlo. • Capturar el <i>Porcentaje de Descuento (%d)</i> que se desea aplicar y validarlo.
T05	Como usuario de la aplicación dedicada a los establecimientos comerciales, necesito observar el catálogo de promociones, escoger un anuncio, actualizarlo y/o eliminarlo.	<p>La aplicación debe ser capaz de:</p> <ul style="list-style-type: none"> • Calcular el <i>Precio con Descuento (pcd)</i> de la promoción al aplicar el %d al pvp. • Mostrar la pantalla <i>Anuncios</i> la cual contine una lista de los anuncios creados. • Permitir que el usuario seleccione un anuncio y lo actualice. • Permitir que el usuario seleccione un anuncio y lo elimine.
T06	Como administrador de la aplicación dedicada a los establecimientos comerciales, quiero que solo las personas que cuenten con un usuario debidamente registrado puedan editar (crear, actualizar o eliminar) anuncios del catálogo de promociones.	<p>La aplicación debe ser capaz de:</p> <ul style="list-style-type: none"> • Mostrar la pantalla <i>Registro de Usuario Nuevo</i> al usuario. • Permitir que el usuario ingrese su nombre de usuario (email) y validarlo. • Permitir que el usuario ingrese el texto que será su contraseña y validarlo. • Permitir que el usuario solicite el registro. • Devolver al usuario a la pantalla de <i>Inicio de Sesión</i> si el registro ha sido exitoso caso contrario informarle del error ocurrido.
T07	Como administrador de la aplicación dedicada a los establecimientos comerciales, quiero que los anuncios sean	<p>La aplicación debe ser capaz de:</p> <ul style="list-style-type: none"> • Mostrar la pantalla <i>Inicio de Sesión</i> al usuario. • Permitir el ingreso de credenciales y solicitud de

	creados por usuarios que hayan ingresado a la aplicación.	<p>inicio de sesión por parte del usuario.</p> <ul style="list-style-type: none"> • Validar las credenciales del usuario. • Permitir la navegación del usuario hasta la pantalla <i>Anuncios</i> siempre y cuando la validación del inicio de sesión haya sido exitosa caso contrario informarle del error ocurrido. • Llevar al usuario desde la pantalla de <i>Inicio de Sesión</i> a la pantalla de <i>Registro de Usuario Nuevo</i> y mostrársela.
T08	Como usuario de la aplicación dedicada a los establecimientos comerciales, quiero ser capaz de gestionar el estado de mi perfil.	<p>La aplicación debe ser capaz de:</p> <ul style="list-style-type: none"> • Mostrar al usuario la pantalla de <i>Perfil y Configuraciones</i>. • Permitir al usuario tomar una fotografía de sí mismo (selfie) con la cámara del smartphone. • Permitir al usuario navegar por la galería de imágenes del smartphone y escoger una de ellas. • Permitir que el usuario pueda cerrar su sesión.
T09	Como usuario de la aplicación dedicada a los clientes, quiero ser informado de las promociones que se ofertan.	<p>La aplicación debe ser capaz de:</p> <ul style="list-style-type: none"> • Lanzar una notificación cuando se encuentre en la zona de cobertura de la baliza BLE. • Reaccionar al toque del usuario sobre la notificación, mostrando la pantalla de <i>Promociones</i>. • Mostrar la pantalla <i>Promociones</i>, la cual contiene la lista de los anuncios que el establecimiento ha publicado.
T10	Como usuario de la aplicación dedicada a los clientes, necesito guardar los anuncios de las promociones que me parecen interesantes tal que no tenga que	<p>La aplicación debe ser capaz de:</p> <ul style="list-style-type: none"> • Mostrar la pantalla <i>Promoción</i> con la información del anuncio que el usuario

	<p>volverlas a buscar en la lista principal.</p>	<p>seleccionó en la pantalla <i>Promociones</i>.</p> <ul style="list-style-type: none"> • Permitirle al usuario marcar como favorito el anuncio de la promoción que se haya elegido. • Mostrar la pantalla <i>Favoritos</i> que contiene la lista de los anuncios de promociones marcados como favoritos por el usuario. • Permitirle al usuario eliminar anuncios de su lista de favoritos.
--	--	---

2.4 Estimación del Product Backlog

En la siguiente tabla, por cada fila se presenta la prioridad y un estimado del tiempo que tomará llevar cada historia de usuario desde su estado inicial al estado final o realizado (done):

Tabla 3 – Priorización y estimación de la duración de la implementación de cada historia de usuario

Código	Prioridad	Duración
T01	alta	2 días
T02	alta	1 día
T03	alta	1 día
T04	alta	2 día
T05	alta	2 días
T06	media	2 días
T07	media	2 días
T08	media	1 día
T09	alta	4 día
T10	alta	2 días

La prioridad se califica en el rango de baja, media y alta. La prioridad de implementación de cada historia de usuario depende del valor que esta le otorga al sistema en general mientras se la compara con las otras historias de usuario. El Product Owner es el encargado de comparar y definir el rango que le corresponde a cada historia de usuario. Dado que este sistema es un prototipo que tiene por objeto validar una propuesta de

negocio entonces solo se ha considerado implementar las características fundamentales del sistema, razón por la cual ninguna historia de usuario ha sido calificada como baja.

Considerando la tabla de historias de usuario además de la tabla de estimación, se lleva a cabo la planificación para las liberaciones (release planning), la misma que se muestra en la siguiente tabla:

Tabla 4 – Distribución de las historias de usuario en cada sprint

Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6
T01	T02	T04	T09	T06	T08
	T03	T05	T10	T07	

2.5 Arquitectura del Sistema

El sistema se compone de cuatro partes, las cuales a continuación se pasan a describir:

- El Pregonero Comercios, es el aplicativo móvil para los establecimientos comerciales. Está compuesto de tres módulos:
 - Gestión de anuncios, le permite al usuario crear, actualizar y/o eliminar anuncios promocionales.
 - Gestión de usuarios, le permite al usuario administrador asignar roles (editor o publicador o ambos) a los usuarios registrados.
 - Gestión de registro y autenticación, le permite al usuario registrarse y autenticarse en la aplicación.
- El Pregonero Clientes, es el aplicativo móvil para los clientes. El módulo *Catálogo de Anuncios* es su único módulo y tiene por objeto albergar toda la lógica asociada con la visualización y marcado (como favorito) de los anuncios.
- El Pregonero Baliza, es el dispositivo electrónico encargado de difundir la URL del catálogo de promociones del establecimiento que los está empleando.
- El Pregonero Backend, es el encargado de los servicios de autenticación, base de datos NoSQL y almacenamiento de información no estructurada (imágenes o fotografías que forman parte del anuncio publicitario de la promoción). Firebase, el backend como servicio (BaaS) de Google (Google LLC, s.f.), es la plataforma que soporta esta parte del sistema.

Las tres primeras partes conforman el frontend del sistema, mientras que la cuarta parte es su backend. En el siguiente diagrama de arquitectura se puede apreciar al sistema y sus partes componentes:

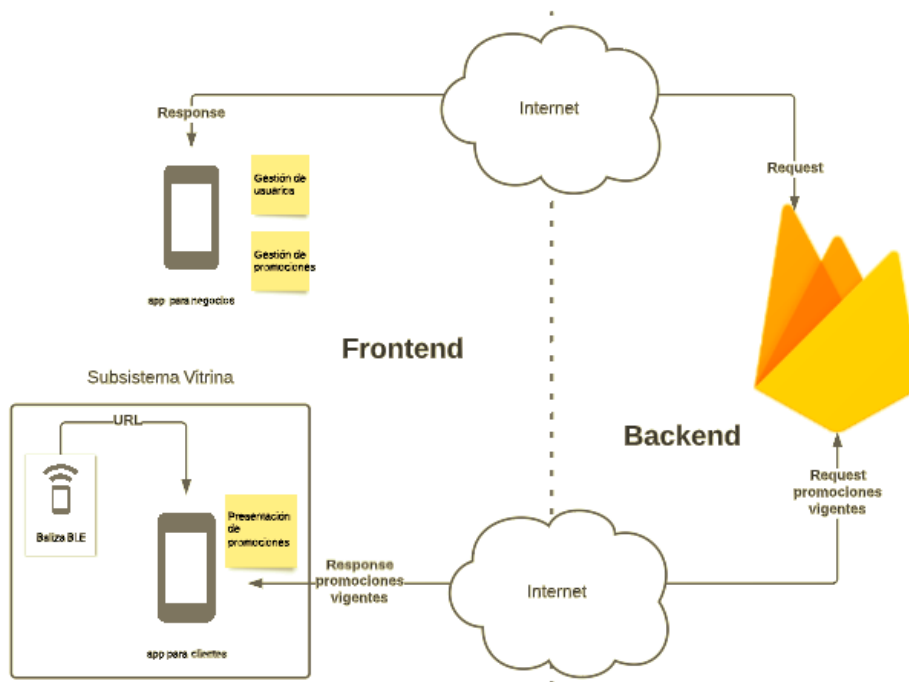


Figura 1 – Arquitectura del Sistema de Anuncios Digitales

2.6 Implementación

En esta parte del documento se pasa a describir las actividades realizadas sprint por sprint con el fin de ensamblar el sistema y finalmente ponerlo en operación.

Además de los 7 sprints mencionados en la tabla 4 se le adhiere el sprint 0. Sprint en el que se agrupan las actividades y tareas relacionados con la preparación del ambiente de desarrollo y la configuración inicial de la Raspberry Pi.

2.6.1 Sprint 0

Este Sprint será utilizado para llevar a cabo la instalación y configuración de las herramientas del ambiente de desarrollo utilizadas para la construcción del sistema. Además, se realiza la instalación de Raspberry Pi OS, una distro GNU/Linux ensamblada a partir de Debian, el cual es el sistema operativo oficial para dispositivos Raspberry Pi (Raspberry Pi Ltd).

En la siguiente tabla se presenta todas las herramientas que conforman el entorno de desarrollo utilizado para la construcción del sistema:

Tabla 5 – Herramientas del entorno de desarrollo

Nombre	Descripción	Alcance	Versión
Flutter	<p>“Flutter es un framework de código abierto de Google para crear aplicaciones hermosas, compiladas de forma nativa y multiplataforma a partir de un único código base.”</p> <p>Es una alternativa a Java y Kotlin (entorno Android) o Swift y Objective-C (entorno iOS) usados para construir aplicaciones móviles nativas además de React Native o frameworks similares (que usan tecnología web) para desarrollar</p>	Frontend	2.10.3

	aplicativos que no dependan de la plataforma (Flutter Dev Team).		
Dart	“Dart es un lenguaje optimizado para el cliente que permite desarrollar aplicaciones rápidas en cualquier plataforma. Su objetivo es ofrecer el lenguaje de programación más productivo para el desarrollo multiplataforma.” (Dart Dev Team)	Frontend	2.16.1
Visual Studio Code	Entorno de Desarrollo Integrado (IDE) de código abierto. Liviano pero al mismo tiempo potente. Su funcionalidad se puede extender a través del uso de extensiones (plug-ins) que le permiten adaptarse para operar con distintos lenguajes y entornos de programación (Microsoft, s.f.).	Frontend y Backend	1.69.2
Firebase	Es la propuesta BaaS de Google. Sus distintos servicios se pueden combinar de diversas maneras tal que se encuentre la configuración adecuada que permita satisfacer los requerimientos de la aplicación.	Backend	N/A
Raspberry Pi	Es un computador de tamaño reducido de bajo costo cuyas especificaciones pertenecen a la Raspberry Pi Foundation afincada en el Reino Unido. Su objetivo fundamental es ser una herramienta a través de la cual se pueda enseñar de informática, pero ahora también es utilizada en el entorno de la domótica (Raspberry Pi Foundation, s.f.).	IoT	3 B+
Raspberry Pi OS	Distro de GNU/Linux que toma a Debian como su base. Este es el SO oficial de los dispositivos Raspberry Pi (Raspberry Pi Ltd).	IoT	
BlueZ	Es la pila de protocolos Bluetooth de fuente abierta, oficial para Linux. A través de este software se puede configurar y manipular el hardware Bluetooth de la Raspberry Pi (BlueZ Project, 2022).	IoT	5.64
Smartphone Android	Dispositivo físico dispuesto para llevar a cabo las pruebas de los aplicativos bajo desarrollo.	Frontend	Marshmallow API Level 23

2.6.1.1 Sprint Planning

El objetivo del presente sprint es preparar y configurar el espacio de trabajo tal que en lo posterior el esfuerzo solo se concentre en la construcción del sistema. Las siete tareas asignadas a este sprint son:

Tabla 6 – Tareas correspondientes al Sprint 0

Código HU	Nº	Tarea
N/A	1	Descargar e instalar el IDE Visual Studio Code (VSC) del sitio oficial: https://code.visualstudio.com/
	2	Descargar el paquete Flutter (el cual incluye al compilador de Dart), en su versión más reciente, estable y que le corresponda al SO subyacente (MacOS para este caso).
	3	Configurar la variable de entorno PATH tal que los binarios de Flutter puedan ser llamados desde la interfaz de línea de comandos (CLI).
	4	A través de la interfaz correspondiente dentro de VSC instalar los plugins correspondientes al framework Flutter, así como las herramientas asociadas al lenguaje Dart tal que trabajar con VSC sea bastante cómodo.
	5	Descargar el SDK de Android para se pueda generar la versión Android de la aplicación bajo desarrollo.
	6	Adherir la ruta de los binarios del SDK Android a la variable de entorno PATH.
	7	Instalar el Raspberry Pi OS en el Raspberry Pi de tal manera que este se encuentre totalmente operativo y funcional.
	8	Crear el proyecto Flutter del aplicativo El Pregonero Comercios.
	9	Crear el proyecto Flutter del aplicativo El Pregonero Clientes.

2.6.1.2 Implementación

A continuación, se presenta tarea por tarea la evidencia asociada:

Tarea 1

La siguiente captura de pantalla muestra la interfaz gráfica del IDE Visual Studio Code con lo cual que en evidencia que el mencionado software se encuentra instalado en la máquina de desarrollo:

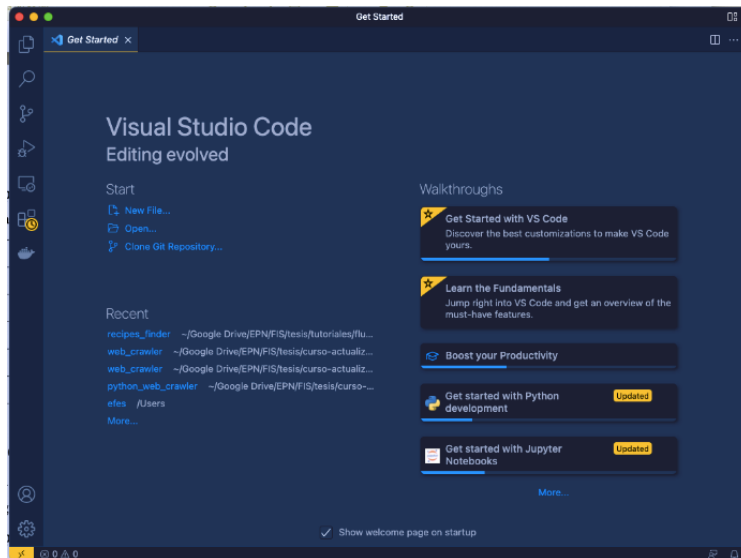


Figura 2 – Interfaz de Usuario principal del IDE VSC

Tarea 2

La pantalla de donde se puede obtener el empaquetado del framework Flutter y su posterior descarga se presentan en las imágenes a continuación:

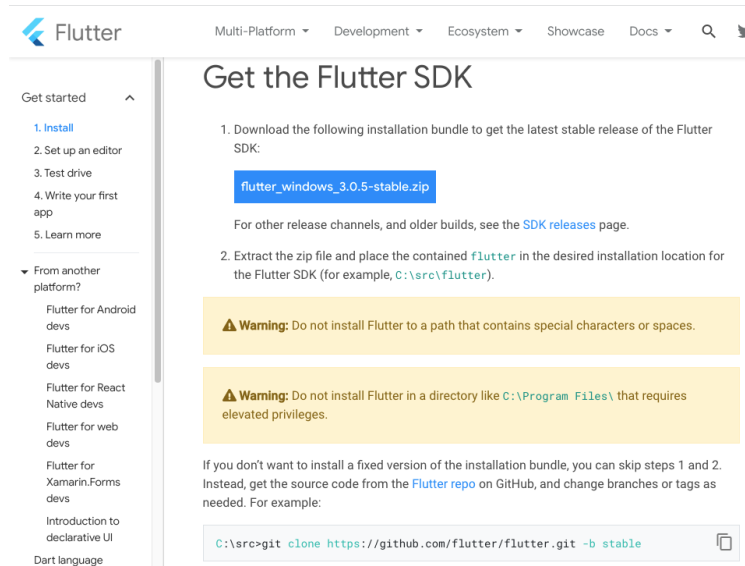


Figura 3 – Muestra del sitio web de Flutter

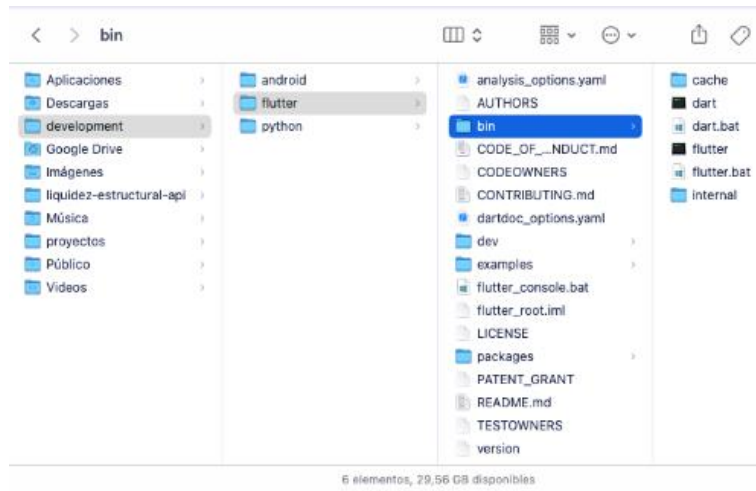


Figura 4 – Ruta del sistema de archivos donde Flutter está alojado

Tarea 3

Luego de realizar la descarga del paquete que contiene todas las herramientas para el desarrollo de aplicaciones con Flutter, se continua con su descompresión para finalizar con la configuración de la variable de entorno PATH para que esta también apunte al directorio `/bin` donde se encuentra los binarios del framework. La siguiente captura muestra la versión del framework Flutter:

```
efes@lolita ~ % echo $PATH
/Users/efes/development/android/platform-tools:/Users/efes/development/android/emulator:/Users/efes/development/android/cmdline-tools/tools/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Library/Apple/usr/bin:/Users/efes/development/flutter/bin
efes@lolita ~ %
```

Figura 5 – Variable de entorno PATH apuntando a Flutter

```
efes@lolita ~ % flutter --version
Flutter 2.10.3 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 7e9793dee1 (7 weeks ago) • 2022-03-02 11:23:12 -0600
Engine • revision bd539267b4
Tools • Dart 2.16.1 • DevTools 2.9.2
efes@lolita ~ %
```

Figura 6 – Resultado de la ejecución del comando `flutter --version`

Tarea 4

Las imágenes a continuación muestran la instalación en VSC de los complementos que corresponden a Flutter y a Dart. Con esto el trabajo relacionado a la configuración del IDE queda concluido.

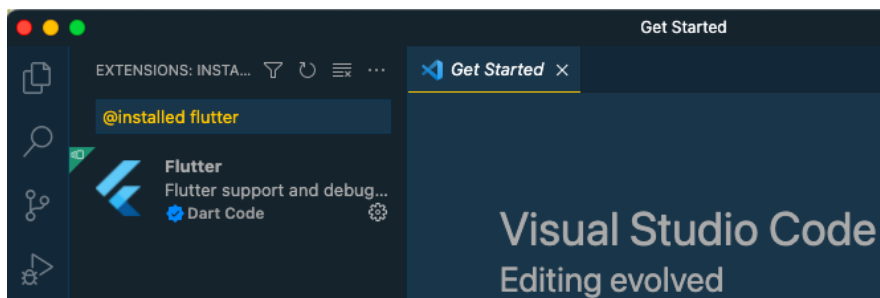


Figura 7 – Complemento Flutter instalado

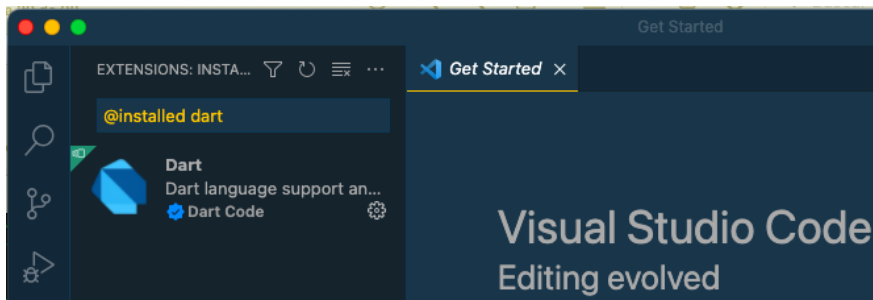


Figura 8 – Complemento Dart instalado

Tarea 5

A continuación, se presenta la ubicación en el sistema de archivos del directorio en el cual se encuentran las herramientas de desarrollo (SDK) para crear aplicaciones para la plataforma Android:

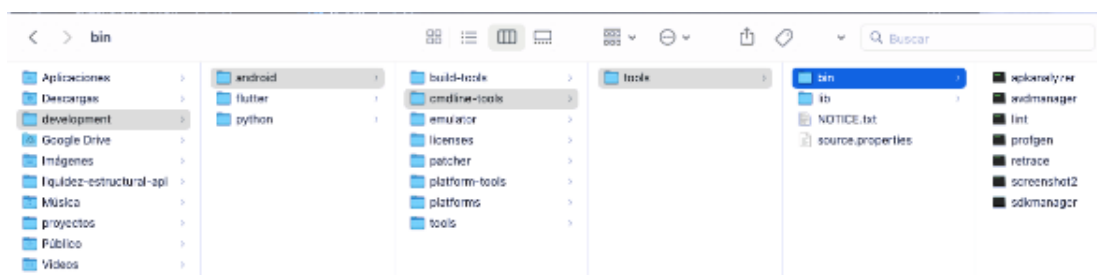


Figura 9 – SDK Android

Tarea 6

Variable de entorno PATH configurada para que apunte al directorio en el cual se encuentra el SDK Android:

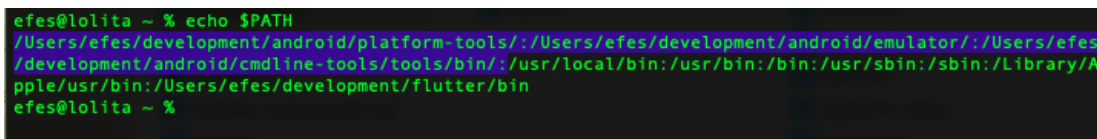


Figura 10 – Variable de entorno PATH apuntando al SDK Android

Tarea 7

Luego de la instalación del sistema operativo Raspberry Pi OS se presenta la pantalla de inicio:

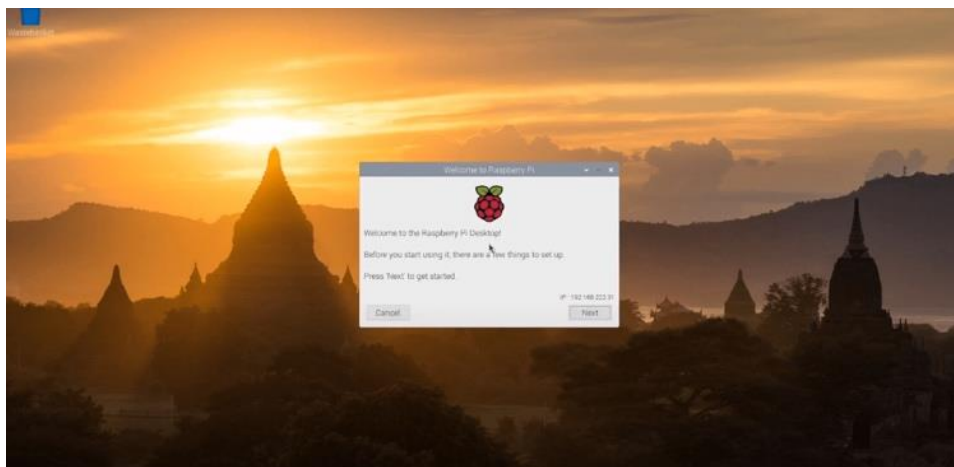


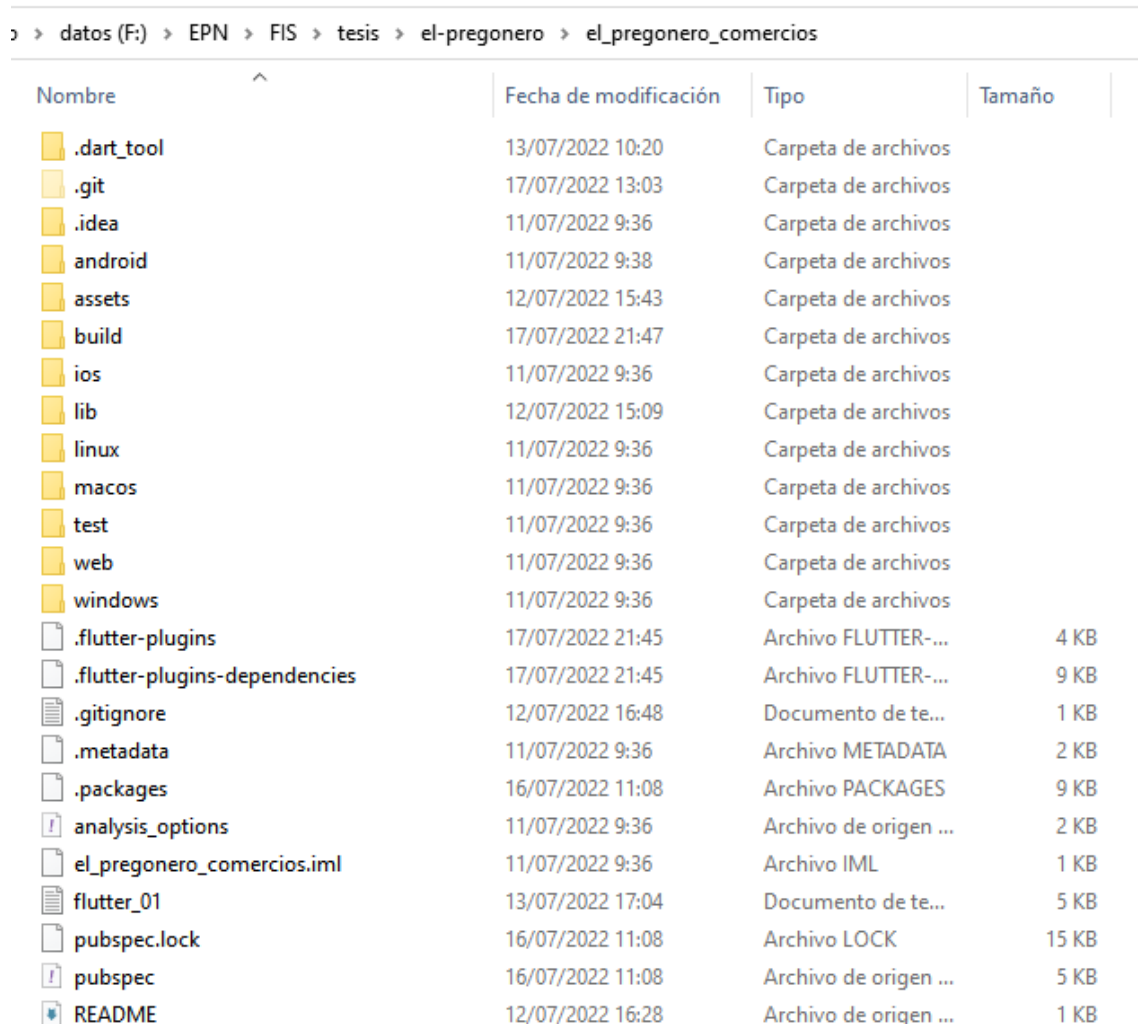
Figura 11 – Pantalla de bienvenida de Raspberry Pi OS

Tarea 8

Para llevar a cabo esta tarea se hace uso del comando `flutter create el_pregonero_comercios`. Como resultado se obtiene la generación automática de un conjunto de directorios y archivos básicos para que el proyecto pueda compilar.

Al compilar el proyecto una aplicación de demostración aparece en la pantalla del dispositivo usado para realizar las pruebas de la aplicación, esto indica que el proyecto no tiene problemas y se puede iniciar con la construcción de la aplicación.

La imagen a continuación muestra la estructura de archivos del directorio del proyecto:



Nombre	Fecha de modificación	Tipo	Tamaño
.dart_tool	13/07/2022 10:20	Carpeta de archivos	
.git	17/07/2022 13:03	Carpeta de archivos	
.idea	11/07/2022 9:36	Carpeta de archivos	
android	11/07/2022 9:38	Carpeta de archivos	
assets	12/07/2022 15:43	Carpeta de archivos	
build	17/07/2022 21:47	Carpeta de archivos	
ios	11/07/2022 9:36	Carpeta de archivos	
lib	12/07/2022 15:09	Carpeta de archivos	
linux	11/07/2022 9:36	Carpeta de archivos	
macos	11/07/2022 9:36	Carpeta de archivos	
test	11/07/2022 9:36	Carpeta de archivos	
web	11/07/2022 9:36	Carpeta de archivos	
windows	11/07/2022 9:36	Carpeta de archivos	
.flutter-plugins	17/07/2022 21:45	Archivo FLUTTER-...	4 KB
.flutter-plugins-dependencies	17/07/2022 21:45	Archivo FLUTTER-...	9 KB
.gitignore	12/07/2022 16:48	Documento de te...	1 KB
.metadata	11/07/2022 9:36	Archivo METADATA	2 KB
.packages	16/07/2022 11:08	Archivo PACKAGES	9 KB
analysis_options	11/07/2022 9:36	Archivo de origen ...	2 KB
el_pregonero_comercios.iml	11/07/2022 9:36	Archivo IML	1 KB
flutter_01	13/07/2022 17:04	Documento de te...	5 KB
pubspec.lock	16/07/2022 11:08	Archivo LOCK	15 KB
pubspec	16/07/2022 11:08	Archivo de origen ...	5 KB
README	12/07/2022 16:28	Archivo de origen ...	1 KB

Figura 12 – Estructura del directorio del proyecto `el_pregonero_comercios`

Tarea 9

Esta tarea es similar a la anterior salvo que el argumento que se le pasa al comando `flutter create` es `el_pregonero_clientes`. La siguiente imagen muestra la estructura de directorios del proyecto:

datos (F:) > EPN > FIS > tesis > el-pregonero > el_pregonero_comercios

Nombre	Fecha de modificación	Tipo	Tamaño
.dart_tool	13/07/2022 10:20	Carpeta de archivos	
.git	17/07/2022 13:03	Carpeta de archivos	
.idea	11/07/2022 9:36	Carpeta de archivos	
android	11/07/2022 9:38	Carpeta de archivos	
assets	12/07/2022 15:43	Carpeta de archivos	
build	17/07/2022 21:47	Carpeta de archivos	
ios	11/07/2022 9:36	Carpeta de archivos	
lib	12/07/2022 15:09	Carpeta de archivos	
linux	11/07/2022 9:36	Carpeta de archivos	
macos	11/07/2022 9:36	Carpeta de archivos	
test	11/07/2022 9:36	Carpeta de archivos	
web	11/07/2022 9:36	Carpeta de archivos	
windows	11/07/2022 9:36	Carpeta de archivos	
.flutter-plugins	17/07/2022 21:45	Archivo FLUTTER-...	4 KB
.flutter-plugins-dependencies	17/07/2022 21:45	Archivo FLUTTER-...	9 KB
.gitignore	12/07/2022 16:48	Documento de te...	1 KB
.metadata	11/07/2022 9:36	Archivo METADATA	2 KB
.packages	16/07/2022 11:08	Archivo PACKAGES	9 KB
analysis_options	11/07/2022 9:36	Archivo de origen ...	2 KB
el_pregonero_comercios.iml	11/07/2022 9:36	Archivo IML	1 KB
flutter_01	13/07/2022 17:04	Documento de te...	5 KB
pubspec.lock	16/07/2022 11:08	Archivo LOCK	15 KB
pubspec	16/07/2022 11:08	Archivo de origen ...	5 KB
README	12/07/2022 16:28	Archivo de origen ...	1 KB

Figura 13 – Estructura del directorio del proyecto *el_pregonero_clientes*

2.6.1.3 Sprint Review

Tabla 7 – Cumplimiento de los criterios de aceptación – Sprint 0

Código HU	Criterio de Aceptación	¿Cumple?
N/A	Descarga e instalación del IDE VSC.	Sí
	Descarga y ubicación del paquete del framework Flutter.	Sí
	Establecer la variable de entorno PATH para que apunte a los binarios de Flutter y Dart.	Sí
	Instalación de los plug-ins para Flutter y Dart en el IDE.	Sí
	Descargar y ubicar el SDK de Android.	Sí
	Establecer la variable de entorno PATH para que apunte a los binarios del SDK de Android.	Sí
	Instalación del Raspberry Pi OS.	Sí

	Creación del proyecto <i>el_pregonero_comercios</i>	Sí
	Creación del proyecto <i>el_pregonero_clientes</i>	Sí

2.6.2 Sprint 1

Para este sprint está planificado llevar a cabo las tareas que corresponden a la configuración de la Raspberry Pi y convertirla en una baliza BLE.

2.6.2.1 Sprint Planning

El presente sprint se ha dividido en nueve tareas que han sido definidas de acuerdo con el objetivo que se requiere alcanzar:

Tabla 8 – Tareas correspondientes al Sprint 1

Código HU	Nº	Tarea
T01	1	Actualizar los paquetes de software que vienen por defecto en Raspbian.
	2	Instalar los siguientes paquetes de software: git, bc, libusb-dev, libdbus-1-dev, libglib2.0-dev, libudev-dev, libical-dev, libreadline-dev.
	3	Descargar y ubicar el paquete BlueZ.
	4	Procesar e instalar el paquete BlueZ.
	5	Verificar la que la instalación haya sido exitosa.
	6	Crear el shell script de configuración del dispositivo Bluetooth de la Raspberry Pi tal que se comporte como una baliza BLE. El script incluye la definición del marco Eddystone.
	7	Probar el shell script al ponerlo en ejecución.
	8	Editar el archivo .bashrc para insertar la instrucción que corresponde a la ejecución del shell script.
	9	Reiniciar la Raspberry Pi.

2.6.2.2 Implementación

Las nueve tareas han sido llevadas a cabo de manera exitosa. A continuación, se presenta tarea por tarea la evidencia asociada:

Tarea 1

La actualización de los paquetes de software de Raspberry Pi OS se lleva a cabo desde la terminal y ejecutando los comandos que se muestran en la imagen a continuación:

```
sudo apt-get update
sudo apt-get upgrade
```

Figura 14 – Actualizando los paquetes del sistema operativo Raspberry Pi

Tarea 2

Instalación de los paquetes de los cuales depende el software BlueZ, el cual opera el dispositivo bluetooth que se encuentra incorporado en la tarjeta Raspberry Pi:

```
sudo apt-get install -y git bc libusb-dev libdbus-1-dev libglib2.0-dev libudev-dev libical-dev libreadline-dev autoconf
```

Figura 15 – Instalación de dependencias

Tarea 3

Una vez las dependencias han sido satisfechas se pasa a descarga y descomprimir el paquete BlueZ:

```
cd ~
wget http://www.kernel.org/pub/linux/bluetooth/bluez-5.49.tar.xz
tar -xvf bluez-5.49.tar.xz
cd bluez-5.49/
```

Figura 16 – Descarga del paquete BlueZ

Tarea 4

Compilación de los archivos fuente descargados se en el paso anterior a través del uso de la herramienta make pero no sin antes ejecutar el script de configuración (configure), el cual establece los directorios que alojarán a los binarios del paquete BlueZ (BlueZ Project, 2022):

```
sudo ./configure --prefix=/usr --mandir=/usr/share/man --sysconfdir=/etc --localstatedir=/var
make
sudo make install
```

Figura 17 – Instalación del paquete BlueZ

Tarea 5

Se prueba que todas las actividades previamente realizadas permiten manipular al dispositivo bluetooth, esto se realiza mediante el uso del comando hciconfig (Gennari):

```
pi@raspberrypi:~ $ hciconfig
hc10:  Type: Primary  Bus: UART
      BD Address: B8:27:EB:7F:F3:F6  ACL MTU: 1021:8  SCO MTU: 64:1
      UP RUNNING
      RX bytes:829 acl:0 sco:0 events:58 errors:0
      TX bytes:4240 acl:0 sco:0 commands:58 errors:0
```

Figura 18 – Resultado de la ejecución del comando *hciconfig*

Tarea 6

Se edita el archivo beacon.sh, el cual es un shell script que contiene el conjunto de comandos que hacen que la dispositivo Raspberry Pi se comporte como una baliza BLE:

```
#!/bin/bash

# Advertising flags
AD_FLAGS="02 01 1a"

# Beacon protocol. All items prefilled here cannot be edited, those which can are taken from
the profile
AD_LENGTH="1b"
AD_TYPE="ff"
MFG_ID="18 01"
BEACON_CODE="be ac"
BEACON_ID="{{company_id}} {{premises_id}} {{beacon_node_id}}"
REFERENCE_RSSI="{{reference_rssi}}"
MFG_RESERVED="01"

Ad_Flags=`echo "$AD_FLAGS"`
Advertisement=`echo "$AD_LENGTH $AD_TYPE $MFG_ID $BEACON_CODE $BEACON_ID $REFERENCE_RSSI
$MFG_RESERVED`

# Commands running on Raspberry Pi
BLE="hci0"

# Turn off BLE
sudo hciconfig $BLE down

# Turn on BLE
sudo hciconfig $BLE up

# Stop LE advertising
sudo hciconfig $BLUETOOTH_DEVICE noleadv

# Start LE advertising (non-connectable)
sudo hciconfig $BLE leadv 3

# Turn scanning off (can sometimes affect advertising)
sudo hciconfig $BLE noscan

# Set the Beacon
sudo hcitool -i $BLE cmd 0x08 0x0008 1f $Ad_Flags $Advertisement
```

Figura 19 – Contenido del archivo beacon.sh

Tarea 7

Una vez el archivo ha sido editado y guardado en el home del usuario pi, se continua con su ejecución a través del llamado al script:

```
sudo sh ./beacon.sh
hci0: Type: Primary Bus: UART
      BD Address: B8:27:EB:7F:F3:F6 ACL MTU: 1021:8 SCO MTU: 64:1
      UP RUNNING
      RX bytes:2277 acl:0 sco:0 events:144 errors:0
      TX bytes:5828 acl:0 sco:0 commands:144 errors:0

< HCI Command: ogf 0x08, ocf 0x0008, plen 15
  1F 02 01 1A 1B FF 18 01 BE AC 00 00 00 00 01
> HCI Event: 0x0e plen 4
  01 08 20 00
```

Figura 20 – Resultado de la ejecución del script beacon.sh

Tarea 8

Con el fin de que el script previamente editado se ejecute de forma automática y con cada reinicio de la Raspberry Pi, se edita el archivo oculto .bashrc para adherir un comando:

```
sudo vi /home/pi/.bashrc
sudo sh /home/pi/beacon.sh
```

Figura 21 – Comando adherido al archivo .bashrc

Tarea 9

Luego de realizadas todas las tareas anteriores, finalmente se pasa a reiniciar la Raspberry Pi y probar que esta se sigue comportando como una baliza BLE:

```

sudo reboot

Linux raspberrypi 4.14.49-v7+ #1 SMP Fri Jun 15 03:10:12 UTC 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jun 29 23:15:13 2018
hc10: Type: Primary Bus: UART
      BD Address: 08:27:EB:7F:F3:F6 ACL MTU: 1021:8 SCO MTU: 64:1
      UP RUNNING
      RX bytes:2277 acl:0 sco:0 events:144 errors:0
      TX bytes:5828 acl:0 sco:0 commands:144 errors:0

< HCI Command: ogf 0x00, ocf 0x0000, plen 15
  1F 02 01 1A 1B FF 18 01 BE AC 00 00 00 00 01
> HCI Event: 0x0e plen 4
  01 08 20 00
  
```

Figura 22 – Después del reinicio la baliza sigue difundiendo el mensaje

2.6.2.3 Sprint Review

Tabla 9 – Cumplimiento de los criterios de aceptación – Sprint 1

Código HU	Criterio de Aceptación	¿Cumple?
T01	Emitir mensajes a través de Bluetooth.	Sí
	Anunciar (difundir) la URL del catálogo de promociones que el establecimiento ha publicado.	Sí

2.6.3 Sprint 2

Para este sprint se encuentra planificado se lleve a cabo la construcción de dos elementos de la interfaz gráfica del aplicativo multiplataforma:

- El primer elemento es un image picker, él cual le dará al usuario la habilidad para de capturar imágenes sean estas de la galería o de la cámara e incorporar la imagen seleccionada al anuncio.
- El segundo elemento es un campo de ingreso de texto, esto con el objeto de que el usuario pueda ingresar una descripción respecto de la promoción que esta por publicar.

2.6.3.1 Sprint Planning

El presente sprint se ha dividido en siete tareas que han sido definidas de acuerdo a los objetivos planteados durante la elaboración de las historias de usuario que se corresponden con el sprint:

Tabla 10 – Tareas correspondientes al Sprint 2

Código HU	Nº	Tarea
Común	1	Editar el archivo <i>pubspec.yaml</i> e ingresar la sentencia asociada con el paquete del plugin <i>image picker</i> .

	2	Crear la pantalla <i>Anuncio</i>
T02	3	Incorporar el widget <i>image picker</i> y los widgets que le acompañan a la pantalla <i>Anuncio</i> . Mientras no se haya elegido una imagen, mostrar un ícono de una cámara en el área que le corresponde a la imagen.
	4	Tomar una fotografía y mostrar en pantalla.
	5	Tomar una imagen de la galería y mostrarla en pantalla.
T03	6	Incorporar un widget campo de texto acompañado de la leyenda <i>Descripción</i> y establecer sus validaciones.
	7	Ingresar el texto de una reseña.

2.6.3.2 Implementación

Las siete tareas han sido llevadas a cabo de manera exitosa. A continuación se presenta tarea por tarea la evidencia asociada:

Tarea 1

Para que flutter pueda hacer uso de un paquete externo, este tiene que ser declarado en el archivo *pubspec.yaml* (Flutter Dev Team) y luego ejecutar el comando *flutter get pub* para recuperar el paquete declarado:

Figura 23 – Vista del archivo *pubspec.yaml*

Tarea 2

La imagen a continuación muestra la pantalla *Anuncio* vacía:

Figura 24 – Pantalla *Anuncio* vacía

Tarea 3

En la siguiente imagen se muestra la pantalla *Anuncio* con el widget *image picker* insertado y mostrando el ícono de una cámara ya que aún no se ha elegido imagen alguna:

Figura 25 – *Image picker* con ícono de una cámara como placeholder

Tarea 4

La pantalla *Anuncio* con una imagen capturada con la cámara del smartphone se muestra a continuación:

Figura 26 – *Image picker* con una imagen capturada con la cámara

Tarea 5

La pantalla *Anuncio* con una imagen recuperada desde la galería del smartphone se muestra a continuación:

Figura 27 – *Image picker* con una imagen recuperada desde la galería

Tarea 6

Ahora en la pantalla *Anuncio* se puede visualizar además del widget *image picker* un campo de texto que se ubica debajo del primero. La distribución escogida para el ordenamiento de los widgets en esta pantalla es vertical. La siguiente imagen muestra el estado de la pantalla:

Figura 28 – *Image picker* junto con widget de inserción de texto

Tarea 7

Se ingresa texto a modo de reseña para probar la operación del widget, la imagen a continuación lo muestra:

Figura 29 – Probando el ingreso de texto

2.6.3.3 Sprint Review

Tabla 11 – Cumplimiento de los criterios de aceptación – Sprint 2

Código HU	Criterio de Aceptación	¿Cumple?
T02	Mostrar la pantalla <i>Anuncio</i> que contiene el formulario con los campos del anuncio.	Sí
	Permitir al usuario tomar una fotografía del producto con la cámara del smartphone.	Sí
	Permitir al usuario navegar por la galería de imágenes del smartphone y escoger una de ellas	Sí
T03	Capturar el texto ingresado por el usuario (reseña del anuncio) y validarlo	Sí

2.6.4 Sprint 3

Para este sprint se encuentra planificado se lleve a cabo la construcción de dos elementos de ingreso de texto y la pantalla que muestra la lista de promociones que se han ingresado:

- El primer elemento corresponde al campo para ingreso del precio de venta al público de los productos de la promoción.
- El segundo elemento corresponde al campo para ingreso del valor porcentual del descuento que se quiere aplicar a la promoción.
- El siguiente paso es la creación de la pantalla que muestra la lista que muestra a todas las promociones que se han ingresado. Junto con esta pantalla se incluye la operaciones de actualización de la promoción o eliminación de esta.

2.6.4.1 Sprint Planning

El presente sprint se ha dividido en siete tareas que han sido definidas de acuerdo con los objetivos planteados durante la elaboración de las historias de usuario que se corresponden con el sprint:

Tabla 12 – Tareas correspondientes al Sprint 3

Código HU	Nº	Tarea
T04	1	Incorporar un widget campo de texto, acompañado de la leyenda <i>Precio de venta al público (pvp)</i> y establecer sus validaciones.
	2	Incorporar un widget campo de texto, acompañado de la leyenda <i>Porcentaje de Descuento (%d)</i> y establecer sus validaciones.

	3	Mostrar al usuario solo el teclado numérico ya que los campos deben recibir solo valores numéricos.
T05	4	Crear el widget <i>Anuncio</i> con los datos ingresados en la pantalla <i>Anuncio</i> . Este widget además debe mostrar el <i>Precio con Descuento</i> (pcd, calculado al aplicar el %d al pvp).
	5	Crear la pantalla <i>Anuncios</i> , cuyos elementos son widgets <i>Anuncio</i> .
	6	Permitir que el usuario seleccione un anuncio de entre los que se encuentran en lista de la pantalla <i>Anuncios</i> y lo edite.
	7	Permitir que el usuario seleccione un anuncio de entre los que se encuentran en lista de la pantalla <i>Anuncios</i> y lo elimine.

2.6.4.2 Implementación

Las siete tareas han sido llevadas a cabo de manera exitosa. A continuación se presenta tarea por tarea la evidencia asociada:

Tarea 1

A continuación se presenta la imagen de la pantalla *Anuncio* modificada en la cual se puede observar que se ha adicionado un nuevo campo de texto con la leyenda *Precio de venta al público*. La distribución continúa siendo vertical:

Figura 30 – Campo de texto *Precio de venta al público*

Tarea 2

La siguiente imagen muestra la adición de otro campo de texto acompañado de la leyenda *Porcentaje de Descuento*. Este elemento se coloca debajo del anterior:

Figura 31 – Campo de texto *Porcentaje de Descuento*

Tarea 3

Al momento que el usuario se ubica sobre cualquiera de los dos widgets recientemente colocados, Android lanza el teclado numérico de tal manera que el usuario no tenga otra opción que ingresar valores numéricos. Las imágenes a continuación muestran el evento:

Figura 32 – Teclado numérico al ingresar al campo pvp o al campo %d

Tarea 4

La imagen a continuación presenta el widget *Anuncio*, el cual agrupa a todos los datos ingresados por el usuario en la pantalla *Anuncio* además de adherir el pcd que se calcula al aplicar el %d al pvp:

Figura 33 – Widget *Anuncio*

Tarea 5

Ya que el widget *Anuncio* está listo entonces se pasa a construir la pantalla *Anuncios*, la cual presenta una lista con todos los anuncios que los usuarios hayan ingresado, estos anuncios se empaquetan en widgets *Anuncio*. La imagen a continuación muestra la pantalla:

Figura 34 – Pantalla *Lista de Anuncios*

Tarea 6

Para editar un anuncio escogido de entre los elementos de la lista de la pantalla *Anuncios*, se programa sobre el gesto toque de cada elemento de la lista tal que la aplicación al detectarlo le lleva al usuario hasta la pantalla *Anuncio*, la cual se presenta con todos sus campos llenos a la espera de que el usuario haga las modificaciones del caso y luego mande a guardar los cambios. A continuación, se presenta un collage que corresponden al flujo descrito:

Figura 35 – Cambios realizados a un *Anuncio*

Tarea 7

Para eliminar un anuncio escogido de entre los elementos de la lista de la pantalla *Anuncios*, se programa sobre el gesto deslizamiento horizontal de derecha a izquierda de cada elemento de la lista tal que la aplicación al detectarlo elimina el anuncio elegido y este desaparece de la pantalla. A continuación, se presenta un collage que corresponden al flujo descrito:

Figura 36 – Eliminar un *Anuncio* de la pantalla *Anuncios*

2.6.4.3 Sprint Review

Tabla 13 – Cumplimiento de los criterios de aceptación – Sprint 3

Código HU	Criterio de Aceptación	¿Cumple?
T04	Mostrar el teclado numérico.	Sí
	Capturar el valor de <i>Precio de venta al público (pvp)</i> del producto ingresado por el usuario y validarlo.	Sí
	Capturar el <i>Porcentaje de Descuento (%d)</i> que se desea aplicar y validarlo.	Sí
T05	Calcular el <i>Precio con Descuento (pcd)</i> de la promoción al aplicar el %d al pvp.	Sí
	Mostrar la pantalla <i>Anuncios</i> la cual contine una lista de los anuncios creados.	Sí
	Permitir que el usuario seleccione un anuncio y lo actualice.	Sí
	Permitir que el usuario seleccione un anuncio y lo elimine.	Sí

2.6.5 Sprint 4

Para este sprint se encuentra planificado se lleve a cabo la construcción de la aplicación del cliente (El Pregonero Clientes). La aplicación está compuesta de tres pantallas, a saber:

- Pantalla *Promociones* que contine una lista vertical cuyos elementos son widgets *Anuncio*, cada uno de los cuales representa a un anuncio de la promoción que ha sido publicada por el establecimiento donde se encuentra la baliza.
- Pantalla *Promoción* que se muestra al usuario una vez este ha elegido algún elemento de la lista. Esta pantalla presenta un widget *Anuncio* que le presenta al cliente información detallada de la promoción. Este widget además le

proporciona al cliente un mecanismo que le permite al cliente marcar el anuncio como favorito.

- Pantalla *Favoritos* que contiene una lista con los anuncios de las promociones marcados como favoritos por el cliente. También proporciona el mecanismo para eliminar elementos de la lista.

Otra característica que la aplicación debe exhibir es la capacidad de mostrar notificaciones al usuario, que se lanzan como respuesta a la detección de la señal de la baliza. La notificación le advierte al cliente que en su entorno cercano hay al menos un establecimiento comercial que está anunciando sus promociones a través del uso de balizas BLE.

2.6.5.1 Sprint Planning

El presente sprint se ha dividido en siete tareas que han sido definidas de acuerdo con los objetivos planteados durante la elaboración de las historias de usuario que se corresponden con el sprint:

Tabla 14 – Tareas correspondientes al Sprint 4

Código HU	Nº	Tarea
T09	1	Crear la pantalla <i>Promociones</i> que presenta la lista de anuncios de las promociones publicadas por el establecimiento comercial para que el usuario pueda visualizarlas.
	2	Hacer que la aplicación reaccione a la presencia de la señal Bluetooth emitida por la baliza BLE, lanzando una notificación al hacia el centro de notificaciones de Android.
	3	Hacer que la aplicación reaccione al toque que el usuario le dé a la notificación, mostrando la pantalla de <i>Promociones</i> .
T10	4	Crear la pantalla <i>Promoción</i> que muestra con mayor detalle el anuncio de la promoción que el usuario ha escogido de entre los anuncios de la lista de la pantalla <i>Promociones</i> .
	5	Incorporar el widget y la lógica necesaria que le permita al usuario marcar y almacenar, si así este lo desea, en su lista de favoritos el anuncio de la promoción que se encuentra en pantalla.
	6	Crear la pantalla <i>Favoritos</i> donde se presenta una lista con todos los anuncios que el usuario ha marcado como favoritos .
	7	Permitir que el usuario seleccione un anuncio de entre los que se encuentran en lista de la pantalla <i>Favoritos</i> y lo elimine.

2.6.5.2 Implementación

Las siete tareas han sido llevadas a cabo de manera exitosa. A continuación se presenta tarea por tarea la evidencia asociada:

Tarea 1

En la imagen que se presenta a continuación se puede apreciar la pantalla *Promociones* con algunos anuncios de promociones de prueba que han sido publicados por el establecimiento:

Figura 37 – Pantalla *Promociones*

Tarea 2

Luego de implementar la lógica necesaria, el aplicativo es capaz de insertar notificaciones en el centro de notificaciones de Android que luego son lanzadas hacia la pantalla del usuario para advertirlo acerca de las promociones que se encuentran en su entorno. La siguiente imagen muestra el resultado:

Figura 38 – Notificación de la aplicación

Tarea 3

Inmediatamente después de que el usuario haya tocado la notificación, Android lanza la aplicación en la pantalla *Promociones* para que este pueda observar la lista de anuncios de promociones que el establecimiento comercial ha publicado. El siguiente collage muestra el flujo descrito:

Figura 39 – Reacción al toque sobre la notificación

Tarea 4

Con el fin de que el usuario pueda apreciar con mayor detalle el anuncio de la promoción que ha elegido en la pantalla *Promociones* se crea la pantalla *Promoción*. El siguiente collage muestra el flujo que trae como consecuencia la presentación de la pantalla *Promoción*:

Figura 40 – Pantalla *Promoción* y el flujo asociado

Tarea 5

La siguiente imagen muestra a la pantalla *Promoción* después de adherir el widget (y la lógica asociada) que le permite marcar a la promoción como favorita:

Figura 41 – Widget de marcado

Tarea 6

La pantalla *Favoritos* y algunos anuncios marcados como favoritos se muestran en la siguiente imagen:

Figura 42 – Pantalla *Favoritos*

Tarea 7

Para eliminar un anuncio escogido de entre los elementos de la lista de la pantalla *Favoritos*, se programa sobre el gesto deslizamiento horizontal de derecha a izquierda de cada elemento de la lista tal que la aplicación al detectarlo elimina el anuncio elegido y este desaparece de la pantalla. A continuación, se presenta un collage que corresponden al flujo descrito:

Figura 43 – Eliminar un *Anuncio* de la pantalla *Favoritos*

2.6.5.3 Sprint Review

Tabla 15 – Cumplimiento de los criterios de aceptación – Sprint 4

Código HU	Criterio de Aceptación	¿Cumple?
T09	Lanzar una notificación cuando se encuentre en la zona de cobertura de la baliza BLE.	Sí

	Reaccionar al toque del usuario sobre la notificación, mostrando la pantalla de <i>Promociones</i> .	Sí
	Mostrar la pantalla <i>Promociones</i> , la cual contiene la lista de los anuncios que el establecimiento ha publicado.	Sí
T10	Mostrar la pantalla <i>Promoción</i> con la información del anuncio que el usuario seleccionó en la pantalla <i>Promociones</i> .	Sí
	Permitirle al usuario marcar como favorito el anuncio de la promoción que se haya elegido.	Sí
	Mostrar la pantalla <i>Favoritos</i> que contiene la lista de los anuncios de promociones marcados como favoritos por el usuario.	Sí
	Permitirle al usuario eliminar anuncios de su lista de favoritos.	Sí

2.6.6 Sprint 5

Para este sprint se encuentra planificado el incremento de las características del aplicativo El Pregonero Comercios. Las nuevas características les darán a los usuarios la posibilidad de:

- Crear su usuario con sus respectivas credenciales (nombre de usuario y contraseña).
- Iniciar sesión en la aplicación tal que el usuario pueda hacer un uso controlado de la aplicación.
- Modificar la información del perfil del usuario.

Las nuevas pantallas que se crean son las siguientes:

- La pantalla *Inicio de Sesión* con un formulario para que el usuario ingrese sus credenciales (email y contraseña) y pueda autenticarse. Si el usuario aún no ha sido registrado, la pantalla dispone de un widget que le llevará al usuario a la pantalla de registro.
- La pantalla *Registro de Usuario Nuevo* que cuenta con un formulario para que el usuario pueda crear sus credenciales (email y contraseña).

2.6.6.1 Sprint Planning

El presente sprint se ha dividido en siete tareas que han sido definidas de acuerdo con los objetivos planteados durante la elaboración de las historias de usuario que se corresponden con el sprint:

Tabla 16 – Tareas correspondientes al Sprint 5

Código HU	Nº	Tarea
T06	1	Crear la pantalla <i>Registro de Usuario Nuevo</i> con el correspondiente formulario el cual se compone de los campos email (nombre de usuario) y contraseña y del botón <i>Regístrate</i> que confirma el registro.

	2	Implementar la lógica que lleva a cabo el proceso de registro del usuario con las credenciales dadas y que se ejecuta cuando el usuario acciona el botón <i>Regístrate</i> . Los campos de credencial se validan en este punto.
	3	Devolver al usuario a la pantalla <i>Inicio de Sesión</i> si el registro es exitoso caso contrario informarle al usuario del error ocurrido.
T07	4	Crear la pantalla <i>Inicio de Sesión</i> con el correspondiente formulario para que el usuario ingrese sus credenciales (email y contraseña) e inicie sesión al accionar el botón <i>Ingresar</i> o se pueda registrar al accionar el botón <i>Regístrate</i> .
	5	Implementar la lógica que lleva a cabo el proceso de inicio de sesión del usuario con las credenciales dadas y que se ejecuta cuando el usuario acciona el botón <i>Ingresar</i> . Los campos de credencial se validan en este punto.
	6	Implementar la navegación hacia la pantalla <i>Registro de Usuario Nuevo</i> desde la pantalla <i>Inicio de Sesión</i> cuando el usuario accione el botón <i>Regístrate</i> .
	7	Llevar al usuario a la pantalla <i>Anuncios</i> si el inicio de sesión ha sido exitoso caso contrario informarle al usuario del error ocurrido.
	5	En la pantalla de perfil, implementar la lógica asociada al cierre de sesión.

2.6.6.2 Implementación

Las siete tareas han sido llevadas a cabo de manera exitosa. A continuación se presenta tarea por tarea la evidencia asociada:

Tarea 1

La imagen a continuación muestra la pantalla *Registro de Usuario Nuevo*, la cual contiene un formulario con dos campos email (nombre de usuario) y contraseña además del botón *Regístrate*:

Figura 44 – Pantalla *Registro de Usuario Nevo*

Tarea 2

En la imagen a continuación se puede observar que los campos han sido llenados y procede a solicitar el registro al accionar el botón *Regístrate*:

Figura 45 – Solicitud de registro

Tarea 3

Después de llevar a cabo el registro del usuario y al ser este exitoso la aplicación devuelve al usuario a la pantalla *Inicio de Sesión*. El collage que se presenta a continuación muestra el flujo mencionado:

Figura 46 – Registro de nuevo usuario exitoso

Tarea 4

La imagen a continuación muestra la pantalla *Inicio de Sesión*, la cual contiene un formulario con dos campos email (nombre de usuario) y contraseña además de dos botones *Ingresar* y *Regístrate*:

Figura 47 – Pantalla *Inicio de Sesión*

Tarea 5

En la imagen a continuación se puede observar que los campos han sido llenados y procede a solicitar el inicio de sesión al accionar el botón *Ingresar*:

Figura 48 – Solicitud de inicio de sesión

Tarea 6

Ya que la pantalla *Inicio de Sesión* es la pantalla con la que arranca la aplicación entonces se requiere que desde esta se pueda navegar hasta la pantalla *Registro de Usuario Nuevo* en el caso de que el usuario aún no se encuentre registrado. El siguiente collage muestra el flujo hacia la pantalla *Registro de Usuario Nuevo*:

Figura 49 – Navegación hacia la pantalla *Registro de Usuario Nuevo*

Tarea 7

Una vez que el usuario ha iniciado sesión de forma exitosa, la aplicación lo lleva hasta la pantalla *Anuncios* donde se listan los anuncios que se encuentran publicados. El collage a continuación muestra el flujo:

Figura 50 – Navegación hacia la pantalla *Anuncios*

2.6.6.3 Sprint Review

Tabla 17 – Cumplimiento de los criterios de aceptación – Sprint 5

Código HU	Criterio de Aceptación	¿Cumple?
T06	Mostrar la pantalla <i>Registro de Usuario Nuevo</i> al usuario.	Sí
	Permitir que el usuario ingrese su nombre de usuario (email) y validarlo.	Sí
	Permitir que el usuario ingrese el texto que será su contraseña y validarlo.	Sí
	Permitir que el usuario solicite el registro.	Sí
	Devolver al usuario a la pantalla de <i>Inicio de Sesión</i> si el registro ha sido exitoso caso contrario informarle del error ocurrido.	Sí
T07	Mostrar la pantalla <i>Inicio de Sesión</i> al usuario.	Sí
	Permitir el ingreso de credenciales y solicitud de inicio de sesión por parte del usuario.	Sí
	Validar las credenciales del usuario.	Sí

	Permitir la navegación del usuario hasta la pantalla <i>Anuncios</i> siempre y cuando la validación del inicio de sesión haya sido exitosa caso contrario informarle del error ocurrido.	Sí
	Llevar al usuario desde la pantalla de <i>Inicio de Sesión</i> a la pantalla de <i>Registro de Usuario Nuevo</i> y mostrársela.	Sí

2.6.7 Sprint 6

Para el sprint final se tiene planificado la construcción de la última pantalla *Perfil y Configuraciones*. Los componentes de esta pantalla son:

- Un widget image picker para que el usuario pueda agregar una foto o imagen de galería a su perfil.
- Una etiqueta de texto en la cual se muestra el nombre de usuario (email).
- Un botón que al accionarlo le permite al usuario cerrar su sesión.

Hay que presentar en la pantalla principal la imagen de perfil.

2.6.7.1 Sprint Planning

El presente sprint se ha dividido en siete tareas que han sido definidas de acuerdo con los objetivos planteados durante la elaboración de las historias de usuario que se corresponden con el sprint:

Tabla 18 – Tareas correspondientes al Sprint 6

Código HU	Nº	Tarea
08	1	Crear la pantalla <i>Perfil y Configuraciones</i> .
	2	Incorporar el widget <i>image picker</i> para la foto de perfil, una etiqueta de texto para mostrar el nombre de usuario y un botón que cuando el usuario lo acciona cierra su sesión.
	3	Implementar la lógica asociada al cierre de sesión. Después de cerrar la sesión la aplicación le muestra al usuario la pantalla <i>Inicio de Sesión</i> .
	4	Crear el widget <i>Perfil y Configuraciones</i> .
	5	Incorporar el widget <i>Perfil y Configuraciones</i> a la pantalla principal.

2.6.7.2 Implementación

Las 5 tareas han sido llevadas a cabo de manera exitosa. A continuación se presenta tarea por tarea la evidencia asociada:

Tarea 1

La imagen siguiente muestra la pantalla *Perfil y Configuraciones* vacía:

Figura 51 – Pantalla *Perfil y Configuraciones* vacía

Tarea 2

A continuación, se muestra la imagen de la pantalla *Perfil y Configuraciones* con el widget image picker, la etiqueta de texto y el botón *Cerrar Sesión*:

Figura 52 – Pantalla *Perfil y Configuraciones* con sus elementos

Tarea 3

Una vez el usuario toque el botón de cierre de sesión la aplicación lo devuelva a la pantalla *Inicio de Sesión*. El collage que sigue muestra el flujo indicado:

Figura 53 – Cierre de sesión

Tarea 4

Se presenta la imagen del resultado del proceso de creación del widget *Perfil y Configuraciones*:

Figura 54 – Widget *Perfil y Configuraciones*

Tarea 5

Se incorpora el widget *Perfil y Configuraciones* a la pantalla principal tal que el usuario pueda acceder a su perfil. Este widget contiene básicamente la fotografía o imagen de perfil de usuario. La imagen a continuación muestra el resultado final:

Figura 55 – Pantalla principal con el widget *Perfil y Configuraciones*

2.6.7.3 Sprint Review

Tabla 19 – Cumplimiento de los criterios de aceptación – Sprint 6

Código HU	Criterio de Aceptación	¿Cumple?
T08	Mostrar al usuario la pantalla de <i>Perfil y Configuraciones</i> .	Sí
	Permitir al usuario tomar una fotografía de sí mismo (selfie) con la cámara del smartphone.	Sí
	Permitir al usuario navegar por la galería de imágenes del smartphone y escoger una de ellas.	Sí
	Permitir que el usuario pueda cerrar su sesión.	Sí

3. RESULTADOS Y DISCUSIÓN

Esta parte del trabajo está dedicada a la presentación y análisis de las pruebas tanto funcionales y no funcionales aplicadas a todo el sistema como las de usabilidad aplicadas a los dos aplicativos móviles.

Una vez que la fase de desarrollo ha concluido, el producto obtenido tiene que ser evaluado. Los ámbitos por valorar del sistema son:

- Funcionalidad, lo que involucra tanto a los requerimientos funcionales como a los no funcionales capturados a través de historias de usuario;
- Usabilidad, es decir cuán satisfactorio es para el usuario interactuar efectiva y eficientemente con la interfaz gráfica (Usability.gov).

3.1 Pruebas de Funcionalidad

Para garantizar que el sistema responde a las necesidades de los usuarios, su operación se debe validar contra los requerimientos funcionales y no funcionales puestos de manifiesto a través de historias de usuario, las mismas que fueron capturadas durante la fase de recolección y análisis de requerimientos. Las pruebas funcionales en general son del tipo caja negra ya que no se evalúa el código de forma particular sino que el sistema como un todo (Hamilton, 2022).

El número de usuarios involucrados en las pruebas es de 10, los cuales llevaron a cabo 2 casos de prueba. Los primeros 5 de ellos, desempeñando el rol de clientes del establecimiento comercial, probaron la aplicación El Pregonero Clientes (visualización de anuncios) mientras que los 5 participantes restantes, como encargados del establecimiento comercial, evaluaron la aplicación El Pregonero Comercios (ingreso de anuncios).

Los resultados de las pruebas se presentan de forma tabular en las siguientes apartados.

3.1.1 Caso de Prueba para Comercios (El Pregonero Comercios)

Las siguientes dos tablas muestran las fichas a aplicarse, la primera para el caso de prueba planteado mientras que la segunda contine los resultados que se esperan de la ejecución de la prueba. La última tabla de este apartado contiene los resultados de la aplicación del caso de prueba.

Tabla 20 – Ficha caso de prueba para comercios

Ingreso de un Anuncio de una Promoción	
Tarea	Anunciar la existencia de una promoción.
Rol	Encargado del establecimiento comercial.
Descripción	El usuario debe iniciar sesión en la aplicación y registrar un anuncio respecto de una promoción que se desea publicar.
Entradas	Nombre de usuario (email)
	Contraseña
	Imagen del producto
	Descripción

	Precio de venta al público	
	Porcentaje de descuento	
Actividades	Iniciar la aplicación.	
	Ingresar las credenciales.	
	Iniciar la sesión en la aplicación.	
	En la pantalla <i>Anuncios</i> darle un toque al botón circular etiquetado con el símbolo de suma.	
	En la pantalla <i>Anuncio</i>	Capturar una imagen del producto con la cámara del smartphone.
		Ingresar la descripción del anuncio de la promoción.
		Ingresar el valor numérico (con o sin decimales) del pvp del producto.
		Ingresar el valor numérico (hasta dos dígitos y sin decimales) del %d que se desea aplicar.
		Darle un toque al botón (etiquetado con un ícono visto) que se encuentra en la barra de la aplicación a la derecha para almacenar el anuncio.
	En la pantalla <i>Anuncios</i> darle un toque al anuncio para cambiar alguno de sus atributos.	
	En la pantalla <i>Anuncio</i>	Cambiar el pvp.
		Volver a guardar el anuncio.
Eliminar el anuncio al deslizarlo de derecha a izquierda.		
Cerrar la sesión.		

Tabla 21 – Ficha resultados esperados

Resultado Esperado	¿Se obtuvo el resultado?		
	Sí	No	En parte
Luego de iniciar sesión la aplicación le muestra al usuario la pantalla <i>Anuncios</i>			
Al tocar el botón el botón circular etiquetado con el símbolo de suma la aplicación le muestra al usuario la pantalla <i>Anuncio</i> .			

Al dar un toque el área de imagen de la pantalla <i>Anuncio</i> la aplicación lanza una pop-up con las opciones para de fuente de la imagen.			
Al seleccionar la opción relacionada con la cámara la aplicación abre la cámara.			
La fotografía tomada se presenta en el área que corresponde.			
El campo descripción puede albergar hasta 150 caracteres. Mientras se ingresa texto se observa que la cantidad restante disminuye.			
Al ingresar en los campos pvp y %d la aplicación muestra solo el teclado numérico.			
Al guardar el anuncio la aplicación muestra la pantalla <i>Anuncios</i> en cuya lista se puede observar el anuncio recién ingresado.			
Al dar un toque sobre el anuncio la aplicación muestra la pantalla <i>Anuncio</i> con el contenido del anuncio seleccionado.			
En la pantalla <i>Anuncios</i> al deslizar un anuncio de derecha a izquierda el anuncio seleccionado se elimina y desaparece de la pantalla.			

Tabla 22 – Resultados obtenidos después de ejecutado el caso de prueba

Resultado Esperado	¿Se obtuvo el resultado?		
	Sí	No	En parte
Luego de iniciar sesión la aplicación le muestra al usuario la pantalla <i>Anuncios</i>	5	0	0
Al tocar el botón el botón circular etiquetado con el símbolo de suma la aplicación le muestra al usuario la pantalla <i>Anuncio</i> .	5	0	0
Al dar un toque el área de imagen de la pantalla <i>Anuncio</i> la aplicación lanza una pop-up con las opciones para de fuente de la imagen.	5	0	0
Al seleccionar la opción relacionada con la cámara la aplicación abre la cámara.	5	0	0
La fotografía tomada se presenta en el área que corresponde.	5	0	0
El campo descripción puede albergar hasta 150 caracteres. Mientras se ingresa texto se observa que la cantidad restante disminuye.	5	0	0

Al ingresar en los campos pvp y %d la aplicación muestra solo el teclado numérico.	5	0	0
Al guardar el anuncio la aplicación muestra la pantalla <i>Anuncios</i> en cuya lista se puede observar el anuncio recién ingresado.	5	0	0
Al dar un toque sobre el anuncio la aplicación muestra la pantalla <i>Anuncio</i> con el contenido del anuncio seleccionado.	5	0	0
En la pantalla <i>Anuncios</i> al deslizar un anuncio de derecha a izquierda el anuncio seleccionado se elimina y desaparece de la pantalla.	5	0	0
Resultados	100%	0	0

Los resultado obtenidos indican que los cinco participantes pudieron llevar a cabo sin novedad alguna la lista de actividades descrita en la tabla 19 y por tanto este caso de prueba ha sido superado de manera satisfactoria.

3.1.2 Caso de Prueba para Clientes (El Pregonero Clientes)

Las siguientes dos tablas muestran las fichas a aplicarse, la primera para el caso de prueba planteado mientras que la segunda contine los resultados que se esperan de la ejecución de la prueba. La última tabla de este apartado contiene los resultados de la aplicación del caso de prueba.

Tabla 23 – Ficha caso de prueba para clientes

Visualización y Marcado de Anuncios	
Tarea	Observar los anuncios publicados por el comercio y seleccionar uno para marcarlo como favorito.
Rol	Cliente del establecimiento comercial.
Descripción	El usuario puede observar los anuncios publicados por el establecimiento comercial, elige y lo marca como favorito.
Entradas	La señal Bluetooth emitida por la baliza BLE
Actividades	Iniciar la aplicación.
	Ubicarse en las proximidades de la baliza BLE.
	Darle un toque a la notificación que aparece en pantalla.
	La pantalla <i>Promociones</i> aparece y muestra las promociones publicadas por el establecimiento comercial.
	Seleccionar una promoción de entre los elementos de la lista.
	En la pantalla <i>Promoción</i> que aparece verificar que el anuncio mostrado es el seleccionado.
	Dar un toque sobre el widget de marcado.

	Ir a la pantalla <i>Favoritos</i> al topar la pestaña correspondiente en fondo de la pantalla.
	Observar que el anuncio marcado como favorito se encuentre en la lista.
	Deslizar el anuncio de derecha a izquierda para eliminar el anuncio de lista de favoritos. El anuncio desaparece de la pantalla.

Tabla 24 – Ficha resultados esperados

Resultado Esperado	¿Se obtuvo el resultado?		
	Sí	No	En parte
La aplicación presenta una notificación al estar en presencia de la señal Bluetooth emitida por la baliza BLE.			
Al tocar la notificación la pantalla <i>Promociones</i> se muestra junto con la lista de las promociones que el establecimiento comercial ha publicado.			
Al dar un toque sobre un anuncio de la lista la aplicación presenta la pantalla <i>Promoción</i> con información detallada del anuncio de la promoción que se seleccionó.			
Al dar un toque sobre el widget de marcado este cambia para indicar que el anuncio ha sido almacenado en lista de favoritos.			
Al navegar hasta la pantalla <i>Favoritos</i> se observa la lista de anuncios marcados como favoritos.			
Al deslizar los elementos de la lista favoritos de derecha a izquierda, el elemento elegido se elimina y desaparece de la pantalla.			

Tabla 25 – Resultados obtenidos después de ejecutado el caso de prueba

Resultado Esperado	¿Se obtuvo el resultado?		
	Sí	No	En parte
La aplicación presenta una notificación al estar en presencia de la señal Bluetooth emitida por la baliza BLE.	5	0	0
Al tocar la notificación la pantalla <i>Promociones</i> se muestra junto con la lista de las promociones que el establecimiento comercial ha publicado.	5	0	0

Al dar un toque sobre un anuncio de la lista la aplicación presenta la pantalla <i>Promoción</i> con información detallada del anuncio de la promoción que se seleccionó.	5	0	0
Al dar un toque sobre el widget de marcado este cambia para indicar que el anuncio ha sido almacenado en lista de favoritos.	5	0	0
Al navegar hasta la pantalla <i>Favoritos</i> se observa la lista de anuncios marcados como favoritos.	5	0	0
Al deslizar los elementos de la lista favoritos de derecha a izquierda, el elemento elegido se elimina y desaparece de la pantalla.	5	0	0
Resultados	100%		

Los resultado obtenidos indican que los tres participantes pudieron llevar a cabo sin novedad alguna la lista de actividades descrita en la tabla 22 y por tanto este caso de prueba ha sido superado de manera satisfactoria.

3.2 Pruebas de Usabilidad

System Usability Scale (SUS) es una herramienta confiable para medir la usabilidad de las interfaces gráficas. SUS es un cuestionario cualitativo compuesto por 10 preguntas cada una de ellas con 5 opciones de respuesta que se encuentran en el rango de Totalmente en desacuerdo (1) hasta Totalmente de acuerdo (5). Fue creado por John Brooke en 1986 y es utilizado para evaluar productos de hardware, dispositivos móviles, por supuesto productos de software como sitios web y aplicaciones y todo aquello que posea una interfaz con la que un ser humano puede interactuar. Entre los beneficios que se conocen de esta herramienta está el hecho de que puede ser usado con muestra poblacionales pequeñas y aun así obtener resultados confiables (Usability.gov).

Para calcular el puntaje de usabilidad al aplicar el cuestionario SUS se tiene el siguiente algoritmo (Thomas):

1. Por cada pregunta impar, restar 1 al puntaje dado.
2. Por cada pregunta par, reste el valor dado de 5.
3. Sumar los nuevos 10 valores.
4. Multiplicar el resultado final por 2,5.

El resultado se debe interpretar de la siguiente manera (Thomas):

- El valor promedio de SUS es 68. Si el puntaje general se encuentra bajo esa medida entonces el producto bajo evaluación presenta serios problemas de usabilidad los cuales se deben resolver.
- Si el puntaje está alrededor de 68 hacia arriba entonces se puede decir que el producto tiene un buen nivel de usabilidad pero puede mejorar.
- Si el puntaje es igual o superior a 80,3 entonces el producto posee un buen nivel de usabilidad.

En la siguiente tabla se presenta las 10 preguntas del cuestionario SUS:

Tabla 26 – La 10 preguntas cuestionario SUS

Nº	Pregunta
1	Creo que me gustaría usar frecuentemente este sistema.
2	Encuentro el sistema innecesariamente complejo.
3	Me pareció que el sistema fue fácil de usar.
4	Pienso que necesitaría soporte de una persona técnica para ser capaz de usar el sistema.
5	Encontré que las diversas funciones del sistema estuvieron bien integradas.
6	Creo que hubo mucha inconsistencia en el sistema.
7	Imagino que mucha gente aprendería a usar el sistema muy rápidamente.
8	El sistema me pareció muy engorroso de utilizar.
9	Me sentí muy confiado al utilizar el sistema.
10	Necesité aprender muchas cosas antes de que pudiera ponerme a trabajar con el sistema.

Con base en los argumentos expuestos se ha elegido al cuestionario SUS como herramienta para evaluar la usabilidad de las interfaces gráficas de los dos aplicativos que forman parte del sistema de marketing de proximidad El Pregonero.

3.2.1 Evaluación de la usabilidad del aplicativo El Pregonero Comercios

Se les solicitó a los mismos 5 participantes que llevaron a cabo la evaluación funcional del aplicativo responder al cuestionario SUS. El resumen de resultados se presenta a continuación:

Tabla 27 – Puntajes obtenidos al aplicar el cuestionario SUS

	Puntajes				
	Evaluador 1	Evaluador 2	Evaluador 3	Evaluador 4	Evaluador 5
SUS 1	4	5	4	3	4
SUS 2	2	1	1	2	2
SUS 3	4	5	5	4	5
SUS 4	1	1	1	2	1
SUS 5	4	5	4	3	5
SUS 6	2	1	2	2	1
SUS 7	5	5	4	4	4
SUS 8	1	1	1	3	1

SUS 9	4	4	4	3	5
SUS 10	1	1	1	2	1

Tabla 28 – Puntajes normalizados y resultado total

	Puntajes Normalizados				
	Evaluador 1	Evaluador 2	Evaluador 3	Evaluador 4	Evaluador 5
SUS 1	3	4	3	2	3
SUS 2	3	4	4	3	3
SUS 3	3	4	4	3	4
SUS 4	4	4	4	3	4
SUS 5	3	4	3	2	4
SUS 6	3	4	3	3	4
SUS 7	4	4	3	3	3
SUS 8	4	4	4	2	4
SUS 9	3	3	3	2	4
SUS 10	4	4	4	3	4
Total	85	97,5	87,5	65	92,5

Una vez aplicado el algoritmo descrito en 3.2 se obtienen los puntajes finales por cada evaluador respecto de la usabilidad del aplicativo El Pregonero Comercios los cuales se observan al pie de la tabla 27.

Al calcular el promedio entre los 5 valores se obtiene el siguiente resultado: 85,5. Según lo indicado en 3.2 respecto de la interpretación de los resultados más el valor promedio indicado se puede concluir que el aplicativo, El Pregonero Comercios, posee un excelente nivel de usabilidad lo que significa que el aplicativo cumple con criterios de usabilidad y por lo tanto los usuarios del aplicativo se sentirán muy cómodos con el mismo.

3.2.2 Evaluación de la usabilidad del aplicativo El Pregonero Clientes

Para esta evaluación también se aprovechó del mismo grupo de 5 personas que llevaron a cabo las pruebas funcionales. A los evaluadores se les presentó y explico en consiste el cuestionario SUS y cuál es su utilidad. Los resultados de la aplicación del cuestionario se presentan a continuación:

Tabla 29 – Puntajes obtenidos al aplicar el cuestionario SUS

	Puntajes				
	Evaluador 1	Evaluador 2	Evaluador 3	Evaluador 4	Evaluador 5

SUS 1	5	5	5	5	5
SUS 2	1	1	1	1	1
SUS 3	5	4	4	5	5
SUS 4	1	1	1	1	1
SUS 5	4	5	5	4	5
SUS 6	2	1	1	1	1
SUS 7	5	5	5	5	5
SUS 8	1	1	1	1	1
SUS 9	5	4	4	5	5
SUS 10	1	1	1	1	1

Tabla 30 – Puntajes normalizados y resultado total

	Puntajes Normalizados				
	Evaluador 1	Evaluador 2	Evaluador 3	Evaluador 4	Evaluador 5
SUS 1	4	4	4	4	4
SUS 2	4	4	4	4	4
SUS 3	4	3	3	4	4
SUS 4	4	4	4	4	4
SUS 5	3	4	4	3	4
SUS 6	3	4	4	4	4
SUS 7	4	4	4	4	4
SUS 8	4	4	4	4	4
SUS 9	4	3	3	4	4
SUS 10	4	4	4	4	4
Total	95	95	95	97,5	100

Una vez aplicado el algoritmo descrito en 3.2 se obtienen los puntajes finales por cada evaluador respecto de la usabilidad del aplicativo El Pregonero Clientes los cuales se observan al pie de la tabla 29.

Al calcular el promedio entre los 5 valores se obtiene el siguiente resultado: 96,5. Según lo indicado en 3.2 respecto de la interpretación de los resultados más el valor promedio indicado se puede concluir que el aplicativo, El Pregonero Clientes, posee un excelente

nivel de usabilidad lo que significa que el aplicativo cumple con criterios de usabilidad y por lo tanto los usuarios del aplicativo se sentirán muy cómodos con el mismo..

3.3 Discusión

Respecto de las pruebas funcionales y los dos casos de uso que se propusieron, los resultados indican que los dos aplicativos móviles El Pregonero Comercios y El Pregonero Clientes, los cuales forman parte del sistema El Pregonero, cumplen con los requerimientos que se encuentran definidos en las historias de usuario que se encuentran en el apartado 2.3 y por lo tanto se da por terminado el ciclo de iteraciones que corresponden al proyecto.

En cuanto a la evaluación de la usabilidad cabe indicar que los valores altos (lo cual indica que los aplicativos cumplen con los criterios de usabilidad adecuados) que han recibido los aplicativos móviles se deben en gran medida a la baja complejidad y diseño minimalista que presentan sus interfaces gráficas. Y esto se debe a que solo se ha implementado un subconjunto de características (el núcleo del sistema) para obtener un Producto Mínimo Viable (MVP por sus siglas en inglés) (Agile Alliance) tal que se pueda evaluar su potencial como idea de negocio y se tenga un producto que pueda ser apreciado por los interesados en este tipo de sistema.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Una vez concluida la construcción del sistema queda claro que la arquitectura diseñada fue la correcta ya que permitió definir de forma clara a cada uno de los componentes y capas que lo conforman. Trabajar sobre cada una de las capas de forma independiente del resto concentró el esfuerzo de desarrollo del sistema en una parte a la vez, aplicando la ya bien conocida estrategia del divide y vencerás.
- Transformar un Raspberry Pi en una baliza BLE no fue complicado bastó con llevar a cabo una actualización de paquetes del sistema operativo e instalar el paquete BlueZ el cual permite la manipulación del dispositivo Bluetooth que viene integrado al Raspberry Pi.
- Dado que la herramienta elegida para construir el aplicativo fue Flutter cabe indicar que hubo inconvenientes para encontrar el paquete adecuado que permita manipular el dispositivo Bluetooth del smartphone. Lo más complicado fue hacer que el equipo reciba los mensajes de la baliza BLE sin necesidad de realizar un emparejamiento previo. Para solucionar el problema se tuvo que ejecutar de forma nativa código Java a través del mecanismo Platform Channels que forma parte del framework Flutter y que fue creado para ejecutar código que no es Dart.
- El uso del framework SCRUM en conjunto con la arquitectura diseñada fue de gran ayuda para construir el sistema de forma incremental y enfocada en los requerimientos que mayor valor le entregan al usuario.
- El haber hecho uso de Firebase, en particular de su servicio de base de datos NoSQL CloudStore, facilitó la construcción del sistema, esto debido a que no hubo la necesidad de desarrollar un backend desde cero.
- Una vez terminado el proceso de desarrollo y ya con el sistema listo, se pasó a evaluarlo.
 - De los resultados obtenidos de las pruebas de funcionalidad realizadas se concluye que todos los requerimientos realizados sobre el sistema han sido satisfechos tal cual se indica en el apartado 3.1 (Pruebas de Funcionalidad) del capítulo 3 (Resultados y Discusión) además de cumplirse con todos los criterios de aceptación impuestos por cada uno de los 6 sprints. Esto garantiza que el sistema ofrece la funcionalidad solicitada.
 - En cuanto a las pruebas de usabilidad cabe mencionar que los resultados indican que los aplicativos móviles (interfaces de usuario del sistema) cumplen con los criterios de usabilidad que SUS evalúa, lo que implica que las interfaces son bastante cómodas y facilitan la interacción del usuario con el sistema.

4.2 Recomendaciones

- Aprovechar todas características que ofrece la tecnología de las balizas BLE construyendo sistemas o aplicaciones que exploten alguna, varias o todas las características de esta tecnología como por ejemplo el posicionamiento al interior de edificios, almacenes, bodegas, etc.
- Se recomienda el uso del framework Flutter para el desarrollo de aplicaciones que posean una interfaz gráfica de usuario y que además se requiera tengan la capacidad de correr sobre distintas plataformas (multiplataforma). La experiencia

adquirida al haber trabajado con el framework (los dos aplicativos móviles) indica que con una única base de código se puede obtener varias versiones de una misma aplicación tal que se la pueda compilar y empaquetar para que trabaje en distintos entornos de ejecución (Android, web, Windows, etc.)

- No hacer uso del marco Eddystone para desarrollos futuros ya que ha dejado de tener soporte por parte de Google y de lo revisado tampoco existe una comunidad open source que lo haya acogido, así que el proyecto se encuentra a la deriva y por lo tanto cualquier desarrollo que confíe en Eddystone va a enfrentar serios problemas.
- El cuestionario SUS es una herramienta para evaluar la usabilidad muy potente pero al mismo tiempo muy fácil de comprender y operar. Sus resultados son muy confiables (Thomas). Por lo tanto su uso es altamente recomendado.
- Si se desea ejecutar código que no es Dart o llamar a las API's de la plataforma subyacente entonces hay que informarse acerca del mecanismo platform channels de Flutter. Este mecanismo permite el envío de mensajes a la plataforma solicitando la ejecución de alguna pieza de código, luego la plataforma devuelve el resultado a Flutter para que este continúe con su operación.

REFERENCIAS BIBLIOGRÁFICAS

- Agile Alliance. (s.f.). *What is a Minimum Viable Product (MVP)?* Obtenido de AgileAlliance.org: [https://www.agilealliance.org/glossary/mvp/#q=~\(infinite~false~filters~\(tags~\(~'mvp\)\)~searchTerm~'~sort~false~sortDirection~'asc~page~1\)](https://www.agilealliance.org/glossary/mvp/#q=~(infinite~false~filters~(tags~(~'mvp))~searchTerm~'~sort~false~sortDirection~'asc~page~1))
- Akpinar, E. (14 de Julio de 2021). *Bluetooth beacons: Everything you need to know*. Obtenido de Pointr: <https://www.pointr.tech/blog/beacons-everything-you-need-to-know>
- Aldrich, C. (15 de Diciembre de 2019). *Flutter: Code Organization (Revised)*. Obtenido de Medium: <https://medium.com/flutter-community/flutter-code-organization-revised-b09ad5cef7f6>
- Android Developers Blog. (25 de Octubre de 2018). *Discontinuing support for Android Nearby Notifications*. Obtenido de Android Developers Blog: <https://android-developers.googleblog.com/2018/10/discontinuing-support-for-android.html>
- Bhargava, M. (Agosto de 2018). *IoT Projects with Bluetooth Low Energy*. Birmingham, UK, United Kingdom: Packt Publishing Limited.
- bin Aftab, M. U. (Abril de 2017). *Building Bluetooth Low Energy Systems*. Birmingham, UK, United Kingdom: Packt Publishing Limited.
- Bluetooth SIG. (2020). *Developer Study Guide - An introduction to Bluetooth® Beacons – Android*. Bluetooth SIG.
- Bluetooth SIG. (2020). *Developer Study Guide - An introduction to Bluetooth® Beacons – Raspberry Pi*. Bluetooth SIG.
- Bluetooth SIG. (s.f.). *Bluetooth® Wireless Technology*. Obtenido de Bluetooth: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>
- BlueZ Project. (19 de Marzo de 2022). *Release of BlueZ 5.64*. Obtenido de BlueZ: <http://www.bluez.org/release-of-bluez-5-64/>
- Boelens, D. (6 de Agosto de 2018). *Widget — State — BuildContext — InheritedWidget*. Obtenido de Medium: <https://medium.com/flutter-community/widget-state-buildcontext-inheritedwidget-898d671b7956>
- Bravo, M. (2015). *Desarrollo del sistema de información para el seguimiento al registro nominal de usuarios de un programa nacional de asistencia alimentaria descentralizado*. Lima, Lima, Perú: Universidad Nacional de Ingeniería. Obtenido de <http://cybertesis.uni.edu.pe/handle/uni/5787>
- Dart Dev Team. (s.f.). *Asynchronous programming: futures, async, await*. Obtenido de Dart: <https://dart.dev/codelabs/async-await>
- Dart Dev Team. (s.f.). *Asynchronous programming: Streams*. Obtenido de Dart: <https://dart.dev/tutorials/language/streams>
- Dart Dev Team. (s.f.). *Dart Programming Language*. Obtenido de Dart: <https://dart.dev/>

Dart Dev Team. (s.f.). *Exception class*. Obtenido de Dart: <https://api.dart.dev/stable/2.7.0/dart-core/Exception-class.html>

dash-overflow.net. (s.f.). *provider 6.0.3*. Obtenido de pub.dev: <https://pub.dev/packages/provider>

devonfw. (20 de Noviembre de 2019). *Flutter Guide*. Obtenido de Github: <https://github.com/devonfw-forge/devon4flutter-non-bloc-arch/wiki>

Environmental Paper Network. (2018). *State Of The Global Paper Industry 2018*. (J. Martin, & M. Haggith, Edits.) Obtenido de Environmental Paper Network: https://environmentalpaper.org/wp-content/uploads/2018/04/StateOfTheGlobalPaperIndustry2018_FullReport-Final-1.pdf

Flutter Dev Team. (s.f.). *Build a form with validation*. Obtenido de Flutter: <https://docs.flutter.dev/cookbook/forms/validation>

Flutter Dev Team. (s.f.). *Build apps for any screen*. Obtenido de Flutter: <https://flutter.dev/>

Flutter Dev Team. (s.f.). *BuildContext class*. Obtenido de Flutter: <https://api.flutter.dev/flutter/widgets/BuildContext-class.html>

Flutter Dev Team. (s.f.). *InheritedWidget class*. Obtenido de Flutter: <https://api.flutter.dev/flutter/widgets/InheritedWidget-class.html>

Flutter Dev Team. (s.f.). *Introduction to widgets*. Obtenido de Flutter: <https://docs.flutter.dev/development/ui/widgets-intro>

Flutter Dev Team. (s.f.). *List of state management approaches*. Obtenido de Flutter: <https://docs.flutter.dev/development/data-and-backend/state-mgmt/options>

Flutter Dev Team. (s.f.). *Performance best practices*. Obtenido de Flutter: <https://docs.flutter.dev/perf/best-practices>

Flutter Dev Team. (s.f.). *State management*. Obtenido de Flutter: <https://docs.flutter.dev/development/data-and-backend/state-mgmt>

Flutter Dev Team. (s.f.). *StatefulWidget Class*. Obtenido de Flutter: <https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html>

Flutter Dev Team. (s.f.). *StatelessWidget Class*. Obtenido de Flutter: <https://api.flutter.dev/flutter/widgets/StatelessWidget-class.html>

Flutter Dev Team. (s.f.). *Writing custom platform-specific code*. Obtenido de Flutter: <https://docs.flutter.dev/development/platform-integration/platform-channels>

Gennari, F. (s.f.). *hciconfig(8) - Linux man page*. Obtenido de Linux man page: <https://linux.die.net/man/8/hciconfig>

Google LLC. (14 de Abril de 2016). *Eddystone Protocol Specification*. Obtenido de Github: <https://github.com/google/eddystone/blob/master/protocol-specification.md>

Google LLC. (12 de Noviembre de 2018). *How StatefulWidget Are Used Best - Flutter Widgets 101 Ep. 2*. Obtenido de Google Developers Official Youtube Channel: <https://www.youtube.com/watch?v=AqCMFXEmf3w>

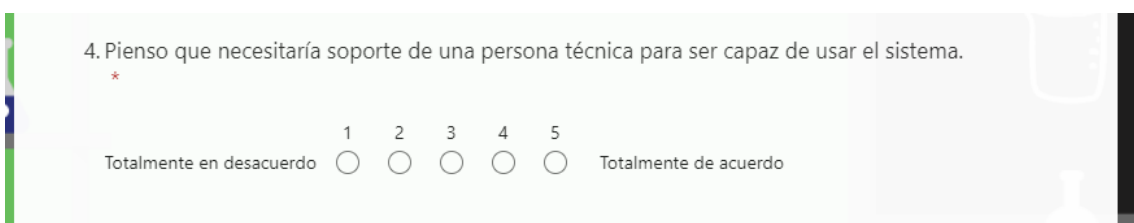
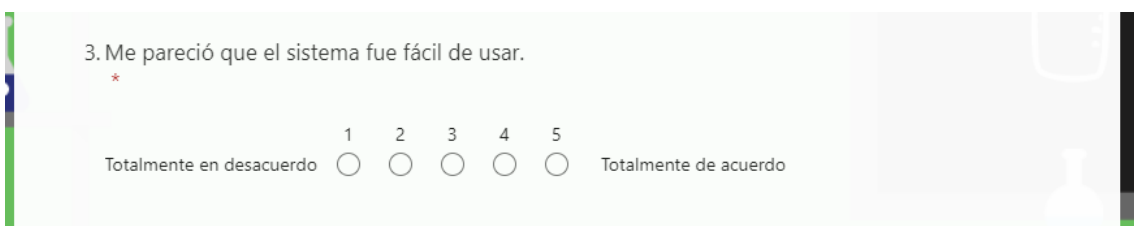
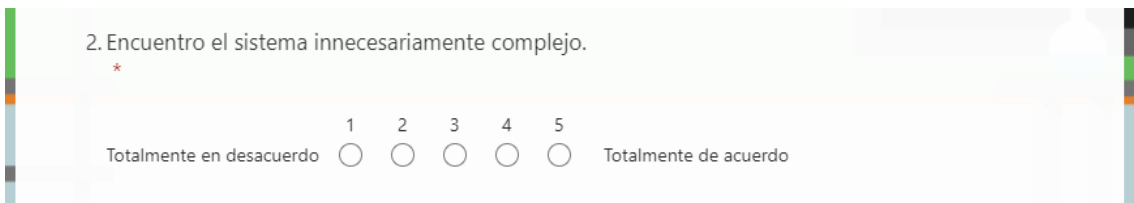
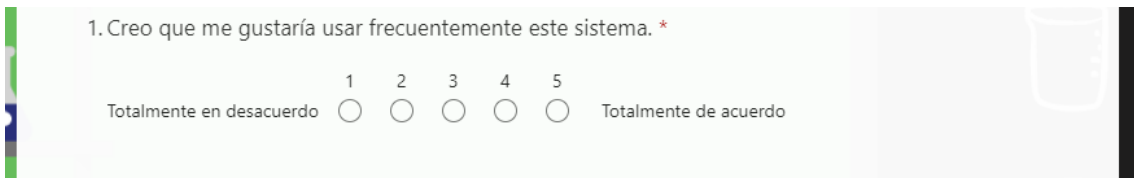
- Google LLC. (5 de Noviembre de 2018). *How to Create Stateless Widgets - Flutter Widgets 101 Ep. 1*. Obtenido de Google Developers Official Youtube Channel: <https://www.youtube.com/watch?v=wE7khGHVkyY>
- Google LLC. (17 de Septiembre de 2019). *How is Flutter different for app development*. Obtenido de Google Developers Official Youtube Channel: <https://www.youtube.com/watch?v=l-YO9CmaSUM>
- Google LLC. (3 de Julio de 2019). *Pragmatic State Management Using Provider (The Boring Flutter Development Show, Ep. 24)*. Obtenido de Flutter Dev Official Youtube Channel: <https://www.youtube.com/watch?v=HrBiNHEqSYU>
- Google LLC. (s.f.). *Android Developers*. Obtenido de Android: <https://developer.android.com/>
- Google LLC. (s.f.). *Firebase*. Obtenido de Firebase: <https://firebase.google.com/>
- Google LLC. (s.f.). *The Physical Web is an open approach to enable quick and seamless interactions with physical objects and locations*. Obtenido de The Physical Web: <https://google.github.io/physical-web/>
- Green America. (2017). *Paper Facts 2017*. Obtenido de Green America: <https://www.greenamerica.org/sites/default/files/inline-files/Paper%20Facts%202017.pdf>
- Hamilton, T. (21 de Junio de 2022). *What is Functional Testing? Types & Examples (Complete Tutorial)*. Obtenido de Guru99: <https://www.guru99.com/functional-testing.html>
- Jeon, K. E., She, J., Soonsawad, P., & Ng, P. C. (2018). BLE Beacons for Internet of Things Applications: Survey, Challenges and Opportunities. *IEEE INTERNET OF THINGS JOURNAL*, 811-828.
- Katz, M., Moore, K., Ngo, V., & Guzzi, V. (2021). *Flutter Apptentice*. Razeware LLC.
- Microsoft. (s.f.). *Code editing. Redifed*. Obtenido de Visual Studio Code: <https://code.visualstudio.com/>
- Navalkrushna, A., Balasaheb, N., & Swapnesh, P. (2016). Beacon for Proximity Target Marketing. *International Journal Of engineering and Computer Science*, 16359-16364.
- Papyrus Ltda. (13 de Junio de 2019). *Publicidad Impresa vs Digital*. Obtenido de Papyrus: <https://papyrusltda.com/publicidad-impresa-vs-digital/>
- ProductPlan. (s.f.). *User Story Examples in Product Development | Definition and Template*. Obtenido de ProductPlan.com: <https://www.productplan.com/glossary/user-story/>
- Raspberry Pi Foundation. (s.f.). *Teach, learn, and make with the Raspberry Pi Foundation*. Obtenido de Raspberry Pi Foundation: <https://www.raspberrypi.org/>
- Raspberry Pi Ltd. (s.f.). *Raspberry Pi Documentation*. Obtenido de Raspberry Pi: <https://www.raspberrypi.com/documentation/computers/os.html>
- Rehkopf, M. (s.f.). *User stories with examples and a template*. Obtenido de Atlassian.com: <https://www.atlassian.com/agile/project-management/user-stories>
- Rocha, Á., José Luis, R., Peter, M., & Bogdanović, Z. (2019). *Marketing and Smart Technologies*. Singapore: Springer.

- Savjoloys, V. (1 de Febreo de 2019). *Flutter app architecture 101: Vanilla, Scoped Model, BLoC*. Obtenido de Medium: <https://medium.com/flutter-community/flutter-app-architecture-101-vanilla-scoped-model-bloc-7eff7b2baf7e>
- Schwaber, K., & Sutherland, J. (Noviembre de 2020). *Scrum Guides*. Obtenido de Scrum Guide: <https://scrumguides.org/scrum-guide.html>
- Scrum.org. (s.f.). *What is a Product Backlog?* Obtenido de Scrum.org: <https://www.scrum.org/resources/what-is-a-product-backlog>
- Scrum.org. (s.f.). *What is Sprint Planning?* Obtenido de Scrum.org: <https://www.scrum.org/resources/what-is-sprint-planning>
- Thomas, N. (s.f.). *How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website*. Obtenido de Usabilitygeek.com: <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>
- Usability.gov. (s.f.). *Glossary*. Obtenido de Usability.gov: <https://www.usability.gov/what-and-why/glossary/u/index.html>
- Usability.gov. (s.f.). *System Usability Scale (SUS)*. Obtenido de Usability.gov: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>
- Zaim, D., Benomar, A., & Bellafkih, M. (2019). Developing A Geomarketing Solution. *Second International Conference on Intelligent Computing in Data Sciences* (págs. 353-360). Elsevier B.V.

ANEXOS

ANEXO 1

A continuación se presenta capturas de pantalla de las 10 preguntas de la encuesta SUS, suministrada a los evaluadores:



5. Encontré que las diversas funciones del sistema estuvieron bien integradas.

*

Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

6. Creo que hubo mucha inconsistencia en el sistema.

*

Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

7. Imagino que mucha gente aprendería a usar el sistema muy rápidamente.

*

Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo



8. El sistema me pareció muy engorroso de utilizar.

*

Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

9. Me sentí muy confiado al utilizar el sistema. *

Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

10. Necesité aprender muchas cosas antes de que pudiera ponerme a trabajar con el sistema. *

Totalmente en desacuerdo 1 2 3 4 5 Totalmente de acuerdo

