

MANUAL TÉCNICO

FEDEFUN (PIQUEFU)

**PROYECTO FINAL LABORATORIO
DE SOFTWARE**

Junio del 2020

Universidad Tecnológica de Pereira

CONTENIDO

INTRODUCCIÓN	2
PROPÓSITO	3
OBJETIVOS	3
CARACTERÍSTICAS MÍNIMAS DEL ENTORNO	3
SOBRE ESTE MANUAL	3
ARQUITECTURA Y DISEÑO DEL CÓDIGO	4
DIAGRAMAS DEL PROYECTO	4
DIAGRAMAS ESTRUCTURALES	4
DIAGRAMAS COMPORTAMENTALES	7
ESTRUCTURA DEL CÓDIGO	11
FRONTEND	12
BACKEND	15
REQUERIMIENTOS Y DEPENDENCIAS	16
CONTACTO	18

TABLA DE IMÁGENES

Imagen 1: Diagrama de paquetes	6
Imagen 2: Diagrama de componentes	7
Imagen 3: Diagrama de clases	8
Imagen 4: Diagrama de casos de uso	9
Imagen 5: Diagrama de actividades general	10
Imagen 6: Diagrama de actividades del vendedor	11
Imagen 7: Diagrama de actividades del supervisor	12
Imagen 8: Diagrama de actividades del administrador	13
Imagen 9: Repositorio principal	14
Imagen 10: Frontend principal	14
Imagen 11: Carpeta src	15
Imagen 12: Carpeta components	16
Imagen 13: Carpeta admin	16
Imagen 14: Carpeta routes	17
Imagen 15: Carpeta supervisor	17
Imagen 16: Carpeta vendedor	18
Imagen 17: Backend principal	18
Imagen 18: Carpeta application	19
Imagen 19: Dependencias frontend	20

INTRODUCCIÓN

El software Fedefun se propone como un apoyo en forma de servicio a las diferentes empresas a nivel local y nacional, sirviendo como refuerzo en la toma de facturas de sus vendedores, además de su óptima visualización y gestión en los tres principales módulos del sistema. El software se desarrolla en los lenguajes de programación python y javascript y es usable en la web, en dispositivos móviles y en el escritorio de un sistema operativo.

PROPÓSITO

El software Fedefun tiene como propósito reflejar los conocimientos adquiridos a través del estudio de la carrera ingeniería de sistemas y computación mediante las diferentes técnicas de ingeniería de software, lógica difusa y paradigmas de programación aprendidas en los semestres anteriores. Se sintetiza toda la información en un proyecto compacto pero conciso que refleja un reto en su implementación en diferentes aspectos y se presenta como entrega final para la materia laboratorio de software.

OBJETIVOS

- Implementar un sistema que cumpla con los requerimientos iniciales del proyecto de la materia laboratorio de software
- Reflejar las actividades de análisis y diseño en un software funcional.
- Sintetizar los conocimientos de las diferentes áreas del conocimiento de la ingeniería de sistemas y computación en el desarrollo de este aplicativo.

CARACTERÍSTICAS MÍNIMAS DEL ENTORNO

Requisitos mínimos de hardware y software

- Sistema operativo windows 7 o versiones superiores.
- Android versión 7.0 o superior.
- IOs versión 12 o superior.
- Chrome versión 23 o superior.
- Mozilla versión 21 o superior.
- Procesador de 1.6GHz o superior
- Memoria RAM de 2Gb o superior

SOBRE ESTE MANUAL

Este manual se propone con el fin de entender el diseño, arquitectura y estructura del código del software. Las explicaciones se basan en la documentación abstraída en la fase de análisis y diseño y en la estructura general del código del repositorio de github del proyecto. No se explica el código desarrollado sino su ubicación y cómo actúa.

ARQUITECTURA Y DISEÑO DEL CÓDIGO

A continuación se presentan los siguientes diagramas UML que reflejan el diseño inicial del software Fedefun:

DIAGRAMAS DEL PROYECTO

En los siguientes apartados se presentan los diagramas comportamentales y estructurales del proyecto con el fin de definir la arquitectura y el diseño del aplicativo desarrollado.

DIAGRAMAS ESTRUCTURALES

Para la comprensión del aplicativo se proponen los siguientes diagramas estructurales:

DIAGRAMA DE PAQUETES

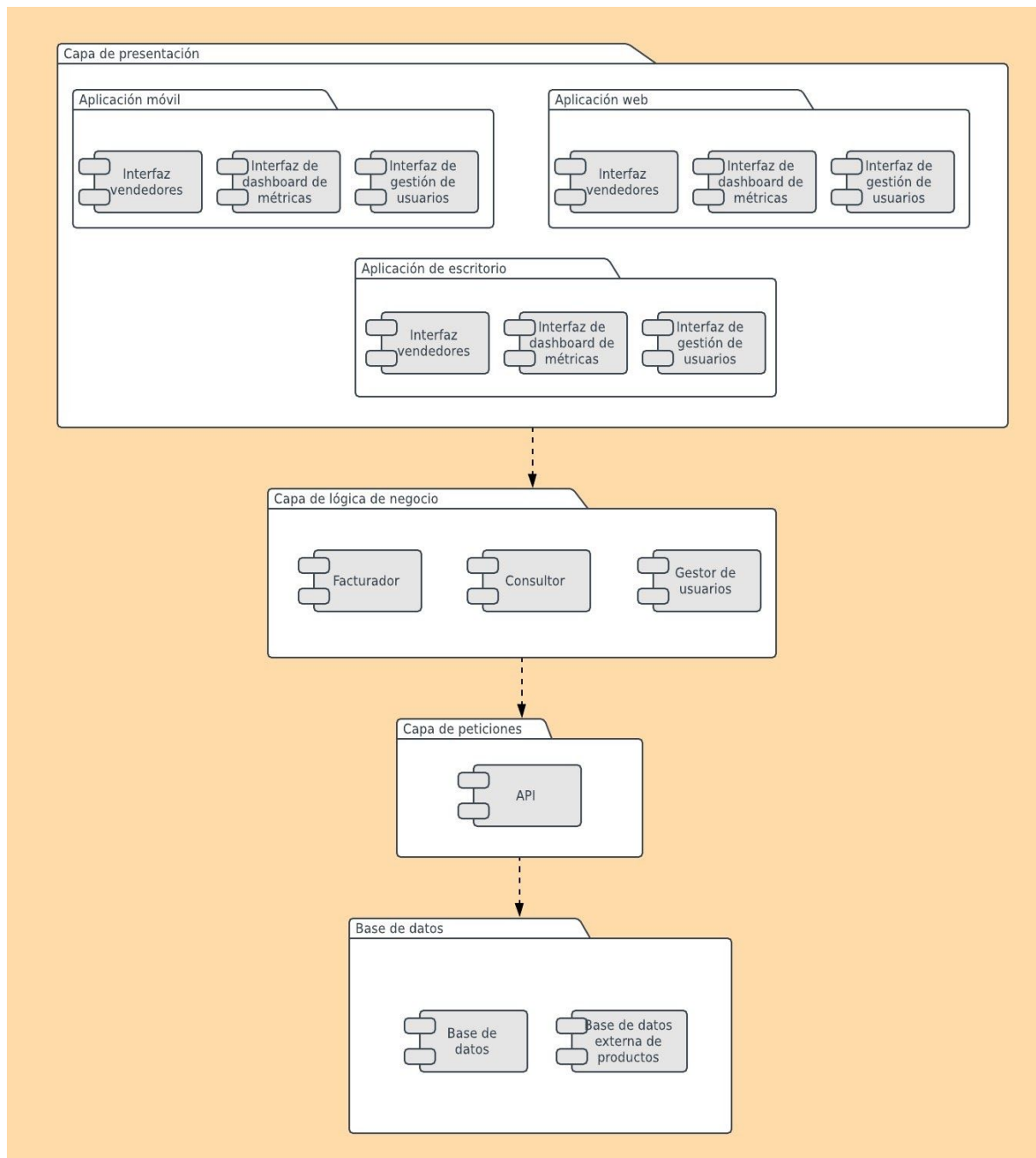


Imagen 1: Diagrama de paquetes

En el diagrama presente se exponen los paquetes del software. Se maneja la capa de presentación como el front-end, la capa lógica de negocio como el back-end que hace peticiones a la API en su respectiva capa la cual conecta con las bases de datos internas y externas para recolectar la información.

DIAGRAMA DE COMPONENTES

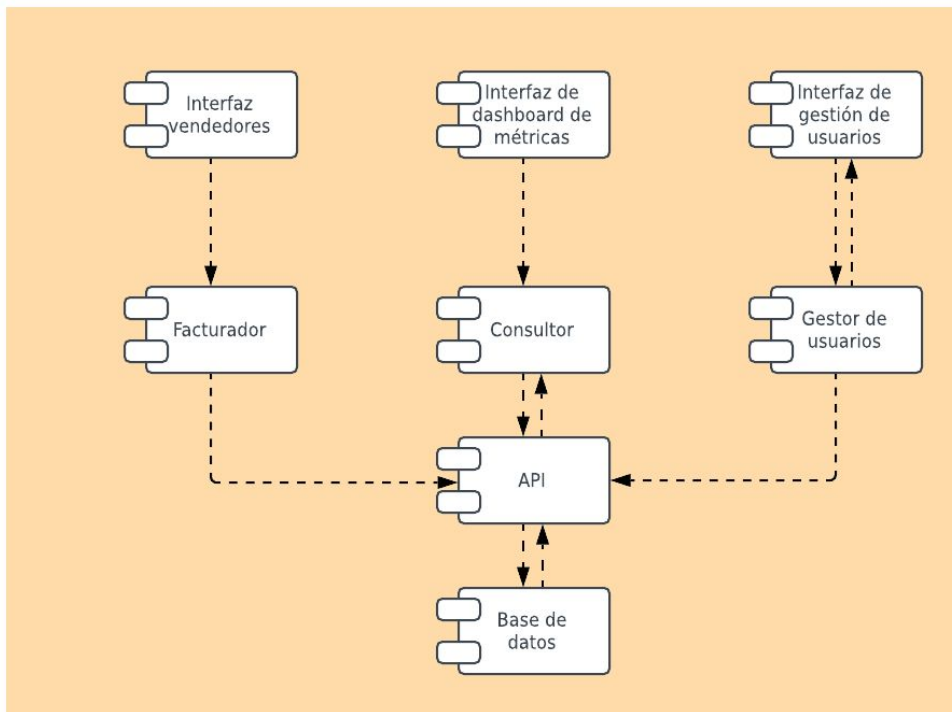


Imagen 2: Diagrama de componentes

En el diagrama presente se exponen los componentes del software. Se explica cómo funciona cada componente del sistema según los paquetes anteriores. Se lee de arriba hacia abajo y muestra el nivel de dependencia de los componentes con las flechas punteadas.

DIAGRAMA DE CLASES

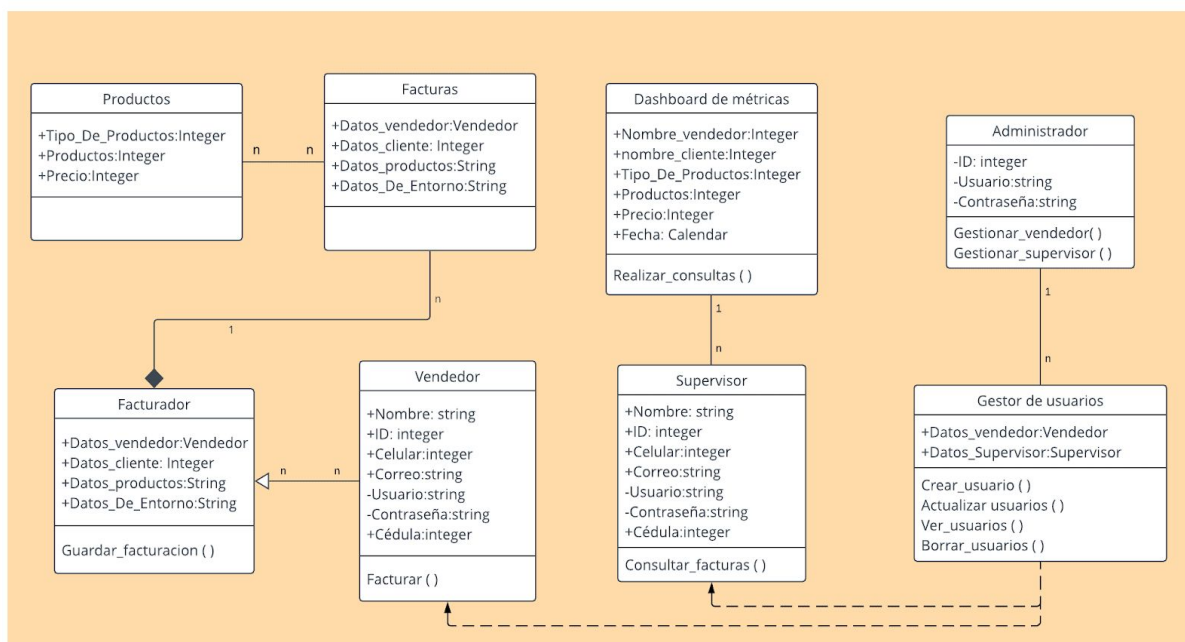


Imagen 3: Diagrama de clases

En el presente diagrama se presentan las clases del aplicativo. Se presenta bajo el mismo esquema anterior mostrando dependencias y demás conexiones que se presentan específicamente en los diagramas estructurales.

DIAGRAMAS COMPORTAMENTALES

Para la comprensión del comportamiento del aplicativo se presenta los siguientes diagramas comportamentales:

DIAGRAMA DE CASOS DE USO

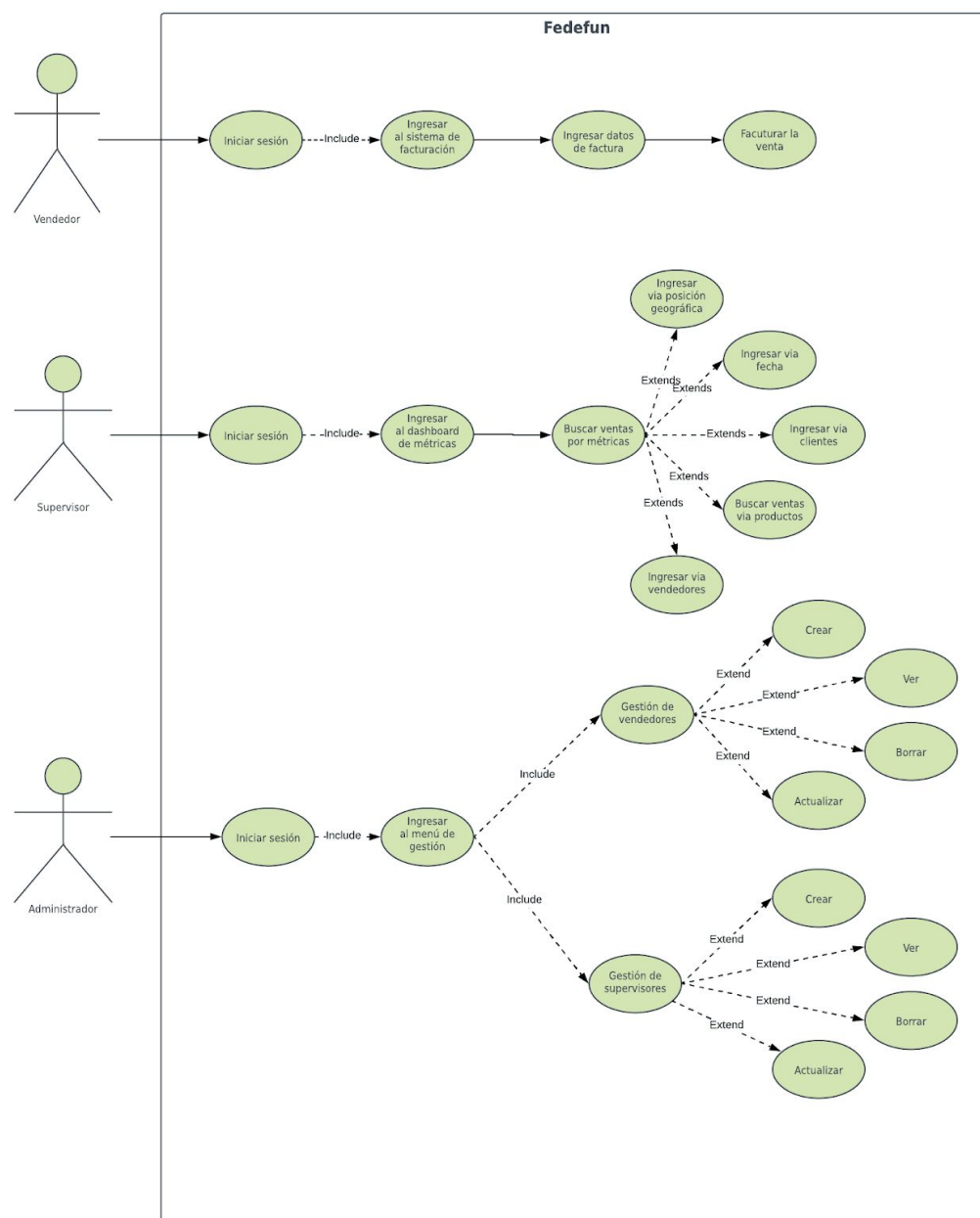


Imagen 4: Diagrama de casos de uso

En el presente diagrama se expone de manera general como cada tipo de usuario interactúa con el sistema.

DIAGRAMAS DE ACTIVIDADES

En los siguientes diagramas se explica el funcionamiento general y específico del sistema

DIAGRAMA DE ACTIVIDADES GENERAL

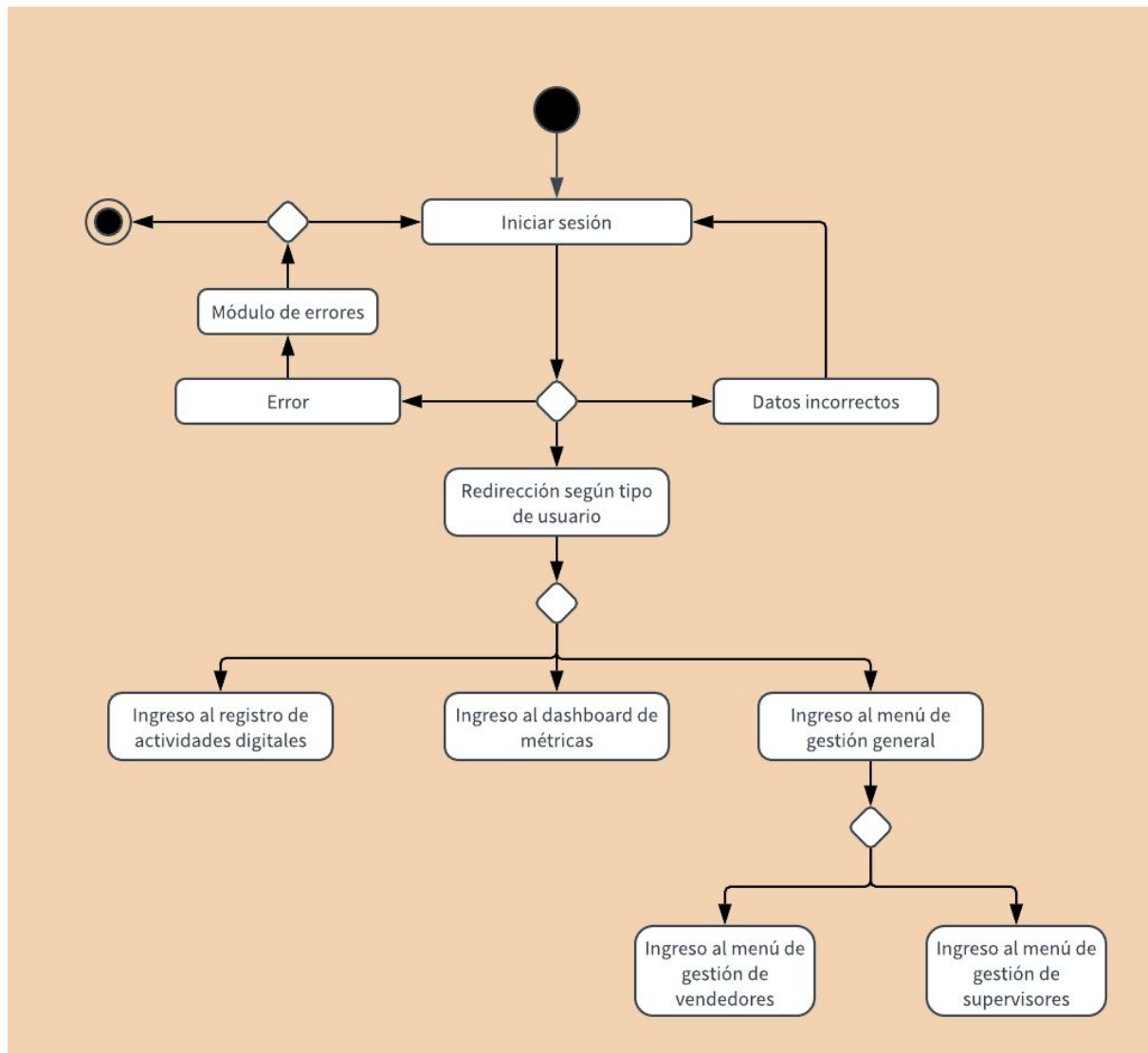


Imagen 5: Diagrama de actividades general

Se expone el diagrama de actividades general del programa, donde se ejecuta el login y los ingresos al resto de vistas/módulos.

DIAGRAMA DE ACTIVIDADES DEL VENDEDOR

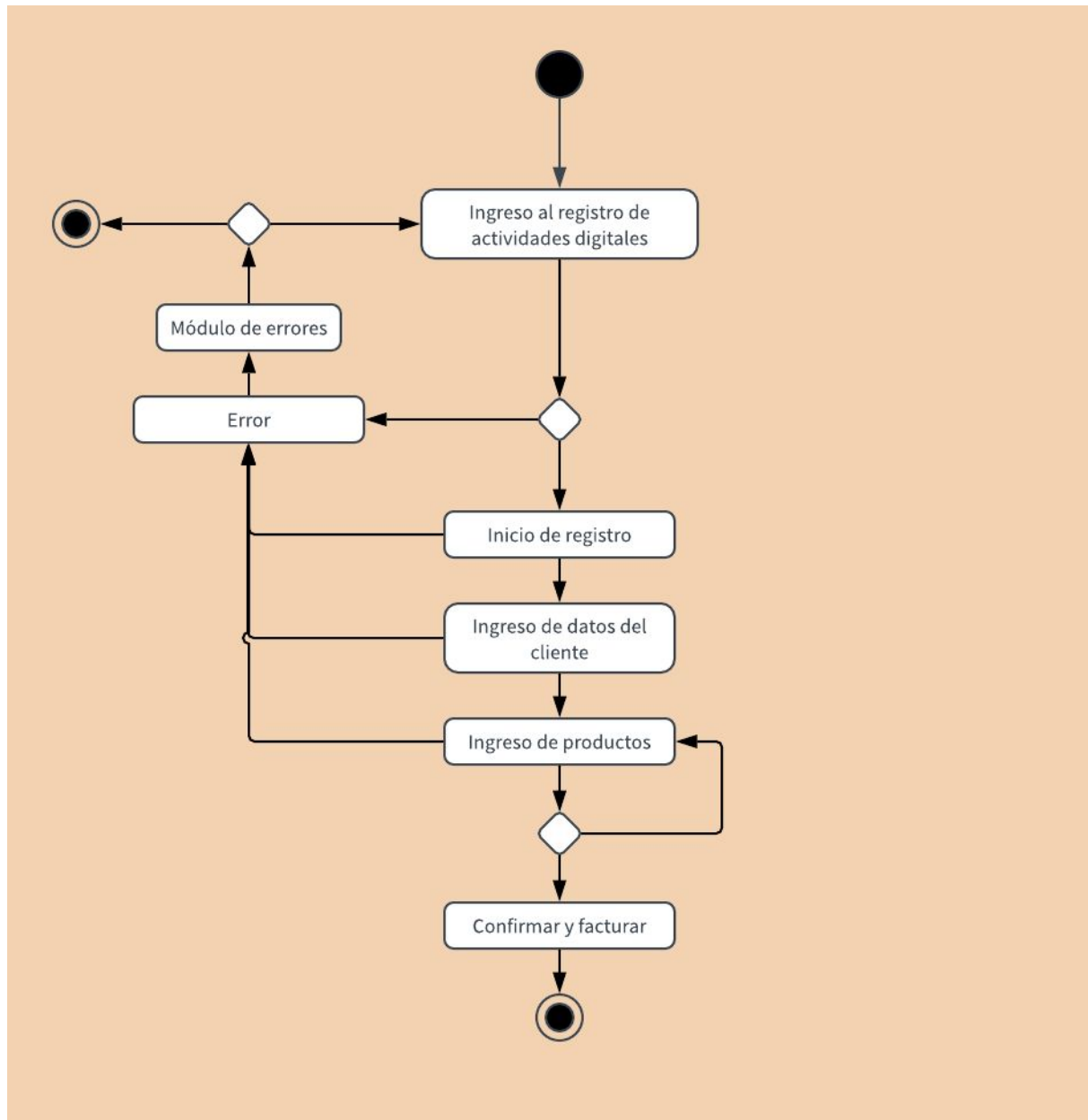


Imagen 6: Diagrama de actividades del vendedor

En el presente diagrama se expone el diagrama de actividades para el vendedor.

DIAGRAMA DE ACTIVIDADES DEL SUPERVISOR

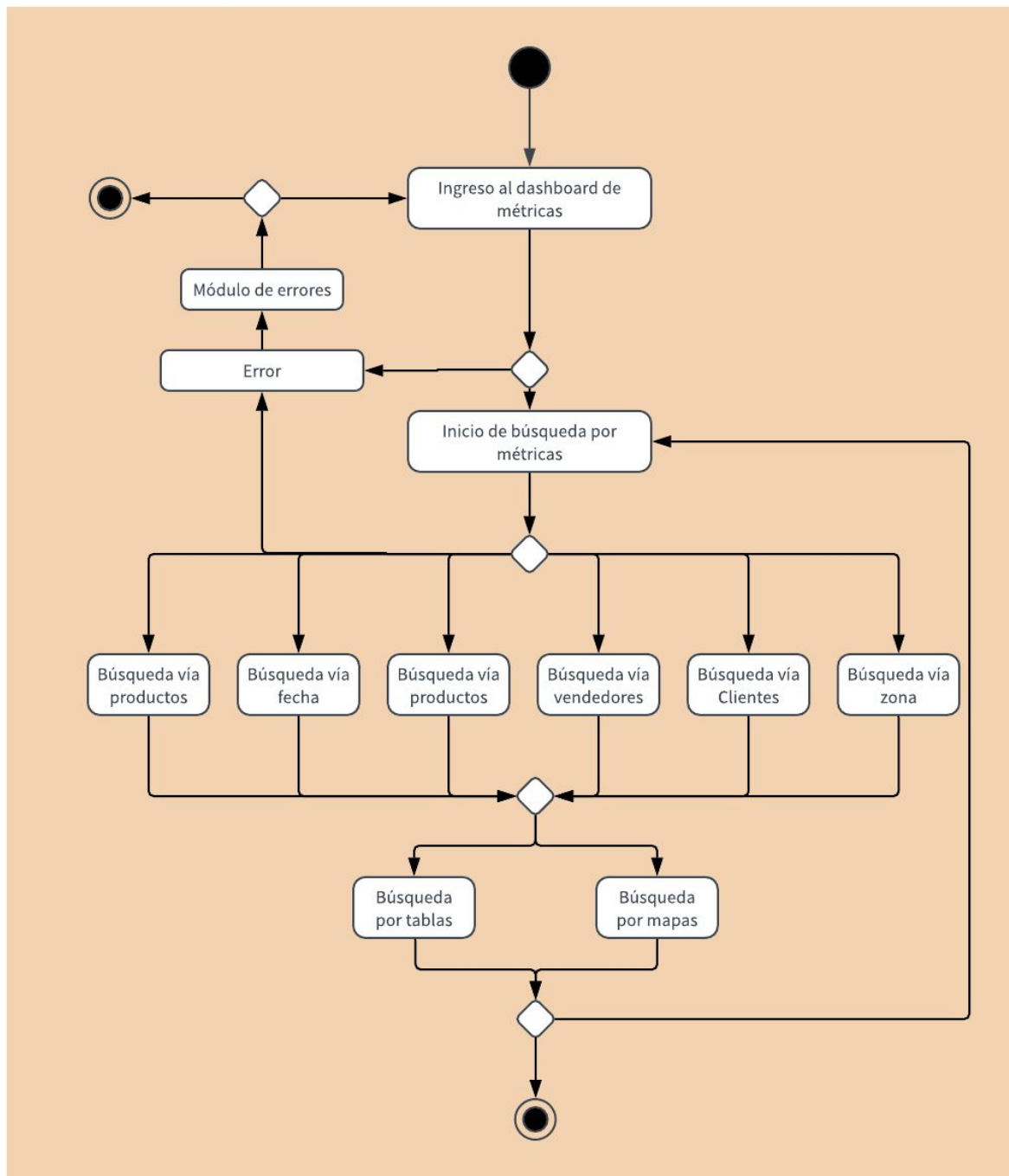


Imagen 7: Diagrama de actividades del supervisor

En el presente diagrama se expone el diagrama de actividades para el supervisor.

DIAGRAMA DE ACTIVIDADES DEL ADMINISTRADOR

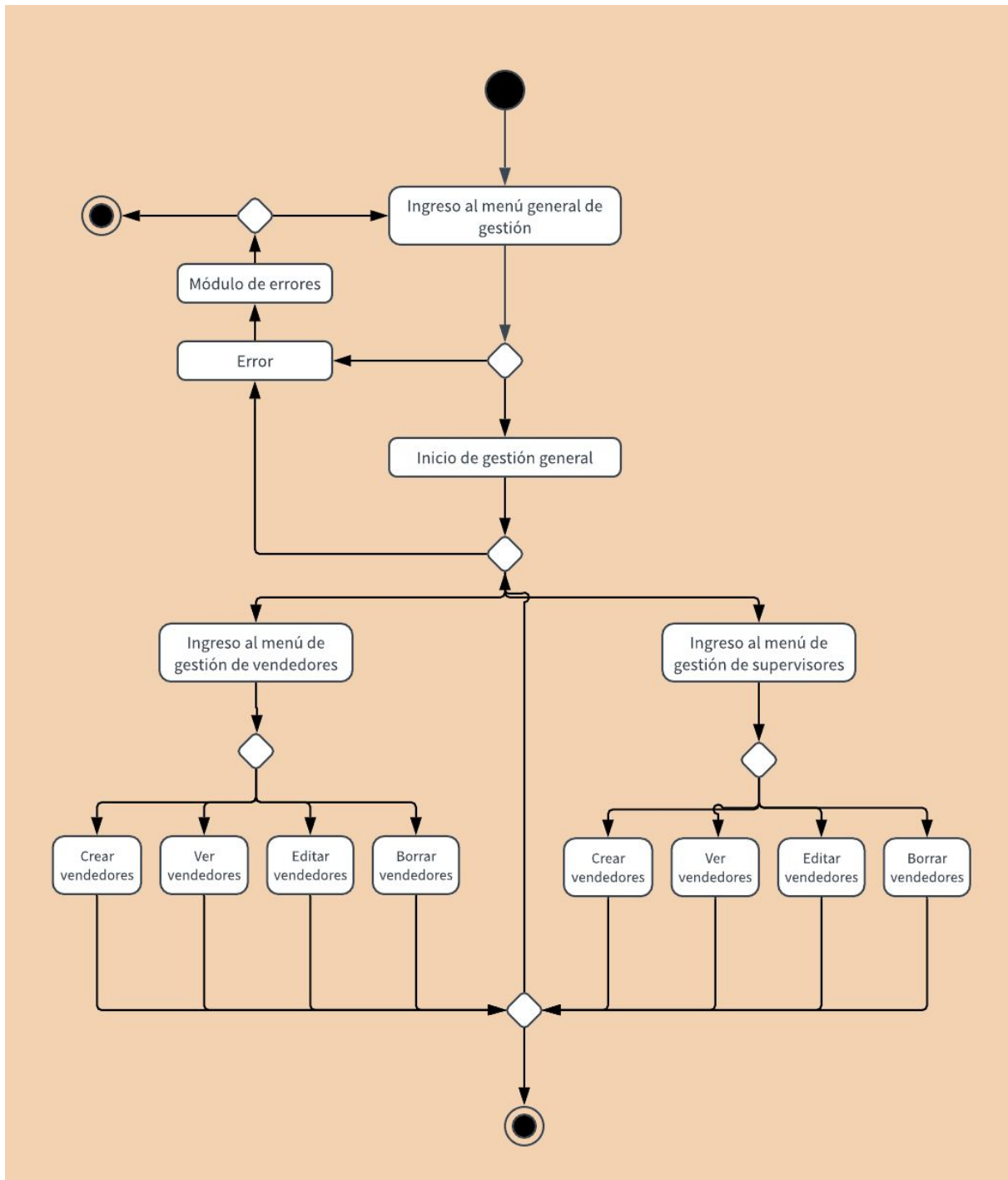


Imagen 8: Diagrama de actividades del administrador

En el presente diagrama se expone el diagrama de actividades para el administrador.

ESTRUCTURA DEL CÓDIGO

En base a los diagramas planteados anteriormente, se desarrolla el aplicativo que se encuentra en el siguiente repositorio:

<https://github.com/Sa-LL/FedeFun---Laboratorio-de-software---2020---1>

A partir del anterior repositorio se analiza la estructura del código.

Aplicación FedeFun - Laboratorio de Software

13 commits

5 branches

0 packages

0 releases


0 contributors

Branch: master

New pull request

Find file

Clone or download

 Santiago Despliegue completo

Latest commit 829a871 6 hours ago

Backend

Desplegado

6 hours ago

Documentación

Actas 13-15

27 days ago

Frontend

Desplegado

6 hours ago

Imagen 9: Repositorio principal

El repositorio presenta 3 carpetas principales llamadas Backend, Frontend y documentación en las cuales se encuentran el diseño, la lógica y los documentos para entender el proyecto. Se inicia en el análisis del front end.

FRONTEND


Branch: master

FedeFun---Laboratorio-de-software---2020---1 / Frontend /

Create new file


Find file

History

 Santiago Desplegado


Latest commit bfee27a 6 hours ago

..

 electron


Desktop app

9 days ago

 public


Desplegado

6 hours ago

 src


Desplegado

6 hours ago

 .env


Cambios en el back, actualizacion de la vista de supervisor y prepara...

4 days ago

 .gitignore


Aplicación versión 1

2 months ago

 README.md


Aplicacion version 1

2 months ago

 package-lock.json

Implementacion del mapa completada

2 days ago

 package.json

Implementacion del mapa completada

2 days ago

Imagen 10: Frontend principal

En el cual se presentan 3 carpetas (electron, public y src) y una serie de archivos los cuales son: .env para las configuraciones de entorno, gitignore para las carpetas o archivos a

ignorar, readme.md para la explicación de la sección y package-lock.json y package.json para las dependencias del sistema.

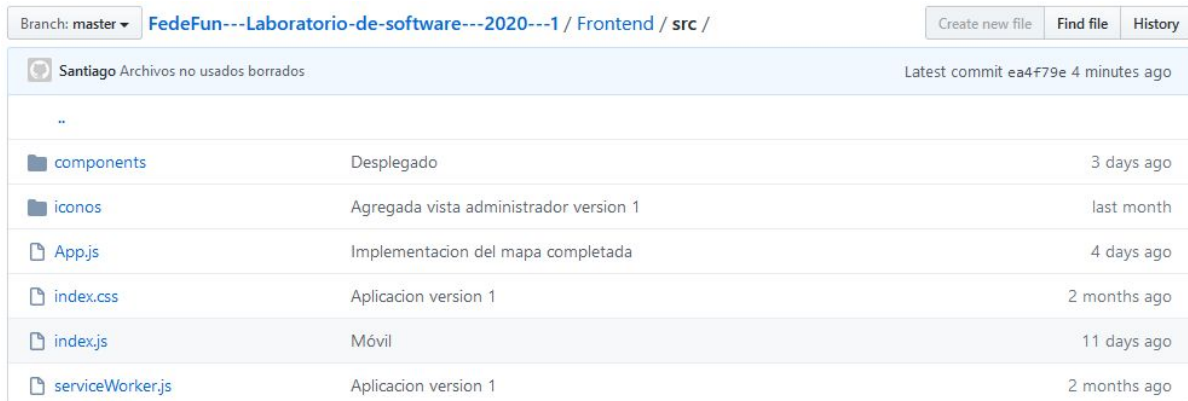
Las carpetas principales contiene el siguiente contenido:

Electron: Contiene el manifiesto para la aplicación de escritorio de Fedefun.

Public: Contiene las variables públicas como documentos o imágenes.

Src: Contiene el código fuente del frontend.

Dentro de Src se encuentran los siguientes elementos:



Branch: master	FedeFun---Laboratorio-de-software---2020---1 / Frontend / src /	Create new file	Find file	History
Santiago Archivos no usados borrados		Latest commit ea4f79e 4 minutes ago		
..				
components	Desplegado	3 days ago		
iconos	Agregada vista administrador version 1	last month		
App.js	Implementacion del mapa completada	4 days ago		
index.css	Aplicacion version 1	2 months ago		
index.js	Móvil	11 days ago		
serviceWorker.js	Aplicacion version 1	2 months ago		

Imagen 11: Carpeta src

Leídos de abajo hacia arriba de la siguiente manera:

ServiceWorker.js: Archivos .js que permite el uso de la aplicación sin conexión a la red, el uso de una aplicación web progresiva (PWA) además de otras funcionalidades.

Index.js: Este archivo renderiza el contenido de app.js.

Index.css: Index del contenido css de la aplicación.

App.js: Esta es la aplicación principal donde se encuentran las rutas de todos los componentes de la carpeta componentes. Se considera la raíz del árbol de componentes.

Íconos: Esta carpeta contiene íconos .png que se usaron para el desarrollo de la aplicación.

components: Carpeta con todos los componentes del frontend.

La carpeta components posee los siguientes elementos:

Branch: master	FedeFun---Laboratorio-de-software---2020---1 / Frontend / src / components /	Create new file	Find file	History
Santiago Desplegado Latest commit bfee27a 3 days ago				
..				
Admin	Desplegado	3 days ago		
Routes	Aplicacion version 1	2 months ago		
Supervisor	Desplegado	3 days ago		
Vendedor	Desplegado	3 days ago		
Copyright.jsx	Agregada vista administrador version 1	last month		
Login.jsx	Desplegado	3 days ago		

Imagen 12: Carpeta components

Leídos de abajo para arriba de la siguiente manera:

Login.jsx: Es la vista completa del login de la aplicación Fedefun (para mayor detalle ver el manual de usuario)

Copyright.jsx Es el componente para el aviso de copyright es todas las pestañas de Fedefun (para mayor detalle ver el manual de usuario)

Vendedor: Contiene toda la vista del tipo de usuario Vendedor.

Supervisor: Contiene toda la vista del tipo de usuario Supervisor.

Routes: Contiene todas las rutas de los componentes.

Administrador: Contiene toda la vista del tipo de usuario Administrador.

La carpeta de administrador contiene los siguientes archivos:

Branch: master	FedeFun---Laboratorio-de-software---2020---1 / Frontend / src / components / Admin /	Create new file	Find file	History
Santiago Desplegado Latest commit bfee27a 3 days ago				
..				
InicioA.jsx	Agregada vista administrador version 1	last month		
Orders.jsx	Desplegado	3 days ago		
Title.jsx	Agregada vista administrador version 1	last month		

Imagen 13: Carpeta admin

Los cuales funcionan para el correcto despliegue y funcionamiento de la aplicación.

La carpeta de routes contiene los siguientes archivos:




Branch: master		FedeFun---Laboratorio-de-software---2020---1 / Frontend / src / components / Routes /		Create new file	Find file	History
 Santiago Aplicacion version 1		Latest commit 0136d26 on 22 Apr				
..						
 ProtectedRoute.jsx	Aplicacion version 1				2 months ago	
 Redirect.jsx	Aplicacion version 1				2 months ago	

Imagen 14: Carpeta routes


Los cuales se encargan del direccionamiento de los componentes del frontend.

La carpeta de supervisor contiene los siguientes archivos:

Branch: master


Create new fileFind fileHistory

FedeFun---Laboratorio-de-software---2020---1 / Frontend / src / components / Supervisor /

 Santiago Desplegado


Latest commit bfee27a 3 days ago

..

 DialogoFactura


Desplegado

3 days ago

 Mapa


Implementacion del mapa completada

4 days ago

 Chart.jsx


Aplicacion version 1

2 months ago

 Deposits.jsx


Aplicacion version 1

2 months ago

 Dialog.jsx


Implementacion del mapa completada

4 days ago

 DialogDate.jsx


Implementacion del mapa completada

4 days ago

 Filtros.jsx


Implementacion del mapa completada

4 days ago

 InicioS.jsx


Desplegado

3 days ago

 ListItems.jsx


Implementacion del mapa completada

4 days ago

 Orders.jsx

Mapa funcional

3 days ago

 Title.jsx

Aplicacion version 1

2 months ago

Imagen 15: Carpeta supervisor


La cual contiene los archivos para el correcto funcionamiento de la vista del supervisor. Contiene 2 carpetas que incluyen el código para las ventanas emergentes de la misma (ver manual de usuario para mayor detalle)

La carpeta de vendedor contiene los siguientes archivos:

Branch: master ▾


Create new fileFind fileHistory

FedeFun---Laboratorio-de-software---2020---1 / Frontend / src / components / Vendedor /

 Santiago Desplegado


Latest commit bfee27a 3 days ago

..

 ListaProductos


Desplegado

3 days ago

 DatosC.jsx


Agregada vista administrador version 1

last month

 DatosV.jsx


Aplicacion version 1

2 months ago

 Factura.jsx


Desplegado

3 days ago

 InicioV.jsx


Desplegado

3 days ago

 ListaProductosV.jsx

Desplegado

3 days ago

 Productos.jsx

Móvil

11 days ago

Imagen 16: Carpeta vendedor

Los cuales contiene todos los archivos para el correcto funcionamiento de la vista de vendedor. La carpeta ListaProductos contiene los productos que se despliegan en el menú de productos (Ver manual de usuario para mayor detalle)

BACKEND

La estructura backend de la aplicación se presenta de la siguiente manera:


Branch: master ▾

FedeFun---Laboratorio-de-software---2020---1 / Backend /

Create new file


Find file

History

 Santiago Desplegado


Latest commit bfee27a 3 days ago

..

 __pycache__


Desplegado

3 days ago

 application


Desplegado

3 days ago

 FedeFun.py


Aplicacion version 1

2 months ago

 Procfile


Desplegado

3 days ago

 config.py

Desplegado

3 days ago

 requirements.txt

Desplegado

3 days ago

Imagen 17: Backend principal

Donde se exponen los siguientes ítems de abajo a arriba:

Requirements.txt: Este archivo incluye todos los requisitos para el correcto funcionamiento del backend en desarrollo.

Config.py: Archivo que configura el entorno del backend.

Procfile: Archivo para la configuración en Heroku.

Fedefun.py: Aplicación principal del backend.

Application: Carpeta principal con el backend.

Pycache: Carpeta con la configuración de python 3.8

Dentro de application se encuentran los siguientes elementos:









Branch: master ▾		FedeFun---Laboratorio-de-software---2020---1 / Backend / application /		Create new file	Find file	History
 Santiago Desplegado		Latest commit bfee27a 3 days ago				
..						
 __pycache__	Desplegado	3 days ago				
 __init__.py	Aplicacion version 1	2 months ago				
 api_adm.py	Desplegado	3 days ago				
 api_res.py	Desplegado	3 days ago				
 auth.py	Cambios en el back, actualizacion de la vista de supervisor y prepara...	6 days ago				
 models.py	Desplegado	3 days ago				
 routes.py	Aplicacion version 1	2 months ago				

Imagen 18: Carpeta application

Leídos de abajo hacia arriba de la siguiente manera:

Routes.py: Contiene todas las rutas del backend.

Models.py: Contiene el modelado de todos los datos que se manejan en la aplicación.

Auth.py: Maneja los permisos y tokenización para los inicios de sesión según los tipos de usuarios (para más detalle ver manual de usuario)

Api_res.py: API para el manejo del módulo de facturación.

Api_adm.py: API para el manejo del módulo de administrador.

Init.py: Programa principal que se encarga de visualizar las variables de entorno del backend.

Pycache: Carpeta con las configuraciones de python 3 necesarias para el back.

REQUERIMIENTOS Y DEPENDENCIAS

Para desplegar el software en ambiente de desarrollo, se requieren los siguientes requerimientos de software:

- Python 2 y 3
- Npm (Sistema de gestión de paquetes de Javascript)
- Editor de texto (Visual studio, Atom, etc)

A partir de los siguientes requerimientos, se necesita instalar o instanciar las siguientes dependencias:

Para el frontend:

```

"dependencies": {
  "@date-io/date-fns": "^1.3.13",
  "@material-ui/core": "^4.9.5",
  "@material-ui/icons": "^4.9.1",
  "@material-ui/lab": "^4.0.0-alpha.55",
  "@material-ui/pickers": "^3.2.10",
  "@material-ui/system": "^4.9.14",
  "@testing-library/jest-dom": "^4.2.4",
  "@testing-library/react": "^9.4.1",
  "@testing-library/user-event": "^7.2.1",
  "axios": "^0.19.2",
  "concurrently": "^5.2.0",
  "date-fns": "^2.14.0",
  "dotenv": "^8.2.0",
  "material-table": "^1.57.2",
  "react": "^16.12.0",
  "react-dom": "^16.12.0",
  "react-map-gl": "^5.2.6",
  "react-resize-observer": "^1.1.1",
  "react-router-dom": "^5.1.2",
  "react-scripts": "3.4.0",
  "typeface-roboto": "0.0.75"
},

```

Imagen 19: Dependencias frontend

Para el backend:

```

aniso8601==8.0.0
bcrypt==3.1.7
cffi==1.14.0
click==7.1.1
DateTime==4.3
Flask==1.1.1
Flask-Bcrypt==0.7.1
Flask-JWT-Extended==3.24.1
Flask-Login==0.5.0
flask-mongoengine==0.9.5
Flask-RESTful==0.3.8
Flask-WTF==0.14.3
gunicorn==20.0.4
itsdangerous==1.1.0
Jinja2==2.11.1
MarkupSafe==1.1.1
mongoengine==0.19.1
pyparser==2.20
PyJWT==1.7.1
pymongo==3.10.1
python-dateutil==2.8.1
pytz==2019.3
six==1.14.0
Werkzeug==1.0.0
WTForms==2.2.1
zope.interface==5.1.0

```

Imagen 20: Dependencias backend

CONTACTO

Para soporte y más información del software puede comunicarse al siguiente correo

sebastian.franco@utp.edu.co

santiago.londono@utp.edu.co

michaelandres.grisales@utp.edu.co

O al teléfono

3110000000