**Vans & Lodging**

Documento de Arquitectura de Software

Versión 1.0

* Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 14/10/2022 | 1.0 | Versión inicial | Grupo 1 |
| 25/10/2022 | 2.0 | Versión final | Grupo 1 |
|  |  |  |  |

**Documento de Arquitectura de Software**

1. **Introducción**

Este documento representa la arquitectura del software “Vans & Lodging” en el que se van a presentar las distintas vistas de la arquitectura:

* **La vista de Casos de Uso** está comprendida por los requerimientos de funcionalidad del sistema, identificando las actividades principales y significativas para la arquitectura. Esta vista está apoyada por el usuario story mapping. Los principales stakeholders son los analistas de requerimientos y en general los dueños de la funcionalidad o quien necesite referirse a las necesidades de la implementación funcional.
* **La vista lógica** describe abstracciones clave, la distribución lógica de los componentes principales de la solución. Los interesados en esta vista son principalmente los diseñadores y desarrolladores.Los stakeholders como usuarios interesados, utilizan dicha vista para comprender cómo las unidades lógicas identificadas van a interactuar e identificar su ubicación.
* **La vista de procesos** permitirá observar las definiciones del tratamiento de las peticiones a nivel de hilos o de procesos de máquina. Los interesados principales son de infraestructura, pues esta perspectiva es un complemento importante para la vista de despliegue.
* **La vista de implementación** generalmente es la más extendida y utilizada para describir la solución por primera vez. También se utiliza para generar contexto arquitectónico y presentar el modelo del diseño. Los interesados son principalmente los desarrolladores y arquitectos, extendiéndose a cualquier interesado del dominio del sistema.
* **La vista de despliegue** está dada para la perspectiva de infraestructura, describiendo nodos y elementos físicos de alto nivel. Los interesados son de infraestructura y pretenden definir, a partir de esta vista, como el sistema será ubicado en la arquitectura de red y lineamientos del área de TI. En general cualquier stakeholder que desee conocer las decisiones de nodos físicos y sus relaciones.

Propósito

Sistema dedicado a la gestión de viajes y alojamientos, orientado a modernizar el sistema de transporte de pasajeros mediante Vans & Lodging utilizando una app móvil que mejore la experiencia de usuario del viajero y del conductor.

Alcance

El alcance del sistema involucra la creación de dos aplicaciones Mobile y una aplicación web para tres tipos de usuarios distintos, siendo estos el Pasajero, Conductor y Administrador de la plataforma.

El sistema opera en todo el territorio Argentino.

No se realizará la configuración ni la colocación de infraestructura del sistema. Se hará uso de ciertas API’s, como Google Maps para realizar todo lo relacionado con ubicación y mapas y MercadoPago para la gestión de pagos de reserva del servicio. No se hará cargo de la gestión de los distintos alojamientos asociados a Vans&Lodging, tales como la validez de datos o costos del servicio. Una vez el pasajero ingresa al alojamiento, Vans&Lodging no será responsable del alojamiento o los actos del pasajero.

Se incluirá una etapa de pruebas de campo de 6 meses posterior a la entrega final del sistema.

* 1. Definiciones, siglas y abreviaturas

|  |  |
| --- | --- |
| **Término** | **Descripción** |
| Pasajero | Actor el cuál hará uso de la aplicación Mobile. Será aquel usuario que realiza el caso de uso Realizar reserva, y utilizará los servicios de paquetes de viajes de Vans & Lodging. |
| Conductor | Empleado de Vans & Lodging que consume los servicios de la aplicación Mobile del conductor. Será el encargado de realizar los viajes y dar una actualización constante del estado del mismo. |
| Administrador | Empleado que consumirá la aplicación web. Será el encargado de realizar la gestión y planificación de los destinos, viajes, paquetes y asignación de transporte a viajes. |
| Alojamiento | Edificio en el cuál los pasajeros residirán mientras el paquete se encuentre activo. |
| Destino | Lugar el cuál el usuario Conductor se dirige. Para las instancias de inicio de viaje será el alojamiento, y para las instancias de vuelta del viaje será el punto de recogida inicial. |
| Van | Refiere a todos los medios de transporte físicos que posee habilitados Vans & Lodging para el transporte de pasajeros. Entre ellos se encuentran combis, autobuses de larga distancia, autos, etc. |
| Viaje | Es el trayecto que realiza un transporte con los pasajeros asignados. Tiene un punto de inicio, uno o más puntos de recogida y uno o más destinos. Para cada pasajero dentro del transporte, el viaje inicia en su punto de recogida y finaliza en su alojamiento. Realizando el camino inverso, el pasajero tiene como punto de recogida el alojamiento, y como destino el punto de recogida inicial. |
| Paquete | Es la suma entre el viaje y el alojamiento. Debido a que el precio final a pagar por el cliente depende de ambos, existe un paquete el cual incluye al total de estos dos.Un cliente puede reservar un paquete para muchas personas |
| Punto de recogida | Lugares físicos los cuáles el conductor debe de recoger previo a dirigirse al destino final del viaje. Dichos puntos serán el inicio del viaje de cada pasajero. |

* 1. Referencias

Zip wireframes:



1. **Representación arquitectónica**

Para la Arquitectura de Software se utilizó el estilo de Arquitectura basado en capas porque nos permite desacoplar las funcionalidades entre componentes cohesivos, donde cada capa tiene una única responsabilidad. Específicamente se emplearán 3 capas: la capa de presentación o interfaz, la capa de modelo o negocio y la capa de acceso a datos o persistencia.

1. **Objetivos de la arquitectura y Restricciones**

El objetivo de la arquitectura es poder visualizar los diferentes servicios y componentes, además de poder balancear satisfactoriamente las cargas, que ocurra un fallo en algún servidor o servicio, el mismo pueda aislarse correctamente.

Las capas elegidas nos permiten relevar los requerimientos de la siguiente manera:

* **La capa de presentación o interfaz**: Es la responsable de manejar la interfaz gráfica con el usuario (pasajero, administrador y conductor) como vistas renderizadas.
* **La capa de modelo o negocio:** Es la responsable de describir las reglas, operaciones y restricciones, del negocio. Consta de las tareas, los pasos de los procedimientos, la información de entrada y salida requerida, y los servicios necesarios para ese procedimiento.
* **La capa de acceso a datos o persistencia:** Es la responsable de almacenar u obtener los datos necesarios para la operatoria del negocio.

**Beneficios**

* **Seguridad (Integridad)**
  + El servidor de datos se encuentra en la última capa limitando el acceso a ataques que puedan afectar su integridad.
* **Mantenibilidad**
  + Debido a que cada capa hace una tarea específica, es fácil detectar dónde ocurrió un error, y cómo corregirlo.
  + Debido a que cada capa tiene una responsabilidad específica, es fácil detectar dónde aplicar cambios de nuevos requerimientos.
  + Los componentes pueden ser reutilizables.

1. **Vista de Casos de Uso**

Se representan los requisitos funcionales mediante el uso de un mapa de historias de usuario.

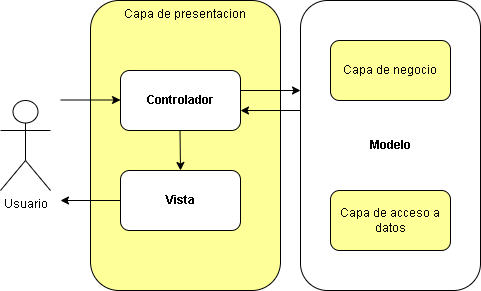
* 1. Realización de los Casos de Uso

No aplica.

1. **Vista Lógica**

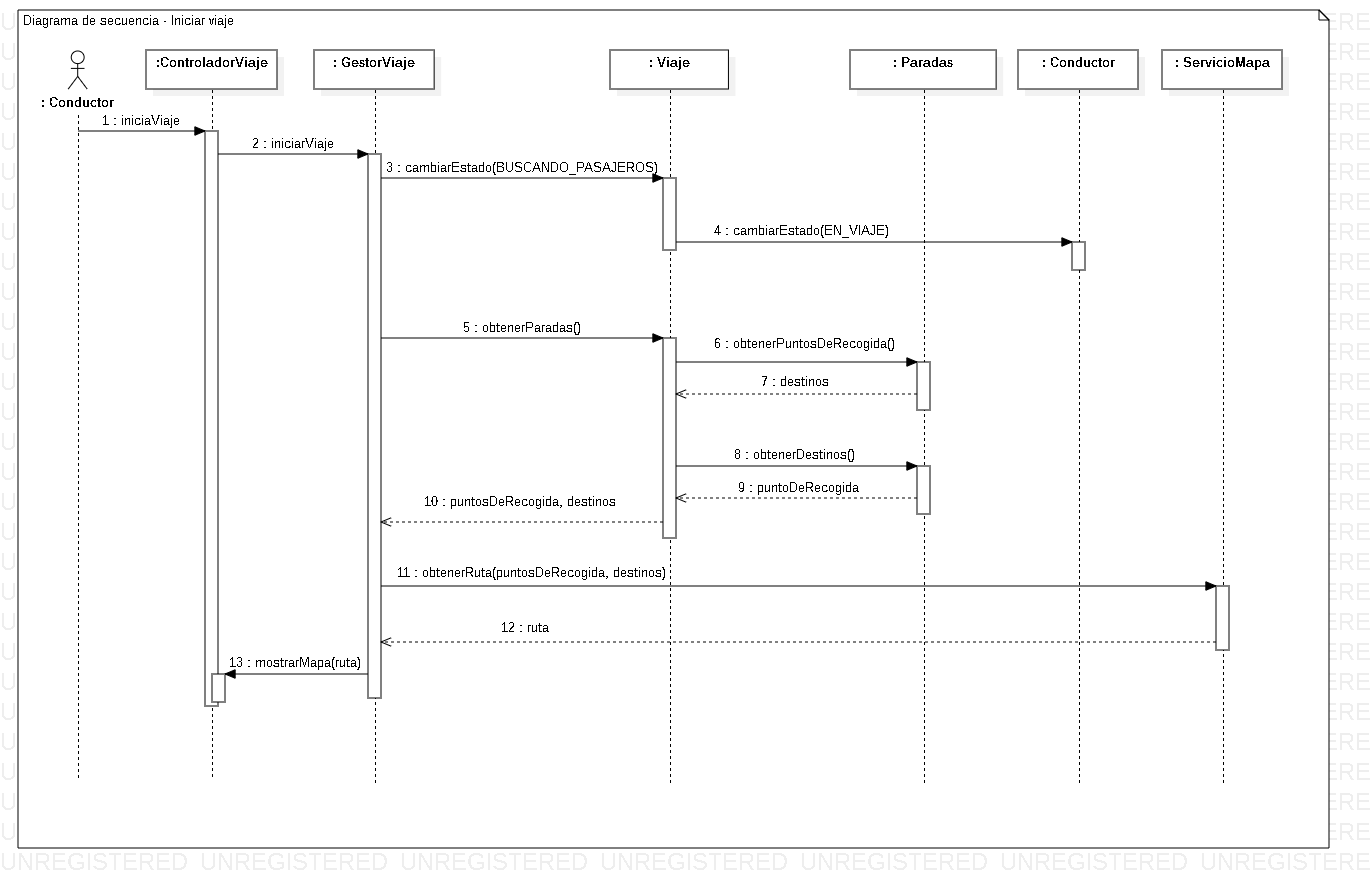
Se aplica el patrón arquitectónico modelo-vista-controlador (MVC) porque nos permite organizar las capas en tres componentes que proporcionan una vista clara de cómo el Usuario va a interactuar con la aplicación, a la vez la responsabilidad que tienen cada componente y cómo deben relacionarse entre sí:

* **Controlador**: Será el responsable de administrar las solicitudes del usuario (pasajero, administrador o conductor), seleccionar el comportamiento del modelo y actualizar las vistas correspondientes.
* **Vista**: Será el responsable de preparar los datos y solicitar actualizaciones del modelo y presentar la vista seleccionada por el controlador.
* **Modelo**: Será el responsable de contener las funciones, los objetos de contenido y de incorporar estados del sistema.

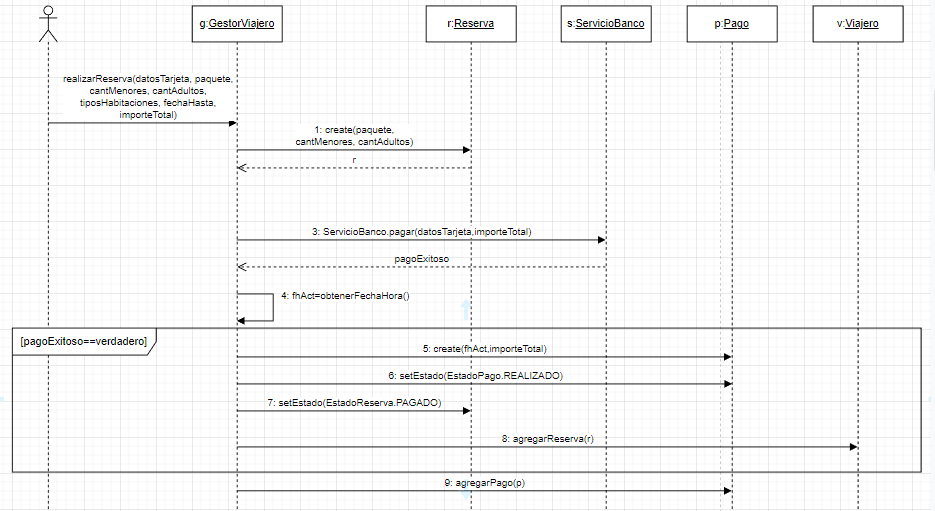


* 1. Visión general
* [Diagrama de clases - Conductor](https://drive.google.com/file/d/1XBdFd2hYULUvEIQxoKSNsTzLhY8ivDBR/view?usp=sharing)
* [Diagrama de clases - Viajero](https://drive.google.com/file/d/1xJMvfN6EphCYej89SCwKTTSZpW4Wyzqs/view?usp=sharing)
* [Diagrama de clases - Administrador](https://drive.google.com/file/d/1eNDE4IEiIE9ubVM-6kma5RmvMWGVuVb_/view?usp=sharing)
* Diagrama de secuencia

Escenario: Iniciar Viaje

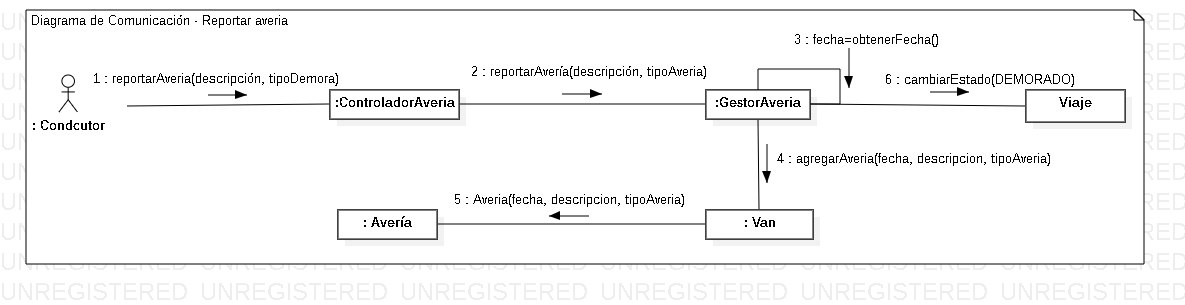


Escenario: finalizar Reserva

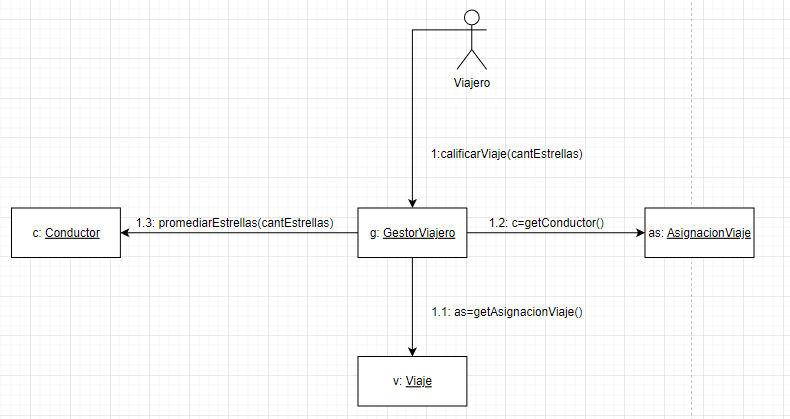


* Diagrama de Colaboración

Escenario: reportar avería



Escenario: Calificar conductor

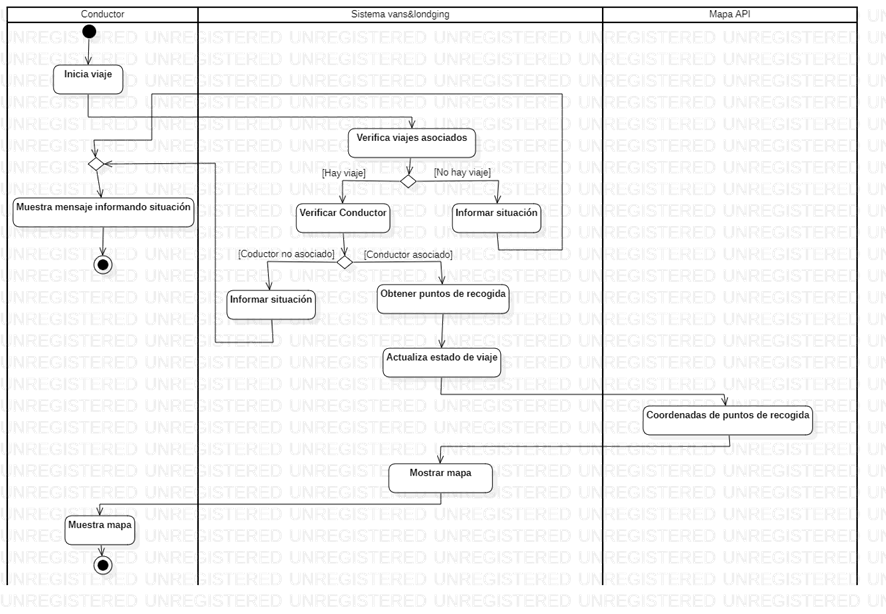


5.2 Paquetes de Diseño Significativos

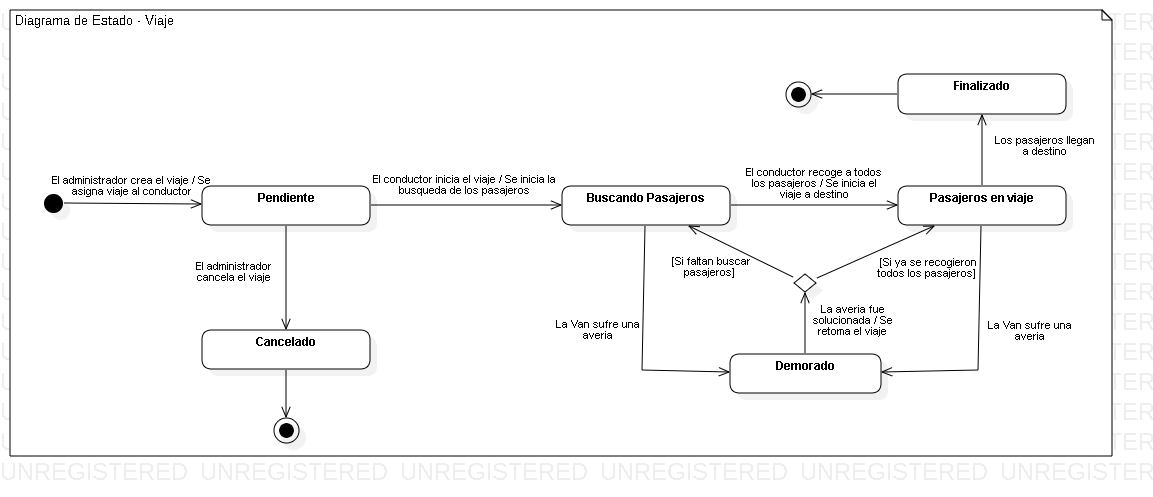
No aplica.

1. **Vista Proceso**
   1. Visión general

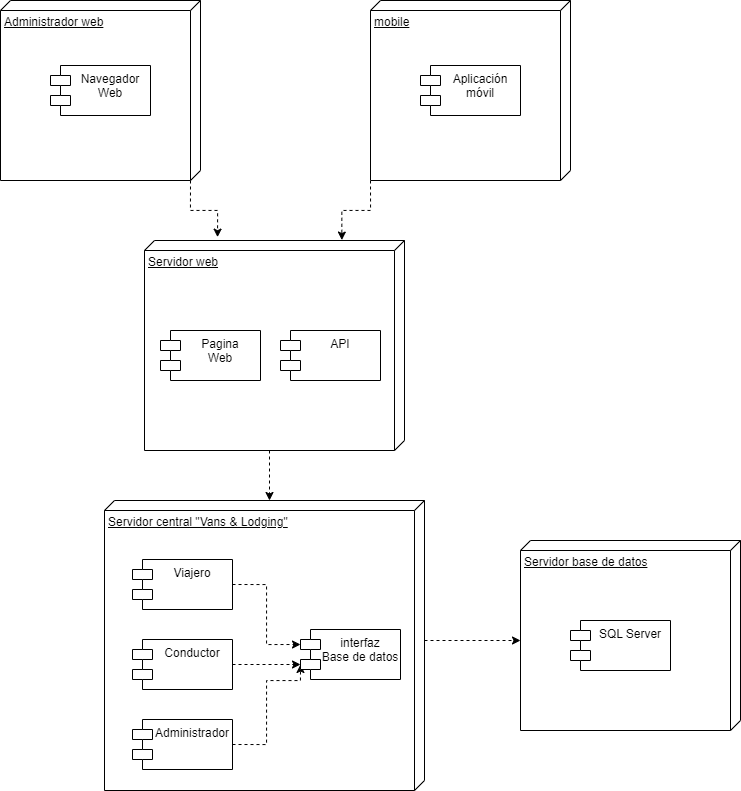
* Diagrama de Actividad - Iniciar viaje



* Diagrama de estado - Viaje



1. **Vista Despliegue**

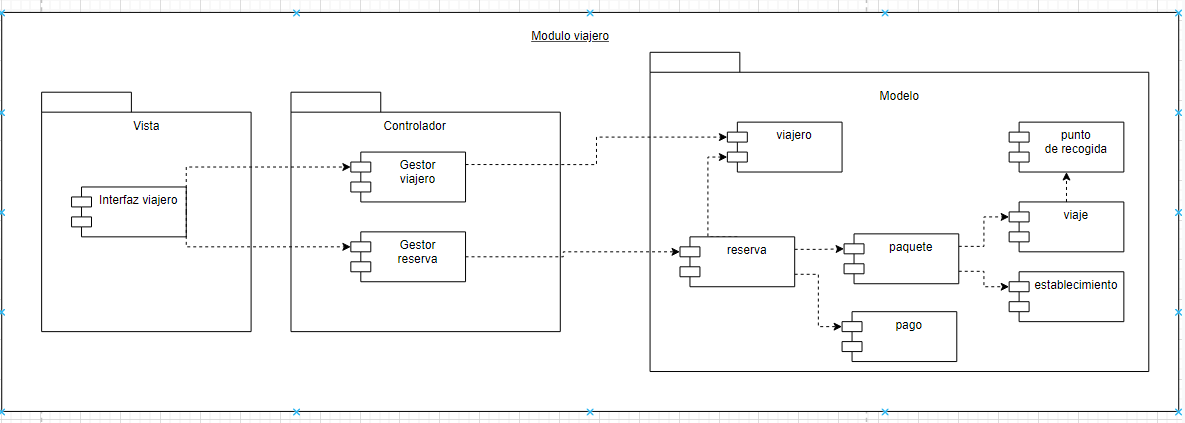


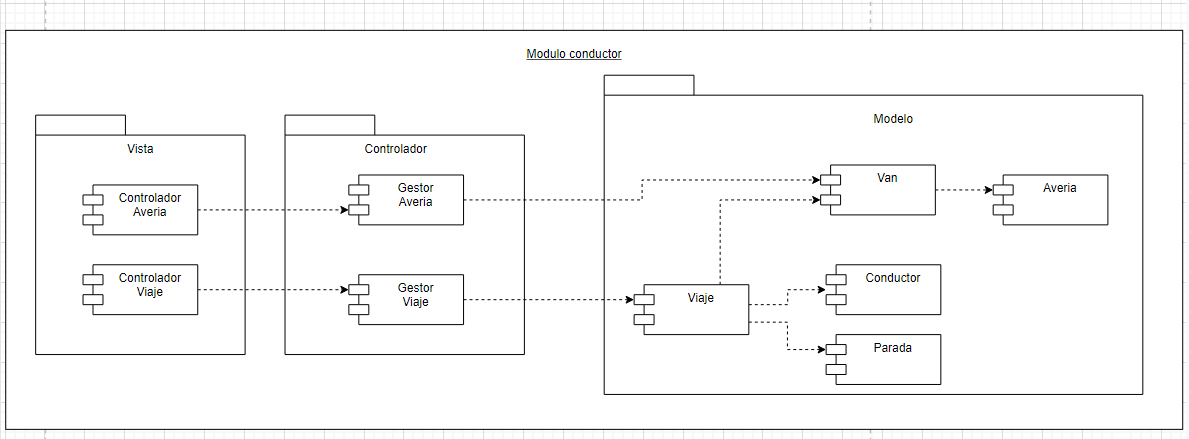
1. **Vista Implementación**

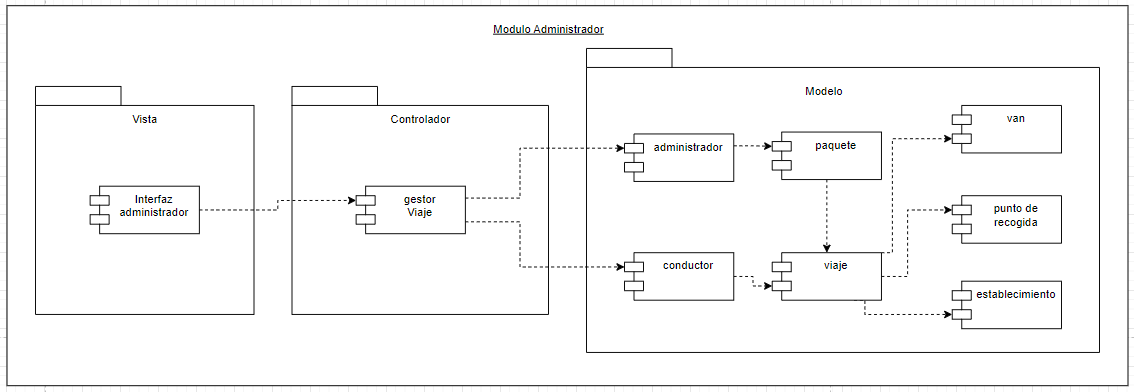
**8.1** **Visión General**

Se separa el diagrama de componentes en los 3 módulos (viajero, conductor, administrador) para un mayor entendimiento de la implementación.

**8.2** **Capas**







1. **Vista de Datos**

No aplica.

1. **Tamaño y performance**

De acuerdo con las estadísticas del año 2021 en Argentina circularon 62 785 para transporte de pasajeros, en un principio esperamos que un 40% del total utilice la aplicación.

De acuerdo con las estadísticas del año 2020 en CABA hay 3.075.646 de habitantes de los cuales esperamos que un 10% utilice la aplicación.

Se espera que el sistema pueda filtrar los requerimientos del pasajero para la elección del paquete en menos de 5 segundos, esto implica que el paquete sea reservado correctamente.

Se establece en el SLA (Service Level Agreement) una disponibilidad del 98%, permitiendo un 2% de margen de error en caídas mensuales.

1. **Calidad**

***Usabilidad:***

Este sistema posee interfaces de usuario sencillas que permiten que los usuarios, tengan el conocimiento que tengan, puedan acceder y entender el sistema de Vans & Lodging sin inconvenientes. Además, se podrá navegar de manera intuitiva y fluida, lo que se traduce en una satisfactoria experiencia de usuario. Esto se demuestra en las pruebas de card sorting donde se determinaron los nombres, menús y submenús de los elementos.

***Estética de la interfaz de usuario:***

La interfaz aplicada al sistema, y por consiguiente la identidad de Vans & Lodging, está orientada al color celeste; debido a su asociación al color del cielo y el mar que son las principales imágenes que se vienen a la mente de una persona al decir vacaciones. Además el logotipo de la empresa tiene alta presencia de azules por lo que es parte de la identidad principal de la compañía y debemos respetarlo.

***Capacidad de aprendizaje:***

El sistema realiza un recorrido por las funcionalidades principales a los usuarios que ingresan por primera vez a modo de tutorial. Esto permite acelerar el aprendizaje de cada usuario.

***Portabilidad:***

El sistema debe permitir para la interfaz web, el uso de navegadores Firefox, Chrome y Microsoft Edge.

El sistema debe permitir para la interfaz móvil, su uso en dispositivos con SO Android (SO Android 7 o superior) y IOS (versión 8 o superior).

***Seguridad (Integridad):***

El sistema debe guardar el historial de registros de viajes, sin permitir alguna alteración. Se mantendrán persistidos los últimos 1000 viajes por cada pasajero. Luego de superar el límite de viajes se irán eliminando los más antiguos a medida que se agregan nuevos manteniendo esta cantidad de 1000 viajes.

El sistema guarda las contraseñas utilizando un método hash and salt como bcrypt.

***Fiabilidad (Tolerancia a fallos):***

El sistema estará compuesto de un balanceador de carga para los 2 servidores Web de la aplicación, distribuyendo las peticiones entre los 2 servidores. En caso de falla de un servidor, el segundo podrá soportar la carga de ambos mientras se vuelve a poner en funcionamiento el primero.

***Eficiencia (comportamiento temporal)***

El tiempo de respuesta para cualquier interacción del usuario con el sistema será inferior a 3 segundos.Con un hardware mínimo de 2 gb de RAM y un procesador de 3 núcleos.

***Mantenibilidad***

Este sistema tiene la capacidad de ser ampliable adaptable y configurable para versiones futuras.