



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO DE UN SISTEMA PARA CONTROL DE INVENTARIO
Y LOGÍSTICA, PARA LA EMPRESA SOLDEINSA CON
APLICACIÓN ANDROID EN DISPOSITIVOS MÓVILES QUE
SE COMUNIQUEN A TRAVÉS DE REDES CELULARES Y A
LA RED WIRELESS DE LA EMPRESA.”**

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

LICENCIADA EN REDES Y SISTEMA OPERATIVOS

LILIANA DENISSE DÁVILA LEÓN

ANDREA CAROLINA HURTARES VALAREZO

GUAYAQUIL – ECUADOR

AÑO: 2018

AGRADECIMIENTO

Mi más sincero agradecimiento a mi familia que con esfuerzo nos han permitido llegar hasta aquí y poder cumplir una meta más en nuestros estudios profesionales.

De igual manera mi agradecimiento a la Escuela Superior Politécnica del litoral, por darme la oportunidad y abrirme las puertas para realizar mis estudios y poder a su vez hacer mi investigación dentro de su establecimiento educativo.

Finalmente quiero expresar mi sincero agradecimiento al Ing. Robert Andrade, quien fue el principal colaborador durante todo este proceso de realización de la tesis, quien, con su conocimiento, enseñanza permitió poder terminar con este trabajo de investigación.

Andrea Carolina Hurtares Valarezo

AGRADECIMIENTO

A Dios por permitirnos estar vivos aún, a nuestros padres y familiares por el gran sacrificio que han hecho por nuestra educación, a nuestros compañeros de estudio, a maestros y amigos, quienes sin su ayuda nunca hubiéramos podido hacer esta tesis. A todos ellos se lo agradecemos con mucho aprecio.

Liliana Denisse Dávila León

DEDICATORIA

El presente proyecto está dedicado a mi familia por haberme impulsado a seguir mis estudios superiores dentro de la ESPOLE ya que han sido mi mayor apoyo a lo largo de toda mi carrera universitaria y a lo largo de mi vida también a todas las personas especiales que me acompañaron en esta etapa de mi vida, aportando a mi formación tanto profesional y como ser humano.

Andrea Carolina Hurtares Valarezo

DEDICATORIA

Dios por ser parte de mí vida en todo momento, mi abuela Plaucia Garcés B, aunque ya no está entre nosotros le agradezco por el impulso a avanzar cada día, un ejemplo digno a seguir, del que sé hubiera sido tan especial esta etapa para ella como lo es para mí. Mi madre Glenda León G, por su amor incondicional, esfuerzo, sabiduría, entrega, constancia, ejemplo y apoyo sin importar nuestras diferencias de opiniones, gracias por todo. Mi tío Guido Vargas, por ser cómplices de vida. A Jennifer Retto por sus consejos, siempre estar dispuesta a escucharme y ayudarme en todo momento, mis amigas Shirley Espinoza y Gisella Bonilla, porque sin el equipo que formamos no hubiera logrado esta meta.

Liliana Denisse Dávila León

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Andrea Carolina
Hurtares Valarezo

Liliana Denisse
Dávila León

RESUMEN

El siguiente trabajo de titulación se elaboró con el propósito de optimizar uno de los procesos más importantes de cualquier empresa como es el control de inventario.

El desarrollo de la propuesta se basó en la investigación a la empresa Soldeinsa con la metodología de Design Thinking. Se buscó información necesaria de la empresa y de cada uno de los empleados con el fin de encontrar la problemática de esta. Se obtuvo como resultado, que un porcentaje mayoritario tenía inconvenientes con el área de compras y bodega, debido a la demora de materiales y el control obsoleto que tenían en forma manual, las obras no finalizaban en el tiempo estimado.

Usando herramientas tecnológicas como Android y Firebase, se investigaron posibles propuestas como solución al área afectada de la empresa, luego de varios prototipos y de una búsqueda exhaustiva se empezó a establecer el alcance del proyecto, lo que ayudó a recopilar todos los requerimientos del usuario. Se realizaron varios bocetos necesarios y que sean del agrado del mismo, para continuar con la programación que se ejecutó iterativamente hasta implementar todas las funcionalidades, se llevaron a cabo las últimas acciones para poder asegurar que el sistema funciona usando de modo que el personal técnico de campo y de bodegas vería en tiempo real los materiales con los cuales contará, haciéndole saber a las áreas respectivas el material faltante, al fin así evitará demoras en la terminación de la obra.

ABSTRACT

Inventory management is a challenging problem area in supply chain management. Today, companies need to have inventories in warehouses in order to execute customer demand and avoid loss of time and money.

This paper present a case of study for the company Soldeinsa whose purchasing and warehousing department can't pack off the purchase orders due to poor control of their inventory. The research also found that company Soldeinsa have a few inventory problems such as unorganized inventory arrangement, no cycle counting and no accurate records balance due to lack of workers

The purpose of this study is to design an inventory and logistic control system base on a mobile application, using the Android and Firebase mobile platforms.

This app will let to purchasing and warehousing department's workers know and management the exact inventory in real time. Therefore, they will be able to communicate to other departments about any problems with their raw materials or finished godos. This new inventory control, will help to the company to finish their projects on time.

ÍNDICE GENERAL

RESUMEN.....	I
ABSTRACT.....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS	V
SIMBOLOGÍA	VI
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS.....	VIII
CAPÍTULO 1.....	1
1.1 Introducción	1
1.2 Descripción del problema	2
1.3 Objetivos.....	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	4
1.3.3 Justificación.....	4
1.4 Marco Teórico.....	5
CAPÍTULO 2.....	24
2. METODOLOGÍA	24
2.1 Fase 1: Empatizar.....	24
2.2 Fase 2: Definir	25
2.3 Fase 3: Idear.....	26
2.4 Fase 4: Prototipar	27
2.5 Fase 5: Evaluar.....	27
CAPÍTULO 3.....	28
3. DESARROLLO DEL SISTEMA DE INVENTARIO SOLDEINSA.....	28

3.1	Especificaciones	28
3.2	Creación y configuración del aplicativo móvil en Android Studio	29
3.3	Conexión de la base de datos Firebase al programa Android Studio	30
3.4	Creación de los Usuarios en la base de datos Firebase.....	32
3.5	Funcionalidades del aplicativo móvil según los cargos de los usuarios.....	33
3.5.1	Funcionalidades del aplicativo móvil	35
CAPÍTULO 4.....		42
4.	Implementación y presupuesto	42
4.1	Plan de Implementación	42
4.2	Presupuesto.....	43
4.2.1	Cuadro de costo del proyecto	43
CONCLUSIONES Y RECOMENDACIONES		45
BIBLIOGRAFÍA.....		47
Anexos.....		47

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
Cia. Ltda	Compañía limitada
XML	Extensible Markup Language
PHP	Hypertext Pre-Processor
JSON	JavaScript Object Notation
Wifi	Wireless Fidelity
LICRED	Licenciatura en redes y sistemas operativos

SIMBOLOGÍA

ÍNDICE DE FIGURAS

Figura 1.1 Organigrama del número de trabajadores por cargo.	1
Figura 1.2 Diagrama de flujo del proceso de atención de las llamadas dentro de la empresa Soldeinsa.	¡Error! Marcador no definido.
Figura 3.1 Diagrama del diseño de la solución	¡Error! Marcador no definido.
Figura 3.2 Archivos .xml de las diferentes interfaces en Android Studio	29
Figura 3.3 Archivos de configuración .java en Android Studio.	30
Figura 3.4 Configuración de Firebase para que se conecte al aplicativo móvil	31
Figura 3.5 Conectividad del aplicativo móvil en Android Studio.....	31
Figura 3.6 Registro de usuarios en Firebase.	32
Figura 3.7 Interfaz de ingreso de usuario al aplicativo móvil.	33
Figura 3.8 Menú principal de cada uno de los roles dentro del aplicativo móvil.	34
Figura 3.9 Opción de ver inventario dentro del aplicativo móvil	35
Figura 3.10 Opción de agregar inventario dentro del aplicativo móvil.....	36
Figura 3.11 Ingreso de un producto a la base de datos.....	36
Figura 3.12 Opción de actualizar un producto en el aplicativo móvil	37
Figura 3.13 Modificación de un producto en la base de datos.....	37
Figura 3.14 Opción de ver órdenes de trabajo en el aplicativo móvil.....	38
Figura 3.15 Opción de actualizar órdenes de trabajo en el aplicativo móvil.	38
Figura 3.16 Modificación de una orden en la base de datos.....	39
Figura 3.17 Opción de rotación de inventario en el aplicativo móvil	39
Figura 3.18 Opción de aumento de órdenes en el aplicativo móvil.....	40
Figura 3.19 Creación de una orden en la base de datos	40
Figura 3.20 Opción de ayuda en el aplicativo móvil.....	41
Figura 4.1 Diagrama de Gantt del plan de implementación del proyecto.....	43

ÍNDICE DE TABLAS

Tabla 1.1 Cuadro comparativo de los proyectos recibidos y realizados.	2
Tabla 3.1 Usuario y contraseña para cada departamento dentro de la aplicación móvil.	32
Tabla 3.2 Funcionalidades del aplicativo móvil según su rol.	33
Tabla 4.1 Costo del proyecto	43

CAPÍTULO 1

1. INTRODUCCIÓN

SOLDEINSA CIA. LTDA. es una compañía dedicada al diseño, provisión e instalación de proyectos de ingeniería mecánica, para la importación de maquinaria industrial en el Ecuador.

La compañía actualmente tiene problemas financieros por lo que ha tenido que hacer reducción de personal, esto ha ocasionado que los empleados que estaban a cargo de un solo departamento deban encargarse de varias funciones. Por ejemplo, la persona encargada del área de compras también tiene delegado el manejo del inventario del área de bodega, además la compañía no utiliza un programa especializado para esta área sino uno básico como Excel para generar hojas de cálculos lo que ha ocasionado pérdidas de tiempo y materiales.

La compañía dispone actualmente con un total de 44 trabajadores de los cuales 14 trabajan a tiempo completo y 30 a medio tiempo distribuidos en todas las áreas de la empresa, que están altamente especializados en las siguientes áreas de: ingeniería comercial, ingeniería mecánica, técnico mecánico, contador. A continuación, se presenta una tabla de la distribución de los diferentes departamentos dentro de la compañía Soldeinsa (Figura1.1).

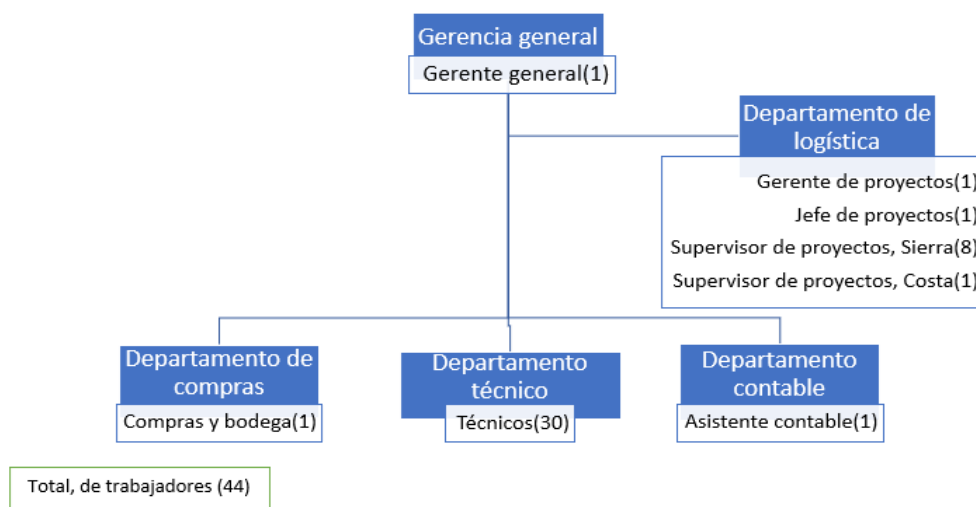


Figura 1.1 Organigrama del número de trabajadores por cargo

La compañía recibe pedidos de trabajos de los cuales al final del mes son aproximadamente entre 40 a 80 trabajos a realizar, después de esto se procede con la realización de estos que al final del mes son aproximadamente entre 40 a 60 trabajos realizados, lo cual permite ver que al final del mes hay un déficit en cuanto a cumplimientos de trabajos, ya que se recolecta más pedidos de los que se pueden llevar a cabo. (Tabla1.1)

Proyectos	Diarios	Semanales	Mensuales
Pedidos recibidos	2 - 4	10 – 20	40 – 80
Trabajos realizados	2 - 3	10 - 15	40 - 60

Tabla 1.1 Cuadro comparativo de los proyectos recibidos y realizados.

1.1 Descripción del problema

En el departamento de compras-bodega se observó que este tiene una alta demanda de requerimientos los cuales son producidos especialmente por el departamento técnico. Además, dicho departamento no puede cumplir con todos los requerimientos ya que registra de forma manual los ingresos y egreso de los materiales, lo que imposibilita llevar un control diario del mismo, debido a esto no hay un abastecimiento oportuno de las bodegas y además causa que la compañía deba comprar productos dentro y fuera del país a un alto costo. Estos atrasos afectan de gran manera a los trabajos de campo, debido a la falta o demora de materiales que se les debe distribuir, lo que finalmente ocasiona pérdida de tiempo y malestar en los empleados que deben realizar la obra en el plazo establecido.

En síntesis, los problemas que sufre la empresa Soldeinsa son los siguientes: pérdida de tiempo, materiales y dinero, todo esto es producido por no tener un debido sistema de control sobre su bodega.

A continuación, se presenta un diagrama de flujo del proceso de atención de llamadas dentro de la empresa Soldeinsa, donde se podrá observar las diferentes actividades que conforman al mismo desde que se recolecta la llamada del cliente hasta la atención de este (figura 1.2):

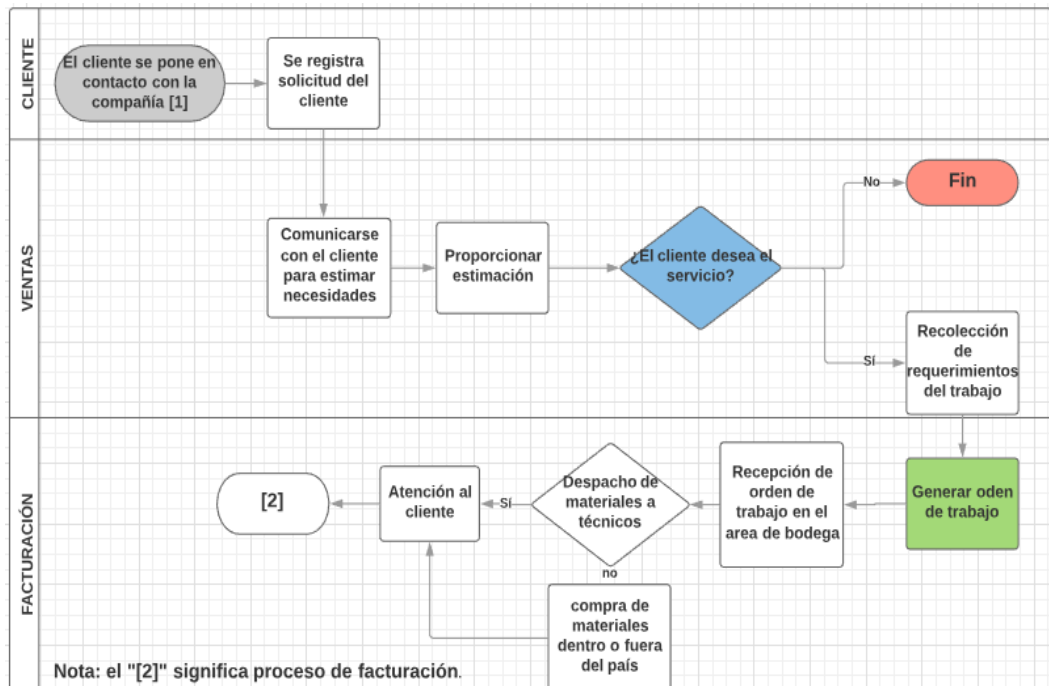


Figura 1.2 Diagrama de flujo del proceso de atención de las llamadas dentro de la empresa Soldeinsa.

1.2 Objetivos

1.2.1 Objetivo General

Generar un sistema que ayude a mejorar los procesos de atención, lo cual permitirá al departamento técnico realizar de manera óptima y eficiente sus actividades en la instalación o reparación de los trabajos a los clientes.

1.2.2 Objetivos Específicos

- Diseñar un sistema de inventario, tomando en consideración las necesidades y recursos de los empleados de Soldeinsa.
- Disminuir tiempos de atención para mejorar la satisfacción del cliente y empleados.
- Mejorar el proceso de recolección de los datos referentes al ingreso y egreso de los materiales por parte de los trabajadores.
- Mostrar datos íntegros y actualizados para mejorar el proceso de abastecimiento oportuno de materiales.

1.3 Justificación

Soldeinsa ha tenido un descontrol en la parte operativa principalmente en los departamentos que están encargados de la compra y venta de productos, esto ocasiona inventarios ineficientes que carece de planificación y que afectan la correcta gestión de las compras.

Soldeinsa al minimizar los tiempos de procesos de atención a los clientes y trabajadores, alcanzará los beneficios esperados en los objetos específicos, ya que empleará un sistema que controle el stock de bodega, para así mejorar la gestión en la compra de material faltante.

Por lo antes mencionado es importante la inclusión de un sistema que controle el inventario para que ayude a la empresa a minimizar los tiempos de entrega y a alcanzar los beneficios esperados, empleando técnicas correctas que controlen el stock de bodega, mejore la gestión de compra de repuestos y de material faltante.

Con esto se pretende tener una mejor organización que satisfaga al cliente dando soluciones rápidas a sus necesidades, también el gerente de la empresa

podrá estar al tanto del stock en bodega pudiendo ser manejado fácilmente desde un dispositivo móvil (celular, Tablet, entre otros.) de manera práctica, ágil y segura, sin pérdidas de información.

1.4 Marco Teórico

Dentro de la investigación, se encontraron usos de diferentes sistemas para el control de inventario relacionados con el objetivo del estudio.

Se citan a continuación investigadores que han contribuido para generar antecedentes sobre el tema de la propuesta del sistema de control de inventario para así mejorar la gestión del departamento de compras.

[1] Valverde Landívar Galo Robayo (febrero, 2018) “Sistema de Control de Inventarios Mediante el Uso de Tecnologías de Internet y Comunicación Inalámbrica”.

Este proyecto desarrolla una herramienta para automatizar el proceso de ingreso y verificación, llamado Sistema de Control de Inventarios, haciendo uso de tecnologías inalámbricas y asistentes digitales personales, mejorando la productividad del personal encargado de bodega, facilitando las tareas que realiza el personal administrativo y así ayudar a que el ingreso de inventarios se haga de manera más rápida y al menor costo posible [1].

[2] Villamil, 2015 “La implementación del control interno de inventarios para el sector droguista de acuerdo con las normas internacionales de auditoría – nías”.

Está encaminado a optimizar y garantizar el adecuado manejo de los inventarios, basado principalmente en el fortalecimiento de políticas y procedimientos administrativos y controles internos contables, el inventario de los productos resulta ser una actividad esencial en el desarrollo de las operaciones teniendo un adecuado almacenamiento, control de existencias físicas, anotaciones de entrada y salidas, conciliaciones con registros, conteos físicos y control en las fechas de vencimiento entre otros[2].

CAPÍTULO 2

2. METODOLOGÍA

En el proceso de observar la problemática y plantear una solución a la empresa Soldeinsa utilizamos la metodología de Design Thinking, esto permitió evaluar de una forma más eficiente la información proporcionada por la misma, a fin de cumplir con los objetivos y resultados esperados.

Esta metodología consta de 5 fases de las cuales son: Empatizar, definir, idear, prototipar y evaluar. Así mismo se pretende explicar brevemente cada una de las fases con sus respectivas herramientas y técnicas utilizadas que sirvieron durante el proceso de desarrollo del proyecto.

2.1 Fase 1: Empatizar

En esta fase se recopiló información de la empresa Soldeinsa tanto de forma interna (empleados) como de forma externa (página web), donde se pudo descubrir de forma eficaz tanto las fortalezas como debilidades de esta. A continuación, se presentará las herramientas utilizadas y proporcionadas por esta fase de las cuales se nombrará las más sobresalientes y que sirvieron como ayuda:

- 1) **Desk research:** Con esta metodología de investigación se pudo recopilar información de la empresa Soldeinsa ([Anexo 1](#)).
- 2) **Preparación de la entrevista:** Se generó preguntas eficaces en base al cargo que ocupaba el empleado de Soldeinsa, a fin de conocer más al fondo los procesos que lleva la empresa y sus empleados ([Anexo 2](#)).
- 3) **Entrevista:** En este paso se realizó el debido acercamiento con el personal de Soldeinsa, con el objetivo saber que es lo que pensaban cada uno de sus empleados de su lugar de trabajo, en cuanto a los problemas que encontraban en el mismo y como ellos creían que podrían resolverlo ([Anexo 3](#)).
- 4) **Análisis DAFO:** Se pudo conocer las fortalezas, oportunidades, debilidades y amenazas dentro de la compañía Soldeinsa, con el

propósito de saber qué es lo que dispone y puede llegar a ser la compañía con nuestra posible solución ([Anexo 4](#)).

5) Flujo de proceso: Se pudo identificar al departamento con mayor afluencia de requerimientos como el de bodega, donde esto permitió centrar las posibles soluciones en plano a dicho departamento ([Anexo 5](#)).

6) ¿Qué? ¿Cómo? ¿Por qué? (Problemática de la empresa): En este paso se plantearon preguntas y respuesta en cuando a: ¿qué problemas tenía?, ¿cómo ocurrían? y ¿por qué ocurrían? Esto permitió ver donde había más problemas y a dónde estaban ocurriendo ([Anexo 6](#)).

2.2 Fase 2: Definir

Después de haber recopilado la información de la fase 1 se identificó las principales falencias de la compañía Soldeinsa, donde por medio de las entrevistas que se realizaron a Freddy Mauricio Paucar Quintuña, Darwin Patricio Quinche, Jaime Gustavo Cofre Unapanta, se concluyó que las principales falencias de Soldeinsa son los siguientes: pérdidas de materiales, tiempo y abastecimiento inoportuno. A continuación, se mencionará las principales herramientas que se utilizaron en esta fase y permitieron descubrir esto:

1) Fichas insights: Estas fichas mostraron porque ocurren cada una de las principales necesidades de la empresa Soldeinsa ([Anexo 7](#)).

2) Point of view: Se estableció la problemática de cada departamento a fin de saber porque se debe resolver cada una de ellas ([Anexo 8](#)).

3) Mapa de afinidades: Es una herramienta que ayudó a clasificar y agrupar ideas con respecto a lo que contaba y debería contar la compañía Soldeinsa ([Anexo 9](#)).

4) Redefinición del problema: Técnica innovadora que se analizan desde un punto de vista alternativo ([Anexo 10](#)).

2.3 Fase 3: Idear

En esta fase se ideó y pensó en las posibles soluciones a los problemas que se identificaron en la fase anterior, donde por medio de análisis se pudo identificar y escoger las 2 propuestas de soluciones óptimas que son las siguientes: rediseño de la estructura organizacional y la creación de un sistema de inventario a través de un aplicativo móvil. A continuación, se mencionará las principales herramientas que se utilizaron para realizar este proceso:

- 1) Brainstoming:** También denominada tormenta de ideas, el fin de esta herramienta es facilitar el surgimiento de nuevas ideas sobre el tema propuesto para Soldeinsa ([Anexo11](#)).
- 2) Matriz importancia – dificultad:** Se mapearon en el eje vertical los proyectos o ideas que actualmente se están ofreciendo a la compañía Soldeinsa. En el eje horizontal los usuarios o clientes existentes y los nuevos de la misma manera ([Anexo 12](#)).
- 3) Matriz de decisiones:** En la matriz de decisiones se enlistó las principales problemáticas de las cuales destacan: pérdidas de materiales y tiempo, falta de personal, mejor manejo operativo de la bodega, rotación adecuada del inventario entre otros. Además, se observó que de las 6 posibles soluciones que se plantearon 2 resolvían en gran manera las problemáticas de la compañía ([Anexo 13](#)).
- 4) Selección de prototipos:** En base a los resultados obtenidos previamente en la matriz de decisiones, se escogió 2 de las 6 soluciones planteadas, ya que estas resolvían en gran manera las problemáticas, una de ellas resolvía 11 de 12 problemas y la otra 10 de 12 problemas ([Anexo 14](#)).

2.4 Fase 4: Prototipar

Se ideó un prototipo en base a una de las posibles soluciones que se proyectó en la fase anterior, donde se presentó las posibles interfaces y funcionalidades de este de las cuales destacan las siguientes: inventario, descarga de inventario, subir inventario y buscar productos, ayuda ([Anexo 15](#)). Sobre este prototipo y la retroalimentación que se obtuvo con los usuarios se definió agregar lo siguiente:

- Crear diferentes interfaces de acuerdo con el cargo que ocupe el empleado.
- Las funcionalidades dependerán del cargo que ocupe el usuario.
- Adaptar las opciones y funciones a una plataforma móvil.
- Agregar más opciones como: ver orden y buscar orden.
- Presentar una interfaz más amigable y adecuada al ámbito empresarial.

2.5 Fase 5: Evaluar

En esta fase se implementó las observaciones de la fase anterior y se ideó un prototipo número 2, donde se les solicitó otra vez a los usuarios su opinión y críticas del nuevo sistema de inventario, Sobre este prototipo y la retroalimentación con los usuarios se definió agregar lo siguiente:

- Agregar la opción de: rotación del inventario.
- Tener 4 roles o interfaces para los cargos de: gerente general, bodega, técnicos de área y jefe de proyectos.
- No mostrar toda la información sino parte de ella para que sea más fácil de entender y usar.

En conclusión, después de esta evaluación se llegó a una solución final del aplicativo móvil que se va a presentar en el capítulo siguiente.

CAPÍTULO 3

3. DESARROLLO DEL SISTEMA DE INVENTARIO SOLDEINSA

3.1 Especificaciones

En cuanto al sistema de control del inventario permitirá a la empresa Soldeinsa resolver los problemas pertenecientes al área de bodega, para tener una adecuada rotación del inventario, permitir a los usuarios realizar consultas en línea y en tiempo real de la información existente. El sistema contará con una base de datos que estará ubicada en la nube, también se tendrá una aplicación que estará conectada a la base de datos en línea, los usuarios dispondrán de interfaces de acuerdo con el rol que desempeñe. La aplicación será desarrollada en el lenguaje de programación Java, a fin de ser instalada en los celulares o Tablet de los empleados de la empresa (Figura 3.1).

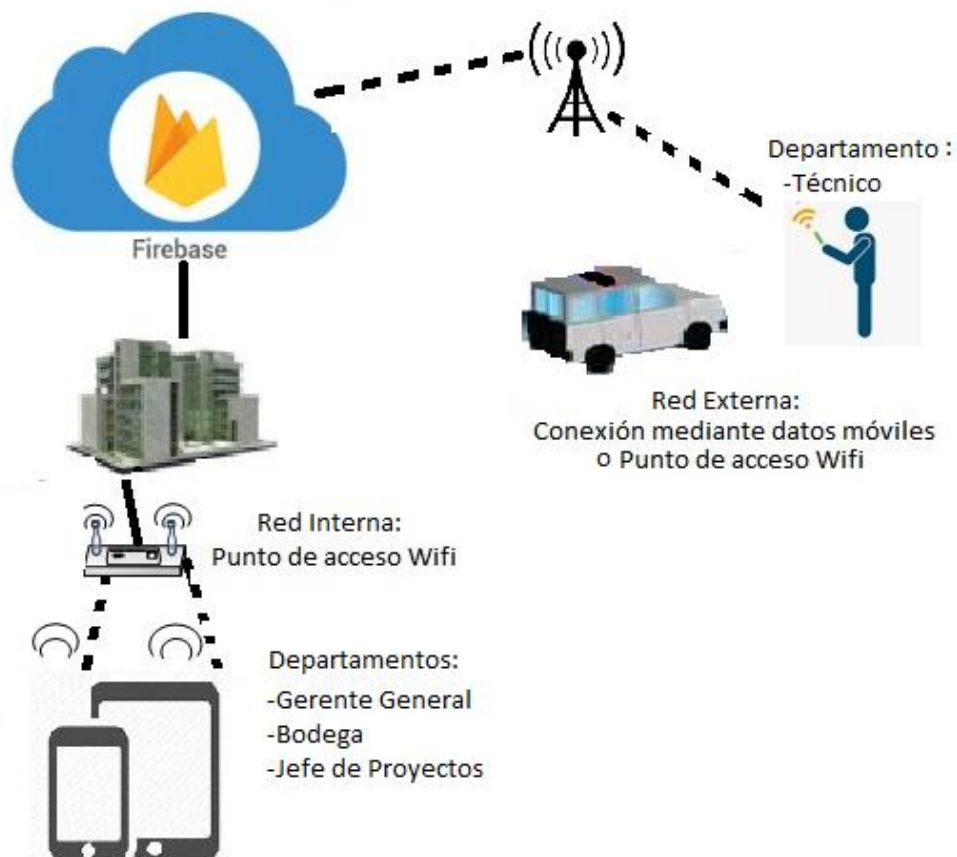


Figura 3.1 Diagrama del diseño de la solución

En la gráfica 3.1 se muestra a los 4 departamentos que usarán la aplicación móvil, donde podrán conectarse a la misma ya sea a través de la red interna (punto de acceso wifi de la empresa) o externa (datos móviles o punto de acceso wifi), además se dispone de un servicio externo de almacenamiento que estará ubicado en la nube y donde toda la información ingresada a través del aplicativo móvil será almacenada en la misma.

3.2 Creación y configuración del aplicativo móvil en Android Studio

Para comenzar con la configuración del aplicativo móvil se crearon las diferentes interfaces que van a ser visualizados por los usuarios [3], con sus respectivos botones y opciones estos archivos son .xml y se encuentran ubicada en la carpeta layout (Figura 3.2).

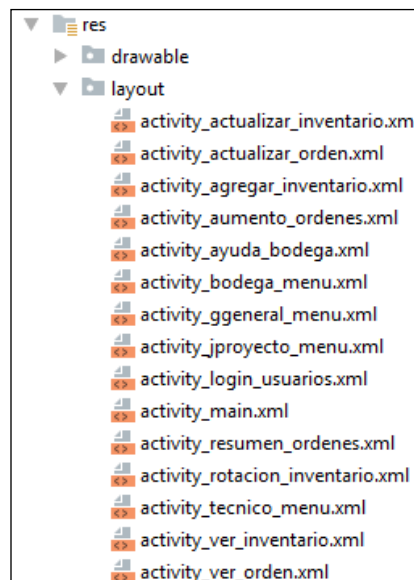


Figura 3.2 Archivos .xml de las diferentes interfaces en Android Studio

Luego de la creación de las interfaces se procedió con la configuración de cada uno de ellos ([Anexo 16](#)). [4] Esto se lo realiza en los archivos .java que se encuentran ubicados en la carpeta java (Figura 3.3).

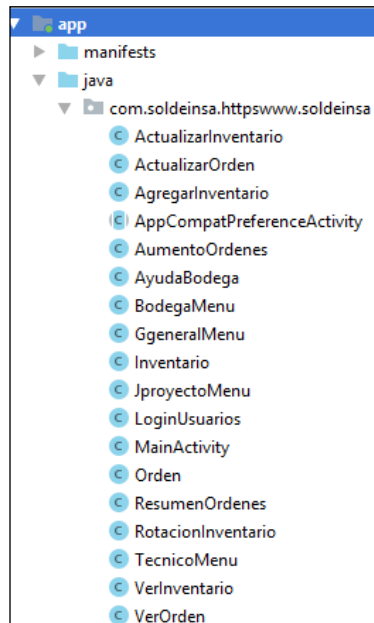


Figura 3.3 Archivos de configuración .java en Android Studio.

3.3 Conexión de la base de datos Firebase al programa Android Studio

Luego de haber creado una cuenta en Firebase se debe de instalar las librerías al programa que se diseñó [5][6], esto se lo realiza en el archivo “build. Gradle” ([Anexo 17](#)). Además en la plataforma Firebase se debe crear la conexión al programa que se diseñó y esto se realiza por medio de realizar varios pasos que pide la plataforma como: la descarga e instalación de un archivo que contiene datos de configuración para que así pueda reconocerse con el programa, el ID(identificador único) de la app, también el nombre del paquete que se encuentra en la app y por último su huella digital esto es opcional (Figura 3.4).

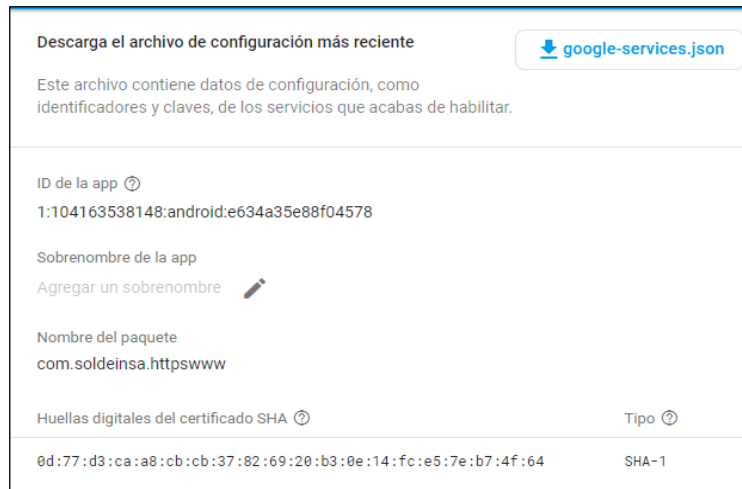


Figura 3.4 Configuración de Firebase para que se conecte al aplicativo móvil

Finalmente se visualizará en el programa de Android Studio la conexión de la base de datos al mismo, también se debe instalar la opción de que la información se ingrese o visualice a tiempo real para esto solo se presiona en la opción y se instalará los archivos [7] (Figura 3.5).

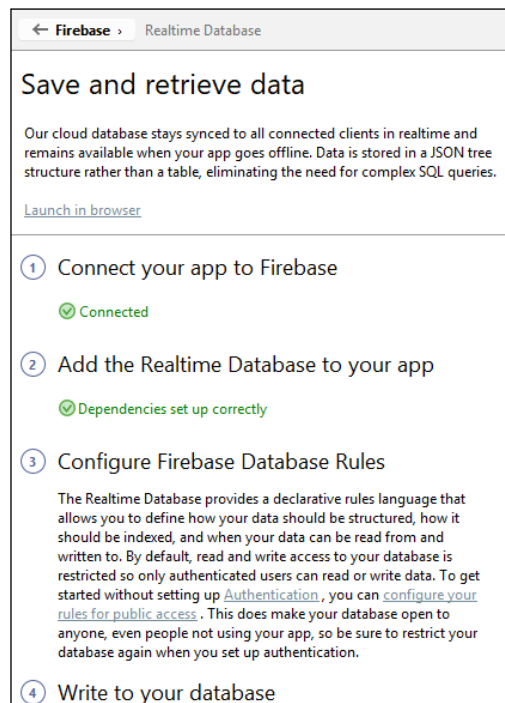


Figura 3.5 Conectividad del aplicativo móvil en Android Studio.

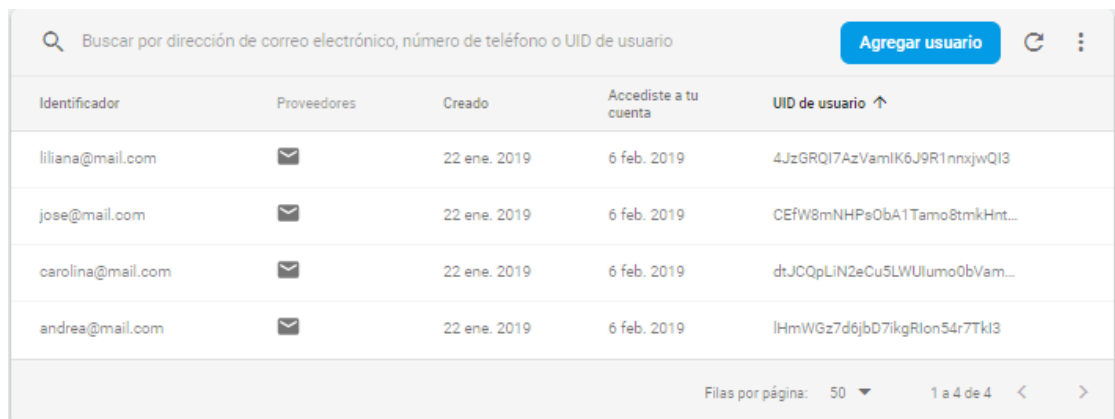
3.4 Creación de los Usuarios en la base de datos Firebase

En primer lugar, para la creación de los usuarios se pensó en la asignación de un usuario y contraseña para los 4 diferentes roles o cargos que dispone el aplicativo móvil (Tabla 3.1).

Cargo	Usuario	Contraseña
Bodega	Andrea@mail.com	123456
Gerente General	jose@mail.com	123456
Jefe de proyecto	Liliana@mail.com	123456
Técnico	carolina@mail.com	123456

Tabla 3.1 Usuario y contraseña para cada departamento dentro de la aplicación móvil.

Después de tener asignado un usuario y contraseña para cada cargo se procedió a su creación en la base de datos Firebase, donde el mismo cuenta con un módulo de autenticación para la creación de estos (Figura 3.6).



Identificador	Proveedores	Creado	Accediste a tu cuenta	UID de usuario ↑
liliana@mail.com	✉	22 ene. 2019	6 feb. 2019	4JzGRQI7AzVamIK6J9R1nnxjwQI3
jose@mail.com	✉	22 ene. 2019	6 feb. 2019	CEfW8mNHPsObA1Tamo8tmkHnt...
carolina@mail.com	✉	22 ene. 2019	6 feb. 2019	dtJCQpLiN2eCu5LWUumo0bVam...
andrea@mail.com	✉	22 ene. 2019	6 feb. 2019	lHmWGz7d6jbD7ikgRlon54r7TkI3

Figura 3.6 Registro de usuarios en Firebase.

Después en el programa de Android estudio se creará un contenedor para las UID de los usuarios, estos son los identificadores de los usuarios dentro de la base de datos y donde deben ser almacenadas en el programa de Android Studio para que así lo reconozca y trabajen simultáneamente ([Anexo 18](#)).

Finalmente, cada usuario registrado podrá ingresar al aplicativo móvil según su usuario y contraseñas (Figura 3.7).

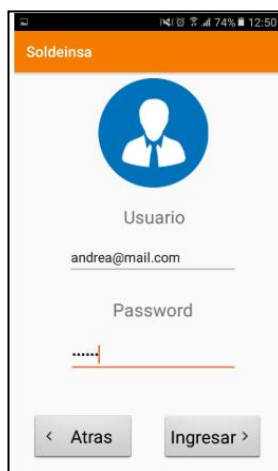


Figura 3.7 Interfaz de ingreso de usuario al aplicativo móvil.

3.5 Funcionalidades del aplicativo móvil según los cargos de los usuarios

Luego de que cada usuario disponga de su usuario y contraseña con el cargo que se le haya designado (Gerente general, gerente de proyecto, técnico, bodega).

Se procedió con la asignación de las funciones dentro del aplicativo móvil de acuerdo con la función que realiza cada uno de ellos dentro de la compañía (Tabla 3.2).

Funciones	Operador de Bodega	Gerente General	Jefe de Proyectos	Técnico
Ver inventario	X	X		X
Agregar inventario	X			
Actualizar inventario	X			
Ver orden	X	X	X	
Actualizar orden	X			
Rotación del inventario	X			
Resumen ordenes				X
Aumento de ordenes			X	
Ayuda	X	X	X	X

Tabla 3.2 Funcionalidades del aplicativo móvil según su rol.

De igual modo cada uno de los roles contara con una interfaz de usuario dentro del aplicativo móvil, de acuerdo con las opciones que dispone (Figura 3.8).



Figura 3.8 Menú principal de cada uno de los roles dentro del aplicativo móvil.

3.5.1 Funcionalidades del aplicativo móvil

Ver Inventario

Esta opción le permite al usuario ver los productos existentes en el inventario, donde previamente fueron almacenados en la base de datos, se puede realizar la búsqueda de un producto por medio del nombre o código donde al final se podrá visualizar el mismo (Figura 3.9).



Figura 3.9 Opción de ver inventario dentro del aplicativo móvil

➤ **Agregar inventario**

Esta opción permite ingresar un producto a la base de datos por medio del botón guardar, cabe recalcar que se debe de llenar todos los campos que son obligatorios, también en caso de que no desee registrar algo e ir al menú principal existe el botón de cancelar (Figura 3.10).



Figura 3.10 Opción de agregar inventario dentro del aplicativo móvil

Ademas se puede verificar y vizualizar lo ingresado en la base de datos Firebase (Figura 3.11).

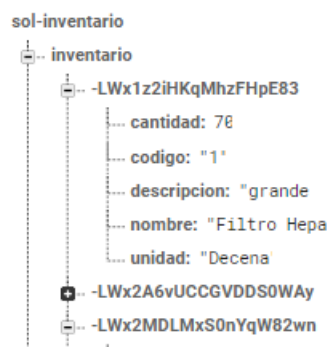


Figura 3.11 Ingreso de un producto a la base de datos

➤ **Actualizar inventario**

Después de haber ingresado el código o nombre del producto se presenta una interfaz con todos los campos llenos de la información que previamente se ingresó de ese producto donde se puede modificar y guardar en la base de datos o cancelar (Figura 3.12).



Figura 3.12 Opción de actualizar un producto en el aplicativo móvil

La modificación de un producto se puede visualizar en la base de datos de forma instantánea (Figura 3.13).

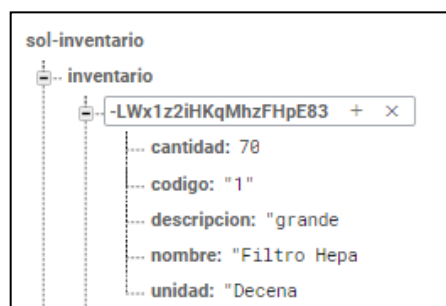


Figura 3.13 Modificación de un producto en la base de datos

➤ **Ver orden**

Esta opción permite ver las ordenes de trabajos que previamente se ingresaron en la base de datos, donde se debe ingresar el código de la orden para que esta se muestre al presionar el botón buscar, también se puede regresar al menú principal al presionar el botón atrás (Figura 3.14).



Figura 3.14 Opción de ver órdenes de trabajo en el aplicativo móvil



Actualizar o resumir ordenes

Esta opción permite actualizar o resumir las ordenes de trabajos que previamente se ingresaron en la base de datos, donde se debe ingresa el código de la orden para que esta se muestre al presionar el botón buscar, también se puede regresar al menú principal al presionar el botón atrás (Figura 3.15).



Figura 3.15 Opción de actualizar órdenes de trabajo en el aplicativo móvil.

La modificación de una orden se puede visualizar en la base de datos de forma instantánea (Figura 3.16).

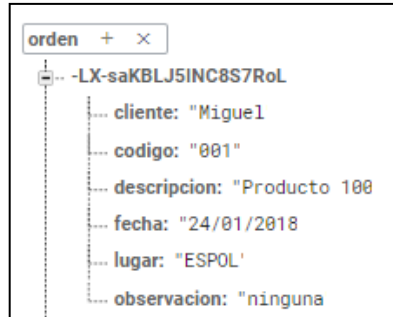


Figura 3.16 Modificación de una orden en la base de datos

➤ Rotación del inventario

Por medio de esta opción se puede verificar si un producto está a punto de acabarse o si se cuenta con el suficiente, esto se logra al ingresar el número que se compra mensualmente del mismo y el código de este, al final se mostrara un mensaje con lo que se dispone y que se debe hacer (Figura 3.17).

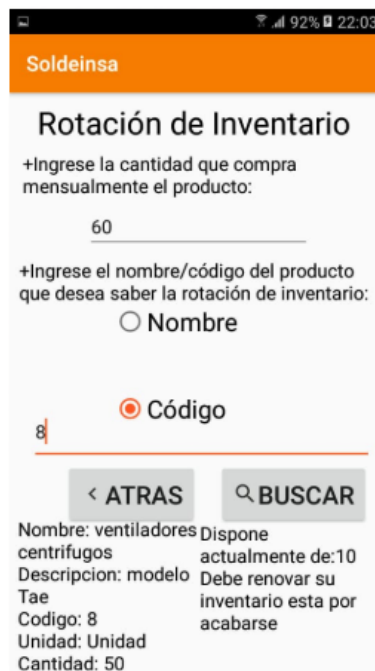


Figura 3.17 Opción de rotación de inventario en el aplicativo móvil

➤ **Aumento de ordenes**

Para pedir el aumento de una orden de trabajo se debe llenar todos los campos y se almacenan al presionar el botón de guardar y para regresar al menú principal está el botón de atrás (Figura 3.18).



Soldeinsa

Aumento de Ordenes

1111111

12.12.2018

pepito

ninguno

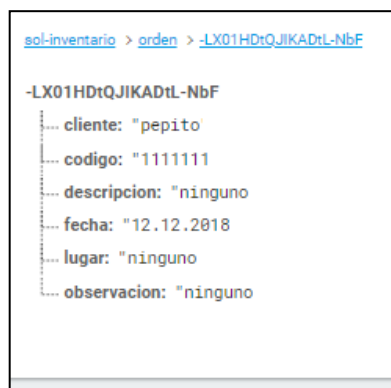
ninguno

ninguno

< ATRAS GUARDAR

Figura 3.18 Opción de aumento de órdenes en el aplicativo móvil.

También se puede constatar esto en la base de datos de forma instantánea la nueva orden de trabajo (Figura 3.19).



```
sol-inventario > orden > -LX01HDtQJIKADtL-NbF

-LX01HDtQJIKADtL-NbF
... cliente: "pepito"
... codigo: "1111111"
... descripcion: "ninguno"
... fecha: "12.12.2018"
... lugar: "ninguno"
... observacion: "ninguno"
```

Figura 3.19 Creación de una orden en la base de datos

➤ **Ayuda**

La opción de ayuda da varios consejos al usuario en caso de que tenga algún problema con el aplicativo móvil (Figura 3.20).

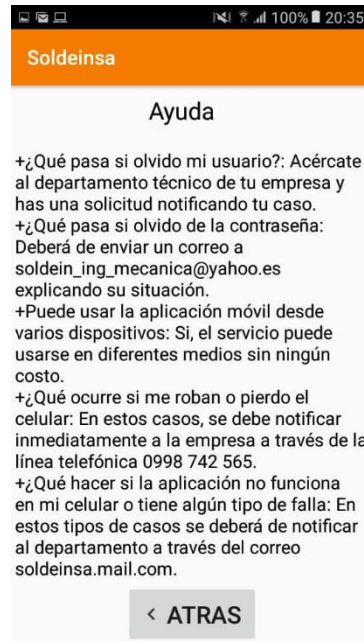


Figura 3.20 Opción de ayuda en el aplicativo móvil

CAPÍTULO 4

4. IMPLEMENTACIÓN Y PRESUPUESTO

4.1 Plan de Implementación

El proyecto tuvo una duración de 75 días de los cuales se creó un cronograma de las actividades que se realizaron, a fin de cumplir con las restricciones y especificaciones del dueño de la compañía se pensó en el cronograma de forma secuencial de acuerdo a lo que se requería, se lo realizó en un plazo de 2 meses y 15 días de jornada laboral. (Figura 4.1)

En cuanto al sistema de inventario después de ser diseñado y probado, se procedió con la instalación de este en los dispositivos autorizados de la empresa, donde por medio de capacitaciones se les enseñó al personal su funcionamiento de acuerdo con su rol, para que así puedan manejar el aplicativo de una forma óptima, luego se les debe entregar un manual del uso y funcionalidad del aplicativo móvil, esto servirá para los futuros empleados. Finalmente es indispensable recolectar toda la información de los diferentes materiales que dispone o utiliza Soldeinsa en bodega para proceder con la asignación de un código o nombre adecuado, esto genera un listado autorizado por la empresa, para el ingreso adecuado de los productos dentro del aplicativo móvil.

Listo	Nombre de la tarea	Fecha de Inicio	Fecha final	Duraci...	Precios por Actividad
✓	DISEÑO DE UN SISTEMA PARA CONTROL DE INVENTARIO Y LOGÍSTICA, PARA LA EMPRESA SOLDEINSA CON APLICACIÓN ANDROID EN DISPOSITIVOS MÓVILES QUE SE COMUNIQUEN A TRAVÉS DE REDES CELULARES Y A LA RED WIRELESS DE LA EMPRESA	15/10/18	25/01/19	75d	
✓	+ Design Thinking	15/10/18	16/11/18	25d	\$15,00
✓	- Desarrollo del aplicativo móvil	19/11/18	25/01/19	50d	\$2.000,00
✓	Creación de interfaces	19/11/18	21/11/18	3d	\$120,00
✓	Configuración de interfaces	22/11/18	07/12/18	12d	\$480,00
✓	conexión de Firebase a Android Studio	10/12/18	10/12/18	1d	\$40,00
✓	Creación de usuarios en Firebase	11/12/18	11/12/18	1d	\$40,00
✓	Unificación de los sistemas	12/12/18	14/12/18	3d	\$120,00
✓	Prueba de funcionalidad del sistema	17/12/18	28/12/18	10d	\$400,00
✓	Instalación del sistema dentro de la empresa	31/12/18	04/01/19	5d	\$200,00
✓	Instalación del sistema dentro de la empresa	31/12/18	04/01/19	5d	\$200,00
✓	Capacitación del personal según su rol	07/01/19	11/01/19	5d	\$200,00
✓	Migración de la información	14/01/19	18/01/19	5d	\$200,00
✓	Evaluación final del funcionamiento del sistema	21/01/19	25/01/19	5d	\$200,00

Figura 4.1 Diagrama de Gantt del plan de implementación del proyecto

4.2 Presupuesto

4.2.1 Cuadro de costo del proyecto

Tarea	Horas	Costo	Total
Desarrollo del aplicativo móvil	128 horas	\$5 por hora	\$640,00
Desarrollo de la base de datos	32 horas	\$5 por hora	\$160,00
Instalación y prueba del sistema	120 horas	\$5 por hora	\$600,00
Capacitación del personal	40 horas	\$5 por hora	\$200,00
Migración de la información	40 horas	\$5 por hora	\$200,00
Evaluación final de funcionalidad	40 horas	\$5 por hora	\$200,00
Gastos varios			\$15,00
Total, de costos			\$2,015

Tabla 4.1 Costo del proyecto

Como podemos observar, el costo total de inversión del proyecto será de \$2.015 dólares.

4.2.1.1 Mano de obra

Por los conocimientos que tenía el personal del proyecto y por lo que se pensaba hacer se decidió cobrar \$5 por cada hora que se haga en el proyecto.

4.2.1.2 Gastos varios

Se hizo uso de varios utensilios como: lápiz, cuadernos, plumas. Para tener un registro aparte de lo que se investigaba no solo de forma digital, además se tuvo que imprimir varias partes de la documentación para su presentación de lo que se iba avanzando en el mismo.

4.2.1.3 Servicios externos

Se necesitó del servicio de una plataforma en línea de la cual provee el servicio de almacenamiento de información para bases de datos de las cuales tienen un costo y un plan de acuerdo con la necesidad de la empresa

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

La investigación se realizó con el fin de saber que funcionalidades debía cumplir el nuevo sistema de inventario para así satisfacer con todos los objetivos planteados, además se pudo observar que el área de bodega y compras eran los que tenían más problemas en la compañía y este a su vez hacía que los demás departamentos sufrieran contratiempos a la hora de realizar sus actividades, por ende, se le otorgó a los empleados de esas áreas, diferentes funcionalidades en el sistema de acuerdo con su rol que desempeñaba en la empresa.

Uno de los objetivos esenciales para este proyecto, era aumentar la rotación de inventario dentro de la empresa, originalmente ellos no contaban con una rotación, dando como resultado incongruencias en materiales en físico y en el sistema. Para realizar la elaboración de este sistema de inventario se necesitó de la ayuda del programa Android Studio aprendido en el transcurso de la carrera, también se utilizó el sistema de base de datos llamado Firebase para el desarrollo de aplicaciones web y móvil, aprendido en el lapso investigación del proyecto. Este sistema permite ingresar los datos de forma automatizada a la base de datos que está ubicada en la nube, para obtener datos reales y actualizados ya que este programa tiene la accesibilidad de irse alimentando semanalmente, con la finalidad de que se pueda obtener un registro histórico del comportamiento de los materiales entrantes y salientes durante toda la jornada laboral.

Luego de culminar con el proyecto, y de ampliar los conocimientos previamente adquiridos se pudo al final, crear y obtener un sistema móvil unificado y compacto para el uso completo dentro y fuera de la compañía Soldeinsa.

Recomendaciones

- Se recomienda tener un buen proveedor de internet, para evitar problemas con el ingreso o actualización del aplicativo móvil.
- Incluir nuevas funcionalidades como por ejemplo un sistema para escritorio para que así se utilice el aplicativo al 100%.
- Se recomienda que el sistema sea configurado y manipulado por personas especializadas en el manejo de base de datos y programación en aplicativos móviles Android.
- Se recomienda realizar capacitaciones al personal responsable de cada departamento que usará el aplicativo móvil.
- Realizar mediciones del rendimiento del inventario periódicamente.

BIBLIOGRAFÍA

- [1] Valverde Landívar Galo, Robayo Ayala Hugo, Romero Gerardo, “Sistema de Control de Inventarios Mediante el Uso de Tecnologías de Internet y Comunicación Inalámbrica”,(febrero, 2018).
- [2] Villamil Torres Diana Paola, La Implementación Del Control Interno De Inventarios Para El Sector Droguista De Acuerdo Con Las Normas Internacionales De Auditoria – Nias (Marzo, 2015).
- [3] Catalán Adrián, Medium, 26 Octubre 2015.[En línea] Available: <https://medium.com/@ykro/mi-primer-app-con-firebase-y-android-parte-1-fd0b7d717e0b>. [Último acceso: 26 Octubre 2015].
- [4] Mkyong, Mkyong, 24 noviembre 2011.[En línea] Available: <https://www.mkyong.com/android/android-radio-buttons-example/>. [Último acceso: 29 Agosto 2012].
- [5] Firebase, Firebase, 29 Marzo 2019.[En línea] Available: <https://firebase.google.com/docs/database/android/start?hl=es-419>. [Último acceso: 29 Marzo 2019].
- [6] Firebase, Firebase, 22 Marzo 2019.[En línea] Available: <https://firebase.google.com/docs/android/setup>. [Último acceso: 22 Marzo 2019].
- [7] Firebase, Firebase, 29 Marzo 2019.[En línea] Available: <https://firebase.google.com/docs/database/android/start?hl=es-419>. [Último acceso: 29 Marzo 2019].
- [8] Guilherme Lautert, Stackoverflow, 31 Agosto 2016. .[En línea] Available: <http://pt.stackoverflow.com/questions/150168/invent%C3%A1rio-com-andro>. [Último acceso: 13 Abril 2017].

Anexos

Anexo 1

Desk research

Se procedió a realizar la debida investigación de escritorio de la empresa Soldeinsa y el medio que se utilizó para realizar la investigación fue a través de internet.

La información de la empresa la encontramos en: <https://www.soldeinsa.com/>

Como se mencionó la empresa cuenta con una página web donde pudimos encontrar más información respecto a los servicios que ofrece, trabajos que ha realizado e información de contacto con la empresa esta última nos sirvió para poder contactarnos con la empresa y a su vez con el dueño de la empresa y así poder ofrecerle la idea de poder realizar un tema de tesis con la resolución de poder resolver alguna problemática de su empresa la cual fue aceptada de buena manera por parte del dueño de la empresa el Sr. Plutarco Salomón Chico Mackliff que es gerente general de la empresa Soldeinsa. A continuación, se presentará lo que tomamos en cuenta para conocer más a profundidad a la empresa Soldeinsa y saber sus problemáticas, se necesitó hacer esto para poder así sacar una posible solución y tema de tesis.

Anexo 2

Preparación para la entrevista

Para la preparación de las entrevistas se realiza el método de las lluvias de ideas, las mismas que se aplicaron para dos tipos de personas:

- Gerente general de la empresa
- Trabajadores de la empresa (en general)

Gerente general

Andrea Hurtares	Liliana Dávila
¿Nombre completo? ¿Cargo dentro de la empresa? ¿Edad? ¿Años desde que se fundó de la empresa? ¿Número total de empleados en la empresa? ¿Cuántos empleados fijos y no fijos tienen contratados? ¿Qué tipo de perfiles profesionales contrata dentro de la empresa? ¿Qué tipo de horarios se labora en la empresa? ¿Horarios de atención a los clientes? ¿Qué tipos de problemas cree que tiene la empresa? ¿Cómo cree usted que se pueda resolver estos problemas? ¿Tiene deseos de poder expandirse a otros mercados de servicios? ¿Tiene planes de expandirse y sacar sucursales en otras partes del país? ¿Cuánto tiempo cree que le tome poder realizar la expansión de su empresa? ¿Cree que pueda resolver totalmente las dificultades de la empresa? ¿Está dispuesto a invertir dinero en poder resolver dichas problemáticas?	¿Nombre completo? ¿Cargo? ¿Tiempo en la empresa? ¿Qué tipo de problema considera usted que tiene la empresa? ¿Utiliza un sistema de inventario para llevar el control de la mercadería? ¿Cuánto tiempo lleva usando el sistema? ¿Ha tenido inconvenientes con el sistema? ¿Desearía llevar un mayor control de la mercadería? ¿Tienes planes de expansión para Soldeinsa dentro y fuera del país? ¿Cuántas personas están a cargo del inventario de la empresa? ¿Cuánto empleado fijo y no fijo tiene? ¿Cuánta hora de jornada laboral tiene la empresa? ¿Cuáles son los horarios de atención al cliente? ¿Cuáles son los cargos que existen en la actualidad de la empresa? ¿Cuánto tiempo lleva en el mercado la empresa? ¿Desearía Soldeinsa como empresa expandirse en otras áreas de servicio al cliente? ¿Con que tipo de perfiles profesionales trabaja la empresa? ¿Cuáles son los problemas principales en cuanto a su empresa? ¿Por qué el nombre Soldeinsa?

¿Hasta cuanto cree que pueda invertir en capital para poder resolver? ¿Hasta cuánto tiempo está dispuesto a esperar por la resolución de una problemática de la empresa? ¿Se considera una persona accesible? ¿Cree que sus empleados cuentan con un buen ambiente laboral? ¿Cree que se trabaja bajo un nivel de presión alto en la empresa? ¿Cree que sus empleados están conforme con su trabajo por el que fueron contratados?	¿Cuándo pasó lo del decrecimiento de la empresa en tiempos difíciles, pensó en hipotecarla? ¿Los empleados tienen remuneraciones?
--	---

Figura 1. Ejemplo de lo que realizamos individualmente cada una de las personas involucradas en el proyecto con respecto a dar lluvias de ideas para el gerente general de Soldeinsa.

Trabajadores de la empresa (en general)

Andrea Hurtares	Liliana Dávila
¿Nombre completo? ¿Cargo dentro de la empresa? ¿Edad? ¿Estado civil? ¿Años trabajando en la empresa? ¿Se siente a gusto de trabajar en Soldeinsa? ¿Se siente cómodo con el ambiente laboral de la empresa? ¿Ha tenido algún tipo de problema con algún compañero de trabajo dentro de la empresa? ¿Se trabaja bajo presión dentro de la empresa? ¿Piensa que hace más trabajo por el cual fue contratado? ¿Cuáles cree que son los problemas que tiene la empresa? ¿Cómo cree que se puede resolver esos problemas? ¿Qué tan urgente ve que se deba resolver esos problemas en la empresa?	¿Nombre completo? ¿Cargo? ¿Tiempo en la empresa? ¿Qué tipo de problema considera usted que tiene la empresa? ¿Cuáles son los horarios de atención al cliente? ¿Tiene remuneración? ¿Qué puede hacer para mejorar la empresa? ¿Qué espera Ud. de esta empresa? ¿Tiene los materiales y equipo necesario para hacer el trabajo de manera correcta? ¿Hay alguien en el trabajo que aliente su desarrollo? ¿Sus compañeros de trabajo están comprometidos en hacer un trabajo de calidad? En el trabajo, ¿sus opiniones cuentan?

--	--

Figura 2. Ejemplo de lo que realizamos individualmente cada una de las personas involucradas en el proyecto con respecto a dar lluvias de ideas para el personal de trabajo de Soldeinsa.

Anexo 3

Entrevista para empalizar

Encuesta Soldeinsa 1

Nombre completo: Plutarco Salomon Chico Mackliff

Cargo: Gerente General

Tiempo en la empresa: 12 años

¿Usted cree que Soldeinsa consta de un buen ambiente laboral? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

1	2	3	4	5	6	7	8	X 9	10
---	---	---	---	---	---	---	---	--------	----

¿Indique el grado de presión en el cual se trabaja? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

1	2	3	4	5	6	7	8	X 9	10
---	---	---	---	---	---	---	---	--------	----

¿Qué tipo de problema considera usted que tiene la empresa?

Inventario, liquidez, logística, control de materiales de obras y herramientas.

¿Utiliza un sistema de inventario para llevar el control de la mercadería? ¿Cuánto tiempo lleva usando el sistema? ¿Ha tenido inconvenientes con el sistema?

Utilizan Excel.

¿Desearía llevar un mayor control de la mercadería?

Si.

¿Tienes planes de expansión para Soldeinsa dentro y fuera del país?

No, por el momento todo lo hacen movilizándolo el material dentro del país.

¿Cuántas personas están a cargo del inventario de la empresa?

1

¿Cuánto empleado fijo y no fijo tiene?

14 afiliadas y 30 eventuales.

¿Cuánta hora de jornada laboral tiene la empresa?

8 am a 5 pm de lunes a viernes.

¿Cuáles son los horarios de atención al cliente?

Oficina: 8am a 5pm y en obra desde las 10pm.

¿Cuáles son los cargos que existen en la actualidad de la empresa?

Gerente, gerente de proyecto, Asistente contable, compras -bodega e inventario, jefe de proyecto y técnico.

¿Cuánto tiempo lleva en el mercado la empresa?

7 años

¿Desearía Soldeinsa como empresa expandirse en otras áreas de servicio al cliente?

No por el momento. Solo se dedican en ingeniería mecánica, gases medicinales y vapor.

¿Con que tipo de perfiles profesionales trabaja la empresa?

Ingeniería mecánica, contadores y técnico.

Nombre completo: Jaime Edmundo Guaita Defaz

Cargo: Gerente de Proyecto

Tiempo en la empresa: 6 años

¿Usted cree que Soldeinsa consta de un buen ambiente laboral? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

1	2	3	4	5	6	7	8	X 9	10
---	---	---	---	---	---	---	---	--------	----

¿Indique el grado de presión en el cual se trabaja? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

1	2	3	4	5	6	7	8	X 9	10
---	---	---	---	---	---	---	---	--------	----

¿Qué tipo de problema considera usted que tiene la empresa?

Logísticas y cartera vencida.

¿Tiene alguna idea de cómo solucionar la problemática que indicó?

Contratar personal calificado. Especialista en cartera vencida.

Encuesta Soldeinsa 3

Nombre completo: Rody Paul Nacevilia Diaz

Cargo: Compras

Tiempo en la empresa: 11 meses

¿Usted cree que Soldeinsa consta de un buen ambiente laboral? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

1	2	3	4	5	6	7	8	9	X 10
---	---	---	---	---	---	---	---	---	---------

¿Indique el grado de presión en el cual se trabaja? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

1	2	3	4	5	6	7	8	X 9	10
---	---	---	---	---	---	---	---	--------	----

¿Qué tipo de problema considera usted que tiene la empresa?

Logística

¿Tiene alguna idea de cómo solucionar la problemática que indicó?

Organizando la movilización.

Encuesta Soldeinsa 4

Nombre completo: Jaime Gustavo Cofre Unapanta

Cargo: Supervisor de Proyecto

Tiempo en la empresa: 7 años

¿Usted cree que Soldeinsa consta de un buen ambiente laboral? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

								X	
1	2	3	4	5	6	7	8	9	10

¿Indique el grado de presión en el cual se trabaja? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

						X			
1	2	3	4	5	6	7	8	9	10

¿Qué tipo de problema considera usted que tiene la empresa?

Compras ya que ellos son el cuello de botella del envío de material, que no llega a tiempo.

¿Tiene alguna idea de cómo solucionar la problemática que indicó?

Aplicación para que compra se organizara mejor.

Encuesta Soldeinsa 5

Nombre completo: Darwin Patricio Quinche

Cargo: Supervisor de Proyector Costa

Tiempo en la empresa: 4 años

¿Usted cree que Soldeinsa consta de un buen ambiente laboral? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

								X	
1	2	3	4	5	6	7	8	9	10

¿Indique el grado de presión en el cual se trabaja? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

							X		
1	2	3	4	5	6	7	8	9	10

¿Qué tipo de problema considera usted que tiene la empresa?

Ventas, demora en despacho de material

¿Tiene alguna idea de cómo solucionar la problemática que indicó?

Contratar más personal que se dedique a cada área.

Encuesta Soldeinsa 6

Nombre completo: Freddy Mauricio Paucar Quintuña

Cargo: Técnico

Tiempo en la empresa: 6 años

¿Usted cree que Soldeinsa consta de un buen ambiente laboral? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

							X		
1	2	3	4	5	6	7	8	9	10

¿Indique el grado de presión en el cual se trabaja? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

				X					
1	2	3	4	5	6	7	8	9	10

¿Qué tipo de problema considera usted que tiene la empresa?

Bodega despacho de material.

¿Tiene alguna idea de cómo solucionar la problemática que indicó?

Contratar más personal.

Encuesta Soldeinsa 7

Nombre completo: Zulema Pilar Guaiña Chiluisa

Cargo: Asistente Contable

Tiempo en la empresa: 2 años

Usted ¿cree que Soldeinsa consta de un buen ambiente laboral? Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

						X			
1	2	3	4	5	6	7	8	9	10

Indique el grado de presión en el cual se trabaja. Marque con una X, Las respuestas serán puntuadas en una escala de 1 a 10, donde 10 es completamente satisfecho y 1 es completamente insatisfecho.

					X				
1	2	3	4	5	6	7	8	9	10

¿Qué tipo de problema considera usted que tiene la empresa?

Falta de coordinación en las actividades

Falta de conocimiento en las funciones de cada empleado

¿Tiene alguna idea de cómo solucionar la problemática que indicó?

Diseño de plan de actividades

Reestructuración, diagrama de actividades y función.

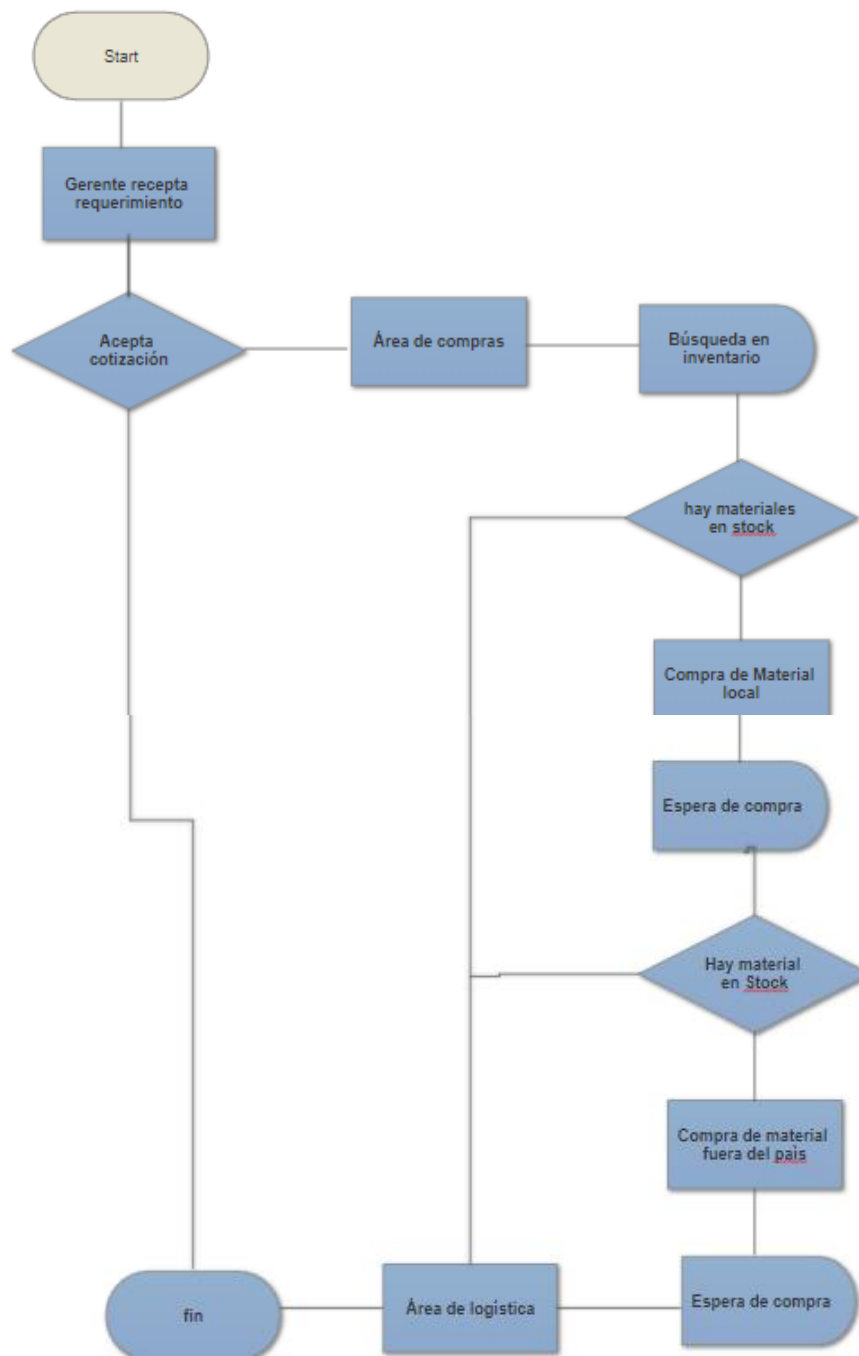
Anexo 4

Análisis DAFO

<u>FORTALEZAS:</u> • Buen ambiente laboral. • calidad de entrega de producto final. • conocimiento del mercado • años de experiencia en el mercado. • las oficinas deben estar bien equipadas y con artículos de oficina en excelente condiciones y calidad. • Servicios de alto nivel. • Base de clientes establecidos. • Diversidad de productos.	<u>OPORTUNIDADES:</u> • Constante crecimiento de la industria. • Alta demanda de servicios o productos. • Bajas tarifas. • Necesidad del producto.
<u>DEBILIDADES:</u> • Salarios deficientes y pagos no puntuales. • No capacitar al personal. • Problemas financieros. • Bajo nivel de venta. • Falta de planeación. • Gerente deficiente. • No dar incentivos o créditos al personal.	<u>AMENAZAS:</u> • Fuerte competencia. • Aumento de precios. • Poca o baja contratación de empleados. • Competencia consolidada en el mercado. • Bajo crecimiento de la organización o empresa. • Espera de pagos de servicios ofrecidos a instituciones públicas. • Sobrecarga de tareas al personal.

Anexo 5

Flujo de procesos



Anexo 6

¿Qué? ¿Cómo? ¿Por qué?

Problema de la empresa

¿Qué?:

¿Qué problema hay en la empresa?

Al no haber un adecuado manejo y control en el inventario de la empresa suele afectar a los demás departamentos de la estructura jerárquica de la empresa ya que es muy común que no se sepa realmente que hay en la bodega de la empresa, ahora revisar la bodega de la empresa causa que se pierda tiempo ya que realizan este proceso en reiteradas ocasiones del día. No solo se pierde tiempo al hacer la búsqueda de un producto sino al hacer la entrega de este al cliente causando molestias y malestar al cliente, esto puede llevar a que el cliente escoja en su próxima compra a otro proveedor que ofrezca los mismos productos que Soldeinsa. La empresa también hace compras en el exterior las cuales importa al país y las vende a sus clientes pero en ocasiones suele quedarse sin productos porque no tuvo un control en el inventario el cual le diga que debía pedir con antelación ciertos productos ocasionando que la empresa deba comprar productos en el país los cuales les salen a costos superiores a los que suele comprar en el exterior, esto causa una pérdida de ganancias al final de realizar un trabajo.

¿Cómo?:

¿Cómo sucede el problema?

El problema se origina porque no hay una buena estructura jerárquica dentro de la empresa la cual podemos notar en la siguiente grafica donde nos muestra cómo ve la empresa el dueño de la empresa.

Organización de la empresa



Figura 3. Grafica proporcionada por el Sr. Plutarco Salomon Chico Mackliff gerente general de la empresa Soldeinsa de cómo ve la estructura jerárquica de la empresa.

Como podemos notar dicha grafica no tiene un orden ni una adecuada unión de tablas de relación donde especifique las tareas que realizan cada empleado y sus funciones, también no une las tareas de acuerdo con las relaciones con sus atributos. Ahora todo esto nos hace notar la falta de organización a nivel jerárquico el cual afecta a la hora de hacer el debido control del inventario ya que al no haber comunicación con todos los departamentos causa desorden y no notifican lo que les llega o no a su departamento.

¿Por qué?:

¿Por qué sucede el problema?

En cuanto al problema sucede porque no se ha tomado el debido estudio del problema y a su vez no se ha tomado en serio las consecuencias que podría causar a la empresa, además esto si sigue ocurriendo podría causar grandes pérdidas a la empresa. Por eso nuestro objetivo es poder ayudar a la empresa y así ayudarla a crecer como institución privada en sus actividades futuras como empresa con nuestra solución.

Anexo 7

Ficha de Insights

La problemática presentada surge como consecuencia de la falta de un orden de inventario en el departamento de ventas de la empresa “Soldeinsa”, el registro y control en concreto son las falencias que necesitan ser atendidas por medio de un sistema de inventario que este a la vanguardia de las tecnologías. En muchas ocasiones al no contar con un inventario moderno ocasiona una pérdida de tiempo al hacer la búsqueda del producto deseado.

Desde el punto de vista del gerente general, hay una falencia o necesidad por parte del departamento de ventas desperdiciando en algunas ocasiones el tiempo a dedicar en otras actividades.

Desde el punto de vista de los empleados, la falta de orden y logística, así como la atención al cliente se torna en un descontrol provocando un bajo rendimiento en las asignaturas de cada departamento.

INSIGHT SOLDEINSA

RECOMENDACIÓN	Soldeinsa debe promover un cambio de perspectiva del personal, para que se adapten nuevas tecnologías.
INSIGHT	Control de inventarios del material saliente y entrante, sin necesidad del recuento.
HALLAZGO	Soldeinsa es una empresa que avanza día a día, con el fin de ser más eficaces y brindar el mejor servicio.
INFORMACIÓN	Los empleados se sienten a gusto con la empresa y les agrada trabajar bajo presión.
DATO	Hay un incremento notable en el trabajo de implementación en Soldeinsa.

Figura 5. Cuadro de especificaciones de la problemática de la empresa Soldeinsa.

Anexo 8

Point of View

USUARIO	NECESIDAD	INSIGHT
SOLDEINSA	Tecnología	Mejora del sistema utilizado para así avanzar como empresa y estar al día con las nuevas tecnologías.
	Orden	Tener un orden es impórtate para poder hacer una atención más eficaz y rápida a los clientes.
	Control	Se podrá constatar si se están cumpliendo: las operaciones, logística y procesos de abastecimiento de la empresa para así avanzar como está previsto.
	Estar Relajados	Tener un mejor ambiente laboral en la empresa ayudara a que los empleados realicen sus actividades de una forma productiva y eficaz.
	Productividad	Al tener recopilada toda la información de los productos que poseen y cuentan, podrán dar un mejor servicio de forma más rápida y oportuna.
	Automatizar	Provee un acceso a toda la información, de forma confiable y verídica, capacidad de compartir datos entre todas las áreas de la empresa.
	Adaptabilidad	Al contar con un nuevo sistema de inventario le permite poder estar a la altura de las nuevas tecnologías que existen hoy y de ser escalable.
	Rentabilidad	Al tener un mejor control en el área de bodega se podrá distribuir mejor los productos y esto ayudará a disminuir el riesgo de caer en costos innecesarios.

Anexo 9

Mapa de afinidades

	COORDINACIÓN ENTRE ELLOS	
--	---------------------------------	--

ESTABLECER DIRECTRICES Y PRIORIDADES EN SU DEPARTAMENTO		PLANIFICAR SU PROPIA GESTIÓN
REUNIONES EFECTIVAS	Mayor planificación	Mantener al día el Stock
MEJOR ORGANIZACIÓN DEPARTAMENTAL	Instrucciones precisas	Tener un manual de procesos
CAPACITACIÓN CONSTANTE DEL PERSONAL	Reducir tiempos de instalación	Tener en conocimiento el organigrama funcional de la empresa
MANTENER LA CANTIDAD NECESARIA POR CADA INSTALACIÓN	Contar con la cantidad necesaria de vehículos para instalaciones múltiples	Calidad de implementos
REDUCIR TIEMPOS DE ACTUALIZACIÓN DE INVENTARIOS	Colaboración y comunicación constante entre áreas	Establecer ideas claras

Anexo 10

Redefinición del problema

Luego de la revisión de los puntos críticos que resalta la empresa, los cuales deben ser redirigidos y planteados se toman en cuenta los siguientes aspectos:

- Falta de tecnología evidente
- Desorden la importancia de la jerarquía administrativa
- Desinterés del control del área de logística
- Descontrol de inventarios en bodega y área de compras

Esto nos revela que Soldeinsa a pesar de su alto crecimiento podría caer en un déficit económico debido a la falta de mejoras que reestructuren su sistema administrativo, así mismo el control en cada área, lo que podría acarrear perdidas irrecuperables.

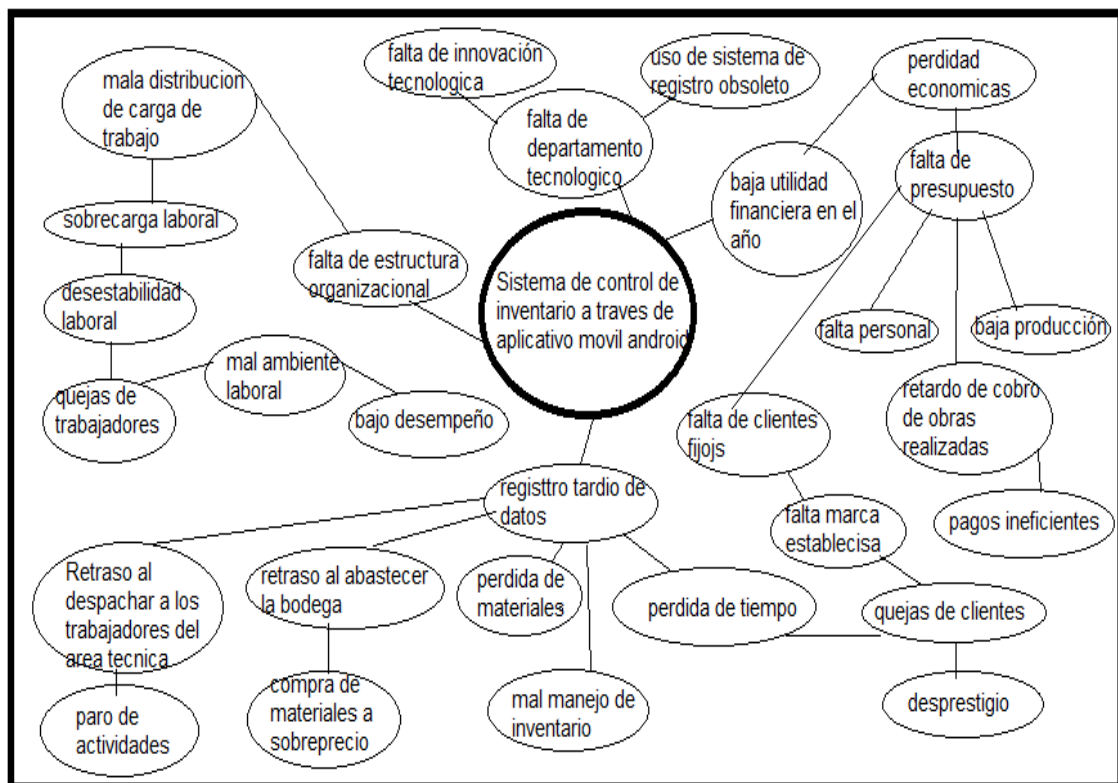
Cabe recalcar que cada área es responsable de llevar su control, pero específicamente las áreas de ventas y logística son las más afectadas en este asunto ya que por su falta de personal esto hace que la carga operativa sea mucho mayor.

Asimismo, el personal debe estar preparado a la orientación de nuevos proyectos para mejoras de la empresa y estar presto a nuevos cambios.

Anexo 11

Brainstorming

SOLDEINSA CIA LTDA



Anexo 12

Matriz Importancia - Dificultad

		MATRIZ IMPORTANCIA - DIFICULTAD	
		DIFICULTAD	
		DIFICULTAD	NO DIFICULTAD
IMPORTANCIA	IMPORTANTE	ATRASOS EN INSTALACION ORGANIZACIÓN DE AREAS INVENTARIOS INCONCLUSOS PLANIFICACIONES	COMPRAS DE IMPLEMENTOS CAPACITACION PERSONAL SITUACIONES DE CRISIS INSTALACIONES CON FECHA DE ENTREGA
	NO IMPORTANTE	FALTA DE PERSONAL REUNIONES INTERRUPCIONES	REGISTROS DE INVENTARIOS EN EXCEL MALA ORGANIZACION LLAMADAS

Anexo 13

1. Matriz de decisión

Criterios orientadores	Proyecto 1: Creación de sistema de inventario a través de aplicación móvil	Proyecto 2: Estrategia para el aumento de accionistas	Proyecto 3: Contratación de personal capacitado	Proyecto 4: Capacitación del personal en atención a clientes	Proyecto 5: Rediseño de estructura organizacional	Proyecto 6: Rediseño de distribución de carga de trabajo
Perdida de Materiales	1	0	1	0	1	1
Retardos al despachar los productos	1	0	0	0	1	1
Mejor manejo operativo de bodega	1	0	0	0	1	1
Falta de personal	0	1	1	0	1	1
Falta de Estabilidad económica	0	1	0	1	1	0
Mejor manejo de distribución de productos	1	0	0	0	1	0
Contar con más Clientes fijos	1	0	1	1	0	0
Se Compra productos a sobreprecios	1	0	0	0	1	1
Quejas de clientes	1	0	1	1	1	1
Mejor Reputación de la empresa	1	0	1	1	1	1
Falta de innovación tecnológica	1	1	0	0	1	1
Rotación adecuada del inventario	1	0	1	0	1	1
TOTAL	10	3	6	4	11	9

Anexo 14

Selección de dos conceptos a prototipar.

De acuerdo con la matriz de decisiones que está en el paso anterior se pudo obtener los dos proyectos para prototipar, se los escogió de acuerdo con su puntuación total las cuales fueron las dos más altas, a continuación, se indicara cuáles fueron los proyectos a prototipar:

1. Proyecto 5: Rediseño de estructura organizacional
2. Proyecto 1: Creación de sistema de inventario a través de aplicación móvil

Anexo 15

Prototipo

**SOLDEINSA** cia Ltda
SOLUCIONES EN PROYECTOS DE INGENIERÍA MECÁNICA



Bienvenido al aplicativo móvil de la compañía Soldeinsa, esta aplicación es solo para empleados de la empresa, si eres empleado dale siguiente.

Siguiente

**SOLDEINSA** cia Ltda
SOLUCIONES EN PROYECTOS DE INGENIERÍA MECÁNICA



USUARIO:

CONTRASEÑA:

ACEPTAR



Marvel 13:00

Agregar Producto

Nombre
Type something here

Descripción
Type something here

Código de barra
Type something here

Cantidad del stock
Type something here

Unidad de ventas
Type something here

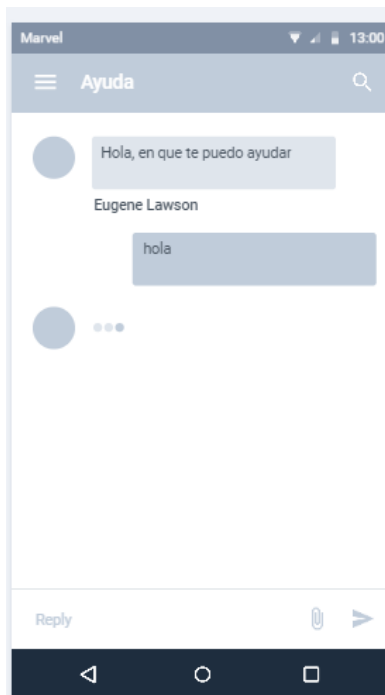
Cancelar Guardar

Marvel 13:00

J James Cameron inventario

Va a descargar el
el inventario de la empresa
que pesa:
50 MB

DISAGREE AGREE



Anexo 16

Configuracion de los archivos .java en Android Studio

Actualizar inventario

```
public class ActualizarInventario extends AppCompatActivity {

    Button btn16;

    private FirebaseDatabase mDataBase;
    private RadioGroup radioGroup;
    private AutoCompleteTextView autoCompleteTextView;
    private Button button;
    private Inventario inventario;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_actualizar_inventario);

        mDataBase = FirebaseDatabase.getInstance();
        radioGroup = findViewById(R.id.rdogActualizarInv);
        autoCompleteTextView = findViewById(R.id.autoCompleteTextView);
        button = findViewById(R.id.button15);

        //BOTON ATRAS
        Button btn16 = (Button) findViewById(R.id.button16);
        btn16.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    int selectedId = radioGroup.getCheckedRadioButtonId();

                    String filtro = "";
                    String value = autoCompleteTextView.getText().toString();

                    switch (selectedId) {
                        case R.id.radioButton4:
                            filtro = "nombre";
                            break;
                        case R.id.radioButton3:
                            filtro = "codigo";
                            break;
                    }

                    final DatabaseReference databaseReference =
```

```

mDataBase.getReference("inventario");
    Query query =
databaseReference.orderByChild(filtro).equalTo(value);

        query.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {
                inventario =
dataSnapshot.getValue(Inventario.class);
                Log.v("RESPUESTA DB", inventario.toString());
                Intent intent = new
Intent(getApplicationContext(), AgregarInventario.class);
                intent.putExtra("MODEL", inventario);
                intent.putExtra("KEY", dataSnapshot.getKey());
                startActivity(intent);
                finish();
            }

            @Override
            public void onChildChanged(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {

            }

            @Override
            public void onChildRemoved(@NonNull DataSnapshot
dataSnapshot) {

            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {

            }

            @Override
            public void onCancelled(@NonNull DatabaseError
databaseError) {
                Log.e("RESPUESTA DB", databaseError.toString());
                Toast.makeText(ActualizarInventario.this,
databaseError.toString(),
                                Toast.LENGTH_SHORT).show();
            }
        });

    } catch (Exception e) {
        Log.e("RESPUESTA DB", e.getMessage());
        Toast.makeText(ActualizarInventario.this, e.getMessage(),
            Toast.LENGTH_SHORT).show();
    }
}
});
}
}

```

Actualizar Ordenes

```
public class ActualizarOrden extends AppCompatActivity {

    private FirebaseDatabase mDataBase;

    Button btn20, btnBuscar;
    private EditText etCodigo;
    private Orden orden;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_actualizar_orden);

        mDataBase = FirebaseDatabase.getInstance();

        etCodigo = findViewById(R.id.editText6);
        btnBuscar = findViewById(R.id.button21);

        btnBuscar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    String value = etCodigo.getText().toString();
                    final DatabaseReference databaseReference =
mDataBase.getReference("orden");
                    Query query =
databaseReference.orderByChild("codigo").equalTo(value);

                    query.addChildEventListener(new ChildEventListener() {
                        @Override
                        public void onChildAdded(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {
                            orden = dataSnapshot.getValue(Orden.class);
                            Log.v("RESPUESTA DB", orden.toString());
                            Intent intent = new
Intent(getApplicationContext(), AumentoOrdenes.class);
                            intent.putExtra("MODEL", orden);
                            intent.putExtra("KEY", dataSnapshot.getKey());
                            startActivity(intent);
                            finish();
                        }

                        @Override
                        public void onChildChanged(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {

                        }

                        @Override
                        public void onChildRemoved(@NonNull DataSnapshot
dataSnapshot) {

                        }

                        @Override
                        public void onChildMoved(@NonNull DataSnapshot
```

```

dataSnapshot, @Nullable String s) {

    }

    @Override
    public void onCancelled(@NonNull DatabaseError
databaseError) {
        Log.e("RESPUESTA DB", databaseError.toString());
        Toast.makeText(ActualizarOrden.this,
databaseError.toString(),
                        Toast.LENGTH_SHORT).show();
    }
});

} catch (Exception e) {
    Toast.makeText(ActualizarOrden.this, e.getMessage(),
        Toast.LENGTH_SHORT).show();
}
}
});

//BOTON ATRAS
Button btn20 = (Button) findViewById(R.id.button20);
btn20.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

}

}

```

Agregar Inventario

```

public class AgregarInventario extends AppCompatActivity {

    Button btn13;
    Button Ginventario;
    private Spinner spinner1;
    EditText Edtxnombre, Edtxdescripcion, Edtxcodigobarra, Edtxcantidad;

    private DatabaseReference mDatabase;
    private Inventario inventario;
    private String mode;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_agregar_inventario);

        Edtxnombre = findViewById(R.id.Nombre);
        Edtxdescripcion = findViewById(R.id.Descripcion);
        Edtxcodigobarra = findViewById(R.id.Codigobarra);
    }
}

```

```

    Edtxcantidad = findViewById(R.id.Cantidad);

    mDatabase = FirebaseDatabase.getInstance().getReference();

    // MENU SPINNER
    spinner1 = (Spinner) findViewById(R.id.spinner2);
    String [] opciones = {"Unidad", "Decena", "Centena"};
    ArrayAdapter <String> adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, opciones);
    spinner1.setAdapter(adapter);

    inventario = getIntent().getParcelableExtra("MODEL");
    if (inventario != null) {
        mode = "UPDATE";
        Edtxnombre.setText(inventario.getNombre());
        Edtxdescripcion.setText(inventario.getDescripcion());
        Edtxcodigobarra.setText(inventario.getCodigo());
        Edtxcantidad.setText(String.valueOf(inventario.getCantidad()));
    } else {
        mode = "CREATE";
    }

    //BOTON DE ATRAS
    Button btn13 = (Button) findViewById(R.id.button13);
    btn13.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });

    // GUARDAR DATOS DEL INVENTARIO AGREGADO
    Ginventario = findViewById(R.id.button14);

    Ginventario.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            try {
                String nombre = Edtxnombre.getText().toString();
                String descripcion =
    Edtxdescripcion.getText().toString();
                String codigobarra =
    Edtxcodigobarra.getText().toString();
                int cantidad =
    Integer.parseInt(Edtxcantidad.getText().toString());
                String unidad = spinner1.getSelectedItem().toString();

                if (mode.equals("UPDATE")) {
                    String key = getIntent().getStringExtra("KEY");
                    inventario.setNombre(nombre);
                    inventario.setCodigo(codigobarra);
                    inventario.setDescripcion(descripcion);
                    inventario.setUnidad(unidad);
                    inventario.setCantidad(cantidad);

    mDatabase.child(String.format("inventario/%s",key)).setValue(inventario);

                } else {
                    inventario = new Inventario(nombre, descripcion,

```

```

codigobarra, unidad,
                                cantidad);

mDatabase.child("inventario").push().setValue(inventario);
    }

    Edtxnombre.setText("");
    Edtxdescripcion.setText("");
    Edtxcodigobarra.setText("");
    Edtxcantidad.setText("");
    spinner1.setSelection(0);

    Toast.makeText(AgregarInventario.this, "Inventario
almacenado correctamente",
                                Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        Toast.makeText(AgregarInventario.this, e.getMessage(),
                                Toast.LENGTH_SHORT).show();
    }

    }
    });
}
}

```

Aumento de ordenes

```

public class AumentoOrdenes extends AppCompatActivity {

    private EditText etCodOrden, etFecha, etCliente ,etLugar, etDesc,
etObservacion;
    private Button btnGuardarOrden;
    private Orden orden;
    private String mode;

    private DatabaseReference mDatabase;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_aumento_ordenes);

        mDatabase = FirebaseDatabase.getInstance().getReference();

        btnGuardarOrden = findViewById(R.id.btnGuardarOrden);

        etCodOrden = findViewById(R.id.etCodOrden);
        etFecha = findViewById(R.id.etFecha);
        etCliente = findViewById(R.id.etCliente);
        etLugar = findViewById(R.id.etLugar);
        etDesc = findViewById(R.id.etDescripcion);
        etObservacion = findViewById(R.id.etObservacion);

        orden = getIntent().getParcelableExtra("MODEL");
        if (orden != null) {
            mode = "UPDATE";
            etCodOrden.setText(orden.getCodigo());
            etFecha.setText(orden.getFecha());

```

```

        etCliente.setText(orden.getCliente());
        etLugar.setText(orden.getLugar());
        etDesc.setText(orden.getDescripcion());
        etObservacion.setText(orden.getObservacion());
    } else {
        mode = "CREATE";
    }

    btnGuardarOrden.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            try {
                String codOrden = etCodOrden.getText().toString();
                String fecha = etFecha.getText().toString();
                String cliente = etCliente.getText().toString();
                String lugar = etLugar.getText().toString();
                String descripcion = etDesc.getText().toString();
                String observacion = etObservacion.getText().toString();

                if (mode.equals("UPDATE")) {
                    String key = getIntent().getStringExtra("KEY");
                    orden.setCodigo(codOrden);
                    orden.setFecha(fecha);
                    orden.setCliente(cliente);
                    orden.setLugar(lugar);
                    orden.setDescripcion(descripcion);
                    orden.setObservacion(observacion);

                    mDatabase.child(String.format("orden/%s", key)).setValue(orden);
                } else {
                    orden = new Orden(codOrden, fecha, cliente, lugar,
descripcion, observacion);
                    mDatabase.child("orden").push().setValue(orden);
                }

                etCodOrden.setText("");
                etFecha.setText("");
                etCliente.setText("");
                etLugar.setText("");
                etDesc.setText("");
                etObservacion.setText("");

                Toast.makeText(AumentoOrdenes.this, "Inventario
almacenado correctamente",
                    Toast.LENGTH_SHORT).show();

            } catch (Exception e) {
                Toast.makeText(AumentoOrdenes.this, e.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        }
    });

    // BOTON Atras
    Button btn50 = (Button) findViewById(R.id.button50);
    btn50.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });

```



```

    });
}
}

```

Login usuarios

```

public class LoginUsuarios extends AppCompatActivity {

    Button btn2;
    Button btn3;
    private FirebaseAuth mAuth;
    private FirebaseUser user;

    private EditText etUsuario, etClave;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_usuarios);

        etUsuario = findViewById(R.id.editex_loginusuario);
        etClave = findViewById(R.id.editex_loginpassword);

        // BOTON DE SIGUIENTE
        final Button btn2 = (Button) findViewById(R.id.button2);
        btn2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = etUsuario.getText().toString();
                String password = etClave.getText().toString();
                try {
                    mAuth.signInWithEmailAndPassword(email, password)
                        .addOnCompleteListener(LoginUsuarios.this, new
OnCompleteListener<AuthResult>() {
                            @Override
                            public void onComplete(@NonNull
Task<AuthResult> task) {

                                if (task.isSuccessful()) {
                                    user = mAuth.getCurrentUser();
                                    FirebaseDatabase database =
FirebaseDatabase.getInstance();
                                    DatabaseReference myRef =
database.getReference(String.format("usuarios/%s", user.getId()));
                                    myRef.addValueEventListener(new
ValueEventListener() {

                                        @Override
                                        public void

onDataChange(DataSnapshot dataSnapshot) {

                                            // This method is called once
                                            // whenever data at this
                                            //String value =
                                            String value =
                                            Log.v("RESPUESTA DB", value);
                                            redirigir(value);
                                        }

                                        @Override
                                        public void

```

```

onCancelled(DatabaseError error) {
    // Failed to read value

    Toast.makeText(LoginUsuarios.this, "No se puede leer",
    Toast.LENGTH_SHORT).show();
    });

    } else {
        Toast.makeText(LoginUsuarios.this,
        "Error de autenticacion",
        Toast.LENGTH_SHORT).show();
    }
    });
});

} catch (Exception e) {
    Toast.makeText(LoginUsuarios.this, e.getMessage(),
    Toast.LENGTH_SHORT).show();
}

}
});

//BOTON DE ATRAS
Button btn3 = (Button) findViewById(R.id.button3);
btn3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent3 = new Intent (v.getContext(),
MainActivity.class);
        startActivityForResult(intent3, 0);
    }
});

// Initialize Firebase Auth
mAuth = FirebaseAuth.getInstance();
user = mAuth.getCurrentUser();
if (user != null) {
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef =
database.getReference(String.format("usuarios/%s", user.getId()));
    myRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            // This method is called once with the initial value and
again
            // whenever data at this location is updated.
            String value = dataSnapshot.getValue(String.class);
            Log.v("RESPUESTA DB", value);
            redirigir(value);
        }

        @Override
        public void onCancelled(DatabaseError error) {
            // Failed to read value
            Toast.makeText(LoginUsuarios.this, "No se puede leer",

```

```

        Toast.LENGTH_SHORT).show();
    }
    });
}

}

private void redirigir(String rol) {
    Intent intent2;
    switch (rol) {
        case "JEFE DE PROYECTO":
            intent2 = new Intent (getApplicationContext(),
JproyectoMenu.class);
            break;
        case "GERENTE GENERAL":
            intent2 = new Intent (getApplicationContext(),
GgeneralMenu.class);
            break;
        case "TECNICO":
            intent2 = new Intent (getApplicationContext(),
TecnicoMenu.class);
            break;
        case "BODEGA":
            intent2 = new Intent (getApplicationContext(),
BodegaMenu.class);
            break;
        default:
            intent2 = new Intent (getApplicationContext(),
MainActivity.class);
    }
    startActivityForResult(intent2, 0);
}
}

```

Ver Inventario

```

public class VerInventario extends AppCompatActivity {

    Button btn12;
    private RadioGroup radioGroup;
    private RadioButton radioButton;

    private FirebaseDatabase mDataBase;
    private AutoCompleteTextView autoCompleteTextView;
    private TextView textView;

    private Inventario inventario;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ver_inventario);

        mDataBase = FirebaseDatabase.getInstance();

        radioGroup = findViewById(R.id.radioGroup);
        autoCompleteTextView = findViewById(R.id.autoCompleteTextView2);
        textView = findViewById(R.id.lista);

        //BOTON DE ATRAS
        Button btn12 = (Button) findViewById(R.id.button12);
    }
}

```

```

btn12.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

// BOTON BUSCAR
Button btn11 = (Button) findViewById(R.id.button11);
btn11.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            int selectedId = radioGroup.getCheckedRadioButtonId();

            String filtro = "";
            String value = autoCompleteTextView.getText().toString();

            switch (selectedId) {
                case R.id.radioButton2:
                    filtro = "nombre";
                    break;
                case R.id.radioButton:
                    filtro = "codigo";
                    break;
            }

            DatabaseReference databaseReference =
mDataBase.getReference("inventario");
            Query query =
            databaseReference.orderByChild(filtro).equalTo(value);

            query.addChildEventListener(new ChildEventListener() {
                @Override
                public void onChildAdded(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {
                    Log.e("RESPUESTA DB", dataSnapshot.toString());
                    inventario =
dataSnapshot.getValue(Inventario.class);
                    textView.setText(inventario.toString());
                }

                @Override
                public void onChildChanged(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {
                }

                @Override
                public void onChildRemoved(@NonNull DataSnapshot
dataSnapshot) {
                }

                @Override
                public void onChildMoved(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {
                }

                @Override

```

```

        public void onCancelled(@NonNull DatabaseError
databaseError) {
            Log.e("RESPUESTA DB", databaseError.toString());
            Toast.makeText(VerInventario.this,
databaseError.toString(),
                        Toast.LENGTH_SHORT).show();
        }
    });

    } catch (Exception e) {
        Log.e("RESPUESTA DB", e.getMessage());
        Toast.makeText(VerInventario.this, e.getMessage(),
            Toast.LENGTH_SHORT).show();
    }
}

});
}
}
}

```

Ver orden

```

public class VerOrden extends AppCompatActivity {

    Button btn19, btnBuscar;
    private TextView tvOrden;
    private EditText etCodigo;
    private FirebaseDatabase mDataBase;

    private Orden orden;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ver_orden);

        mDataBase = FirebaseDatabase.getInstance();

        btnBuscar = findViewById(R.id.btnBuscarOrden);
        tvOrden = findViewById(R.id.tvOrden);
        etCodigo= findViewById(R.id.editText5);

        //BOTON ATRAS
        Button btn19 = (Button) findViewById(R.id.button19);
        btn19.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        Log.d("RESPUESTA DB", "VER ORDEN");
        btnBuscar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    Log.d("RESPUESTA DB", "CLICK");
                    String value = etCodigo.getText().toString();

```

```

        DatabaseReference databaseReference =
mDataBase.getReference("orden");
        Query query =
databaseReference.orderByChild("codigo").equalTo(value);
        query.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {
                Log.d("RESPUESTA DB", dataSnapshot.toString());
                orden = dataSnapshot.getValue(Orden.class);
                tvOrden.setText(orden.toString());
            }

            @Override
            public void onChildChanged(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {
            }

            @Override
            public void onChildRemoved(@NonNull DataSnapshot
dataSnapshot) {
            }

            @Override
            public void onChildMoved(@NonNull DataSnapshot
dataSnapshot, @Nullable String s) {
            }

            @Override
            public void onCancelled(@NonNull DatabaseError
databaseError) {
                Log.e("RESPUESTA DB", databaseError.toString());
                Toast.makeText(VerOrden.this,
databaseError.toString(),
                    Toast.LENGTH_SHORT).show();
            }
        });
    } catch (Exception e) {
        Toast.makeText(VerOrden.this, e.getMessage(),
            Toast.LENGTH_SHORT).show();
    }
}
});
}
}

```

Anexo 17

Archivo de configuración para agregar librerías

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.soldeinsa.httpswww.soldeinsa"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
        "android.support.test.runner.AndroidJUnitRunner"
        vectorDrawables.useSupportLibrary = true
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'),
            'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.android.support:support-v4:28.0.0'
    implementation 'com.android.support:support-vector-drawable:28.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-
core:3.0.2'
    implementation 'com.google.firebase:firebase-core:16.0.6'
    implementation 'com.google.firebase:firebase-database:16.0.5'
    implementation 'com.google.firebase:firebase-auth:16.1.0'
    implementation 'com.google.firebase:firebase-database:16.0.5'
}

apply plugin: 'com.google.gms.google-services'
```

Anexo 18

Archivos de configuración para órdenes e inventario

Ordenes

```
public class Orden implements Parcelable {

    String codigo, fecha, cliente, lugar, descripcion, observacion;

    public Orden() {}

    public Orden(String codigo, String fecha, String cliente, String lugar,
String descripcion,
        String observacion) {
        this.codigo = codigo;
        this.fecha = fecha;
        this.cliente = cliente;
        this.lugar = lugar;
        this.descripcion = descripcion;
        this.observacion = observacion;
    }

    protected Orden(Parcel in) {
        codigo = in.readString();
        fecha = in.readString();
        cliente = in.readString();
        lugar = in.readString();
        descripcion = in.readString();
        observacion = in.readString();
    }

    public static final Creator<Orden> CREATOR = new Creator<Orden>() {
        @Override
        public Orden createFromParcel(Parcel in) {
            return new Orden(in);
        }

        @Override
        public Orden[] newArray(int size) {
            return new Orden[size];
        }
    };

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(codigo);
        dest.writeString(fecha);
        dest.writeString(cliente);
        dest.writeString(lugar);
    }
}
```



```

        dest.writeString(descripcion);
        dest.writeString(observacion);
    }

    public String getCodigo() {
        return codigo;
    }

    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }

    public String getFecha() {
        return fecha;
    }

    public void setFecha(String fecha) {
        this.fecha = fecha;
    }

    public String getCliente() {
        return cliente;
    }

    public void setCliente(String cliente) {
        this.cliente = cliente;
    }

    public String getLugar() {
        return lugar;
    }

    public void setLugar(String lugar) {
        this.lugar = lugar;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getObservacion() {
        return observacion;
    }

    public void setObservacion(String observacion) {
        this.observacion = observacion;
    }

    @Override
    public String toString() {
        return "codigo: " + codigo + '\n' +
            "fecha: " + fecha + '\n' +
            "cliente: " + cliente + '\n' +
            "lugar: " + lugar + '\n' +
            "descripcion: " + descripcion + '\n' +
            "observacion: " + observacion;
    }

```

```
}  
}
```

Inventario

```
public class Inventario implements Parcelable {  
  
    private String nombre, descripcion, codigo, unidad;  
    private int cantidad;  
  
    public Inventario() {}  
  
    public Inventario(String nombre, String descripcion, String codigo,  
String unidad, int cantidad) {  
        this.nombre = nombre;  
        this.descripcion = descripcion;  
        this.codigo = codigo;  
        this.unidad = unidad;  
        this.cantidad = cantidad;  
    }  
  
    protected Inventario(Parcel in) {  
        nombre = in.readString();  
        descripcion = in.readString();  
        codigo = in.readString();  
        unidad = in.readString();  
        cantidad = in.readInt();  
    }  
  
    public static final Creator<Inventario> CREATOR = new  
Creator<Inventario>() {  
        @Override  
        public Inventario createFromParcel(Parcel in) {  
            return new Inventario(in);  
        }  
  
        @Override  
        public Inventario[] newArray(int size) {  
            return new Inventario[size];  
        }  
    };  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getDescripcion() {  
        return descripcion;  
    }  
  
    public void setDescripcion(String descripcion) {  
        this.descripcion = descripcion;  
    }  
  
    public String getCodigo() {  
        return codigo;  
    }  
}
```

```

    }

    public void setCodigo(String codigo) {
        this.codigo = codigo;
    }

    public String getUnidad() {
        return unidad;
    }

    public void setUnidad(String unidad) {
        this.unidad = unidad;
    }

    public int getCantidad() {
        return cantidad;
    }

    public void setCantidad(int cantidad) {
        this.cantidad = cantidad;
    }

    @Override
    public String toString() {
        return "Nombre: " + nombre + '\n' +
            "Descripcion: " + descripcion + '\n' +
            "Codigo: " + codigo + '\n' +
            "Unidad: " + unidad + '\n' +
            "Cantidad: " + cantidad;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(nombre);
        dest.writeString(descripcion);
        dest.writeString(codigo);
        dest.writeString(unidad);
        dest.writeInt(cantidad);
    }
}

```