

**RTF2: Diagrama de despliegue
(deployment)**



Célula de trabajo Gryffindor

Docente de la asignatura

ROBINSON CORONADO GARCIA

1. Descripción del proyecto

Este proyecto de arquitectura de software presenta la metodología de desarrollo utilizada en la realización de una solución de monitoreo para vehículos pesados con dispositivos GPS, el software que se presenta, forma parte del soporte de monitoreo de vehículos pesados que se desea ofrecer por medio de una página web. Se describe y especifica la arquitectura del sistema a través de diagramas de las vistas de la arquitectura.

La solución propuesta agrupa las funcionalidades representativas de los mejores atributos de calidad y las presenta de manera objetiva a través de una propuesta completa y puntual donde el usuario final pueda dar uso de la solución a través de un navegador web, así mismo se muestra el diseño de la solución utilizando los estándares de calidad y patrones que exige la ingeniería de software.

Los usuarios de este software pertenecen a diferentes sectores, como: minería, pesca, transporte de valores, transporte interdepartamental, transporte de alimentos, fuerzas armadas, etc. Todos estos usuarios tienen vehículos pesados de diferentes tipos que recorren el territorio nacional en toda su extensión y requieren de la información necesaria para monitorear la posición exacta, la ruta que siguen estos y estar informados de cualquier eventualidad. Nuestra solución les ofrece a estos clientes el servicio de transmisión de datos que se obtenga con el rastreo satelital.

El usuario deberán tener como dispositivo GPS la sim card en un celular con la finalidad de monitorear la posición geográfica de sus vehículos, este monitoreo es realizado a través de una aplicación de monitoreo vehicular en un servidor web

2. Fases del proyecto

2.1 Análisis de requerimientos

En esta fase del proyecto abordaremos el problema que se presenta con algunos propietario de vehículos de carga pesada, al no contar con información de sus vehículos en tiempo real.

Estos propietarios de vehículos pesados tienen la necesidad de contar con un sistema de monitoreo que les permita llevar un mejor seguimiento de sus vehículos, que a su vez sea fácil de manejar; el monitoreo consiste en llevar a cabo un registro de todos los datos en tiempo real del vehículo, las rutas frecuentes y demás información pertinente. Con el fin de que los propietarios por medio de una página web pueden consultar por la información de sus vehículos

2.2 Diseño

Se desea implementar una interfaz web intuitiva. Una infraestructura de comunicación disponible y eficiente. Herramientas de geo-información que ayuden al fácil entendimiento de la información. Herramientas Open-Source para la Arquitectura del Sistema. La arquitectura será cliente servidor donde el cliente es el navegador web y el servidor comparte funciones como web server, servidor de comunicaciones, servidor de base de datos y servidor de mapas. GPRS (General Packet Radio Service) como uno de los medios de comunicación entre la unidad vehicular y el servidor, y la información recolectada del GPS es almacenada en el servidor de base de datos.

Se contempla usar comunicación vía sockets, un servidor web y un servidor de mapas. El servidor de comunicación vía sockets será el componente central que establecerá conexión con las unidades rastreadas, establece una conexión TCP/IP y es capaz de comunicarse con múltiples unidades utilizando múltiples hilos. Cuando una unidad vehicular se conecta, el servidor la autentica y reconoce al cliente. Luego, el servidor procederá a recibir la información del cliente y almacenarlo en la base de datos. La aplicación web tomará la data de la base de datos y la procesa cuando el usuario haga futuras peticiones. Así mismo la aplicación web es una herramienta de mantenimiento del sistema, ya que un usuario puede ingresar al sistema y en función a su nivel de permisos, puede acceder a diversos módulos funcionales de la aplicación.

2.3 Desarrollo

Para el desarrollo del software adoptaremos la metodología de trabajo en cascada, ya que es adecuado para este proyecto dado que sólo cuando se finaliza una fase, comienza la otra. Se procurará realizar una revisión antes de iniciar la siguiente fase de desarrollo, lo que permitirá la posibilidad de cambios (lo que puede incluir un proceso de control formal de cambio). Las revisiones también serán utilizadas para asegurar que la fase anterior ha sido totalmente finalizada; los criterios para completar una fase serán dados por el analista. Al seguir este modelo evitamos visitar y revisar fases que ya se han completado.

2.4 Prueba

Las pruebas del software serán ejecutadas por el Tester, en esta fase nos cercioramos de cumplir con los requisitos funcionales y no funcionales establecidos en la etapa de diseño del software, se verificará que se cumpla con los protocolos y se informará cualquier inconveniente que presente el software que le impida cumplir su función de monitoreo de manera eficiente.

En primer lugar tendremos las Pruebas de Software Funcionales. Aquí encontraremos el comportamiento del sistema, subsistema o componente software descrito en especificaciones de requisitos o casos de uso, aunque también puede no estar documentado . Es decir, con las funciones establecemos “lo que el sistema hace”.

Estas pruebas se definen a partir de funciones o características (como decimos, bien descritas en documentos o bien interpretadas por los probadores) y su interoperabilidad con sistemas específicos, pudiendo ejecutarse en todos los niveles de pruebas (componentes, integración, sistema, etc).

En segundo lugar figuran las Pruebas Software no Funcionales que incluyen las pruebas de: Rendimiento, Carga, Estrés, Usabilidad, Mantenibilidad, Fiabilidad o Portabilidad, entre otras. Por tanto se centran en características del software que establecen “cómo trabaja el sistema”.

2.5 Mantenimiento

En la fase de mantenimiento el equipo se encargará de la corrección de defectos en el software, creación de nuevas funcionalidades en el software por nuevos requisitos de usuario y mejora de la funcionalidad y del rendimiento.

3. Objetivos

3.1 objetivo principal

- Crear un sistema de monitoreo satelital de una red de soporte para transporte pesado.

3.2 Objetivos específicos

- Se podrá ingresar al sistema y en función a el nivel de usuario, poder acceder a diversas funcionales de la aplicación.
- El sistema tendrá compatibilidad con los hardware(sim cards) más usados en el mercado de las telecomunicaciones.
- Ofrecer la ubicación en tiempo real de los vehículos.
- Suplir Pérdida de productividad y de tiempo en el transporte de empleados.
- Falta de un sistema adecuado para guardar un registro del recorrido de vehículos pesados.

4. Alcance

El software a construir tiene, como objetivo principal, ofrecer un sistema de monitoreo satelital de una red de soporte para transporte pesados. Se desea facilitar, fundamentalmente, la ubicación en tiempo real y el guardado de los registros del recorrido de los vehículos, de diferentes propietarios. En cuanto a la contratación del servicio, debe facilitarse su gestión integral, desde el momento en que un propietario ingresa un vehículo, pasando por el pago de la suscripción al servicio, la gestión de sus vehículos.

5. Análisis de requerimientos

5.1 Requisitos funcionales

- **Identificación y autenticación:** Se contará con un usuario y contraseña para la validación de la identidad.
- **Gestión de seguridad:** Definición de usuarios y niveles de acceso asociado a su función.
- **Gestión de ubicación:** El sistema debe brindar con exactitud y en tiempo real la ubicación del vehículo.
- **Gestión de información:** El sistema debe brindar la información pertinente de los vehículos a los cuales se esté monitoreando.
- **Gestión de almacenamiento:** El sistema cuenta con un mecanismo de almacenamiento persistente.
- **Gestión de membresía:** El sistema contará con un método de membresía para el registro de usuarios en la web.
- **Compatibilidad:** El sistema contará con compatibilidad a los hardware(sim card) más utilizados del mercado de telecomunicaciones en el país de Colombia.
- **Gestión de vehículo:** El sistema contará con una interfaz para el ingreso de nuevos vehículos.
- **API:** El sistema deberá contar con su debido funcionamiento con la API de Google Maps.
-
- Ordenar los requisitos según el impacto de los objetivos del negocio
 - **Gestión de ubicación**
 - **Gestión de información**
 - **API**
 - **Gestión de seguridad**
 - **Identificación y autenticación**
 - **Gestión de vehículo**
 - **Compatibilidad**
 - **Gestión de almacenamiento**
 - **Gestión de membresía**

5.2 Requisitos no funcionales

Experiencia de usuario(UX): El sistema contará con una interfaz amigable, agradable e intuitiva para el usuario.

Escalabilidad: Nuestro sistema contará con la capacidad de crecer en cualquier aspecto que se le requiera debido a las nuevas necesidades de los clientes, ya que en la fase de testeo se implementarán todo tipo de pruebas, específicamente pruebas de estrés.

Rendimiento: El sistema podrá responder de manera rápida y eficiente sin importar la carga que se le presente. También se harán las pruebas necesarios para satisfacer este requerimiento.

Seguridad: Se brindará total seguridad tanto del sistema, como en el ambiente en el que desenvuelve este.

Eficiencia: El sistema realizará cualquier labor para la que fue construido sin problema alguno y en cualquier momento sin importar las condiciones en las que se encuentre.

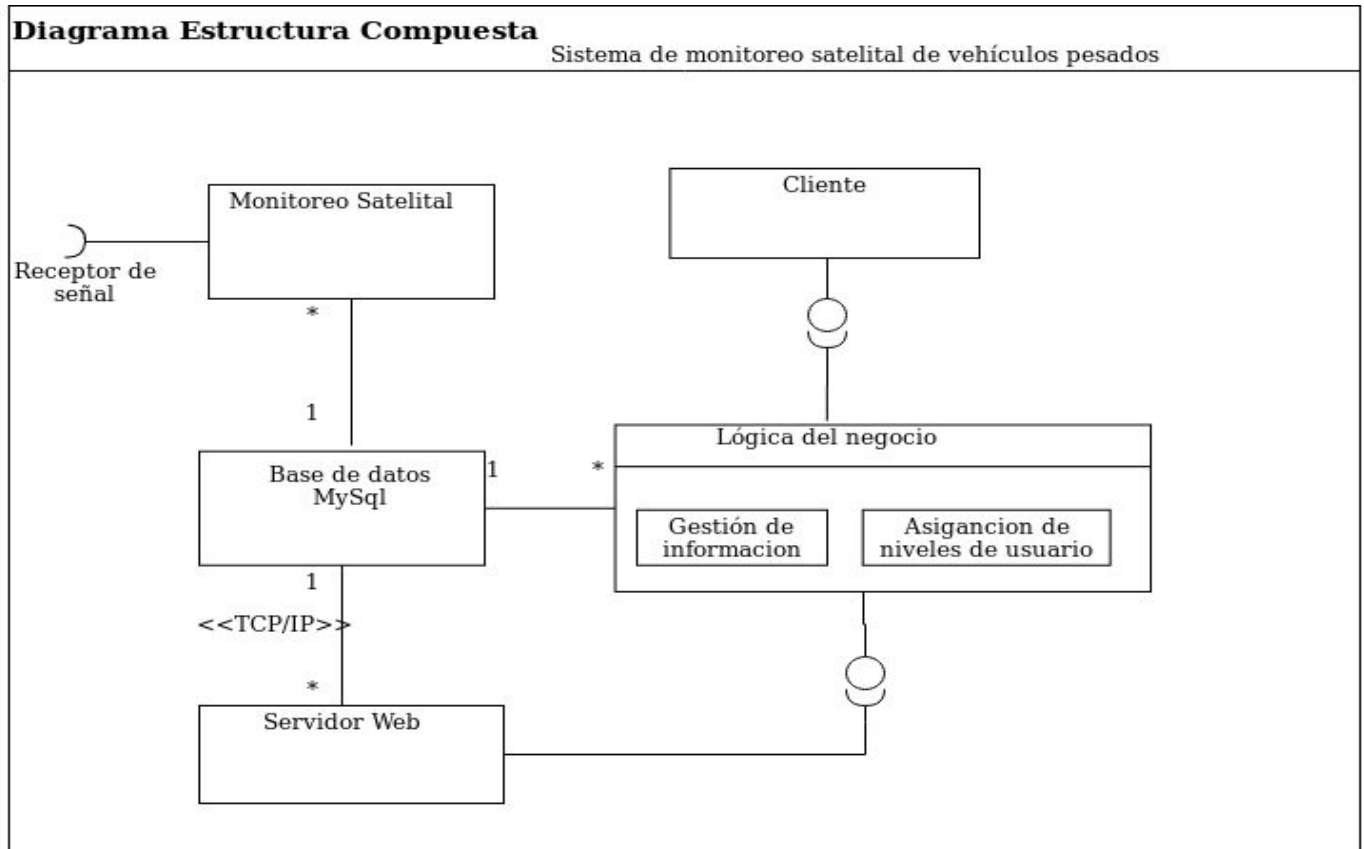
Soporte: El sistema contará con un soporte rápido y eficaz para cualquier inquietud o problema que este le presenta al usuario.

Documentación: El sistema contará con su respectivo manual de uso, además de que cada parte del desarrollo como tal de dicho sistema estará documentado.

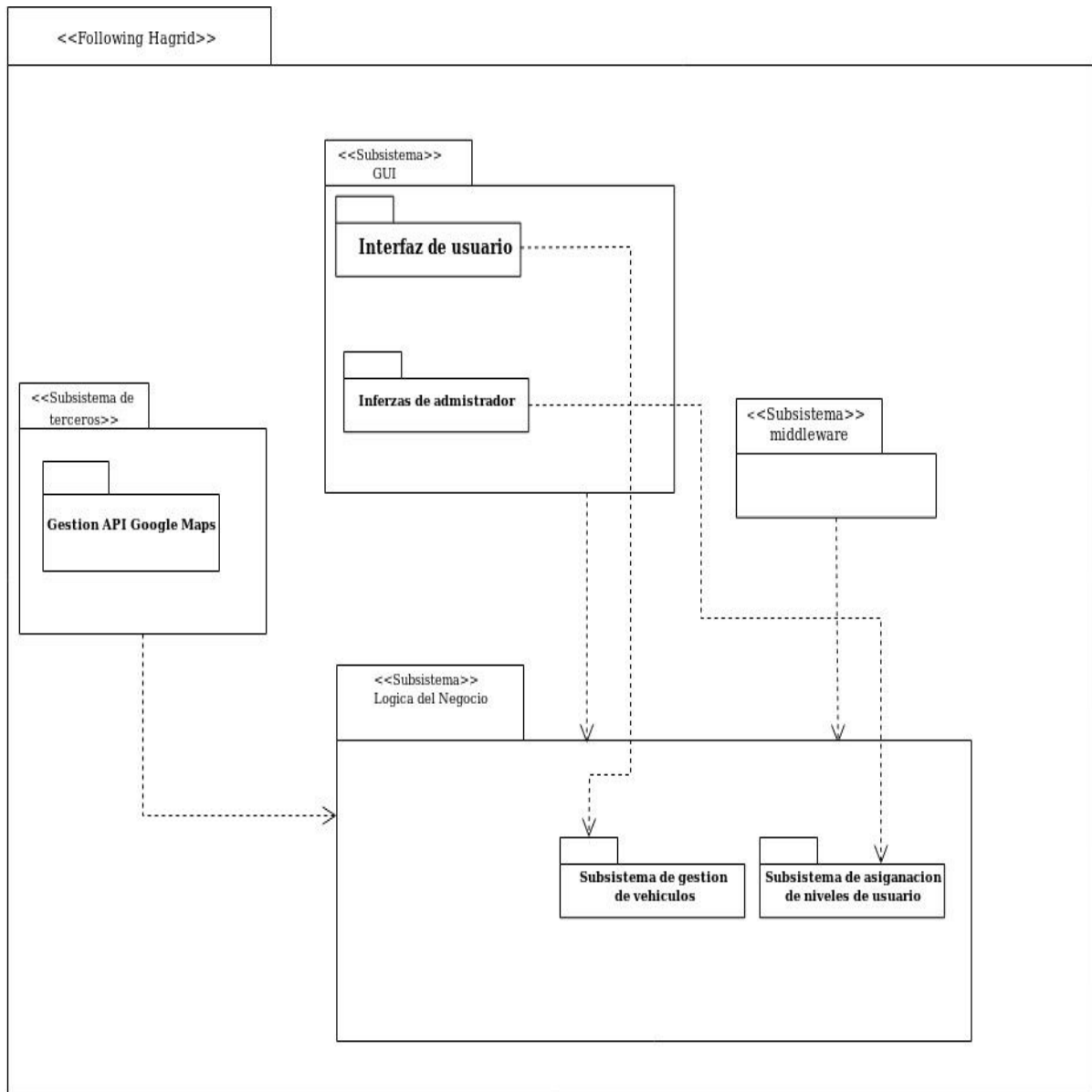
Compatibilidad: Se tendrá total compatibilidad con los principales sistemas operativos, navegadores y demás herramientas tanto de software como de hardware que se necesiten para su perfecto funcionamiento.

Portabilidad: El sistema podrá ser llevado en cualquier dispositivo que se necesite.

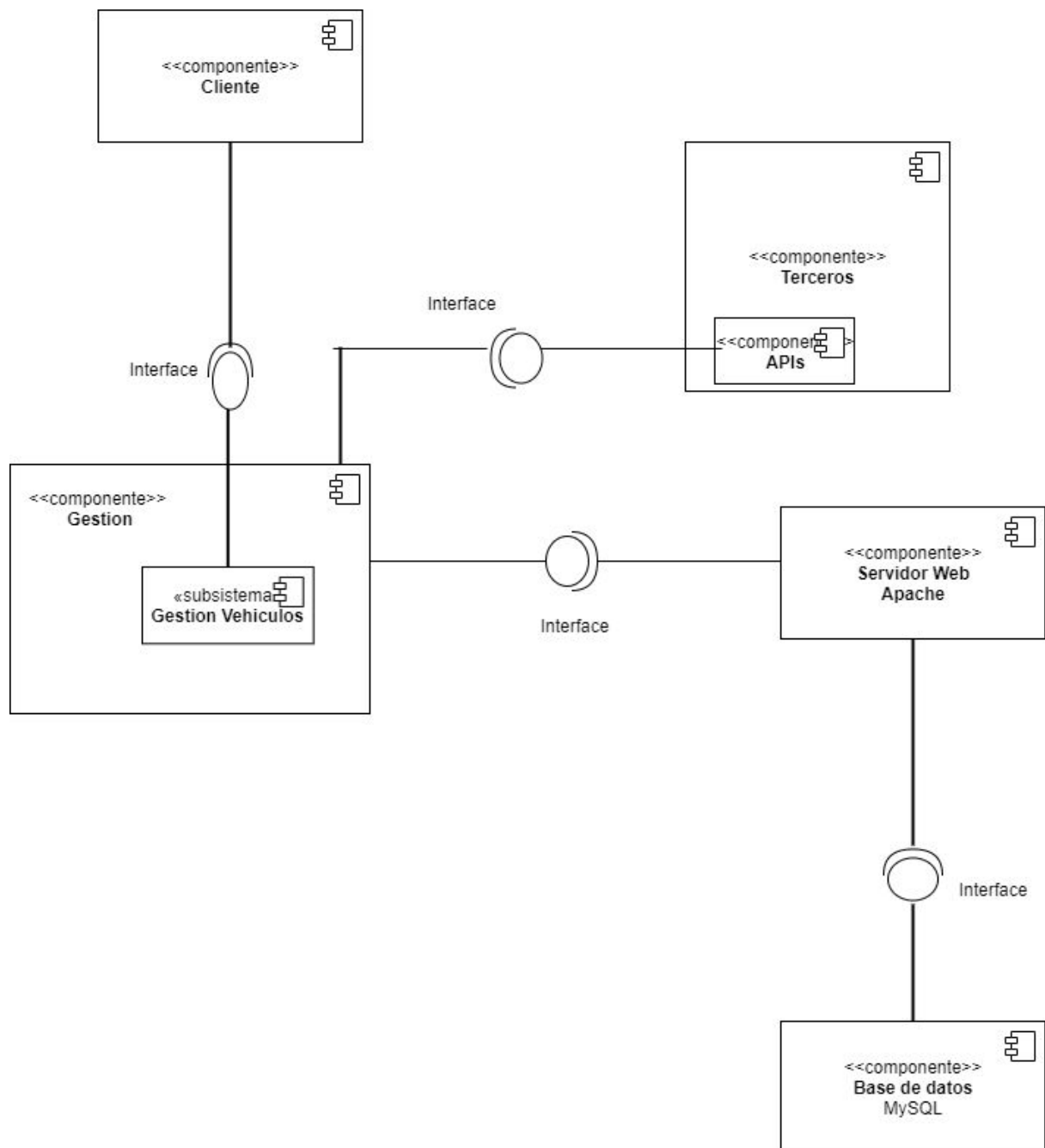
6. Diagrama de estructura compuesta(estructura interna de la clase y las colaboraciones)



7. Diagrama de paquetes



8. Diagrama de componentes



9. Escenarios básicos para cubrir los atributos de calidad:

*Se ingresará por una pantalla de Login, en la cual se autenticará con su usuario y contraseña, para luego ser dirigido a la nueva pantalla en donde se encontrarán todos sus vehículos registrados, junto con un mapa, en el cual se mostrará la ubicación en tiempo real de cada uno de estos.

Al hacer *hover* con el mouse en uno de los vehículos señalados en el mapa, se desplegará un cuadro en el que se podrá apreciar toda la información relacionada y pertinente a este, como la ruta que lleva, estado del vehículo, nivel de gasolina, e información correspondiente tanto del vehículo como del conductor.

Experiencia de usuario(UX): El cliente se autentica, y rápidamente puede acceder a la información de sus vehículos ya que el sistema permite un fácil e intuitivo manejo del mismo.

Escalabilidad: El sistema necesita poder abarcar la cantidad de vehículos en días y horas especiales en las cuales la afluencia de estos, es mucho mayor a la normal.

Rendimiento: Actualización rápida y en tiempo real de la ubicación de los vehículos solicitados.

Seguridad: El cliente necesita que la información de sus vehículos sea protegida ya que el saber la posición de estos puede generar riesgos potenciales.

Eficiencia: Cada vez que el cliente requiera utilizar nuestro sistema en alguna de las funcionalidades con las que va a contar, como por ejemplo, en conocer la información exacta de un vehículo, la podrá realizar sin problema alguno.

Soporte: Cuando al cliente se le presente un inconveniente con la aplicación, como por ejemplo que no se muestre la ubicación de algún vehículo, le llevaremos un soporte inmediato para resolver estos problemas.

Documentación: Si el cliente no sabe como llegar a la pestaña en donde se le mostrarán todos sus vehículos, podrá realizar una búsqueda en nuestro manual de usuario, en el que siguiendo los pasos que estarán ahí, podrá solucionar su inconveniente.

Compatibilidad: Si el cliente utiliza como navegador "Internet Explorer", podrá utilizar nuestra aplicación sin ningún problema de compatibilidad.

Portabilidad: Si el cliente necesita utilizar nuestra aplicación desde un dispositivo móvil con un sistema operativo IOS, lo podrá hacer sin ningún problema.

10. StakeHolders y sus funciones

- **Alejandro Muñoz: Tester**

La función principal del planificador para este proyecto será el de llevar a cabo pruebas del software resultante para comprobar si funciona correctamente. Identificará el riesgo que pueda tener el software para presentar errores y lo comunicará a el equipo de trabajo .

- **Manuel Calle: Analista**

El analista será el encargado del desarrollo de aplicaciones en lo que respecta a su diseño y obtención de los algoritmos, así como de analizar las posibles utilidades y modificaciones necesarias de los diferentes sistemas para una mayor eficacia del software. Otra misión del analista para este proyecto es dar apoyo técnico a los usuarios

- **Santiago López: Desarrollador**

es un especialista en informática que es capaz de concebir y elaborar sistemas informáticos (paquetes de software), así como de implementarlos y ponerlos a punto, utilizando uno o varios lenguajes de programación.

El desarrollador puede contribuir a la visión general del proyecto más a nivel de aplicación que a nivel de componentes, así como en las tareas de programación individuales.

Conforme pasa el tiempo, la separación entre el diseño de sistemas informáticos, el desarrollo de software, y la programación, se van haciendo más claras y diferenciadas. En el mercado laboral suele encontrarse una diferenciación entre programadores y desarrolladores, siendo estos últimos los que diseñan la estructura o jerarquía de clases. Incluso esos desarrolladores se convierten en arquitectos de sistemas informáticos, o sea, aquellos que diseñan la arquitectura a varios niveles o las interacciones entre componentes de un proyecto de software grande.

Diagrama de despliegue

