UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA

**FACULTAD DE INGENIERIA EN INFORMATICA Y SISTEMAS**



**INFORME FINAL DE PRÁCTICAS PRE-PROFESIONALES CONSTRUCCION DEL APLICATIVO MÓVIL DONCA ERP**

**Presentado por:** Manuel Alejandro Rojas Curitima

**Asesor:** Brian Cesar Pando Soto

**Periodo de Prácticas Pre Profesionales:** 15/07/2022 – 15/10/2022

**Lugar de las Prácticas Pre Profesionales***:* Lead Working Partner SA

**TINGO MARIA – Octubre 2022**

2



*“Año del Fortalecimiento de la Soberanía Nacional”*

**DICTAMEN Nº 005-BP-2022**

**PARA : Ing. Edwin Jesús Vega Ventocilla**

**Presidente de la comisión de prácticas pre profesionales**

**DE : BRIAN CESAR SOTO PANDO**

**Asesor de Manuel Alejandro Rojas Curitima**

**FECHA : 17-10-2022**

**ASUNTO : Conformidad de redacción de Informe PPP**

**--------------------------------------------------------------------------------------------------**

De mi especial consideración:

Habiendo revisado y validado el Informe Final de **Manuel Alejandro Rojas Curitima** titulado

**“CONSTRUCCIÓN DEL APICATIVO MOVIL ERP EN LA EMPRESA LWP S.A.C”**

Para tal efecto emito el dictamen con opinión **FAVORABLE** para ser aceptado por la Comisión de prácticas Pre Profesionales, continuando con el trámite correspondiente.

Sin otro particular, me suscribo de usted.

Cordialmente,





**“Año del Fortalecimiento de la Soberanía Nacional”**

**CERTIFICADO DE PRÁCTICAS PRE-PROFESIONALES**

3

Mediante la presente se CERTIFICA que:

**MANUEL ALEJANDRO ROJAS CURITIMA**

Realizó sus prácticas pre profesionales satisfactoriamente en nuestra empresa, desempeñándose como PRACTICANTE DE CONSTRUCCION DEL APLICATIVO MOVIL Donca ERP en nuestra área de desarrollo de software, desde el 15/07/2022 hasta el 15/10/2022, habiendo cumplido satisfactoriamente las siguientes actividades:

* Análisis de documentación del código fuente del Backend del aplicativo web.
* Análisis de buenas prácticas del desarrollo de software
* Análisis de la documentación de la tecnología usada para la construcción del aplicativo móvil.
* Codificación de los módulos presentados por parte de la empresa.
* Participante de las reuniones semanales sobre los avances del proyecto.

Se expide el presente documento a solicitud del interesado.

*Tingo Maria,17 de octubre del 2022*

*ATENTAMENTE*



**DEDICATORIA**

Al Señor nuestro Dios, por darme la vida y las fuerzas para poder continuar con mis sueños, protagonista de este proyecto.

A mis madres Susy Curitima y Lady Curitima, pilares de mi crecimiento personal y profesional.

A mis hermanos Adriel y Kevin por su apoyo moral y por preocupación.

# AGRADECIMIENTO

A nuestro señor Jesucristo, por bendecirme y guiar mis pasos, brindándome las fuerzas necesarias para cumplir mis metas.

Al equipo de lwp, quienes influyeron en mi crecimiento profesional al darme la oportunidad de trabajar con ellos.

A mi asesor Brian Cesar Pando Soto, por su disposición y asesoramiento en el correcto desarrollo de mis Practicas Pre Profesionales.

# RESUMEN

Lead Working Partner S.A, es una empresa desarrolladora de soluciones y sistemas de software, utilizando tecnologías de alto nivel, también cuenta con un producto de software ERP, este es una aplicación web que permite a las mypes y pymes administrar sus procesos como son de venta, compra, almacenes, manejo de stocks, reportes entre otras cosas.

Actualmente la empresa tiene demanda de su ERP, para conseguir una mejor accesibilidad y experiencia de sus clientes busca desarrollar un aplicativo móvil que permita la conexión a su sistema erp, desde los equipos móviles de sus clientes, para que puedan realizar pedidos, ventas, apertura y cierre de cajas, registro de marcas y categorías, sin la necesidad de utilizar una laptop o computador.

La presente practica preprofesional tiene como objetivo construir el aplicativo móvil, usando previamente los prototipos con las que cuenta la empresa, para ello se utilizará la metodología ágil Kanban, que permitirá organizar el trabajo de construcción de acuerdo con las actividades propuestas.

**ÍNDICE GENERAL**

[AGRADECIMIENTO 5](#_bookmark0)

[RESUMEN 6](#_bookmark1)

[INTRODUCCIÓN 11](#_bookmark2)

[CAPITULO I 12](#_bookmark3)

[CARACTERISTICAS GENERALES DE LA ORGANIZACIÓN 12](#_bookmark4)

* 1. [Descripción de la Organización 12](#_bookmark5)
  2. [Misión y Visión 12](#_bookmark6)
  3. [Organigrama estructural 12](#_bookmark7)

[CAPITULO II 13](#_bookmark8)

[MARCO TEORICO 13](#_bookmark9)

* 1. [Ingeniería de Software 13](#_bookmark10)
  2. [Patrones de Arquitectura de software 13](#_bookmark11)
     1. [Patrón de arquitectura en N capas 13](#_bookmark12)
  3. [Metodología de desarrollo ágil 14](#_bookmark13)
     1. [Metodología Kanban 14](#_bookmark14)
  4. [Sistema de Gestión Comercial 15](#_bookmark16)
     1. [ERP 15](#_bookmark17)
  5. [Interfaz de programación de aplicaciones (API) 16](#_bookmark18)
  6. [Transferencia del estado de representación (REST) 16](#_bookmark19)
  7. [Backend 17](#_bookmark21)
  8. [Frontend 17](#_bookmark22)
  9. [Java 17](#_bookmark23)
  10. [Spring Framework 17](#_bookmark24)
  11. [PostgreSQL 18](#_bookmark25)
  12. [Dart 18](#_bookmark26)
  13. [Flutter 18](#_bookmark27)
  14. [Gestión de Versiones de código fuente 18](#_bookmark28)

[CAPITULO III 19](#_bookmark29)

[ACTIVIDADES DESARROLLADAS 19](#_bookmark30)

* 1. [Situación Actual 19](#_bookmark31)
     1. [Problemática 19](#_bookmark32)
  2. [Objetivos 19](#_bookmark33)
  3. [Alcance 20](#_bookmark34)
  4. [Cronograma de actividades 20](#_bookmark35)
  5. [Escenario de trabajo 20](#_bookmark36)
     1. [Dirección del proyecto 20](#_bookmark37)
     2. [Equipo de trabajo 21](#_bookmark38)
     3. [Recursos 21](#_bookmark39)
  6. [Descripción de la arquitectura del proyecto 22](#_bookmark40)
  7. [Método de trabajo 26](#_bookmark41)
  8. [Módulo de Configuraciones 27](#_bookmark43)
     1. [Requisitos del módulo de Configuraciones 27](#_bookmark44)
     2. [Prototipos del módulo Configuraciones 28](#_bookmark45)
     3. [Construcción del módulo de Configuraciones 28](#_bookmark46)
  9. [Modulo Inicio de Sesión 35](#_bookmark51)
     1. [Requisitos del módulo inicio de sesión 35](#_bookmark52)
     2. [Prototipos del módulo Inicio de sesión 35](#_bookmark53)
     3. [Construcción del módulo Inicio de sesión 37](#_bookmark55)
     4. [Pruebas del módulo Inicio de sesión 39](#_bookmark57)
  10. [Modulo Categoría 39](#_bookmark58)
      1. [Requisitos del módulo Categoría 40](#_bookmark59)
      2. [Prototipos del módulo Categoría 40](#_bookmark60)
      3. [Construcción del módulo Categoría 42](#_bookmark62)
  11. [Modulo Marcas 45](#_bookmark64)
      1. [Requisitos del módulo Marcas 46](#_bookmark65)
      2. [Prototipos del módulo Marcas 46](#_bookmark66)
      3. [Construcción del módulo Marcas 48](#_bookmark68)
  12. [Modulo Caja 51](#_bookmark71)
      1. [Requisitos del módulo Caja 51](#_bookmark72)
      2. [Prototipos del módulo Caja 51](#_bookmark73)
      3. [Construcción del módulo Caja 53](#_bookmark76)
  13. [Modulo Ventas 56](#_bookmark77)
      1. [Requisitos del módulo Ventas 56](#_bookmark78)
      2. [Prototipos del módulo Ventas 57](#_bookmark79)
      3. [Construcción del Ventas 61](#_bookmark81)
  14. [Modulo Pedidos 64](#_bookmark82)
      1. [Requisitos del módulo Pedidos 64](#_bookmark83)
      2. [Prototipos del módulo Pedidos 65](#_bookmark84)
      3. [Construcción del módulo Pedido 68](#_bookmark85)

[CONCLUSIONES 71](#_bookmark86)

[RECOMENDACIONES 72](#_bookmark87)

[BIBLIOGRAFÍA 73](#_bookmark88)

ÍNDICE DE FIGURAS

[**Figura 1:Organigrama Estructural LWP S.A.C** 12](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975684)

[**Figura 2: Arquitectura de N capas** 13](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975685)

[**Figura 3: Modelo del tablero Kanban** 15](#_bookmark15)

[**Figura 4: Flujo de trabajo de una API** 16](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975687)

[**Figura 5:Comunicacion REST** 17](#_bookmark20)

[**Figura 6: Arquitectura Frontend** 22](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975689)

[**Figura 7: Arquitectura Backend del proyecto** 23](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975690)

[**Figura 8: diagrama de arquitectura del proyecto** 24](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975691)

[**Figura 9:Distribución del Proyecto en el IDE** 24](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975692)

[**Figura 10:Distribución del Backend en el IDE** 25](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975693)

[**Figura 11: Distribución del aplicativo móvil en el IDE** 25](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975694)

[**Figura 12:Tablero Kanban en trello** 26](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975695)

[**Figura 13:Miembros participantes en el tablero Kanban** 26](#_bookmark42)

[**Figura 14:Flujo de subida de código al repositorio en bitbucket** 27](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975697)

[**Figura 15:Prototipo – Registro de Configuraciones** 28](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975698)

[**Figura 16:Flujo de trabajo para la construcción del módulo configuraciones** 29](#_bookmark47)

[**Figura 17:Fragmento de Código del contenedor del módulo configuraciones** 30](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975700)

[**Figura 18:Codificacion de los atributos, ruc, url, Código de acceso.** 31](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975701)

[**Figura 19:Implementación de Alertas.** 31](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975702)

[**Figura 20:Creacion de servicios que se conectaran al Backend.** 32](#_bookmark48)

[**Figura 21:Creacion del controlador en el Backend.** 33](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975704)

[**Figura 22: método de generación y validación de token** 33](#_bookmark49)

[**Figura 23:Implementacion de la capa de servicio para el registro de dispositivo** 34](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975706)

[**Figura 24: Capa de repositorio para el registro del dispositivo** 34](#_bookmark50)

[**Figura 25:Prototipo del módulo inicio de sesión** 35](#_bookmark54)

[**Figura 26:Prototipo del menú.** 36](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975709)

[**Figura 27: Flujo de trabajo para la construcción del módulo inicio de sesión** 37](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975710)

[**Figura 28: Fragmento de código de la vista Login.** 37](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975711)

[**Figura 29: Fragmento de Código del servicio del login que conecta al Backend de la aplicación.** 38](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975712)

[**Figura 30:Fragmento de Código del controlador del Backend, que recibe la solicitud de login** 38](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975713)

[**Figura 31: Fragmento de Código del servicio del Backend.** 39](#_bookmark56)

[**Figura 32:Fragmento de Código del repositorio del Backend** 39](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975715)

[**Figura 33:Prototipo del Módulo categoría - Listar categorías** 40](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975716)

[**Figura 34:Prototipo de Nueva categoría.** 41](#_bookmark61)

[**Figura 35:Prototipo de edición de categoría** 41](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975718)

[**Figura 36: Prototipo de la opción de eliminar la categoría** 42](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975719)

[**Figura 37: Flujo de trabajo para la construcción del módulo categoría.** 42](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975720)

[**Figura 38: Fragmento de código de listar categoría.** 43](#_bookmark63)

[**Figura 39:Fragmento de código de la opción de guardar categoría** 43](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975722)

[**Figura 40: Fragmento de código de la opción eliminar categoría** 44](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975723)

[**Figura 41:Fragmento de código del servicio categoría** 44](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975724)

[**Figura 42: Fragmento de código del controlador del Backend - categoría** 45](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975725)

[**Figura 43:Fragmento de código del servicio y repositorio del Backend- categoría** 45](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975726)

[**Figura 44: Prototipo listar marcas.** 46](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975727)

[**Figura 45:Prototipo- crear marca.** 47](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975728)

[**Figura 46:Prototipo- editar marca** 47](#_bookmark67)

[**Figura 47:Prototipo- eliminar marca** 48](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975730)

[**Figura 48:Flujo de trabajo para la construcción del módulo marca.** 48](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975731)

[**Figura 49: Fragmento de código- listar marcas** 49](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975732)

[**Figura 50: Fragmento de código del servicio- listar marca.** 49](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975733)

[**Figura 51:Fragmento de código del controlador del Backend-listar marca.** 50](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975734)

[**Figura 52: Fragmento de condigo del servicio del Backend- listar marca.** 50](#_bookmark69)

[**Figura 53:Fragmento de código del repositorio del Backend- listar marca.** 50](#_bookmark70)

[**Figura 54:Prototipo-listar cajas** 51](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975737)

[**Figura 55: Prototipo de apertura de caja.** 52](#_bookmark74)

[**Figura 56:Prototipo de cierre de caja.** 52](#_bookmark75)

[**Figura 57:Prototipo de eliminar caja.** 53](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975740)

[**Figura 58:Flujo de trabajo para la construcción del módulo caja.** 53](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975741)

[**Figura 59:Fragmento de código- listar cajas** 54](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975742)

[**Figura 60:Fragmento de código del servicio- listar cajas.** 54](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975743)

[**Figura 61: Fragmento de código del controlador del Backend- listar cajas.** 55](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975744)

[**Figura 62:Fragmento de código del servicio del Backend- listar cajas.** 55](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975745)

[**Figura 63:Fragmento de código del repositorio del Backend - listar cajas.** 55](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975746)

[**Figura 64:Prototipo- listar ventas.** 57](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975747)

[**Figura 65:Prototipo -selección de productos para una venta.** 57](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975748)

[**Figura 66:Prototipo lista de productos con imagen.** 58](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975749)

[**Figura 67:Prototipo- detalles de venta.** 58](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975750)

[**Figura 68: Prototipo de selección de cliente de una venta.** 59](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975751)

[**Figura 69:Prototipo-pagar una venta.** 59](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975752)

[**Figura 70:Prototipo-pagar una venta.** 59](#_bookmark80)

[**Figura 71:Prototipo- imprimir venta.** 60](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975754)

[**Figura 72:Prototipo de editar una venta.** 60](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975755)

[**Figura 73:Prototipo - Eliminar venta.** 61](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975756)

[**Figura 74:Flujo de trabajo para la construcción del módulo ventas.** 61](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975757)

[**Figura 75:Fragmento de código -lista de ventas.** 62](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975758)

[**Figura 76:Fragmento de código del servicio - listar ventas** 62](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975759)

[**Figura 77:Fragmento de código del controlador del Backend- listar ventas.** 63](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975760)

[**Figura 78:Fragmento de código del servicio del Backend - listar ventas** 63](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975761)

[**Figura 79:Fragmento de código del repositorio del Backend- listar ventas.** 63](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975762)

[**Figura 80:Prototipo - lista de pedidos.** 65](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975763)

[**Figura 81:Lista de productos a seleccionar.** 65](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975764)

[**Figura 82:Prototipo detalle de pedidos.** 66](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975765)

[**Figura 83: Prototipo de edición de pedidos** 66](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975766)

[**Figura 84:Prototipo de impresión de pedido.** 67](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975767)

[**Figura 85:Prototipo de eliminar pedido.** 67](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975768)

[**Figura 86: Flujo de trabajo para la construcción del módulo pedidos.** 68](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975769)

[**Figura 87:Fragmento de código -lista de pedidos** 68](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975770)

[**Figura 88: Fragmento de código del servicio - listar pedidos** 69](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975771)

[**Figura 89:Fragmento de código del controlador del Backend -listar pedidos.** 69](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975772)

[**Figura 90:Fragmento de código del servicio del Backend - listar pedidos** 70](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975773)

[**Figura 91:Fragmento de código del repositorio del Backend - listar pedidos** 70](https://unasedu-my.sharepoint.com/personal/manuel_rojas_unas_edu_pe/Documents/ppp/informe%20de%20PPP%20rojas_curitima_manuel_alejandro.docx#_Toc116975774)

# INTRODUCCIÓN

Cada año se crean nuevas empresas con diferentes innovaciones que compiten con las otras que ya llevan algún tiempo en el mercado, para ello la clave de supervivencia de una empresa es la constante actualización de los participantes con ayuda de la tecnología, donde se pueden reducir costos, automatizaciones, entre otras necesidades de la empresa, brindando una experiencia única y accesible a sus clientes.

Lead Working Partner S.A, busca construir un aplicativo móvil que comparta la funcionalidad de su aplicación web, para que sus clientes puedan acceder desde un dispositivo móvil y tener una experiencia mejorada. Para la cual en la presente practica preprofesional se construyó el aplicativo móvil.

**En el capítulo 1:** Se describe las características de la empresa Lead Working Partner SA, organización donde se desarrolló las practicas preprofesionales, que se encuentra en la ciudad de tingo maría. La asistencia fue de forma presencial en horarios de oficina.

**En el capítulo 2:** Se describe el marco teórico y conceptual que sirvió como base para el desarrollo y entendimiento de la practica preprofesional.

**En el capítulo 3:** Se explica el desarrollo de las actividades realizadas durante el periodo de duración de las practica preprofesional: construcción del aplicativo móvil.

Finalmente se detalla los resultados, conclusiones y anexos.

# CAPITULO I

# CARACTERISTICAS GENERALES DE LA ORGANIZACIÓN

## Descripción de la Organización

* + - **Nombre:** Lead Working Partner S.A.C
    - **Ubicación Geográfica** Departamento: Huánuco Provincia: Leoncio Prado Distrito: Rupa Rupa Dirección: Jr. Arequipa #1085
  1. **Misión y Visión Misión**

Ser una empresa líder en desarrollo y mantenimiento de software con la

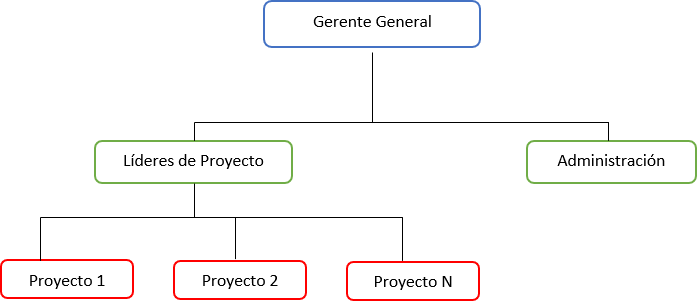
vanguardia de las últimas tecnologías, brindando al cliente productos más livianos.

**Visión**

Ser reconocidos por nuestro mercado, contar con productos maduros(software) para ofertar en el mercado mundial.

## Organigrama estructural

***Figura 1:Organigrama Estructural LWP S.A.C***



Fuente: Elaboración propia.

## Ingeniería de Software

**CAPITULO II MARCO TEORICO**

(Pressman, 2010) define la ingeniera de software como una disciplina de cuatro capas: herramientas, métodos, procesos y un enfoque de calidad. Se complementa con la definición en cinco áreas de conocimiento relacionada a la capa de procesos de desarrollo de software: requisitos, diseño, construcción, pruebas y mantenimiento de software, para obtener un producto que cumpla con los estándares de calidad y las expectativas del cliente (Ieee Computer Society, 2014).

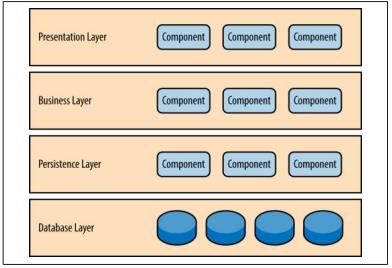
## Patrones de Arquitectura de software

Los patrones de arquitectura describen los diseños de los componentes que forman un sistema, es decir la estructura con el cual estará compuesta el sistema(Jacobson & Stimson, 2017).

### Patrón de arquitectura en N capas

Es el patrón más usado en los proyectos con tecnología Java EE, y muy conocido en la comunidad de arquitectos, diseñadores y desarrolladores de software(García- Holgado & García-Peñalvo, 2016).

Los componentes del patrón de arquitectura por capas se ordenan de manera horizontal. Además, no especifica el numero o tipos de capa que deben existir en este patrón. Generalmente están compuesto de cuatro capas como se muestra a continuación.



***Figura 2: Arquitectura de N capas***

Fuente: (Richards, 2015)

Como se muestra en la figura 2, la arquitectura en capas generalmente se compone de cuatro capas donde:

* + - * **Capa de Presentación**, es la encargada de manejar la interfaz de usuario y la lógica de comunicación con el navegador.
      * **Capa de Negocio**, es la encargada de ejecutar las reglas de negocio.
      * **Capa de Persistencia**, es la encargada de realizar las llamadas hacia la capa de base de datos para obtener la información solicitada.
      * **Capa de Base de datos**, es la encargada de almacenar la información.

Este patrón tiene como principio el aislamiento, es decir las capas se acceden de forma horizontal, por ejemplo, la capa de presentación no puede conectarse directamente con la capa de persistencia, para ello de pasar por la capa de negocio. El concepto de capas significa que los cambios realizados en una capa de la arquitectura generalmente no afectan a los componentes de las otras capas(Richards, 2015).

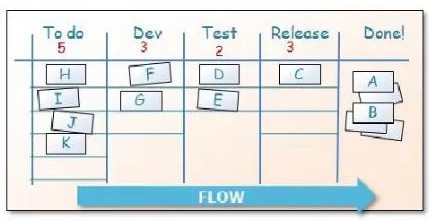
## Metodología de desarrollo ágil

El enfoque de esta metodología se basa en un modelo iterativo e incremental, los requisitos y las soluciones son producto de la colaboración entre los equipos multifuncionales(McCormick, 2012).

### Metodología Kanban

Es una metodología de desarrollo de software lean, que se centra en la entrega “Just inTime” de la funcionalidad. Esta metodología fue introducida en la industria de fabricación japonesa(Ahmad et al., 2013).

El tablero Kanban proporciona visibilidad al proceso de software, ya que muestra el trabajo asignado a cada desarrollador, y estos se centran solo en esos pocos elementos en un momento dado, esta metodología pretende adaptar rápidamente el proceso utilizando bucles de retroalimentación más cortos(Kniberg & Skarin, 2009).

***Figura 3: Modelo del tablero Kanban***

Fuente:(Kniberg & Skarin, 2009)

tabla 1.

Esta metodología se basa en los siguientes principios, como se muestran en la

***Tabla 1: Principios de la Metodología Kanban***

|  |  |
| --- | --- |
| **Principios del Desarrollo de software lean** | **Principios Kanban** |
| Eliminar los residuos | Visualice el flujo de trabajo |
| Incorporación de calidad | Limita el trabajo en curso |
| Crear conocimiento | Medir y gestionar el flujo |
| Aplazar el compromiso | Explicitar las políticas de los  procesos |
| Entrega rápida | Mejorar la colaboración |
| Respetar a las personas |
| Optimizar el conjunto |

Fuente:(Ahmad et al., 2013)

## Sistema de Gestión Comercial

Son sistemas que atienden los procesos comerciales de una empresa, facilitando el trabajo y la atención de los clientes.

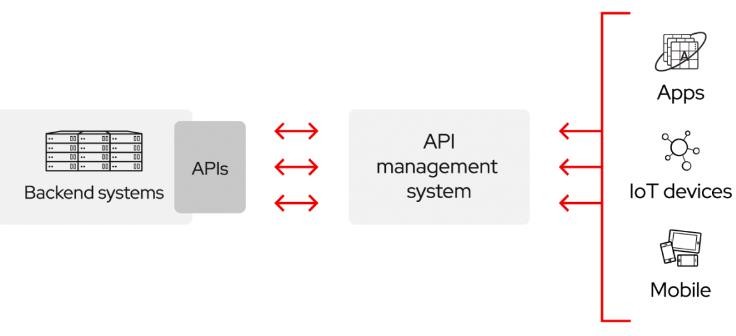
### ERP

Se refiere a un tipo de software que las organizaciones usan para administrar las actividades comerciales diarias, como la contabilidad, el rendimiento, etc.(Abugabah & Sanzogni, 2010).

## Interfaz de programación de aplicaciones (API)

Es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de aplicaciones.

Las Apis permiten que sus productos y servicios se comuniquen entre sí, sin necesidad de saber cómo están implementados. Las Apis son un medio simplificado para conectar su propia infraestructura a través del desarrollo de aplicaciones nativas de la nube, pero también permiten compartir los datos con clientes y usuarios externos como se muestra en la figura 4 (Bortenschlager et al., 2020).



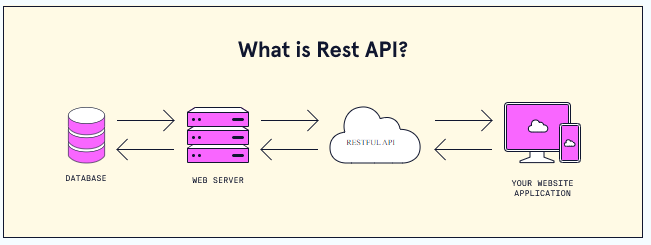
***Figura 4: Flujo de trabajo de una API***

Fuente: (Bortenschlager et al., 2020)

## Transferencia del estado de representación (REST)

Es un estilo arquitectónico para proporcionar estándares de comunicación entre sistemas como se muestra en la figura 5 (¿What Is REST? | Codecademy, n.d.).

***Figura 5:Comunicacion REST***



Fuente: (¿What Is REST? | Codecademy, n.d.)

## Backend

Es la parte que se ejecuta en un servidor, es el responsable de tratar los datos en términos general, esto significa que es responsable de recibir las peticiones de los clientes para que se les proporcione datos, para que puedan visualizarlos (Filipova & Vilão, 2018).

## Frontend

El desarrollo Frontend, también conocido como desarrollo del lado del cliente, es la práctica de producir HTML, CSS Y JavaScript para un sitio web o una aplicación web para que un usuario pueda verlos e interactuar directamente con ellos (Gerasimov et al., 2020).

## Java

Es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionaran a menos que tenga java instalado y cada día se crea más. Java es más rápido, seguro y fiable(Oracle, 1995).

## Spring Framework

Es un marco de trabajo de código libre que proporciona soporte de infraestructura para desarrollar aplicaciones Java. Spring, uno de los marcos de Java Enterprise Edition (Java EE) más populares con alto rendimiento utilizando objetos

simples en la programación orientada a objetos(*Spring | Projects*, n.d.).

## PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacionales de objetos (ORDBMS) basado en POSTGRES, desarrollado en el departamento de ciencias de la computación de la universidad de Berkeley (*PostgreSQL: Documentation: 14: PostgreSQL 14.4 Documentation*, n.d.).

## Dart

Dart es un lenguaje optimizado para aplicaciones rápidas en cualquier plataforma, esta soportado por Google, se creó como una alternita a java en cuanto a desarrollo móvil, pero poco a poco fue creciendo hasta ser multiplataforma, es decir el programador solo crea un aplicativo para distintas plataformas gracias a la interoperabilidad de Dart, además este soporta desarrollo nativo tanto para java o kotlin como Swift y Objective-C (*Dart Programming Language | Dart*, n.d.).

## Flutter

Flutter es un Frameworks de interfaz de usuario creado por Google, el lenguaje de programación con el que trabaja es Dart, consta de una colección de elementos de interfaces de usuario reutilizables, que puedes modificar(*Flutter - Build Apps for Any Screen*, n.d.).

## Gestión de Versiones de código fuente

El control de versiones, también conocido como “control de código fuente”, es la práctica de rastrear y gestionar los cambios en el código fuente del software (*Qué Es El Control de Versiones | Atlassian Git Tutorial*, n.d.).

## Situación Actual

### Problemática

**CAPITULO III ACTIVIDADES DESARROLLADAS**

La empresa “Lead Working Partner S.A.C”, tiene como uno de sus productos, un sistema de gestión comercial web llamado “DONCA ERP”. Debido a la acogida que tiene el sistema, la empresa quiere innovar presentando una nueva opción a sus clientes para que puedan acceder desde sus dispositivos móviles mediante una aplicación.

Específicamente la empresa buscaba desarrollar un aplicativo móvil basado en el Backend de la aplicación web ya existente, para ello previamente la empresa cuenta con los diseños y requisitos que se deben seguir para su construcción.

Para comenzar con la construcción del aplicativo móvil se propuso el uso de la metodología Kanban.

## Objetivos

* **General**

Construir el aplicativo móvil “Donca erp”

* **Específicos**

1. Construir el módulo de configuración de empresa.
2. Construir el módulo de inicio de sesión.
3. Construir el módulo ventas.
4. Construir el módulo pedido.
5. Construir el módulo caja.
6. Construir el módulo marcas.
7. Construir el módulo categorías.

## Alcance

Este trabajo está limitado al alcance que tendrán los módulos de:

* Configuraciones.
* Inicio de sesión.
* Ventas.
* Pedido.
* Caja
* Marcas.
* Categorías.

## Cronograma de actividades

***Tabla 2: Cronograma de actividades***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actividades** | **Julio** | **Agosto** | **Septiembre** | **Octubre** |
| Análisis de los requisitos presentados por la empresa, para la construcción del aplicativo móvil.   * Arquitectura del sistema * Comunicación entre Frontend y Backend * Arquitectura de la base de datos * Políticas de codificación |  |  |  |  |
| Construcción del módulo Configuraciones: |  |  |  |  |
| Construcción del módulo Inicio de Sesión. |  |  |  |  |
| Construcción del módulo Ventas. |  |  |  |  |
| Construcción del módulo Pedidos. |  |  |  |  |
| Construcción del módulo Caja. |  |  |  |  |
| Construcción del módulo Marca |  |  |  |  |
| Construcción del módulo Categoría. |  |  |  |  |

Fuente: Elaboración propia

## Escenario de trabajo

### Dirección del proyecto

El jefe del proyecto es Brian Pando Soto.

### Equipo de trabajo

A continuación, se presenta los miembros del equipo del proyecto erp:

***Tabla 3:Equipo de trabajo***

|  |  |
| --- | --- |
| **Líder del proyecto – Analista programador Senior** | |
| **Nombre** | Brian Pando Soto |
| **Categoría profesional** | Ingeniero en sistemas - analista programador sénior – QA sénior –  DevOps sénior |
| **Contacto** | [Brian.pando@unas.edu.pe](mailto:Brian.pando@unas.edu.pe) |
| **Analista programador sénior** | |
| **Nombre** | Jhil Palacios Mariano |
| **Categoría profesional** | Analista programador sénior |
| **Contacto** | [Jhil.palacios@unas.edu.pe](mailto:Jhil.palacios@unas.edu.pe) |
| **Analista programador** | |
| **Nombre** | Manuel Rojas Curitima |
| **Categoría profesional** | Analista programador - Practicante |
| **Contacto** | [manuel.rojas@unas.edu.pe](mailto:manuel.rojas@unas.edu.pe) |

Fuente: Elaboración propia.

### Recursos

Durante el tiempo de desarrollo de las practicas preprofesionales se usaron los siguientes recursos:

* + - * **Eclipse Java EE IDE.** Este es un entorno de desarrollo integrado utilizado en la programación, mayormente utilizado para proyectos basado en Java.
      * **DBeaver.** este es un software que se usa para la administración de base de datos.
      * **Visual Studio Code.** Este es un editor de código fuente.
      * **Android studio.** Es un entorno de desarrollo para aplicaciones móviles creado por Google.

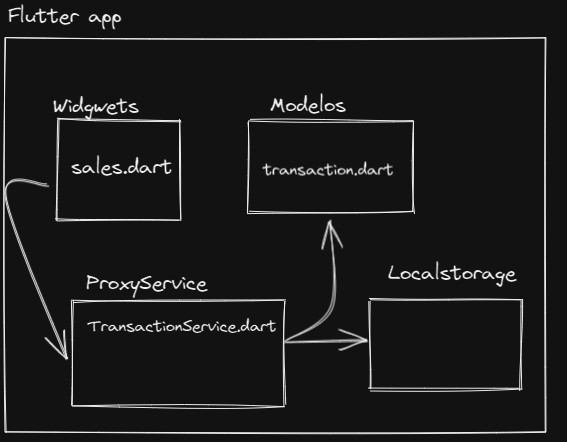
## Descripción de la arquitectura del proyecto

El proyecto este compuesto de un Frontend y un Backend a continuación se presenta un diagrama:

* **Frontend**

La figura 6 muestra la arquitectura Frontend del proyecto basado en el modelo de capas, para ello se usó la tecnología de Flutter, donde los widgets vienen a ser la capa de presentación, los modelos vienen a ser la capa de dominio y el proxyservice y el localstorage vienen a ser la capa de datos.

***Figura 6: Arquitectura Frontend***



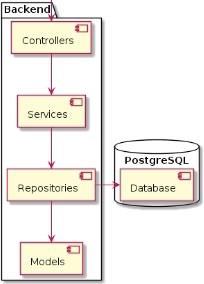
Fuente: Lead Working Partner S.A

Los archivos tendrán una extensión. Dart, ya que Flutter trabaja con el lenguaje de programación Dart, a su vez los proxyservice interactúan con el Backend, mediante una conexión REST.

* **Backend**

La figura 7 muestra la arquitectura Backend del proyecto, este compuesto por una arquitectura en capas.

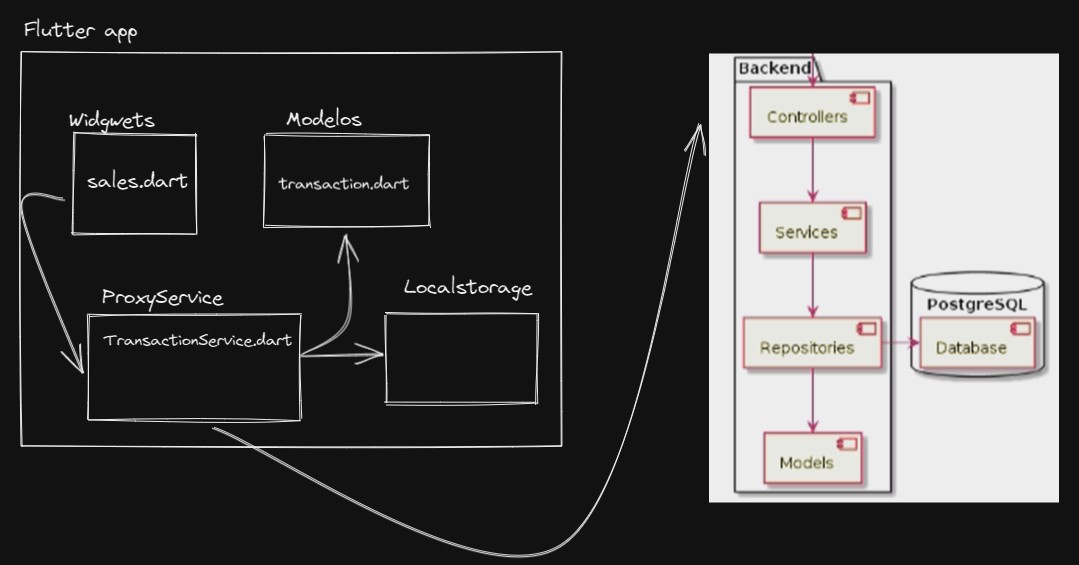
***Figura 7: Arquitectura Backend del proyecto***



Fuente: Lead Working Partner S.A

Las peticiones que llegan desde el servicio del Frontend son recibidos por el controlador del Backend, quien será el encargado de dirigir la solicitud hacia el servicio, el repositorio y finalmente la base de datos, dependiendo de la operación que se solicite desde el Frontend.

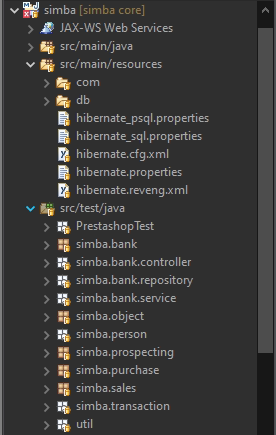
La arquitectura conjunta quedaría de la siguiente manera como se presenta en la figura 8.

***Figura 8: diagrama de arquitectura del proyecto***

Fuente: Lead Working Partner S.A.

A continuación, se representa en la figura 9 la distribución del proyecto en el IDE eclipse:

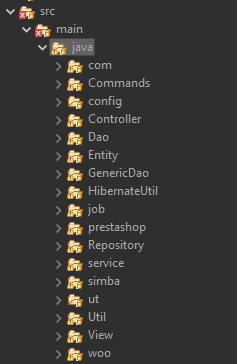
***Figura 9:Distribución del Proyecto en el IDE***



Fuente: Lead Working Partner S.A

Distribución del Backend del proyecto:

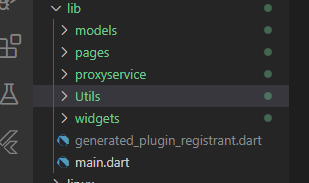
***Figura 10:Distribución del Backend en el IDE***



Fuente: Lead Working Partner S.A

Para desarrollar los nuevos módulos se debe trabajar con la estructura ya planteada en la figura 10, para ello se deberá crear o trabajar en un mismo controlador, servicio, repositorio y modelo.

Distribución del Aplicativo móvil:

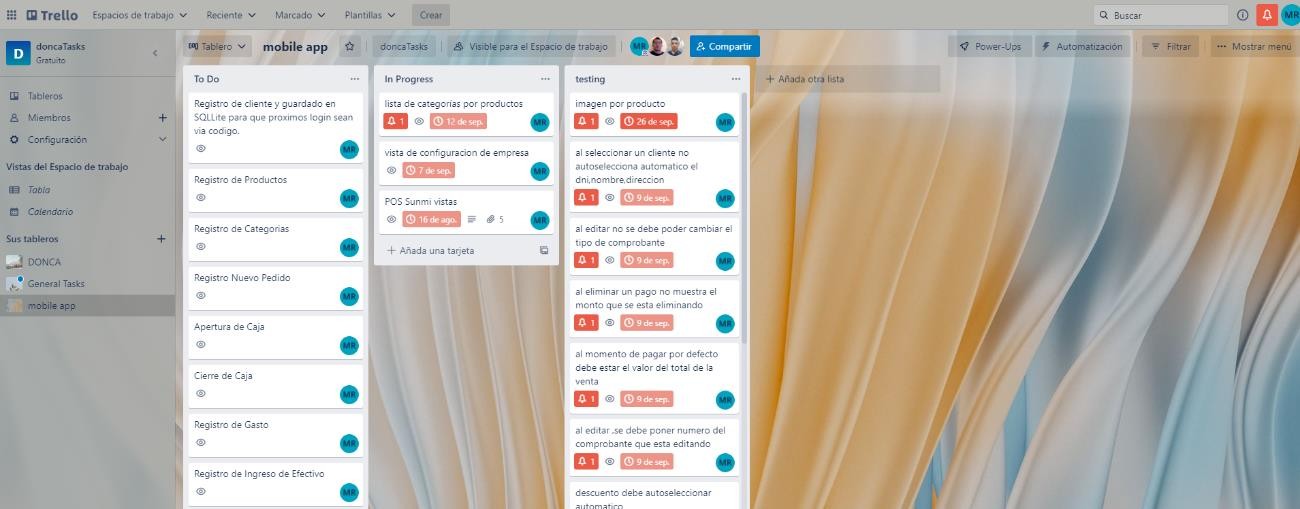
***Figura 11: Distribución del aplicativo móvil en el IDE***

Fuente: Lead Working Partner S.A

La figura 11 muestra la distribución del aplicativo móvil, para comenzar a desarrollar se debe de crear un widget quien será encargado de la lógica de la vista con la que interactúa el usuario.

## Método de trabajo

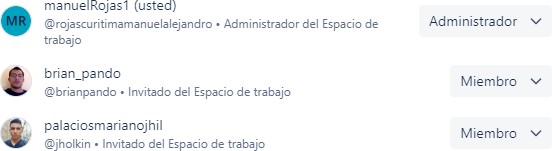
La metodología para usar en el desarrollo de los módulos es la metodología Kanban. Para ello se creó un tablero usando la herramienta Trello donde se asignaron las tareas de acuerdo con los prototipos y requisitos presentados al practicante.

***Figura 12:Tablero Kanban en trello***

Fuente: Elaboración propia

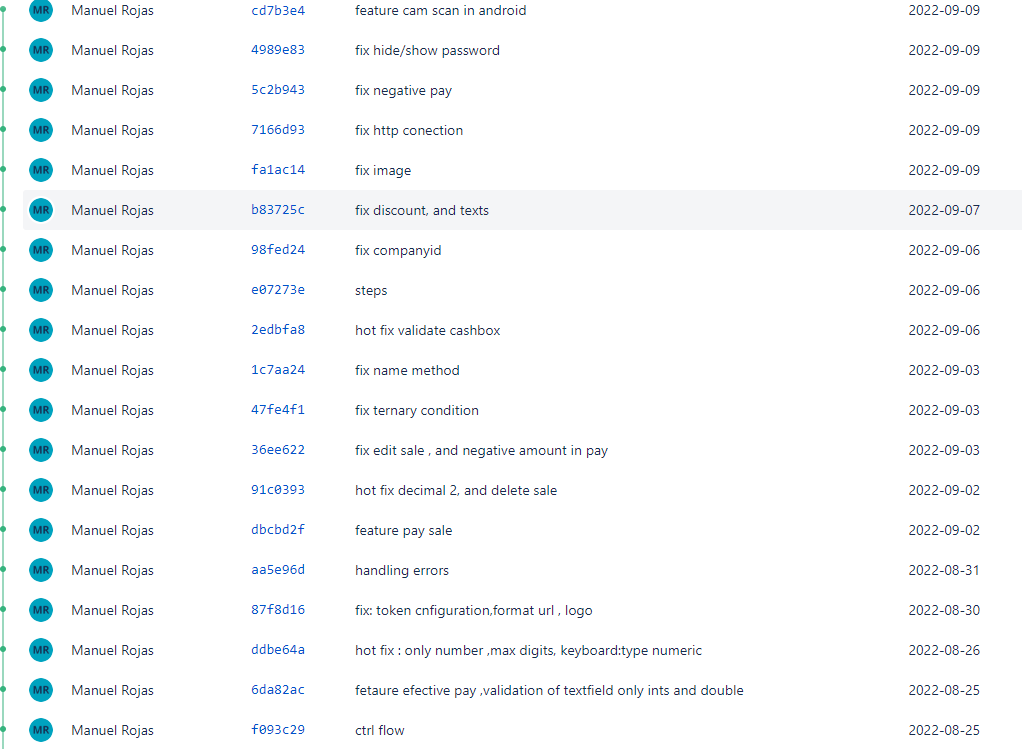
La figura 12 muestra el tablero Kanban, donde se encuentran las tareas y revisiones que va realizando el practicante durante el desarrollo de los módulos del aplicativo móvil, para luego ser revisados por los miembros del equipo detallado en la sección equipo de trabajo.

***Figura 13:Miembros participantes en el tablero Kanban***



Fuente: Elaboración propia

Se uso como repositorio de trabajo Bitbucket, donde se tiene un control de lo que el practicante va subiendo como se muestra en la figura 14.

***Figura 14:Flujo de subida de código al repositorio en bitbucket.***

Fuente: Lead Working Partner SA

## Módulo de Configuraciones

El módulo configuraciones será el encargado de gestionar el registro de las empresas que utilizaran el aplicativo móvil, esta será usada por el usuario administrador de la empresa, donde se registrara el número de ruc y la url de conexión a su servidor.

A continuación, se detallarán algunos de los requisitos y prototipos que se plantearon al practicante para su desarrollo.

### Requisitos del módulo de Configuraciones

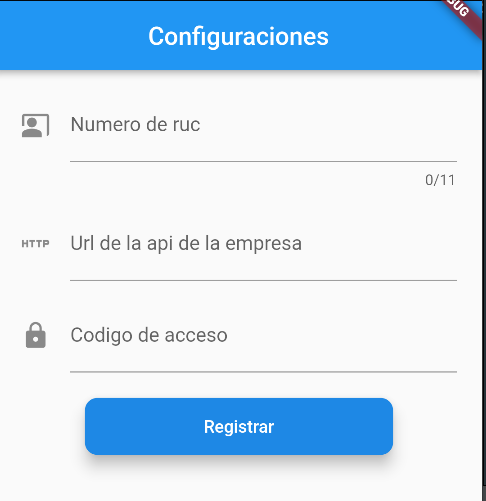
***Tabla 4: Requisitos del módulo configuraciones***

|  |  |  |  |
| --- | --- | --- | --- |
| **Modulo** | **Código** | **Descripción** | **Prioridad** |
| Configuraciones | RFC01 | Registro de empresa con ruc, url del servidor y un código único de  acceso | Alta |
| RFC02 | Eliminar registro de empresa | Alta |

Fuente: Lead Working Partner S.A

### Prototipos del módulo Configuraciones

***Figura 15:Prototipo – Registro de Configuraciones***



Fuente: Lead Working Partner S.A

La figura 15 presenta el prototipo de la vista principal del módulo configuraciones, que tiene como finalidad registrar la empresa donde se usara el aplicativo móvil, a su vez este validara la data con el Backend de la aplicación web principal.

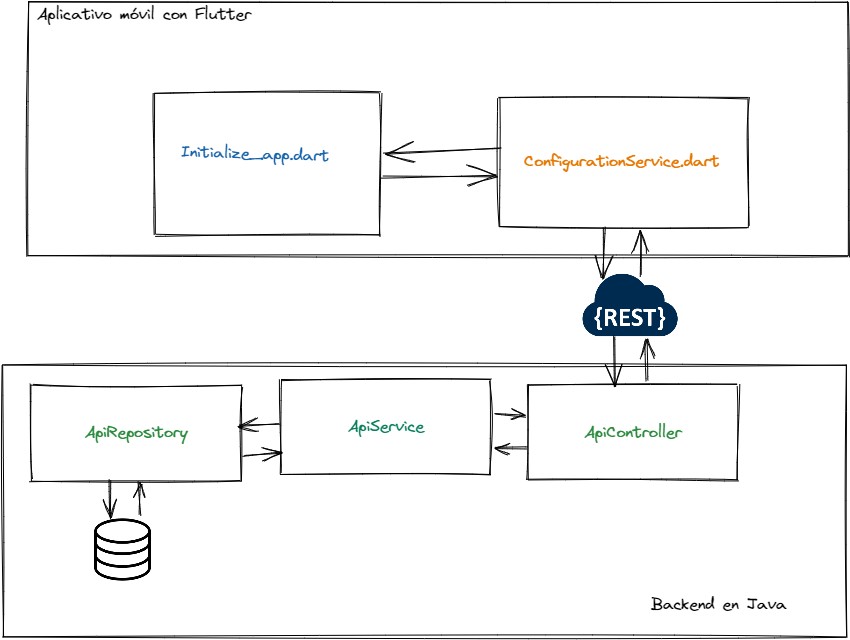
Las validaciones que se tendrán en cuenta son:

* + - * Cantidad de dígitos en el ruc (11).
      * Teclado numérico para el ingreso de ruc.

### Construcción del módulo de Configuraciones

Para iniciar con la construcción del módulo se tendrá en cuenta el siguiente diagrama:

***Figura 16:Flujo de trabajo para la construcción del módulo configuraciones.***

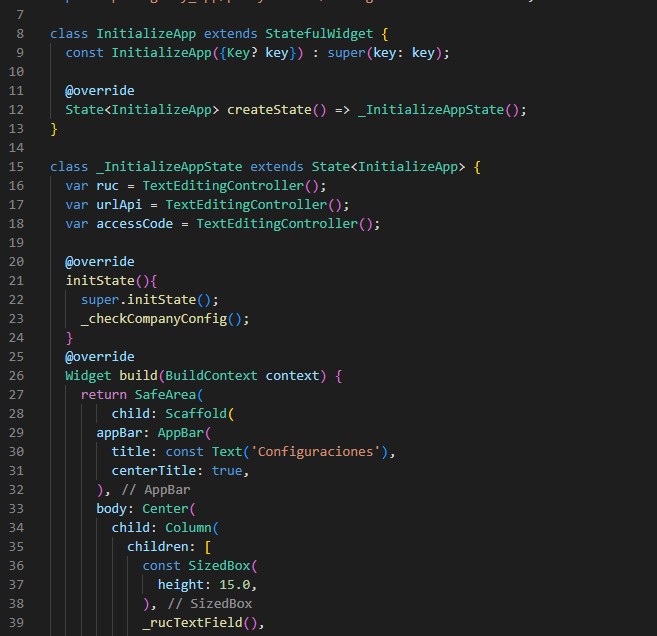


Fuente: Elaboración propia

Se comenzó a trabajar desde el aplicativo móvil, para ello se creó el archivo **Initialice\_app.dart,** que contendrá todas las vistas que interactuaran con el usuario final.

En Flutter para indicar un contenedor se usa la palabra clave **Scaffold** dentro del metodo principal build, como se muestra en la figura 17.

***Figura 17:Fragmento de Código del contenedor del módulo configuraciones.***

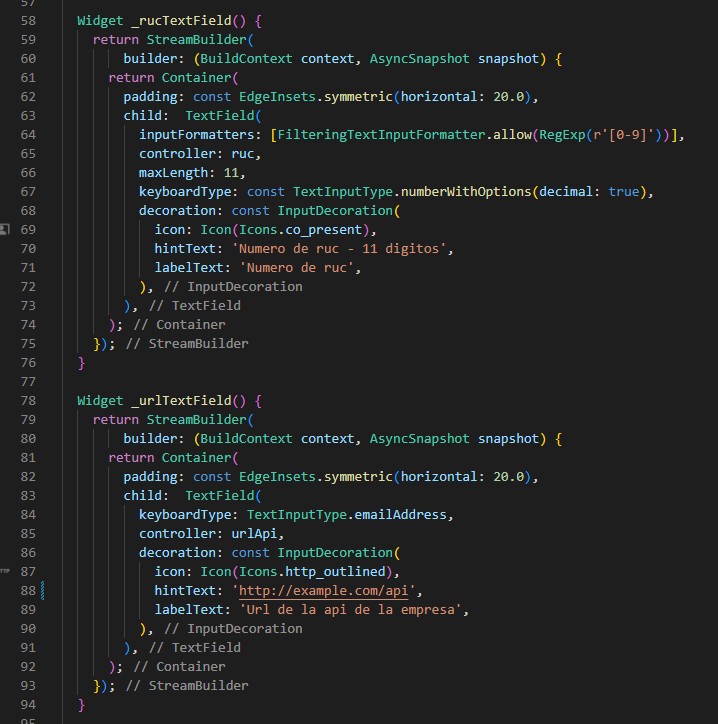


Fuente: Elaboración propia.

Una vez creado el contenedor comenzamos a crear los atributos como son, el ingreso ruc, url y código acceso como se muestra en la figura 18.

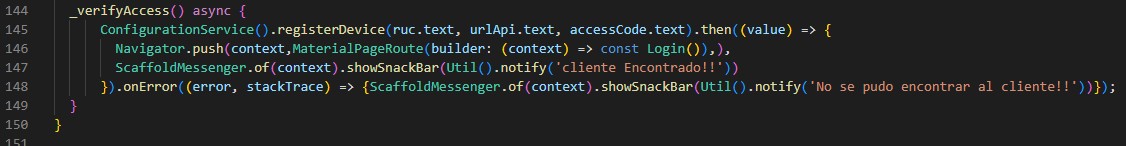
Además, se hizo uso de expresiones regulares, que permiten ingresar solo números enteros y de cierto tamaño de dígitos como se muestra en la figura 18, así como también se activó el teclado numérico cuando el usuario se encuentre ingresando

datos del ruc, para ello se usó la opción **TextInputType** que provee Flutter.

***Figura 18:Codificacion de los atributos, ruc, url, Código de acceso.***

Fuente: Elaboración propia

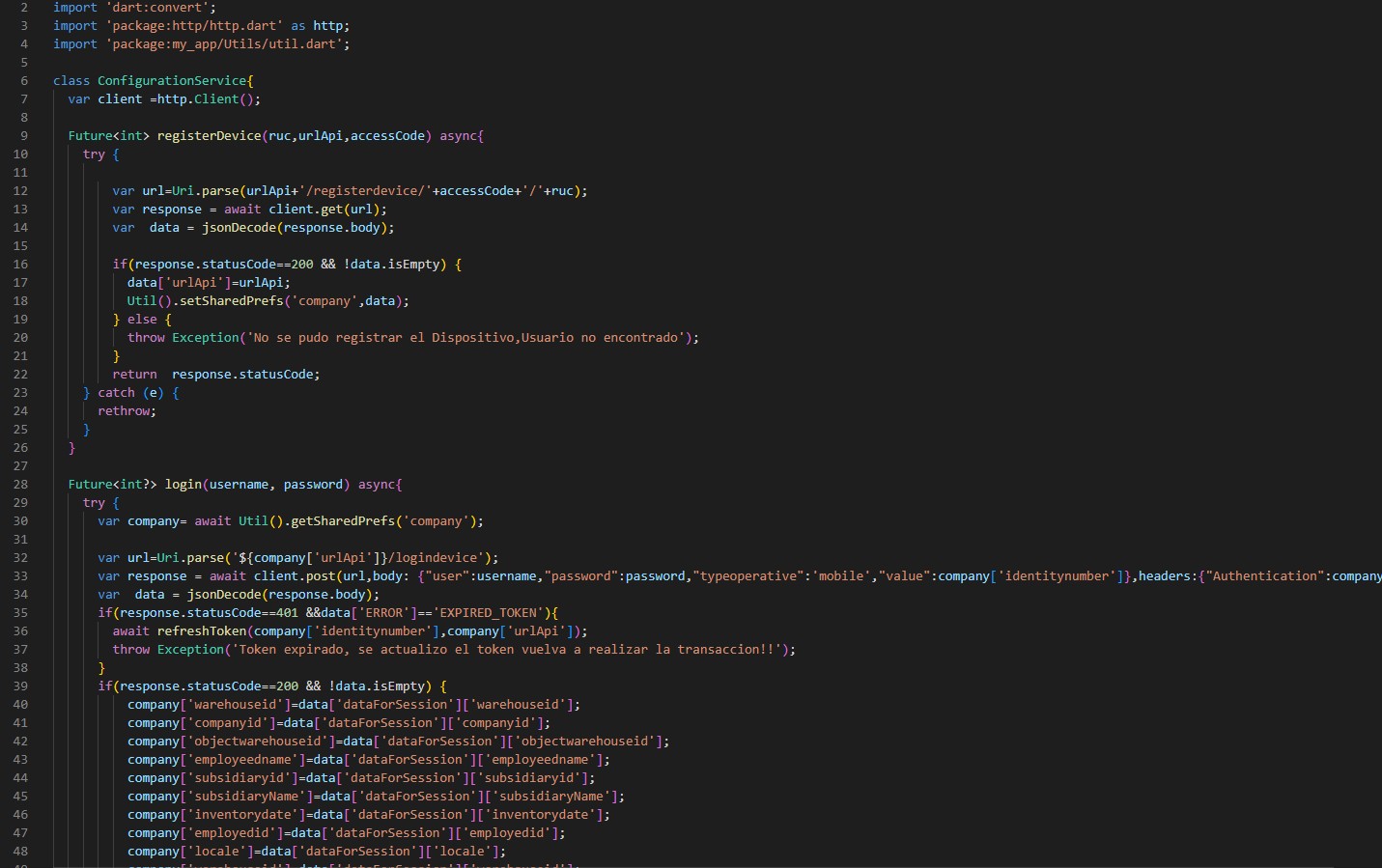
También se hizo uso de notificaciones que ayuden al usuario a informar sobre lo que se encuentra realizando, la implementación se muestra en la figura 19.

***Figura 19:Implementación de Alertas.***

Fuente: Elaboración propia

Para proveer y verificar los datos de la aplicación, se creó un archivo llamado

**configurationservice,** quien es el encargado de comunicarse con el Backend, para ello se crearon 3 servicios como se muestra en la figura 20.

***Figura 20:Creacion de servicios que se conectaran al Backend.***

Fuente: Elaboración propia.

Una vez codificada la parte de configuraciones del aplicativo móvil en Flutter, pasamos a crear los controladores en el Backend que serán los encargados de recibir la solicitud desde el aplicativo móvil.

El Backend está desarrollado en el lenguaje Java, usando spring como marco de trabajo, en la figura 21 se muestra cómo se creó el servicio que registra el dispositivo en la base de datos, primero se debe crear el controlador.

Se debe tener en cuenta, para indicar a spring que la clase será un controlador se debe especificar la etiqueta **@RestController** y finalmente indicarle a la función con la etiqueta **@RequestMapping** que esta se activará cuando se solicite de acuerdo con la determinada ruta.



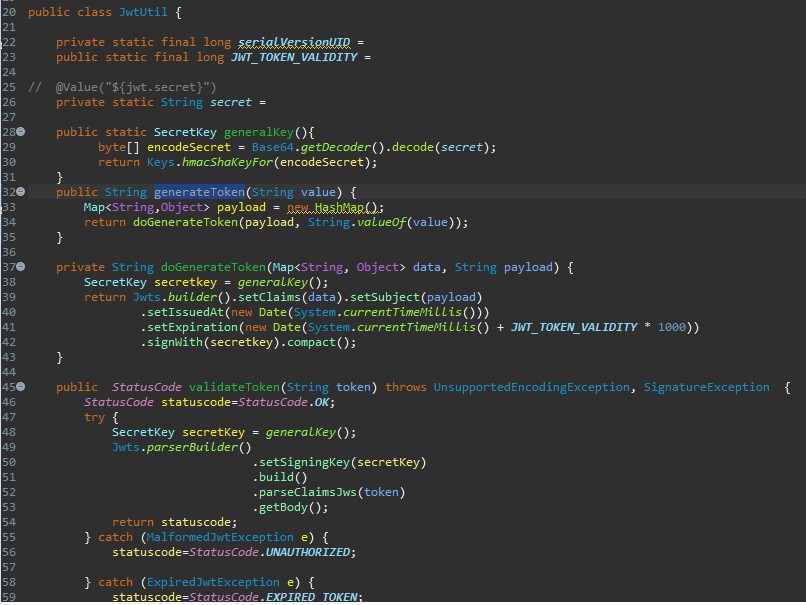
***Figura 21:Creacion del controlador en el Backend.***

Fuente: Elaboración propia

Para que otros sistemas no puedan acceder a la ruta del api, se hizo una validación por token usando json web tokens, quien será el encargado de verificar si es una fuente confiable la que está realizando la petición.

El solicitante deberá adjuntar en la cabecera de la petición un campo llamado **Authentication**, que tendrá como valor el token que se le asigno, previa solicitud de registro como se muestra en la figura 21.

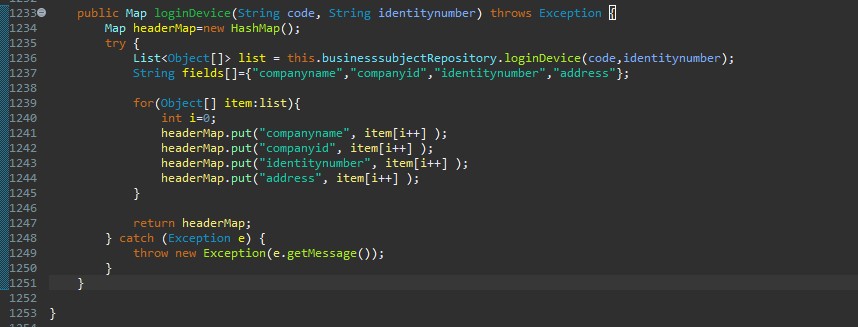
***Figura 22: método de generación y validación de token***



Fuente: Elaboración propia.

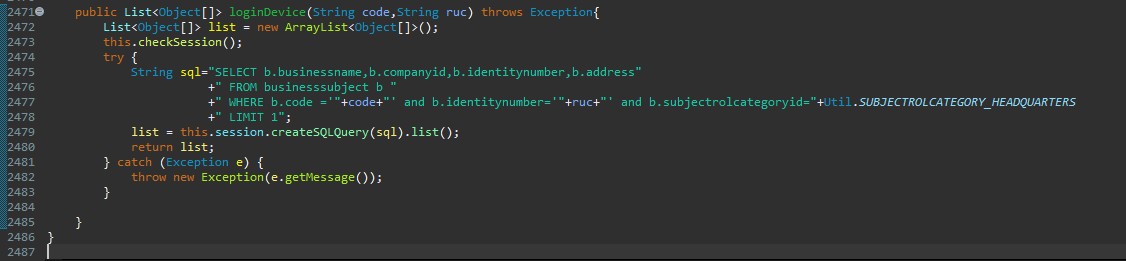
Una vez validada la solicitud, el controlador pasa la solicitud hacia el servicio donde se hará la lógica necesaria para responder o recibir las solicitudes como se muestra en la figura 22.

***Figura 23:Implementacion de la capa de servicio para el registro de dispositivo***



Fuente: Elaboración propia

Posteriormente se pasa a la capa de repositorio, quien es el encargado de escribir y leer en la base de datos, para ello se hace uso del orm hibernate que gestionas las conexiones a la base de datos como se muestra en la figura 23.

***Figura 24: Capa de repositorio para el registro del dispositivo***

Fuente: Elaboración propia

Finalmente, si la solicitud ha sido exitosa el Backend devuelve, la información que se hizo en la solicitud.

Si la respuesta fue exitosa, se almacena todos los datos en el localstorage del dispositivo, para finalmente terminar el proceso mostrando la ventana de inicio de sesión.

## Modulo Inicio de Sesión

El módulo de inicio de sesión será la vista principal que se mostrará a los usuarios finales, este estará sincronizado con los datos de los usuarios registrados en la aplicación web.

A continuación, se presentan algunos requisitos y prototipos que fueron proporcionados al practicante.

### Requisitos del módulo inicio de sesión

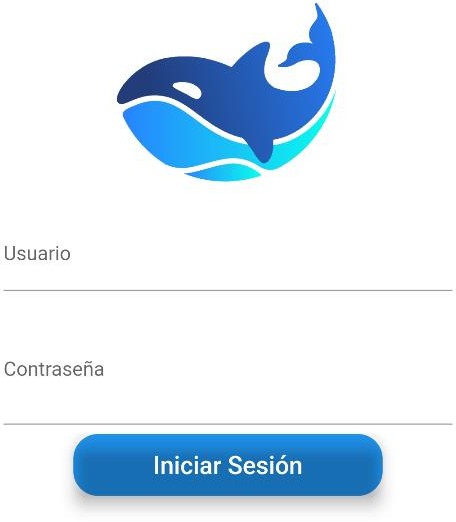
***Tabla 5:Requisitos del módulo inicio de sesión***

|  |  |  |  |
| --- | --- | --- | --- |
| **Modulo** | **Código** | **Descripción** | **Prioridad** |
| Inicio de sesión | RFDP01 | Inicio de sesión sincronizado con  los datos de la aplicación web | Alta |
| RFDP02 | Validación de usuario y clave  incorrecta | Alta |
| RFDP03 | Cargar los datos de variables globales en localstorage por  usuario logrado. | Alta |

Fuente: Lead working Partner SA

### Prototipos del módulo Inicio de sesión

***Figura 25:Prototipo del módulo inicio de sesión.***

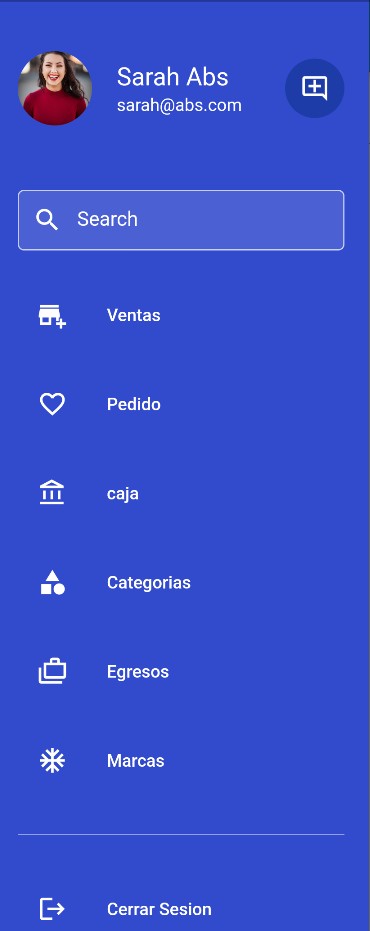


Fuente: Lead Working Partner SA

La figura 25 muestra el prototipo de inicio de sesión, para codificar este módulo se debe tener en cuenta lo siguiente:

* + - * Los usuarios deben validarse con los datos registrados desde la aplicación, es decir que si el administrador registro usuarios desde la página web, estos también deben poder ser utilizados desde el aplicativo móvil.

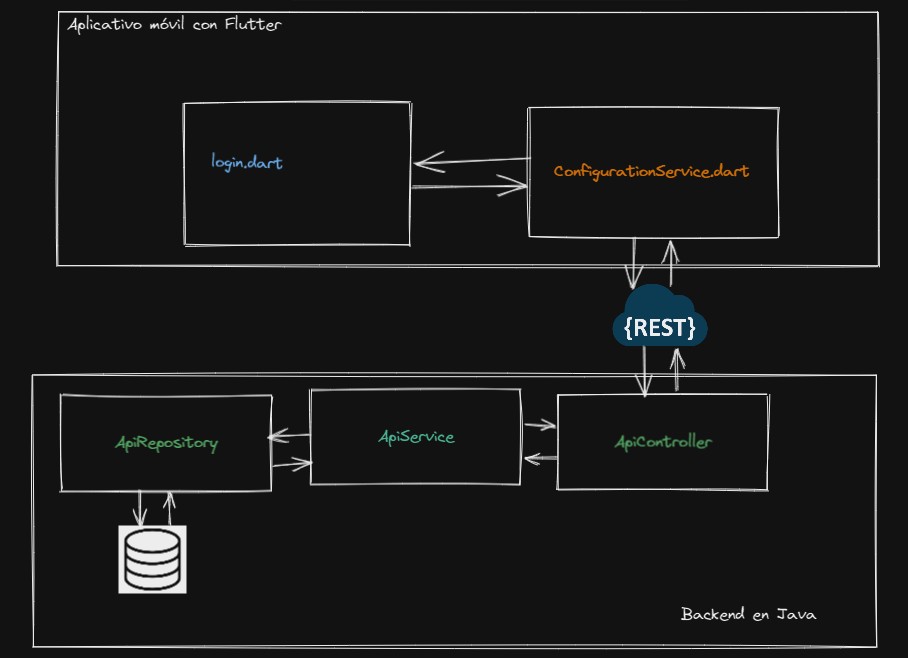
En la figura 26 se muestra el prototipo del menú, que contendrá algunas de las opciones principales de la aplicación web.

***Figura 26:Prototipo del menú.***

Fuente: Lead working partner SA

### Construcción del módulo Inicio de sesión

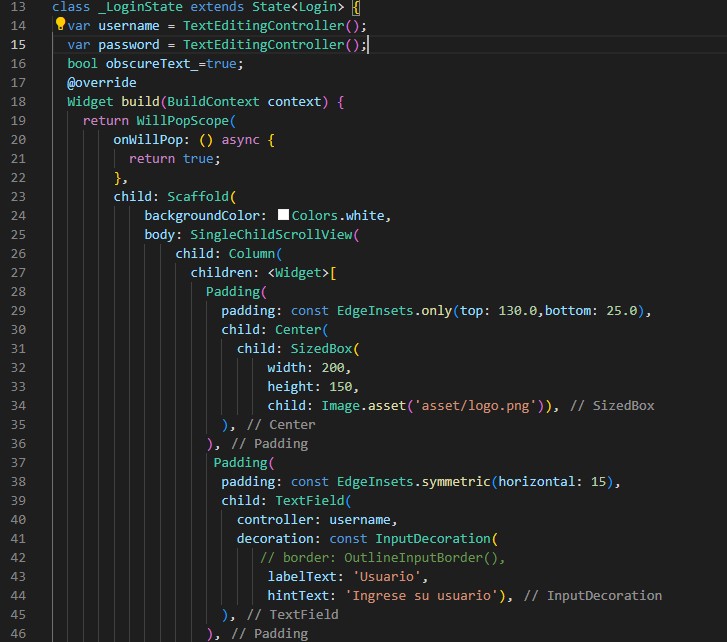
Para la construcción del módulo inicio de sesión se tuvo en cuenta el siguiente flujo de trabajo:



***Figura 27: Flujo de trabajo para la construcción del módulo inicio de sesión***

Fuente: Elaboración propia

Se inicio creando el archivo **login.dart**, quien será el encargado de interactuar con el usuario como se muestra en la figura 28.

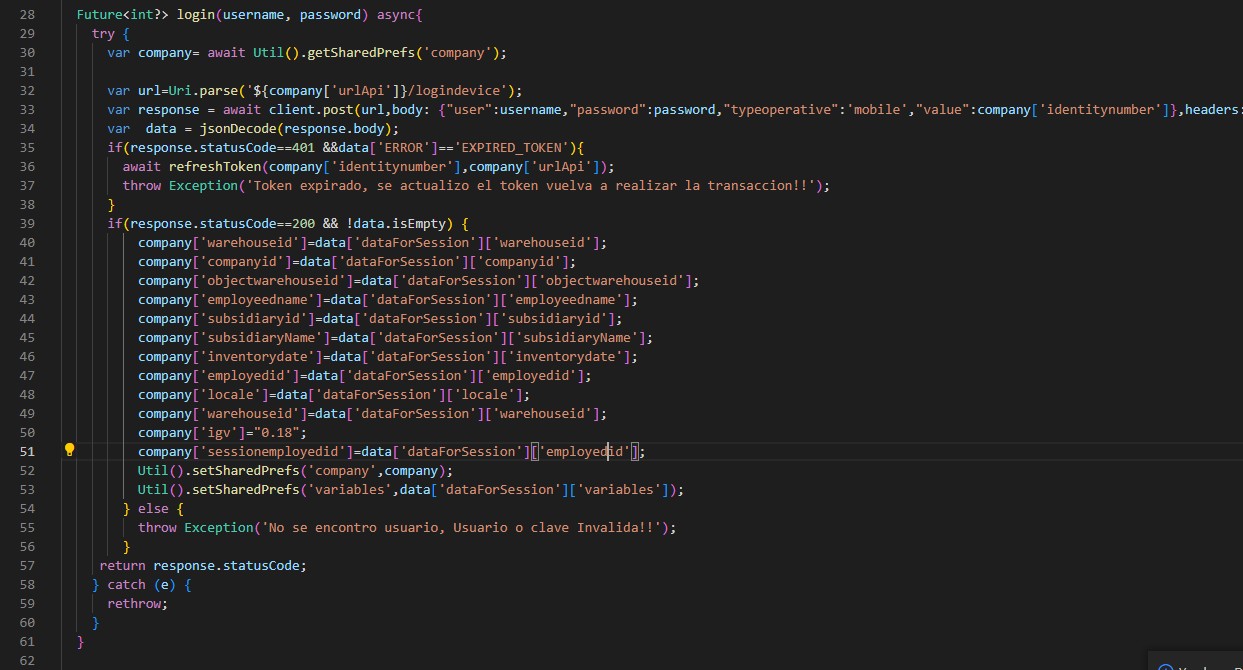
***Figura 28: Fragmento de código de la vista Login.***

Fuente: Elaboración propia

Luego este **login.dart** se conectará con un servicio, donde se encuentran los

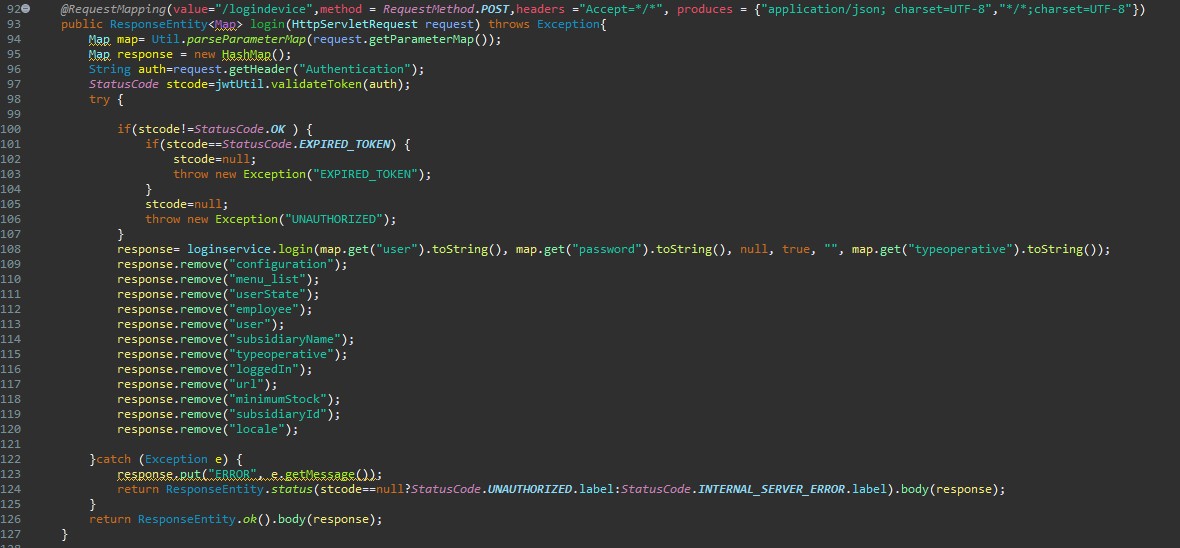
puntos de acceso para realizar las peticiones hacia el Backend como se muestra en la figura 29.

***Figura 29: Fragmento de Código del servicio del login que conecta al Backend de la aplicación.***



Fuente: Elaboración propia

En la parte del Backend, será recibido por un controlador, como se muestra en la figura 30.



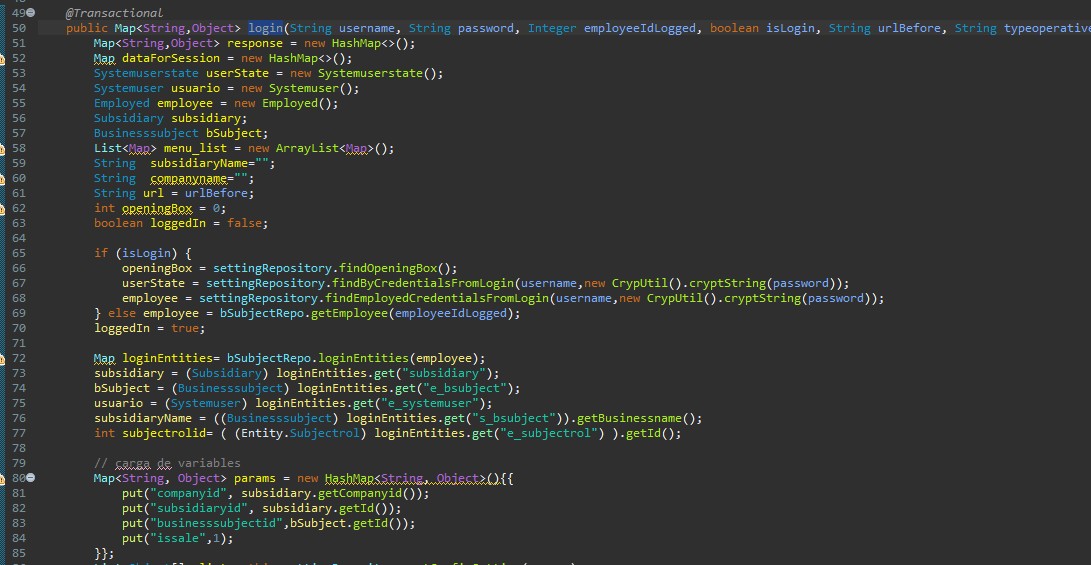
***Figura 30:Fragmento de Código del controlador del Backend, que recibe la solicitud de login.***

Fuente: Elaboración propia

El controlador, luego de validar si es una petición de una fuente confiable, este

pasara a la siguiente capa de servicio, como se muestra en la figura 31.

***Figura 31: Fragmento de Código del servicio del Backend.***



Fuente: Elaboración propia

Finalmente, el servicio enviará la petición hacia el repositorio para que se resuelva de acuerdo con los parámetros enviados, y si no ocurre ningún error este devolverá una respuesta satisfactoria, caso contrario responderá un error, como se muestra en la figura 32.

***Figura 32:Fragmento de Código del repositorio del Backend***

Fuente: Elaboración propia

### Pruebas del módulo Inicio de sesión

## Modulo Categoría

En este módulo se registrarán las categorías de los productos que la empresa ofrece a sus clientes.

De igual forma para el desarrollo de este módulo se reutilizarán componentes genéricos que se desarrollaron anteriormente.

A continuación, se presentan algunos requisitos y prototipos que fueron proporcionados al practicante.

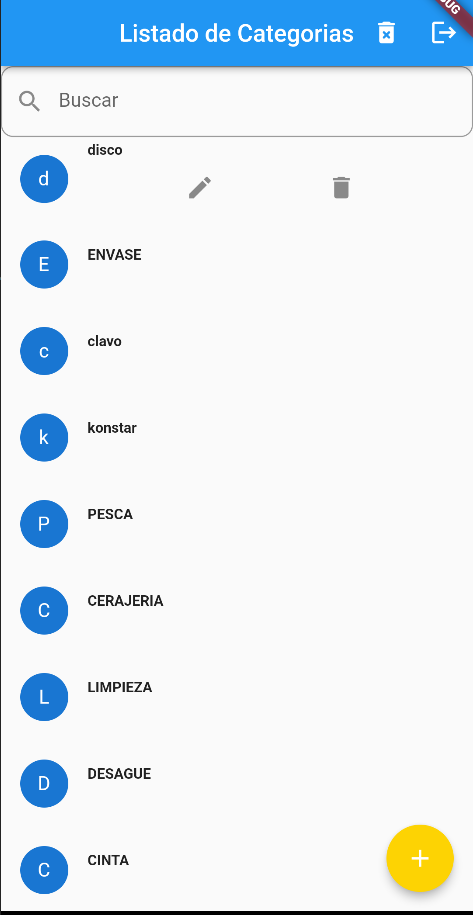
### Requisitos del módulo Categoría

***Tabla 6:Requisitos del módulo categoría.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Modulo** | **Código** | **Descripción** | **Prioridad** |
| Categorías | RFDP01 | Listar categorías. | Alta |
| RFDP02 | Eliminar categorías | Alta |
| RFDP03 | Editar categorías | Alta |
| RFDP10 | Filtro de búsqueda por nombre  de categoría | Baja |

Fuente: Lead Working Partner SA

### Prototipos del módulo Categoría

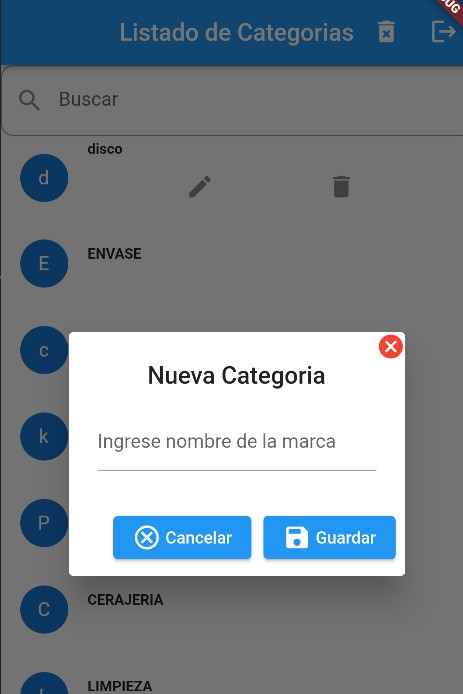


***Figura 33:Prototipo del Módulo categoría - Listar categorías***

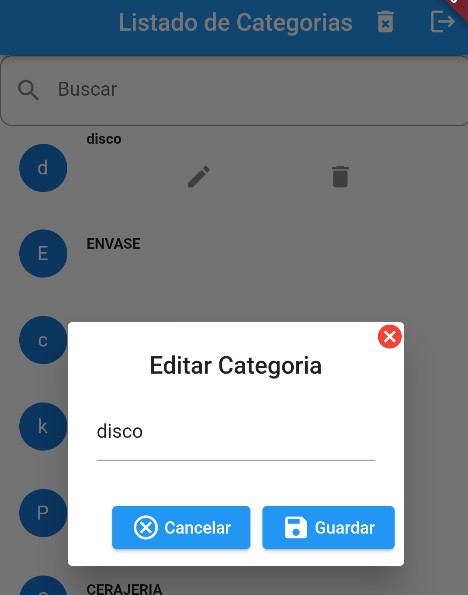
Fuente: Lead Working Partner SA

La figura 33 muestra el prototipo del módulo categoría, específicamente el

listado, que será la ventana principal que encontrará el usuario. En la figura 34 se muestra el prototipo de crear una nueva categoría, que será accionado desde el botón de color amarillo, con el símbolo de cruz.

***Figura 34:Prototipo de Nueva categoría.***

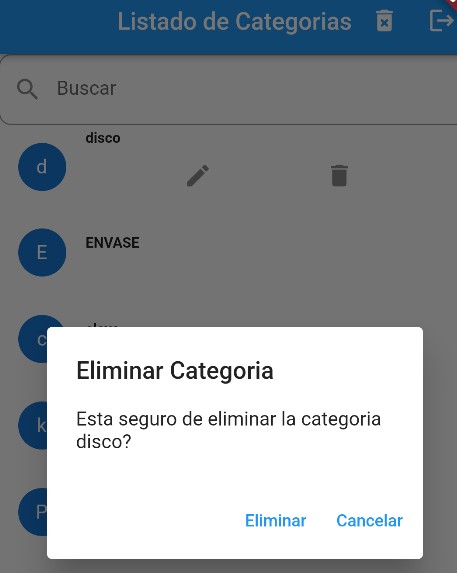
Fuente: Lead Working Partner SA

***Figura 35:Prototipo de edición de categoría***

Fuente: Lead Working Partner SA

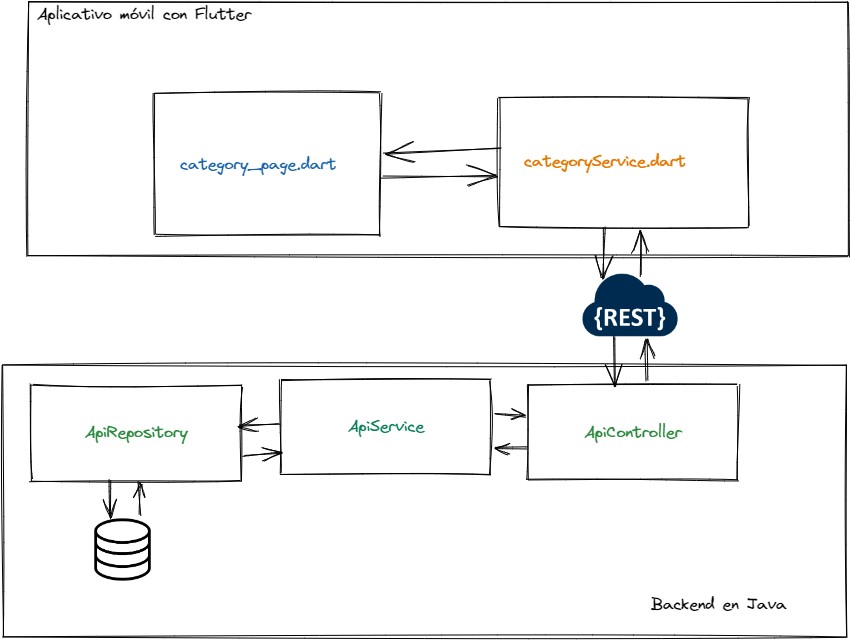
La figura 35 muestra la opción de editar una categoría, esta será activada cuando el usuario presione el botón que contiene un icono de lápiz. La figura 36 muestra la opción de eliminar una categoría, para acceder a esa opción el usuario deberá presionar en el botón con icono de un tacho, luego le aparecerá una alerta para que confirme la acción.

***Figura 36: Prototipo de la opción de eliminar la categoría****.*



Fuente: Lead Working Partner SA

### Construcción del módulo Categoría



***Figura 37: Flujo de trabajo para la construcción del módulo categoría.***

Fuente: Elaboración propia

Se trabajo con el flujo mostrado en la figura 37, se comenzó por el archivo **category\_page.dart**, que será con el que el usuario interactuará primero, a continuación, se muestra un fragmento de código de la opción de listar categoría en la figura 38.

***Figura 38: Fragmento de código de listar categoría.***

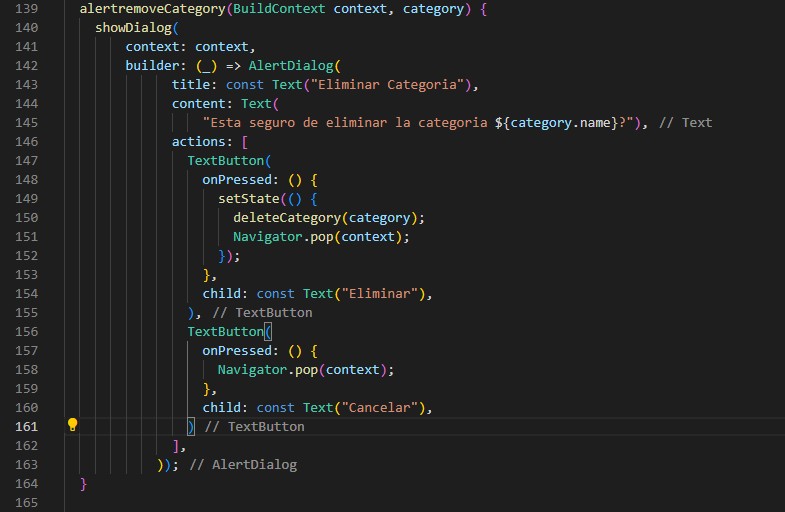
Fuente: Elaboración propia

A continuación, la figura 39 y 40, muestran los fragmentos de código la opción guardar categoría y eliminar categoría.

***Figura 39:Fragmento de código de la opción de guardar categoría***

Fuente: Elaboración propia

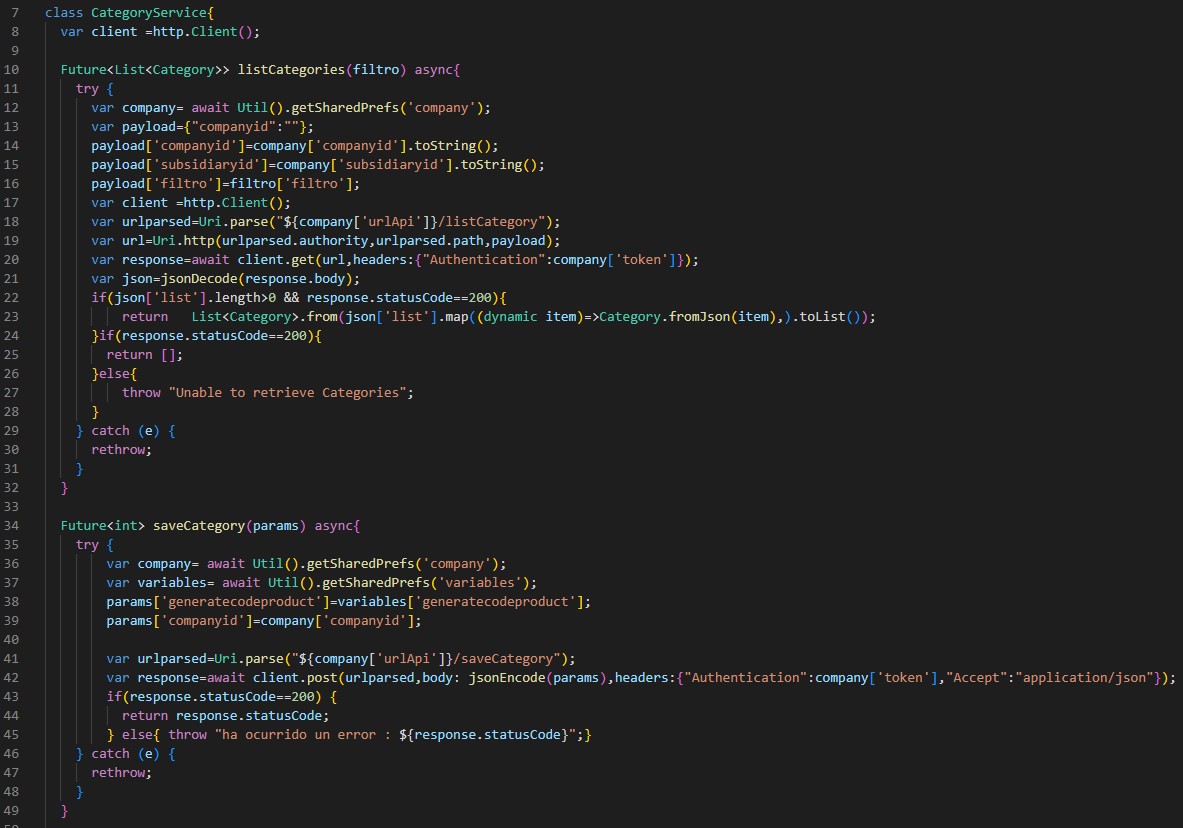
***Figura 40: Fragmento de código de la opción eliminar categoría***



Fuente: Elaboración propia

Luego este se conectaba con un servicio llamado **CategoryService.dart**, quien envía la petición a través de https hacia el controlador del Backend como se muestra en la figura 41.

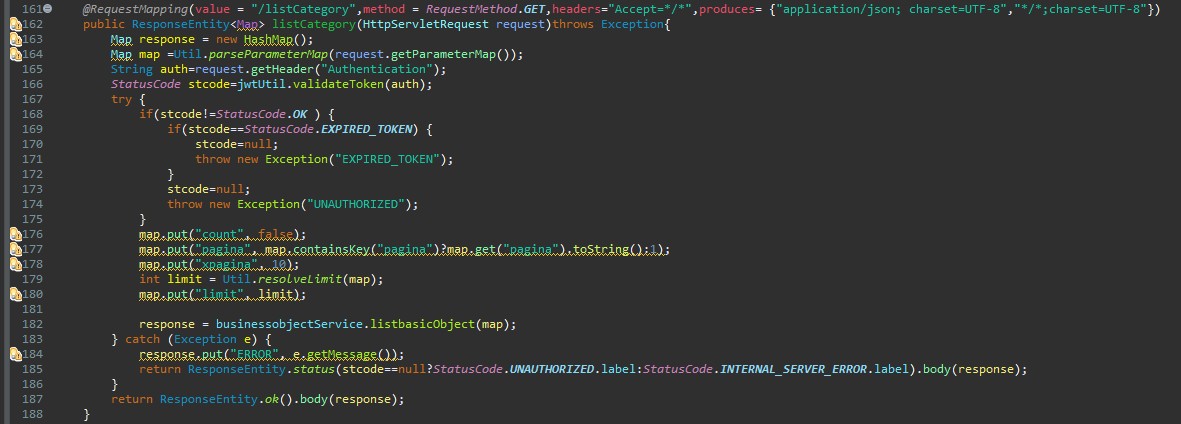
***Figura 41:Fragmento de código del servicio categoría***



Fuente: Elaboración propia.

Luego en el Backend el controlador, recibe la petición y valida que sea de una fuente confiable, como se muestra en la figura 42.

***Figura 42: Fragmento de código del controlador del Backend - categoría***



Fuente: Elaboración propia

Finalmente, el controlador envía la petición hacia el servicio y luego este envía al repositorio para que resuelva y responda satisfactoriamente o un error en caso lo hubiera como se muestra en la figura 43.

***Figura 43:Fragmento de código del servicio y repositorio del Backend- categoría***

Fuente: Elaboración propia

## Modulo Marcas

En este módulo se registrarán las marcas de los productos que ofrece la empresa a sus clientes.

A continuación, se presentan algunos requisitos y prototipos que fueron proporcionados al practicante.

### Requisitos del módulo Marcas

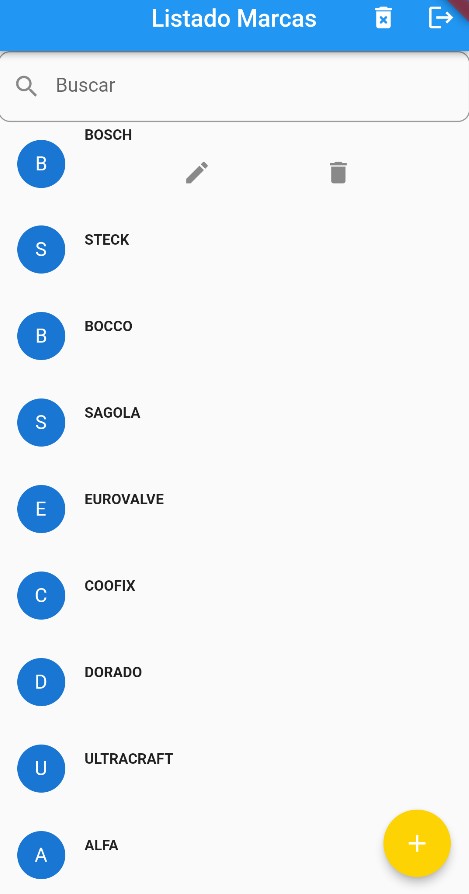
***Tabla 7:Requisitos del módulo marca.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Modulo** | **Código** | **Descripción** | **Prioridad** |
| Marcas | RFDP01 | Listar marcas. | Alta |
| RFDP02 | Eliminar marcas. | Alta |
| RFDP03 | Editar marcas. | Alta |
| RFDP04 | Filtro de búsqueda por nombre  de marca | Alta |

Fuente: Lead Working Partner SA

### Prototipos del módulo Marcas

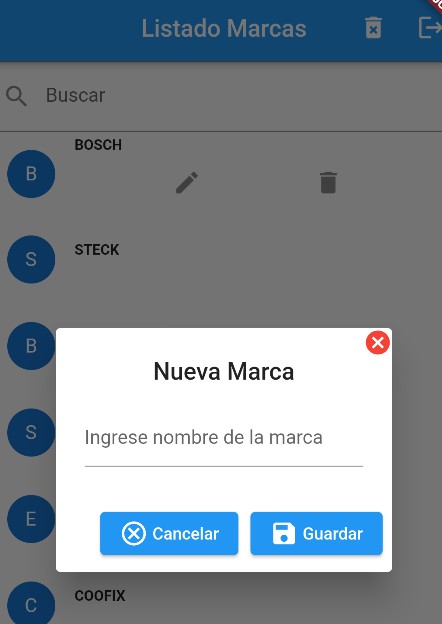
***Figura 44: Prototipo listar marcas.***



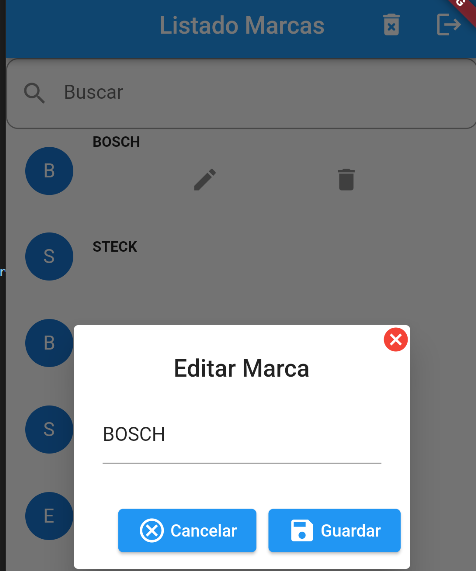
Fuente: Lead Working Partner SA

La figura 44 muestra el prototipo de listado de marcas, cada una de ellas tendrá la opción de editar y eliminar, para crear una nueva marca el usuario deberá presionar en el botón de color amarillo con el símbolo de cruz, como se muestran en las figuras 45 y 46.

***Figura 45:Prototipo- crear marca.***

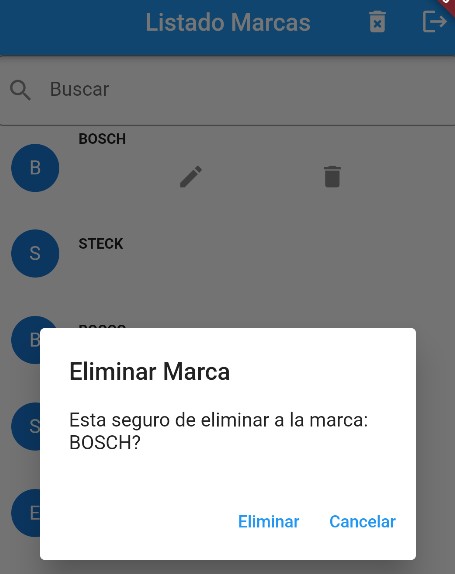


Fuente: Lead Working Partner SA

***Figura 46:Prototipo- editar marca***

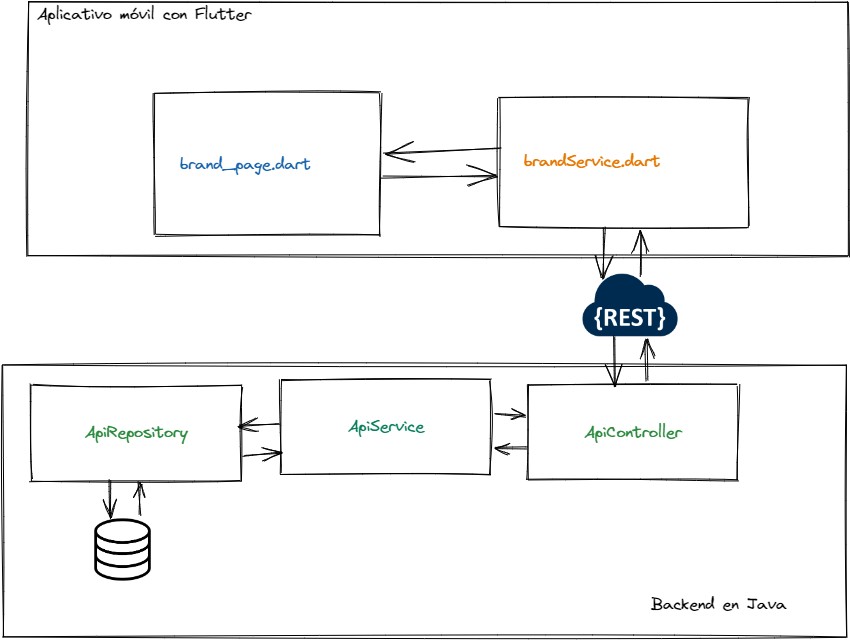
Fuente: Lead Working Partner SA

La figura 47 muestra el prototipo de eliminar una marca, para ello se deberá mostrar un dialogo para que el usuario pueda confirmar la acción.

***Figura 47:Prototipo- eliminar marca***

Fuente: Lead working partner SA

### Construcción del módulo Marcas



***Figura 48:Flujo de trabajo para la construcción del módulo marca.***

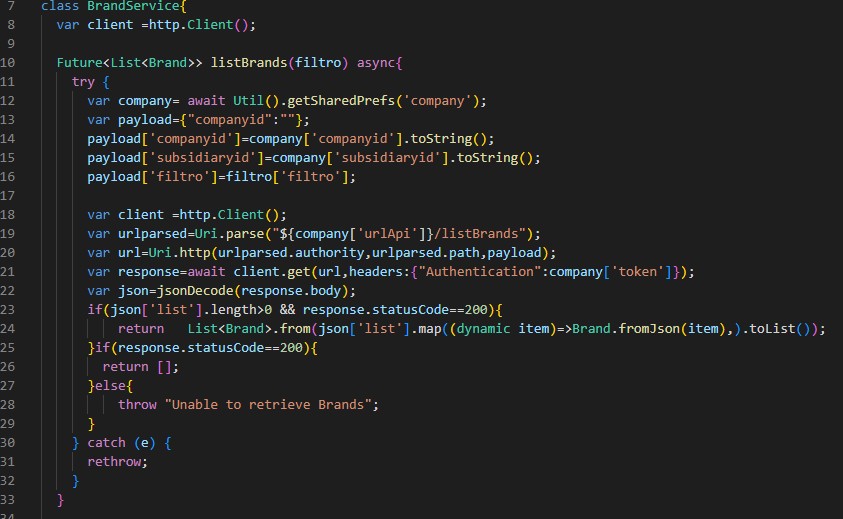
Fuente: Elaboración propia

La figura 49, muestra el flujo de trabajo para la construcción del módulo marcas, para ello se inició creando el archivo **brand\_page.dart**, quien será el encargado de interactuar con el usuario, como se muestra en la figura 49.

***Figura 49: Fragmento de código- listar marcas.***

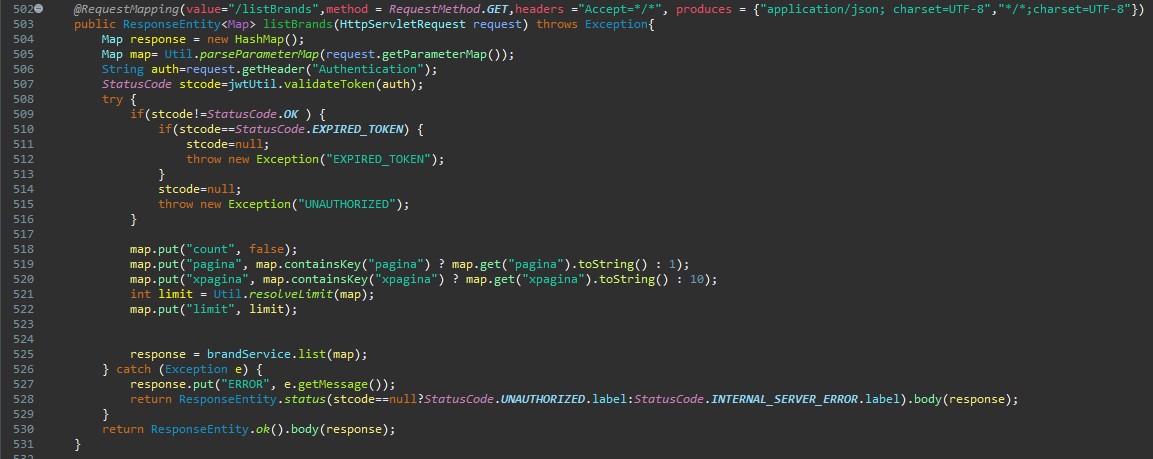
Fuente: Elaboración propia

El modelo a su vez se conecta con en servicio quien le provee los datos necesarios para mostrar al usuario como se muestra en la figura 50.

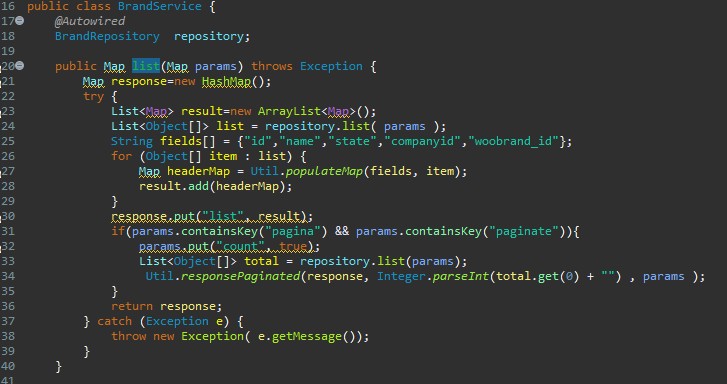
***Figura 50: Fragmento de código del servicio- listar marca.***

Fuente: Elaboración propia

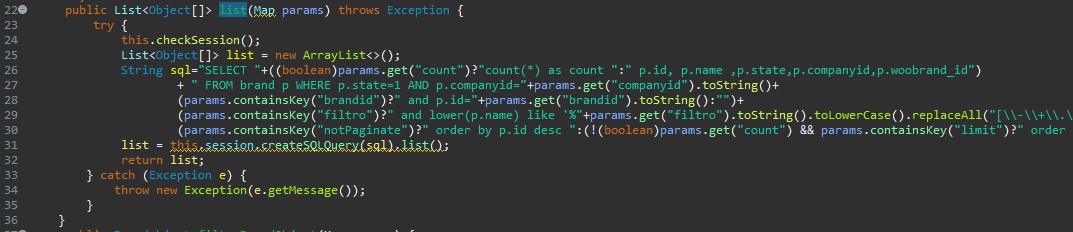
Luego este será recibido por el controlador del Backend, quien luego de revisar si es una petición valida, lo pasara hacia el servicio, que finalmente enviara al repositorio para que resuelva y responder satisfactoriamente o un error si lo hubiera como se muestran en las figuras 51,52,53.

***Figura 51:Fragmento de código del controlador del Backend-listar marca.***

Fuente: Elaboración propia

***Figura 52: Fragmento de condigo del servicio del Backend- listar marca.***

Fuente: Elaboración propia

***Figura 53:Fragmento de código del repositorio del Backend- listar marca.***

Fuente: Elaboración propia

## Modulo Caja

El módulo caja será el encargado de administrar las entradas y salidas de dinero que se realice en la aplicación. Mostrando los comprobantes que se cobran durante el intervalo de la apertura y cierre de la caja.

A continuación, se presentan algunos requisitos y prototipos que fueron proporcionados al practicante.

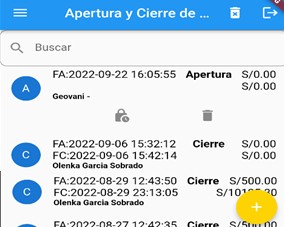
### Requisitos del módulo Caja

***Tabla 8:Prototipo del módulo caja.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Modulo** | **Código** | **Descripción** | **Prioridad** |
| Caja | RFDP01 | Listar las cajas | Alta |
| RFDP02 | Opción de apertura de caja | Alta |
| RFDP03 | Opción de cierre de caja | Alta |
| RFDP04 | Eliminar caja, siempre y cuando esta no tenga  ninguna transacción. | Alta |
| RFDP05 | Vista de reporte de pagos que se van realizando | Alta |

Fuente: Lead working Partner SA

### Prototipos del módulo Caja

***Figura 54:Prototipo-listar cajas***

Fuente: Lead Working Partner SA

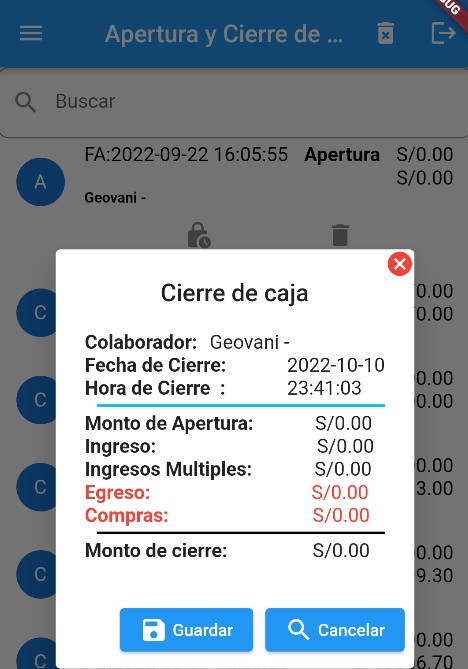
La figura 54 muestra el prototipo del módulo caja, la lista debe contener las opciones de cerrar caja en caso este abierta, además debe tener la opción de eliminar, para crear una nueva caja, el usuario deberá presionar el botón amarillo con el icono de una cruz, como se muestran en la figura 55,56,57.

***Figura 55: Prototipo de apertura de caja.***



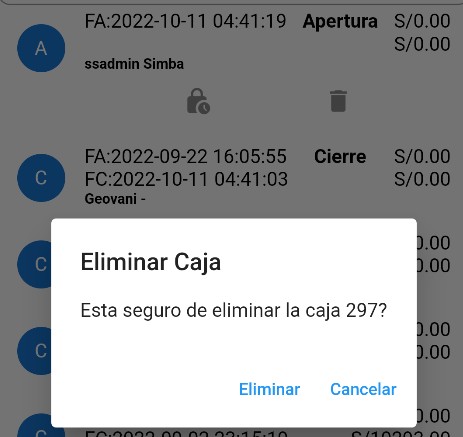
Fuente: Lead Working Partner SA

***Figura 56:Prototipo de cierre de caja.***



Fuente: Lead working partner SA

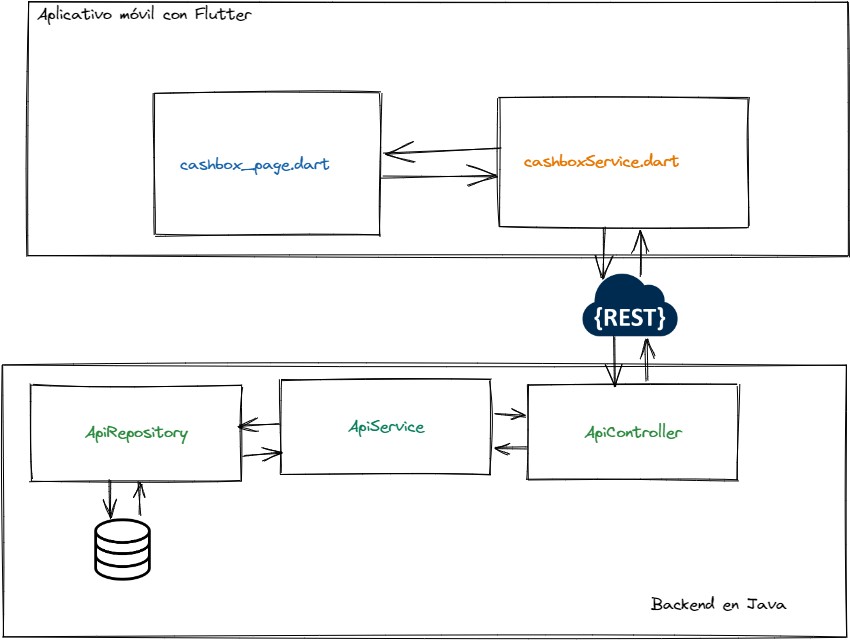
***Figura 57:Prototipo de eliminar caja.***



Fuente: Lead working partner SA

### Construcción del módulo Caja

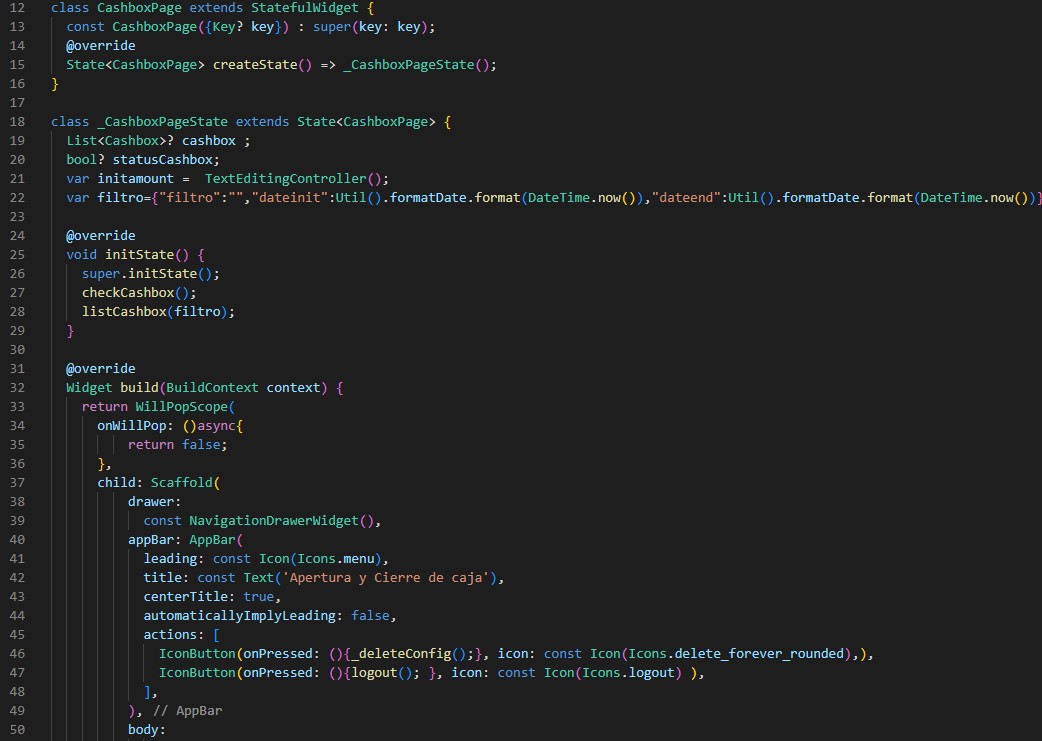
***Figura 58:Flujo de trabajo para la construcción del módulo caja.***



Fuente: Elaboración propia

Se comenzó por el archivo **cashbox\_page.dart**, donde estará el código de la

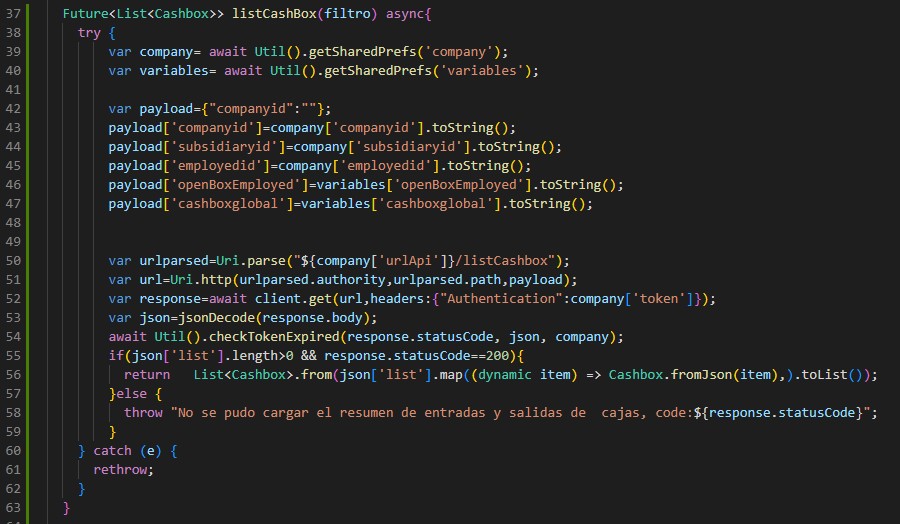
vista con la que interactuará el usuario, como se muestra en la figura 59.

***Figura 59:Fragmento de código- listar cajas.***

Fuente: Elaboración propia

Además, se creó un servicio el que tendrá la finalidad de proveer los datos necesarios al modelo, como se muestra en la figura 60.

***Figura 60:Fragmento de código del servicio- listar cajas.***



Fuente: Elaboración propia

Luego el controlador del Backend recibe la petición, valida si es correcto, para luego pasar entre las capas de servicio y repositorio quien resolverá la petición satisfactoriamente o con un error en caso lo hubiera como se muestra en la figura

61,62,63.

***Figura 61: Fragmento de código del controlador del Backend- listar cajas.***



Fuente: Elaboración propia

***Figura 62:Fragmento de código del servicio del Backend- listar cajas.***



Fuente: Elaboración propia

***Figura 63:Fragmento de código del repositorio del Backend - listar cajas.***

Fuente: Elaboración propia

## Modulo Ventas

El módulo Ventas será el encargado de realizar transacciones de tipo boletas o facturas para los clientes, este módulo también estará sincronizado con la aplicación web, desde donde se generará los correlativos correspondientes a cada venta realizada.

También este módulo contará con un servicio que será el encargado de consultar los datos personales de los remitentes de los comprobantes.

A continuación, se presentan algunos requisitos y prototipos que fueron proporcionados al practicante.

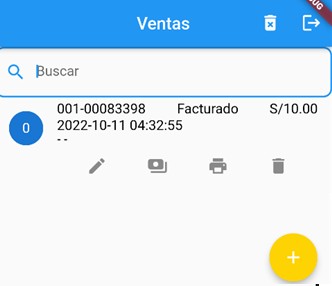
### Requisitos del módulo Ventas

***Tabla 9:Prototipo del módulo ventas***

|  |  |  |  |
| --- | --- | --- | --- |
| **Modulo** | **Código** | **Descripción** | **Prioridad** |
| Ventas | RFDP01 | Listar las ventas. | Alta |
| RFDP02 | Eliminar Ventas. | Alta |
| RFDP03 | Editar las ventas. | Alta |
| RFDP04 | Búsqueda de clientes por servicios web. | Alta |
| RFDP05 | Búsqueda de ventas por correlativo, cliente. | Alta |
| RFDP06 | Selección de productos mediante lectura de  código de barra, usando la cámara del dispositivo | Alta |
| RFDP07 | Impresión de comprobante, usando sunmi  printer | Media |
| RFDP08 | Pagos de comprobante, por partes o contado. | Baja |
| RFDP08 | Auto seleccionar valores en las entradas de  datos. | Baja |
| RFDP09 | Calcular descuento de una venta. | Baja |
| RFDP10 | Listado de productos con imagen | baja |

Fuente: Lead working partner SA

### Prototipos del módulo Ventas

***Figura 64:Prototipo- listar ventas.***

Fuente: Lead Working Partner SA

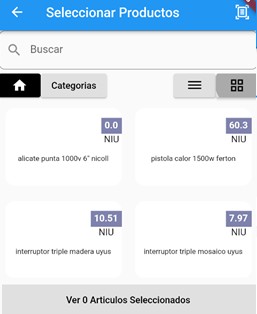
La figura 64 muestra el prototipo de la lista de ventas, cada ítem tendrá las opciones de editar, pagar, imprimir y eliminar. Para crear una nueva venta el usuario debe presionar en el botón amarillo, una vez presionado se le mostrara la lista de productos que ofrece a la venta la empresa para que el usuario pueda escoger como se muestra en la figura 65, también se le ofrece una vista por imágenes de los productos como se muestra en la figura 66 .

***Figura 65:Prototipo -selección de productos para una venta.***



Fuente: Lead Working Partner SA

***Figura 66:Prototipo lista de productos con imagen.***



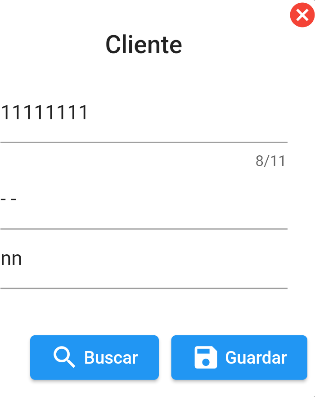
Fuente: Lead working partner SA

***Figura 67:Prototipo- detalles de venta.***



Fuente: Lead Working Partner SA

La figura 67 muestra el prototipo de la vista de detalles de la venta, para ello los importes, el descuento y el total deben calcularse automaticamente, la figura 68 muestra el prototipo de selección del cliente ya sea por dni o ruc.

***Figura 68: Prototipo de selección de cliente de una venta.***

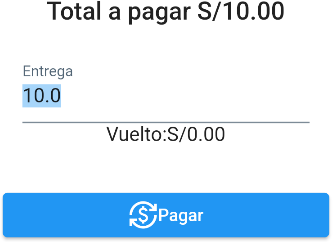
Fuente: Lead Working Partner SA

***Figura 69:Prototipo-pagar una venta.***

Fuente: Lead working partner SA

La figura 69 muestra el prototipo de pagar una venta, el usuario deberá pagar en efectivo, además el aplicativo deberá calcular el vuelto en caso sea necesario como se muestra en la figura 70.

***Figura 70:Prototipo-pagar una venta.***



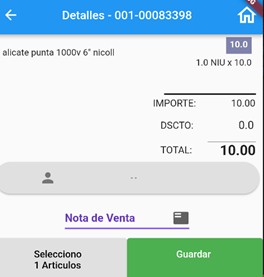
Fuente: Lead working partner SA



***Figura 71:Prototipo- imprimir venta.***

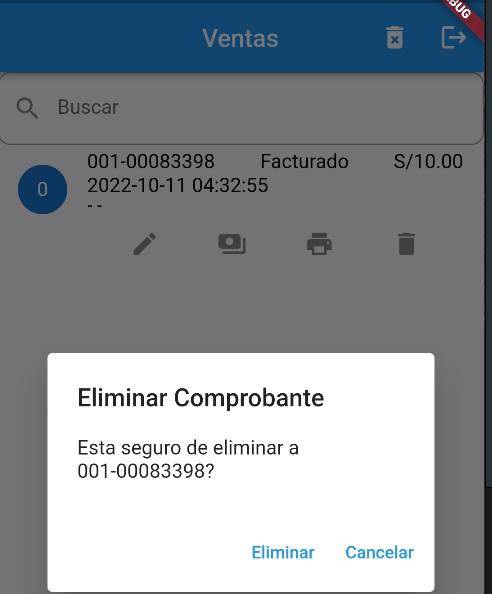
Fuente: Lead Working Partner SA

La figura 71 muestra el prototipo de impresión de una venta, este tendrá dos opciones imprimir o crear una nueva venta, la figura 72 muestra el prototipo de edición de una venta, este mostrará al usuario le vista de detalles directamente, mostrando en la cabecera el número de venta que está editando.

***Figura 72:Prototipo de editar una venta.***

Fuente: Lead Working Partner SA

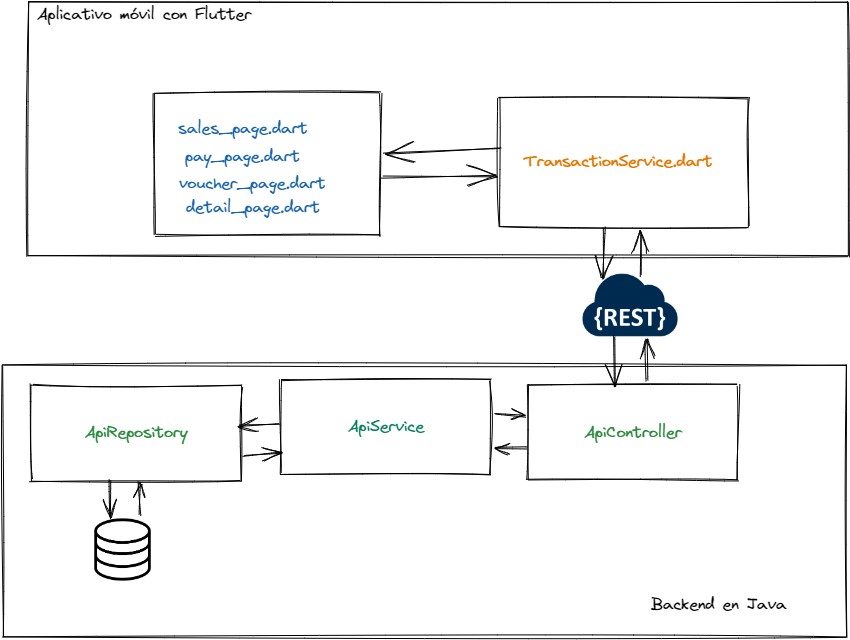
La figura 73, muestra el prototipo de eliminar una venta, se podrá eliminar siempre y cuando sea una nota de venta.

***Figura 73:Prototipo - Eliminar venta.***

Fuente: Lead working partner SA

### Construcción del Ventas

***Figura 74:Flujo de trabajo para la construcción del módulo ventas.***



Fuente: Elaboración propia

La figura 74 muestra el flujo de trabajo que se siguió para la construcción del módulo ventas, se inició creando el archivo **sales\_page.dart**, quien será el modelo con

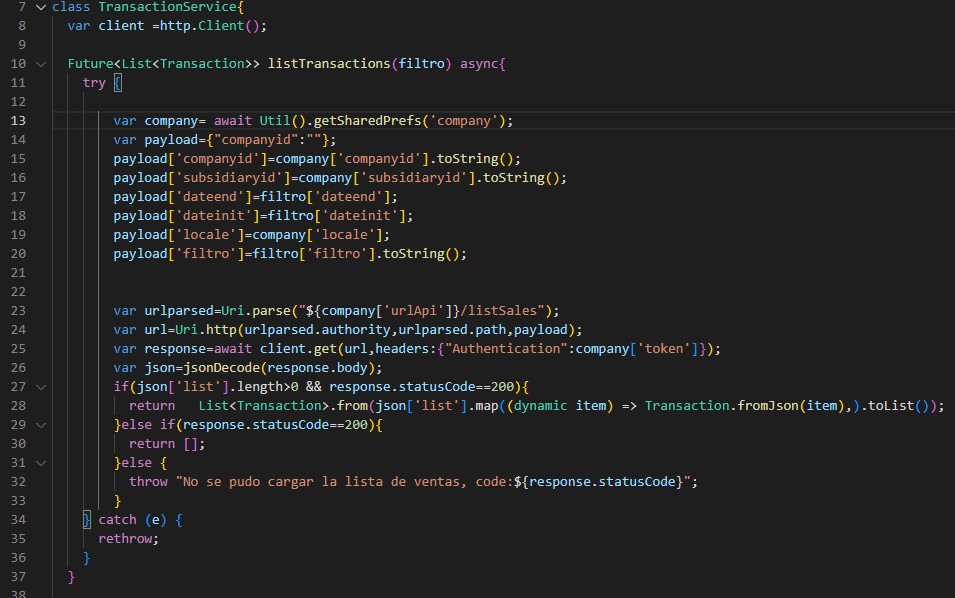
el que el usuario interactuará como se muestra en la figura 75.

***Figura 75:Fragmento de código -lista de ventas.***

Fuente: Elaboración propia

La figura 76 muestra el servicio que provee de datos al modelo de la lista ventas.

***Figura 76:Fragmento de código del servicio - listar ventas.***

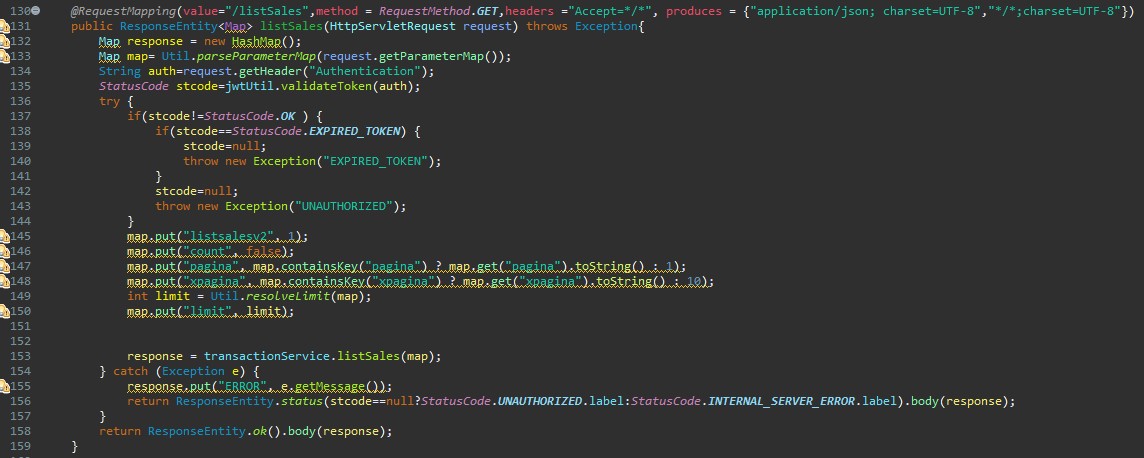


Fuente: Elaboración propia

Luego el controlador del Backend recibirá la petición validando si es de una fuente confiable, para luego pasarlo al servicio y finalmente al repositorio quien será el

encargado de resolver satisfactoriamente o un error en caso contrario, como se muestra en la figura 77,78,79.

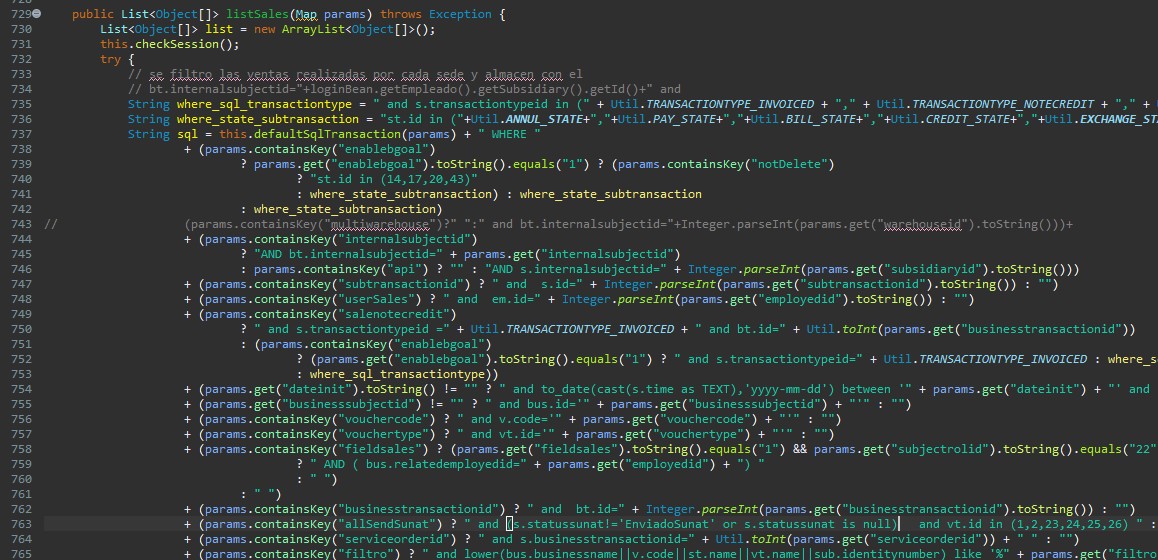
***Figura 77:Fragmento de código del controlador del Backend- listar ventas.***



Fuente: Elaboración propia

***Figura 78:Fragmento de código del servicio del Backend - listar ventas***

Fuente: Elaboración propia

***Figura 79:Fragmento de código del repositorio del Backend- listar ventas.***

Fuente: Elaboración propia

## Modulo Pedidos

El módulo de pedidos será el encardo de mostrar una orden previa a realizar una venta, servirá como una alternativa a un comprobante, para que cuando el cliente se sienta seguro de realizar la compra, el pedido directamente se pueda transformar en un comprobante. Cabe resaltar que este módulo también estará sincronizado con la aplicación web

De igual forma para el desarrollo de este módulo se reutilizarán componentes genéricos que se desarrollaron anteriormente.

A continuación, se presentan algunos requisitos y prototipos que fueron proporcionados al practicante.

### Requisitos del módulo Pedidos

***Tabla 10:Requisitos del módulo pedidos.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Modulo** | **Código** | **Descripción** | **Prioridad** |
| Pedidos | RFDP01 | Listar los pedidos. | Alta |
| RFDP02 | Eliminar pedidos. | Alta |
| RFDP03 | Editar los pedidos. | Alta |
| RFDP04 | Búsqueda de clientes por  servicios web. | Alta |
| RFDP05 | Búsqueda de pedidos por correlativo, cliente. | Alta |
| RFDP06 | Selección de productos mediante lectura de código de barra, usando la cámara del  dispositivo | Alta |
| RFDP07 | Impresión de pedido, usando  sunmi printer | Media |

Fuente: Lead working Partner SA

### Prototipos del módulo Pedidos

***Figura 80:Prototipo - lista de pedidos.***

Fuente: Lead Working Partner SA

La figura 80 muestra el prototipo de la lista de pedidos, cada ítem deber contar con 3 opciones las cuales serán editar, imprimir y eliminar, la figura 81 muestra el prototipo de lista productos a seleccionar para el pedido, y en la figura 82 muestra los detalles del pedido, donde se calculan los totales y descuentos si lo hubiera.

***Figura 81:Lista de productos a seleccionar.***



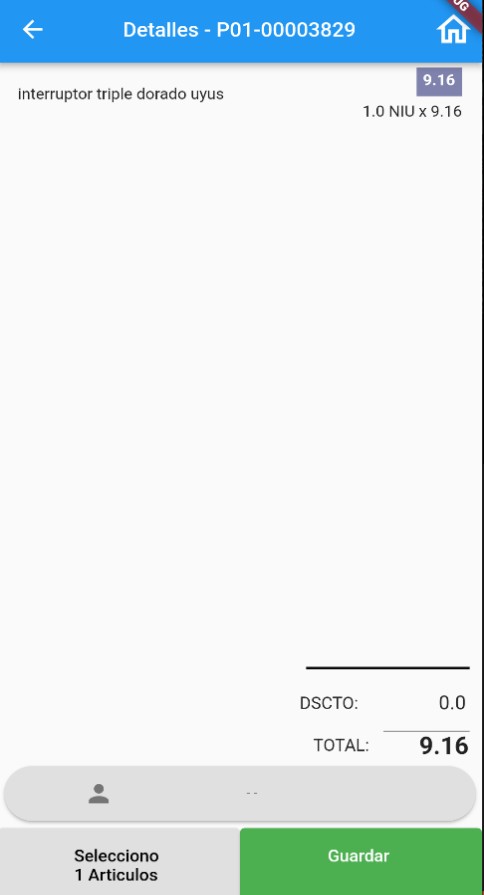
Fuente: Lead Working Partner SA

***Figura 82:Prototipo detalle de pedidos.***

Fuente: Lead Working Partner SA

La figura 83 muestra el prototipo de edición de pedidos, este lleva el correlativo en la parte superior con la finalidad de identificar que pedido está editando.

***Figura 83: Prototipo de edición de pedidos.***

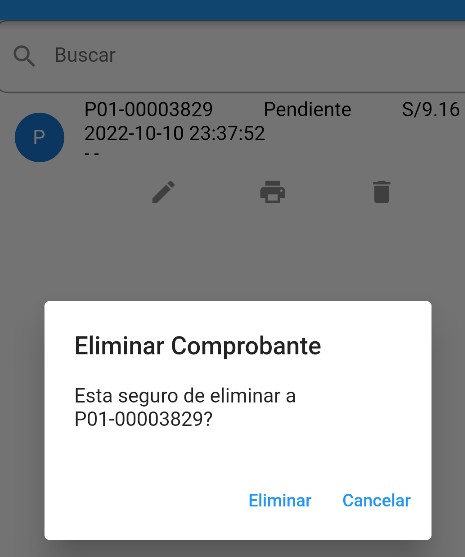


Fuente: Lead Working Partner SA

La figura 84 muestra el prototipo de impresión de pedido, el cual contara con dos opcion, crear un nuevo pedido o imprimir, la figura 85 muestra el protipo de eliminar un pedido, para ello se le mostrara un dialogo para que el usuario confirme la accion.

***Figura 84:Prototipo de impresión de pedido.***

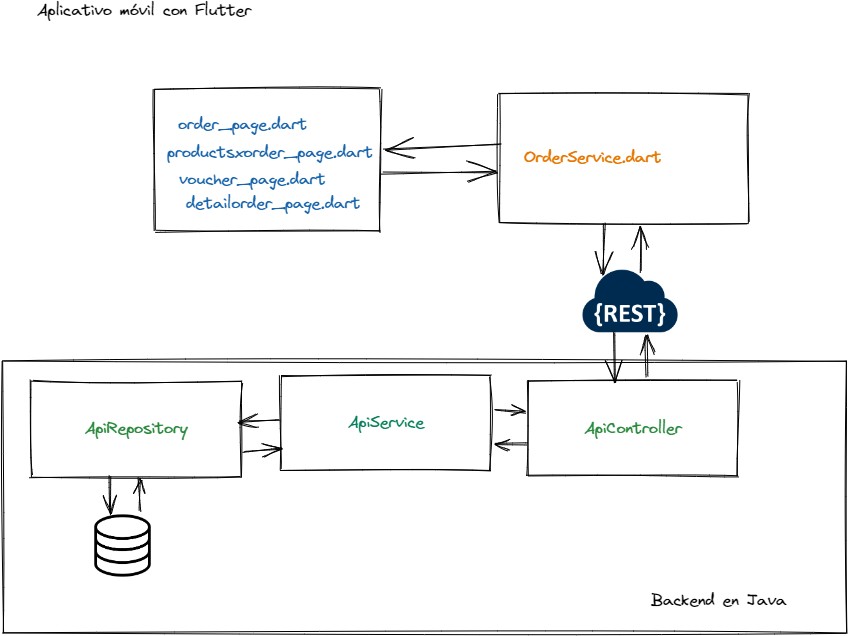
Fuente: Lead Working Partner SA

***Figura 85:Prototipo de eliminar pedido.***

Fuente: Lead working partner SA

### Construcción del módulo Pedido

***Figura 86: Flujo de trabajo para la construcción del módulo pedidos.***



Fuente: Elaboración propia

La figura 86 muestra el flujo de trabajo que se siguió para construir el módulo de pedidos, para ello se comenzó con el archivo **order\_page.dart**, quien es el modelo con vista que interactuara con el usuario como se muestra en la figura 87.

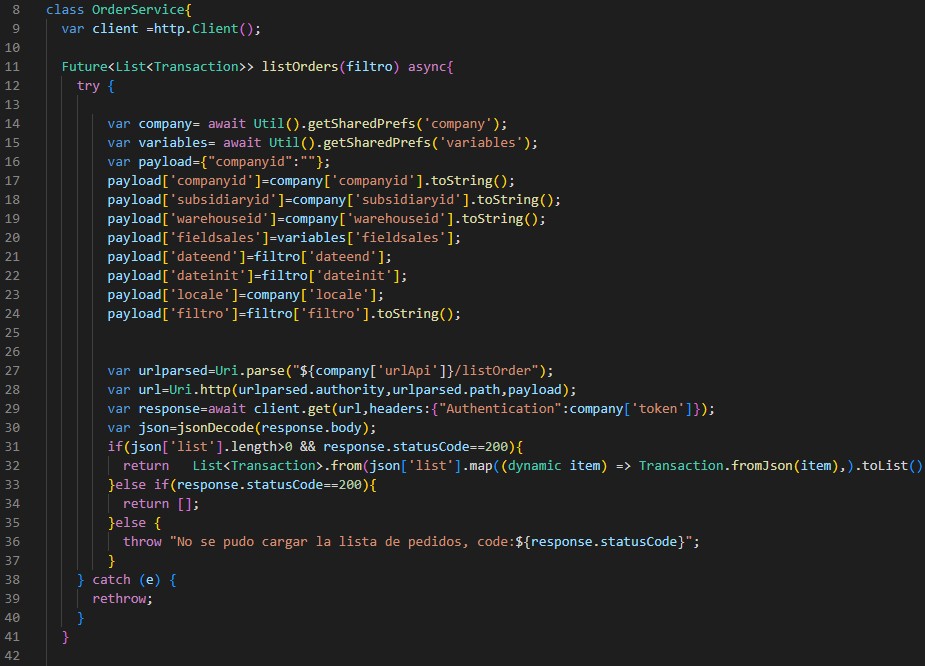
***Figura 87:Fragmento de código -lista de pedidos.***



Fuente: Elaboración propia

Para proveer de los datos necesarios al modelo, se creó un servicio quien se comunica con el Backend para solicitar los datos, como se muestra en la figura 88.

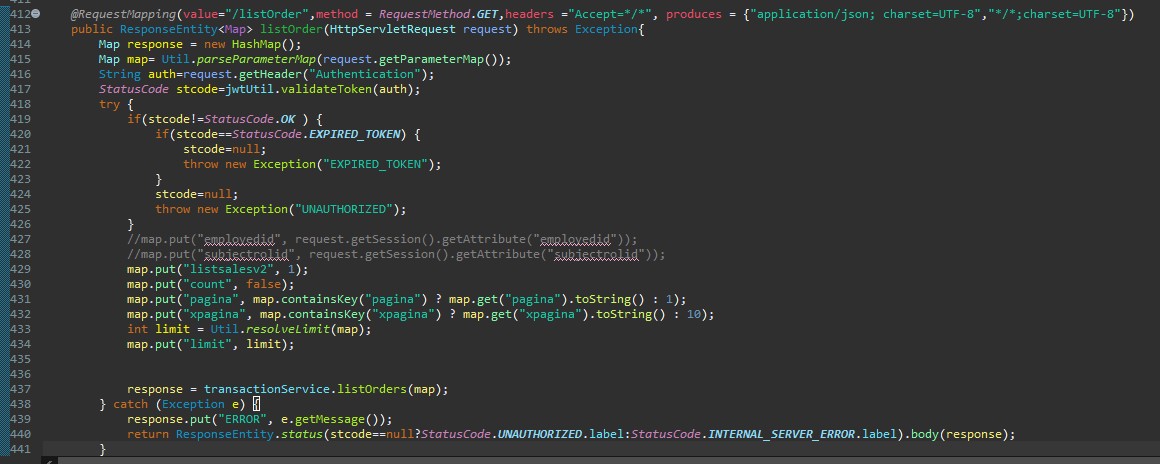
***Figura 88: Fragmento de código del servicio - listar pedidos.***



Fuente: Elaboración propia

Luego el controlador recibe la petición y valida si es una fuente confiable, para luego pasarle al servicio y finalmente al repositorio, como se muestra en la figura 89,90,91.

***Figura 89:Fragmento de código del controlador del Backend -listar pedidos.***

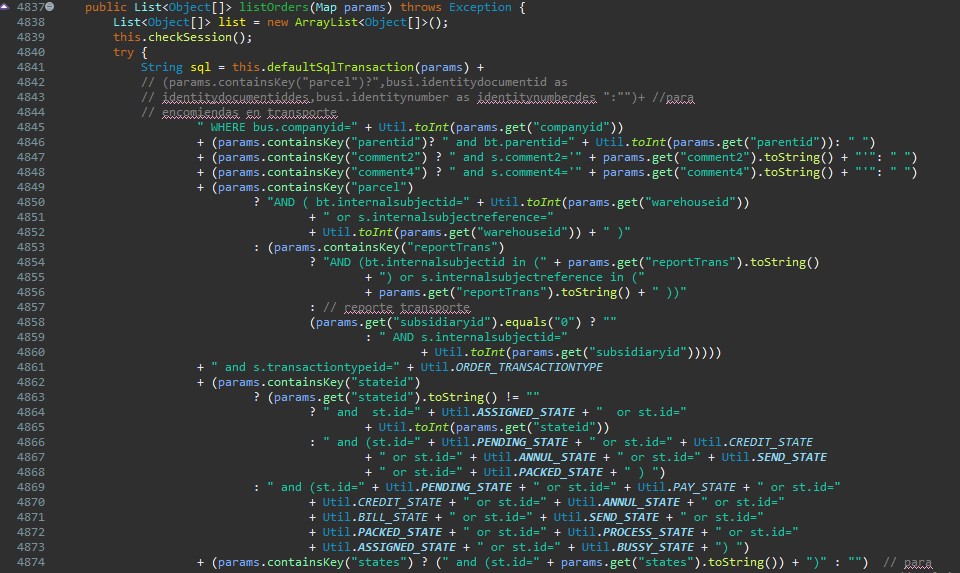


Fuente: Elaboración propia

***Figura 90:Fragmento de código del servicio del Backend - listar pedidos***



Fuente: Elaboración propia

***Figura 91:Fragmento de código del repositorio del Backend - listar pedidos***

Fuente: Elaboración propia

# CONCLUSIONES

Se logro construir el aplicativo móvil Donca erp con los módulos planteados, además del uso de buenas prácticas del desarrollo de software y código limpio.

Se logro construir el módulo de configuraciones, aplicando las buenas prácticas de desarrollo de software, así como también los principios de código limpio.

Se logro construir el módulo de inicio de sesión, usando las buenas prácticas de desarrollo de software, así como también los principios de código limpio.

Se logro construir el módulo de categorías, usando las buenas prácticas de desarrollo de software, así como también los principios de código limpio.

Se logro construir el módulo de marcas, usando las buenas prácticas de desarrollo de software, así como también los principios de código limpio.

Se logro construir el módulo de caja, usando las buenas prácticas de desarrollo de software, así como también los principios de código limpio.

Se logro construir el módulo de ventas, usando las buenas prácticas de desarrollo de software, así como también los principios de código limpio.

Se logro construir el módulo de pedidos, usando las buenas prácticas de desarrollo de software, así como también los principios de código limpio.

# RECOMENDACIONES

Se recomienda utilizar funciones como servicios (faaS), para implementar los servicios que son menos utilizados o que contienen una sola responsabilidad.

Se recomienda refactorizar algunas funciones en los controladores del Backend que no valida si la petición es de una fuente confiable.

Se recomienda trabajar con los principios de código limpio y las buenas prácticas de software para mejorar la legibilidad del código fuente.

Se recomienda el uso de plataformas como Amazon web Service amplify o app Service de Microsoft Azure para el despliegue de aplicaciones móviles.

# BIBLIOGRAFÍA

Abugabah, A., & Sanzogni, L. (2010). Enterprise resource planning (ERP) system in higher education: A literature review and implications. *World Academy of Science, Engineering and Technology*, *71*, 49–53.

Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. *Proceedings - 39th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2013*, 9–16. <https://doi.org/10.1109/SEAA.2013.28>

Bortenschlager, M., Codelli, D., & Willmott, S. (2020). *The API owner ’ s manual Best practices of successful API teams*.

*Dart programming language | Dart*. (n.d.). Retrieved September 13, 2022, from <https://dart.dev/>

Filipova, O., & Vilão, R. (2018). Software Development From A to Z. In *Software Development From A to Z*. <https://doi.org/10.1007/978-1-4842-3945-2>

*Flutter - Build apps for any screen*. (n.d.). Retrieved September 13, 2022, from <https://flutter.dev/>

García-Holgado, A., & García-Peñalvo, F. J. (2016). Architectural pattern to improve the definition and implementation of eLearning ecosystems.

*Science of Computer Programming*, *129*, 20–34. <https://doi.org/10.1016/j.scico.2016.03.010>

Gerasimov, A., Heuser, P., Ketteniß, H., Letmathe, P., Michael, J., Netz, L., Rumpe, B., & Varga, S. (2020). Generated enterprise information systems: MDSE for maintainable co-development of frontend and backend. *CEUR Workshop Proceedings*, *2542*, 22–30.

Ieee Computer Society. (2014). *Guide to the Software Engineering Body of Knowledge Version 3.0 (SWEBOK Guide V3.0)*.

Jacobson, I., & Stimson, R. (2017). Escaping Method Prison – On the Road to Real Software Engineering. In *The Essence of Software Engineering*. <https://doi.org/10.1007/978-3-319-73897-0_3>

Kniberg, H., & Skarin, M. (2009). Kanban and Scrum making the most of both.

*Crisp*, *1*, 1–41.

McCormick, M. (2012). Waterfall and Agile Methodology. *MPCS Inc*, *8/9/2012*, 1–8.

Oracle. (1995). *¿ Qué es la tecnología Java y para qué la necesito ? He leído las condiciones de Java Virtual Machine y de JVM . ¿ Se trata de software de*?

*PostgreSQL: Documentation: 14: PostgreSQL 14.4 Documentation*. (n.d.). Retrieved June 16, 2022, from <https://www.postgresql.org/docs/current/>

Pressman, R. S. (2010). Software Quality Engineering: A Practitioner’s Approach. In *Software Quality Engineering: A Practitioner’s Approach* (Vol. 9781118592). <https://doi.org/10.1002/9781118830208>

*Qué es el control de versiones | Atlassian Git Tutorial*. (n.d.). Retrieved June 16, 2022, from https:/[/www.atlassia](http://www.atlassian.com/es/git/tutorials/what-is-version-control)n[.com/es/git/tutorials/what-is-version-control](http://www.atlassian.com/es/git/tutorials/what-is-version-control)

Richards, M. (2015). Software Architecture Patterns - Understanding Common Architecture Patterns and When to Use Them. In *Encyclopedia of Database Systems*.

*Spring | Projects*. (n.d.). Retrieved June 16, 2022, from https://spring.io/projects

*What is REST? | Codecademy*. (n.d.). Retrieved September 26, 2022, from https://[www.codecademy.com/article/what-is-rest](http://www.codecademy.com/article/what-is-rest)