



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

**Desarrollo de una Aplicación móvil de transporte
público “Combi App”**

Curso: “Soluciones Móviles 1”

Docente: *Ing. Patrick Cuadros Quiroga*

Integrantes:

- *LUPACA MAMANI RONAL DANIEL* 2020067146
- *POMA MANCHEGO, RENE MANUEL* 2017057491
- *AGUILAR PINTO, VICTOR ELEAZAR* 2017057405
- *CHINO CONDE, OSWALDO JESUS* 2017057434
- *POMA CHURA, JHON ROMARIO* 2019064022

**Tacna – Perú
2022**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	Lupaca mamani ronal daniel, Poma manchego rene Manuel, Aguilar pinto victor Eleazar, chino conde oswaldo jesus , Jhon Romario Poma Chura			02/12/2023	Versión 1.0
1.5	Lupaca mamani ronal daniel, Poma manchego rene Manuel, Aguilar pinto victor Eleazar, chino conde oswaldo jesus , Jhon Romario Poma Chur			08/12/2023	Ajustes de la implementación del tema
2	Lupaca mamani ronal daniel, Poma manchego rene Manuel, Aguilar pinto victor Eleazar, chino conde oswaldo jesus , Jhon Romario Poma Chur			14/12/2023	Reflexión de impacto e ámbito legal y utilidad

Desarrollo de una Aplicación móvil de transporte público “Combi App” Documento de Arquitectura de Software

Versión2.0

ÍNDICE GENERAL

1. INTRODUCCIÓN	4
1.1. Propósito (Diagrama 4+1)	4
1.2. Alcance	4
1.3. Definición, siglas y abreviaturas	4
1.4. Organización del documento	4
2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS	5
2.1. Priorización de requerimientos	5
2.1.1 Requerimientos Funcionales	5
2.1.2 Requerimientos No Funcionales – Atributos de Calidad	5
2.2. Restricciones	5
3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA	6
3.1. Vista de Caso de uso	6
3.1.1 Diagramas de Casos de uso	6
3.2. Vista Lógica	AD7
3.2.1 Diagrama de Subsistemas (paquetes)	7
3.2.2 Diagrama de Secuencia (vista de diseño)	8
3.2.3 Diagrama de Colaboración (vista de diseño)	9
3.2.4 Diagrama de Objetos	9
3.2.5 Diagrama de Clases	10
3.2.6 Diagrama de Base de datos (relacional o no relacional)	10
3.3. Vista de Implementación (vista de desarrollo)	11
3.3.1 Diagrama de arquitectura software (paquetes)	11
3.3.2 Diagrama de arquitectura del sistema (Diagrama de componentes)	11
3.4. Vista de procesos	11
3.4.1 Diagrama de Procesos del sistema (diagrama de actividad)	11
3.5. Vista de Despliegue (vista física)	11
3.5.1 Diagrama de despliegue	11
4. ATRIBUTOS DE CALIDAD DEL SOFTWARE	12
4.1. Escenario de Funcionalidad	12
4.2. Escenario de Usabilidad	12
4.3. Escenario de confiabilidad	12
4.4. Escenario de rendimiento	12
4.5. Escenario de mantenibilidad	12
4.6. Otros Escenarios	12

1.Introducción

1.1. Propósito (Diagrama 4+1)

El propósito de este proyecto es desarrollar una aplicación móvil y poner en funcionamiento la aplicación “Combi app”, el cual nos permitirá a los usuarios de la ciudad de Tacna ubicar y rastrear los medios de transporte público en tiempo real de tal manera que el objetivo principal es, mejorar la experiencia de los usuarios al brindarles la ubicación en tiempo real del transporte público y por lo cual obtendremos una optimización, eficiencia y comodidad de los desplazamientos diarios que tendría el cada transporte público, también buscamos fomentar y promover una movilidad más sostenible en la ciudad de Tacna

1.2. Alcance

El alcance del proyecto incluirá el desarrollo de una aplicación móvil, donde se realizará el diseño, la programación y la implementación hasta su lanzamiento oficial dentro de la ciudad de Tacna.

La aplicación está planificada para que cuente con funciones que permitan a los usuarios poder tener a su conocimiento y ubicar el medio de transporte público cercanos o tener en cuenta donde se encuentran y así planificar su ruta, además de establecer colaboraciones con los operadores de cada transporte público para así poder garantizar la disponibilidad y precisión de los datos que sean necesarios para el funcionamiento de dicha aplicación, por lo cual nuestra propuesta podría ser financiada por las entidades estatales y de empresas privadas de transporte público.

1.3. Definición, siglas y abreviaturas

- "Combi App": Nombre de la aplicación móvil de transporte público desarrollada en el proyecto.
- GPS: Sistema de posicionamiento global.
- API: Interfaz de programación de aplicaciones.

- Usuario: Persona que utiliza la aplicación "Combi App" para acceder a los servicios de transporte público.
- Tiempo real: Información actualizada en tiempo presente, sin demoras significativas.
- Rutas: Trayectos establecidos por los buses de transporte público en la ciudad de Tacna.
- Notificaciones: Mensajes o alertas enviados a los usuarios para informarles sobre eventos relevantes relacionados con el transporte público.
- Movilidad sostenible: Enfoque de transporte que busca minimizar el impacto ambiental, promoviendo modos de transporte más eficientes y menos contaminantes.
- Usuario registrado: Persona que ha creado una cuenta en la aplicación "Combi App" para acceder a funcionalidades adicionales y personalizadas.
- Autoridades de transporte: Organismos encargados de regular y supervisar el sistema de transporte público en la ciudad de Tacna.

1.4. Organización del documento

Para la representación de la arquitectura del sistema se diseñó siguiendo el modelo 4+1 vistas.

Vista de casos de Uso: La descripción de la arquitectura se ilustra utilizando un conjunto de casos de uso que genera la quinta vista. Los escenarios describen secuencias de interacciones entre objetos, y entre procesos.

Vista lógica: Está enfocada en describir la estructura y funcionalidad del sistema. Los diagramas UML utilizados para representar esta vista son los diagramas de paquetes, secuencia, colaboración, objetos, clases y el relacional de la base de datos.

Vista de implementación: Ilustra el sistema desde la perspectiva del programador y está enfocado en la administración de los artefactos de software. Utilizamos los diagramas de paquetes y componentes.

Vista de proceso: Trata los aspectos dinámicos del sistema, explica los procesos de sistema y cómo se comunican. Se enfoca en el comportamiento del sistema en tiempo de ejecución. Para esta vista utilizamos el diagrama de actividad.

Vista de despliegue: Relacionada con la topología de componentes de software en la capa física, así como las conexiones físicas entre estos componentes. Utilizamos el diagrama de despliegue.

Por último, describimos los atributos de calidad del software como son funcionalidad, usabilidad, confiabilidad, rendimiento, y mantenibilidad.

2.OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS

2.1. Priorización de requerimientos

Se procede a desplegar los requerimientos funcionales y no funcionales desde una perspectiva de priorización, mediante una tabla resumen donde pueda desplegar los requerimientos del sistema de la siguiente forma:

ID	Descripción	Prioridad	Orden
CU01	El sistema debe permitir al registrar un usuario y un conductor	Alta	1
CU02	El sistema debe permitir al usuario y conductor iniciar sesión	Alta	2
CU03	El sistema debe permitir al usuario y conductor cerrar sesión	Alta	3
CU04	El sistema debe permitir al usuario y conductor agregar, buscar y editar un usuario y conductor	Media	4
CU05	El sistema debe permitir al usuario visualizar rutas y horarios	Alta	5
CU06	El sistema debe permitir al administrador, agregar, editar y una Compra	Alta	6
CU07	Consultar filtrar rutas	Alta	7
CU08	Generar notificaciones	Baja	8
CU09	El lenguaje debe ser C#	Alta	-
CU10	El sistema debe permitir al usuario y conductor ver imagen de perfil	Alta	-
CU11	El sistema debe permitir al usuario pagar tarifas del autobús a través de la aplicación.	Media	-
CU12	El sistema debe permitir al usuario visualizar la ubicación en tiempo real	Media	-

2.1.1 Requerimientos Funcionales

ID.	Descripción	Prioridad	Orden
CU01	El usuario podrá registrar sus datos (DNI, nombre, dirección, teléfono, email).	Alta	1
RF-02	El usuario podrá iniciar sesión en la aplicación.	Media	2
RF-03	El usuario podrá cerrar su sesión en la aplicación.	Media	

RF-04	El usuario podrá visualizar su perfil, incluyendo historial de viajes y preferencias.	Alta	3
RF-05	El usuario podrá editar su perfil, incluyendo sus datos personales y preferencias de viaje.	Media	4
RF-06	El usuario podrá añadir o cambiar su imagen de perfil.	Alta	5
RF-07	El usuario podrá ver la lista y detalles de las rutas de autobuses y sus horarios.	Alta	6
RF-08	El usuario podrá aplicar filtros a la búsqueda de rutas basados en diversos criterios como ubicación, hora, etc.	Alta	7
RF-09	El usuario recibirá notificaciones y alertas sobre su ruta de viaje, como llegadas de autobuses y cambios de horario.	Media	8
RF-10	El usuario podrá proporcionar feedback sobre su experiencia de viaje y la aplicación.	Media	9
RF-11	El usuario podrá pagar las tarifas del autobús a través de la aplicación.	Media	10
RF-12	El usuario podrá visualizar la ubicación en tiempo real de los autobuses en su ruta seleccionada.	Media	11
RF-14	El usuario podrá ver el perfil del conductor del autobús, incluyendo calificaciones y comentarios de otros usuarios.	Media	12

2.1.2 Requerimientos No Funcionales – Atributos de Calidad

ID	Requerimientos no funcionales	Prioridad
----	-------------------------------	-----------

CN01	Facilidad de instalación	Alta.
CN02	SDK mínimo	Media
CN03	Solicitud de permisos	Alta
CN04	Sistema de base de datos	Alta
CN05	Gestión de sesiones	Alta
CN06	Identificación mediante correo electrónico	Media
CN07	Unicidad de correo electrónico	Media
CN08	Identificador único de usuarios	Alta
CN09	Almacenamiento de imágenes de perfil	Alta
CN10	Tiempo de respuesta	Media

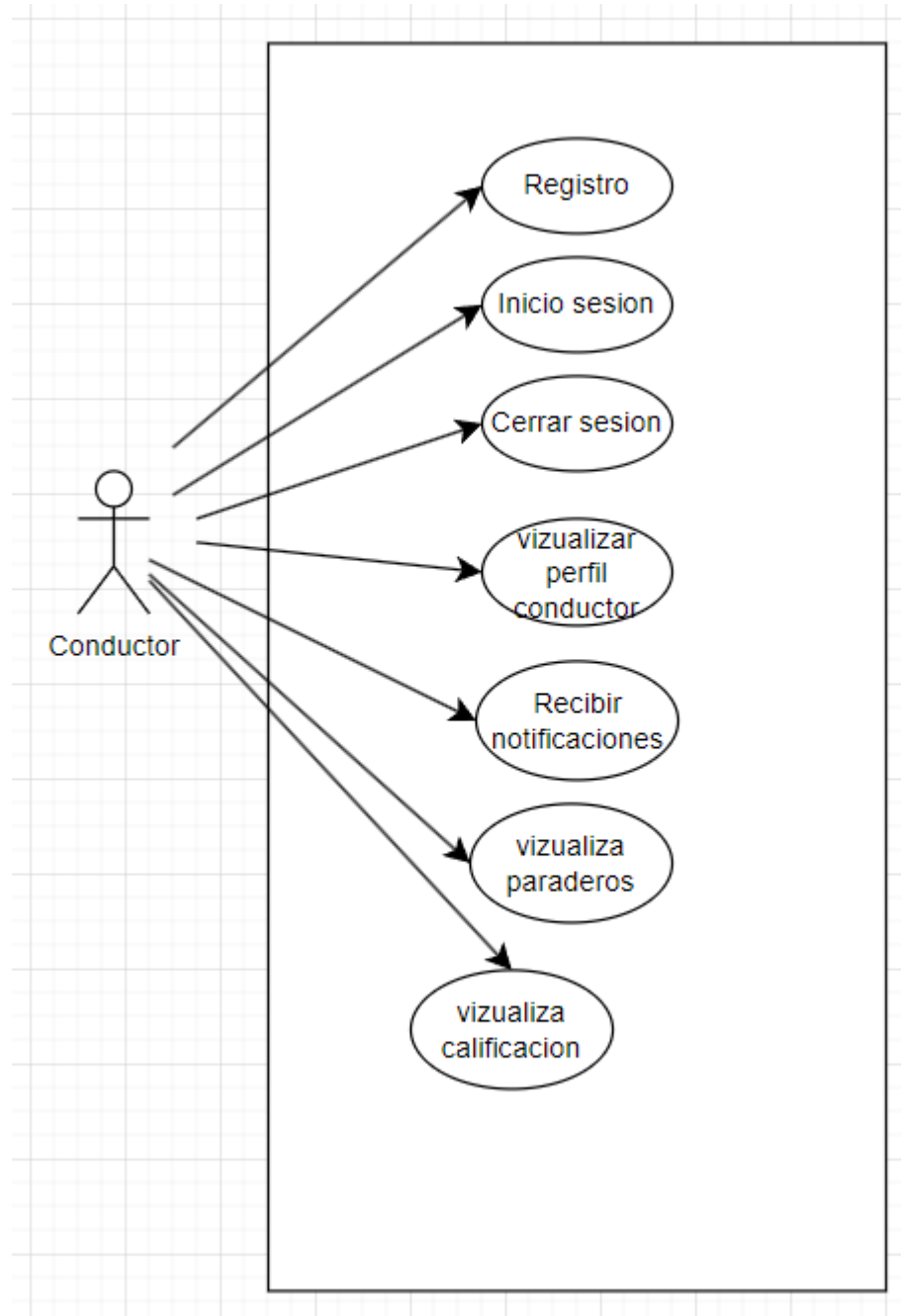
2.2. Restricciones

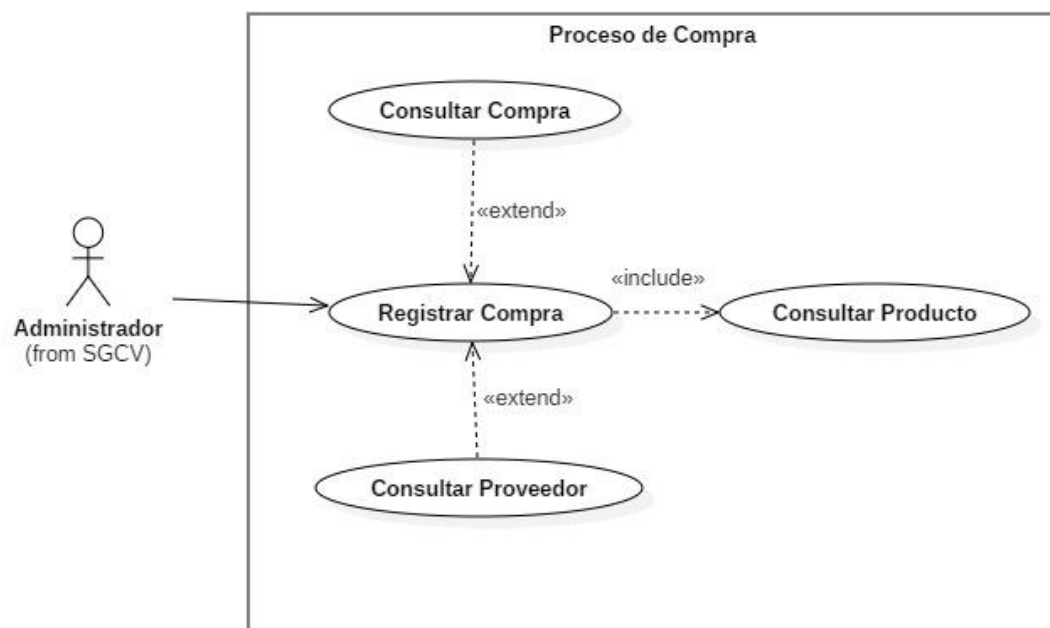
- Diseño basado en componentes de propósito claro y concreto, con alto grado de cohesión y bajo acoplamiento que permita el fácil reemplazo de los mismos.
- Componentes altamente reutilizables.

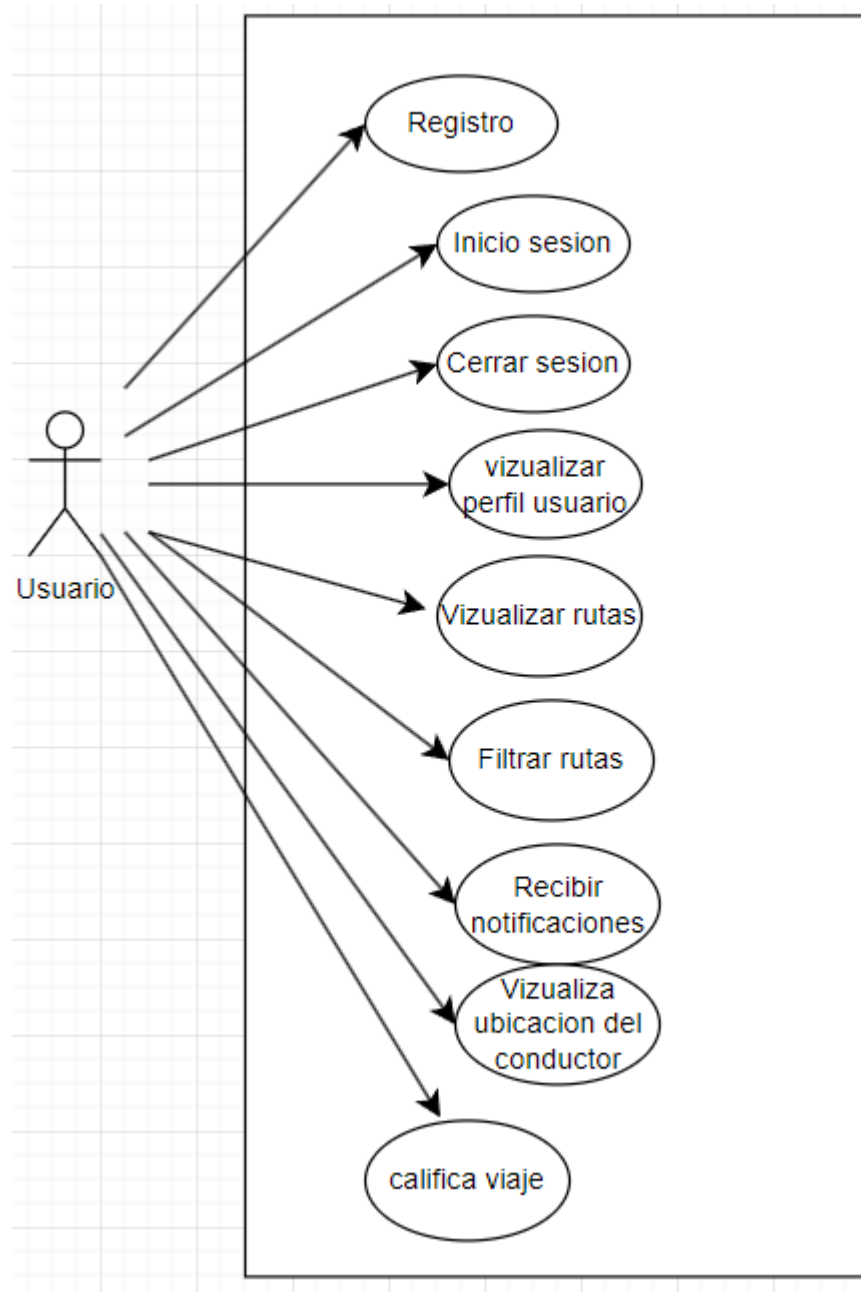
3.Representación de la arquitectura del sistema

3.1. Vista de Caso de uso

3.1.1 Diagramas de Casos de uso







3.2. Vista Lógica

3.2.1 Diagrama de Subsistemas (paquetes)

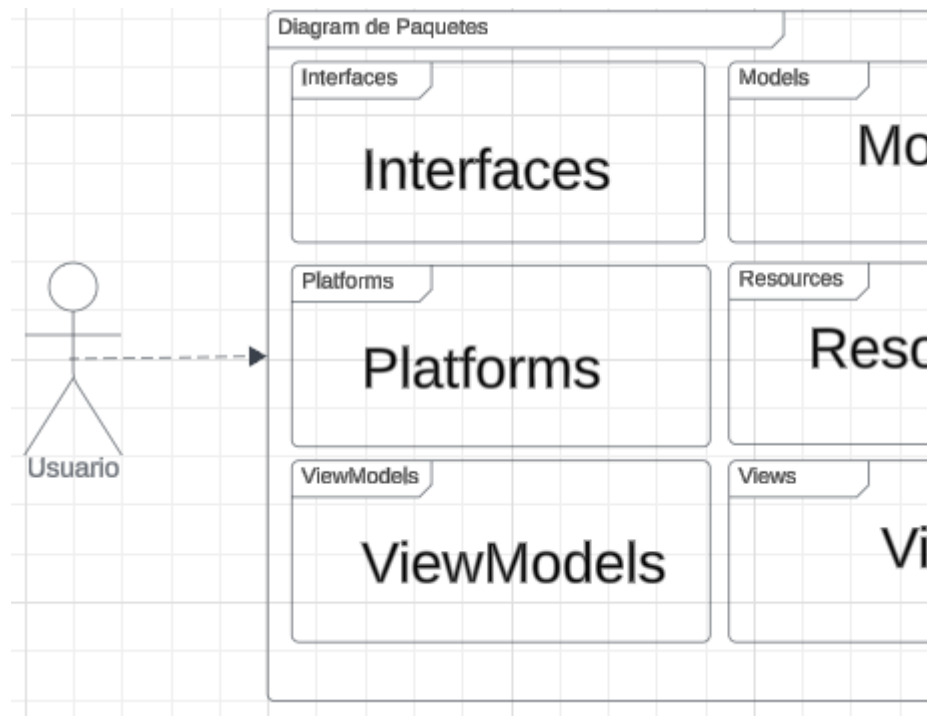
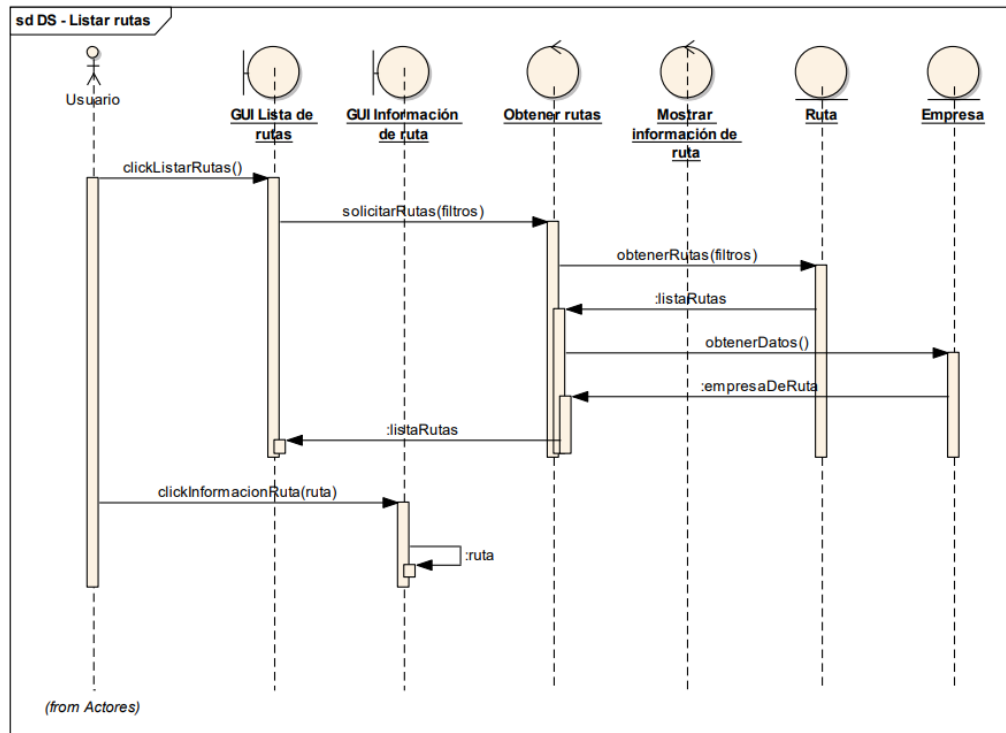
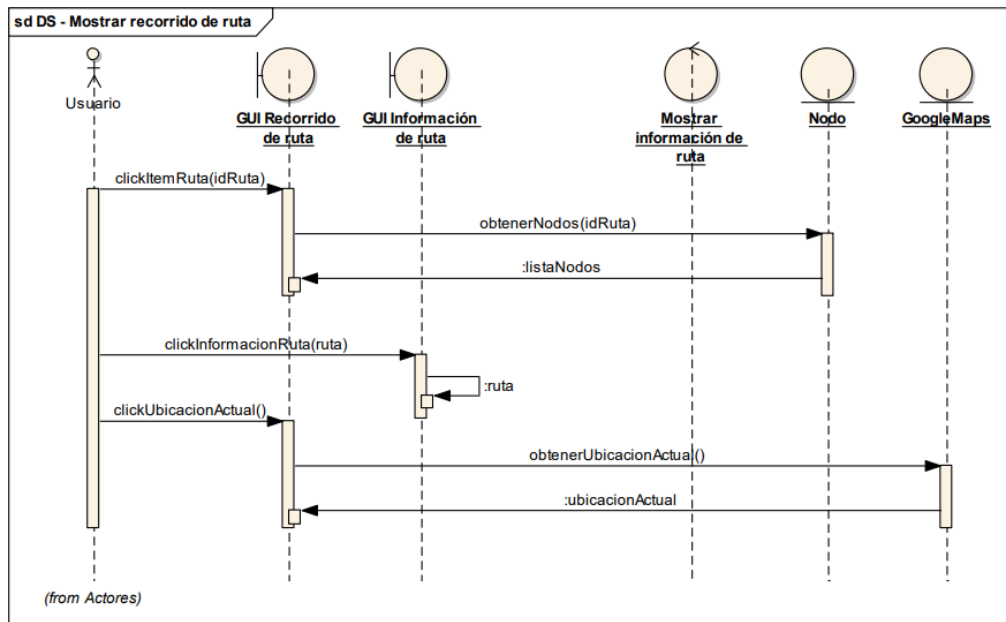


Diagrama de Secuencia (vista de diseño)

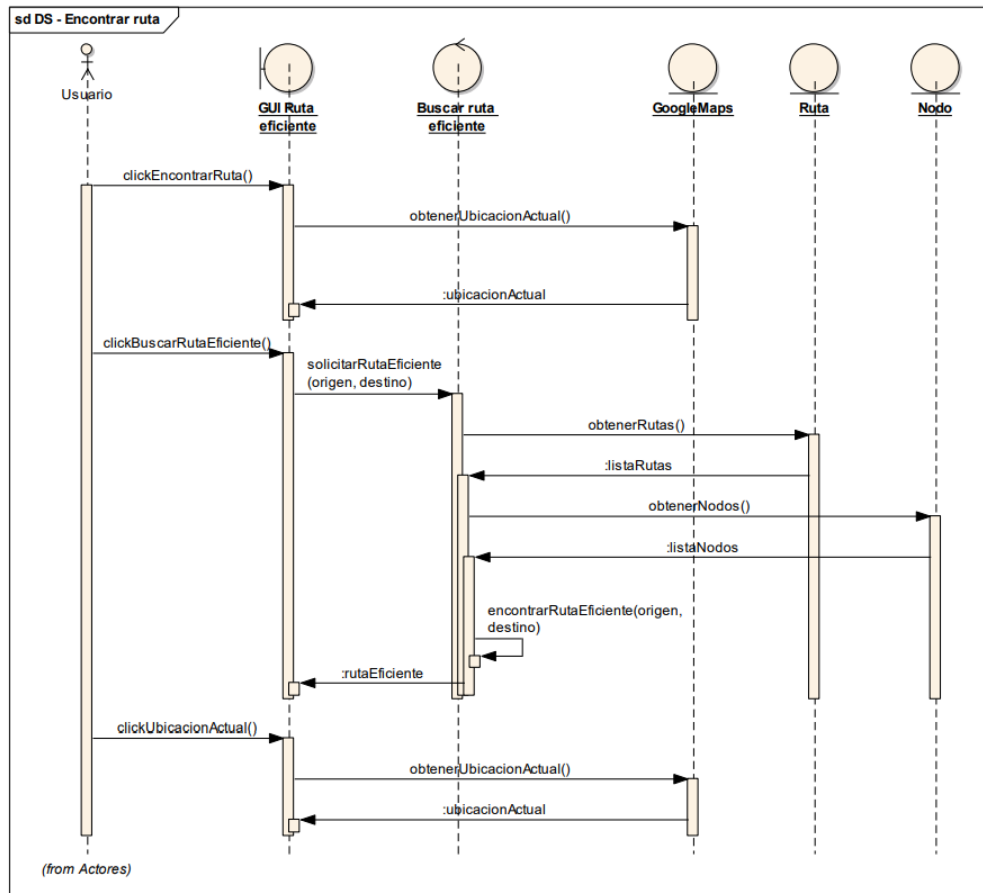
Vista listar Rutas



Vista mostrar el recorrido de las rutas



Vista encontrar ruta



3.2.2 Diagrama de Colaboración (vista de diseño)

Diagrama encontrar ruta...(análisis de objetos)

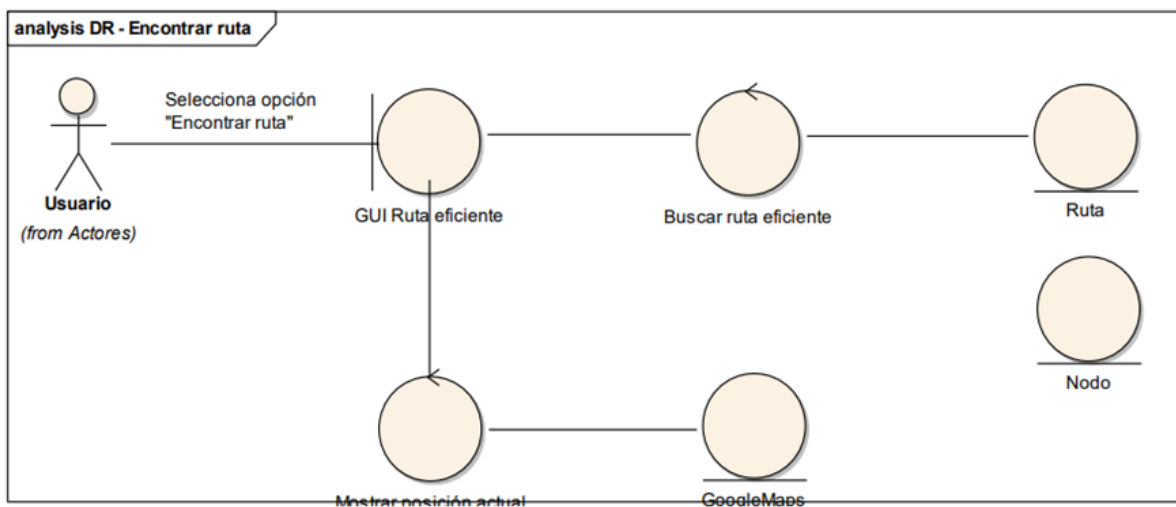


Diagrama listar rutas

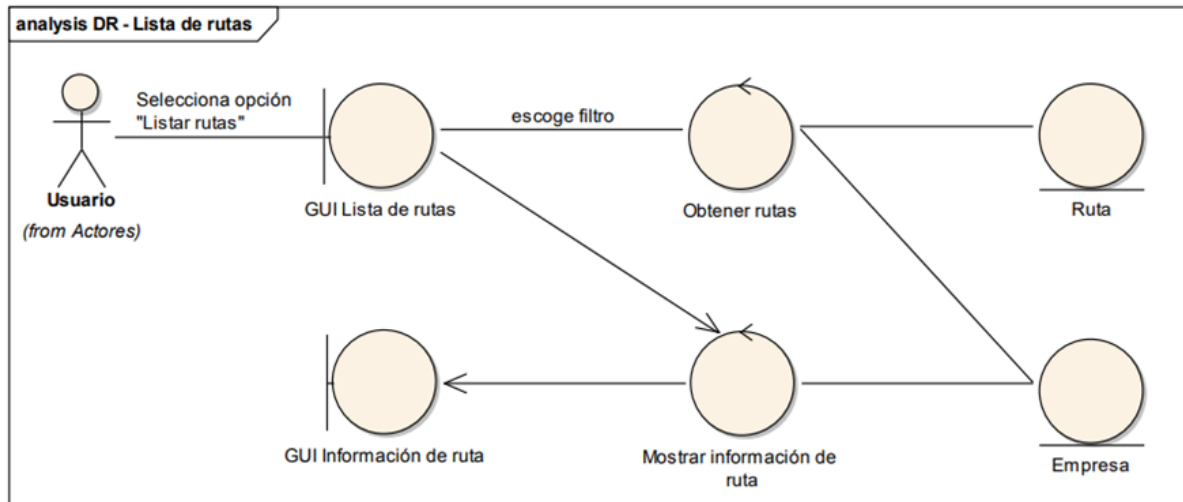
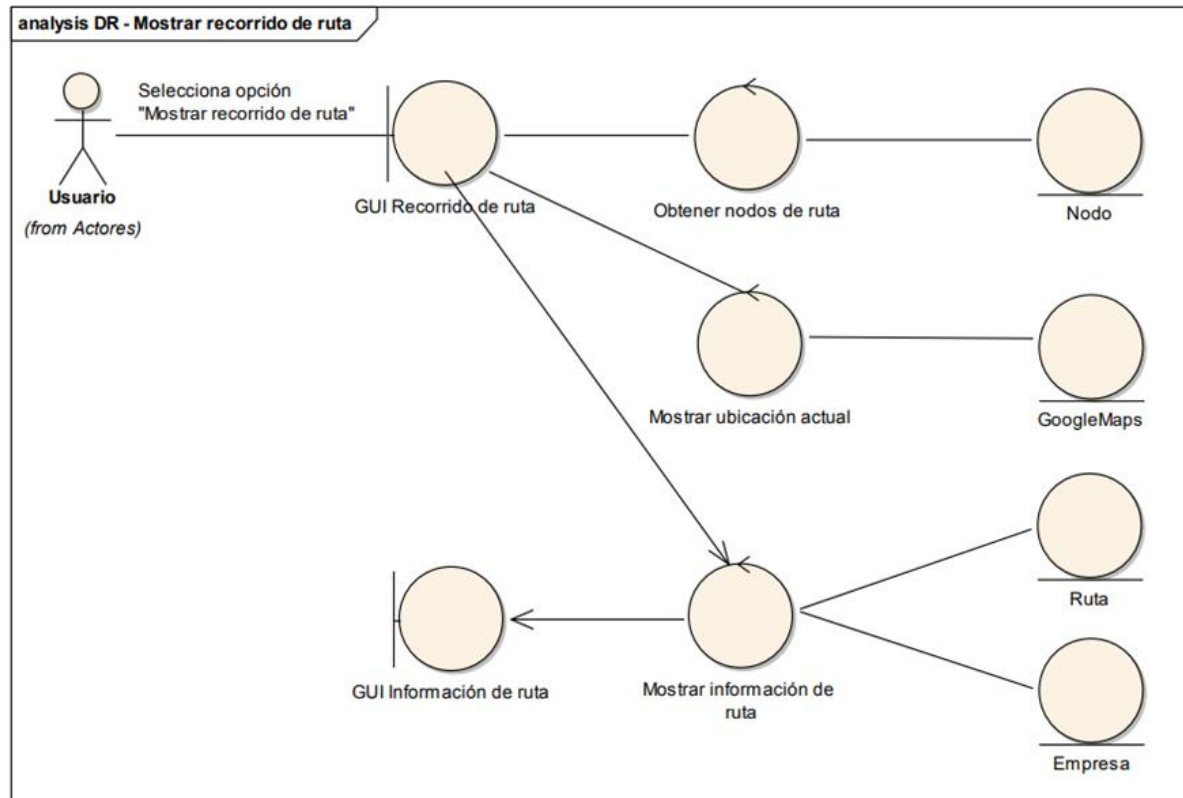
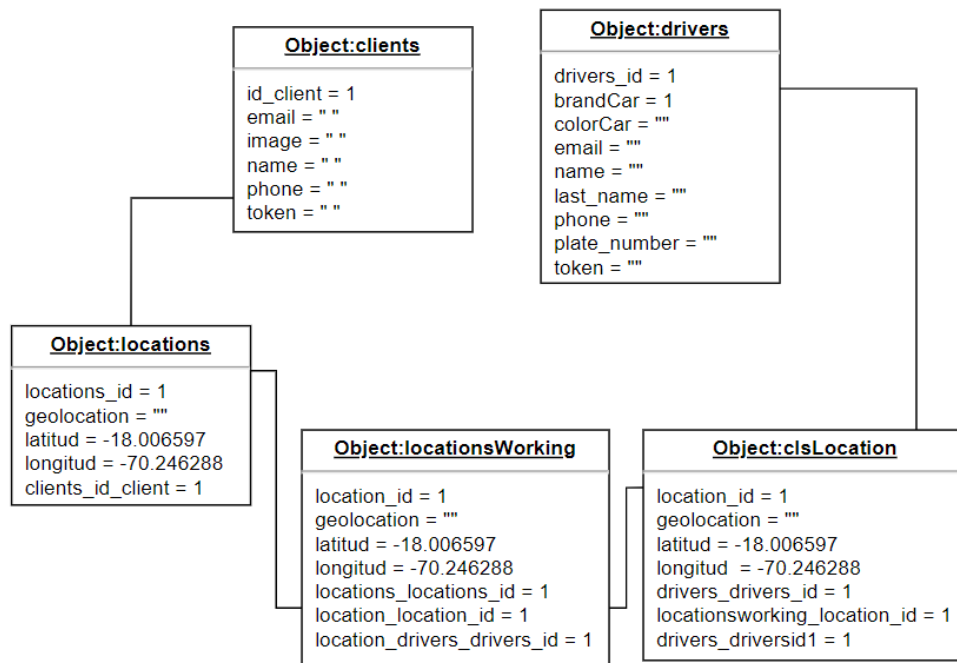


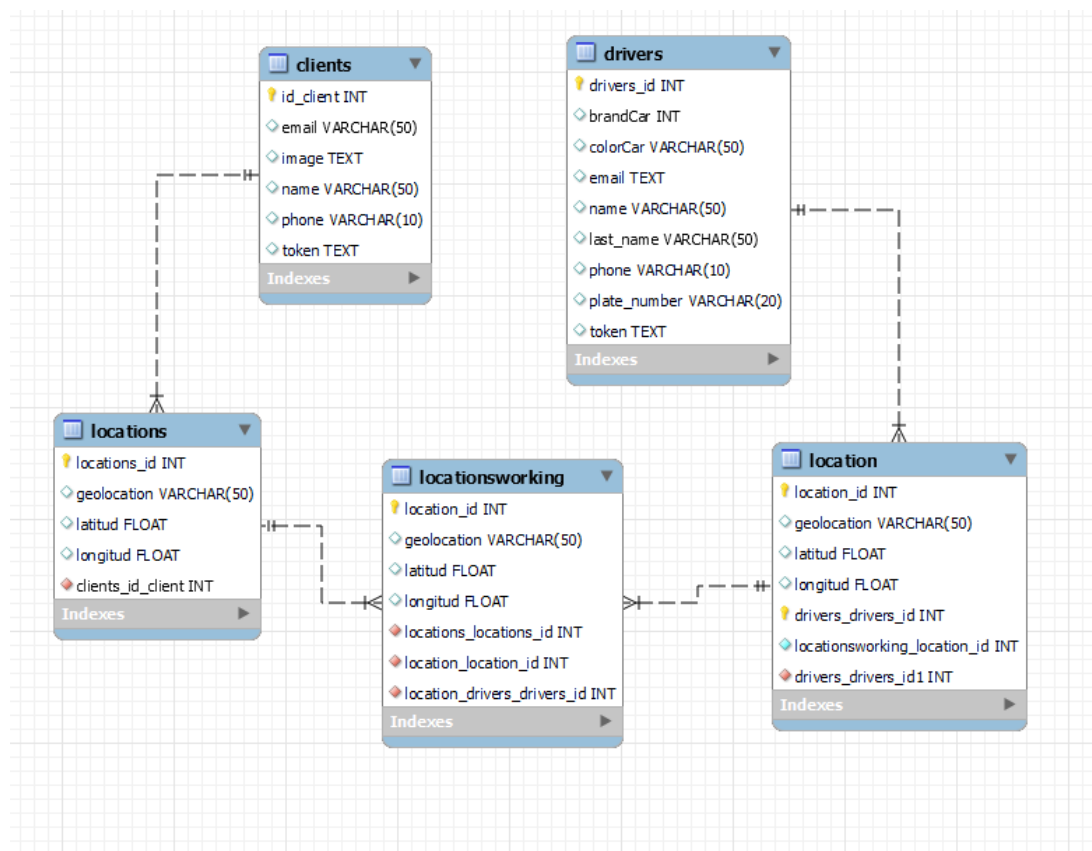
Diagrama mostrar recorrido de ruta



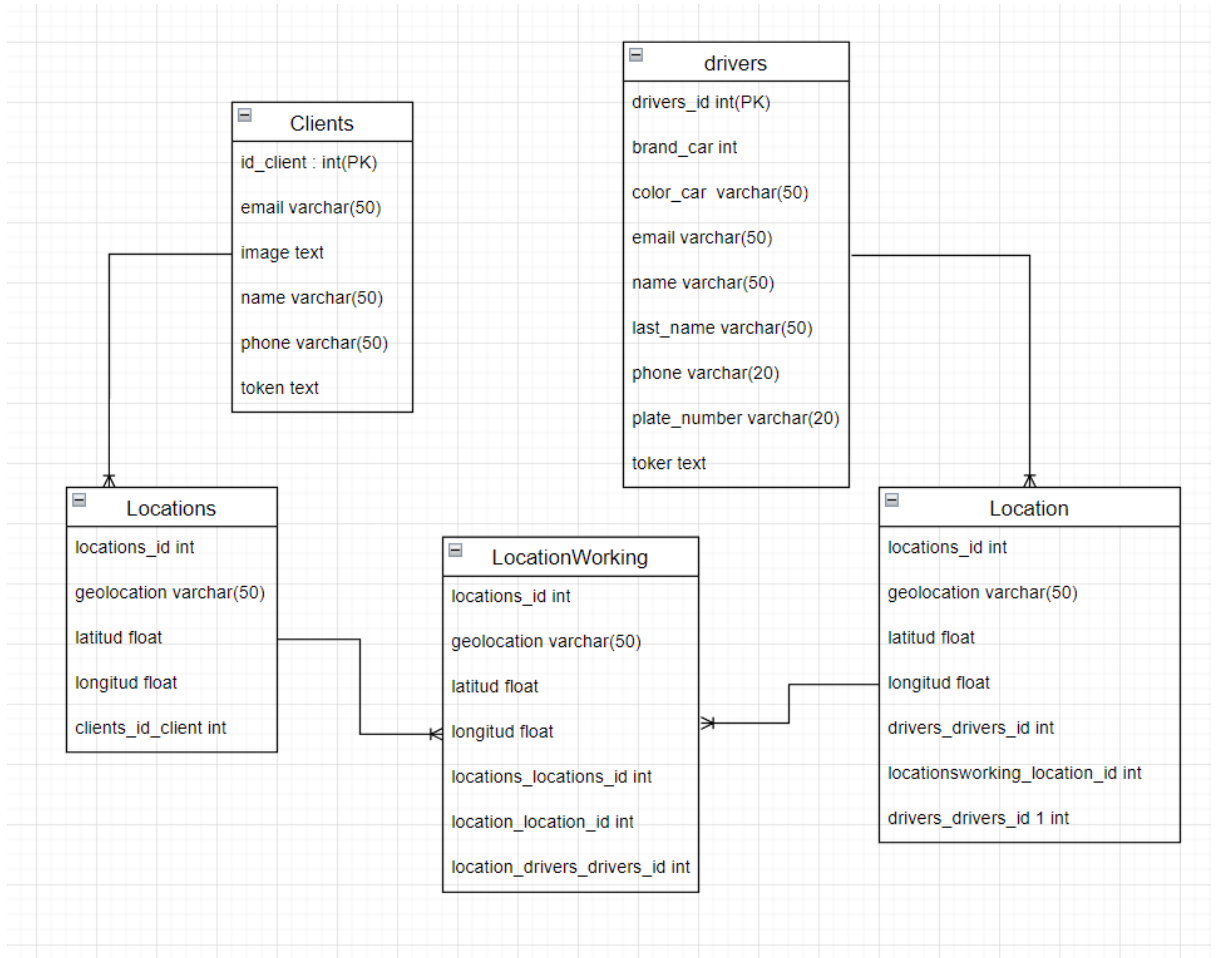
3.2.3 Diagrama de Objetos



3.2.4 Diagrama de Clases

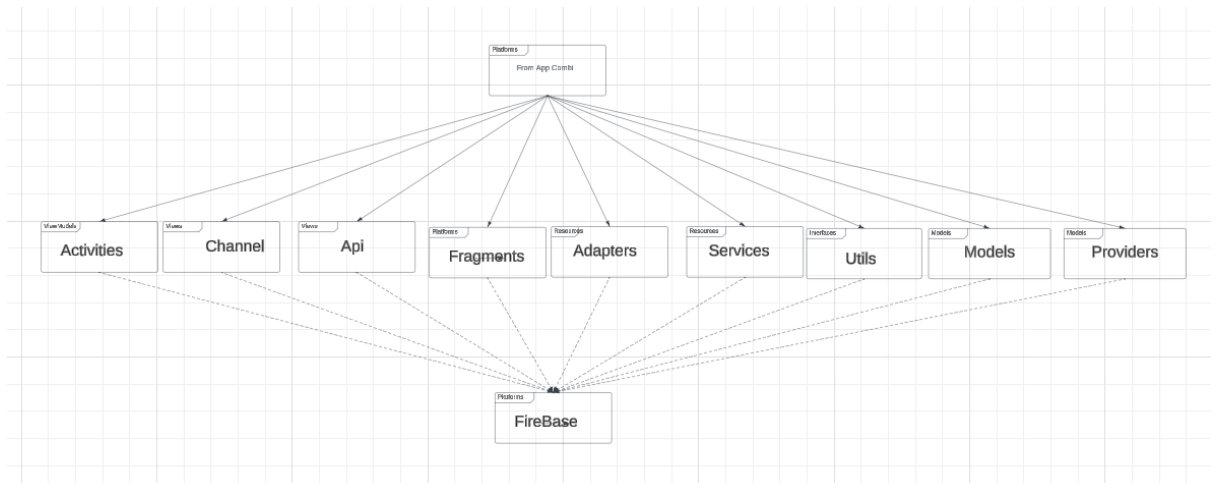


3.2.5 Diagrama de Base de datos (relacional o no relacional)

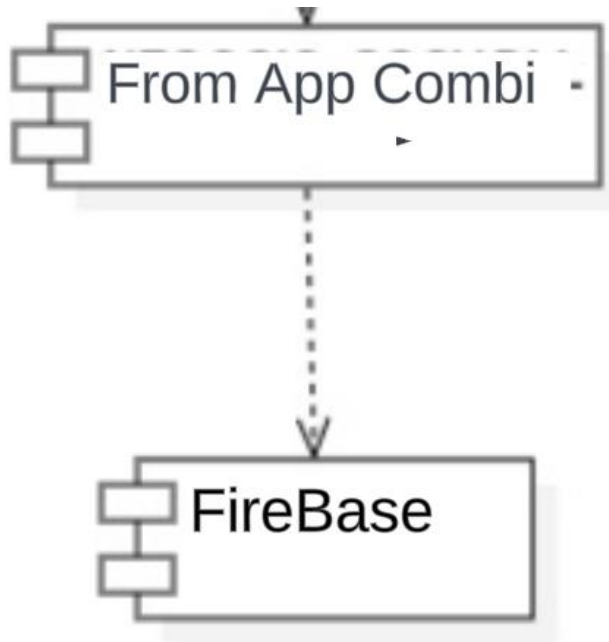


3.3. Vista de Implementación (vista de desarrollo)

3.3.1 Diagrama de arquitectura software (paquetes)



3.3.2 Diagrama de arquitectura del sistema (Diagrama de componentes)



3.4. Vista de procesos

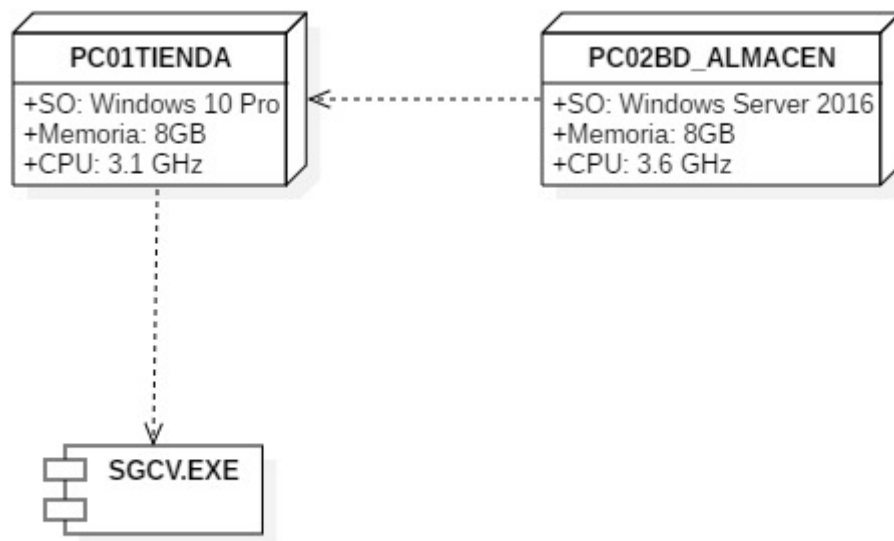
[Describe la descomposición del sistema procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican.]

3.4.1 Diagrama de Procesos del sistema (diagrama de actividad)

[Se realizará un diagrama del o los procesos del sistema donde se exponga las actividades donde interviene el sistema propuesto, adicionando diagramas que definan el detalle la descomposición del sistema en procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican]

3.5. Vista de Despliegue (vista física)

3.5.1 Diagrama de despliegue



4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

Los Atributos de Calidad (QAs) son propiedades medibles y evaluables de un sistema, estas propiedades son usadas para indicar el grado en que el sistema satisface las necesidades de los stakeholders [Wojcik 2013].

Los QAs además son concebidos como aquellos requerimientos que no son funcionales. De hecho, la funcionalidad es mayormente ortogonal a los QAs; un diseño puede cumplir con la funcionalidad deseada y fallar a la hora de satisfacer sus requerimientos de calidad. De esta manera, se entiende a la funcionalidad como la capacidad del sistema para hacer el trabajo para el cual fue pensado, independientemente de la estructura.

Existen QAs mayormente usados que se suelen identificar en numerosos sistemas y se tienen que describir, aunque la lista no es fina ya que muy a menudo hay situaciones en que podrían identificarse y proponerse nuevas propiedades para las diversas necesidades de stakeholders.

4.1. Escenario de Funcionalidad

[se califica de acuerdo con el conjunto de características y capacidades del programa, la generalidad de las funciones que se entregan y la seguridad general del sistema.]

4.2. Escenario de Usabilidad

Los usuarios pueden usarlo de forma eficiente y satisfactoria.

El administrador de compra necesita saber el stock de productos para ver cual es el que falta, entonces ingresa al nuevo sistema implantado se dirige al menú Reportes y selecciona la opción generar reporte de productos.

4.3. Escenario de confiabilidad

[Es el equilibrio entre la confidencialidad, la integridad, la irrefutabilidad y la disponibilidad de la información y datos manipulados por el sistema. Se trata del estado de un sistema, el cual puede ser transitorio y volátil. La seguridad de un sistema se caracteriza por mecanismos y técnicas empleados para intentar reducir los más posible el impacto provocado por un ataque, y las amenazas (entendidas como los caminos mediante los cuales se pueden provocar un ataque).

Abarca los planos de observación físico, lógico y humanos. Posee tres tipos de enfoque: prevención, precaución y reacción.]

4.4. Escenario de rendimiento

[Se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos, el conjunto y la eficiencia.] (Pressman 2010, pág. 187)

4.5. Escenario de mantenibilidad

La mantenibilidad es la habilidad de un sistema de someterse a cambios con un grado de facilidad. Esos cambios pueden afectar componentes, servicios, características e interfaces cuando se agrega o se cambia la funcionalidad corrigiendo errores y satisfaciendo nuevos requerimientos de negocios.

4.6. Otros Escenarios

Performance: El atributo de calidad Performance se refiere a la capacidad de responder, ya sea el tiempo requerido para responder a eventos determinados, o bien, la cantidad de eventos procesados en un intervalo de tiempo dado. La Performance caracteriza la proyección en el tiempo de los servicios entregados por el sistema.