**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA INFORMÁTICA**

**Diseño e Implementación de un Sistema de Tracking y Solicitud de Pedidos para Plantas Distribuidoras de Combustible**

TRABAJO INSTRUMENTAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

**INGENIERO EN INFORMÁTICA**

**REALIZADO POR**

**DANIEL MONTIEL**

**TUTOR EMPRESARIAL**

**FRANCISCO GAMALLO**

**TUTOR ACADÉMICO**

**GLORIA TARRÍO**

**FECHA**

**JUNIO 2017**

# Dedicatorias

**A mis padres, Diego y Teresa**.

Por haberme apoyado durante toda mi carrera universitaria, por los ejemplos de ser una persona responsable y constante, por los ejemplos de ser un buen profesional, por haberme dado todas las herramientas que necesite en el transcurso de mi carrera universitaria y en especial a mi madre por prepararme buenos almuerzos durante 6 años.

Este trabajo es posible gracias a ellos.

**A mi tía y abuelo, Sonia y Giuseppe.**

Por todo su amor y cariño transmitido a lo largo de mi carrera universitaria.

**A mi hermano, Andrés**.

Por ser el ejemplo de un buen estudiante.

**A Dios.**

Por haberme dado salud para lograr mis objetivos propuestos.

**Contenido**

[Dedicatorias I](#_Toc484205856)

[Índice de Figuras y Tablas V](#_Toc484205857)

[Sinopsis VII](#_Toc484205858)

[Capítulo I – Presentación 1](#_Toc484205859)

[Necesidades de la empresa 1](#_Toc484205860)

[Solución propuesta 3](#_Toc484205861)

[Objetivo general 5](#_Toc484205862)

[Objetivos específicos 5](#_Toc484205863)

[Aporte tecnológico 6](#_Toc484205864)

[Aporte funcional 6](#_Toc484205865)

[Alcance 6](#_Toc484205866)

[Limitaciones 13](#_Toc484205867)

[Justificación 14](#_Toc484205868)

[Capítulo II – Marco Teórico 15](#_Toc484205869)

[1. Sistema de Tracking 15](#_Toc484205870)

[2. Protocolo RFID 15](#_Toc484205871)

[3. Raspberry PI 15](#_Toc484205872)

[4. Servidor de aplicaciones WildFly 17](#_Toc484205873)

[5. Patrones de diseño 17](#_Toc484205874)

[5.1 Comando 17](#_Toc484205875)

[5.2 Dao 18](#_Toc484205876)

[5.3 Fabrica 18](#_Toc484205877)

[6. Arquitectura MVVM 18](#_Toc484205878)

[7. Arquitectura RESTFul 19](#_Toc484205879)

[7.1 Json 19](#_Toc484205880)

[8. Aplicaciones híbridas 20](#_Toc484205881)

[9. Aplicaciones nativas 20](#_Toc484205882)

[10. Java 21](#_Toc484205883)

[11. Maven 21](#_Toc484205884)

[12. Git 21](#_Toc484205885)

[13. Autenticación por Token 22](#_Toc484205886)

[Capítulo III – Marco de trabajo 23](#_Toc484205887)

[1. Eventos 23](#_Toc484205888)

[1.1 Requisitos 24](#_Toc484205889)

[1.2 Actividades 24](#_Toc484205890)

[1.3 Sprint o Ciclo 24](#_Toc484205891)

[1.4 Planeación de la iteración 25](#_Toc484205892)

[1.5 Reunión diaria 25](#_Toc484205893)

[2. Roles 25](#_Toc484205894)

[2.1 SCRUM team 25](#_Toc484205895)

[2.2 Product Owner 26](#_Toc484205896)

[3.3 SCRUM Master 26](#_Toc484205897)

[3. Software rodando 26](#_Toc484205898)

[4. Justificación de la metodología. 28](#_Toc484205899)

[Capítulo IV – Desarrollo 29](#_Toc484205900)

[1. Sprint #1 29](#_Toc484205901)

[1.1 Actividades 29](#_Toc484205902)

[1.2 Desarrollo 29](#_Toc484205903)

[1.3 Software rodando 33](#_Toc484205904)

[2. Sprint #2 33](#_Toc484205905)

[2.1 Actividades 33](#_Toc484205906)

[2.2 Desarrollo 34](#_Toc484205907)

[2.3 Software rodando 38](#_Toc484205908)

[3. Sprint #3 38](#_Toc484205909)

[3.1 Actividades 38](#_Toc484205910)

[3.2 Desarrollo 38](#_Toc484205911)

[3.3 Software rodando 41](#_Toc484205912)

[4. Sprint #4 41](#_Toc484205913)

[4.1 Actividades 42](#_Toc484205914)

[4.2 Desarrollo 42](#_Toc484205915)

[4.3 Software rodando 43](#_Toc484205916)

[5. Sprint #5 43](#_Toc484205917)

[5.1 Actividades 43](#_Toc484205918)

[5.2 Desarrollo 44](#_Toc484205919)

[5.3 Software rodando 44](#_Toc484205920)

[6. Sprint #6 44](#_Toc484205921)

[6.1 Actividades 45](#_Toc484205922)

[6.2 Desarrollo 45](#_Toc484205923)

[6.3 Software rodando 47](#_Toc484205924)

[7. Sprint #7 47](#_Toc484205925)

[7.1 Actividades 47](#_Toc484205926)

[7.2 Desarrollo 47](#_Toc484205927)

[7.3 Software rodando 48](#_Toc484205928)

[8. Sprint #8 49](#_Toc484205929)

[8.1 Actividades 49](#_Toc484205930)

[8.2 Desarrollo 49](#_Toc484205931)

[8.3 Software rodando 50](#_Toc484205932)

[9. Sprint #9 51](#_Toc484205933)

[9.1 Actividades 51](#_Toc484205934)

[9.2 Desarrollo 51](#_Toc484205935)

[9.3 Software rodando 58](#_Toc484205936)

[Capítulo V – Resultados 59](#_Toc484205937)

[Capítulo VI – Conclusiones y recomendaciones 80](#_Toc484205938)

[Conclusiones 80](#_Toc484205939)

[Recomendaciones 81](#_Toc484205940)

[Referencias bibliográficas 82](#_Toc484205941)

[Apéndices 86](#_Toc484205942)

[Apéndice A - Arquitectura del sistema V-track en acoplamiento con la plataforma V-TAS 86](#_Toc484205943)

[Apéndice B – Fases del proceso de llenado 86](#_Toc484205944)

[Apéndice C – Diagramas de actividades del proceso de solicitud de pedido 89](#_Toc484205945)

[Apéndice D – Modelo Entidad-Relación 91](#_Toc484205946)

[Apéndice E – Evaluación de Framework de aplicaciones híbridas 93](#_Toc484205947)

[Apéndice F – Historias de usuario 102](#_Toc484205948)

# Índice de Figuras y Tablas

[Ilustración 1: Metodología SCRUM 24](#_Toc484205949)

[Ilustración 2: Especificación de objetivos y aportes desarrollados en cada sprint. 28](#_Toc484205950)

[Ilustración 3: Arquitectura API Restful 36](#_Toc484205951)

[Ilustración 4 : Proceso de autenticación 41](#_Toc484205952)

[Ilustración 5: Diseño de módulo de sincronización 46](#_Toc484205953)

[Ilustración 6: Proceso de autoregistro de usuario 49](#_Toc484205954)

[Ilustración 7: Interacción con FCM 51](#_Toc484205955)

[Ilustración 8: Diseño del módulo de captura de estados 55](#_Toc484205956)

[Ilustración 9: Log del módulo de captura de estado 58](#_Toc484205957)

[Ilustración 10: Arquitectura de API RESTful en implementación 61](#_Toc484205958)

[Ilustración 11: Pantalla login Aplicación Web 62](#_Toc484205959)

[Ilustración 12: Pantalla login Aplicación Web – Zoom 63](#_Toc484205960)

[Ilustración 13: Pantalla login Aplicación Móvil – Zoom 64](#_Toc484205961)

[Ilustración 14: Pantalla Pedidos Aplicación web 65](#_Toc484205962)

[Ilustración 15: Pantalla Pedidos Aplicación web - Zoom 1 65](#_Toc484205963)

[Ilustración 16: Pantalla Pedidos Aplicación web - Zoom 2. 66](#_Toc484205964)

[Ilustración 17: Pantalla Detallado pedido Aplicación web. 66](#_Toc484205965)

[Ilustración 18: Pantalla Detallado pedido Aplicación web - Zoom 1. 67](#_Toc484205966)

[Ilustración 19: Pantalla Detallado pedido Aplicación web - Zoom 2. 67](#_Toc484205967)

[Ilustración 20: Pantalla Pedidos Aplicación móvil 68](#_Toc484205968)

[Ilustración 21: Pantalla Detallado pedido Aplicación móvil 69](#_Toc484205969)

[Ilustración 22: Pantalla Solicitar pedido - Confirmar pedido - Aplicación web 70](#_Toc484205970)

[Ilustración 23: Pantalla solicitar pedido - Aplicación móvil 70](#_Toc484205971)

[Ilustración 24: Pantalla generar pedido - Aplicación móvil 71](#_Toc484205972)

[Ilustración 25: Servicios Windows de módulo de sincronización 72](#_Toc484205973)

[Ilustración 26: WSDL WSActualizador 1 72](#_Toc484205974)

[Ilustración 27: WSDL WSActualizador 2 73](#_Toc484205975)

[Ilustración 28: Servicios Windows de módulo de notificación 74](#_Toc484205976)

[Ilustración 29: Correo recibido del módulo de notificación 75](#_Toc484205977)

[Ilustración 30: Push recibido del módulo de notificación 76](#_Toc484205978)

[Ilustración 31: Máquina virtual de Windows Server 2012 77](#_Toc484205979)

[Ilustración 32: Despliegues en WildFly 77](#_Toc484205980)

[Ilustración 33: Servicios Windows de V-Track en ejecución 78](#_Toc484205981)

[Ilustración 34: Modulo de captura de estado 79](#_Toc484205982)

[Tabla 1: Implementaciones realizadas en la API de V-Track 37](#_Toc485072473)

[Tabla 2:Características de modelos de Raspberry PI evaluados 52](#_Toc485072474)

[Tabla 3: Implementaciones hechas en módulo de captura de estados 55](#_Toc485072475)

[Tabla 4: Escala de calificación para la evaluación del uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil. 95](#_Toc485072476)

[Tabla 5: Tabla de ponderación para la evaluación del el uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil. 100](#_Toc485072477)

[Tabla 6:Ponderación final de evaluación del uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil. 101](#_Toc485072478)

# Sinopsis

El siguiente trabajo de grado titulado **“Diseño e implementación de un sistema de tracking y solicitud de pedidos para plantas distribuidoras de combustibles”** fue desarrollado con la finalidad de brindarle a los clientes de las plantas distribuidoras de combustible el apoyo de saber el estado de sus pedidos en tiempo real, ofreciendo una mejor supervisión y control de su flota de transporte, y por consiguiente una optimización en los procesos de suministro de combustibles a sus estaciones de servicios.

La creación de módulos de comunicación con el Sistema de Control de Terminal con el objetivo de actualizar los estatus de los pedidos, la creación de procesos con la funcionalidad de informar y reportar a los clientes el estado de sus pedidos, son algunos de los objetivos planteados para este proyecto.

Se trabajó bajo las normativas de la metodología ágil SCRUM, ésta se escogió ya que permite comprobar de manera regular si se están cumpliendo con los objetivos planteados según el criterio del líder de proyecto, desde el principio se pueden tomar decisiones informadas a partir de resultados y dirige estos resultados del proyecto hacia los objetivos planteados.

# Capítulo I – Presentación

## Necesidades de la empresa

Vanesoft empresa integradora en el Diseño, Desarrollo, Implementación y soporte de Software para la Automatización Terminal (TAS), ha creado V-TAS, una solución para la automatización de todos los procesos operacionales en Terminales de Almacenamiento y Despacho de Combustibles.

  La implementación de V-TAS como herramienta de trabajo facilita al negocio un óptimo control de las operaciones de la terminal de manera segura y confiable, soportando a la gerencia en la toma de decisiones. Los sistemas modernos de Automatización de Terminales o Plantas de Distribución de Combustibles, deben adaptarse a las cambiantes necesidades del mercado. Si bien comparten muchos de los mismos factores de prácticas operacionales e infraestructura, también presentan problemas que le son comunes a muchos de ellos.

Los clientes de las Plantas de Distribución de Combustibles deben enviar sus transportes para retirar las cargas asociadas a sus respectivos pedidos. En la actualidad una vez que el cliente envía un transporte a la Planta de Distribución de Combustibles a retirar la carga asociada a uno de sus pedidos, no tiene información a ciencia cierta sobre la hora de llegada a la Planta de Distribución de Combustible, Tiempos en cada uno de los procesos de carga, así como tampoco hora precisa de salida de la Terminal, información que cada día se hace más necesaria al cliente para la planificación y toma de decisiones en sus procesos de procura que le permitan solucionar o evitar problemas de suministro de los diferentes combustibles en sus Estaciones de Servicios.

El no contar con una herramienta de software automatizada que le suministre en tiempo real a los clientes de la Terminal, toda la información referente a los procesos de carga de sus pedidos, le origina la pérdida parcial de la supervisión y control de su flota de transporte y por consiguiente no lograr una óptima planificación en los procesos de procura y suministro de combustibles a sus Estaciones de Servicios, lo que dificulta garantizar y ofrecer cada uno de los diferentes combustibles en sus estaciones, afectando directamente en la rentabilidad y misión del negocio.

Actualmente la gerencia de Vanesoft ha recibido constantemente una serie de críticas del V-TAS de parte los clientes de las Plantas de Distribución de Combustibles, las cuales tratan de que desconocen el estado de sus pedidos y que no tienen un proceso de solicitud de pedidos que no sea de manera presencial en las Plantas de Distribución de Combustible.

Vanesoft desea mejorar el servicio prestado a sus clientes, por lo tanto, se plantea la creación de un sistema de tracking y solicitud de pedidos el cual trabaje en conjunto con el sistema V-TAS. Este sistema permitirá mejorar la comunicación y atención hacia los clientes de la Terminal, mediante un portal web y un aplicativo móvil. Vanesoft es una empresa venezolana que tiene como objetivo el desarrollo de soluciones informáticas.

## Solución propuesta

“V-Track: Sistema de Tracking y solicitud de pedidos” será desarrollado por la empresa Vanesoft, se contará con un líder de proyecto el cual tendrá como responsabilidad revisar cuidadosamente de que los requerimientos se estén desarrollando según se desee.

La arquitectura del sistema V-Track estará compuesta por cinco capas: 1) Portal web 2) Capa con la responsabilidad de gestionar datos (base de datos) 3) Servicios web 4) Aplicación móvil 5) Captador de estado:

* Portal web: los clientes de la terminal podrán acceder al portal web mediante la autenticación de sus credenciales. Podrán solicitar pedidos y consultar el estado de sus pedidos en tiempo real. Se propone para el desarrollo de esta capa el uso de un “framework” basado en *MVVM*.[[1]](#footnote-1)
* Capa con la responsabilidad de gestionar datos (base de datos): esta capa tendrá como funcionalidad principal realizar las operaciones de crear, eliminar, consultar y modificar registros de la base de datos. Es importante mencionar que esta base de datos será centralizada.
* Capa de servicios web: esta capa tendrá como funcionalidad principal comunicar los datos principales de la aplicación móvil y el portal web. Es importante mencionar que esta capa se trata de una interfaz de programación de la aplicación (API) bajo la arquitectura *RESTful*.[[2]](#footnote-2)
* Aplicación Móvil: Los clientes de la terminal podrán visualizar toda la información de sus pedidos en tiempo real y generar pedidos a través de un Smartphone. Para esta capa se propone el uso de un framework para el desarrollo de aplicaciones híbridas, con el objetivo de generar una aplicación móvil compatible con los dos sistemas operativos de Smartphone más demandados actualmente (Android y iOS).
* Captador de estado: capa que tendrá como funcionalidad principal llevar el estado del tanque camión, barcaza o tren registrado por las *tarjetas RFID* [[3]](#footnote-3)hacia la base de datos de V-Track. Es importante mencionar que en esta capa se utilizará como componente un Raspberry PI.

## Objetivo general

Diseñar e implementar un sistema de solicitud de pedidos y tracking para el monitoreo en tiempo de real de los pedidos basado en una aplicación móvil y un portal web

## Objetivos específicos

1. Diseñar e implementar el modelo de base de datos para el sistema.
2. Diseñar e implementar una API RESTful que suministre datos a los componentes del sistema.
3. Diseñar e implementar módulo de autoregistro para el portal web.
4. Diseñar e implementar módulo de autenticación y recuperación de usuarios para el portal web y la aplicación móvil.
5. Diseñar e implementar módulo de reportes de pedidos para el portal web y la aplicación móvil.
6. Diseñar e implementar módulo de solicitud de pedidos para el portal web y aplicación móvil.
7. Diseñar e implementar módulo de captura de estados de pedidos.
8. Diseñar e implementar módulo de sincronización de datos entre el sistema V-TAS y V-Track.
9. Diseñar e implementar módulo de notificación de envió y recibo de pedidos.
10. Implementar un ambiente de despliegue de V-Track en los servidores de Vanesoft.

### Aporte tecnológico

* Evaluar distintos modelos de Raspberry PI como componente en los sistemas de tracking.
* Evaluar el uso de “*frameworks*[[4]](#footnote-4)” de aplicaciones híbridas para la implementación del aplicativo móvil.

### Aporte funcional

* Rediseñar el proceso de solicitud de pedidos utilizado por las Plantas de Distribución de Combustibles a través de una aplicación móvil y un portal web.
* Diseñar la modalidad de realizar consultas de los pedidos hechas por los clientes de las Plantas de Distribución de Combustibles.

## Alcance

A continuación se presenta el alcance de cada uno de los objetivos planteados anteriormente:

**1. Diseñar e implementar el modelo de base de datos para el sistema.**

Para este objetivo se diseñará e implementará un modelo de entidad-relación de base de datos, el cual será utilizado del lado del servidor y será una fuente de datos tanto para el portal web como la aplicación móvil.

**2. Diseñar e implementar una API RESTful que suministre datos a los componentes del sistema.**

Esta API RESTful representará la capa de servicios web y permitirá que los componentes del sistema intercambien información a través de peticiones HTTP y el formato utilizado para el intercambio de datos será JSON (JavaScript Object Notation).

Es importante mencionar que se utilizarán los principios de seguridad para preservar y proteger la información sensible.

Para efectos de este trabajo se evaluará los componentes con que cuenta el lenguaje de programación Java para el desarrollo de esta API RESTful.

**3. Diseñar e implementar módulo de autoregistro para el portal web.**

Para este objetivo se propone el diseño e implementación de un módulo que permitirá realizar el registro de nuevos usuarios en el sistema V-Track. Se tendrá que validar la información suministrada por el usuario nuevo con la información almacenada en la base de datos de clientes del sistema V-TAS.

Para este módulo se utilizará él envio de clave provisional por correo electrónico, formularios amigables y se tomará en cuenta que la creación del usuario sólo será necesaria si no posee una cuenta ya creada.

**4. Diseñar e implementar módulo de autenticación y recuperación de usuarios para el portal web y la aplicación móvil.**

Para el sistema de tracking es necesario un módulo de autenticación de los usuarios permitiendo limitar las acciones que podrán realizar los clientes de Terminal en el sistema. Los clientes podrán ingresar al sistema mediante un usuario y una clave. En la autenticación es necesario la autenticación del dispositivo para que no sea robada la sesión del usuario, se propone el uso de envio de Token en la cabecera de la petición HTTP.

Dentro de este módulo se contemplan los procesos necesarios para la recuperación de usuario y clave mediante confirmación del usuario por envio de correo.

**5. Diseñar e implementar módulo de reportes de pedidos para el portal web y la aplicación móvil.**

Se plantea un módulo de reportes de pedido en el cual el usuario pueda ingresar mediante por el portal web o la aplicación móvil y visualizar el estado de todos sus pedidos en tiempo real.

Este módulo suministrará información necesaria para el cliente como hora de llegada del camión, barcaza o tren, en qué fase de llenado se encuentra, hora estimada de salida de la refinería, hora de salida de la refinería. Este módulo será diseñado e implementado con interfaces amigables.

**6. Diseñar e implementar módulo de solicitud pedidos para el portal web y aplicación móvil.**

Se diseñará e implementará un módulo de solicitud de pedidos, el cual incluya procesos en el cual el usuario pueda solicitar pedidos mediante el portal web o la aplicación móvil. A continuación se describen los procesos:

• El usuario podrá describir con exactitud el pedido que desea realizar mediante formularios amigables.

• Enviar la solicitud al sistema V-Track.

• V-Track generará una clave vinculada al pedido y la enviará al correo registrado del cliente.

• El usuario deberá ingresar la clave vinculada al pedido mediante el portal web o la aplicación móvil para poder liberar el pedido y la Planta de Distribución de Combustibles realice los procedimientos correspondientes.

**7. Diseñar e implementar módulo de captura de estados de pedidos.**

Actualmente el sistema V-TAS no cuenta con procesos para la captura de estados en el flujo de llenado de tanque camión, barcaza o tren. Para efectos de este proyecto es necesario diseñar e implementar un módulo de captura de estados de los pedidos, de manera que el estado del pedido pueda ser almacenado a nivel de base de datos y la capa de Web Servicies consuma esta información anteriormente descrita.

Se propone diseñar e implementar un módulo con el siguiente flujo:

1. A partir de un Raspberry PI (se realizará una evaluación para seleccionar el modelo más sofisticado) se esperarán señales de las lectoras de RFID (esto indicará el cambio de estado del pedido).

2. El Raspberry PI registrará a nivel de base de datos el nuevo estado del pedido en el sistema V-TAS.

3. Posteriormente estos registros serán actualizados en el sistema V-Track mediante el módulo de sincronización de datos entre el sistema V-TAS y V-Track.

**8. Diseñar e implementar módulo de sincronización de datos entre el sistema V-TAS y V-Track.**

Actualmente, V-TAS es el sistema que realiza el llenado de los pedidos y contiene los datos de los clientes. Por esto, se requiere un módulo que permita la automatización del proceso de extraer e importar la información de los clientes suministrada por las Plantas de Distribución de Combustibles para posteriormente almacenarla en la base de datos centralizada de V-Track. Esta información importada de los clientes será utilizada posteriormente por la capa de Web services.

**9. Diseñar e implementar módulo de notificación envió y recibo de pedidos.**

Se diseñará un módulo que cada cierto tiempo consulte a nivel de base de datos el estado de los pedidos. Cuando un pedido sea nuevo en el sistema o haya finalizado su proceso de llenado en la Planta de Distribución de Combustibles, este módulo notificará al cliente dueño del pedido sobre el estado de su pedido de dos maneras: mediante correo electrónico y recibiendo una notificación push a la aplicación móvil.

**10. Implementar un ambiente de despliegue de V-Track en los servidores de Vanesoft.**

Actualmente Vanesoft no cuenta con servidores virtuales en los cuales realizar los despliegues de la plataforma “V-Track” antes de ser definitivos, por esta razón, se propone implementar un servidor virtual de Vanesoft que funcione bajo el sistema operativo Windows 2012 server para el despliegue de “V-Track”.

**Aporte funcional:**

**Rediseñar el proceso de solicitud de pedidos utilizado por las Plantas de Distribución a través de una aplicación móvil y un portal web.**

Actualmente la comunicación entre las Plantas de Distribución de Combustibles y sus clientes suele ser presencial o por vía telefónica. El proceso actual es el siguiente:

1. El cliente llama al call center de la Planta de Distribución de Combustible en busca de una cita de solicitud de pedidos.

2. Un grupo de operadores atiende la llamada del cliente.

3. El operador consulta la disponibilidad de una cita de solicitud de pedidos y el estatus del cliente (si posee alguna deuda con la Planta de Distribución de Combustible o no).

4. El operador asigna una cita de solicitud de pedidos al cliente.

5. El operador le comunica al cliente la fecha de la cita.

6. El cliente asiste a la cita de solicitud de pedidos. Describe todas las solicitudes de que desea realizar.

7. El personal confirma la disponibilidad para satisfacer las solicitudes del cliente.

Se plantea que en el nuevo proceso, la creación de un portal web y una aplicación móvil acerque al cliente y la Planta de Distribución de Combustible de forma automática, permitiendo crear comunicación efectiva en tiempo real y que realice la gestión de solicitud de pedidos de manera más rápida que la actual.

La gestión de solicitud de pedidos se llevará a cabo a través de solicitudes HTTP y formularios para la descripción de los pedidos.

**Diseñar la modalidad de realizar consultas de los pedidos hechas por los clientes de las Plantas de Distribución de Combustible.**

Actualmente las Plantas de Distribución de Combustible que utilizan V-TAS no cuentan con procesos para que sus clientes consulten el estado de sus pedidos. Esto es una crítica que muchos clientes de terminal han realizado, ya que los mismos no pueden realizar planificaciones o mejorar su toma de decisiones. Se plantea un nuevo proceso en el cual el cliente accediendo por el portal web o la aplicación móvil pueda consultar todos sus pedidos. De estos pedidos el cliente podrá consultar el detallado de cada uno, pudiendo saber la hora de llegada, en qué fase del proceso de llenado se encuentra, la hora estimada de salida, y la hora de salida finalmente luego de que termine el proceso de llenado.

También se plantea la modalidad de que el usuario reciba notificaciones cuando el pedido del cliente ha iniciado el proceso de llenado y cuando el proceso de llenado ha sido finalizado mediante correo electrónico y notificaciones push.

**Aporte tecnológico:**

**Evaluar los distintos modelos de Raspberry PI como componente en los sistemas de tracking.**

Se realizará una investigación y evaluación de los distintos modelos de Raspberry PI existente en el mercado para ser utilizado como un componente en los sistemas de tracking. Adicionalmente se evaluará el uso de la Raspberry PI como un componente en el sistema V-Track, tomando en cuenta sus tiempos de respuestas, su adaptabilidad al sistema y su capacidad de almacenamiento.

**Evaluar el uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil.**

Se realizará una evaluación tomando en cuenta las ventajas y desventajas de los distintos frameworks que existen actualmente para la creación de aplicaciones híbridas. Esto se realiza con el fin de que la aplicación móvil de V-Track pueda funcionar en las distintas plataformas más demandadas actualmente (Android y iOS).

## Limitaciones

* Para poder utilizar el portal web y la aplicación móvil es de vital importancia que el usuario mantenga conexión a internet.
* La aplicación móvil podrá ser utilizada por usuarios con dispositivos que contengan sistemas operativos Android y iOS.
* Para poder realizar el autoregistro en el sistema V-Track es necesario que el usuario sea cliente de la Planta de Distribución de Combustible que utilice V-TAS.
* El desarrollo se realizará en una máquina con las siguientes características:

1. Procesador: Intel® Core™ i3-3250 CPU @ 3.50GHz 3.49 GHz.
2. Memoria (RAM): 8,00 GB (7,71 GB utilizable).
3. Tipo de sistema: Sistema operativo de 64 bits.
4. Edición de Windows: Windows 10.

## Justificación

Debido a la modernización de los negocios en el mundo, Vanesoft se ve en la necesidad de invertir en el mejoramiento de su producto V-TAS, para darles un mejor servicio a sus clientes, posicionarse como una buena empresa en el mercado, brindarles a sus clientes las posibilidades de ser una mejor organización y tener una mejor planificación al suministrar combustibles a sus estaciones de servicios.

Para brindar un servicio moderno y eficiente, se propone desarrollar un proyecto informático que permita a los clientes de las plantas de combustibles consultar en tiempo real el estado de sus pedidos y solicitar pedidos de combustible de manera no presencial. Previamente se deben desarrollar los procesos encargados de registrar en qué fase del procedimiento de llenado se encuentra el pedido, así como también un módulo de sincronización que permita llevar los datos registrados en el sistema V-TAS al sistema V-Track, para que sean consultados posteriormente por la aplicación web y la aplicación móvil.

# Capítulo II – Marco Teórico

## 1. Sistema de Tracking

El usuario podrá saber el estado de sus pedidos mediante un sistema de tracking. Este es un sistema que permite conocer la localización de las mercancías dentro de la empresa y mantener en línea al cliente, gracias a las nuevas tecnologías. Mediante este sistema fácil y sencillo, los clientes pueden conocer en todo momento el estado de sus pedidos. (RFLCargo, 2016)

## 2. Protocolo RFID

El módulo de captura de estado está compuesto por lectoras de tarjetas RFID, esta utiliza una tecnología de transmisión de datos automática que emplea radiofrecuencia para comunicar información entre la lectora y una tarjeta; es decir, utiliza ondas electromagnéticas para hacer llegar información a la lectora desde la tarjeta. Este sistema permite la transmisión de datos sin la necesidad de hacer contacto entre la lectora y la tarjeta. (RFID Tecnología de identificación por Radiofrecuencia y sus principales aplicaciones, 2007)

## 3. Raspberry PI

Para captar las señales de las lectoras RFID se hará uso de un dispositivo Raspberry PI, este no es más que una computadora del tamaño de una tarjeta de crédito la cual se puede conectar a un monitor y un teclado. Este es capaz de ser utilizado en proyectos de electrónica, y muchas cosas que hace un PC normal, como hojas de cálculo, navegar en internet, jugar juegos y reproducir videos de alta definición. (Raspberry, 2017)

Los componentes de este procesador son los siguientes:

Conexión eléctrica

La Raspberry Pi usa como fuente de energía una toma de micro-USB, esta es utilizada hoy en día por muchos teléfonos móviles o cámaras fotográficas. El único requisito es que esta alimentación sea de 5Vy 700 mA. (Arboix, 2013)

Memoria de almacenamiento

Dado que el tamaño de la Raspberry PI es muy reducido conlleva a la ausencia de disco duro, por lo que el almacenamiento se realiza mediante una tarjeta SD de 4 GB, 8 GB o hasta 32 GB. Si se desarrollaran proyectos que van a necesitar de más memoria podemos optar por dos soluciones: configurar distintas tarjetas SD o ajustar la Raspberry PI para que pueda aceptar tarjetas superiores a las de 32 GB (esta última opción conlleva conocimientos técnicos avanzados). (Arboix, 2013)

Internet

La Raspberry PI gestiona la conexión a internet de dos maneras distintas: la primera opción es mediante un cable de Ethernet y, la segunda, utilizando un USB-WiFi. Para ambas opciones no es necesario instalar algún driver, puesto que el propio sistema operativo de la Raspberry PI ya los incluye. (Arboix, 2013)

## 4. Servidor de aplicaciones WildFly

El servidor de aplicaciones tanto en el ambiente de desarrollo como en el ambiente de producción será WildFly. Este es un servidor de aplicaciones Java de código abierto que puede ser utilizado en cualquier sistema operativo en el que se encuentre disponible la máquina virtual de Java (Cloudbuilder Next, 2017)

Sus principales características son:

* Despliegue ágil y capacidad de editar recursos sin la necesidad de redespliegue.
* Cada servicio desplegado en el servidor puede ser iniciado y detenido aisladamente.
* Realiza la gestión de memoria de manera eficiente.
* Enfoque modular. (facilCloud, 2015)

## 5. Patrones de diseño

Los patrones de diseño son un conjunto de buenas prácticas que ayudan a prevenir los problemas frecuentes de la programación orientada a objetos, además de esto, facilita la creación de sistemas reutilizables y fáciles de mantener.

Actualmente existen muchos patrones de diseño, a continuación se mencionan y se explican los aplicados en la arquitectura del proyecto:

### 5.1 Comando

El patrón de diseño comando tiene como función principal encapsular una petición o acción dentro de un objeto, de esta manera, se permite parametrizar a los clientes con distintas peticiones, encolarlas, guardarlas en un registro o implementar un mecanismo de deshacer/repetir (Gala, 2015).

### 5.2 Dao

La función principal del patrón de diseño DAO es encapsular el acceso a la base de datos, es decir, cuando la capa lógica de negocio necesite algún recurso de la base de datos, va a hacerlo a través de métodos que le ofrece DAO.

Generalmente DAO consiste en métodos CRUD (Create, Read, Update, y Delete). Por ejemplo, cuando la capa lógica de negocio necesite almacenar un dato va a llamar a un método create().

DAO Consiste básicamente en una clase que es la que interactúa con la base de datos. (Castellanos, 2013).

### 5.3 Fábrica

La función principal del patrón de diseño fábrica es liberar al desarrollador sobre la forma correcta de crear objetos. Se define una interfaz de un cierto tipo de objeto, esta interfaz permite a sus subclases que decidan la instancia de los objetos. (Juarez, 2011)

## 6. Arquitectura MVVM

La aplicación web estará desarrollada bajo la arquitectura MVVM, MVVM es un patrón de diseño de arquitectura de software, esta arquitectura separa la interfaz de usuario de la lógica de negocio y el modelo de datos de la aplicación. Este patrón de arquitectura es frecuente en las tecnologías .NET como C#, ya que aquí es donde se inició originalmente. (Bennett, 2017)

## 7. Arquitectura RESTFul

En el lado del servidor se creará un servicio RESTFul, este servicio lo utilizará tanto la aplicación web como la aplicación móvil. Los servicios web RESTFul son diseñados e implementados para funcionar mejor en la web. Representational State Transfer (REST) es un estilo de arquitectura que especifica restricciones, como la interfaz uniforme. Si estas restricciones son aplicadas a un servicio web se obtendrán propiedades deseables, como rendimiento, escalabilidad y modificabilidad, estas propiedades permiten que los servicios funcionen mejor en la web. En este estilo arquitectónico, los datos y la funcionalidad se consideran recursos y se accede mediante URI (Uniform Resource Identifiers).

Los recursos se utilizan con cuatro operaciones fijas de creación, lectura, actualización y eliminación PUT, GET, POST y DELETE. PUT crea un recurso en el sistema, que puede eliminarse mediante DELETE. GET consulta el estado del recurso y POST modifica el recurso utilizado. (Oracle, 2013).

Para el intercambio de datos entre la arquitectura RESTful, la aplicación web y la aplicación móvil, se decidió utilizar el formato Json, a continuación se describe

### 7.1 Json

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato de intercambio de información. Leer y escribir este formato es simple para los humanos, mientras que para las computadoras es simple la interpretación y la generación. Está basado en un subconjunto del lenguaje de programación JavaScript Standard ECMA-262 3rd Edition - Diciembre 1999. Este formato de texto utiliza convenciones que son ampliamente conocidos por los programadores de C, C++, C#, Java, JavaScript, Perl, Python, y otros más. (JSON, 2017)

## 8. Aplicaciones híbridas

Para el aplicativo móvil del sistema V-Track se ha propuesto el desarrollo de una aplicación híbrida. Las aplicaciones híbridas son desarrolladas usando un único stack de lenguajes de programación y son empaquetadas para ser instaladas en múltiples sistemas operativos de Smartphone, con diferentes tamaños de pantalla y fabricantes. Estas aplicaciones permiten el desarrollo en tecnologías simples como HTML, CSS y JavaScript. (Urrego, 2016)

## 9. Aplicaciones nativas

Las aplicaciones nativas son llamadas así porque se desarrollan en el lenguaje nativo del propio dispositivo móvil. Dependiendo de la plataforma para la que queramos crear nuestra aplicación móvil, desarrollaremos en un lenguaje en específico. Para desarrollar en Android se programa en Java, en iOS Objetive C, y ahora también el nuevo Swift, en Windows Phone se usa C# y Visual Basic .NET.

Estas aplicaciones tienen acceso a diferentes recursos como la cámara, el GPS, entre otros. Además de esto, una gran ventaja es que, al estar diseñada para el software del dispositivo móvil, tendrán un rendimiento optimizado y una interfaz gráfica más adaptada al sistema operativo al cual el usuario está acostumbrado a utilizar. La distribución de estas aplicaciones se hace a través de las tiendas virtuales oficiales de cada sistema operativo. (QODE, 2014)

## 10. Java

Para el desarrollo de módulos en el servidor se desarrollará bajo el lenguaje de programación Java. Java es un lenguaje de programación con el que se puede crear cualquier tipo de programa. En la actualidad es un lenguaje muy extenso y cada vez cobra más importancia tanto en el desarrollo web como en la informática en general. Fue desarrollado por la compañía Sun Microsystems con mucha dedicación y enfocado a cubrir las necesidades tecnológicas más punteras. (Alvarez, Qué es Java, 2001)

## 11. Maven

Para lograr la modularidad en este proyecto se hizo uso de la herramienta “Maven”. Maven fue utilizado como gestor de dependencias entre los módulos de V-Track.

Maven es una popular herramienta de creación de proyectos empresariales de Java, diseñada para sacar mucho partido del proceso de construcción. Maven utiliza un enfoque declarativo, donde se describe la arquitectura del proyecto. Esto ayuda a cumplir los estándares de desarrollo de toda la compañía y reduce el tiempo necesario para escribir y mantener scripts de compilación. (Smart, 2005)

## 12. Git

Para llevar un control de versiones en este proyecto se hizo uso de la herramienta “Git”.

Git es en la actualidad la implementación más popular de un sistema de control de versión distribuido. Git se origina en el desarrollo del kernel de Linux y fue creado en el 2005. Hoy en día es utilizado en miles de proyectos de código abierto, por ejemplo, desarrolladores de Android o Eclipse. (Lars Vogel (c), 2015)

## 13. Autenticación por Token

Como medida de seguridad en la aplicación web y en la aplicación móvil se utilizará la autenticación por Token. Este mecanismo trabaja de la siguiente manera. El usuario se autentica con un usuario /contraseña, o través de un proveedor de identificación como lo son Twitter, Facebook o Google por ejemplo. Luego de ser autenticado, cada petición HTTP que haga el usuario estará acompañada de un Token en la cabecera. Este Token es una firma cifrada que permite al API de nuestro sistema identificar al usuario. Este Token no es almacenado en el servidor, si no en el lado del cliente y el API se encarga de descifrar ese Token y redirigir el flujo de la aplicación.

Dado que los Tokens son almacenados en el lado del cliente, no hay información de estado y la aplicación se vuelve bastante escalable. El API desarrollado se puede utilizar para diferentes tipos de aplicación (Web, Mobile, Android, iOS,…), como desarrolladores debemos preocuparnos únicamente de enviar los datos en formato Json, generar y descifrar Tokens en la autenticación para posteriormente validarlos en las peticiones HTTP. (Azaustre, 2015).

# Capítulo III – Marco metodológico.

En este proyecto se utilizó la metodología SCRUM, con sus respectivas modificaciones en las etapas de desarrollo, las cuales optimizaron la adaptación del programador al marco de trabajo de la compañía Vanesoft. Permitiéndole así cumplir con los objetivos descritos anteriormente en este proyecto.

El marco de trabajo que se llevó a cabo en la metodología SCRUM se basó en dos componentes, eventos y roles. Cada uno de estos componentes dentro de la metodología sirvió con un propósito específico y fueron esencial para el éxito del proyecto. En la Ilustración 1 puede visualizarse el esquema de SCRUM.

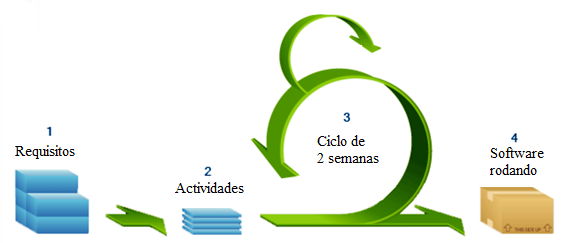


Ilustración 1: Metodología SCRUM

*Fuente: Elaboración propia.*

## 1. Eventos

Los eventos dentro del marco de trabajo de Scrum se usan para reducir la necesidad de reuniones no planeadas. Todos estos eventos deben tener una duración definida, de forma tal que el evento no se extienda más allá de la duración definida. (ids, 2014).

A continuación se explica cada uno de los eventos de la metodología utilizada en este proyecto:

### 1.1 Requisitos

Se analizaron cada uno de los requerimientos planteados a desarrollar en el proyecto.

### 1.2 Actividades

Consistió en el establecimiento de todos los requerimientos que serían desarrollados en el ciclo o sprint. Cada ciclo tuvo una duración de dos semanas, para así realizarse 9 interacciones con la intención de abarcar un tiempo de 18 semanas de desarrollo.

### 1.3 Sprint o Ciclo

En este período de tiempo se desarrollaron las actividades que se establecieron en el paso anterior (Actividades), para efectos de este proyecto se decidió que estos ciclos de trabajo tuvieran una duración de 2 semanas. Al final de cada uno de estos ciclos se entregaron los requerimientos desarrollados durante el sprint para que fueran corregidos y se verificará la calidad del producto que estaba siendo desarrollado.

### 1.4 Planeación de la iteración

Esta fase de la metodología representa reuniones que se llevaron a cabo cada dos semanas con la finalidad de organizar las actividades en curso y las actividades que quedaban, para así ir organizando el siguiente sprint o iteración.

### 1.5 Reunión diaria

Se utilizaron reuniones presenciales cortas. Esto se utilizó para llevar a cabo la pizarra de requerimientos y esta se basa en tres columnas principales para el histórico de los requerimientos, las cuales son tareas por hacer, en proceso y finalizadas, haciendo una variación de la pizarra original de la metodología SCRUM para adaptarse al marco de trabajo de la compañía.

## 2. Roles

En el marco de trabajo de SCRUM, el equipo se concentra en construir software de calidad. La gestión de un proyecto en el marco de trabajo SCRUM se focaliza en definir cuáles son las cualidades que debe tener el producto a desarrollar (qué construir, qué no y en qué orden) y en eliminar los obstáculos que pudieran aparecer en el camino del equipo de desarrollo. (SOFTENG, 2010).

Para el marco de trabajo de SCRUM en la compañía Vanesoft, el equipo SCRUM está formado por los siguientes roles:

### 2.1 SCRUM team

Representa el equipo de trabajo el cual estuvo enfocado en el desarrollo del sistema bajo los términos de la metodología SCRUM. Para efectos de este proyecto este rol fue asumido por mi persona.

### 2.2 Product Owner

Esta persona toma las decisiones, y es la que conoce realmente el negocio del cliente y los requerimientos del producto. Esta persona también se encarga de escribir las ideas del cliente y las ordena por prioridad. (Trigas Gallego, Metodología Scrum, 2015). Este rol fue asumido por el tutor empresarial Francisco Gamallo.

### 3.3 SCRUM Master

Es responsable de asegurarse que el trabajo del equipo vaya bien, siguiendo las bases de SCRUM. Además, se encarga de remover cualquier obstáculo que pueda encontrar el equipo de desarrollo. Este rol fue asumido por el tutor empresarial Francisco Gamallo.

## 3. Software rodando

Al terminar cada ciclo de desarrollo, se realizó una muestra del subproducto desarrollado al equipo de trabajo para su respectiva evaluación y la verificación de todos los requerimientos que fueron planteados para la iteración pertinente.

A efectos de la metodología llevada por la compañía Vanesoft, este evento es llamado también “Incremento”, este según las bibliografías es definido de la siguiente manera:

“El incremento es la porción de producto desarrollada en un sprint, y la característica principal de esta porción es estar completamente terminada y operativa, en condiciones de ser entregada al cliente”. (Trigas Gallego, Metodología Scrum, 2015).

En la Ilustración 2 se definen los sprints de la metodología empleada con el detalle de los objetivos seleccionados para cada sprint:

Ilustración 2: Especificación de objetivos y aportes desarrollados en cada sprint.

*Fuente: Elaboración propia.*

## 4. Justificación de la metodología.

Dado que el sistema V-Track es un proyecto que debe ser desarrollado ágilmente y bajo la supervisión de un jefe que conozca ampliamente las necesidades del cliente, se decidió utilizar la metodología ágil SCRUM. Cada incremento debe ser evaluado y aceptado por el Product Owner del equipo de trabajo. Además diariamente se realizan reuniones donde el equipo de desarrollo puede expresar las dificultades que se están presentando y el Product Owner puede realizar críticas sobre mejoras que se pueden realizar sobre el producto. Estas reuniones diarias reducen el riesgo potencial de desviarse de los objetivos planteados en este trabajo de grado.

Por otra parte, una gran ventaja de la metodología SCRUM en el proyecto V-Track es la adaptabilidad al cambio, a pesar que al principio del desarrollo se definió un plan, existe una mejora constante del producto, dado que a medida que se realiza el desarrollo se pueden realizar modificaciones cuando se solicitan, es decir, se está empleando una metodología adaptativa.

En la metodología SCRUM se lleva una organización más sencilla del proyecto, para saber el estado del proyecto simplemente basta con acercarse a la pizarra de requerimientos y leerla. Una organización sencilla elimina niveles de gestión y acerca más al equipo de trabajo. Dado que V-Track es un producto interno de Vanesoft, no es necesario una documentación extensa, sin embargo, se hará uso de las historias de usuarios que son originarias de la metodología XP.

# Capítulo IV – Desarrollo

Para el desarrollo de este trabajo de grado se decidió que el producto debe ser adaptable a la plataforma V-TAS, esto se logró gracias a la aplicación de patrones de diseño. La planificación de cada uno de los sprints se realizó bajo lo establecido en el marco metodológico, a continuación se detalla el desarrollo de cada uno de los sprints:

## 1. Sprint #1

En este primer sprint se procedió a analizar todos los requerimientos del sistema a desarrollar, diseñar su modelo de base de datos, diseñar el nuevo modo de realizar consultas de los pedidos y rediseñar el proceso de solicitud de pedidos. Para lograr un diseño óptimo, también se realizó el diseño de la arquitectura del sistema V-Track.

### 1.1 Actividades

* Diseñar e implementar el modelo de base de datos para el sistema.
* Diseñar la modalidad de realizar consultas de los pedidos hechas por los clientes de las Plantas de Distribución de Combustibles.
* Rediseñar el proceso de solicitud de pedidos utilizado por las Plantas de Distribución de Combustibles a través de una aplicación móvil y un portal web.

### 1.2 Desarrollo

Para el diseño de la nueva modalidad de realizar consultas de los pedidos y el rediseño del proceso de solicitud de pedidos fue necesario realizar el diseño de la arquitectura del sistema V-Track en acoplamiento con la plataforma V-TAS (Ver apéndice A). Los clientes de las Plantas de Distribución de Combustibles podrán realizar consultas de sus pedidos mediante una aplicación web y una aplicación móvil, en ambas aplicaciones los clientes podrán ver tanto la información general de todos sus pedidos como la información detallada de cada uno de sus pedidos, estas dos aplicaciones realizaran peticiones HTTP al sistema V-Track, el cual responderá con la información correspondiente. V-Track tendrá las siguientes características:

* V-Track estará diseñado con el estilo arquitectónico RESTFul.
* Intercambiará información con la aplicación web y la aplicación móvil mediante la transmisión de Json.
* Para el desarrollo de V-Track se aplicará el uso de los patrones de diseño: Comando, Fábrica y DAO.

El sistema V-Track mantendrá una comunicación cada cierto tiempo con el sistema V-TAS, este tiempo será parametrizable y será registrado en un archivo de configuración del sistema. Esta comunicación se logrará con el desarrollo de un módulo de sincronización el cual tendrá las siguientes responsabilidades y características:

* El módulo de sincronización en el sistema V-TAS estará compuesto por un Web Service, el cual responderá las solicitudes hechas desde el sistema V-Track para ser actualizado.
* El módulo de sincronización en el sistema V-Track estará compuesto por tres Servicios Windows, los cuales solicitarán información actualizada al sistema V-TAS sobre los pedidos, los eventos sucedidos en el proceso de llenado de cada pedido y sobre los clientes de las plantas distribuidoras.
* Deberá realizar la sincronización de la información general de los pedidos mediante el protocolo SOAP.
* Deberá realizar la sincronización de los eventos ocurridos en el proceso de llenado de los pedidos mediante el protocolo SOAP.
* Deberá realizar la sincronización de la información de los clientes de las plantas de distribución de combustible mediante el protocolo de SOAP.
* Deberá iniciar procesos con la finalidad de notificar a los clientes cuando sus pedidos inician y terminan el proceso de llenado.
* Deberá mantener la base de datos del sistema V-Track actualizada con respecto a la información extraída de la plataforma V-TAS. La información almacenada en esta base de datos será entregada a los clientes mediante la aplicación web y la aplicación móvil.

El módulo de sincronización extraerá la información generada en el sistema V-TAS por el módulo de captura de estados y la llevará al sistema V-Track. Para el desarrollo del módulo de captura de estados se instalarán lectoras de tarjetas RFID en cada fase del proceso de llenado (Ver apéndice B) y al iniciar el proceso de llenado, se le asignará a cada pedido una tarjeta RFID, la cual el chofer deberá acercar a la lectora de tarjetas RFID de cada fase de llenado. El módulo de captura de estados tendrá las siguientes responsabilidades:

* Captar mediante las lectoras de tarjetas RFID, en qué fase del proceso de llenado se encuentra el pedido.
* Con la señal captada desde las lectoras de tarjetas RFID, registrar el cambio de la fase en la que se encuentra el pedido. Este registro se hará en la base de datos del sistema V-TAS y se realizará con el uso de un Raspberry PI. Esta información registrada, es la que extraerá el módulo de sincronización.

La otra modalidad con que los clientes tendrán conocimientos sobre sus pedidos es mediante notificaciones. El módulo de notificación tendrá las siguientes responsabilidades:

* Detectar los pedidos que se encuentran en la fase de “En espera” y en la fase de “Cargado”.
* Enviar un correo a la dirección de correo electrónico registrada del cliente dueño del pedido que se encuentra en la fase de “En espera” o en la fase de “Cargado”.
* Enviar una notificación push al cliente dueño del pedido que se encuentra en la fase de “En espera” o en la fase de “Cargado”.

Actualmente el proceso de solicitud de pedidos se realiza presencialmente en las instalaciones de las plantas distribuidoras de combustibles. En el apéndice C se muestra el diagrama de actividades del proceso antiguo y el diagrama de actividades del nuevo proceso .Se rediseñó el proceso de solicitud de pedidos y se definió el siguiente proceso:

* El usuario podrá describir con exactitud el pedido que desea realizar mediante formularios amigables en la aplicación web y la aplicación móvil.
* Enviará la solicitud al sistema V-Track.
* V-Track generará una clave vinculada al pedido y la enviará al correo registrado del cliente.
* El usuario deberá ingresar la clave vinculada al pedido mediante el portal web o la aplicación móvil para poder liberar el pedido, se enviará esta solicitud con su descripción al Web Service correspondiente del sistema V-TAS y la Planta de Distribución de Combustibles realizará los procedimientos correspondientes.

Por último, se diseñó la estructura de la base de datos a través de un modelo Entidad-Relación (Ver Apéndice D). Se eligió como sistema de gestión de base de datos SQL SERVER, en el cual se llevó a cabo la implementación de la estructura de la base de datos.

### 1.3 Software rodando

* Se implementa esquema de base de datos en SQL SERVER.
* Se define el flujo de trabajo de las consultas que podrá realizar el cliente.
* Se define el flujo de trabajo del nuevo proceso de solicitud de pedidos.

## 2. Sprint #2

En este sprint se elige el diseño e implementación del API RESTFul que suministrará información tanto a la aplicación web como a la aplicación móvil.

### 2.1 Actividades

* Diseñar e implementar una API RESTful que suministre datos a los componentes del sistema.

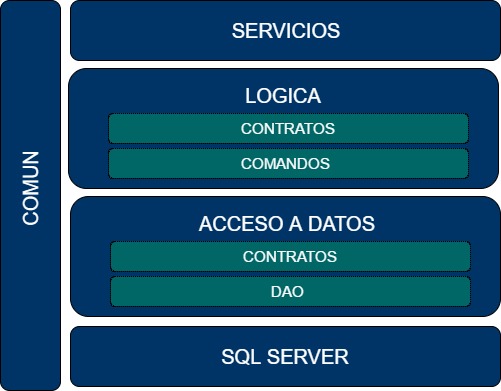
### 2.2 Desarrollo

Se definió el diseño de la arquitectura del API, se decidió que estaría divido en cuatro módulos: Módulo de Acceso a datos, Módulo de Lógica, Módulo de Servicios y Módulo Común. A continuación se describe la funcionalidad de cada uno de ellos.

* Módulo de Lógica: en este módulo se definen las reglas de negocio del sistema V-Track, se diseñó bajo el patrón de diseño “Comando”, recibe datos del módulo de servicios y posteriormente los envía al módulo de acceso a datos según las necesidades del negocio.
* Módulo de Servicios : este módulo es el que se encarga de recibir las peticiones HTTP de la aplicación web y la aplicación móvil para posteriormente ejecutar los comandos creados en el módulo de lógica.
* Módulo de Común: este módulo contiene funcionalidades que se utilizan en los otros tres módulos como lo son: entidades base, funcionalidad de encriptar, funcionalidad de desencriptar, funcionalidad de extraer propiedades del sistema descritas en archivos de configuración etc.
* Módulo de Acceso a datos: este módulo proporciona las operaciones de los datos persistentes en la base de datos. Es el único módulo que tiene acceso a la base de datos de V-Track.

En la Ilustración 3 se presenta la arquitectura API Restful.

Es importante mencionar que la dependencia entre cada uno de los módulos se logró gracias al uso de la herramienta maven. Esta herramienta también provee un gran paquete de librerías que se utilizaran en el transcurso del desarrollo del sistema V-Track.



*Ilustración 3: Arquitectura API Restful*

*Fuente: elaboración propia*

Luego de haber construido la arquitectura del API se procedió a implementar cada uno de los módulos mencionados de la manera como se indica en la Tabla 1.

En el módulo de Servicios se utilizó el API[[5]](#footnote-5) de java “JAX-RS” para implementar los servicios RESTFul. JAX-RS es una API Java utilizada para el desarrollo de aplicaciones REST de forma rápida. JAX-RS proporciona una manera más rápida de desarrollo de aplicaciones web que los servlets, el objetivo principal de JAX-RS es crear servicios RESTFul (IBM, 2015).

|  |  |
| --- | --- |
| **Módulo** | **Implementaciones** |
| Módulo Común | 1. Se crearon las excepciones a utilizarse en cada módulo. 2. Se crearon las entidades bases a utilizarse en todo el sistema. 3. Se crearon utilidades a usarse en los otros módulos (armador de notificaciones, encriptador, desencriptador y Archivos de configuración de cada uno de los módulos). |
| Módulo de Acceso a datos | 1. Se implementaron los DAO de cada una de las entidades base del sistema. 2. Se implementaron los contratos de los DAO creados anteriormente. |
| Módulo de Lógica | 1. Se implementaron los comandos necesarios para cumplir con los requerimientos de la aplicación web y la aplicación móvil. 2. Se implementaron los contratos necesarios para la creación de los comandos creados anteriormente. |
| Módulo de Servicios | 1. Se crearon las 4 operaciones (GET, PUT, DELETE Y POST) de las entidades bases creadas anteriormente. Estas operaciones pueden ser accedidas mediante URI[[6]](#footnote-6)´s |

Tabla 1: Implementaciones realizadas en la API de V-Track

*Fuente: elaboración propia*

En el módulo de Acceso a datos se utilizó el API de java “JDBC” para realizar las operaciones necesarias en la base de datos de V-Track. JDBC (Java Database Connectivity) es una API de java que permite la conexión con bases de datos y realizar operaciones en ellas con instrucciones SQL desde un programa desarrollado con el lenguaje de programación Java. JDBC provee una interfaz para conectarse a una base de datos sin tomar en cuenta si el proveedor de la base de datos es MySQl, Oracle, SQLServer o cualquier otro tipo de base de datos (Salas, 2013)

### 2.3 Software rodando

Se desarrolló el Servicio RESTFul que proveerá información a la aplicación web y aplicación móvil. Se puede realizar peticiones HTTP al módulo de Servicios con las operaciones GET, PUT, DELETE Y POST.

## 3. Sprint #3

Una vez creado el servicio RESTFul se procede a iniciar con el desarrollo de la aplicación web y la aplicación móvil que consumirán los recursos del servicio RESTFul.

### 3.1 Actividades

* Evaluar el uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil.
* Diseñar e implementar módulo de autenticación y recuperación de usuarios para el portal web y la aplicación móvil

### 3.2 Desarrollo

Al realizar la evaluación del uso de frameworks de aplicaciones híbridas para la implementación de la aplicación móvil, se determinó que es conveniente el uso de algún framework de aplicaciones híbridas (Ver la sustentabilidad de esta evaluación en el apéndice E), sin embargo, cuando se trata del desarrollo de una aplicación la cual tendrá mucho procesamiento y hará uso de los recursos del dispositivo móvil, se recomienda el desarrollo de una aplicación móvil con los lenguajes de programación nativos. Se decidió utilizar el framework de aplicaciones híbridas Ionic.

En este Sprint se dio inicio con el desarrollo de la aplicación web de V-Track, luego de evaluar los distintos frameworks para el desarrollo de aplicaciones web, se decidió utilizar el framework AngularJS por las siguientes razones:

* AngularJS es un framework respaldado por Google, lo que da a entender que es un framework muy confiable y bastante soportado.
* AngularJS soporta el desarrollo de aplicaciones web con el uso del patrón de diseño MVVM.
* AngularJS goza de una numerosa comunidad de desarrolladores en la web. Una simple búsqueda de AngularJS en github nos arroja un resultado de 74.922 repositorios.
* En el servidor de aplicaciones WildFly se puede desplegar contenido desarrollado con el framework AngularJS.
* AngularJS hace sencillo la trasferencia de datos entre una aplicación web y los servicios RESTFul.
* AngularJS puede ser sincronizado con el framework de aplicaciones híbridas Ionic.

Se procedió a diseñar la pantalla de login tanto para la aplicación web como la aplicación móvil, posteriormente se implementaron ambas pantallas. En ambas pantallas se encuentra un botón que permite la autenticación, este botón se encuentra en la parte central de la pantalla. Tanto en la aplicación web como en la aplicación móvil se utilizó la autenticación por Token. En el lado del servicio RESTFul se utilizó la librería de Apache “oltu.oauth2”.La siguiente ilustración muestra el proceso de autenticación en el sistema V-Track.

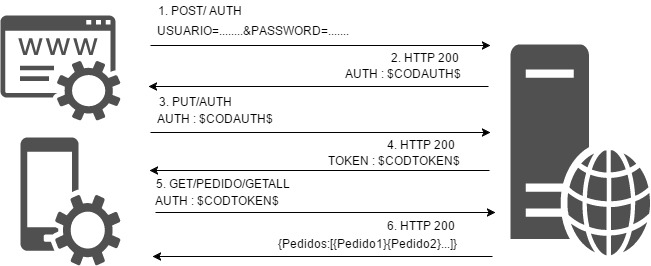


Ilustración 4 : Proceso de autenticación

*Fuente: elaboración propia*

1. La aplicación móvil o aplicación web envía una solicitud de autenticación al servicio RESTFul con los datos del usuario.
2. El servicio RESTFul valida los datos del usuario y envía un código de autorización a la aplicación cliente
3. La aplicación cliente recibe el código de autorización y lo envía de nuevo al servidor en solicitud de un Token.
4. El servicio RESTFul valida que este código de autorización fue enviado a ese usuario en específico y envía el código Token.
5. La aplicación cliente guarda el Token, luego es redirigida a la vista de información general de pedidos y solicita esta información al servidor enviando en la cabecera de la petición HTTP el código Token.
6. A partir de ahora, todas las solicitudes hechas hacia el servidor por el cliente serán validadas por la programación de interceptores[[7]](#footnote-7) en el servicio RESTFul. El interceptor en el servidor valida el Token enviado y luego el servicio responde con la información solicitada.

Luego de desarrollar la autenticación, se procedió a desarrollar la recuperación de usuario, se agregó un botón en las pantallas de login, el cual puede realizar la recuperación de usuario con el siguiente proceso:

1. Se solicita el usuario de V-Track a recuperar.
2. V-Track valida la existencia del usuario en el sistema.
3. V-Track envía al correo registrado una clave provisional para la autenticación del usuario.

Además de lo anteriormente descrito, se desarrolló un mecanismo de bloqueo de usuario, en el momento que un usuario realiza un 3er intento fallido de contraseña, V-Track bloquea el usuario y envía por correo una clave provisional. Esta clave provisional la deberá ingresar el usuario mediante la aplicación web siguiendo los pasos de desbloqueo de usuario.

### 3.3 Software rodando

El sistema V-Track cuenta con un proceso de autenticación para la aplicación móvil y la aplicación web. Todas las solicitudes hechas después de realizar el proceso de autenticación son validadas por interceptores en el servidor. La aplicación web y la aplicación móvil ya cuentan con la pantalla de login y cuentan con mecanismos de recuperación de usuario. Además, se desarrolló un mecanismo de bloqueo de usuario por intentos fallidos de contraseña.

## 4. Sprint #4

Para este Sprint se elige el diseño e implementación de la pantalla de información general de pedidos y la pantalla de información detallada de un pedido en específico. Estas dos pantallas serán implementadas tanto en la aplicación web como en la aplicación móvil.

### 4.1 Actividades

* Diseñar e implementar módulo de reportes de pedidos para el portal web y la aplicación móvil. Ver apéndice F para mayor información.

### 4.2 Desarrollo

Continuando con el desarrollo de la aplicación web y la aplicación móvil, se diseñó e implementó la pantalla de información general de pedidos en la aplicación Web. Se creó una barra lateral para que el usuario pueda realizar la redirección hacia las otras pantallas de la aplicación. En esta pantalla de información general también se agregó un reporte que indica cuantos vehículos de transporte del usuario se encuentran en las fases más importantes del proceso de llenado. Se agregó una tabla donde se muestran todos los pedidos del usuario, dando la siguiente información: código del pedido, estado y fecha de solicitud, al dar click a alguno de estos pedidos, la aplicación se redirige a la pantalla de información detallada del pedido seleccionado.

Una vez implementada esta pantalla en la aplicación web, se diseñó y se implementó la pantalla de información detallada de un pedido en específico, esta pantalla muestra la siguiente información del pedido: el código, el estado, la fecha de solicitud, la cola[[8]](#footnote-8), el cabezote[[9]](#footnote-9), nombre del chofer, la hora de inicio de llenado, la hora estimada de finalización de llenado, los eventos sucedidos durante el llenado del pedido y si el pedido ya finalizo su llenado, la hora de finalización. Es importante mencionar que estas pantallas utilizan la tecnología “Responsive[[10]](#footnote-10) Web Design”.

Al igual que en la aplicación web, se agregó una barra lateral en la aplicación móvil para que el usuario pueda redirigirse a las otras pantallas de la aplicación. Se diseñaron las pantallas anteriormente descritas y se implementaron de la misma manera que en la aplicación web, pero esta vez en la aplicación móvil.

### 4.3 Software rodando

Se puede consultar tanto información general de todos los pedidos como información detallada de un pedido en específico. Estas consultas se pueden realizar mediante la aplicación web y la aplicación móvil.

## 5. Sprint #5

Se desarrolló un módulo de solicitud de pedidos, para el desarrollo de este módulo se utilizó el rediseño hecho en el sprint # 1, en el cual, el usuario puede solicitar un pedido mediante el uso de la aplicación web y la aplicación móvil.

### 5.1 Actividades

* Diseñar e implementar módulo de solicitud de pedidos para el portal web y aplicación móvil (ver apéndice F).

### 5.2 Desarrollo

Se agregó la pantalla de “Solicitar pedido” tanto en la aplicación web como en la aplicación móvil, esta pantalla se encuentra dividida en dos partes. Una parte donde el usuario puede ingresar un código de liberación de pedido y la otra parte donde el usuario puede describir un pedido mediante el uso de un formulario. En esta pantalla también existen dos botones, en uno de ellos el usuario puede confirmar la solicitud del pedido luego de haber rellenado los campos del formulario, en el momento que el usuario presiona este botón, el sistema V-Track envía una solicitud HTTP al sistema V-TAS, en el cual se valida que se pueda cumplir con el pedido solicitado. De poderse cumplir con el pedido, V-Track envía un código de liberación de pedido a la cuenta de correo del usuario registrado, este código de liberación deberá ser ingresado por el usuario en la opción de “Ingresar código de liberación del pedido” de la pantalla “Solicitar pedido”.

Una vez confirmado el código de liberación del pedido, el sistema V-Track procede a enviar una solicitud HTTP hacia el sistema V-TAS, en el cual se asentará la solicitud del pedido de combustible.

### 5.3 Software rodando

Se implementó el rediseño de la manera de realizar la solicitud de pedidos en las plantas de distribución de combustible (sprint#1). La aplicación web y la aplicación móvil cuenta con procesos automáticos para la solicitud de pedidos.

## 6. Sprint #6

Continuando con el desarrollo del sistema V-Track, se diseñó y se implementó el módulo de sincronización. Este módulo es el encargado de realizar la extracción de la información que se encuentra en el sistema V-TAS.

### 6.1 Actividades

* Diseñar e implementar módulo de sincronización de datos entre el sistema V-TAS y V-Track.

### 6.2 Desarrollo

En la Ilustración 5 se muestra el diseño del módulo de sincronización definido

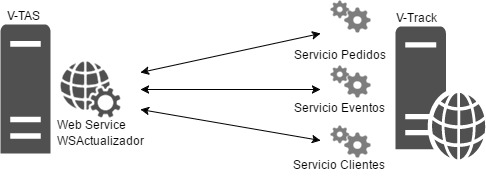


Ilustración 5: Diseño de módulo de sincronización

*Fuente: elaboración propia*

Este módulo está compuesto por 4 componentes, a continuación se muestra las características de cada uno de ellos

* WSActualizador: este componente se encuentra en el servidor de V-TAS y ofrece la invocación de tres métodos. El método pedidos retorna una lista de pedidos que están siendo atendidos o fueron atendidos por el sistema V-TAS, cada uno de estos pedidos contiene su información detallada. El método eventos retorna todos los eventos ocurridos en el proceso de llenado de un pedido en específico. El método clientes retorna una lista de todos los clientes de la planta distribuidora de combustible.
* Servicio Pedidos: este componente es un servicio Windows que tiene como funcionalidad invocar cada cierto tiempo el método pedidos del componente WSActualizador. La información consultada por este componente posteriormente es almacenada o modificada en la base de datos del sistema de V-Track.
* Servicio Eventos: este componente es un servicio Windows que tiene como funcionalidad invocar cada cierto tiempo el método eventos del componente WSActualizador. La información consultada por este componente posteriormente es almacenada o modificada en la base de datos del sistema de V-Track.
* Servicio Clientes: este componente es un servicio Windows que tiene como funcionalidad invocar cada cierto tiempo el método clientes del componente WSActualizador. La información consultada por este componente posteriormente es almacenada o modificada en la base de datos del sistema de V-Track.

Para el desarrollo de los servicios Windows se hizo uso de la librería de Apache “Commons Daemon”, gracias al uso de esta librería es posible la creación e instalación de Servicios Windows desarrollados en el lenguaje de programación Java. La comunicación entre los Servicios Windows y el componente WSActualizador se realiza mediante el protocolo SOAP. Para el desarrollo del componente WSActualizador y los servicios Windows se utilizó el API de java “JAX-WS”.

Con el uso del API JAX-WS, se simplifica el desarrollo de servicios web y clientes con una mayor independencia de la plataforma para las aplicaciones Java mediante el uso de anotaciones Java y proxies dinámicos (IBM, 2014)

### 6.3 Software rodando

Luego del 6to sprint el sistema V-Track cuenta con procesos que automatizan la extracción de información generada en el sistema V-TAS. Esta información extraída es posteriormente es consultada por la aplicación web y la aplicación móvil.

## 7. Sprint #7

Una vez creado el módulo de sincronización se procede a diseñar e implementar un proceso en la aplicación web que permite registrar el usuario por primera vez en el sistema V-Track, cambiando el estado del usuario de “NUEVO” a “ACTIVO”. Para que un usuario se pueda autoregistrar, debe ser cliente de las plantas distribuidoras de combustibles.

### 7.1 Actividades

* Diseñar e implementar módulo de autoregistro para el portal web

### 7.2 Desarrollo

Una de las funcionalidades del módulo de sincronización desarrollado es extraer información de los clientes nuevos desde el sistema V-TAS. Al momento que el módulo de sincronización extrae un cliente nuevo al sistema V-Track, envía mediante correo electrónico una clave provisional a la dirección de correo electrónico registrado del cliente nuevo. Partiendo de lo anteriormente explicado, se agregó un botón en la parte central de la pantalla de login que permite el autoregistro de un usuario en el sistema V-Track a través del proceso que se muestra en la Ilustración 6.

Ilustración 6: Proceso de autoregistro de usuario

*Fuente: elaboración propia*

Permitiendo al usuario registrarse en el sistema V-Track. Esta nueva contraseña es la misma que utilizará para autenticarse mediante la aplicación móvil.

### 7.3 Software rodando

La sincronización de clientes y el módulo de autoregistro trabajan en conjunto, el módulo de sincronización es encargado de extraer los clientes del sistema V-TAS y el módulo de autoregistro es encargado de permitirles el primer acceso al sistema.

## 8. Sprint #8

Se diseñó y se implementó el módulo de notificación del sistema V-Track. Este módulo tiene como funcionalidad el envió de notificaciones cuando los pedidos inician y terminan el proceso de llenado.

### 8.1 Actividades

* Diseñar e implementar módulo de notificación de envío y recibo de pedidos.

### 8.2 Desarrollo

Para iniciar con el desarrollo de este módulo, se creó un servicio Windows que constantemente consulta la base de datos del sistema V-Track en búsqueda de pedidos que se encuentren en la fase de “En espera” o en fase de “Cargado” (Ver Apéndice B). A los clientes dueños de los pedidos que se encuentran en las fases antes mencionadas, se les envía notificaciones mediante dos vías.

* Correo electrónico : se envía un correo a la cuenta registrada del cliente con el código del pedido, el estado, la fecha de solicitud, la cola, el cabezote, el chofer, la hora de inicio de llenado, la hora estimada de finalización de llenado y si el pedido ya finalizo su llenado, la hora de finalización. Para él envió del correo con el lenguaje de programación Java se hizo uso de la librería “javax.mail”.
* Notificación push: se envía una notificación push a los Smartphone que tengan instalada la aplicación móvil de V-Track y estén autenticados con la cuenta de correo registrada del cliente dueño del pedido. En esta notificación se envía únicamente el estado del pedido .Para él envió de las notificaciones push se hizo uso del servicio de “Firebase Cloud Messaging[[11]](#footnote-11)” (FCM), para poder hacer uso de este servicio se crearon las respectivas credenciales en su portal. La interacción con FCM se realiza como muestra la Ilustración 7.

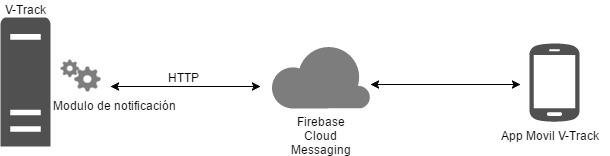


Ilustración 7: Interacción con FCM

*Fuente: elaboración propia*

Luego de que el sistema selecciona los pedidos que debe notificar, por cada pedido envía una petición HTTP al servicio REST de FCM con las credenciales creadas anteriormente en el portal del FCM, la cuenta identificadora del Smartphone y el mensaje para el cliente.

Luego de que el servicio Windows realiza la notificación por correo o por notificación push, se procede a identificar a nivel de base de datos que el cliente dueño fue notificado, esto con el fin de que la notificación no sea realizada varias veces.

### 8.3 Software rodando

El sistema V-Track cuenta con procesos automatizados para el envió de notificaciones cuando el pedido inicia y termina el proceso de llenado. La notificación se realiza por correo electrónico y mediante el envío de un mensaje a la aplicación móvil de V-Track.

## 9. Sprint #9

Para el desarrollo de este sprint se decidió realizar una evaluación sobre qué modelo de Raspberry PI es el más sofisticado a utilizarse en V-Track y posteriormente implementar un módulo de captura de estados de pedidos.

Una vez desarrollados los distintos módulos del servicio RESTful planteados en este trabajo de grado, se procedió a implementar un ambiente de despliegue para estos módulos. Se decidió utilizar Windows server 2012 como sistema operativo

### 9.1 Actividades

* Aporte tecnológico: Evaluar distintos modelos de Raspberry PI como componente en los sistemas de tracking.
* Diseñar e implementar módulo de captura de estados de pedidos
* Implementar un ambiente de despliegue de V-Track en los servidores de Vanesoft

### 9.2 Desarrollo

Las características mínimas con que debe contar el dispositivo Raspberry PI para ser adaptable al sistema V-TAS son las siguientes:

* RAM: es utilizada por el procesador para recibir instrucciones y guardar resultados. La memoria RAM mínima que debe tener la Raspberry PI es de 512 MB.
* Precio: costo del modelo de Raspberry PI publicado por su fabricante. El costo máximo con que podrá ser adquirida la Raspberry PI es de 50$.
* CPU: realiza el procesamiento de todas las instrucciones que provienen del hardware y el software. El CPU de la Raspberry PI debe tener aunque sea un núcleo y 500 MHz.
* Tamaño: dimensiones que ocupará el modelo Raspberry PI en el panel de control del sistema V-TAS. El tamaño máximo que puede tener la Raspberry PI es de 115.0mm x 100.0 mm.

Además se espera que el tiempo de respuesta de la Raspberry PI desde que recibe la señal de la lectora de RFID hasta la modificación de los datos del pedido sea de máximo 2 segundos.En La Tabla 2 se muestran las características de los distintos modelos de Raspberry PI que se encuentran en el mercado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Raspberry PI 3B | Raspberry PI B+ | Raspberry 2B | Raspberry A |
| RAM | 1 GB | 512 MB | 1 GB | 512 MB |
| Precio | 33.96$ | 35.99$ | 38.99$$ | 35.00$$ |
| CPU | ARM cuatro núcleos a 1.2 Ghz | ARM1176JZF-S a 700 MHz | ARM cuatro núcleos a 900 MHz | ARM1176JZF-S a 700 MHz |
| Tamaño | 85.6 x 56.5 mm | 85.6 x 56.5 mm | 85.6 x 56.5 mm | 85.6 x 56.5 mm |

Tabla 2:Características de modelos de Raspberry PI evaluados

*Fuente: www.raspberrypi.org*

A continuación se muestran las evaluaciones hechas en las características de los modelos Raspberry PI:

* Tamaño: los 4 modelos de Raspberry tienen el mismo tamaño, el cual es 85.6 mm x 56.5 mm, es decir, el tamaño aproximado de una tarjeta de crédito. Gracias a estas dimensiones se puede decir que cualquier modelo de Raspberry PI podrá ser ubicado fácilmente en el sistema de llenado.
* CPU: el CPU mejor ofrecido por los 4 modelos, es el del modelo Raspberry Pi 3B. Cuenta con un CPU de ARM cuatro núcleos a 1.2 GHz, lo que quiere decir que trabajaría cómodamente mientras se ejecutan distintos procesos.
* Precio: el precio mejor ofrecido por los 4 modelos, es el del modelo Raspberry Pi 3B, el precio es de 33.96$. El precio más bajo después de este, es el del modelo Raspberry A, el cual es 35.00$.
* RAM: la memoria mejor ofrecidas por los cuatros modelos, son las del modelo Raspberry Pi 3B y Raspberry 2B. Siendo el doble de rápidas que la memoria de los otros dos modelos.

Se compararon las características mínimas que debe tener la Raspberry PI con las características de los distintos modelos de Raspberry PI que existen actualmente en el mercado, y se decidió seleccionar la Raspberry PI 3B como componente del sistema de tracking. Luego de seleccionar el modelo de Raspberry PI más sofisticado para el sistema de tracking, se procedió a diseñar e implementar un módulo de captura de datos. En la Ilustración 8 se muestra el diseño de este módulo:

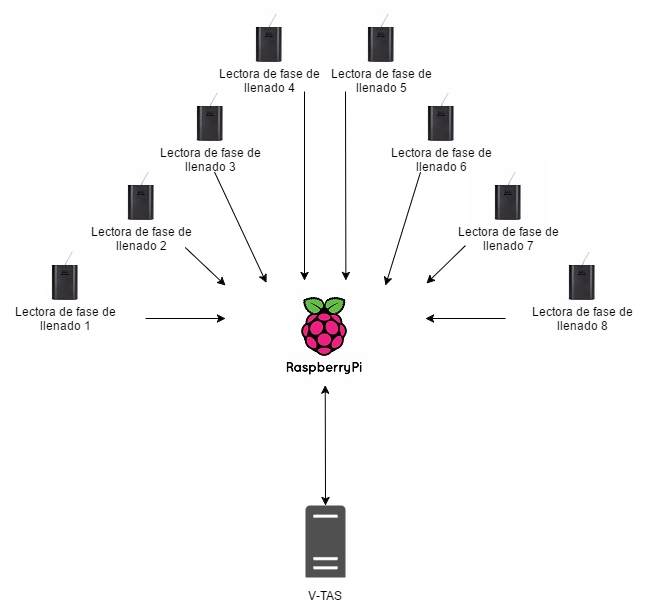


Ilustración 8: Diseño del módulo de captura de estados

*Fuente: elaboración propia*

Luego de haber diseñado el módulo de captura de datos, se procedió a implementar cada una de sus partes de la manera expuesta en la Tabla 3.

|  |  |
| --- | --- |
| **Parte del módulo** | **Implementación** |
| Lectoras de fase de llenado | Se utilizaron lectoras de tarjetas de RFID, las cuales se conectan al Raspberry PI. Para este módulo fue necesario el uso de un hub de puertos USB para poder conectar todas las lectoras. Se utilizó el modelo OMNIKEY® 5421 como modelo de lectoras de RFID. Es importante mencionar que el chofer del transporte debe acercar la tarjeta a las lectoras de RFID de las fases de llenado. Esta tarjeta es entregada al iniciar el proceso de llenado y contiene el código del pedido. |
| Raspberry PI | En este dispositivo se desarrolló e inicio un programa el cual se encuentra en un ciclo infinito y lee la entrada de las lectoras de tarjetas RFID. Para el desarrollo de este programa se hizo uso de la librería “pyusb”, la cual permite el acceso a los puertos USB del dispositivo. Por otra parte se utilizó el controlador o SDK (kit de desarrollo de software) del fabricante de las lectoras para obtener el código de las tarjetas RFID desde la lectora. Este programa desarrollado, ya tiene identificado a que puerto USB está conectado cada lectora RFID del proceso de llenado, de manera que cuando este Raspberry recibe la entrada de un código de tarjeta de RFID (código de pedido) por uno de los puertos USB, el programa identifica en qué fase de llenado se encuentra el pedido. Luego de que este programa identifica en qué fase de llenado se encuentra el pedido, se modifica el estado del pedido en la base de datos del sistema V-TAS. |
| V-TAS | En la base de datos del sistema V-TAS se modifica el estado del pedido. Esta información es extraída posteriormente por el módulo de sincronización. |

Tabla 3: Implementaciones hechas en módulo de captura de estados

*Fuente: elaboración propia*

Por medidas de seguridad se conectó el dispositivo Raspberry PI a una fuente de protección de sobre consumo y sobre temperatura. Tanto la fuente y el dispositivo Raspberry fueron colocados sobre un carril DIN.

Luego de utilizar la Raspberry PI en el sistema de tracking se definieron los siguientes resultados:

* Tiempo de respuesta: se realizaron pruebas en donde la Raspberry PI tenía como funcionalidad dejar escrito en un log el momento en que se recibía una señal desde las lectoras de RFID, el momento en que se enviaba el query de modificación de datos al sistema V-TAS y el momento en que se confirmaba la modificación de datos en el sistema V-TAS. La Raspberry PI cumple con los tiempos de respuestas esperados, ya que sus tiempos de respuestas son de menos de 2 segundos. Esto se visualiza en la Ilustración 9

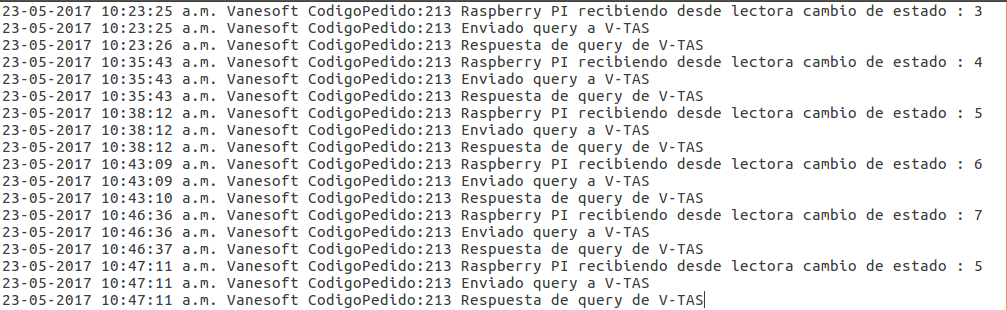


Ilustración 9: Log del módulo de captura de estado

*Fuente: elaboración propia.*

* Adaptabilidad: la Raspberry PI se consideró altamente adaptable al sistema de tracking ya que su pequeño tamaño permite ubicarlo fácilmente en algún lugar del sistema. En este caso fue ubicado en un panel de control junto con otros componentes del sistema V-TAS. Gracias a su puerto Ethernet se pudo agregar a la red del V-TAS.
* Capacidad de almacenamiento: con una memoria de 8 GB expandible, se considera con poca capacidad de almacenamiento de datos. Para efectos de este proyecto, no se utilizó la Raspberry PI como fuente de datos, lo que significa que esta característica no afecta los procesos del sistema de tracking.

Para lograr la virtualización del ambiente de despliegue, se procedió instalar la herramienta “VirtualBox[[12]](#footnote-12)” en uno de los servidores de la compañía Vanesoft. Esta herramienta nos permitió realizar la virtualización del sistema operativo “Windows Server 2012”. Se creó una máquina virtual compuesta por 2 procesadores, 4 GB de memoria RAM, 256 MB de memoria de video y 256 GB de disco duro.

Luego de lograr iniciar la máquina virtual, se procedió a instalar el paquete “jdk1.8.0\_91” y crear la variable de entorno JAVA\_HOME con la dirección del paquete anteriormente mencionado.

Se descargó WildFly en su versión “9.0.1 Final” y se creó un usuario para poder ingresar a la consola administrativa de este servidor de aplicaciones web. Se procedió a modificar el archivo de configuración “standalone.xml” para que WildFly recibiera solicitudes remotas. Para terminar con la configuración de WildFly, se subieron los archivos “.war” del sistema V-Track con el uso de la consola administrativa de WildFly.

Con el uso de librería de Apache “commons daemon”, se instalaron y se iniciaron los servicios Windows desarrollados en el módulo de sincronización y en el módulo de notificación.

### 9.3 Software rodando

La compañía Vanesoft cuenta con un ambiente de despliegue para realizar pruebas del sistema V-Track. Se pueden crear nuevos ambientes de despliegues con el uso de la herramienta Virtual Box.

# Capítulo V – Resultados

En el presente capítulo se mostrarán los resultados obtenidos en la elaboración de este trabajo instrumental de grado, basándose en los objetivos y aportes planteados:

**Objetivo:**

* Diseñar e implementar el modelo de base de datos para el sistema.

**Resultados:**

Para el desarrollo de este objetivo, se recopilaron todos los datos e información necesaria para cumplir con los objetivos de este trabajo de grado. Se representó a través de un modelo entidad-relación toda la información recopilada anteriormente (Ver apéndice C). Se crearon las tablas: vtrack\_pedido, vtrack\_usuario, vtrack\_token, vtrack\_evento, vtrack\_plantilla, vtrack\_notificacion, vtrack\_parametro, vtrack\_sincronizacion, vtrack\_plantilla\_parametro, vtrack\_solicitud\_pedido. Ver sprint #1 en el capítulo IV Desarrollo.

**Objetivo:**

* Diseñar e implementar una API RESTful que suministre datos a los componentes del sistema.

**Resultados:**

Para el desarrollo de este objetivo se analizaron todos los requerimientos que debían cumplir la aplicación web y la aplicación móvil. Posteriormente se procedió a diseñar y desarrollar una API RESTful que ayudará a cumplir requerimientos analizados anteriormente. Ver sprint #2 en el capítulo IV Desarrollo. En la Ilustración 10 se visualiza la arquitectura del API RESTful desde el ambiente de desarrollo.

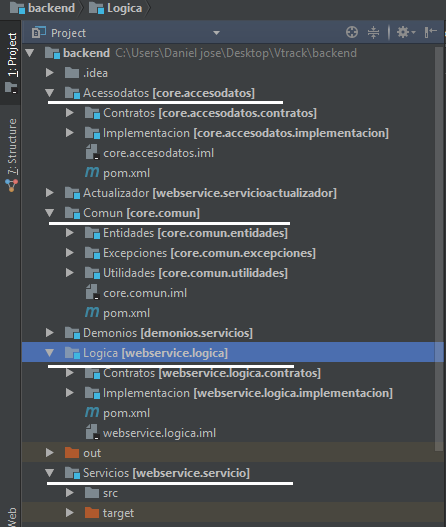


Ilustración 10: Arquitectura de API RESTful en implementación

*Fuente: elaboración propia*

**Objetivo:**

* Diseñar e implementar módulo de autenticación y recuperación de usuarios para el portal web y la aplicación móvil

**Resultado:**

Para cumplir con este objetivo, se diseñó y se implementó un módulo de autenticación que permite validar las credenciales del usuario en el inicio de sesión y en las solicitudes recibidas en el servidor RESTful luego de haber iniciado sesión. Esta autenticación y la validación de credenciales se desarrollaron con el uso de autenticación por Token e interceptores. También se creó un mecanismo de recuperación de usuario mediante la confirmación de usuario por correo electrónico. Además, se agregó un proceso de bloqueo de usuario por intentos fallidos de clave. Se agregó en la aplicación web y en la aplicación móvil la pantalla de login, las cuales utilizan los procesos explicados anteriormente. Ver Sprint # 3 en el capítulo IV Desarrollo. La pantalla de login de la aplicación web puede visualizarse en la Ilustración 11 e Ilustración 12. La pantalla de login de la aplicación móvil puede visualizarse en la Ilustración 13.

****

Ilustración 11: Pantalla login Aplicación Web

*Fuente: elaboración propia*

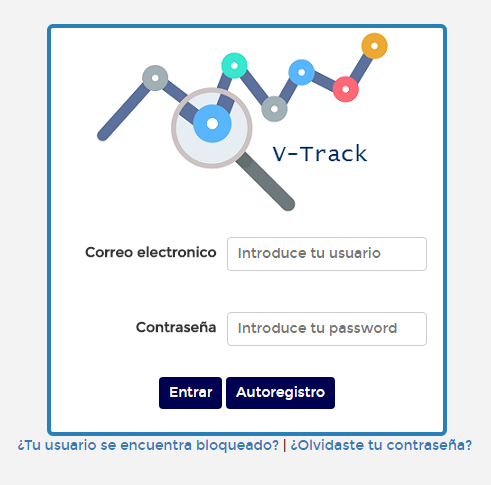
****

Ilustración 12: Pantalla login Aplicación Web – Zoom

*Fuente: elaboración propia*

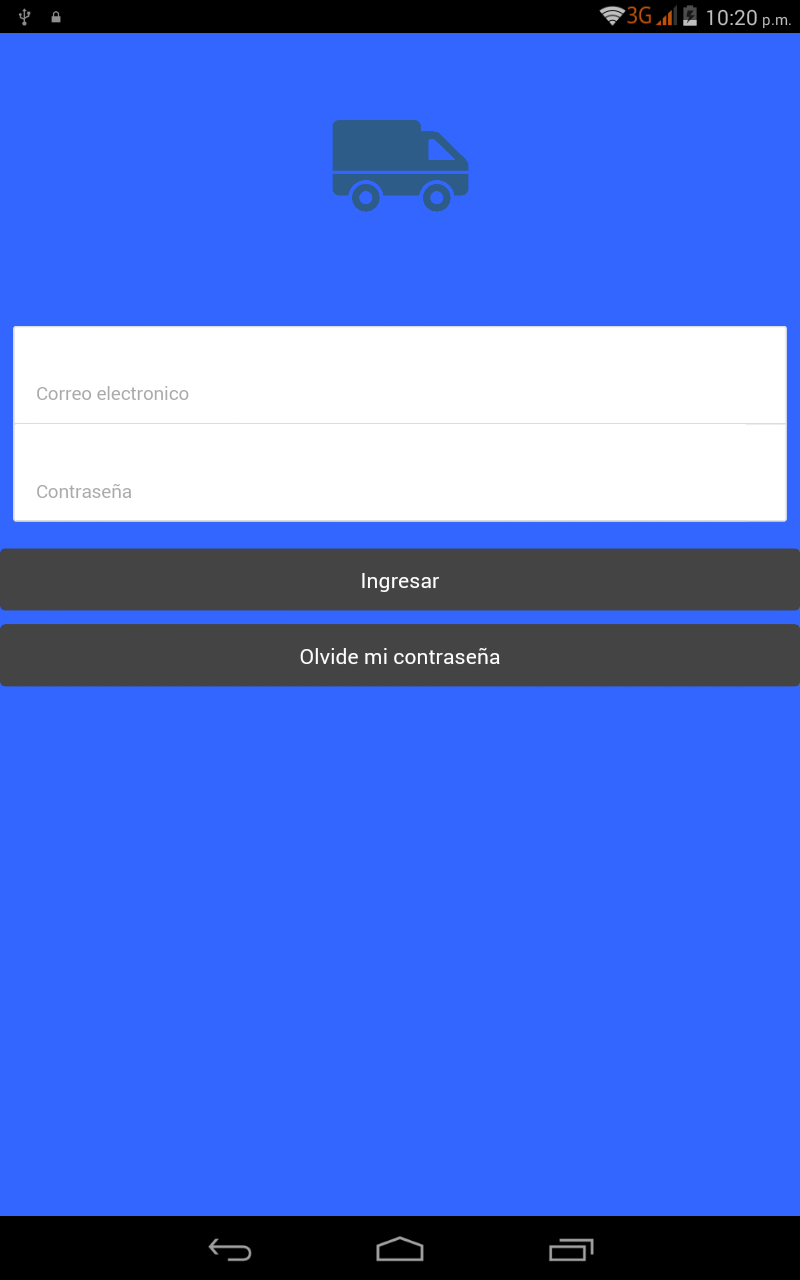
**

Ilustración 13: Pantalla login Aplicación Móvil – Zoom

*Fuente: elaboración propia.*

**Objetivo:**

* Diseñar e implementar módulo de reportes de pedidos para el portal web y la aplicación móvil.

**Resultado:**

En la aplicación móvil se crearon dos reportes, en los cuales se muestran todos los pedidos en general del usuario y en otro la información detallada de un pedido en específico. En la aplicación web se crearon tres reportes, en uno se muestran los pedidos en general del usuario, en otro la información detallada de un pedido en específico y por último, uno en el que se muestra cuantos transportes de carga se encuentran en las fases de llenado más importantes. Ver Sprint # 4 en el capítulo IV Desarrollo. La pantalla de pedidos de la aplicación web puede visualizarse en la Ilustración 14, Ilustración 15 e Ilustración 16.

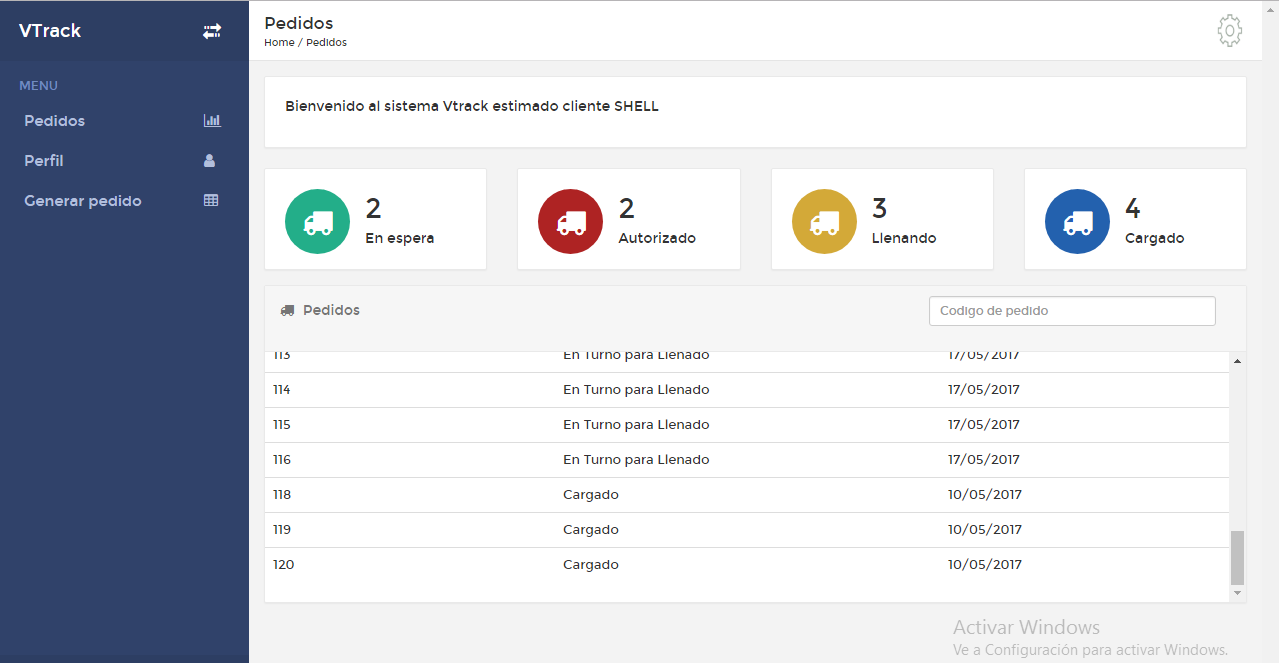


Ilustración 14: Pantalla Pedidos Aplicación web

*Fuente: elaboración propia.*

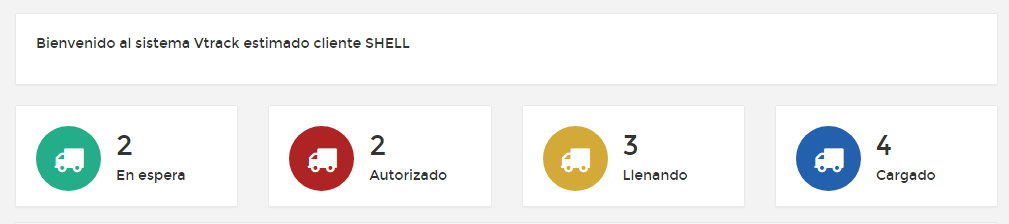


Ilustración 15: Pantalla Pedidos Aplicación web - Zoom 1

*Fuente: elaboración propia.*



Ilustración 16: Pantalla Pedidos Aplicación web - Zoom 2.

*Fuente: elaboración propia.*

La pantalla de detallado de pedido de la aplicación web puede visualizarse en la Ilustración 17, Ilustración 18 e Ilustración 19.

**

Ilustración 17: Pantalla Detallado pedido Aplicación web.

*Fuente: elaboración propia.*

**

Ilustración 18: Pantalla Detallado pedido Aplicación web - Zoom 1.

*Fuente: elaboración propia.*

**

Ilustración 19: Pantalla Detallado pedido Aplicación web - Zoom 2.

*Fuente: elaboración propia.*

La pantalla de pedidos de la aplicación móvil puede visualizarse en la Ilustración 20.

**

Ilustración 20: Pantalla Pedidos Aplicación móvil

*Fuente: elaboración propia.*

La pantalla de detallado de pedido de la aplicación móvil puede visualizarse en la Ilustración 21.

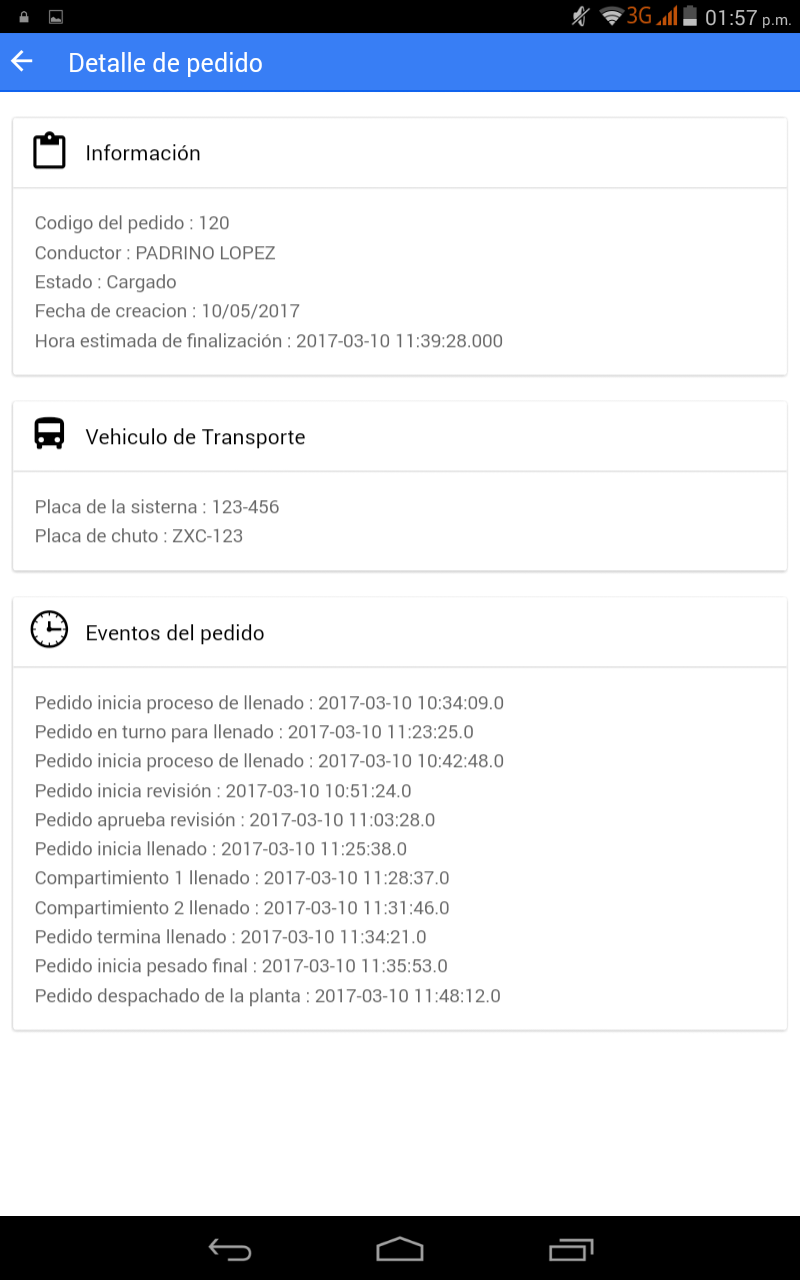


Ilustración 21: Pantalla Detallado pedido Aplicación móvil

*Fuente: elaboración propia.*

**Objetivo:**

* Diseñar e implementar módulo de solicitud de pedidos para el portal web y aplicación móvil.

**Resultados:**

Para cumplir con este objetivo se desarrolló un módulo tanto en la aplicación web como en la aplicación móvil. En este módulo el usuario puede solicitar un pedido mediante el uso de formularios y confirmar su pedido a través de un código de confirmación enviado a su correo electrónico. Ver Sprint # 5 en el capítulo IV Desarrollo.

La pantalla de solicitud de pedido de la aplicación web puede visualizarse en la Ilustración 22.

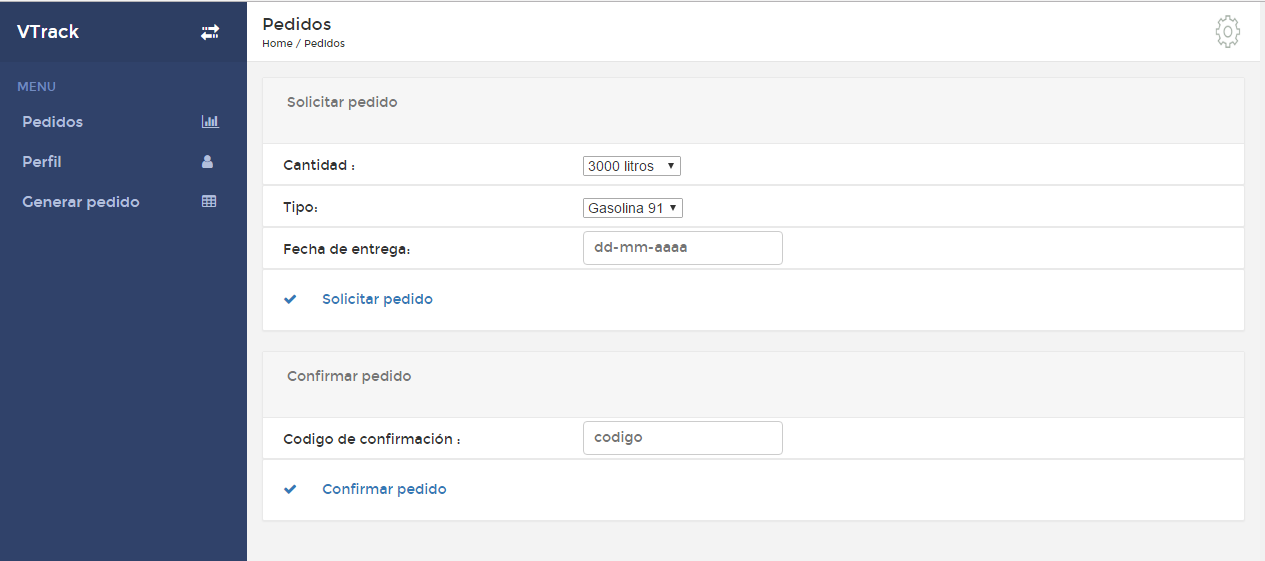


Ilustración 22: Pantalla Solicitar pedido - Confirmar pedido - Aplicación web

*Fuente: elaboración propia.*

La pantalla de solicitud de pedido de la aplicación móvil puede visualizarse en la Ilustración 23 e Ilustración 24.

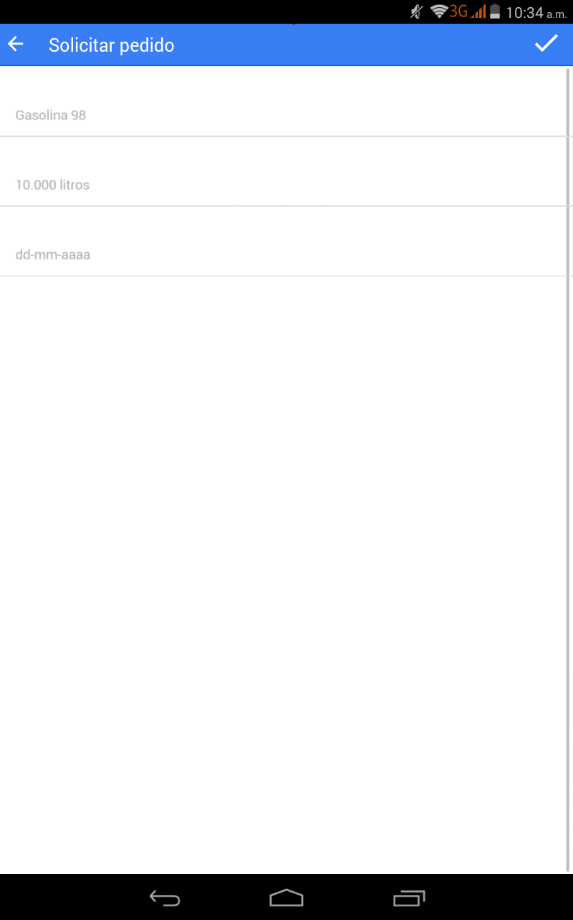
**

Ilustración 23: Pantalla solicitar pedido - Aplicación móvil

*Fuente: elaboración propia.*

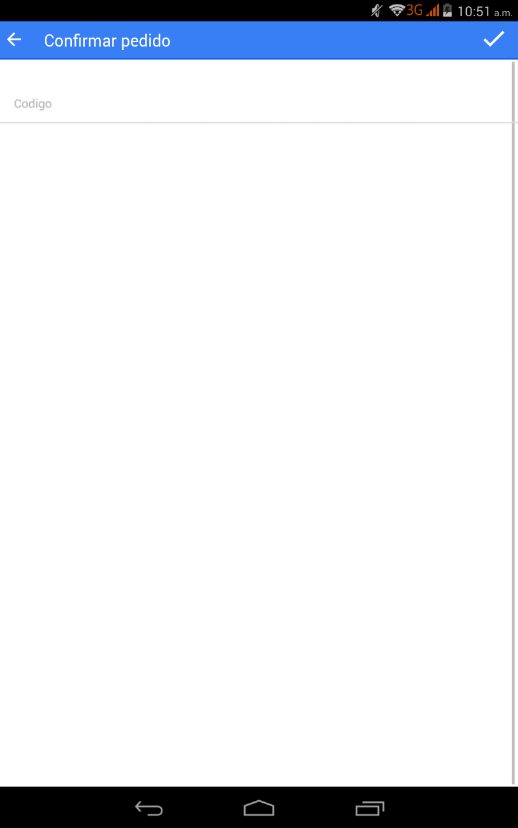


Ilustración 24: Pantalla generar pedido - Aplicación móvil

*Fuente: elaboración propia.*

**Objetivo:**

* Diseñar e implementar módulo de sincronización de datos entre el sistema V-TAS y V-Track.

**Resultados:**

Para cumplir con este objetivo se desarrolló un módulo compuesto por un servicio web en el lado del sistema V-TAS y 3 servicios Windows en el lado del sistema V-Track. Estos componentes se comunican con el protocolo SOAP y sincronizan la información a utilizarse en el sistema V-Track. Ver Sprint # 6 en el capítulo IV Desarrollo. En la Ilustración 25 se pueden visualizar los servicios Windows del módulo de sincronización.

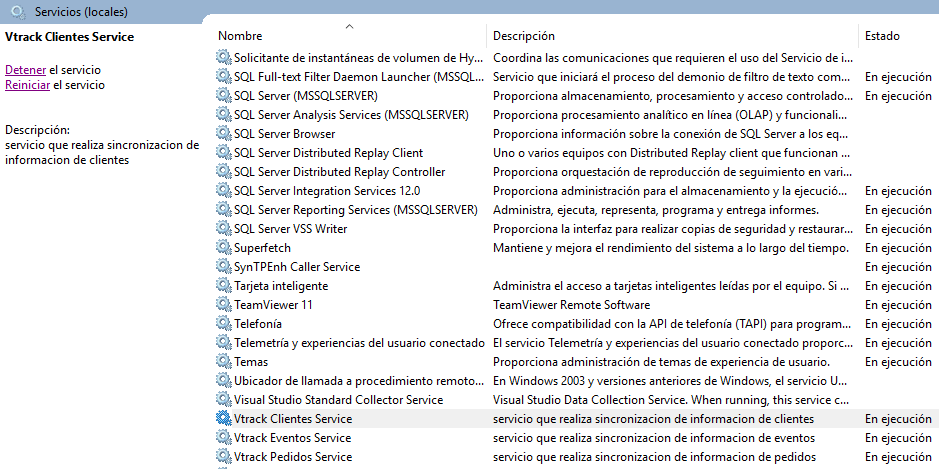


Ilustración 25: Servicios Windows de módulo de sincronización

*Fuente: elaboración propia.*

En la Ilustración 26 e Ilustración 27 se puede visualizar el WSDL del web service WSActualizador perteneciente al módulo de sincronización.

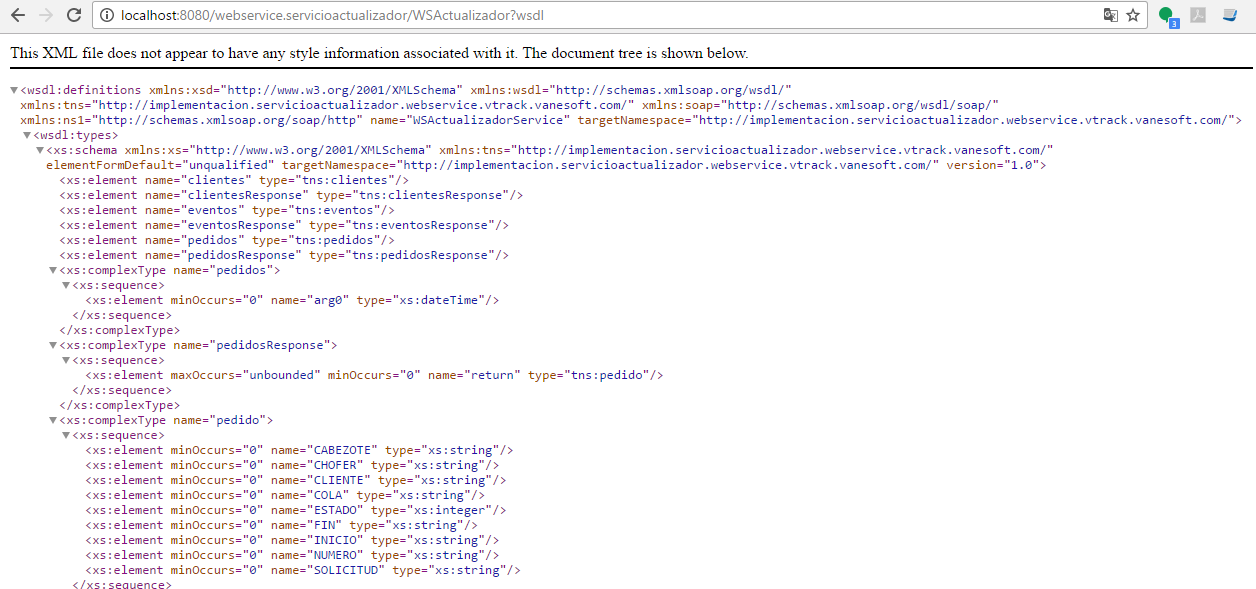
**

Ilustración 26: WSDL WSActualizador 1

*Fuente: elaboración propia.*

**

Ilustración 27: WSDL WSActualizador 2

*Fuente: elaboración propia.*

**Objetivo:**

* Diseñar e implementar módulo de autoregistro para el portal web

**Resultado:**

Para cumplir con este objetivo se desarrolló un módulo el cual registra al usuario por primera vez luego de haber sido extraído del sistema V-TAS por el módulo de sincronización. Para realizar este autoregistro el usuario debe confirmar su clave provisional enviada por correo para posteriormente cambiar su clave. El usuario debe realizar el proceso de autoregistro obligatoriamente para empezar hacer uso del sistema V-Track. En la ilustración 9 se puede notar el botón de “Autoregistro”. Ver Sprint # 7 en el capítulo IV Desarrollo.

**Objetivo:**

* Diseñar e implementar módulo de notificación de envió y recibo de pedidos.

**Resultados:**

Para cumplir con este objetivo se desarrolló un módulo compuesto por un servicio Windows el cual cada cierto tiempo consulta la base de datos del sistema V-Track en búsqueda de pedidos que se encuentren en la fase de “En cola” o “Cargado” y que su cliente dueño no haya sido notificado sobre su estado. Los pedidos que cumplan con la condición anteriormente descrita, son seleccionados para posteriormente notificarle a su cliente dueño sobre su estado mediante un correo electrónico y una notificación push. Ver Sprint # 8 en el capítulo IV Desarrollo. En la Ilustración 28 se puede visualizar el servicio Windows del módulo de notificación.

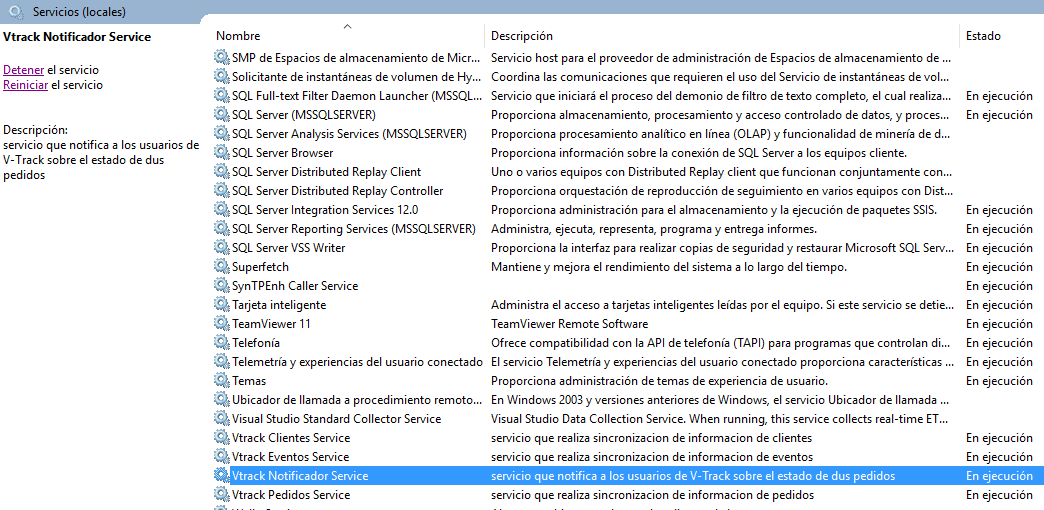


Ilustración 28: Servicios Windows de módulo de notificación

*Fuente: elaboración propia.*

En la Ilustración 29 se puede visualizar un correo recibido del módulo de notificación.

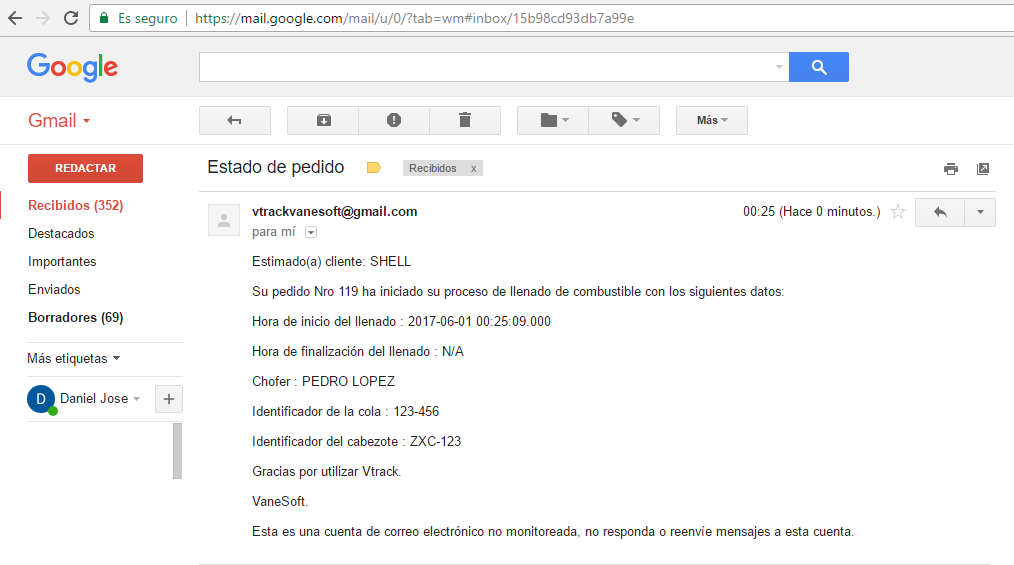
****

Ilustración 29: Correo recibido del módulo de notificación

*Fuente: elaboración propia.*

En la Ilustración 30 se puede visualizar un Push recibido del módulo de notificación.

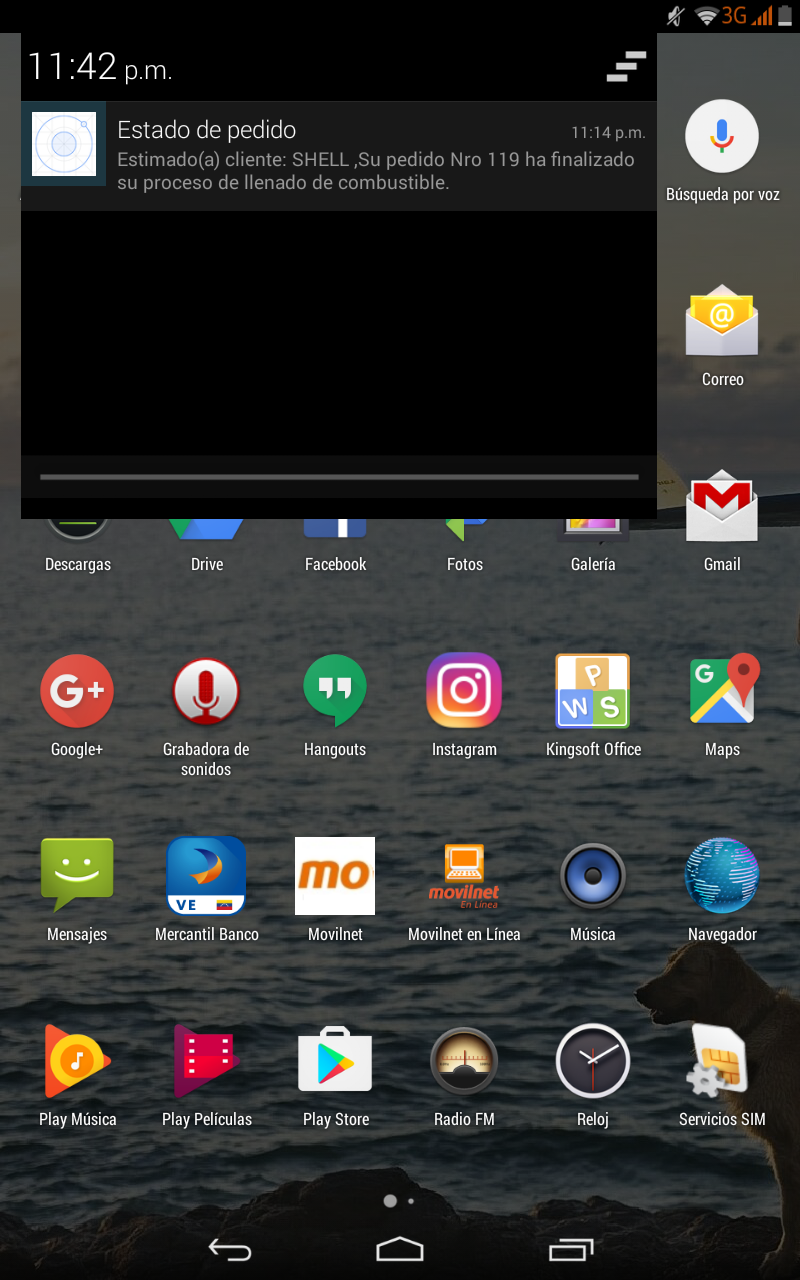
**

Ilustración 30: Push recibido del módulo de notificación

*Fuente: elaboración propia.*

**Objetivo:**

* Implementar un ambiente de despliegue de V-Track en los servidores de Vanesoft.

**Resultados:**

Para cumplir con este objetivo se creó una máquina virtual con el sistema operativo Windows Server 2012, para la virtualización se hizo uso de la herramienta VirtualBox. En esta máquina virtual se instaló el paquete “jdk1.8.0\_91” y se creó la variable de entorno JAVA\_HOME con la dirección del paquete anteriormente mencionado. Se utilizó WildFly como servidor de aplicaciones, se realizaron las respectivas configuraciones y se desplegaron los archivos “.war” generados del sistema V-Track. Por otra parte se procedieron instalar e iniciar los servicios del módulo de sincronización y notificación. Ver Sprint # 10 en el capítulo IV Desarrollo. En la Ilustración 31 se puede visualizar la máquina virtual creada con Windows Server 2012.

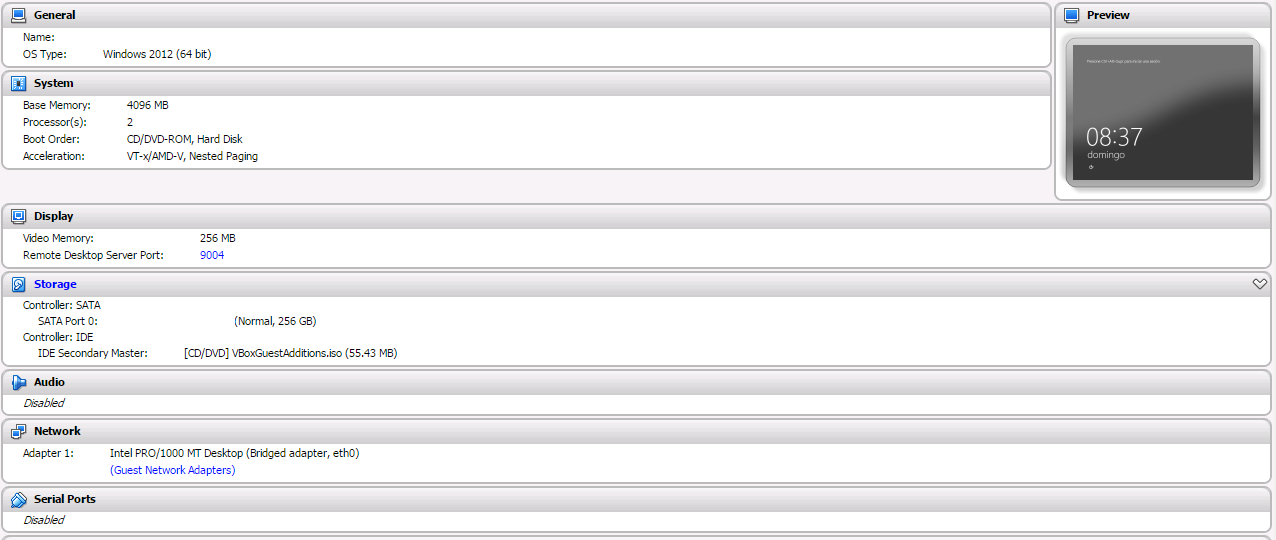


Ilustración 31: Máquina virtual de Windows Server 2012

*Fuente: elaboración propia.*

En la Ilustración 32 se puede visualizar los “.war” desplegados en WildFly.

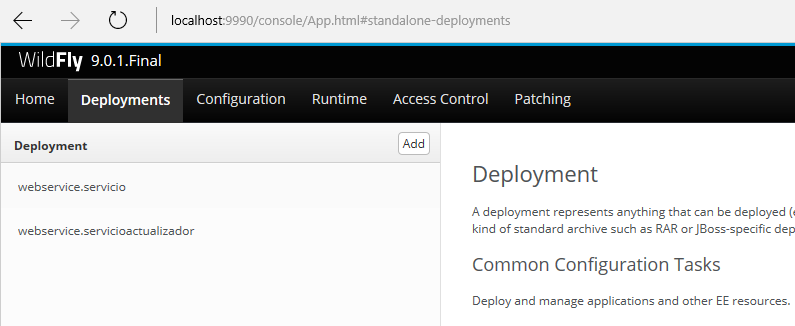
**

Ilustración 32: Despliegues en WildFly

*Fuente: elaboración propia.*

En la Ilustración 33 se puede visualizar los servicios Windows de V-Track en el ambiente de pruebas.

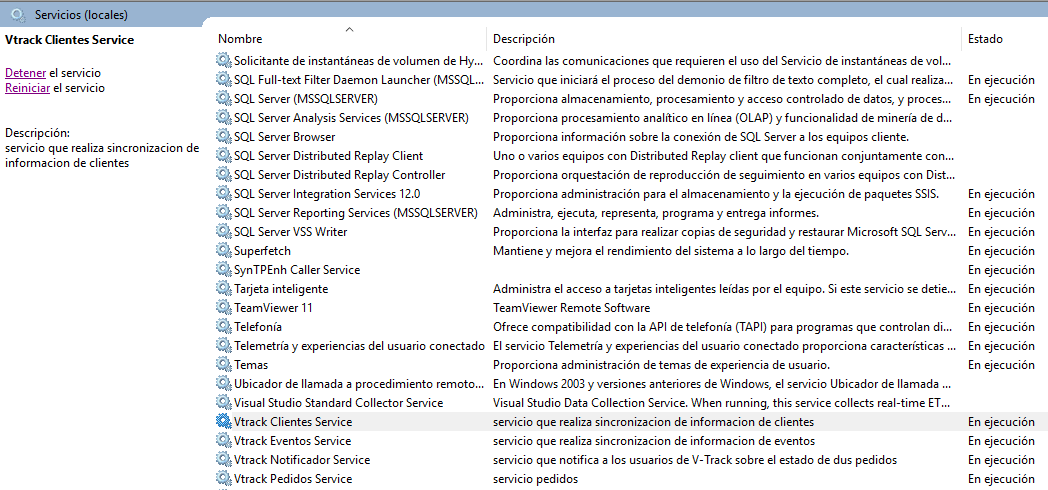


Ilustración 33: Servicios Windows de V-Track en ejecución

*Fuente: elaboración propia.*

**Objetivo:**

* Diseñar e implementar módulo de captura de estados de pedidos.

**Resultados:**

Para cumplir con este objetivo se implementó un módulo el cual tiene como función identificar en qué fase del proceso de llenado se encuentra el pedido. En cada fase de llenado se identifica el código del pedido mediante las lectoras de tarjeta de rfid, el chofer del trasporte desliza la tarjeta en cada lectora y esta señal es enviada al Raspberry PI, por último este dispositivo modifica el estado del pedido en la base de datos del sistema V-TAS. Ver Sprint # 9 en el capítulo IV Desarrollo.

En la Ilustración 34 se puede visualizar parte del módulo de captura de estado.



Ilustración 34: Módulo de captura de estado

*Fuente: elaboración propia.*

**Aporte funcional:**

* Rediseñar el proceso de solicitud de pedidos utilizado por las Plantas de Distribución de Combustibles a través de una aplicación móvil y un portal web.

**Resultados:**

Para el cumplimiento de este aporte, se creó una aplicación web y una aplicación móvil, en ellas el usuario puede solicitar pedidos a las plantas distribuidoras de combustibles. Estas dos aplicaciones interactúan constantemente con el sistema V-TAS.

**Aporte funcional:**

* Diseñar la modalidad de realizar consultas de los pedidos hechas por los clientes de las Plantas de Distribución de Combustibles.

**Resultados:**

Para el cumplimiento de este aporte, se creó un sistema compuesto por varios módulos. Mediante este sistema el usuario puede realizar consultas de sus pedidos mediante una aplicación móvil y una aplicación web. El usuario también es informado sobre sobre sus pedidos mediante la recepción de correos electrónicos y notificaciones push.

**Aporte tecnológico:**

* Evaluar el uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil.

**Resultados:**

Para el cumplimiento de este aporte, se realizó una evaluación utilizando una escala de ponderaciones para determinar las ventajas y desventajas del uso de los frameworks de aplicaciones de hibrida. Como resultado de esta evaluación, se concluyó que era conveniente el uso de un framework de aplicaciones hibrida para la implementación del aplicativo móvil, sin embargo, se determinó que cuando se trate de una aplicación que requiera de gran rendimiento y accesibilidad a los recursos del dispositivo, es conveniente el desarrollo de aplicaciones nativas.

**Aporte tecnológico:**

* Evaluar distintos modelos de Raspberry PI como componente en los sistemas de tracking.

**Resultados:**

Para el cumplimiento de este aporte, se realizó una evaluación de los distintos modelos de Raspberry que existen actualmente en el mercado, estos fueron comparados con las características mínimas con que debía cumplir la Raspberry PI en el sistema V-Track. Se determinó que la Raspberry PI 3B es la mejor opción para ser utilizado como componente del sistema V-Track. Además se determinó que la Raspberry PI 3B cumplía con los tiempos de respuesta esperando y es de gran adaptabilidad en los sistemas tracking.

# Capítulo VI – Conclusiones y recomendaciones

Este capítulo tiene como objetivo mostrar las conclusiones y recomendaciones recopiladas por los objetivos desarrollados a lo largo de este trabajo de grado.

## Conclusiones

* El análisis del sistema V-TAS fue de mucha importancia para identificar la manera en que se podría diseñar e implementar un sistema de tracking altamente acoplable a dicho sistema, también gracias a esto se pudo diseñar una estructura de base de datos para almacenar toda la información necesaria en el tracking de los pedidos.
* El módulo de sincronización y módulo de captura de estados permiten obtener toda la información solicitada por el usuario en la aplicación móvil y la aplicación Web, gracias a esto se pudo rediseñar la manera en que los clientes de plantas distribuidoras de combustibles ejecutan sus consultas para tener conocimientos sobre sus pedidos. Esto permite a los usuarios mejorar su planificación y evitar problemas de suministro de combustible en sus estaciones de servicio.
* La selección de un framework para el desarrollo de una aplicación híbrida, permite que el desarrollo sea más rápido y menos costoso. Además este tipo de desarrollo posee una documentación completa y una comunidad activa, ayudando así en la resolución de problemas durante el desarrollo.
* El cliente se mantiene notificado recibiendo de manera inmediata correos electrónicos y notificaciones push cuando sus pedidos inician y terminan el proceso de llenado.
* El módulo de solicitud de pedidos permite que el usuario pueda solicitar un pedido de cualquier tipo mediante la aplicación móvil o la aplicación web, gracias a esto se pudo rediseñar la manera en que los clientes de plantas distribuidora de combustibles solicitan sus pedidos.
* El modelo de Raspberry PI 3B es en la actualidad el modelo más sofisticado para utilizarse en la implementación de un sistema de tracking. También se concluye que el Raspberry PI se comporta como un buen componente en el sistema V-Track, gracias a su pequeño tamaño, su escalable capacidad de almacenamiento y su capacidad de responder en tiempos cortos.

## Recomendaciones

* Se propone incorporación de un sistema de GPS en la flota de unidades de transporte de los clientes de las plantas distribuidora de combustible. Este sistema de GPS puede ser acoplado con el sistema V-Track, de manera que se muestre la ubicación del transporte.
* Se propone el desarrollo de un módulo de gestión de roles y permisología del sistema V-Track.
* Al desarrollar una aplicación móvil que requiera de gran rendimiento y accesibilidad a los recursos del dispositivo, se recomienda el desarrollo de una aplicación nativa.

# Referencias bibliográficas

a3m. (2014). *NOVEDADES EN TARJETAS DE PROXIMIDAD RFID*. Recuperado el 01 de 03 de 2017, de NOVEDADES EN TARJETAS DE PROXIMIDAD RFID: http://www.a3m.eu/es/tarjetas-plasticas/tarjetas-de-proximidad-rfid.html

Alvarez, M. A. (18 de julio de 2001). *Qué es Java.* Recuperado el 09 de 04 de 2017, de Qué es Java: https://desarrolloweb.com/articulos/497.php

Alvarez, M. A. (19 de enero de 2015). *Qué es Responsive Web Design.* Recuperado el 18 de 04 de 2017, de Qué es Responsive Web Design: https://desarrolloweb.com/articulos/que-es-responsive-web-design.html

appio. (05 de 06 de 2017). *TIPOS DE APPS: NATIVAS, HÍBRIDAS Y WEB APPS.* Recuperado el 09 de 03 de 2017, de TIPOS DE APPS: NATIVAS, HÍBRIDAS Y WEB APPS: http://appio.es/tipos-de-apps/

*Apps Híbridas vs Nativas vs Generadas. ¿Qué decisión tomar?* (23 de 02 de 2014). Recuperado el 09 de 03 de 2017, de Apps Híbridas vs Nativas vs Generadas. ¿Qué decisión tomar?: http://www.innovaportal.com/innovaportal/v/696/1/innova.front/apps-híbridas-vs-nativas-vs-generadas-que-decision-tomar

Arboix, N. (28 de Octubre de 2013). *Raspberry Pi: componentes y montaje.* Recuperado el 06 de 04 de 2017, de Raspberry Pi: componentes y montaje: http://ididactic.com/edblog/raspberry-pi-componentes-y-montaje/

Azaustre, C. (19 de 02 de 2015). *¿Qué es la autenticación basada en Token?* Recuperado el 17 de 04 de 2017, de ¿Qué es la autenticación basada en Token?: https://carlosazaustre.es/blog/que-es-la-autenticacion-con-token/

Bennett, D. (06 de 04 de 2017). *MVVM with RxJava.* Obtenido de MVVM with RxJava: https://www.codevate.com/blog/13-better-android-software-development-mvvm-with-rxjava

Castellanos, M. (8 de Julio de 2013). *DATA ACCESS OBJECT (DAO).* Recuperado el 06 de 04 de 2016, de DATA ACCESS OBJECT (DAO): http://castellanosmiguel.blogspot.com/2013/07/dataaccess-object-dao-definicion-dao-es.html

Castro, L. (12 de Marzo de 2016). *Significado de URI, y la diferencia con URL.* Recuperado el 15 de 04 de 2017, de Significado de URI, y la diferencia con URL: http://aprenderinternet.about.com/od/ConceptosBasico/a/Que-Es-Uri.htm

Cloudbuilder Next. (08 de Abril de 2017). *WildFly, el servidor de aplicaciones Java que multiplica su rendimiento en Cloud.* Recuperado el 2017 de 04 de 06, de WildFly, el servidor de aplicaciones Java que multiplica su rendimiento en Cloud: https://blog.arsys.es/wildfly-cloud/

Consultec. (30 de 01 de 2017). *¿Desarrollar una App nativa o híbrida?* Recuperado el 09 de 03 de 2017, de ¿Desarrollar una App nativa o híbrida?

EcuRed. (05 de Marzo de 2009). *EcuRed*. Recuperado el 29 de 03 de 2017, de EcuRed: https://www.ecured.cu/Framework

facilCloud. (17 de Agosto de 2015). *WILDFLY.* Recuperado el 06 de 04 de 2017, de WILDFLY: http://facilcloud.com/docs/es\_ES/wildfly/

Firebase. (10 de 09 de 2015). *Firebase Cloud Messaging.* Recuperado el 30 de 05 de 2017, de Firebase Cloud Messaging: https://firebase.google.com/docs/cloud-messaging/

Gala, F. J. (22 de 01 de 2015). *Qué es el patron de diseño Command.* Recuperado el 06 de 04 de 2017, de Qué es el patron de diseño Command: https://rootear.com/desarrollo/patron-command

IBM. (11 de 05 de 2014). *JAX-WS.* Recuperado el 01 de 06 de 2017, de JAX-WS: https://www.ibm.com/support/knowledgecenter/es/SSEQTP\_8.0.0/com.ibm.websphere.base.doc/info/aes/ae/cwbs\_jaxws.html

IBM. (02 de 04 de 2015). *Iniciación a IBM JAX-RS.* Recuperado el 15 de 04 de 2017, de Iniciación a IBM JAX-RS: https://www.ibm.com/support/knowledgecenter/es/SSAW57\_8.0.0/com.ibm.websphere.nd.doc/info/ae/ae/twbs\_jaxrs\_getstarted.html

ids. (19 de Noviembre de 2014). *Scrum, 3era. parte – Eventos.* Recuperado el 12 de 04 de 2017, de Scrum, 3era. parte – Eventos: http://www.ids.com.mx/desarrollo-profesional/comunidad-ids/blog/scrum-3era-parte-eventos

innovaage. (12 de 02 de 2014). *Apps Híbridas vs Nativas vs Generadas. ¿Qué decisión tomar?* Obtenido de Apps Híbridas vs Nativas vs Generadas. ¿Qué decisión tomar?: http://www.innovaportal.com/innovaportal/v/696/1/innova.front/apps-híbridas-vs-nativas-vs-generadas-que-decision-tomar

JSON. (06 de 04 de 2017). *Introducción a JSON.* Obtenido de Introducción a JSON: http://www.json.org/json-es.html

Juarez, M. (31 de 05 de 2011). *Factory.* Recuperado el 06 de 04 de 2017, de Factory: http://migranitodejava.blogspot.com.es/2011/05/factory-method.html

LANCETALENT. (20 de 02 de 2014). *Los 3 Tipos De Aplicaciones Móviles: Ventajas E Inconvenientes.* Recuperado el 09 de 03 de 2017, de Los 3 Tipos De Aplicaciones Móviles: Ventajas E Inconvenientes: https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/

Lars Vogel (c). (10 de 08 de 2015). *Git - Tutorial.* Recuperado el 12 de 04 de 2017, de Git - Tutorial: http://www.vogella.com/tutorials/Git/article.html#gitterminlogy

Marquez, F. J. (06 de 02 de 2014). *Interceptores.* Recuperado el 17 de 04 de 2017, de Interceptores: https://infow.wordpress.com/2014/02/06/interceptores/

Oracle. (2013). *Oracle Java Documentation*. Recuperado el 01 de 03 de 2017, de Oracle Java Documentation: http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html

Piriz, F. M. (09 de 06 de 2010). Recuperado el 01 de 03 de 2017, de https://fernandomachadopiriz.com/2010/06/09/una-simple-introduccin-al-patrn-model-view-viewmodel-para-construir-aplicaciones-silverlight-y-windows-presentation-foundation/

QODE. (3 de Agosto de 2014). *¿Qué es una App Nativa?* Recuperado el 16 de 04 de 2017, de ¿Qué es una App Nativa?: http://qode.pro/blog/que-es-una-app-nativa/

raona. (11 de 05 de 2013). *¿App nativa, web o híbrida?* Recuperado el 09 de 03 de 2017, de ¿App nativa, web o híbrida?: http://www.raona.com/es/Solutions/Template/163/App-nativa-web-o-h%C3%ADbrida-

Raspberry. (06 de 04 de 2017). *FAQS.* Obtenido de FQS: https://www.raspberrypi.org/help/faqs/#introWhatIs

*RFID Tecnología de identificación por Radiofrecuencia y sus principales aplicaciones.* (2007). Madrid: (Junta de Castilla y Leon. Recuperado el 06 de 04 de 2017

RFLCargo. (2016). *¿Qué es el sistema Traking? ¿Cómo se aplica a la logística?* Recuperado el 06 de 04 de 2017, de ¿Qué es el sistema Traking? ¿Cómo se aplica a la logística?: http://rflcargo.com/2016/03/16/que-es-el-sistema-tracking-como-se-aplica-a-la-logistica/

RFLCargo. (2016).

Salas, I. (17 de Septiembre de 2013). *TUTORIAL JDBC CON APLICACIONES DE EJEMPLO.* Recuperado el 15 de 04 de 2017, de TUTORIAL JDBC CON APLICACIONES DE EJEMPLO: http://programandoointentandolo.com/2013/09/tutorial-jdbc-con-aplicaciones-de-ejemplo.html

Smart, J. F. (5 de Diciembre de 2005). *An introduction to Maven 2.* Recuperado el 10 de 04 de 2017, de An introduction to Maven 2: http://www.javaworld.com/article/2072203/build-ci-sdlc/an-introduction-to-maven-2.html

SOFTENG. (01 de 05 de 2010). *Proceso y Roles de Scrum.* Recuperado el 12 de 04 de 2017, de Proceso y Roles de Scrum: https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html

Trigas Gallego, M. (07 de Agosto de 2015). *Metodología Scrum*. Recuperado el 12 de 04 de 2017, de Metodología Scrum: http://www.quimbiotec.gob.ve/sistem/auditoria/pdf/ciudadano/mtrigasTFC0612memoria.pdf

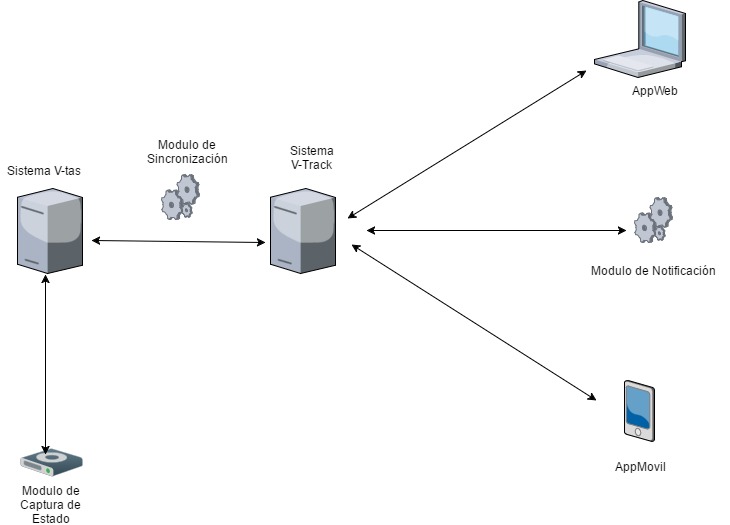
Trigas Gallego, M. (07 de Agosto de 2015). *Metodología Scrum*. Recuperado el 07 de Julio de 2016, de Metodología Scrum: http://www.quimbiotec.gob.ve/sistem/auditoria/pdf/ciudadano/mtrigasTFC0612memoria.pdf

Urrego, J. S. (01 de 03 de 2016). *Aplicaciones híbridas: estigmas, realidad y futuro.* Recuperado el 07 de 04 de 2017, de Aplicaciones híbridas: estigmas, realidad y futuro: http://acis.org.co/revista138/content/aplicaciones-h%C3%ADbridas-estigmas-realidad-y-futuro

Wikipedia. (16 de marzo de 2017). *API Java.* Recuperado el 15 de 04 de 2017, de API Java: https://es.wikipedia.org/wiki/API\_Java

# Apéndices

## Apéndice A - Arquitectura del sistema V-track en acoplamiento con la plataforma V-TAS



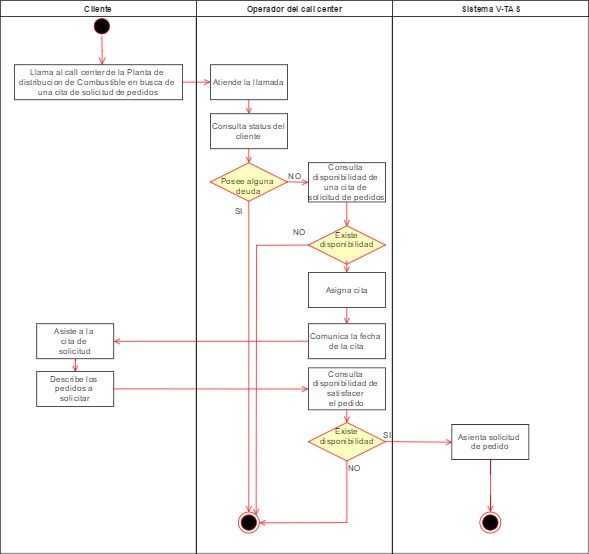
## Apéndice B – Fases del proceso de llenado

A continuación se describe las fases del proceso de llenado que actualmente se realizan en las plantas distribuidoras de combustible que utilizan V-TAS:

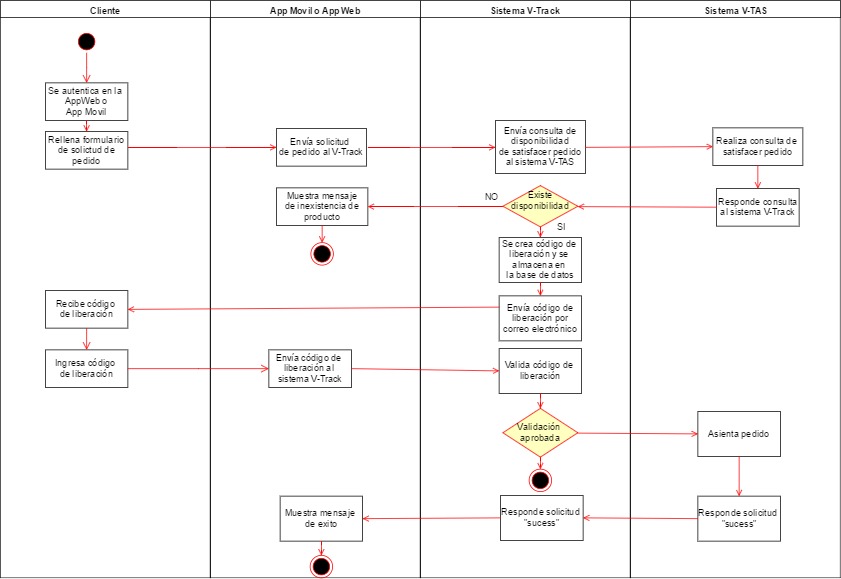
1. En espera : en esta fase el transporte de carga que es enviado por el cliente llega a la planta distribuidora y es registrado en el sistema V-TAS para posteriormente iniciar su proceso de llenado, si al ser registrado existe otro pedido, el pedido inicial es puesto en una cola a la espera de ser llamado para su turno de llenado.
2. En Turno para Llenado: en esta fase se le indica al chofer del transporte de carga que ha llegado su turno de llenado y este debe conducir el transporte de carga correspondiente hacia el punto de inicio de llenado de la planta distribuidora.
3. Autorizado y Llamado a Llenado: en esta fase el transporte de carga se encuentra en el punto de inicio de llenado de la planta distribuidora. Los operarios antes de iniciar el proceso de llenado realizan una revisión del transporte de carga y los componentes del sistema de llenado. Una vez que los operarios aprueban la revisión, autorizan a iniciar el proceso de llenado.
4. Pesado Inicial : en esta fase el transporte de carga es ubicado en una balanza para registrar el peso del mismo por los siguientes motivos :
   1. Asegurar que ninguno de los compartimientos de carga de combustible del transporte contiene combustible de pedidos anteriores.
   2. Registrar el peso inicial del transporte de carga, este registro es utilizado como punto de comparación a medida que los compartimientos son llenados.
5. Llenando: luego de registrar el peso inicial, el sistema V-TAS realiza un cálculo de cuanto deberá pesar el transporte de carga cuando haya cargado la cantidad de combustible solicitada por el cliente (peso esperado). El transporte de carga es puesto en el punto de llenado de la planta distribuidora y se empieza con el llenado de los compartimientos, en este punto, el transporte de carga también es puesto sobre una balanza. A medida que los compartimientos se van llenando se compara el peso del transporte con el peso esperado, en el momento que el transporte de carga llega al peso esperado, el proceso de llenado de los compartimientos es detenido.
6. Llenado: en esta fase el transporte es retirado del punto de llenado y se lleva a una balanza donde será pesado por última vez.
7. Pesado Final: en esta fase el transporte de carga es pesado nuevamente para asegurarse que este no retire más combustible de lo solicitado por el cliente.
8. Cargado: en esta fase el transporte de carga se retira de la planta distribuidora de combustible.

## Apéndice C – Diagramas de actividades del proceso de solicitud de pedido

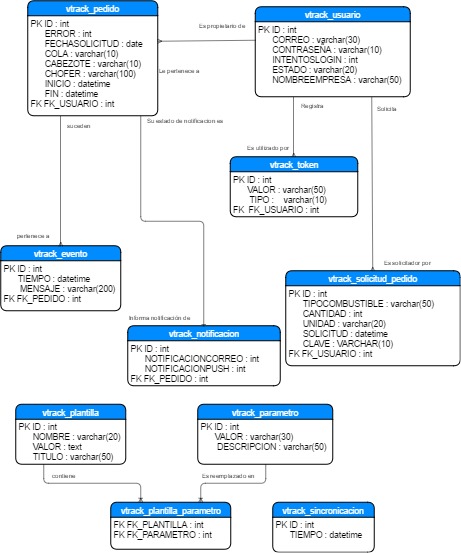
1. Proceso de solicitud antiguo



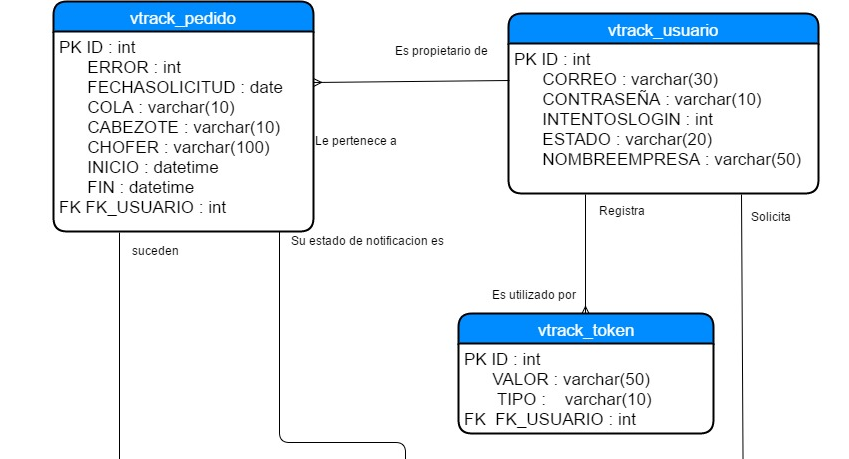
2. Proceso de solicitud nuevo



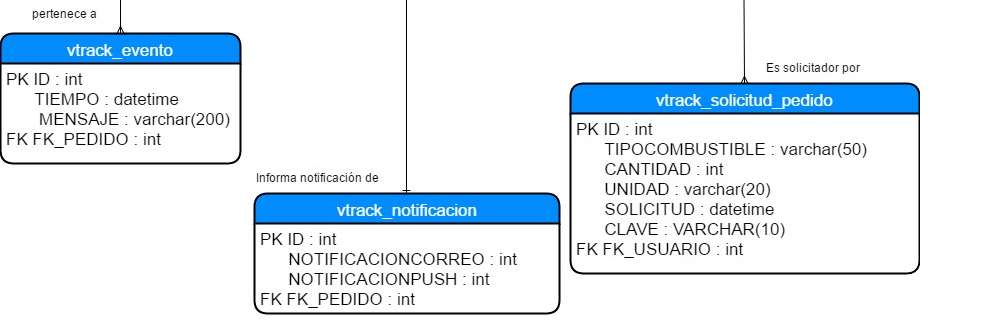
## Apéndice D – Modelo Entidad-Relación



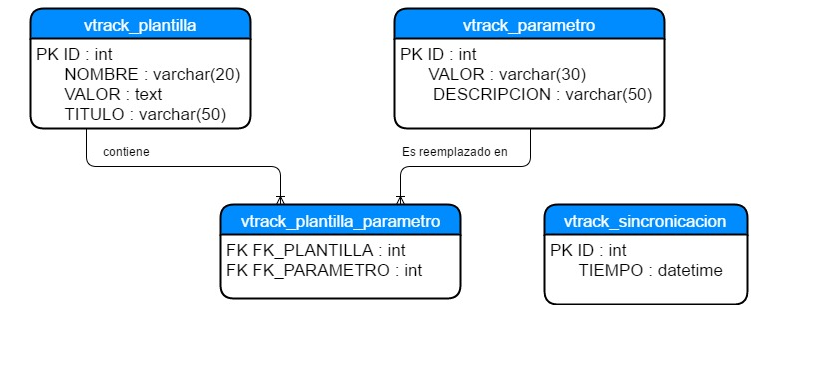
**Sección # 1**

****

**Sección # 2**

****

**Sección #3**

****

## Apéndice E – Evaluación de Framework de aplicaciones híbridas

Actualmente existen dos maneras de desarrollar aplicaciones móviles, una manera es el desarrollo de aplicaciones móviles haciendo uso de su lenguaje nativo (aplicaciones nativas) y otro modo es el desarrollo haciendo uso de algún framework para la creación de una aplicación hibrida (aplicaciones híbridas), es decir, de no hacer uso de los frameworks de aplicaciones híbridas, se tendría que recurrir al desarrollo de una aplicación nativa. Para la evaluación del uso de frameworks de aplicaciones híbridas para la implementación del aplicativo móvil, se decidió tomar como comparación el desarrollo de aplicaciones móviles haciendo uso de su lenguaje nativo.

Se decidió evaluar estos dos tipos de desarrollo en tres tipos de experiencias que se explican a continuación:

* Experiencia para el usuario final: percepción que tendrá el usuario al utilizar la aplicación móvil al finalizar el desarrollo de la misma.
* Experiencia para el desarrollador: percepciones que se tendrán durante el desarrollo de la aplicación móvil.
* Experiencia para el proyecto: efectos que se tendrán en el proyecto V-Track

Estos tipos de experiencias se dividen en varios criterios a evaluar .Para estos criterios, se tomó como escala de calificación la siguiente:

|  |  |  |
| --- | --- | --- |
| Efectividad | Calificación | Definición |
| Excelente | 4 | Siempre supera la satisfacción del criterio evaluando. |
| Muy bueno | 3 | A veces supera la satisfacción del criterio evaluando. |
| Bueno | 2 | Satisface el criterio evaluando. |
| Regular | 1 | Satisface el criterio evaluando con dificultades. |
| Malo | 0 | No satisface el criterio evaluando. |

Tabla 4: Escala de calificación para la evaluación del uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil.

*Fuente: elaboración propia.*

A continuación se explica el método de evaluación de cada experiencia y sus criterios a evaluar.

Experiencia para el usuario final: se le entregó un Smartphone con una aplicación nativa instalada y una aplicación hibrida instalada a tres clientes de la empresa Vanesoft. En esta experiencia se evaluaran los siguientes criterios.

* Rendimiento: es la rapidez con que la aplicación se ejecuta en el dispositivo móvil. Se le indico al cliente que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué la rapidez con que se ejecuta la aplicación en el Smartphone

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

* Interfaz gráfica: es el medio con que el usuario final puede comunicarse con el dispositivo móvil. Se le indico al cliente que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué la amigabilidad de las pantallas de la aplicación en el Smartphone

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

Experiencia para el desarrollador : se le hizo una serie de preguntas a 5 desarrolladores de la empresa Vanesoft luego de haberse documentado sobre las ventajas y desventajas de ambos tipos de desarrollo. En esta experiencia se evaluaron los siguientes criterios.

* Acceso al hardware del dispositivo**:** es la accesibilidad a los recursos del dispositivo, es decir, accesibilidad a la cámara, al velocímetro del dispositivo, botones laterales etc. Se le indico al desarrollador que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué la accesibilidad al hardware del Smartphone para el tipo de desarrollo indicado:

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

* Curva de aprendizaje: es el tiempo requerido para que el desarrollador llegue a dominar el tipo de desarrollo. Se le indico al desarrollador que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué la curva de aprendizaje para el tipo de desarrollo indicado:

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

* Rapidez de desarrollo: se define como la velocidad con que se lleva acabo el desarrollo de la aplicación móvil. Se le indico al desarrollador que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué la rapidez de desarrollo para el tipo de desarrollo indicado:

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

* Comunidad desarrolladora: se define como los espacios para plantear problemáticas y sus posibles soluciones. Se le indico al desarrollador que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué la comunidad desarrolladora para el tipo de desarrollo indicado:

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

Experiencia para el proyecto: se le hizo una serie de preguntas a 5 gerentes de proyecto de la empresa Vanesoft luego de haberse documentado sobre las ventajas y desventajas de ambos tipos de desarrollo. En esta experiencia se evaluaron los siguientes criterios.

* Fácilmente escalable: se define como la facilidad de agregar nuevas funcionalidades a la aplicación en un futuro. Se le indico al gerente de proyecto que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué la facilidad de agregar nuevas funcionalidades a la aplicación para el tipo de desarrollo indicado:

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

* Integración con el negocio: se define como las librerías que presta el tipo de desarrollo para la creación de la aplicación móvil del sistema V-Track. Se le indico al gerente de proyecto que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué la integración con el negocio para el tipo de desarrollo indicado:

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

* Licenciamiento: se define como la autorización que sede el autor a otros programadores para utilizar los frameworks o lenguajes de programación. Se le indico al gerente de proyecto que respondiera la siguiente pregunta.

“En una escala del 0 (mala) al 4(excelente) evalué el licenciamiento para el tipo de desarrollo indicado:

Aplicación nativa: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aplicación hibrida: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”

En la Tabla 5 se muestra la matriz donde se puede observar el promedio de las evaluaciones obtenidas en cada criterio a evaluar.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Criterio a evaluar | Desarrollo de aplicación nativa | Desarrollo de aplicación hibrida |
| Experiencia para el usuario final | Rendimiento | 4 | 2 |
| Interfaz de usuario | 3 | 4 |
| Experiencia para el desarrollador | Acceso al hardware del dispositivo | 4 | 2 |
| Curva de aprendizaje | 1 | 4 |
| Rapidez de desarrollo | 1 | 3 |
| Comunidad desarrolladora | 4 | 3 |
| Experiencia para el proyecto | Fácilmente escalable | 2 | 4 |
| Integración con el negocio | 4 | 4 |
| Licenciamiento | 4 | 4 |

Tabla 5: Tabla de ponderación para la evaluación del el uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil.

*Fuente: elaboración propia*

Para obtener una ponderación final para cada tipo de desarrollo, se decidió sumar de cada experiencia, el promedio de las calificaciones obtenidas en los criterios a evaluar, es decir, se aplicó la siguiente formula:

*Fuente: recomendación de tutor empresarial.*

CCEE = calificación de criterio a evaluar de experiencia

En la tabla 6 se muestra la matriz donde se puede observar las ponderaciones finales de los dos tipos de desarrollo.

|  |  |
| --- | --- |
| Tipo de desarrollo | Ponderación final |
| Desarrollo de aplicación nativa | 10 |
| Desarrollo de aplicación hibrida | 9,3 |

Tabla 6: Ponderación final de evaluación del uso de “frameworks” de aplicaciones híbridas para la implementación del aplicativo móvil.

*Fuente: elaboración propia*

A continuación se hará un análisis sobre los criterios que se evaluaron anteriormente.

Rendimiento: se concluyó lo siguiente (LANCETALENT, 2014).

* Desarrollo de aplicación nativa: este tipo de desarrollo ofrece un excelente rendimiento en comparación al desarrollo de aplicación hibrida, ya que con este desarrollo se puede acceder directamente a las librerías del sistema y las ejecuciones van directamente dirigidas al sistema del dispositivo móvil.
* Desarrollo de aplicación hibrida: este tipo de desarrollo ofrece un rendimiento más bajo, ya que las órdenes se ejecutaran primero en el navegador nativo del sistema y luego en el sistema del dispositivo móvil, por otra parte no se puede acceder a las librerías del sistema.

Interfaz de usuario: se concluyó lo siguiente (raona, 2013)

* Desarrollo de aplicación nativa: este tipo de desarrollo ofrece una gran guía de diseño de parte de cada plataforma, es decir, iOS y Android ofrece una línea de librería para el diseño de pantallas.
* Desarrollo de aplicación hibrida: en este tipo desarrollo, la aplicación se adaptará a una sola línea de diseño, por consecuencia, hay que realizar un buen diseño para que la aplicación sea aceptada por Android y iOS.

Acceso al hardware del dispositivo: se concluyó lo siguiente (appio, 2017):

* Desarrollo de aplicación nativa: este tipo de desarrollo ofrece una gran accesibilidad a los recursos dado que se pueden utilizar las librerías del sistema.
* Desarrollo de aplicación hibrida: este tipo de desarrollo ofrece una baja accesibilidad a los recursos del sistema dado que no cuenta con las librerías del sistema, sin embargo, mediante la instalación de algunos plugins se puede obtener acceso a algunos recursos del dispositivo.

Curva de aprendizaje: se concluyó lo siguiente (innovaage, 2014)

* Desarrollo de aplicación nativa: en este tipo de desarrollo la curva de aprendizaje es fuerte, dado que el desarrollador debe programar una versión para cada sistema operativo.
* Desarrollo de aplicación móvil: en este tipo de desarrollo la curva de aprendizaje es ligera, dado que el desarrollo es orientado a web, es decir, el desarrollador tendrá que programar una sola versión de la aplicación.

Rapidez de desarrollo: se concluyó lo siguiente (raona, 2013).

* Desarrollo de aplicación nativa: en este tipo de desarrollo la rapidez de desarrollo es baja, ya que se deben programar varias versiones de la aplicación.
* Desarrollo de aplicación hibrida: en este tipo de desarrollo la rapidez es alta, ya que solo se debe programar una versión de la aplicación, además, el desarrollo será más rápido si el equipo cuenta con conocimientos del desarrollo web.

Comunidad desarrolladora**:** Ambos tipos de desarrollo cuentan con una gran comunidad desarrolladora en el sitio web *“https://stackoverflow.com/”,* una simple búsqueda en este portal sobre cada tipo de desarrollo nos arroja más de 40.000 resultados.

Fácilmente escalable: se concluyó lo siguiente (Consultec, 2017)

* Desarrollo de aplicación nativa: en este tipo de desarrollo es más complejo agregar nuevas funcionalidades ya que se debe estudiar para cada versión de la aplicación las posibilidades de agregar nuevas funciones.
* Desarrollo de aplicación hibrida: en este tipo de desarrollo resulta más fácil agregar nuevas funcionalidad, ya que se le deben agregar a una sola versión de la aplicación y el estudio de las posibilidades de agregar nuevas funcionalidades se realizaran sobre una sola versión.

**Integración con el negocio**: ambos tipos de desarrollos cuentan con librerías para realizar llamadas al servicio RESTFul del sistema V-Track.

**Licenciamiento**: se define como la autorización que sede el autor a otros programadores para utilizar los frameworks o lenguajes de programación. Ambos tipos de desarrollo son de código abierto.

## Apéndice F – Historias de usuario

**Historia #1 – Consulta de información general de pedidos de usuario.**

Como usuario del sistema V-Track, quiero consultar todos los pedidos de mi propiedad existentes en la Planta Distribuidora de Combustible.

**Condiciones de satisfacción:**

* Una vez realizada la autenticación en la aplicación, la aplicación se redirigirá a una ventana donde se muestre información general de todos mis pedidos existentes en la planta distribuidora de combustible. De cada pedido se mostrara el código, la fecha de solicitud y el estado.
* En el menú principal de la aplicación, se podrá seleccionar la opción “Pedidos” y la aplicación se redirigirá a una ventana donde se muestre información general de todos mis pedidos existentes en la planta distribuidora de combustible. De cada pedido se mostrara el código, la fecha de solicitud y el estado

**Historia#2 – Consulta de información detallada de pedido en específico.**

Como usuario del sistema V-Track, quiero consultar la información detallada de un pedido en específico.

**Condiciones de satisfacción:**

* En la opción de “Pedidos”, al dar click en la información general de uno de los pedidos, la aplicación se redirigirá a una ventana donde se mostrará información detallada del pedido seleccionado. Del pedido se deberá mostrar el código, el estado, la fecha de solicitud, la cola, el cabezote, el chofer, la hora de inicio de llenado, la hora estimada de finalización de llenado, los eventos sucedidos durante el llenado del pedido y si el pedido ya finalizo su llenado, la hora de finalización.

**Detalle adicional**

* En la aplicación web, la pantalla de la opción “Pedidos”, deberá tener un buscador para poder filtrar los pedidos por código.

**Historia#3 – Solicitud de pedido de combustible**

Como usuario del sistema V-Track, quiero solicitar un pedido de combustible a la planta distribuidora.

**Condiciones de satisfacción:**

* Al seleccionar la pantalla solicitar pedido, la aplicación se deberá redirigir a una pantalla donde el usuario pueda describir una solicitud de combustible.
* Luego de que el usuario llene el formulario sobre la descripción de su pedido, el cliente deberá recibir en su correo, un mensaje con un código de liberación de pedido.
* El cliente en la opción de “Solicitar pedido”, podrá ingresar el código de liberación de pedido para que posteriormente el sistema V-TAS realice los procedimientos correspondientes.

1. **MVVM**: “es un sucesor de otro patrón bien conocido y exitoso como es el **Model View Controller** (MVC).” (Piriz, 2010) [↑](#footnote-ref-1)
2. **RESTful** : “Es un estilo arquitectónico que especifica restricciones” (Oracle, 2013) [↑](#footnote-ref-2)
3. **Tarjetas RFID**: “son tarjetas chip en las que la transacción electrónica se efectúa a través de comunicación radiofrecuencia” (a3m, 2014) [↑](#footnote-ref-3)
4. **Framework:** es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado (EcuRed, 2009) [↑](#footnote-ref-4)
5. **API** : “La API Java es una [interfaz de programación de aplicaciones](https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones) (API, por sus siglas del [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s): *Application Programming Interface*) provista por los creadores del [lenguaje de programación Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java), que da a los programadores los medios para desarrollar aplicaciones Java.” (Wikipedia, 2017) [↑](#footnote-ref-5)
6. **URI** : “sirve para identificar recursos en Internet, tiene un formato estándar definido y su propósito es permitir interacción entre recursos disponibles en Internet” (Castro, 2016) [↑](#footnote-ref-6)
7. Interceptores : nos permiten realizar alguna acción antes de que los métodos sean llamados (Marquez, 2014) [↑](#footnote-ref-7)
8. **Cola**: placa de vehículo cisterna que lleva un pedido. [↑](#footnote-ref-8)
9. **Cabezote**: placa del vehículo que conduce o lleva al vehículo cisterna que lleva un pedido. [↑](#footnote-ref-9)
10. **Responsive Web Design**: “es la técnica que nos permite crear sitios adaptables a las condiciones del ordenador o dispositivo desde donde se van a acceder, sobre todo en lo que tiene relación con la pantalla del sistema donde se están visualizando.” (Alvarez, Qué es Responsive Web Design, 2015) [↑](#footnote-ref-10)
11. Firebase Cloud Messaging: es una solución multiplataforma que te permite enviar, de forma gratuita y segura, mensajes y notificaciones (Firebase, 2015). [↑](#footnote-ref-11)
12. **VirtualBox:** es una potente solución de virtualización de sistemas operativo de 32 y 64 bits, tanto para entornos corporativos como para entornos domésticos. [↑](#footnote-ref-12)