

Universidad del Quindío

Proyecto Sistema de Gestión de Renta de vehículos

Documento Arquitectura de Software

Versión <1.0>

Revisión Histórica

Fecha	Versión	Descripción	Autor
15/04/2016	1.0	El Enfoque del proyecto sistema de gestión de renta de vehículos , que consiste en crear Un negocio de renta de vehículos con sede en el Quindío, además de crear un sistema de gestión de toda la información involucrada en el alquiler de un Auto.	Gonzalo Andres Salazar Florez. Juan David Velasquez Rojas. Erick Vargas Hernandez. Juan Esteban Bolaños Quintero. Mauricio Collazos Murillo.

Tabla de Contenido

1. [Introducción.](#)
 - 1.1 [Propósito.](#)
 - 1.2 [Alcance.](#)
 - 1.3 [Definiciones, acrónimos y abreviaturas.](#)
 - 1.4 [Referencias.](#)
 - 1.5 [Visión General.](#)
2. [Representación Arquitectural.](#)
3. [Objetivos de Arquitectura y Limitaciones.](#)
4. [Vistas de Caso de Uso.](#)
 - 4.1. [Diagramas de secuencia.](#)
 - 4.2. [Diagramas de colaboración.](#)
5. [Vista Lógica](#)
 - 5.1 [Visión General](#)
 - 5.2 [Paquetes de gran importancia para el diseño arquitectónico](#)
 - 5.3 [Realización de Casos de Uso](#)
6. [Vista de Procesos](#)
7. [Vista de Deployment](#)
8. [Vista de Implementación](#)
 - 8.1 [Visión General](#)
 - 8.2 [Capas](#)
9. [Tamaño y actuación](#)
10. [Calidad](#)

Documento de Arquitectura de Software

1. Introducción.

A continuación, mostramos el enfoque del Sistema de Gestión de Renta de vehículos, ilustrando las características principales de éste. Con el fin de exponer una visión general de las decisiones arquitectónicas generales realizadas al sistema. Evaluamos las características generales del diseño del sistema, los requisitos solicitados por el cliente final y los modelos y vistas que lo definen.

Propósito.

Este documento tiene como propósito brindar una visión comprensible del Sistema de Gestión de Renta de vehículos, sus funcionalidades y vistas requeridas para el sistema.

Alcance.

El documento se centra en el desarrollo de un sistema de gestión de toda la información involucrada en el alquiler de vehículos en el departamento del Quindío.

El sistema deberá permitir a sus usuarios:

1. La aplicación debe registrar, modificar y eliminar en un catálogo la información de cada carro: especificación del tipo de carro, foto, la marca, la capacidad, el precio de la renta por hora, el color, el modelo, la placa, el kilometraje y si está o no disponible en el momento.
2. La aplicación debe Registrar, modificar y eliminar las solicitudes de renta, que contiene la fecha, la hora, el kilometraje con el que se entrega, los datos del cliente: nombre, número de cédula, número de licencia de conducción, el teléfono y la dirección de su lugar de alojamiento.
3. La aplicación debe Registrar la devolución de un carro alquilado, indicando la fecha, la hora, el kilometraje actual, el estado del vehículo y observaciones cuando lo considere necesario.
4. Se debe Retribuir en bono canjeable, la cantidad de horas acumuladas por un cliente usando el servicio de alquiler. (Por cada 100 horas acumulados se dará un bono de 50.000)
5. Se presentará Facturar en cada servicio contratado.
6. Se debe generar reportes sobre las rentas hechas en un periodo de tiempo, sobre kilometraje de los carros para determinar la necesidad de someterlos a mantenimiento o darlos de baja.

Universidad del Quindío

Definiciones, acrónimos y abreviaturas.

Funcionalidad

El sistema tendrá que cumplir con varios requisitos para que sea realmente funcional, entre ellos ser preciso, adecuado para los usuarios y el administrador(a) de los sistemas y sobre todo que tenga un grado muy alto de seguridad.

Fiabilidad

Es importante que todo sistema tenga gran rango de fiabilidad, eso incluye la madurez que pueda tener el software, sin contar con que el sistema tendrá que ser tolerante a errores, tener un plan b que haga al sistema tener un alto rango de recuperabilidad.

Usabilidad

La usabilidad de la aplicación tendrá cómo principio entender que no solo será importante basar la usabilidad en el usuario final, sino dentro de los miembros internos y funcionales de la misma, como lo son el administrador del sistema y usuarios, que son los principales involucrados en este proyecto.

Eficiencia

La eficiencia del proyecto final se refleja en el servicio ofrecido a los usuarios, reduciendo las fallas que se presentan normalmente con la asignación de vehículos, en los que se han evidenciado, por no tener una agenda adecuada y disponibilidad clara de los vehículos agendados.

Mantenibilidad

Para garantizar que el software sea mantenible, antes tenemos que hacer que el software sea escalable y compatible con nuevas versiones, esto quiere decir que a la hora de poder intervenir se puedan hacer cambios rápidos y eficaces. Es importante también tener en cuenta que la estabilidad del proyecto podrá sustentarse con un desarrollo sencillo y las herramientas adecuadas.

Portabilidad

El sistema tendrá la característica responsive y podrá ser desplegado en cualquier plataforma o dispositivo diferente al ambiente de producción o donde se este desarrollo. Lleva inmersa la despleabilidad y podrá ser visualizado sin ningún inconveniente.

Referencias.

Documento de especificación de Requisitos del Sistema de Gestión de Renta de vehículos.

Documento de Visión del Sistema de Sistema de Gestión de Renta de vehículos.

Visión General.

Este documento cuenta con una breve descripción de los procesos con los que cuenta el Sistema de Gestión de Renta de vehículos, los cuales se encuentran representados en modelos.

2. Representación Arquitectural.

El Sistema de Gestión de Renta de vehículos cuenta con procesos como: registrar, modificar y eliminar en un catálogo la información de cada carro, registrar, modificar y eliminar las solicitudes de renta, registrar la devolución de un carro alquilado, retribuir en bono canjeable, facturación en cada servicio contratado, generar reportes sobre las rentas hechas en un periodo de tiempo..

El modelo propuesto por RUP para representar la arquitectura utiliza el siguiente conjunto de vistas:

- Vistas de caso de uso: Listamos los casos de uso del modelo de casos de uso que representen funcionalidades centrales del sistema final.
- Vista Lógica: Describimos las partes arquitectónicamente significativas del modelo de diseño.
- Vista de Procesos: Indicamos que procesos se comunican o interactúan entre sí y la manera en que se comunican.
- Vista de Deployment: Describimos los escenarios de distribución física del Sistema de Gestión de Renta de vehículos sobre los cuales se ejecutará.
- Vista de implementación: Describimos la estructura general del modelo de implementación y el mapeo de los subsistemas, paquetes y clases de la Vista lógica.
- Vista de Datos: Describimos los elementos principales del Modelo de Datos.

3. Objetivos de Arquitectura y Limitaciones.

Las propiedades esenciales que se va a definir en el Sistema de Gestión de Renta de vehículos son los aspectos que influyen en la etapa de desarrollo.

Objetivos

1. El Sistema de Gestión de Renta de vehículos permitirá a los usuarios finales acceder al sistema desde cualquier lugar con acceso a internet.
2. EL sistema contará con registro de vehículos y renta de vehículos, el cual contará con dos tipos de usuario, el administrador quien maneja los horarios y disponibilidad de los vehículos y por otro para los usuarios los cuales podrán gestionar el alquiler y devolución de los vehículos rentados.
3. EL sistema contará con un login, el cual contará con dos tipos de usuario, el administrador quien maneja los horarios y otro para los usuarios los cuales tendrán deshabilitados las funciones del administrador.

Limitaciones

1. Los usuarios deberán contar con un dispositivo móvil (Smartphone).

4. Vistas de Caso de Uso

Se visualizarán las actividades o procesos a realizar por los usuarios dentro de la aplicación, se demostrara gráficamente en las descripciones formales de los casos de uso :



Figura 1. Diagrama de casos de uso relevantes para el sistema.

Nombre: CRUD Usuario
Descripción: En este caso de uso se realiza la gestión referente a los usuarios, tales como: Crear usuario Editar usuario Eliminar usuario
Actores:

Universidad del Quindío

Administrador.
Sistemas involucrados: No aplica.
Pasos: Crear usuario: <ol style="list-style-type: none">1. El administrador debe recibir toda la información por parte del usuario para poder realizar la creación del usuario.2. Una vez se reciba la información el sistema crea un usuario y le redirecciona a la vista principal donde encontrará el listado de vehículos para rentar.. Eliminar usuario: <ol style="list-style-type: none">1. El administrador se dirige a la lista de los usuarios, selecciona un usuario y elimina la cuenta.2. Una vez eliminada la cuenta el administrador verifica si fue exitosa la eliminación del usuario. Editar Usuario: <ol style="list-style-type: none">1. El administrador selecciona un usuario de la lista de usuarios.2. Realiza la petición de los datos que el usuario desea cambiar o actualizar3. Ingresa al sistema la información proporcionada por el usuario para su edición.
Precondiciones: · El administrador deberá tener un usuario para realizar el proceso de Creación, Edición y Eliminación.
Suposiciones: · El administrador debe de estar logueado como administrador para poder realizar las respectivas modificaciones.
Manejo de situaciones anormales: En caso de no poder realizar la creación de un usuario el administrador deberá informar al usuario que no se ha podido completar el procedimiento. En caso de no poder realizar la eliminación de un usuario el administrador deberá informar al usuario que no se ha podido completar el procedimiento. En caso de no poder realizar la edición de un usuario el administrador deberá informar al usuario que no se ha podido completar el procedimiento.

Nombre:
CRUD Vehiculo

Universidad del Quindío

Descripción: En este caso de uso se realiza la gestión referente a los vehículos, tales como: Crear vehículos Editar vehículos Eliminar vehículos
Actores: Administrador
Sistemas involucrados: No aplica.
Pasos: Crear vehiculo: <ol style="list-style-type: none">1. El administrador deberá ingresar al sistema con una cuenta de administrador.2. Una vez haya ingresado al sistema el administrador debe ingresar la información completa sobre el vehículo que va a crear. Tales como, placa, kilometraje, precio, disponible. Etc.3. Una vez los datos fueron ingresar el administrador prosigue a crear el vehículo. Editar vehiculo: <ol style="list-style-type: none">1. El administrador deberá ingresar al sistema con una cuenta de administrador.2. Una vez haya ingresado al sistema el administrador debe ingresar la información nueva que va a tener el vehículo.3. Una vez la información nueva se ha agregado el administrador deberá guardar los cambios realizados. Eliminar vehiculo: <ol style="list-style-type: none">1. El administrador deberá ingresar al sistema con una cuenta de administrador.2. Una vez haya ingresado el administrador deberá escoger un vehículo de una lista de vehículos para darle de baja.
Precondiciones: <ul style="list-style-type: none">· El administrador debe tener la información referente a un vehículos, ya sea para crearlo, editarlo o las estadísticas de funcionalidad para eliminarlo.
Suposiciones: <ul style="list-style-type: none">· El administrador deberá estar logueado con una cuenta de administrador.
Manejo de situaciones anormales: En caso de no poder realizar la creación de un vehículo el administrador deberá revisar la que la información sea correcta para completar el procedimiento. En caso de no poder realizar la eliminación de un vehículo el administrador deberá verificar que el vehículo esté en disposición para ser eliminado y poder completar el procedimiento.

Universidad del Quindío

En caso de no poder realizar la edición de un vehículo el administrador deberá verificar la información que ha ingresado para poder completar el procedimiento.

Nombre:

CRUD Solicitud renta

Descripción:

En este caso de uso se realiza la gestión referente a las solicitudes de renta, tales como:

Crear solicitudes de renta

Editar solicitudes de renta

Eliminar solicitudes de renta

Actores:

Usuario, Administrador

Sistemas involucrados:

No aplica.

Pasos:

Crear solicitudes de renta:

1. El usuario debe seleccionar el vehículo el cual desea rentar.
2. El usuario debe ingresar la información referente al alquiler, como fecha de alquiler, fecha de devolución. Ect.
3. El usuario deberá enviar la información ingresada a la aplicación y con esto termina su procedimiento.
4. Posteriormente después que el usuario envía la solicitud de renta, el administrador realiza la creación de la solicitud,
5. El administrador verifica que los datos proporcionados por el usuario sean correctos y procede a cambiarle el estado de disponibilidad al vehículo por el cual se realizó la solicitud.

Editar solicitudes de renta:

1. El usuario debe seleccionar la solicitud para poder realizar los cambios respectivos.
2. Una vez hecho esto el administrador recibe la información proporcionada por el usuario y procede a cambiar los datos por los nuevos.
3. Después el administrador notifica al usuario que los cambios se han realizado con éxito.

Eliminar solicitudes de renta:

Universidad del Quindío

<ol style="list-style-type: none"> 1. El usuario debe seleccionar la solicitud realizada por un vehículo. 2. El administrador recibe la información referente a la solicitud y procede a eliminarla. 3. El administrador notifica al usuario que la solicitud ha sido eliminada con éxito.
<p>Precondiciones: El usuario deberá escoger el vehículo para poder realizar la creación, edición o eliminación de la solicitud de renta, Además por parte del administrador este deberá tener la solicitud realizada por el usuario.</p>
<p>Suposiciones: El usuario debe estar logueado y seleccionar un vehículo para poder realizar la creación, edición o eliminación de la solicitud de renta.</p>
<p>Manejo de situaciones anormales:</p> <p>En caso que administrador no pueda realizar la creación de la solicitud, esté deberá informar al usuario que verifique si la información proporcionada es correcta.</p> <p>En caso que administrador no pueda realizar la edición de la solicitud, esté deberá informar al usuario que verifique si la información proporcionada es correcta.</p> <p>En caso que administrador no pueda realizar la eliminación de la solicitud, esté deberá informar al usuario que verifique si la información proporcionada es correcta.</p>

<p>Nombre: Registrar devolucion</p>
<p>Descripción: En este caso de uso se realiza todo el proceso para la devolucion de un vehiculo por parte de un usuario.</p>
<p>Actores: Administrador, Usuario.</p>
<p>Sistemas involucrados: No aplica.</p>
<p>Pasos:</p> <ol style="list-style-type: none"> 1. El usuario debe ingresar al sistema en la fecha indicada en el momento de la renta. 2. Una vez en el sistema deberá entregar la información correspondiente al estado del vehículo, como por ejemplo su kilometraje, observaciones generales. Etc. 3. El usuario envía la información al sistema. 4. El administrador verifica la información enviada por parte del usuario.. 5. Una vez verificada la información el administrador envía al usuario una notificación de que el vehículo ha sido entregado de forma correcta. 6. El administrador cambia el estado de disponibilidad del vehículo.

Universidad del Quindío

Precondiciones: El usuario debió haber rentado un vehículo. El usuario debe de estar en la fecha indicada para su posterior devolución.
Suposiciones: El usuario debe haber ingresado correctamente a la aplicación y tener acceso a esta funcionalidad.
Manejo de situaciones anormales: En caso de no poder realizar la devolución de forma correcta del vehículo por parte del usuario, este debe verificar que la fecha sea la indicada. En caso de que el usuario no pueda realizar la devolución del vehículo, el usuario debe recibir una notificación indicándole que la información del vehículo no concuerda con la que tiene el administrador.

Nombre: Facturar servicio.
Descripción: Se generará cobro por cada uno de los servicios contratados por los usuarios en la aplicación.
Actores: Usuario Administrador
Pasos: <ol style="list-style-type: none">1. El usuario ingresa su nombre y contraseña.2. El usuario escoge el servicio que desea contratar.3. Crea la petición formal al sistema.4. El sistema comprueba la disponibilidad del servicio y lo carga a la cuenta del sistema.5. Genera facturación por servicios prestados.
Precondiciones: <ul style="list-style-type: none">· El Usuario deberá estar registrado en nuestra aplicación.· El Usuario deberá estar registrado en nuestras bases de datos.· El usuario deberá tener cuentas bancarias asociadas para realizar los pagos de los servicios.

Universidad del Quindío

Suposiciones: <ul style="list-style-type: none">· El usuario debe haber ingresado correctamente a la aplicación y tener acceso a esta funcionalidad.· El usuario deberá tener saldo suficiente para contratar los servicios.· Se crearán las facturas pertinentes a los servicios prestados.
Manejo de situaciones anormales: <ul style="list-style-type: none">· Si el usuario no pudiera hacer el login en nuestro aplicativo, podrá notificarlo a través de una llamada o un mensaje al correo electrónico.· En caso de algún error en el sistema, se debe informar al usuario.· En caso de no realizar la transacción se generará un mensaje de alerta en el sistema y una notificación al usuario.

Nombre: Generar reporte por rentas.
Descripción: Se generará a través del sistema un reporte por la renta de los vehículos a los usuarios, con la descripción detallada del servicio, el tipo de vehículo, marca, modelo, kilometraje y valor asociados a la renta .
Actores: Usuario.
Sistemas involucrados: No aplica.
Pasos: <ol style="list-style-type: none">1. El usuario debe ingresar al sistema y realizar una renta.2. El sistema recibe la solicitud de renta.3. Evalúa la renta y realiza un reporte de la renta.5. Envía el reporte de renta al usuario.
Precondiciones: <ul style="list-style-type: none">· El Usuario deberá estar registrado en nuestra aplicación.· El Usuario deberá estar registrado en nuestras bases de datos.· El usuario deberá tener cuentas bancarias asociadas para realizar los pagos de los servicios.· El usuario debe realizar una renta.
Suposiciones:

Universidad del Quindío

- El usuario debe haber ingresado correctamente a la aplicación y tener acceso a esta funcionalidad.
- El usuario deberá tener saldo suficiente para contratar los servicios.
- Se debe haber realizado una renta.

Manejo de situaciones anormales:

- Si el usuario no pudiera hacer el login en nuestro aplicativo, podrá notificarlo a través de una llamada o un mensaje al correo electrónico.
- En caso de algún error en el sistema, se debe informar al usuario.
- En caso de no realizar la transacción se generará un mensaje de alerta en el sistema y una notificación al usuario por correo electrónico.

Nombre:

Generar reporte por kilometraje.

Descripción:

Se generará a través del sistema un reporte por el kilometraje de los vehículos con la descripción detallada del servicio, el tipo de vehículo, marca, modelo, kilometraje, además se determinará la necesidad de someterlos a mantenimiento o darlos de baja..

Actores:

Administrador

Sistemas involucrados:

No aplica.

Pasos:

1. Se establecerán una auditoría o revisión periódica para los vehículos.
2. Parametros de desempeño de los vehículos.
3. Se realizará una revisión del kilometraje y se compara con el historial de de cada vehículo.
4. Según lo resultados de la revisión y la auditoría de los vehículos se genera un reporte del estado de los mismos.

Precondiciones:

- El vehículo deberá estar registrado en la aplicación.
- El vehículo deberá estar registrado en el catálogo de la aplicación.
- El vehículo deberá contar con un historial de de usuarios.
- El vehículo deberá contar con un historial de revisiones.

Universidad del Quindío

Suposiciones:

- El vehículo debe estar registrado en la bases de datos, con sus respectivas revisiones por kilometraje.
- Si el vehículo no cuenta con sus respectivas revisiones por kilometraje no puede estar habilitado en el catálogo.
- Si el vehiculo no cumple con los requerimientos tecnicos y mecanicos establecido en la auditoría de los vehículos se dará de baja.
- Se debe generar reporte de los vehículos, en cuanto al kilometraje y el estado mecánico de los mismos.
- Se Genera reporte de los vehículos dados de baja por el sistema..

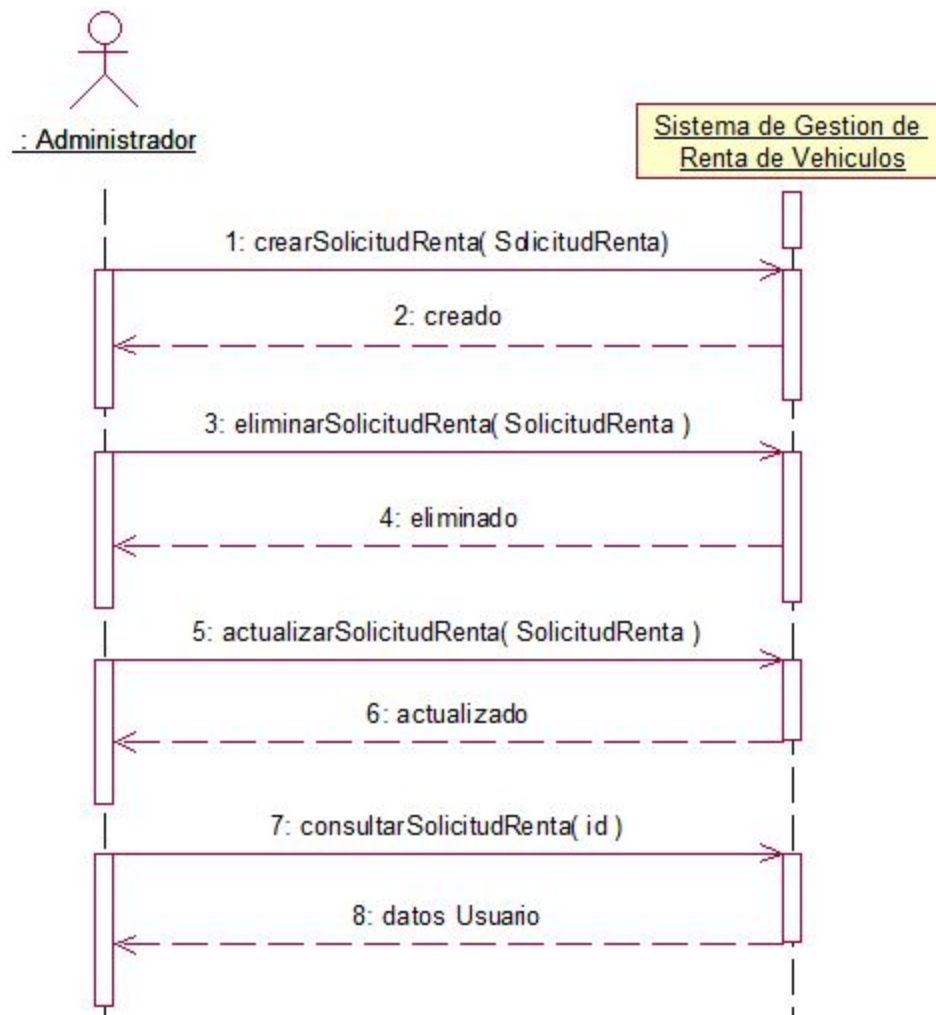
Manejo de situaciones anormales:

- En caso tal en el que el vehículo se encuentre en funcionamiento y no cuente con las revisiones pertinentes por kilometraje.
- Que se presente un vehículo en el catálogo el cual haya sido dado de baja.
- se generen reportes sin haber hecho las revisiones pertinentes.

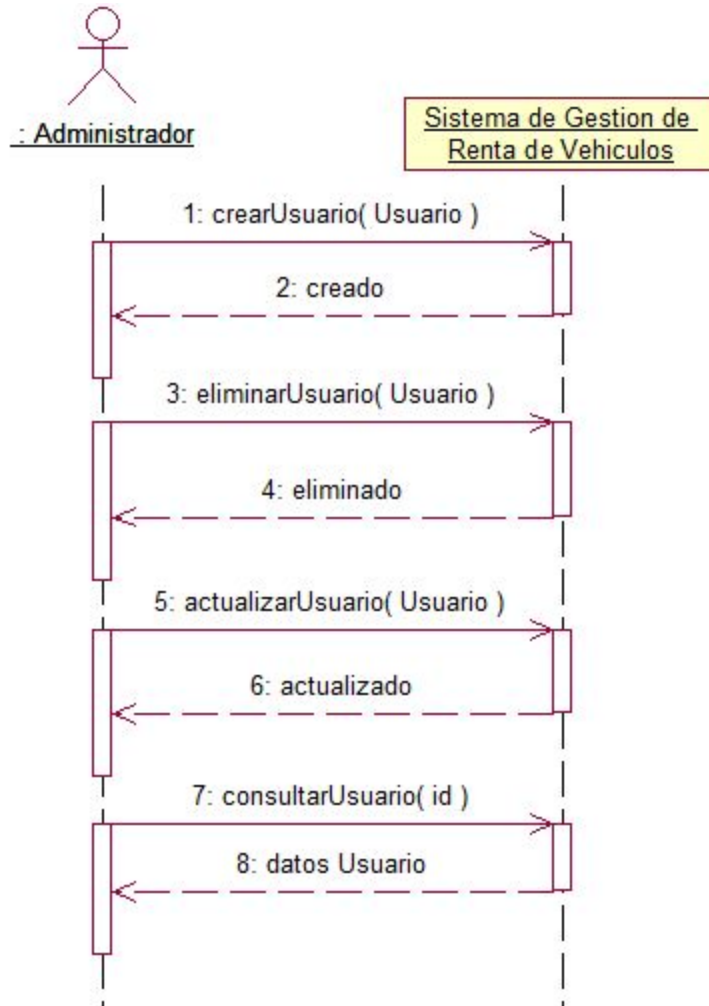
4.1 Diagramas de secuencia

CRUD Solicitud Renta

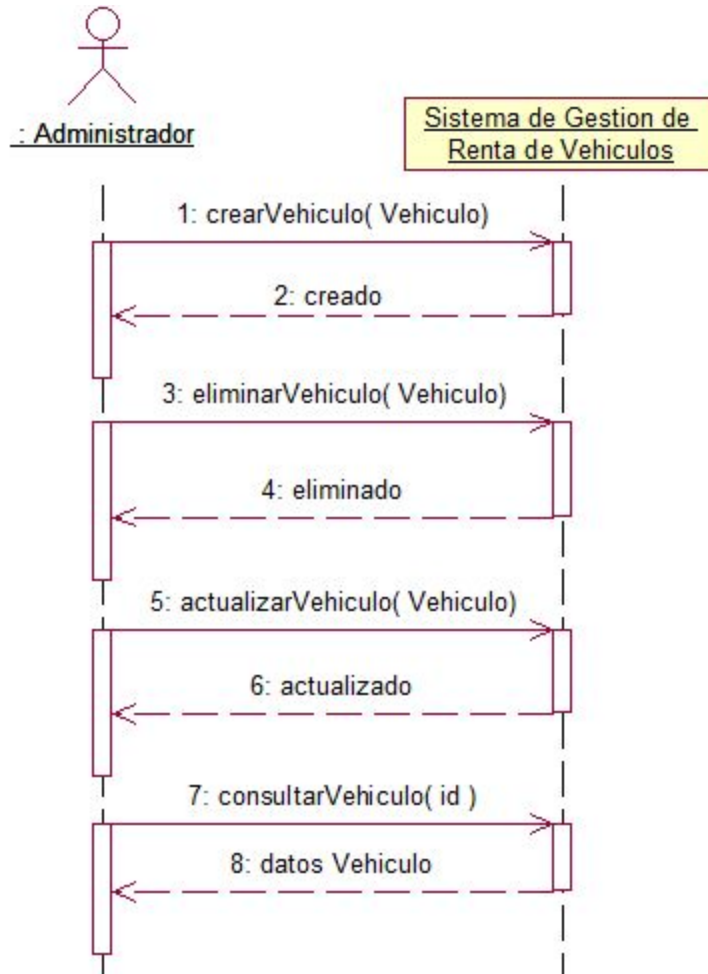
Universidad del Quindío



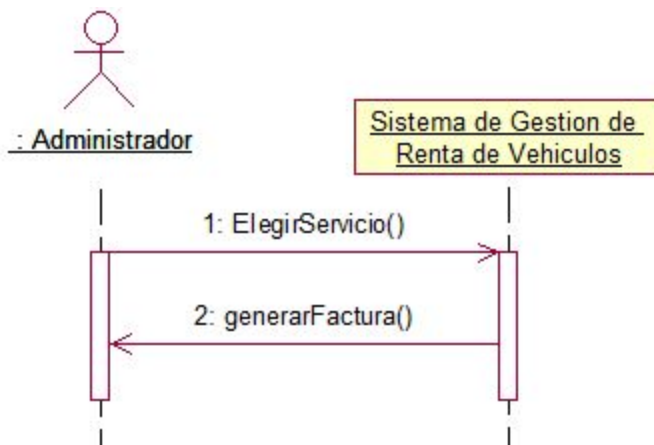
CRUD Usuario



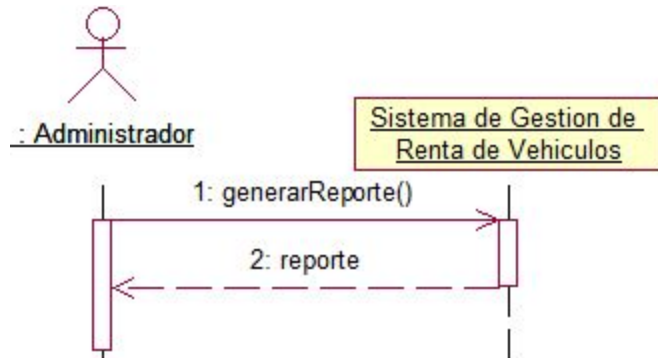
CRUD Vehiculos



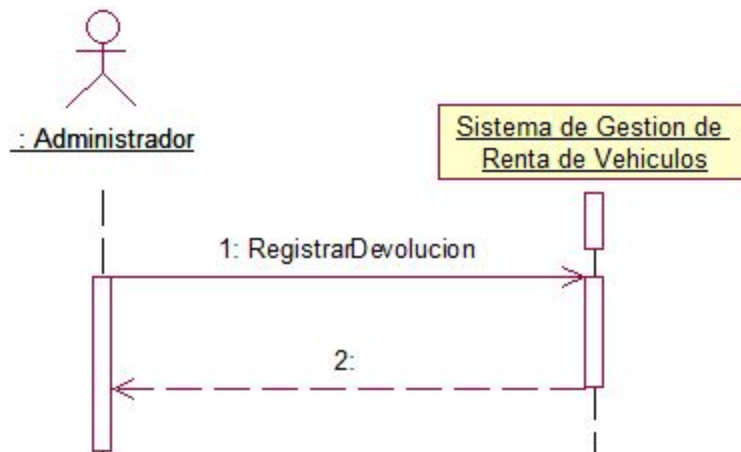
Facturar Servicio



Generar Reportes Rentas, Generar Reporte Kilometraje

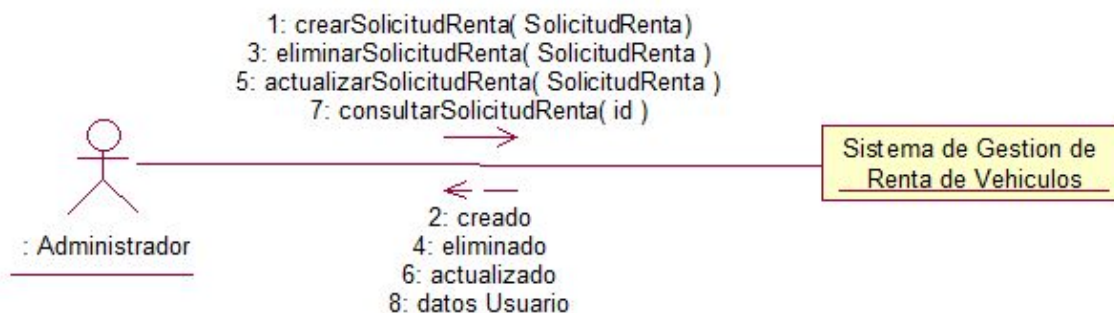


Registrar Devolución



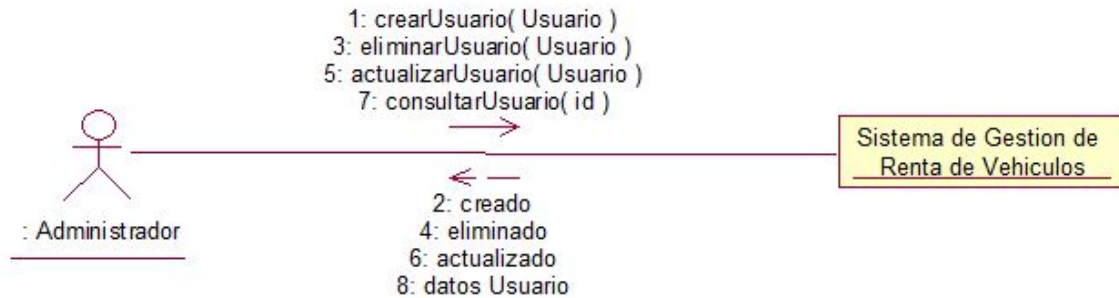
4.2 Diagramas de colaboración

CRUD Solicitud Renta

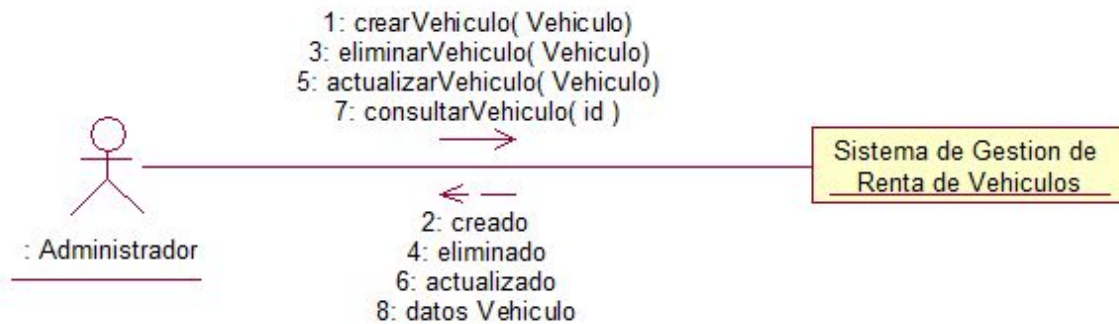


Universidad del Quindío

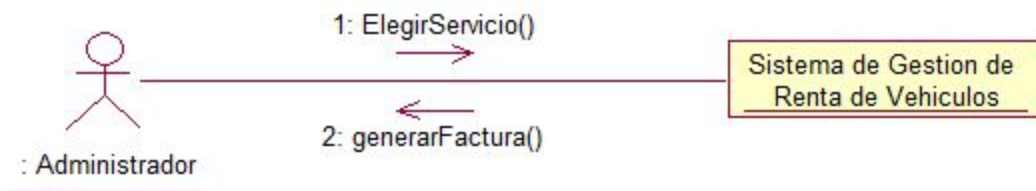
CRUD Usuario



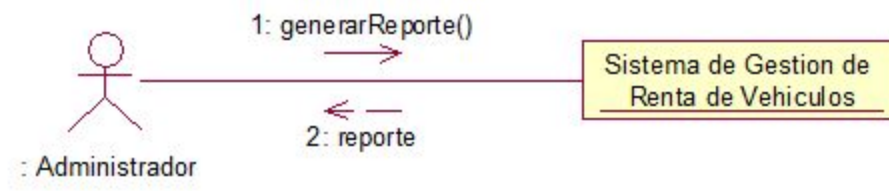
CRUD Vehiculo



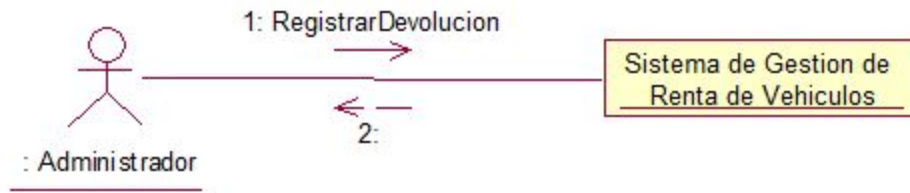
Facturar Servicio



Generar Reportes



Registrar Devolución



5. Vista Lógica

Visión General

La vista lógica del **Sistema de Gestión** está compuesta de cuatro paquetes principales: Interfaz de usuario (GUI), lógica, objetos de procesamiento (Entidades) y objetos de persistencia.

Paquetes de gran importancia para el diseño arquitectónico

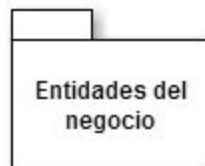
El paquete de Interfaz de usuario contiene la parte visual de la aplicación



El paquete de lógica contiene las clases que llevan a cabo los servicios de procesamiento como: Mostrar el reporte diario entre otros.



El paquete de entidades contiene las clases que utilizará el Sistema de Gestión de Renta de vehículos como por ejemplo: Usuario, Reporte...



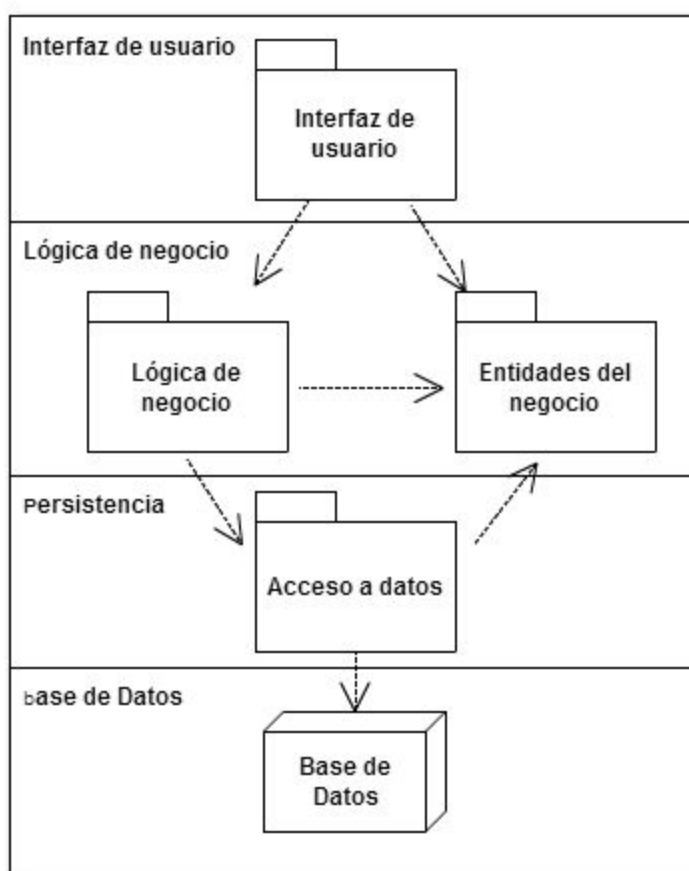
Universidad del Quindío

El paquete de Base de Datos contiene las clases que el Sistema de Gestión de Renta de vehículos utilizara para la persistencia.



Diseño de Capas

Representamos en la siguiente imagen la distribución de los paquetes en cada una de las capas del sistema.



Interfaz de usuario

Esta capa contiene el paquete de interfaz de usuario el cual almacena las clases con las cuales el usuario puede interactuar.

Lógica del negocio

Esta capa contiene el paquete de la lógica para el manejo de las operaciones del negocio.

Universidad del Quindío

Persistencia

Esta capa contiene el paquete de Base de Datos que brinda la persistencia al Sistema de Gestión de Renta de vehículos.

Realización de Casos de Uso

[This section illustrates how the software actually works by giving a few selected use-case (or scenario) realizations, and explains how the various design model elements contribute to their functionality.]

6. Vista de Procesos

Diagrama de clases

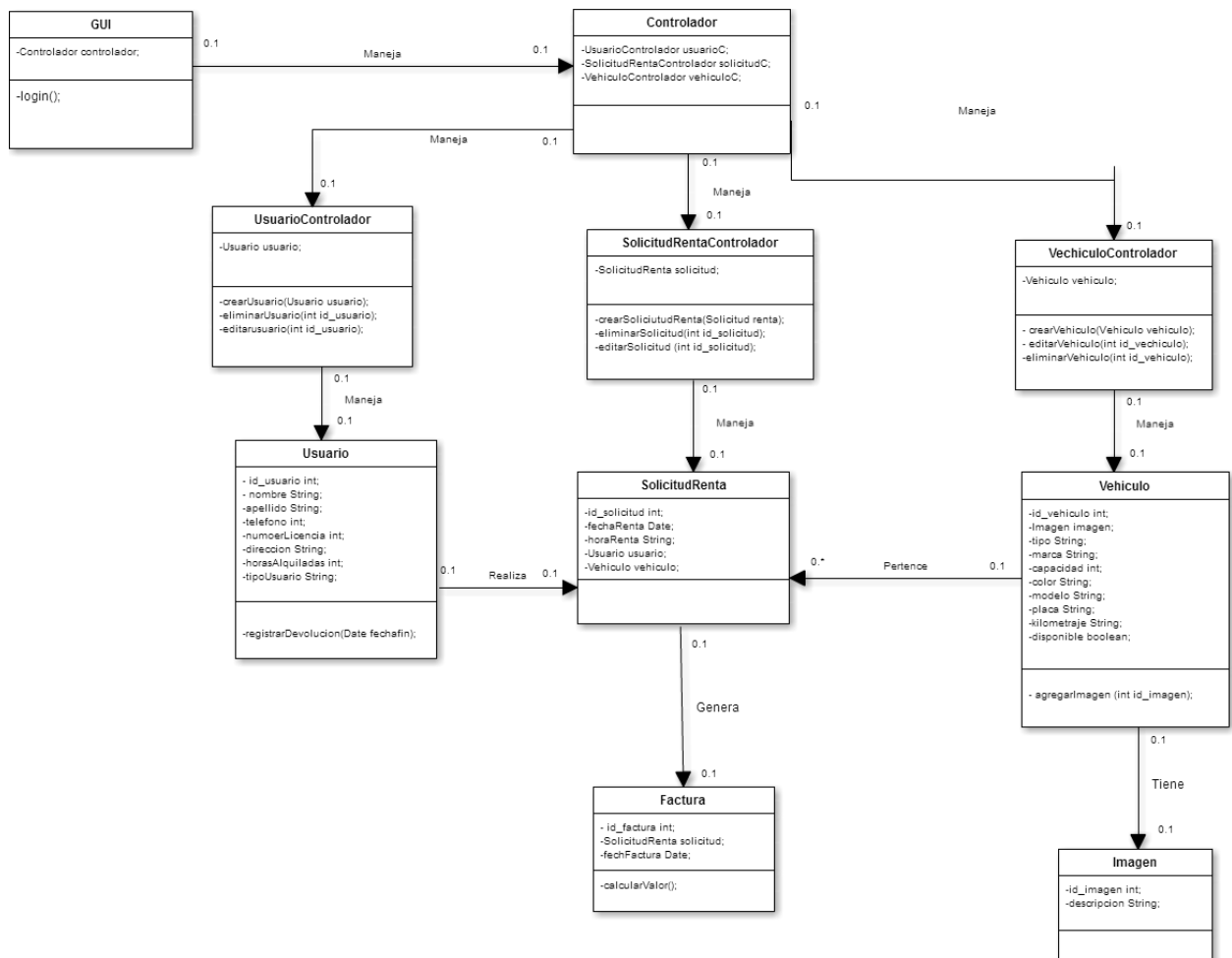


Figura 2. Diagrama de clases del proyecto.

7. Vista de Deployment

A continuación se presenta de manera gráfica los nodos que conforman el sistema.

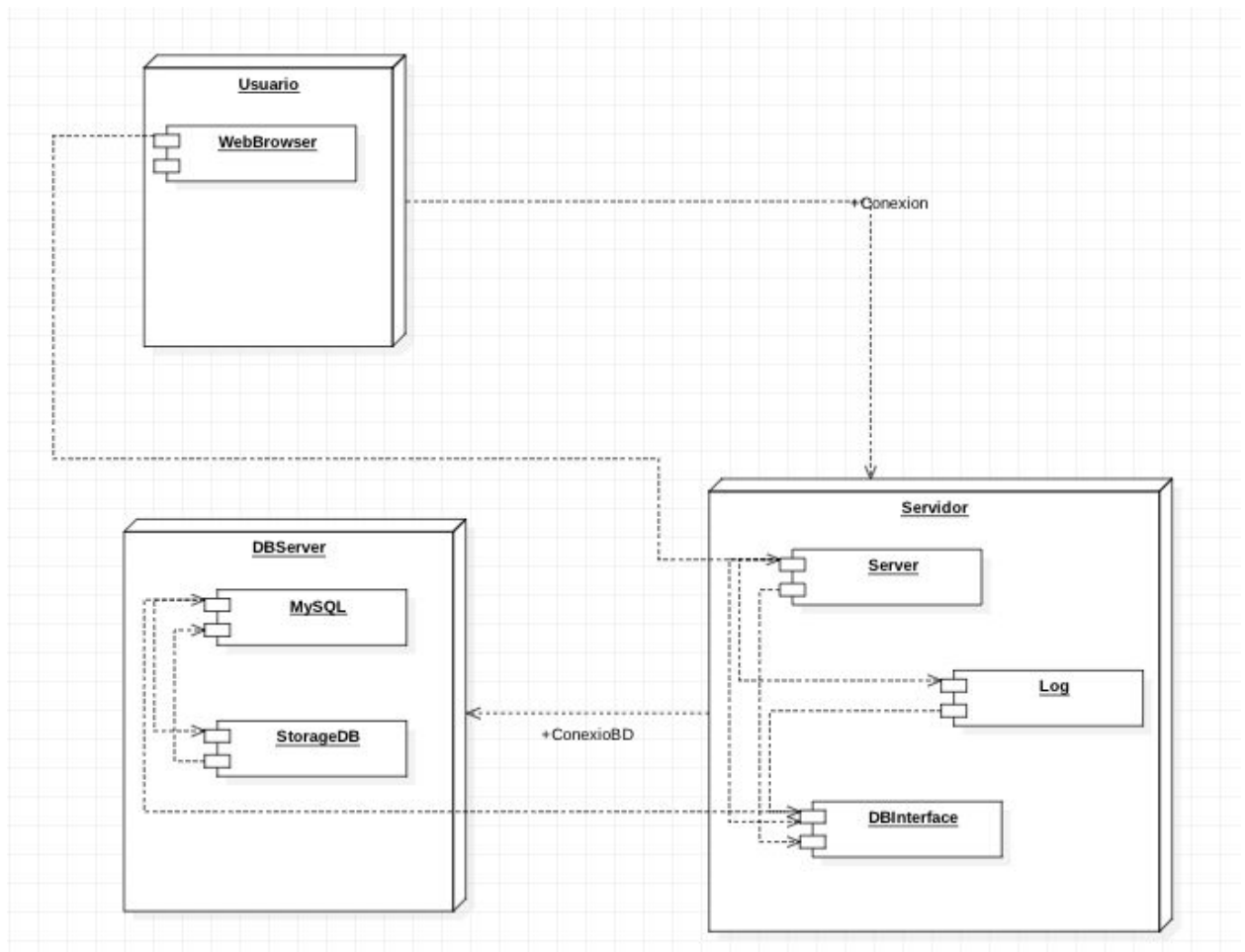


Figura 3. Diagrama de despliegue..

8. Vista de Implementación

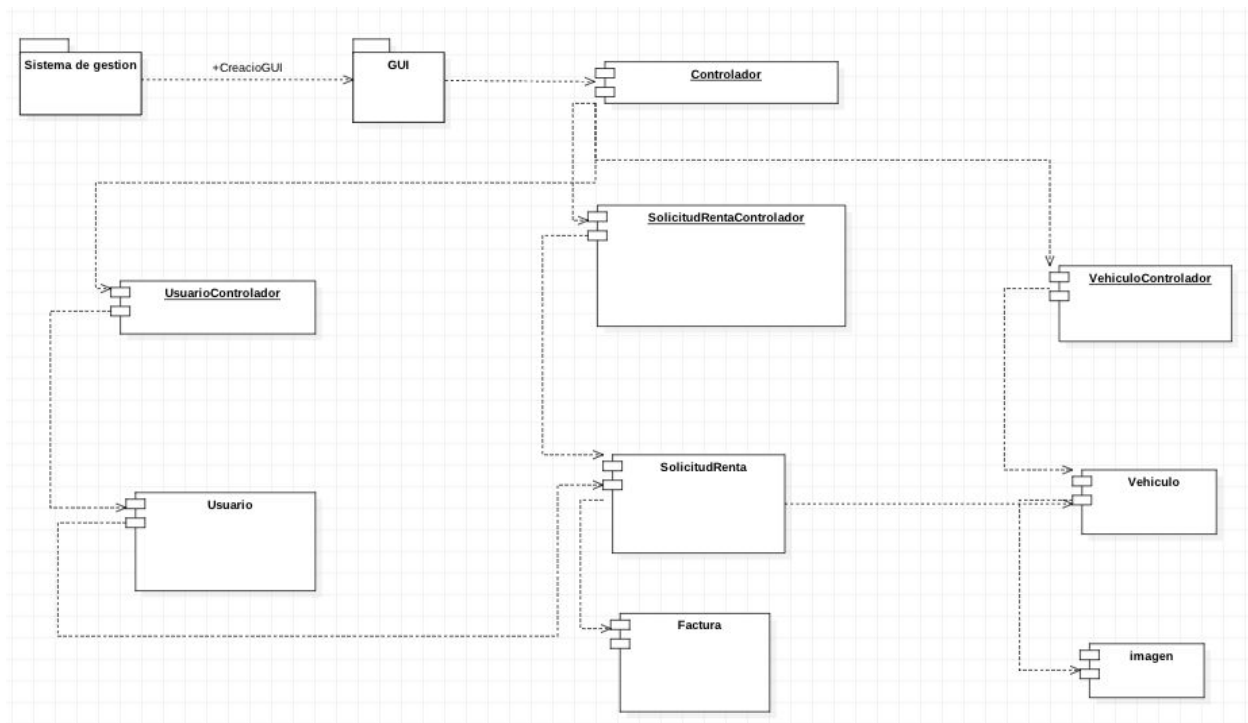


Figura 4. Diagrama de componentes.

Visión General

La vista de implementación seguirá el patrón de diseño Modelo Vista Controlador, el cual nos divide el problema en tres capas: La capa de modelo, la capa de controlador y la capa de vista.

Capas

- **Capa GUI**

El componente GUI maneja las clases que permiten la interacción del usuario con el sistema.

- **Capa Lógica**

El componente de Lógica permite manejar la lógica de negocio de todas las áreas.

- Usuarios.
- Reporte.
- Administración del sistema.
- Renta de vehículos.
- Devolucion de vehículos.

- **Capa de Base de Datos**

Este componente contiene la persistencia de los datos que maneja el Sistema.

9. Calidad

A continuación damos los siguientes requerimientos de calidad:

Funcionalidad

El sistema tendrá que cumplir con varios requisitos para que sea realmente funcional, entre ellos ser preciso, adecuado para los usuarios y el administrador(a) de los sistemas y sobre todo que tenga un grado muy alto de seguridad seguridad.

Fiabilidad

Es importante que todo sistema tenga gran rango de fiabilidad, eso incluye la madurez que pueda tener el software, sin contar con que el sistema tendrá que ser tolerante a errores, tener un plan b que haga al sistema tener un alto rango de recuperabilidad.

Usabilidad

La usabilidad de la aplicación tendrá cómo principio entender que no solo será importante basar la usabilidad en el usuario final, sino dentro de los miembros internos y funcionales de la misma, como lo son el administrador del sistema y usuarios , que son los principales involucrados en este proyecto.

Eficiencia

La eficiencia del proyecto final se refleja en el servicio ofrecido a los usuarios, reduciendo las fallas que se presentan normalmente con la asignación de vehículos, en los que se han evidenciado, por no tener una agenda adecuada y disponibilidad clara de los vehículos agendados.

Mantenibilidad

Para garantizar que el software sea mantenible, antes tenemos que hacer que el software sea escalable y compatible con nuevas versiones, esto quiere decir que a la hora de poder intervenir se puedan hacer cambios rápidos y eficaces. Es importante también tener en cuenta que la estabilidad del proyecto podrá sustentarse con un desarrollo sencillo y las herramientas adecuadas.

Portabilidad

El sistema tendrá la característica responsive y podrá ser desplegado en cualquier plataforma o dispositivo diferente al ambiente de producción o donde se este desarrollo. Lleva inmersa la despleabilidad y podrá ser visualizado sin ningún inconveniente.