Especificación de Requerimientos de Software IEEE 29148

para

Banco de Ropa, Corporación Minuto de Dios

Versión 1.0

Propuesto por: Urquijo, J. Talero, A.

Pontificia Universidad Javeriana

8 de mayo de 2023

Índice general

1 Introducción

1.1. Propósito

Actualmente, dentro del programa del Banco de Ropa existen 16 roperos repartidos en diferentes ciudades de Colombia donde son distribuidas las prendas. En cada uno de estos roperos, existe personal encargado de realizar el registro de las ventas y atender a la población que desee compras las prendas. En algunos casos, el personal que atiende los roperos pueden ser voluntariados de personas que quieren apoyar con su trabajo al programa o personas contratadas para atender los establecimientos, ya que se requiere tiempo completo. De igual manera, se busca seguir abriendo más roperos para abarcar más población con escasos recursos y que estos puedan acceder a prendas en buen estado pagando poco por ellas.

El banco de ropa tiene limitaciones en cuanto al uso de herramientas tecnológicas para el manejo de su inventario de prendas. Actualmente, el ingreso de nuevas donaciones se realiza de forma manual, llenando formatos impresos según el tipo de prenda que se recibe, lo cual indica que el control de inventarios es deficiente. Además, aunque en algunos de los roperos se utilizan computadores o teléfonos móviles para registrar las ventas diarias, no todos tienen acceso a internet, se utiliza un formato en Excel para consolidar las ventas diarias, que luego se remite a la directora del banco para su validación.

Para el control de inventarios en los roperos, se emplea otro formato en Excel que se envía al banco de ropa para solicitar prendas específicas. Una vez recibida la solicitud, el ropero busca y prepara las prendas solicitadas, y registra la cantidad enviada en el mismo formato.

Es importante que el banco de ropa haga uso de herramientas tecnológicas que permitan automatizar el registro de las donaciones y ventas, así como el control de inventarios. Esto puede mejorar significativamente la eficiencia y precisión en el manejo de la información, lo cual puede traducirse en una mejor gestión del banco de ropa y en una mejor atención a sus usuarios.

1.1.1. Objetivo General

Diseñar e implementar un aplicativo que permita automatizar el control de inventarios dentro del banco de ropa del minuto de Dios

1.2. Alcance

Se busca desarrollar una aplicación móvil para el sistema operativo **Android (versión 8.1)** que permita a los colaboradores del Banco de Ropa del Minuto de Dios registrar información acerca del inventario de ropa, registrar ventas y crear bonos virtuales. La aplicación estará conectada a un sistema de persistencia de datos distribuido que permitirá consignar la información incluso cuando no haya conexión a internet.

1.3. Descripción del Producto

1.3.1. Perspectiva del Producto

Interfaces de Usuario

La aplicación será el componente de interfaz de usuario a través del cuál los colaboradores del Banco de Ropa del Minuto de Dios registren los ingresos de ropa y consoliden las ventas, la *figura 1.1* representa las diferentes interfaces de la aplicación.

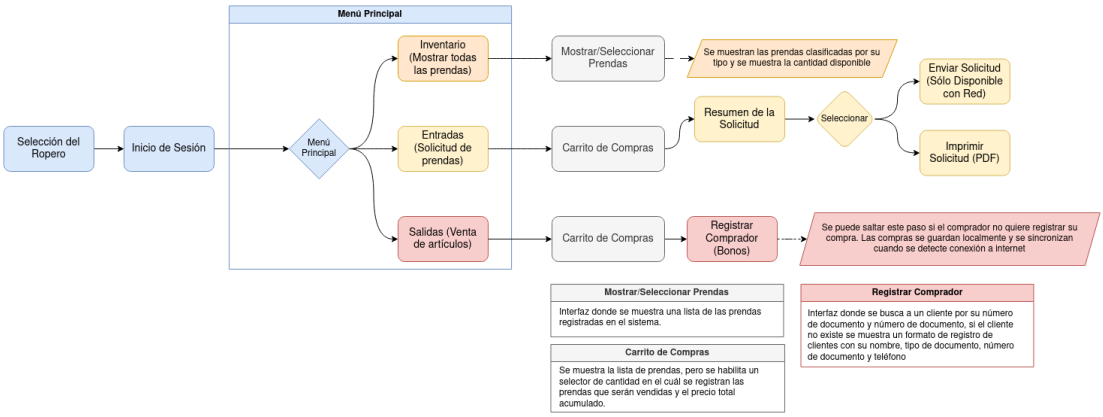

Figura 1.1: Flujo de la Interfaz de Usuario

Como es descrito en la *Sección 1.3.1*, la tecnología utilizada durante la construcción de las interfaces de usuario será **Jetpack Compose** [1], el cuál se basa en el paradigma de programación imperativa, lo que quiere decir que las interfaces se moldearán a partir de un conjunto de datos llamados *estado* y no al revés, por lo que proveer un esquema robusto de la interfaz en este momento no es posible

Interfaces de Hardware

Se contempla que la aplicación sea ejecutada en celulares inteligentes con interfaz táctil y una pantalla con dimensiones mayores a 720x1080 px con orientación vertical, el dispositivo debe además poseer la capacidad de conectarse a internet, todos los dispositivos

en el cuál se ejecuté la aplicación tendrán que conectarse a un servidor centralizado en donde se almacenará el consolidado de la información que luego podrá ser consultada por otros medios, se busca que este servidor haga parte de la infraestructura existente de la organización, sin embargo; debido a las actuales limitaciones se considerarán otras alternativas de infraestructuras alojadas en la nube.

Interfaces de Software

Para el desarrollo del sistema de persistencia de datos se requiere cumplir con el siguiente requerimiento: *"Se pueden alterar los contenidos de la base de datos incluso cuando no existe una conexión a internet, esta información se almacenará en caché hasta que la conexión con el sistema central sea posible y los cambios puedan ser consolidados"*, debido a los postulados del *Teorema de CAP* [2] una base de datos tolerante a particiones, disponible y consistente no es posible, por lo que tendremos que lidiar con problemas de inconsistencias en los datos entre las réplicas particionadas a falta de una conexión a internet. El método mediante el cuál pretendemos solucionar las inconsistencias es mediante consulta directa al usuario, y resolución automática cuando sea posible mediante el uso de bases de datos no estructuradas *noSQL*, existen múltiples bases de datos que proveen funcionalidades de este tipo como: CassandraDB o CouchDB, sin embargo; al no contar con la infraestructura necesaria será necesaria una base de datos alojada en la nube, las bases de datos mencionadas anteriormente podrán ser consideradas en una implementación futura que busque reemplazar la infraestructura en nube una vez sean adquiridos los recursos en la organización.

El software requerido para el correcto funcionamiento de la solución será el siguiente:

- Sistema Operativo **Android**, versión 8.1
- Infraestructura en nube con **Firebase** (*véase* 1.3.1 *Interfaces con Servicios*)

Para el desarrollo de estas funcionalidades se requieren las siguientes herramientas de software:

- SDK de desarrollo de Android, versión API 27
- Lenguaje de Programación Kotlin, versión 1.8
- Jetpack Compose [1], versión 1.4

Interfaces de Comunicación

Para lograr la comunicación entre el aplicativo y el servidor centralizado será necesario una conexión a internet, incluso si es intermitente, a través de Wi-Fi o redes móviles (GSM o VoIP).

Como la comunicación se dará a través de internet se utilizará el modelo de comunicaciones estándar 'TCP/IP' apoyado sobre el uso de protocolos de transporte TCP y de internet IPv4

La mayor parte de la comunicación será administrada por el SDK de Firebase para Android y por lo tanto no se detallará el modelo de comunicaciones del servidor centralizado pues es administrado por (Google Cloud Platform)

Interfaces con Servicios

Como ha sido anteriormente mencionado, actualmente no se cuenta con la infraestructura necesaria para alojar el servidor centralizado por lo que se harán uso de algunas herramientas de nube a través de la plataforma de Firebase (Google Cloud Platform), las herramientas necesarias hacen parte del plan gratuito "Spark" [3] y serán los siguientes:

- Cloud Firestore
- Authentication
- Cloud Messaging

Restricciones de Memoria

No se han contemplado estrictas restricciones de memoria sobre la aplicación, sin embargo; considerando que muchos de los colaboradores de los bancos de ropa poseen, en su mayoría, celulares de media o baja gama, se espera que la aplicación pueda ser ejecutada correctamente en estos dispositivos de recursos limitados (menos de 4GB de memoria principal "RAM" y un tamaño de la aplicación de alrededor 100MiB)

Para el servidor centralizado se deben cumplir con las restricciones del plan "Spark" de Firebase [1], por lo que la información almacenada en el servidor estará sujeta a la siguientes limitaciones:

- 1GB de almacenamiento para Cloud Firestore.

Requerimientos de adaptación del sitio

Será necesario que todos los operarios del ropero cuenten con un teléfono inteligente que cumpla todas las restricciones de software y de hardware descritas anteriormente mencionadas además de una conexión a internet, la disponibilidad de esta red debe ser de al menos una vez al día. El éxito del aplicativo estará en función de la capacidad de la organización de garantizar los requisitos anteriormente mencionados

1.3.2. Funciones del producto

Operaciones de usuario El usuario podrá realizar tres principales operaciones dentro de la aplicación

Listar prendas Mostrar todas las prendas registradas en el ropero categorizadas con sus respectivas cantidades

Registrar ingreso de nuevas prendas (Registrar solicitud) Una vez realizada una solicitud al banco de ropa, el ropero ingresará la información de las nuevas prendas que han llegado para agregarlas al inventario

Registrar salida de prendas (Realizar venta) Se registran las prendas que saldrán del ropero, se calculará automaticamente el precio y, si el cliente lo desea, se registrarán los bonos de compra en el sistema

Replicación y disponibilidad de la base de datos Las transacciones en la base de datos, en presencia de una conexión a internet, se sincronizan inmediatamente con la nube, sin embargo; la aplicación puede funcionar sin una conexión a internet, en este caso se almacenan los cambios en caché y una vez exista una conexión, los datos almacenados localmente se persisten a través de las múltiples réplicas en línea

1.3.3. Características de los Usuarios

Los usuarios de la aplicación serán los *colaboradores* en los roperos del **Banco de Ropa del Minuto de Dios**, estas personas, en su mayoría voluntarios, se encuentran en todos los rangos de edades y estratos socio-económicos, es por este motivo que describir a este usuario es una tarea compleja, debido a esto se asumirá que los usuarios son personas *moderadamente relacionadas con herramientas tecnológicas* capaces de hacer uso de las funciones básicas de un teléfono inteligente con sistema operativo *Android*

1.4. Definiciones

colaboradores Personal encargado de realizar el registro de las ventas y atender a la población que desee compras las prendas. En algunos casos, el personal que atiende los roperos puede ser voluntariados de personas que quieren apoyar con su trabajo al programa o personas contratadas para atender los establecimientos

documento En el contenxto de una *Base de datos documental*, un documento es un tipo de base de datos no relacional que ha sido diseñada para almacenar y consultar datos no estructurados como documentos de tipo JSON

2 Requerimientos

2.1. Funciones

2.1.1. Listar inventario

Permite al usuario ver el listado de todos los elementos almacenados en el sistema, se debe mostrar una lista perezosa *"Lazy List"* que progresivamente cargue los elementos en pantalla, adicionalmente se mostrará un cuadro de texto para realizar la búsqueda de prendas y se mostrarán filtros de: edad, género y tipo de prenda.

2.1.2. Registrar ingresos al inventario

Permite al usuario registrar la llegada de nuevas prendas, se mostrará una interfaz tipo *Carrito de Compras* en la cuál se mostrarán las prendas a añadir, para añadir una prenda se mostrará una nueva vista en la cual se deberá registrar el nombre de la prenda, su categoría, género, edad y cantidad. La fecha de ingreso se asigna automáticamente y su precio no es almacenado junto a la prenda, este precio se asigna en base a las características anteriores y se almacena en un documento diferente. Una vez agregada la prenda, la misma deberá ser mostrada en el carrito de compras, desde esta vista se podrá nuevamente agregar una nueva prenda o confirmar la inserción.

2.1.3. Registrar salidas al inventario (Realizar una venta)

Permite al usuario registrar una nueva venta en la cuál se registrarán las salidas de prendas del inventario, nuevamente se mostrará una ventana tipo carrito de compra 2.1.2, agregar una prenda en este caso abre una ventana de listado de inventarios 2.1.1 pero esta ventana contará además, con un componente tipo *Spinner* que permite seleccionar la cantidad. Una vez agregadas las prendas, se podrá confirmar la selección la cuál mostrará una nueva ventana con el subtotal calculado, se le preguntará al usuario si desea registrar el documento del usuario, de ser aceptada se muestra una ventana en donde se registra el documento del usuario para generar un bono virtual. Una vez concretada la venta se muestra un resumen y se da la posibilidad de imprimir el formato.

2.1.4. Control de Errores

Errores de Hardware

Durante la ejecución puede ocurrir un error de almacenamiento al no haber espacio suficiente para replicar la base de datos, en este caso, la aplicación no permitirá acceder al menú principal y se le notificará al usuario que debe liberar espacio de almacenamiento.

Errores de Comunicación

Durante la replicación puede haber un error de comunicación que afecte la integridad de los datos en la réplica, en caso de que exista este error, el *SDK de Firebase* garantizará la integridad de la base de datos local durante su sincronización, otras implementaciones deberán garantizar la *atomicidad en sus operaciones*. El error será notificado al usuario el cuál podrá reintentar la operación.

2.2. Requerimientos de la Base de Datos

Se requiere de una *base de datos documental* NOSQL como CassandraDB o CouchDB que facilite el control y automatización de réplicas, como primera implementación *Cloud Firestore* 1.3.1 será utilizado, la cuál será administrada por *Google Cloud Platform*

2.2.1. Modelo de Datos

Se tendrá un documento general con el listado de los precios de cada una de las categorías de las prendas, adicionalmente cada una de las ubicaciones del banco de ropa tendrá asociado un documento para su inventario. éste documento de inventarios consiste de un arreglo de prendas con su respectiva, categoría: tipo de prenda, género y edad

2.2.2. Conexión a Internet

El listado de prendas y precios debe ser replicado desde la nube al dispositivo de manera local para garantizar su disponibilidad aún sin internet, esta réplica se realizará en el primer acceso a la aplicación y se sincronizará con la nube siempre que al iniciar la aplicación se cuente con una conexión estable a internet, además se debe garantizar que la réplica no tenga una antigüedad mayor a 3 días, en tal caso no se permitirá el acceso a la aplicación sin internet.

2.3. Restricciones de Diseño

La arquitectura de la aplicación deberá desarrollarse bajo los estándares impuestos por la *"Guía de desarrollo y mejores prácticas para Android y Kotlin"* [4] la cuál guiará los lineamientos de desarrollo, además; se hará uso de la tecnología de *Jetpack Compose* [1] por lo que para facilitar el proceso de desarrollo se deberán usar los lineamientos de diseño de interfaces de *Material Design 3* [5]

3 Verificación

Para la verificación se realizarán dos comprobaciones con el usuario durante el ciclo de desarrollo además de una verificación final cercana a la fecha de entrega, además se validarán algunas de las funciones fundamentales del sistema a través de *Pruebas Unitarias*

4 Referencias

[1] Android Developers, "Jetpack Compose, el kit de herramientas para el desarrollo de IU de apps - Android Developers | Jetpack Compose." [Online]. Available: https://developer.android.com/jetpack/compose?hl=es-419

[2] "¿Qué es el teorema CAP? | IBM," Apr. 2023, [Online; accessed 6. Apr. 2023]. [Online]. Available: https://www.ibm.com/mx-es/topics/cap-theorem

[3] Firebase, "Firebase Pricing." [Online]. Available: https://firebase.google.com/pricing

[4] "Guide to app architecture," Mar. 2023, [Online; accessed 19. Apr. 2023]. [Online]. Available: https://developer.android.com/topic/architecture

[5] "Material Design," Dec. 1979, [Online; accessed 19. Apr. 2023]. [Online]. Available: https://m3.material.io