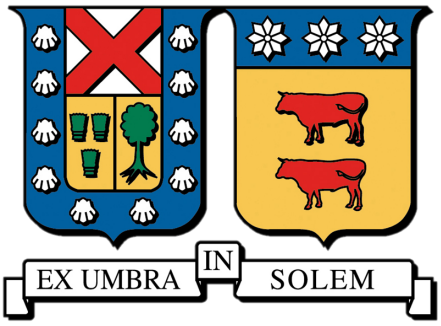
2020-09

DESARROLLO DE UN SISTEMA DE SEGUIMIENTO DE CARGAMENTOS PORTUARI

ARANCIBIA OJEDA, EDUARDO IGNACIO

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



"DESARROLLO DE UN SISTEMA DE SEGUIMIENTO DE CARGAMENTOS PORTUARIOS"

EDUARDO IGNACIO ARANCIBIA OJEDA
GERSON EDUARDO PINCHEIRA VICENCIO

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL TELEMÁTICO

PROFESOR GUÍA1: **MOHAMED ABDELHAMID**

PROFESOR GUÍA2: **NICOLÁS JARA**

SEPTIEMBRE 2020

Agradecimientos Eduardo

A todos los que me han acompañado en este viaje, Familia, Amigos y Compañeros, sin dudas muchas personas que me han acompañado y ayudado a formar la persona que soy hoy en día, si no fuera por ellos, nada de esto seria posible.

En especial a mi *Padre*, por darme las facilidades, comodidades para estudiar y así obtener mi título de Ingeniero, a mi *Abueli* por brindarme su cariño incondicional y su eterna preocupación, a mi polola *Maju* por su gran contención, amor, apoyo y ánimos en especial al final de esta etapa universitaria y por último a mi *compa de cuatro patas, Fox* Q.E.P.D, por acompañarme en las largas noches de estudio todos estos años y enseñandome una de las cosas más difíciles de la vida, el amor desinteresado.

Eduardo Ignacio Arancibia Ojeda

Agradecimientos Gerson

Para nadie es un camino fácil llegar a este punto. Quisiera agradecer a muchas personas que me acompañaron a lo largo de este camino, gracias por el apoyo, la compañia y el creer en mi en todo momento. Principalmente quisiera agradecerle a mi familia que de alguna forma siempre me apoyo y estuvieron presentes, a mi madre y a abuela sin las cuales no podría estar donde estoy ahora.

A todos mis amigos de la universidad con los que compartí en algún momento, ya sea carreteando , estudiando, aguantando estrés o en alguna actividad distinta. A todos mis amigos fuera de la universidad que siempre creyeron en mi, los cuales comprendían cada momento que no pude estar presente por estar cumpliendo esta meta en mi vida.

También me gustaría agradecer a algunos profesores con los cuales aprendí mucho y realmente son profesionales dignos de admirar, en este punto quiero agradecer principalmente a mi profesor guía, espero que sigas siendo tan buen profe como lo eres ahora.

Podría seguir agradeciendo mucho mas, pero en general les agradezco a todos los que estuvieron presentes en este camino de alguna o otra forma, ya que todos de alguna manera ayudaron a que llegara a este punto.

A todos muchas gracias

Resumen

En la actualidad se mueven millones de toneladas de cargas entre puertos a nivel mundial, transportándose una amplia gama de productos desde distintos países. Según el estudio elaborado por *ProChile* [1] en el año 2018, Chile lidero la exportaciones en 29 categorías de productos a nivel mundial. Por lo cual este tema se vuelve de suma importancia para el desarrollo económico del país.

En este trabajo de título se investiga sobre una manera eficiente de dar seguimiento a los cientos de cargamentos que salen día a día desde los puertos del país, de modo que las empresas tengan un mejor control de sus productos y se puedan anteponer a cualquier inconveniente que fuera a ocurrir antes de que estos lleguen a su cliente.

Se propone la creación de una aplicación web que guarde registro y actualice el estado de todos los cargamentos que estén actualmente en proceso de transporte, principalmente, nos centraremos en los contenedores que se transportan en vía marítima ya que son estos últimos los que presentan un mayor desafió a la hora de poder darle un seguimiento preciso.

Es muy difícil que una empresa que exporte productos utilice la misma naviera para el transporte de sus envíos, ya sea por costos, destino final, tiempos de embarque, etc. Es por esto que la idea principal de la aplicación es que el cliente pueda registrar todos los datos principales de cada uno de sus contenedores, de esta forma la aplicación puede interactuar con la información entregada por cada naviera , dando seguimiento al contenedor en un solo lugar. La aplicación web constará de ciertos requisitos funcionales, tales como: sistema de alarmas al cambiar los tiempos de arribo, visualización en tiempo real de los cargamentos, despliegue informativo sobre los contenedores, entre otras.

El sistema de alarma, permite notificar al operador el estado del tiempo de llegada del contenedor, permitiendo poder tomar alguna acción en el caso de ser necesario.

Poder visualizar gráficamente donde se encuentran espacialmente los contenedores, permite identificar cual es la última posición del contenedor, en caso que se extravíe o este se quede varado en algún puerto.

Asimismo, la posibilidad de identificar todos los contenedores que se poseen a través de una tabla, permite conocer información relativa a este, como a la naviera a la que pertenece, modelo del buque que lo transporta, fecha estimada de arribo a puerto, entre otras.

keywords: ETA (Estimated Time of Arrival), ETD (Estimated time of Departure), naviera, seguimiento de contenedores, buques, aplicación web .

Abstract

Currently, millions of tons of cargoes are moved between ports worldwide, transporting a wide range of products from different countries. According to the study elaborated by ProChile in the year 2018, Chile led the exports in 29 categories of products at world-wide level. Therefore, this issue becomes very important for the economic development of the country.

In this title work, we investigate an efficient way to track the hundreds of shipments that leave the country's ports every day, so that companies have better control of their products and can anticipate any inconvenience that might occur before they reach their customers.

We propose the creation of a web application that keeps records and updates the status of all shipments that are currently in the process of transport, Mainly, we will focus on the containers that are transported by sea as it is the latter that present a greater challenge to be able to give a precise follow up.

It is very difficult for a company that exports products to use the same shipping company for the transportation of its shipments, whether it is for cost, final destination, shipping times, etc. That is why the main idea of the application is that the customer can register all the main data of each of its containers, In this way, the application can interact with the information delivered by each shipping company, following the container in one place. The web application will have certain functional requirements, such as: alarm system when changing the arrival times, real time visualization of the shipments, information display about the containers, among others.

The alarm system allows the operator to be notified of the time of arrival of the container and allowing him to take action if necessary.

To be able to visualize graphically where the containers are spatially located, allows to identify which is the last position of the container, in case it gets lost or stranded in some port.

Likewise, the possibility of identifying all the containers that you have through a table, allows you to know information related to it, such as the shipping company to which it belongs, model of the ship that transports it, estimated date of arrival to port, among others.

Glosario

Agrosuper: Es un holding de empresas alimentarias chilenas, dedicadas particularmente a la producción, distribución y comercialización de alimentos frescos y congelados de cerdo, aves (pollos y pavos), salmones y productos procesados (cecinas).

Contenedores: Es un recipiente de carga para el transporte terrestre, multimodal y marítimo o fluvial.

Aplicación web: Es una herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

Requisito funcional: Define una función del sistema de software o sus componentes, que es descrita como un conjunto de entradas, comportamientos y salidas.

Operario: Persona que tiene un oficio de tipo manual, en este caso observar el estado de arribo de los contenedores.

Visualización: Hacer visible por algún procedimiento o dispositivo lo que normalmente no se puede ver a simple vista.

Estiba: Distribución y colocación adecuada de la carga en una embarcación.

Aduana: Oficina pública del Estado, situada en las fronteras, puertos o aeropuertos, donde se registran los géneros y mercancías que se importan o exportan y se cobran los derechos que adeudan según el arancel correspondiente.

Arribar: Llegada de una embarcación al puerto que es su destino o en el que se puede refugiar.

Departamento de Logística: Su función es la planificación y gestión del flujo de materiales de la manera más eficaz entre los proveedores y clientes finales, incluyendo la creación e implementación de sistemas de control y mejora.

Cadena de Suministro: Es el nombre que se le otorga a todos los pasos involucrados en la preparación y distribución de un elemento para su venta.

Packing List: Es una lista con una relación de contenidos del contenedor que completa la información de la factura y debe ser emitida por la persona que realiza el envío, el remitente.

Draft: Es un borrador digital, el cual es un documento que se debe enviar para confirmar todo el contenido del cargamento.

Bill of Landing: Es un documento que establece las reglas de la relación contractual entre el cargador, el destinatario y el transportista, dando confianza a cada parte respecto al comportamiento de las otras.

Free On Board: Es una cláusula de comercio internacional que se utiliza para operaciones de compraventa, en las que el transporte de la mercancía se realiza por barco, ya sea marítimo o fluvial.

Less Container Load: Cuando un cargamento es Less Container Load, implica que la mercancía ocupa menos que el espacio total de un contenedor completo, lo que significa que el contenedor será compartido y en él viajarán distintas mercancías de varios proveedores.

Full Container Load: significa simplemente que el vendedor o expedidor es el responsable de llenar el contenedor y de sufragar los correspondientes gastos.

RFID: Es un método de almacenamiento y recuperación de datos que por medio de una señal de radiofrecuencia, puede transmitir los datos que contiene en su interior.

GPS: Es un sistema que permite determinar en toda la Tierra la posición de cualquier objeto o Ser, con una precisión de hasta centímetros, aunque lo habitual son unos pocos metros de precisión.

GSM: Es un tipo de red que se utiliza para la transmisión móvil de voz y datos.

GNSS: Es una constelación de satélites que transmite rangos de señales utilizados para el posicionamiento y localización en cualquier parte del globo terrestre, ya sea en tierra, mar o aire.

FM: Es una técnica para transmitir información a través de una onda, en la cual se varia su frecuencia.

Machine Learning: Es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente, donde aprenden en el contexto de identificar patrones complejos en millones de datos.

Front-end: Corresponde a la capa de la vista en la aplicación web, basado en el esque-

ma Modelo-Vista-Controlador.

HTML: Lenguaje de marcado de hipertexto, que le permite al usuario crear y estructurar secciones, párrafos, encabezados, enlaces y elementos de cita en bloque para páginas web y aplicaciones.

SVG: Es un tipo de formato el cual especifica un gráfico vectorial con gran facilidad para escalar en diversos tamaños de pantallas.

Back-end: Capa de acceso a los datos, donde se desarrolla el motor de la aplicación.

HTTP: Es un protocolo de acceso para las páginas web a través de Internet.

Base de datos: Es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite.

Flujo de la aplicación: Es la forma en la cual van cambiando las vistas dentro de la página web.

Flujo de datos: Es el modo en el cual los datos se van transmitiendo dependiendo de las diferentes acciones del usuario.

Single Page Applications: Es una aplicación web de una sola página, con el propósito de dar una experiencia más fluida a los usuarios.

Login: Es el proceso que controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas a este.

API: Es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software.

API Fetch: Proporciona una interfaz para recuperar recursos (incluso a través de la red).

JSON: Es un formato de texto sencillo para el intercambio de datos.

JWT:Se utiliza para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de privilegios del mismo.

Acrónimos

ETA: Estimated Time of Arrival

ETD: Estimated Time of Departure

AIS: Automatic Identification System

JSON: JavaScript Object Notation

JWT: JSON Web Token

MVC: Model View Controller

B/L: Bill of Landing

FOB: Free On Board

LCL: Less Container Load

FCL: Full Container Load

API: Application Programming Interface

IoT: Internet of Things

GPS: Global Positioning System

GSM: Global System for Mobile Communications

GNSS: Global Navigation Satellite System

FM: Frequency Modulation

HTML: HyperText Markup Language

HTTP: Hypertext Transfer Protocol

SVG: Scalable Vector Graphics

SIM: Subscriber Identity Module

ML: Machine Learning

Índice de figuras

Índice general

Capítulo 1

Introducción

Según *ProChile* [1], Chile lideró la exportación de 29 categorías de productos a nivel mundial. Específicamente en cuanto a las exportaciones no tradicionales nuestro país destacó como el mayor exportador de un largo listado de productos, entre ellos: uvas frescas, cerezas frescas, filetes de salmones frescos, arándanos frescos, ciruelas frescas y deshidratadas, filetes de truchas congelados, erizos de mar y algas de uso industrial. Por otra parte, el sector de los alimentos, específicamente el de las carnes y pescados tuvo durante el 2018 un incremento en las exportaciones en el sector de las carnes (cerdo, pollo, pavos, bovinos y ovinos) de un 24 % mayor respecto al 2017, superando la barrera de US$ 1.000 millones y, por otra parte, la industria salmonicultora nacional obtuvo retornos históricos por un total de US$ 5.157 millones, cifra que representa un aumento de 11,4 % en contraste con el resultado consolidado del año 2017, según información del *Banco Central de Chile* [2].

Sin embargo, estas actividades de exportación antes mencionadas presentan un alto riesgo durante el proceso de transporte necesario para que el cargamento pueda llegar a su destino. Entre tales riesgos, se encuentra el posible retraso en la fecha de llegada del cargamento, debido a los tiempos de demora entre puertos por el número de escalas, retrasos en transbordos, excesos de estadía y tramitación de la carga, inspecciones o paradas y la frecuencia del servicio, entre otros motivos [3]. Lo cual representa un riesgo de pérdida de la mercancía para las empresas si estas sobrepasan los tiempos de vencimiento, provocando un aumento de los costos de producción y, en consecuencia, deficiencia del servicio, del negocio y problemas en la fidelización de los clientes.

Tal problemática es la que enfrenta la empresa *Agrosuper* [4], que se estudiará y cuyo desafío se aborda en el contexto del Programa de Memorias Multidisciplinarias de la Universidad Técnica Federico Santa María, año 2019.

En la actualidad, la empresa en cuestión posee un sistema de seguimiento a través de la plataforma *Super Tracking* [5] que le resulta poco eficiente y que no le permite prevenir futuras pérdidas, por retrasos en Aduana al no estar al tanto del tiempo de la llegada del cargamento enviado al puerto de destino. Considerando lo descrito anteriormente, el estudio que se desarrollará a continuación aborda el punto de vista comercial del problema que enfrenta el equipo del Programa de Memorias Multidisciplinarias, y quienes describen la solución a través de una plataforma web, llamada *SMART TRACKING*, la cual ofrece una solución global y responde a las necesidades de *Agrosuper* y su departamento de logística, como también a otros potenciales clientes que enfrentan necesidades similares.

1.1. Objetivos

1.1.1. Objetivo General

Desarrollar una solución plausible para la gestión y supervisión de cargamentos transportados por vía marítima, la cual permitiría al usuario informar y actualizar la posición y tiempo de arribo de cada contenedor registrado.

1.1.2. Objetivos Específicos

- Investigar distintos mecanismos para dar seguimiento al transporte de contenedores vía marítima en la actualidad.
- Definir funcionalidades claves para la aplicación Web, esto para todos sus distintos perfiles.
- Investigar y definir tecnologías en concordancia a funcionalidades requeridas y compatibilidad entre ellas.
- Diseñar el flujo de la aplicación y base de datos.
- Desarrollo backend de la plataforma para la conexión de las distintas funcionalidades.
- Desarrollo frontend para la plataforma para el envió y recepción de información desde backend.

1.2. Estructura

Este documento se estructura de la siguiente forma: en el capítulo 2, se hace una revisión de todo el proceso de logística para llevar acabo el transporte del contenedor; en el capítulo 3, se presentan diversos estudios sobre el seguimiento de contenedores tanto por aire, tierra y mar, donde de este último se muestras las principales aplicaciones web que realizan funciones similares a la de esta memoria; en el capítulo 4 se presentan diversas herramientas para el desarrollo de aplicaciones web, comparaciones, ventajas y desventajas, además los diagramas, arquitectura y componentes que conforman *SMART TRACKING*; mientras que en el capítulo 5, se describe el desarrollo final de la aplicación. Por último, en el capítulo 6, se mencionan las conclusiones y se detallan las futuras mejoras que se pueden llegar a implementar a la solución.

Capítulo 2

Marco Teórico

El seguimiento marítimo de los activos de hoy es crucial para una cadena de suministro global eficiente. Un aspecto particular de la industria del transporte es su alto grado de desfragmentación vertical, es decir, se realizan las operaciones de: estiba y desestiba, carga y descarga, almacenamiento en tránsito en explanada o tinglado, el transporte y la recepción, y evacuación de la mercancía, como entidades separadas. Por lo tanto, existen mucho puntos de inflexión donde al contenedor podría tener algún problema y sufrir retrasos en su tiempo estimado de arribo. Como resultado, las soluciones que proporcionan la capacidad de seguimiento de contenedores deben ser autosuficientes en términos de localización y comunicación [6] .

2.1. Estiba del contenedor marítimo

La carga y estiba de la mercancía en el contenedor marítimo es responsabilidad de quien la realiza, y debe considerarse que en el tránsito internacional el contenedor marítimo sufrirá movimientos violentos y cambios climáticos bruscos. Algunas recomendaciones e instrucciones generales para realizar este proceso correctamente son las siguientes:

2.1.1. Recomendaciones para el correcto estibamiento

- Antes de cargar la mercancía en el contenedor se debe inspeccionar el buen estado del mismo.
- Se deben cargar y estibar las cargas más ligeras sobre las más pesadas.
- La distribución de los pesos debe hacerse de forma uniforme tratando de situar el centro de gravedad lo más bajo y centrado posible.
- No se deben mezclar cargas secas con húmedas (temperatura controlada) y se debe cuidar la mezcla de olores.

Donde se puede evidenciar, que en esta etapa la posibilidad de tener algún inconveniente con el cargamento es alta, generando una posibilidad de que se pierda el contenedor.

2.2. Proceso de exportación

El proceso de gestión y contratación del transporte variará sustancialmente en función de múltiples parámetros: naturaleza de la carga o de la mercancía, cliente, mercado, confianza y seguridad en el cobro. A la llegada del contenedor al almacén (tras su revisión y descarte si presenta anomalías) se carga la mercancía (este proceso se suele fotografiar y se envían las imágenes por correo electrónico al cliente) y se entrega un juego de facturas junto con la mercancía (varios ejemplares junto al packing list o lista de contenido, que se requieren para el despacho de exportación). En caso de realizar un servicio de grupaje/consolidación, se desarrolla una operación similar a la expuesta hasta ahora, pero en vez de recibir el contenedor vacío en la empresa para cargar la mercancía, se contrata el transporte nacional hasta el almacén transitorio donde se va a efectuar la carga de la mercancía junto a los envíos de otros exportadores en un solo contenedor para compartir hasta destino, donde se desconsolidará. El contenedor se posiciona en puerto, se manipula y se tramita su exportación y servicios auxiliares hasta su embarque en el buque.

Antes de la salida del buque y cuando el contenedor ya suele estar en el terminal, el transitario (o naviera, si se ha contratado directamente con ella) debe remitir por correo electrónico un borrador (draft) digital o propuesta del *B/L (Bill of Lading)* para comprobar si todos los datos del envío son correctos y si falta alguno o se debe rectificar algo (se puede remitir por correo electrónico dicho borrador o copia del B/L al importador para que también dé su conformidad respecto a cualquier detalle relativo a los datos del B/L que se deban rectificar a efectos del crédito documentario, aduanas, etc).

2.3. Proceso de importación

Una vez que el proveedor ha fabricado la mercancía, comunica la fecha de entrega del pedido y se le comunica al transitario elegido, proporsionándole los datos del proveedor para que lo contacte (su corresponsal en el país del proveedor) y proceda a la recogida del envío en sus instalaciones (envío del contenedor) hasta el puerto de exportación. Estos costos los paga el vendedor, incluido el despacho de exportación, pues se supone que es venta en condiciones, denominada *FOB (Free On Board)*, hasta el puerto de exportación. Desde ahí los costos son del comprador: flete, recargos y costos en el puerto de importación, despacho de importación y costos hasta el almacén. Es posible que el proveedor haya solicitado al inicio del proceso los datos de su transitario habitual para poder dar precio con base en los costos que el propio proveedor tendrá que asumir.

Una vez que el contenedor se ha embarcado en el buque en el puerto de exportación, el proveedor consigue del transitario (el corresponsal del importador) toda la documentación que necesita para presentarla en su banco en el caso de crédito documentario: B/L [7] que funciona como contrato de transporte, packing list, facturas, etc. Si todo está conforme, podrá cobrar (a la vista o a plazo). El banco del proveedor remite (por courier) la documentación al importador, quien la revisa, carga en cuenta el dinero y la entrega para que se haga llegar al transitario del importador, y una vez llegado el buque, se pueda despachar el contenedor de importación. En caso de *LCL (Less Container Load)*, el proceso es similar, pero con las fases de consolidación y desconsolidación vistas en el proceso de exportación.

2.4. Seguimiento

Durante el tránsito internacional de la mercancía en el contenedor marítimo, se efectúa su seguimiento y se hace principalmente mediante el transitario (con apoyo en la web de las navieras y de los puertos). Es de esperar que no haya ninguna incidencia (trasbordo en puerto intermedio, congestión en puerto, etc) y que se cumpla el tiempo de tránsito (transit time o ETA) indicado con las condiciones marítimas aceptables.

Una vez llegado el contenedor a puerto y obtenida del banco la documentación requerida para su despacho (al menos el B/L original, factura y packing list), se le hace llegar al transitario, quien procede a su despacho de importación y gestión para su salida del terminal. Tras estas operaciones, se indicará la fecha en que llegará el contenedor a las instalaciones. Una vez sacado del terminal, se procede a su desconsolidación y se acuerda con el transitario la gestión del transporte del envío en camión normal hasta el almacén del importador. Si surgen problemas en el despacho aduanero (inspección documental y física), el transitario informa del proceso y su posible resolución.
Es específicamente, en este proceso donde *SMART TRACKING* ayuda a resolver este dolor, puesto que dentro de la plataforma actual que posee Agrosuper, no se realizan las actualizaciones de los ETA de sus contenedores, al no utilizar las APIs que proporcionan las navieras.

2.5. Aspectos Comerciales

La empresa que contrata el transporte marítimo en contenedor, lo puede hacer básicamente mediante dos canales de comercialización: contratando directamente con la naviera (con sus oficinas comerciales o con sus agentes consignatarios) o a través de intermediarios (transitarios y otros operadores logísticos). Las navieras son los porteadores del transporte marítimo, empresas que explotan comercialmente sus buques realizando servicios todo tipo.

Los transitarios *(freight forwarders)* son operadores de transporte marítimo que, sin disponer de buques, intermedian en la contratación entre los cargadores y las navieras que ejecutan el transporte. Ofrecen una función comercial reservando y alquilando espacios *(slot charter)*, cuya capacidad de carga y transporte ponen a disposición de sus clientes finales. Desde el punto de vista de los cargadores, su oferta se valora como un servicio más adaptado a las necesidades del cliente que el de las navieras, ya que pueden desarrollar toda la cadena logística internacional en colaboración con sus corresponsales en otros países.

La contratación del transporte en contenedores marítimos se puede llevar a cabo en función de las características del envío de dos grandes formas o grupaje:

- FCL *(Full Container Load)*
- LCL *(Less Container Load)*

2.5.1. Full Container Load

Se recurre a esta fórmula cuando el envío a transportar en un contenedor ocupa un espacio suficiente como para llenarlo por completo o su mayor parte. Mediante este sistema se eliminan los riesgos de manipular la mercancía a lo largo de su transporte y de compartir el contenedor con otras existencia, pues se carga en origen, se precinta y no se descarga hasta destino.

2.5.2. Less Container Load

Este sistema, también conocido como régimen de consolidación, es adecuado para envíos de reducido tamaño (2 palés, 4 cajas, 16 sacos, 7 bultos, etc) que no alcanzan a completar la capacidad de carga del contenedor. En estos casos, para reducir su costo de transporte, las mercancías van a compartir el contenedor con otros de un origen y destino común.

2.6. Precio del Transporte

Para obtener el precio del transporte, se deberá solicitar al porteador (naviera o transitario) la cotización de dicho transporte. La solicitud debe indicar las características básicas del tipo de transporte que se quiere contratar: puertos de origen y de destino, tipo y cantidad de contenedores necesarios, descripción de la mercancía, y modalidad de servicio, por ejemplo, si se desea contratar desde el almacén del vendedor hasta el puerto de destino *(door-pier)*, desde dicho almacén hasta puerto de exportación o en condiciones *CFR/CIF (Cost and Freight/Cost Insurance and Freight)* o *CPT/CIP (Carriage Paid To/Carriage and Insurance Paid To)*.

Toda cotización puede incluir hasta tres grupos de elementos que serían:

- Costos previos al embarque.
- Flete marítimo entre puertos.
- Costos en el país de destino.

2.6.1. Costos previos al embarque

Son aquellos gastos previos al embarque del contenedor marítimo en el buque. Los más habituales son:

- **Transporte por carretera desde el almacén del exportador al puerto:** Se trata de un transporte que suele resultar mas costoso con relación al costo total de la cotización.
- **Recepción y manipulación de la carga en puerto:** Se suele aglutinar dentro de lo que se conoce como "costos por manipulación en terminal" o *THC (Terminal Handling Charge)*. Costo asociado a la manipulación del contenedor en la terminal desde su llegada hasta que se carga a bordo del buque.
- **Tarifa de Ia mercancía o muellaje:** Tasa que paga la mercancía y el contenedor por el uso de las instalaciones del puerto.
- **Expedición del conocimiento de embarque:** Copias adicionales que pueden implicar costos.
- **Despacho aduanero:** En exportaciones en contenedor, costos por la ejecución de las formalidades aduaneras de exportación, que en su caso solicite el cargador.
- **Recargo por seguridad :** Costos del puerto derivados de la ejecución de medida de seguridad
- **Otros:** Costos por certificaciones, almacenaje y consolidación, limpieza del contenedor, comprobación de precinto, etc.

2.6.2. Flete marítimo entre puertos y sus recargos

El flete básico (también conocido como *Ocean Freight*) es el precio del transporte marítimo del contenedor puerto a puerto, es decir, muelle a muelle. Entre los principales recargos, no normalizados y por tanto, sujetos al libre mercado y acuerdo entre las partes, se suelen encontrar los siguientes:

- **BAF *(Bunker Adjustment Factor)*:** Recargo por costo del combustible, para introducir correcciones al precio debido a la distorsión generada por la alta volatilidad del precio del petróleo.
- **CAE *(Currency Adjustment Factor)*:** Recargo por corrección de divisas, la cual depende de las cotizaciones que tenga la moneda local y la moneda con la que se establece el flete, que suele ser en dólares.
- **Derecho de Obtención de Divisa:** Son los gastos y comisiones bancarias por la obtención de divisa para el pago del flete.
- **CS *(Congestion Surcharge)*:** Recargo por congestión en los puertos.
- **Collect Surcharge:** Recargo por cobrar el flete en destino. Si queremos que el flete lo pague el importador, suele ser aproximadamente un 3 % adicional al costo normal.

2.6.3. Costos en el país de destino

Principalmente, son los mismos que los costos previos, pero en el país donde se desembarcan los contenedores, donde se agregan costos como: recargos en el puerto de destino, acarreo en destino, despacho de importación, etc.

SMART TRACKING desea reducir los costos de muellaje, tasa que se paga por usar un espacio en el puerto junto a las instalaciones de este mismo. Además, de la manipulación que deben tener los contenedores críticos, los cuales en ocasiones pierden su contenido por interrupción en la cadena de frío.

Finalmente, en la Figura. 2.1 resume todo el proceso de envio de los contenedores, pasando por todos los pasos anteriormente descritos.

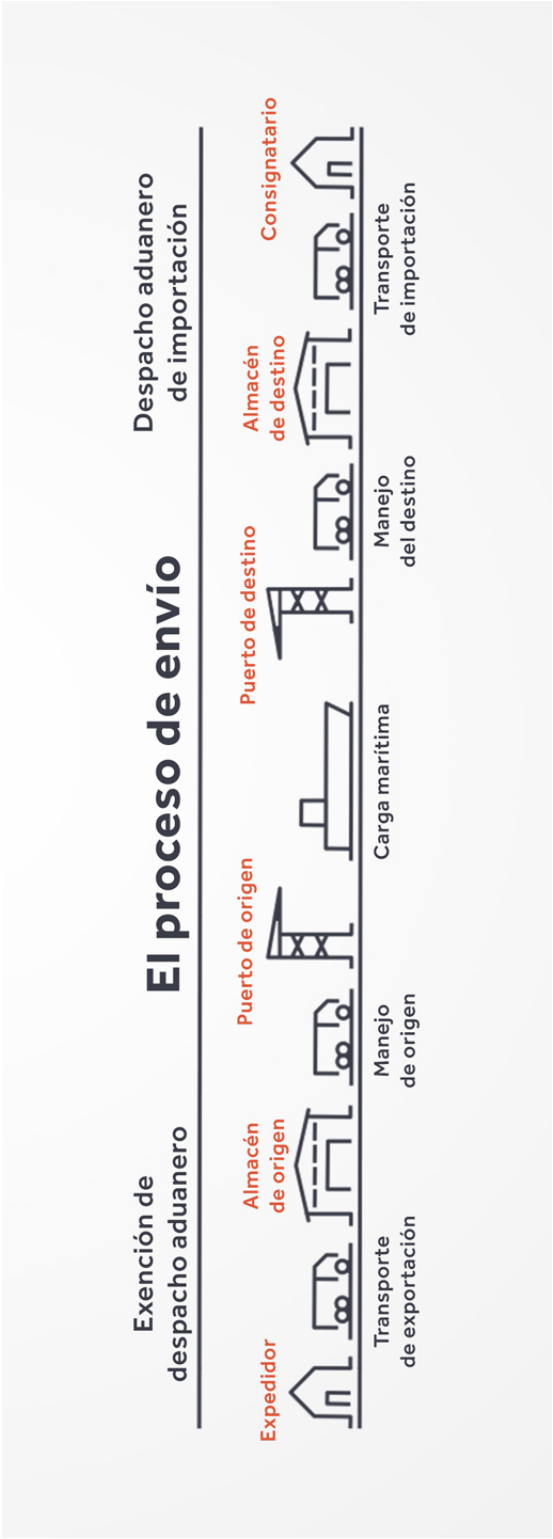


Figura 2.1: Resumen de la logística para el envío de un contenedor.

Capítulo 3

Estado del Arte

En el capítulo anterior se ha descrito en que consiste el proceso del seguimiento de un contenedor, los posibles problemas que puede tener y los costos que significa el retraso de la mercancía. En este capítulo, se exponen las diversas medidas para el envío de contenedores y los principales competidores en el mercado marítimo específicamente, que mediante una aplicación web desean resolver el problema de seguimiento, actualizando sus ETAs correspondientes.

El comercio mundial depende del envío eficiente de bienes, ya que el 90 % [8] de todos los bienes se envían en contenedores, la capacidad de realizar el seguimiento de la ubicación de dichos contenedores e inspeccionar de forma continua y remota su estado, ayuda tanto a las compañías navieras como a sus clientes. Debido a su éxito, la tecnología IoT se utiliza cada vez más para el seguimiento global de contenedores, es por ello que para cada tipo de transporte existen diversos estudios sobre seguimiento de contenedores, como los que se describirán a continuación.

3.1. Transporte de contenedores por vía aérea

Al momento de buscar estudios sobre seguimiento de cargas por vía aérea, se encontró el siguiente trabajo [9], el cual propone una solución IoT, la cual considera indispensable dispositivos inalámbricos que tengan capacidad de adquisición automática de datos, de tamaño pequeño, larga vida útil y capacidad de comunicación de largo alcance. Una de las formas de cumplir con estos requisitos es adoptar sistemas de identificación por radiofrecuencia (RFID) activos, dado que es más ventajosa que los RFID pasivos y permite un mayor rendimiento de lectura de datos en distancias más largas. También se realizaron pruebas en logística global para el seguimiento del transporte de contenedores marítimos/aéreos utilizando sistemas RFID activos. Los resultados de la prueba fueron que las etiquetas RFID activas tienen suficiente capacidad y bajo consumo de energía para poder soportar bien el transporte marítimo/aéreo y el servicio de logística.

Por otro lado, en el siguiente paper [10], se propone que el objetivo de un servicio de seguimiento de contenedores es lograr una mayor claridad durante el proceso de la cadena de suministro, la cual se encarga de la coordinación de las tareas a cumplir en todas las etapas de los bienes a transportar. También se debe realizar un directo seguimiento a todos los containers, debido a que pasan por múltiples lugares. Este documento presenta el esquema CTS (*Container Tracking Service*) que utiliza el satélite LEO (*Low Earth Orbital*) para mejorar la visibilidad global de los contenedores. El LEO recopila los datos periódicamente y los envía al servidor web y finalmente, al PDA (*Personal Digital Assistant*) de los clientes. Esto proporciona a los agentes una carga de información más sólida, como el estado de la puerta y el estado del contenedor.

Finalmente, el último documento [11] estudiado se concentra en el comercio electrónico, dándole un mayor énfasis al modelo de negocio en la Web. Un infomediario de bajo costo basado en la creación de una plataforma Web para la industria de carga aérea,

que ayuda a integrar a los proveedores de servicios de Air Cargo con sus clientes, así mejorando la productividad de la cadena logística. Los clientes pueden acceder a la información de vuelo de los transportistas de carga a través de una aplicación web de una manera mucho más simplificada, por lo que se puede tener un mayor control en la logística para operaciones posteriores recibido el contenedor. Por último se desarrolló un prototipo funcional de sistema para demostrar su capacidad para aceptar solicitudes de navegadores generales, recuperar datos de diferentes plataformas y presentar datos en un formato coherente y personalizado.

En conclusión, el principal requerimiento que se necesita para el seguimiento de cargas, es un enlace de comunicación para poder enviar los datos, donde el IoT parece la mejor solución para ello, ya sea por medio de una conexión satelital o sistemas de RFID activos, pero en el último documento se abarca la idea de una aplicación web, la cual se centra en la recuperación de datos por medio de diferentes plataformas para entregárselas a los clientes de forma fácil y coherente para la toma de decisiones.

3.2. Transporte de contenedores por vía terrestre

Se plantea en el siguiente documento [12] que el seguimiento de vehículos cada vez se está volviendo más esencial en el mundo actual, sobretodo en el mercado logístico, puesto que la demora en la entrega y retiro tardío de los contenedores genera altas perdidas de dinero. Hacer un seguimiento de los vehículos que transportan la carga ayuda a administrar los recursos de una manera más eficiente, aumentando las ganancias a la empresa, además, en este trabajo se propone un seguimiento del vehículo por GPS. Este sistema plantea un almacenamiento con los detalles completos sobre el viaje del vehículo, como la ruta, la distancia recorrida, el control del conductor sobre el vehículo y las precauciones ocurridas, los cuales se podrían utilizar posteriormente para aplicar algoritmos de *Machine Learning* y realizar predicciones para futuras decisiones.

De la misma manera, en este otro paper [13] se presenta un sistema en tiempo real para monitorear la seguridad de los camiones que transportan contenedores. El cual integra un sistema de monitoreo que desarrolla un algoritmo de rastreo de camiones y detección de incidentes que se basa en una combinación de datos de telemetría del vehículo con los datos GPS obtenidos y diversas aplicaciones encargada de la elección de la ruta del camión que lleva el contenedor, al igual que en la publicación anterior, propone un sistema usando GPS.

Una forma novedosa para actualizar los ETAs de forma correcta es el uso de *Machine Learning* para realizar una predicción sobre el tiempo en el que debería llegar el contenedor, en el siguiente documento [14] es aplicado a unos centros de distribución que conectan las corrientes de transporte a la cadena de suministro. La sincronización de carga de entrada y salida en un centro de distribución requiere tiempos de llegada precisos, para lograr esto se necesita un método confiable de predicción para obtener los tiempos de llegada, donde por medio de una revisión a la literatura, que consistía en encontrar los factores importantes para predecir la hora de llegada, el resultado de esta revisión son que la congestión, el clima, la hora del día e incidentes, son usados para la predicción del tiempo de viaje. Luego, a partir de lo anterior se espera obtener el tiempo de llegada, dado que este es la consecuencia del tiempo de viaje en combinación con el tiempo de salida, aunque la literatura hable solo sobre el tiempo de viaje, es aplicable el factor humano involucrado en la planificación del tiempo de salida que puede afectar al tiempo de llegada, pero los resultados determinaron que el poder predictivo no es tan alto como se esperaba; otros factores, como los factores humanos u organizativos, podrían influir en el tiempo de llegada, y se concluye que dichos factores organizativos deberían considerarse en futuros modelos predictivos.

Por lo que la mejor alternativa que se tiene al momento de realizar el seguimiento de los contenedores por tierra es por medio de GPS, pero notar que esta no es una muy

buena opción por la limitante que se tiene al no poseer una conexión estable a internet para realizar el envío de las coordenadas GPS. Sin embargo, hay ideas sobre aplicar métodos predictivos para poder tomar decisiones a partir de estos resultados.

3.3. Transporte de contenedores por vía marítima

Lo que se plantea en este documento [15], es que las empresas están poniendo más recursos en hacer que sus cadenas de suministro sean cada vez más eficientes. Desde la perspectiva de la cadena de suministro, los transportistas siempre buscan aumentar su rentabilidad, donde el seguimiento de contenedores les ayuda a reducir sus costos, les ahorra tiempo y les brinda un mejor servicio al cliente, y a la vez cada destinatario debe ser informado de la última posición y situación de su carga. Para esto se definen tres métodos distintos sobre el seguimiento de los contenedores en las navieras basadas en los satélites, el *Identificador de Radiofrecuencia* (RFID) y el *Sistema Global para Comunicaciones Móviles* (GSM), donde solo se consideraron los contenedores desde la primera etapa (liberación al remitente) hasta la última etapa (devolución por consignatario) en destino.

En este trabajo [16] se habla sobre que los sistemas actuales para el seguimiento de contenedores en todo el mundo utilizan *Sistemas de Navegación Global por Satélite* (GNSS) para determinar la posición geográfica de un contenedor y por medio de un módem satelital transmitir la información a un servidor central. Sin embargo, la comunicación resulta significativa en costos operativos y ambos servicios requieren un alto grado de línea de vista entre el contenedor y los satélites correspondientes, en particular este último requisito es difícil de cumplir una vez que los contenedores están apilados. Este documento explica el enfoque alternativo basado en el *Sistema de Identificación Automática* (AIS), que en los últimos años experimentó un auge global después de que varias compañías lanzaron satélites que pueden detectar la señal de forma remota des-

de el espacio, permitiendo la detección global de señales AIS. Este sistema tiene dos funciones, en primer lugar, la información AIS transmitida por las naves circundantes se usa junto con información adicional para calcular la posición de un contenedor, por lo que se evita el uso de un receptor GNSS; en segundo lugar, las funciones integradas en el protocolo de mensajes AIS se utilizan para transmitir la posición del contenedor de forma transparente, evitando interferencias con el caso de uso original. La cobertura global se habilita a través de la disponibilidad de numerosos satélites de teledetección dedicados a AIS, así como cargas auxiliares de AIS a bordo de satélites de comunicaciones.

Por otro lado, siguiendo con las soluciones IoT y de las telecomunicaciones, este paper [17] habla sobre que las tecnologías actuales de rastreo de contenedores basadas en GPS consumen muchos recursos, son costosas, requieren la línea de vista con el satélite y, a menudo, son demasiado grandes para ser cubiertos, por lo que propone un sistema de seguimiento en la señal de transmisión FM, que es una señal artificial alternativa que es razonablemente ubicable, proporcionando un espectro de frecuencia geográficamente único y es aproximadamente 100.000 veces más fuerte que un satélite GPS, permitiendo el desarrollo de un receptor FM en miniatura, de bajo costo que puede grabar espectros de frecuencia y compararlos con datos conocidos para rastrear el camino que el contenedor ha tomado. Como beneficio adicional de seguridad, lograron que las etiquetas también sean capaces de detectar aberturas en las puertas de los contenedores.

Cambiando de tecnología, en este documento [18] se habla que en la industria del transporte marítimo, hay ganancias sustanciales por un acertado tiempo estimado de llegada (ETA), donde velocidades relativamente bajas y otras influencias externas, causan grandes fluctuaciones en el tiempo de viaje de un barco. Este estudio se centra en predecir futuros tiempos de llegada de buques tanque, desarrollando diversos modelos de predicción con *Machine Learning*, siendo pioneros en usar redes neuronales para la

predicción ETA marítima. Los datos que se utilizan para el entrenamiento y evaluación de los modelos, son los datos emitidos por los AIS de los buques de cierto tamaño que se envían regularmente, dando como resultado que no se pueden predecir con exactitud los ETAs de las navieras, dejando a trabajo futuro la evaluación de nuevas variables para el modelo.

Siguiendo en la línea de *Machine Learning*, este trabajo [19] apunta a que una de las mayores preocupaciones en las operaciones es que se cumplan los trabajos en los tiempos indicados. La gestión del factor tiempo se ha convertido en un tema crucial en las operaciones de envío de línea de hoy, sin embargo, la puntualidad del buque se ve afectada por muchos factores, como el puerto y las condiciones del buque y los efectos de demoras. Como resultado, este documento desarrolló un modelo para analizar y predecir la puntualidad de llegada de un buque mediante el uso de una técnica híbrida de toma de decisiones, basada en algoritmos de predicción, donde los resultados obtenidos son bastante acertados, con errores de entre un 4 % y 7 % considerados razonables. Este modelo es capaz de ayudar a los operadores de transporte marítimo de línea a predecir la puntualidad de llegada de su embarcación a un puerto de escala en particular.

En síntesis, las soluciones para el seguimiento de contenedores se podrían separar en tres opciones distintas:

- Internet de las Cosas (IoT) basado en sistemas de telecomunicaciones.
- Técnicas de Aprendizaje de Maquinas.
- Uso de Sistemas de Identificación Automática (AIS).

Donde la alternativa de usar IoT queda descartada por no tener acceso a los contenedores, no contar con los recursos ni los permisos necesarios para poder intervenir los buques y navieras. Para el uso de Machine Learning se necesitaría de mucha data histórica para llevar acabo la solución y, además, tener acceso a los datos provenientes

del AIS para poder obtener una mejor predicción, pero los datos del AIS deben ser proporcionados por las navieras, dado que ellas son dueñas de los datos de sus embarcaciones. Por lo que como última opción, se recurre a la aplicación web, que utilizando las APIs que liberan las navieras, podemos actualizar los ETAs correspondientes a los contenedores que deseamos realizar el seguimiento y de esta forma no tener inconvenientes con la fecha incierta de la entrega a los clientes.

Algunas de las aplicaciones web que realizan esta labor son las siguientes:

3.3.1. Searates

Esta plataforma [20], al igual que SMART TRACKING, ofrece su producto a través de una página web con diversidades de planes y servicios, según cada perfil. El plan básico de "seguimiento de contenedores", permite determinar la posición actual de un contenedor dado en el mapa (por medio de monitoreo de datos AIS). También procesa los datos portuarios y calcula el tiempo de permanencia en el puerto de congestión, entregando informes y estadísticas. Además, les permite a los usuarios rastrear la ubicación del contenedor al especificar el número y la línea de envío. Este plan, puede ampliarse a un diseño personalizado junto con la optimización del tiempo de transporte de acuerdo al cliente en un plan que se llama "Integración con Sitios Web", el cual provee una API para poder incorporar los datos al sitio web propio.

Posee otros perfiles con sus funcionalidades enfocadas a transportistas y usuarios de cadenas de suministro del rubro del comercio internacional y seguimiento de sus cargas.

3.3.2. Arviem's

Es otro tipo de servicio [21] de monitoreo de contenedores a través de una página web y, además, se encuentra disponible como aplicación móvil para sus usuarios,

que permite seguir la ubicación y el estado de los envíos en tiempo real. Asimismo, esta puede monitorear otros parámetros de las necesidades específicas de los clientes como: cambios de temperatura y humedad dentro del contenedor, localización (la cual para ellos contempla distintos tipos de clientes: envíos aéreos, marítimos, terrestres y ferroviarios) y además, el cliente pueda predeterminar instrucciones como la apertura del contenedor del envío ante posibles robos. Por otra parte, basa sus estimaciones de localización en el cálculo dinámico de ETA, para que sus usuarios puedan planificar mejor y tomar medidas eficientes y proactivas en caso de demoras.

Este servicio no cuenta con datos públicos del costo de contratar su servicio, pero permite solicitar una demostración para probarla. Finalmente, el usuario puede visualizar por contenido de información en las distintas secciones en una lista de detalle, mapa de ubicación y gráfica de tiempos de retraso.

3.3.3. Ocean Insights

Esta plataforma web [22] de seguimiento de contenedores se enfoca en envío a clientes para toda su cadena de suministro como, por ejemplo: algún problema en el puerto de salida, cambio de buque, embarque y descarga en el puerto destino, incluye la visibilidad de contenedores a través de todo los procesos y transportistas con datos confiables, el transporte de la carga marítima en tiempo real de los buques, posiciones de mapas y gestión de excepciones.

El sistema permite tomar medidas que generen “notificaciones de alertas tempranas” antes de que ocurra un inconveniente en su cadena de suministro de logística, por medio del cálculo de la posición del envío respecto a la posición en que debiesen estar, la cual es determinada por el cliente. El cálculo del recorrido naviero, también realiza el seguimiento actualizado constante de ETA y el sistema de alerta notifica sobre demoras, altas del envío, transferencias o tiempos de espera inusuales.

3.3.4. Monitoreo de carga refrigerada

Esta es una solución específica, donde las navieras con contenedores refrigerados, cuentan con dispositivos para que sus clientes puedan realizar el seguimiento de estos [23].

Remote Container Management: Es una solución actual en el mercado, herramienta creada por la naviera *Maersk Line*, la cual entrega al cliente la posibilidad de monitorear la carga en el contenedor refrigerado desde su origen hasta su destino final, por medio de un dispositivo GPS, un módem y una tarjeta SIM, el que permite conocer las condiciones atmosféricas, permitiendo tener una visión completa y obtener datos en tiempo real de los productos durante el transcurso de su viaje. Una de las ventajas de este servicio, es que cuenta con el seguimiento directo de transmisores satelitales ubicados en 400 de los buques propiedad de la compañía.

3.4. Resumen

En el Cuadro 3.1, se realiza una comparación entre Searates, Arviem's y Ocean Insights,señalando funcionalidades en que estan presentes y/o faltan en las aplicaciones investigadas, para poder enfocarse en los principales problemas que debe solucionar nuestra aplicación.

	Searates	Arviem's	Ocean Insights
Permitir busqueda de un contenedor	✓	✓	✓
Dashboard de administración	✓	✓	✓
Despliegue de Mapa	✓	✓	✓
Actualización de ETA	✓	✓	✓
Interfaz amigable	✓	✓	✓
Monitoreo de Temperatura		✓	✓
Aplicación Movil	✓		
Alertas		✓	
Integración con webs navieras		✓	
Multiples perfiles			
Carga de archivos en formato .xlsx			

Cuadro 3.1: Comparación de las soluciones actualmente disponibles.

Capítulo 4

Desarrollo de la plataforma

En esta sección se presenta la arquitectura a utilizar para el desarrollo de la plataforma *SmartTracking* propuesta como solución de esta memoria. En primer lugar se definen las tecnologías a utilizar y el porqué estas fueron elegidas. Se realiza un análisis comparativo de las opciones que ofrece el mercado actual,Por el lado de desarrollo frontend y de backend, se analizan distintos aspectos de cada tecnologías para contrastarlas con las que se eligieron en la arquitectura final. También se hace un análisis de que sistema de base de datos usar para la persistencia de la información.También se hace un análisis de los requerimientos de la aplicación y por último se define los aspectos tanto del cliente como lo del servidor donde son descritos los diseños y todo lo que respecta a base de datos y flujo de la plataforma, separando cada capa que conforma el sistema.

4.1. Tecnologías

La plataforma propuesta tiene como finalidad almacenar toda la información referente a los contenedores pertenecientes al cliente, de manera que se pueda actualizar con información nueva de forma automática y notifique al usuario en caso de que ocurriera algún evento en especial (por ejemplo el retraso de un contenedor). Para poder realizar esto se necesita de ciertas tecnologías para el desarrollo del software final. Estas

tecnologías deben considerar 2 aspectos fundamentales que son:

- Cliente: Es la parte encargada de mostrar la información al usuario de una manera entendible, fácil de usar y intuitiva.

- Servidor: Es el encargado de manejar la información para poder enviarla al usuario.

Dado que nos encontramos en una etapa temprana del producto debemos elegir tecnologías que puedan mutar fácilmente con el tiempo, ya que es común que exista cierta incertidumbre con respecto al modelo de negocios, funcionalidades y diseño en un comienzo. De esta forma lo ideal es que cada componente del software sea independiente del resto basado en una arquitectura modular. Es en este punto donde entran los conceptos de frontend y backend, separando completamente la lógica del servidor con la de cliente. Tomando en cuenta lo anterior, podemos listar las tecnologías que pueden ayudar a el proceso de creación de software, para esto es importante señalar 3 conceptos los que son:

- **Lenguaje de programación:** Es un lenguaje formal que, mediante una serie de instrucciones, le permite a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para, de esa forma, crear programas que controlen el comportamiento físico y lógico de una máquina.

- **Framework:** Es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

- **Stack:** Es una lista de todos los servicios tecnológicos utilizados para construir y ejecutar una sola aplicación.

4.2. Análisis de tecnologías para Frontend

Para hacer el análisis de las tecnologías utilizadas para el desarrollo de la parte de cliente, descartaremos las que están quedando obsoletas, y solo nos centraremos en las que son más usadas en el mundo de desarrollo actual, las cuales son las siguientes:

- React JS
- Vue JS
- Angular JS

Estas tecnologías tienen aspectos en común, como por ejemplo: las tres utilizan Javascript, orientados a programación reactiva, es decir, la aplicación *reacciona* cada vez que se produzca un cambio en un flujo de datos. Como también, las tres opciones anteriores tienen en común el desarrollo basado en componentes, el cual ayuda a modularizar la aplicación, pudiendo separar, organizar y estructurar el código final.
A continuación, se realizará una comparación entre cada una de ellas, analizando tanto sus ventajas como desventajas.

4.2.1. React Js

React JS [24] es una biblioteca de Javascript *Open Source* creada por Facebook, diseñada para crear interfaces de usuario mediante componentes, estos pueden ser cada una de las piezas que forman la interfaz, como también la propia interfaz completa. Cada componente contiene tanto la lógica como la parte visual, de este modo podemos reutilizar los componentes dentro de otros componentes. React facilita construir aplicaciones en las cuales los datos varían en el tiempo.

Ventajas

- ***Amplia comunidad:*** Al estar basado en Javascript, cuenta con una amplia comunidad que dejan a disposición un gran número de bibliotecas externas con una muy buena documentación para las distintas funcionalidades que se requieran desarrollar.

- ***Composición de componentes:*** Las aplicaciones se elaboran con la composición de varios componentes donde se encapsulan el comportamiento, una vista y un estado de una variable, por lo que la complejidad que tome el proyecto no es una preocupación, dado que el comportamiento reside dentro de cada componente y el resultado final será un conjunto de componentes generando una facilidad de mantenimiento, depuración y escalabilidad de la aplicación.

- ***Flujo de datos unidireccional:*** El patrón de funcionamiento que posee React hace que los componentes superiores propaguen los datos a los que están en un orden inferior, estos trabajarán con esos datos y cuando cambian su estado envían los eventos hacia los de orden superior para actualizarse.

Desventajas

- ***No ionizado:*** Lo que significa que los desarrolladores a veces tienen muchas opciones de desarrollo.

- ***Renderización:*** El proceso de generación de las imágenes puede tomar más tiempo del estimado si es que hay grandes volúmenes de datos involucrados, provocando que la aplicación pueda ver afectado su rendimiento.

- ***Integración del JSX:*** Es una extensión en React que permite mezclar sintaxis HTML con JavaScript, esta mezcla de lenguajes puede producir confusión durante el desarrollo.

4.2.2. Vue Js

Vue [25] es un framework progresivo, lo que quiere decir que puede ser usado tanto para tareas básicas, como para tareas más complejas. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single-Page Applications cuando se utiliza en combinación con herramientas modernas y librerías de apoyo.

Ventajas

- ***Gran Escala:*** Se pueden desarrollar plantillas reutilizables como una alternativa más simple, cualquier HTML válido es también una plantilla Vue válida, haciendo mucho más fácil la migración progresiva de las aplicaciones.
- ***Documentación detallada:*** Está bien estructurada y cubre todos los temas posibles, describiendo con precisión todo, desde la instalación hasta aspectos más detallados, como la reactividad y el escalado de la aplicación.
- ***Documentación de iniciación:*** Vue.js tiene una documentación muy circunstancial que puede ajustar la curva de aprendizaje para los desarrolladores y ahorrar mucho tiempo para desarrollar una aplicación utilizando solo los conocimientos básicos de HTML y JavaScript.
- ***Rendimiento:*** Vue.js puede pesar alrededor de 20 KB manteniendo su velocidad y flexibilidad que permite alcanzar un rendimiento mucho mejor en comparación con otros framework y librerias.

Desventajas

- ***Exceso de flexibilidad:*** A veces, Vue.js puede tener problemas al integrarse en grandes proyectos y todavía no hay experiencia con posibles soluciones.

- ***Falta de recursos:*** aún tiene poca cuota de mercado comparado con Angular o React, lo que significa que los recursos disponibles de este framework aún están en su fase inicial.

- ***Barrera del idioma:*** Al estar desarrollada principalmente en empresas chinas como Xiaomi y Alibaba, gran cantidad de los desarrollos surgen en chino, lo cual crea una barrera idiomática.

4.2.3. Angular JS

HTML es excelente para declarar documentos estáticos, pero falla cuando intentamos usarlo para declarar vistas dinámicas en aplicaciones web. AngularJS [26] permite ampliar el vocabulario HTML para su aplicación. El entorno resultante es extraordinariamente expresivo, legible y rápido de desarrollar.

Ventajas

- ***Desarrollado por Google:*** Al tener un respaldo como Google se puede estar seguro de que el código es confiable y eficiente.

- ***Enfocado en el modelo MVC:*** Una de las mejores alternativas cuando se desea una interacción con la página web y los datos proporcionados. Para esto, Angular crea una solución ligera que mantiene el equilibrio entre cliente-servidor.

- ***Enlaces de datos Bidireccionales:*** Significa que se puede realizar cualquier cambio relacionado con los datos e inmediatamente se propagaría a las vistas correspondientes y cuando se realiza cualquier cambio en la vista, eso también ocurriría en el modelo subyacente. Tan pronto como cambien los datos de la aplicación, también habrá cambios correspondientes en la interfaz de usuario.

Desventajas

- ***Bibliotecas para Angular son muy específicas:*** El desarrollo no funciona muy bien con herramientas o bibliotecas que no son específicas de AngularJS.

- ***Curva de aprendizaje lenta:*** La dificultad de aprender este Framework es alta comparada a otros. Además, la documentación limitada disponible puede afectar aún más el proceso de aprendizaje.

- ***Soporte de JavaScript obligatorio:*** Si está deshabilitado JavaScript en el computador los usuarios asociados no podrán acceder a su sitio web o aplicaciones web.

4.3. Análisis de tecnologías para Backend

Según la encuesta anual realizada por *Stack Overflow*[27] entre los framerworks mas utilizados (solo tomando en cuenta los de desarrollo backend) se encuentran:

- Express (Javascript)[28]
- Spring (Java)[29]
- Django (Python)[30]
- Ruby on Rails (Ruby)[31]

Hay que destacar que estas no son las únicas alternativas para el desarrollo por parte del servidor, pero al ser las mas usadas cuentan con comunidades mas grandes que las respaldan, lo que conlleva menor cantidad de errores, mas documentación y mayor variedad de librerías que facilitan el desarrollo. A continuación , se realizara una comparación entre cada una de ellas, considerando aspectos como la documentación disponible, curva de aprendizaje y escalabilidad.

4.3.1. Express

Según la definición de su web oficial[28]. “Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles.“ Este framework funciona en el lenguaje de programación javascript , un lenguaje que desde sus inicios fue pensando para la parte del cliente y no la del servidor, para hacer esto posible nace nodeJS[32] que es un entorno que trabaja en tiempo de ejecución, de código abierto, multi-plataforma.

Documentación disponible

Express[28] es hoy por hoy la herramienta de desarrollo desde el punto de vista del servidor mas popular del momento, gracias a que usa NodeJS[32] es posible usar todas las librerías que tiene disponible NPM(Node Package Manager)[33], el cual es uno de los gestores de paquetes de código mas grande del mundo, lo que significa que hay una comunidad muy grande que desarrolla código que podemos incluir en nuestro desarrollo, optimizando los tiempos y teniendo código que ya fue testeado y depurado por muchas mas personas.

Curva de aprendizaje

Nace de una premisa minimalista, esto quiere decir que te otorga los componentes mínimos para establecer la comunicación y puedas comenzar a desarrollar sin problemas, lo que trae consigo sus ventajas y desventajas. Debido que al ser minimalista es muy fácil adaptarlo a cualquier tipo de estructura y desarrollo. La desventaja de esto que es que le da mucha libertad al desarrollador para crear el software, lo que en casos de desarrolladores de poca experiencia podría decaer en código mal estructurado y desordenado, lo que dificultaría futuras actualizaciones de la aplicación.

Escalabilidad

Al ser altamente configurable es muy fácil escalar y refactorizar en cualquier etapa de desarrollo, junto con esto Express siempre tuvo entre premisas ser una herramienta para el desarrollo de micro-servicios, lo que como su nombre lo indica es poder separar un servicio grande en varios servicios pequeños mas fáciles de manejar y mejorar.

4.3.2. Spring

Spring[29] es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java. Según su definición oficial[29] " Spring hace que la programación de Java sea más rápida, fácil y segura para todos. El enfoque de Spring en la velocidad, la simplicidad y la productividad lo ha convertido en el marco de trabajo Java más popular del mundo ". La principal ventaja de esta herramienta es que simplifico en gran medida el desarrollo en java, algo que genera mucha expectativa ya que java es un lenguaje de programación con gran trayectoria y mucha potencia.

Documentación disponible

Spring cuenta con gran cantidad de material proporcionado desde su sitio web oficial,entre lo que destaca una documentación bastante completa, vídeos tutoriales. También, hacen conferencias cada cierto tiempo explicando distintas partes del framework. En términos generales hay mucha información y aun tiene una comunidad muy activa a pesar del tiempo que lleva en el mercado, lo que lo hace una muy buena opción para cierto tipo de proyectos.

Curva de aprendizaje

El aprender a usar bien Spring[29] es complicado, debido a que hay muchas configuraciones que manejar y además usa java un lenguaje que no destaca por su simplicidad, es por esto que para aprender Spring es indispensable haber trabajado anterior-

mente en java, esto no pasa con frameworks basados en lenguajes de tipificado simple como lo son python y ruby , ya que no hay que dedicarle tanto tiempo a solo aprender el lenguaje base para poder hacer uso del sus herramientas.

Escalabilidad

Spring funciona muy bien tanto en proyectos pequeños como proyectos muy grandes, esto debido a su estructura solida bien definida. Este framework se basa en la filosofía "convención sobre configuración", reduciendo al mínimo el número de pasos que un desarrollador debe dar en la configuración inicial del proyecto antes de ponerse a trabajar en la parte dura del mismo centrando sus esfuerzos en lo importante. Esto ayuda a dar mayor orden a los proyectos aun para programadores con poca experiencia.

4.3.3. Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como MVC. Según su web oficial[30] " Django es un framework de alto nivel para la web en Python que fomenta un rápido desarrollo y un diseño limpio y pragmático. Construido por desarrolladores experimentados, se encarga de gran parte de las molestias del desarrollo Web, de modo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es libre y de código abierto".

Documentación disponible

Django[30] desde su web oficial ofrece una basta documentación, la cual incluye tutoriales,vídeos,acceso a una gran comunidad y una documentación que esta en varios idiomas incluyendo ingles y español. También podemos decir que tiene una comunidad muy activa con mas de 5000 usuarios activos en la actualidad.

Curva de aprendizaje

La principal ventaja de Django es que esta escrito en python un lenguaje que combina eficiencia con simplicidad, python cuenta con una sintaxis muy simple basada en legibilidad de código, soporta orientación a objetos y es hoy por hoy uno de los lenguajes más usados para análisis de datos. Python es relativamente fácil de aprender, haciendo que usar Django tenga un alcance mucho mayor. Si solo nos centramos en el framework, este trae consigo muchas aplicaciones integradas lo cual es una ventaja para el desarrollo ya que hace de este tremendamente adaptable a las necesidades del negocio, el punto bajo de esto, es que al tener tantas aplicaciones lo vuelve un tanto difícil de manejar en un principio, por lo que aprender a utilizar este framework resulta algo difícil.

Escalabilidad

Esto claramente es una ventaja de Django[30] su documentación oficial te enseña como manejar el framework para aplicaciones pequeñas y para aplicaciones grandes, es tremendamente adaptable y estable. Cuando ya se sabe usar, Django puede adaptarse al crecimiento de la aplicación, esto debido a que el framework trae consigo muchas aplicaciones para diversos ámbitos.

4.3.4. Ruby on Rails

Ruby on Rails[31] es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, que al igual que Django sigue un patrón MVC. Su principal virtud según sus creadores, es que hace el desarrollo mucho mas rápido y ágil que su competencia, pudiendo realizarse aplicaciones simples como un foro en tan solo 5 minutos. Esto quedo demostrado en varios vídeos realizado por la comunidad donde programadores con experiencia pudieron realizar este desafió.

Documentación disponible

Ruby on Rails fue uno de los framwork de desarrollo web más usados en sus inicios creando una comunidad muy grande que lo respalda, su web oficial cuenta con una documentación bastante detallada de cada aspecto de uso, por lo cual sera muy fácil encontrar información que se adapte a cualquier tipo de desarrollo.

Curva de aprendizaje

La ventaja de este framework es que tiene un sistema gestión de persistencia de datos (CRUD) automático llamado "scaffold", el cual crea toda la estructura inicial de un objeto tanto en la vista como el controlador, la ventaja de esto es que acelera mucho el proceso de crear nuevos objetos o asociaciones, pero la desventaja es que muchas veces genera código inútil o basura que no se usa en el desarrollo, por lo cual utilizar esta herramienta en etapas avanzadas del desarrollo no se considera como recomendado, pero si es muy útil en etapas iniciales para desarrolladores inexpertos.

Escalabilidad

Quizás un punto bajo en RoR es la excesiva creación de código automático, lo que a la larga genera grandes cantidades de código por todos lados, lo que lo hace difícil de mantener y de escalar, a menos que se tenga un buen control de todo lo que se agrega desde un principio, dando un orden y estructura definida para cada acción, de forma que escalar una aplicación desarrollada con este framework sea menos engorroso.

4.4. Base de datos

Para que los datos que los usuarios guarden en nuestra plataforma sean persistentes, se debe crear una estructura de base de datos que sea capas de manejar de forma ordenada toda la información. En la actualidad existen dos formas de manejar los datos, las que son las bases de datos relacionales y las bases de datos no relacionales, ambas cuentan con varios sistemas que proporcionan sus servicios como por ejemplo mySQL, postgresSQL, Microsoft SQL Server, mongoDB, dynamoDB, etc. En esta sección solo nos centraremos en que tipo de base de datos usar y no en los sistemas, dado que la elección del sistema depende muchas veces de las tecnologías que se usen para el desarrollo y la compatibilidad con estas.

4.4.1. Base de datos relacional

Son una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas, desde donde se puede acceder a los datos o volver a montarlos de muchas maneras diferentes sin tener que reorganizar las tablas de la base. La interfaz estándar de programa de usuario y aplicación a una base de datos relacional, es el Lenguaje de Consultas Estructuradas (SQL). Los comando SQL se utilizan tanto para consultas interactivas como para obtener información de una base de datos relacional y la recopilación de datos para informes.

Las bases de datos relacionales se basan en la organización de la información en partes pequeñas que se integran mediante identificadores, estos identificadores son conocidos como llaves (o keys), los que generan una relación o enlace entre los datos.

La principal desventaja de este tipo de base de datos, proviene de su rigidez en la estructura, debido a que el usuario debe definir previamente como estarán estructurados cada uno de los elementos, tanto en sus relaciones como en atributos.

4.4.2. Base de datos no relacional

Están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Son ampliamente reconocidas porque son fáciles de desarrollar, tanto en funcionalidad como en rendimiento a escala. Usan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en-memoria y búsqueda.

Las bases de datos no relacionales (NoSQL) son las que, a diferencia de las relacionales, no tienen un identificador que sirva de relación entre un conjunto de datos y otros. Como veremos, la información se organiza normalmente mediante documentos y es muy útil cuando no tenemos un esquema exacto de lo que se va a almacenar.

Con relación a formatos, la información de una base de datos puede ser almacenada en tablas o documentos. Cuando los datos son organizados en un archivo de Excel, es en formato tabla, pero cuando simplemente son datos escritos como cartas, fórmulas o recetas, son datos en formato documento. Esto aplica para los dos tipos de bases de datos.

Habitualmente los datos almacenados en tablas son bases de datos relacionales, porque existe la posibilidad de enlazar los datos de una tabla con los de otra y los datos almacenados en documentos son no relacionales, aunque no siempre tiene que ser así. Por ejemplo, los datos de una tabla pueden ser transcritos a un documento, todo depende del punto de vista y la necesidad del problema que se vaya enfrentar.

A diferencia de su contra parte las base de datos no relacionales no tienen una estructura rígida para el manejo de datos, por lo que si se quisiera hacer cambios en las estructura de los datos en etapas avanzadas de un desarrollo, no seria tan engorroso ya que difícilmente estos cambios afecten el trabajo previamente realizado.

4.5. Conclusión acerca del Framework a utilizar

4.5.1. FrontEnd

A modo de resumen, se confeccionaron tres gráficos, comparando las principales características. El gráfico de la Figura 4.1, compara la “Curva de aprendizaje”versus la “Experiencia de desarrollo”, en el cual tiene superioridad React Js, por la gran cantidad de bibliotecas que este posee y existencia de herramientas que facilitan su desarrollo (*React Developer Tools*), mientras que Angular, queda en último lugar, por su empinada curva de aprendizaje que puede llegar hasta ser frustrante.

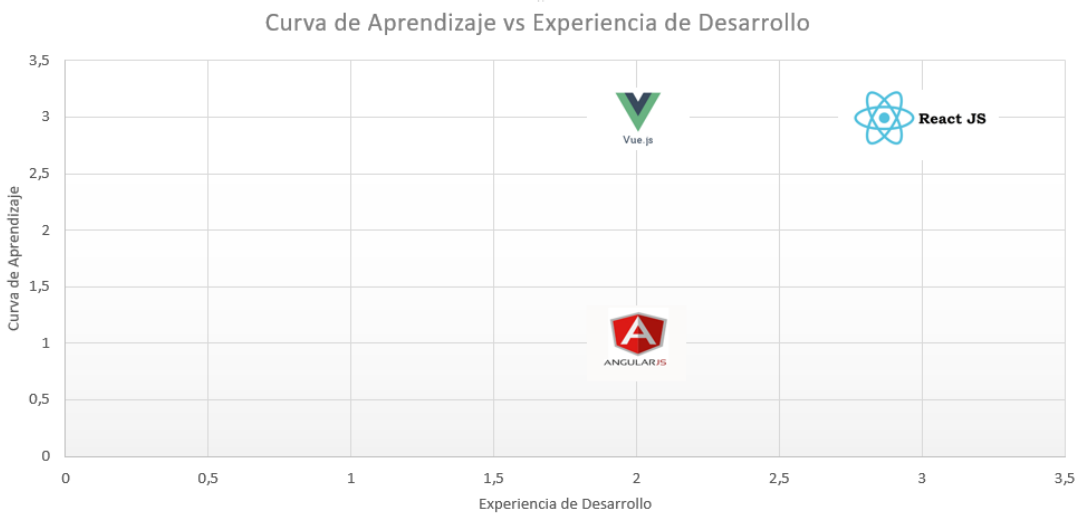


Figura 4.1: Curva de Aprendizaje vs Experiencia de Desarrollo.

En la Figura 4.2 se compara la “Mantención de código”v/s “Aplicaciones de gran escala”, donde Angular es mejor en esto último, puesto que fue confeccionado para aplicaciones grandes, pero falla en la mantención de códigos al poseer muchas acciones que funcionan como caja negra, por su parte Vue JS está en un punto medio entre ambas categorías, pero React JS, supera a este, por su forma orientada a componentes, pudiendo modularizar de mejor forma el código, llegando a ser auto explicativo.

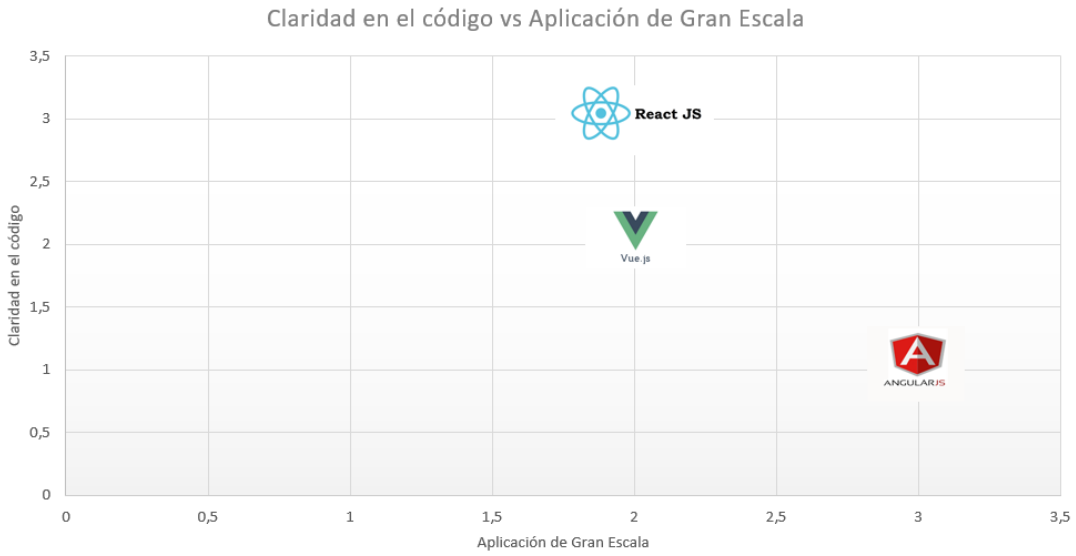


Figura 4.2: Claridad de Código vs Aplicación de Gran Escala.

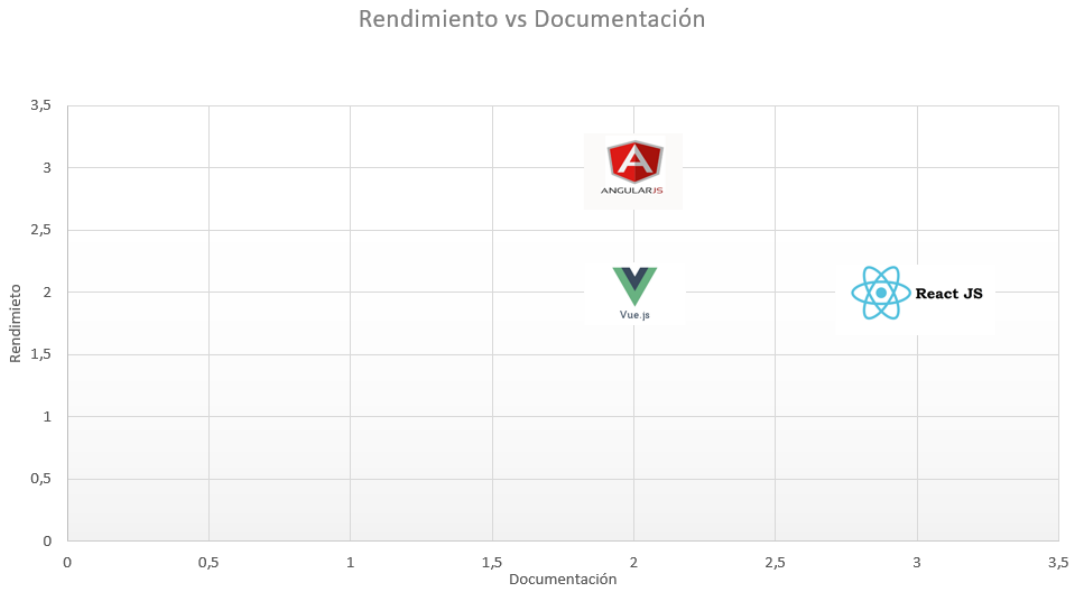


Figura 4.3: Rendimiento vs Documentación.

Por último, en la Figura 4.3, se compara Rendimiento v/s Documentación, en términos de rendimiento, Angular es superior a los otros dos framework. React y Vue JS, se manejan de similar manera, pero al comparar su documentación, React gana en ese aspecto, por la gran comunidad que hay detrás de este Framework, mientras que Vue

JS, posee un rendimiento medio y documentación buena, pero la barrera del idioma limita que esta documentación podría ser mejor. Por lo tanto, se decidió elegir React JS, por su amplia documentación, una buena curva de aprendizaje y una fácil forma de mantener un orden al momento de programar, dejando atrás características como, por ejemplo, si es que será una aplicación de gran escala o rendimiento, puesto que solo se desea conseguir un mínimo producto viable para esta etapa.

A modo de resumen general, se muestra el siguiente cuadro 4.1, el cual según los criterios anteriormente señalados, sería la mejor tecnología para el desarrollo Frontend:

FrameWorks de desarrollo Frontend	Puntaje final
React JS	16
Vue JS	13
Angular JS	11

Cuadro 4.1: Ordenamiento de mejor a peor Framework de desarrollo Frontend en base a los puntajes de las gráficas anteriores.

4.5.2. BackEnd

Hay que destacar que todas las herramientas de desarrollo definidas en este documento, son suficientes y podrían cumplir con el desarrollo de lado de servidor de nuestra plataforma. Es por esto que para la elección se consideraron criterios como conocimiento previo del lenguaje y cual tiene mejor proyección en el futuro dado que aún nos encontramos en etapas tempranas del desarrollo.

A continuación, se muestra una tabla de calificación de lenguajes de programación según conocimiento previo con una escala de 1 a 10, donde 1 representa tener un completo desconocimiento de su uso y 10 representa un conocimiento avanzado proporcionado por el uso y experiencias previas.

Lenguaje	Calificación
Javascript	9
Java	4
Python	7
Ruby	7

Cuadro 4.2: Calificación de lenguajes según conocimiento previo.

Para los criterios definidos en 4.3 se utiliza evalúan según una calificación de 1 a 10. detallados a continuación

Documentación disponible

Una calificación de 1 representa que tiene poca o nula documentación oficial o de su comunidad, por otro lado una calificación de 10 representa que tiene una solida documentación para distintos niveles de aprendizaje(principiante, semi avanzado y avanzado).

Curva de aprendizaje

Se consideran para este punto tanto el lenguaje de programación como el funcionamiento del framework, por lo que una calificación de 1 representa que tanto el lenguaje

Framework	Calificación
Express JS	9
Spring	7
Django	5
Ruby on rails	9

Cuadro 4.3: Calificación de frameworks según documentación disponible .

como el framwork tienen una dificultad muy elevada para usarlos, esto quiere decir que el aprender dichas tecnologías requiere un tiempo mayor o igual a 2 semanas. Mientras que una calificación de 10 representa que tanto el lenguaje como el framework son de aprendizaje rápido por lo cual se podrían empezar a usar sin dificultades en un tiempo menor o igual a 2 días. Esto considera un nivel de aprendizaje básico.

Framework	Calificación
Express JS	7
Spring	4
Django	5
Ruby on rails	9

Cuadro 4.4: Calificación de frameworks según curva de aprendizaje .

Escalabilidad

Para este punto se considera modularización y casos de éxito en aplicaciones maduras, donde una calificación de 1 representa una estructura poco escalable y ninguna aplicación madura que los respalde, mientras que una calificación de 10 representa una fácil modularización y múltiples casos de éxito que la respaldan.

Framework	Calificación
Express JS	9
Spring	8
Django	6
Ruby on rails	7

Cuadro 4.5: Calificación de frameworks según escalabilidad .

A continuación se presenta una tabla resumen con cada punto evaluado en esta sección.

	Express JS	**Spring**	**Django**	**Ruby on rails**
Conocimiento previo	9	4	7	7
Documentación disponible	9	7	5	9
Curva de aprendizaje	7	4	5	9
Escalabilidad	9	8	6	7

Cuadro 4.6: Resumen tecnologías backend .

4.5.3. Base de datos

Sabemos que nuestra plataforma aún esta en una etapa muy temprana de desarrollo, por lo cual es probable que tenga múltiples cambios a nivel de producto en un corto plazo. Es por esto que se vuelve indispensable tener una estructura de base de datos dinámica, que se adapte a este requerimiento, la cual nos debe proporcionar la mayor flexibilidad para cambiar su estructura y comportamiento. Tomando en cuenta esos requerimiento es la razón por la que se decide usar una base de datos de tipo no relacional. Dado que nos ofrece una estructura menos rígida la cual podríamos mutar fácilmente.

4.5.4. Resumen general de Tecnologías

En estas sección se describieron varias tecnologías utilizadas en la actualidad para el desarrollo de plataformas web, tanto para el lado de cliente como de servidor. El objetivo principal es elegir aquellas tecnologías que mejor se adapten a nuestros requerimiento y que no nos obliguen a migrar a otra en un mediano plazo. Es por eso que tomando en cuenta todo el análisis previo se llega a la conclusión que una alternativa aceptable sería usar por el lado de cliente la librería ReactJs[24] y por el lado de servidor crear una interfaz de programación de aplicaciones(API) en ExpressJs[28] de modo que nuestra API se conecte tanto con la base de datos por un lado, como al cliente por otro. En la actualidad existe un conjunto de tecnologías (conocidos en in-

gles como *stack*) que incluyen las a nuestras ambas propuestas este *stack* es conocido como MERN el cual utiliza JavaScript tanto en el cliente como en el servidor es decir Full Stack JavaScript, lo cual es una ventaja ya que todo nuestro desarrollo estaría en el mismo lenguaje.

En la figura 4.4 se puede apreciar como se divide en dos la aplicación *SmartTracking*, separándose claramente el Backend y el Frontend con sus respectivas tecnologías a utilizar, a continuación de la imagen, se procede a explicar cada una de estas tecnologías.

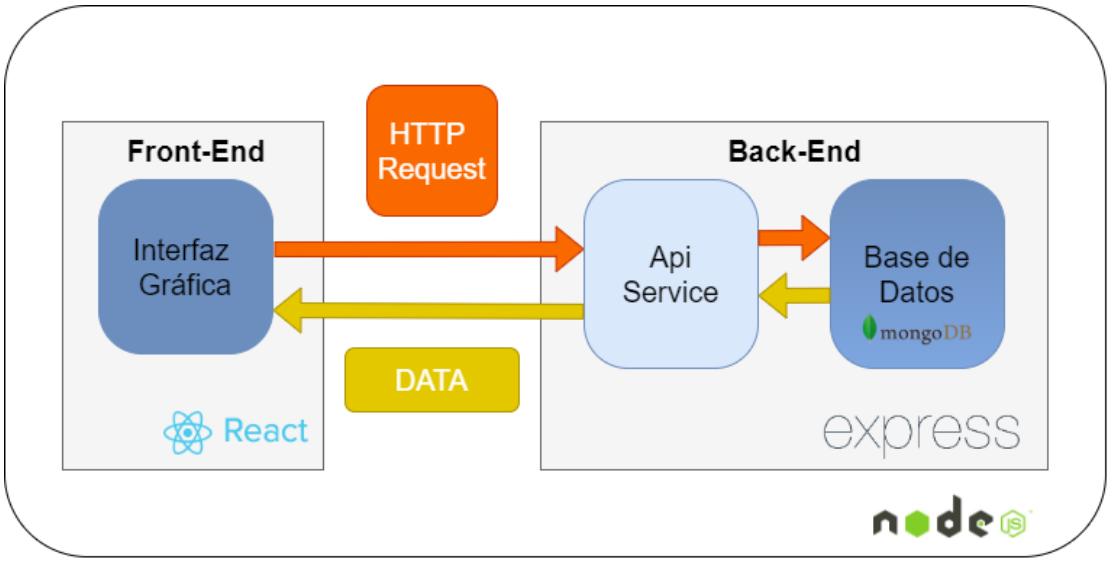


Figura 4.4: Diagrama de arquitectura.

MERN significa la mezcla de 4 tecnologías anteriormente señaladas:

- **Mongo**: Es una base de datos no relacional, donde se guardará, toda la informaciones de los contenedores y usuarios.
- **Express**: Es parte del backend, donde se construyen las APIs que se encargaran de ver qué petición se han hecho desde el cliente y reaccionará devolviendo los datos requeridos.
- **React**: Framework de desarrollo para realizar la interfaz web.

- **Node**: Es el entorno ejecución multiplataforma.

4.6. Requerimientos del sistema

A continuación se definen una serie de requisitos con los cuales tiene que contar el sistema, estos son obtenidos gracias a múltiples reuniones con la contra-parte de este proyecto y explorando las principales falencias de su sistema actual. El requisito principal del sistema es informar y actualizar de forma correcta los tiempos de arribo de los contenedores. Para poder desarrollar dicho objetivo es que se definen una serie de requisitos funcionales y no funcionales.

4.6.1. Requisitos funcionales

- Ingreso y registro de personas.
- Registro de información de los contenedores.
- Desplegar mapa con la ubicación e información de los contenedores.
- Actualización de los ETA de forma periódica.
- Despliegue de alarmas en casos de la variación de los ETA

4.6.2. Requerimientos no funcionales

- El sistema tiene que tener una disponibilidad superior al 95 % de modo que el acceso a los datos sean continuo.
- Interfaz amigable para el despliegue de los resultados.

4.7. Capa de datos (base de datos)

A partir de los requisitos de la plataforma se puede elaborar un diagrama de las colecciones que tendrá nuestra base de datos no relacional. La principal diferencia de un diagrama de colecciones es que como proviene de una base de datos no relacional, este no tiene una estructura fija y tampoco relaciones forzadas, esto no quiere decir que no se puedan usar relaciones, si no que estas no son obligatorias, pudiendo manejar los datos en un solo documento.

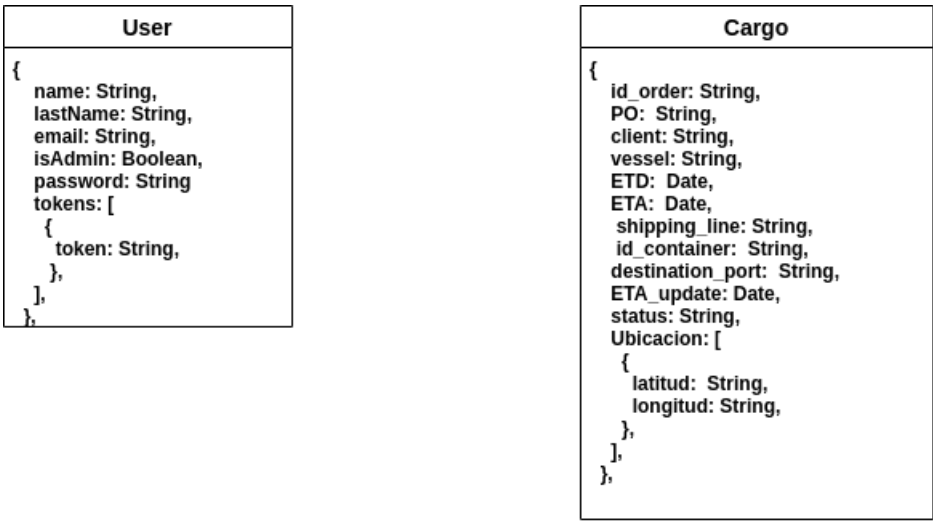


Figura 4.5: Estructura de base de datos

En la figura 4.5 se expone la estructura de las colecciones que componen la base de datos, donde se define la estructura básica sus documentos. En una primera iteración solo consideramos dos de estas Usuario y Cargamentos. La colección de usuario define las características que tendrá que ingresar cada persona que vaya a utilizar la aplicación, mientras que la colección de cargamentos guardara todos los datos relacionado con los cargamentos del cliente.

4.8. Capa de negocios(backend)

Tomando en cuenta los requerimientos del sistema y la arquitectura vista en 4.4 es que se desarrolla una API en ExpressJs el cual usa como Modelado de datos de objetos(ODM) mongoose[34]. Esta API se encargará de generar la comunicación entre el cliente y la base de datos, además de ejecutar tareas de actualización conectándose con servicios externos pertenecientes a las navieras.

A continuación, se definen como estará estructurado cada endpoint a partir de los requisitos del sistema.

Definición/ Ruta	**Método**	**Header**	**Body**

Iniciar sesión
 	
”Host/api/v1/users/login
	POST	Content-Type: application/json	{
			email: String,
			password: String
			}

Finalizar sesión ”Host/api/v1/users/logout	POST	Content-Type: application/json authorization: token	{}
Finalizar sesión en todos los dispositivos ”Host/api/v1/users/logout_all	POST	Content-Type: application/json authorization: token	{}

Registro de usuario
 	
”Host/api/v1/users
	POST	Content-Type: application/json
		authorization: token
			{
			name: String,
			lastName: String,
			email: String,
			password: String
			}

Lista de usuarios ”Host/api/v1/users	GET	Content-Type: application/json authorization: token	{}

Editar usuario
 	
”Host/api/v1/users/:id_user
	PUT	Content-Type: application/json
		authorization: token
			{
			
			name: String,
			lastName: String,
			email: String,
			password: String
			
			}

Ver informacion de usuario ”Host/api/v1/users/:id_user	GET	Content-Type: application/json authorization: token	{}
Eliminar usuario ”Host/api/v1/users/:id_user	DELETE	Content-Type: application/json authorization: token	{}

Cuadro 4.7: Tabla de accesos de usuario

Definición/ Ruta	**Método**	**Header**	**Body**
Lista de cargamentos ”Host/api/v1/cargos	GET	Content-Type: application/json authorization: token	{}

Crear cargamento
 	
”Host/api/v1/cargos
	POST	Content-Type: application/json
		authorization: token
			{
			id_order: String,
			PO: String,
			client: String,
			vessel: String,
			ETD: Date,
			ETA: Date,
			shipping_line: String,
			id_container: String,
			destination_port: String,
			ETA_update: Date,
			status: [
			ÏN PRODUCTION”,
			”PROGRAMMED”,
			”DISPATCHED”,
			”SHIPPED”,
			],
			location:[
			{
			latitud: String,
			longitud: String
			}
			
			]
			}

Importar información desde excel
 	
”Host/api/v1/cargos/upload_informacion
	POST	Content-Type: application/json
		authorization: token
			{
			file: Base64
			}

Resumen de estado de cargamentos ”Host/api/v1/cargos/estado_alarmas	GET	Content-Type: application/json authorization: token	{}
Ver información de un cargamento ”Host/api/v1/cargos/:id_cargo	GET	Content-Type: application/json authorization: token	{}

Definición/ Ruta	**Método**	**Header**	**Body**

Editar información de un cargamento
 	
”Host/api/v1/cargos/:id_cargo
	PUT	Content-Type: application/json
		authorization: token
			{
			id_order: String,
			PO: String,
			client: String,
			vessel: String,
			ETD: Date,
			ETA: Date,
			shipping_line: String,
			id_container: String,
			destination_port: String,
			ETA_update: Date,
			status: [
			ÏN PRODUCTION”,
			”PROGRAMMED”,
			”DISPATCHED”,
			”SHIPPED”,
			],
			location:[
			{
			latitud: String,
			longitud: String
			}
			
			]
			}

Eliminar cargamento ”Host/api/v1/cargos/:id_cargo	DELETE	Content-Type: application/json authorization: token	{}

Cuadro 4.8: Tabla de accesos de contenedores

4.9. Capa de presentación (frontend)

En esta sección se define como es que se comunica la interfaz con la API, definiendo cada una de sus interacciones y el flujo que tendrá la aplicación final.

4.9.1. Ingreso de usuarios

SMART TRACKING debe verificar la identidad de cada usuario que ingresa a la plataforma, dado que dentro de la aplicación existen dos tipos de usuarios:

- **Operario:** Este usuario puede acceder a algunas de las funcionalidades de la aplicación, que son las siguientes:
  - Visualización de la lista de contenedores.
  - Filtrar la lista de contenedores.
  - Sistema de alerta en la lista de contenedores para aquellos que están con retraso.
  - Visualización de contenedores en mapa global.
- **Administrador:** Puede acceder a todas las funcionalidades anteriormente descritas y, además, puede crear usuarios, subir archivos en formato Excel con información de contenedores y agregar la información de un solo contenedor.

En la vista de ingreso, el componente contiene dos estados:

- SignInEmail.
- SignInPassword.

Estas dos variables son las encargadas de almacenar los valores que ingresará el usuario y confirmar sus credenciales, luego se utiliza la biblioteca *Fetch API*, la que proporciona métodos para obtener recursos (incluso a través de la red), otorgando un conjunto de características más potentes y flexibles. Con estos métodos se realizan las

cargas *HTTP Request* como: GET, POST, DELETE, PUT, un HTTP Request puede incluir un *Header* y un *Body*, este método retorna un *Response* y donde se obtiene la respuesta.

En este caso, se realiza una *Request* de tipo *POST* a la URL *"http://localhost:4000/api/v1/users/login"* puesto que se envía información cifrada desde el cliente para que sea procesada en el *backend* y así actualice la información en el servidor. En el *Header*, se indica que los datos de entrada será en formato *JSON*, esto se logra agregando la linea *'Content-Type': 'application/json'*. Por último, en el *Body*, se agregan los estados *SignInEmail* y *SignInPassword*, para verificar si las credenciales son correctas.

Luego, la respuesta de la API tiene 2 alternativas:

- Credenciales Incorrectas: Desde la API, en la Response, no se envia el *Token*, el cual es nuestro mecanismo de seguridad para el consumo de la API, además informa que las credenciales son incorrectas , luego se despliega un mensaje con ayuda de la biblioteca *SweetAlert* indicando que el correo o la contraseña son inválidas, manteniéndose en el Login e indicando que se revisen los datos.
- Credenciales Correctas: La Response indica que las credenciales son correctas, en la Response se envía el *Token* y se almacena, también se almacenan los datos de sesión para identificar si es un operador o un administrador y dependiendo de esto último, se procederá a redirigirse a la vista principal para administrador u operario, donde se encuentra la tabla con los datos de los contenedores.

Definición y funcionamiento de los Tokens

Hoy en día, las APIs son parte importantísima del ecosistema de desarrollo, gracias a ellas se tiene acceso a los datos en distintas empresas u organismos, dado que nos permiten mandar o pedir información y utilizar servicios de terceros de forma sencilla.

Sin embargo, dejar expuesto un servicio permite que cualquier persona tenga acceso a este, por el hecho de estar en la red, lo que provoca que estas APIs deban protegerse, para que así no cualquiera pueda realizar peticiones a los servicios.

Hay distintos métodos para poder dar acceso a un API, por ejemplo: la autenticación a través de tokens, la que se utiliza en la aplicación *SMART TRACKING*. En términos simples, un servicio basado en autenticación por token, es aquel que a través de una cadena de caracteres enviadas desde el cliente hacia la API (como por ejemplo una contraseña) detecta el *token* para así poder revisar a qué usuario pertenece, si es válida o no, además mantener registros de peticiones, y una seguridad en la aplicación impidiendo que cualquier persona pueda acceder a los servicios.

La autenticación por tokens de *SMART TRACKING* cuenta de los siguientes pasos:

1. Autenticación usando credenciales regulares (usuario-contraseña).
2. Una vez autentificado en el servidor, se genera una cadena de caracteres que contiene el token proporcionado por la biblioteca *JWT* integrada.
3. Envío del token al cliente.
4. Almacenamiento de este en el lado del cliente.
5. Envío del token al servidor en cada petición que se realiza.
6. Validación del token en el servidor, y otorgamiento (o no) de acceso al recurso que el cliente solicita.

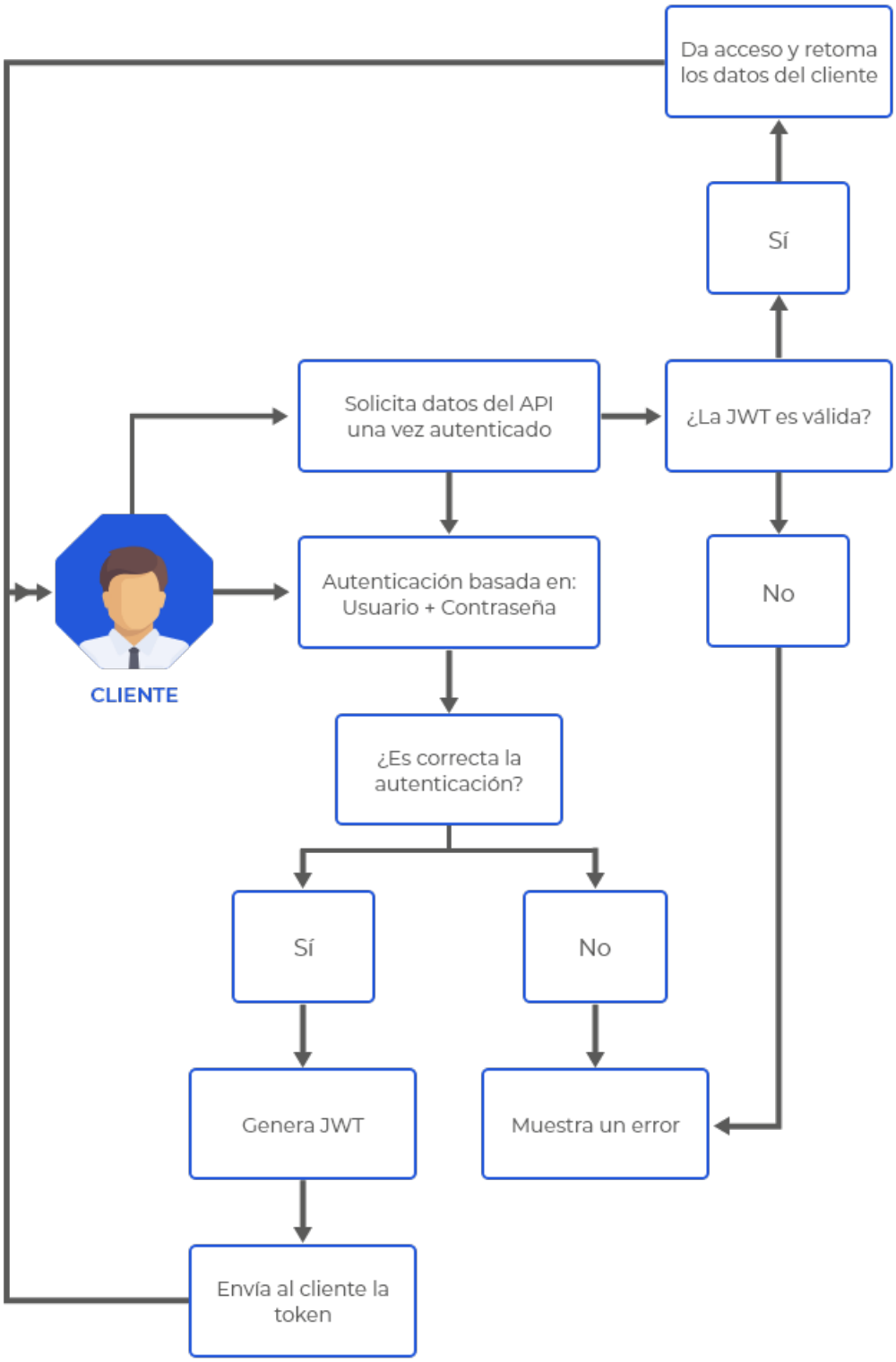


Figura 4.6: Diagrama de flujo de los Token

Por lo tanto, el flujo del módulo de login queda de la siguiente forma:

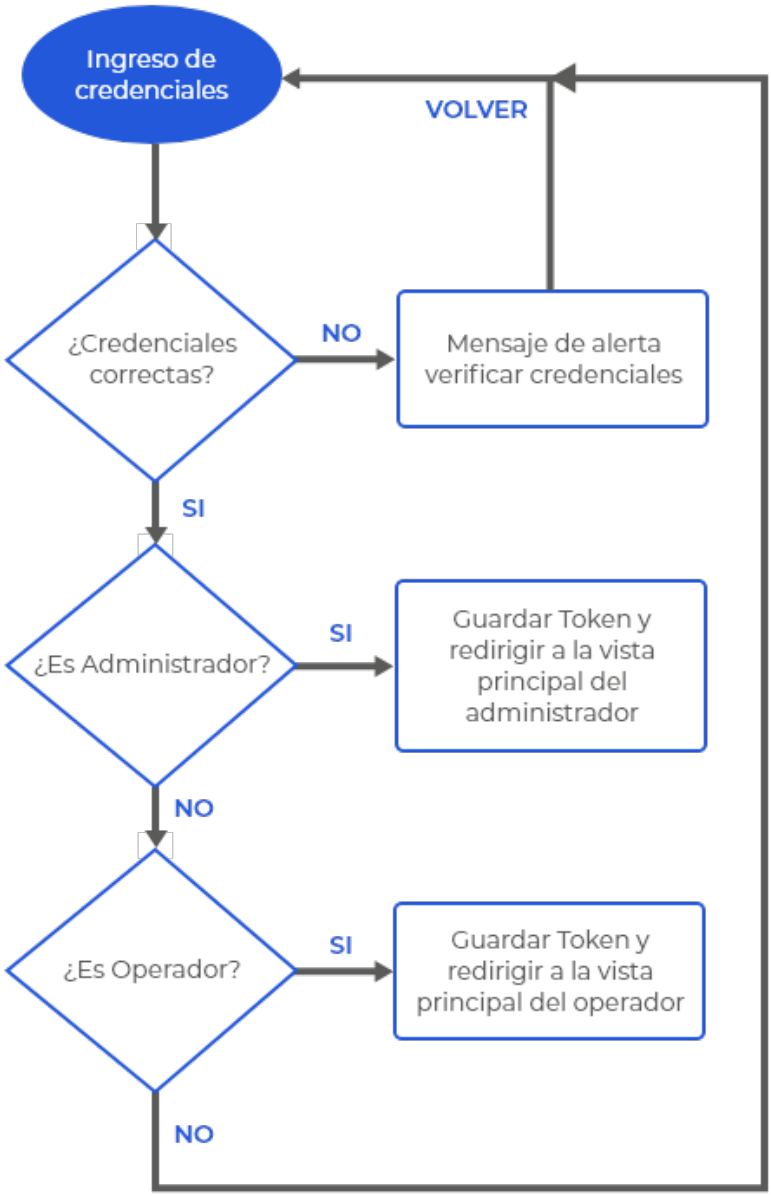


Figura 4.7: Diagrama de flujo del módulo de Login.

4.9.2. Vista Principal

Este módulo, depende principalmente del tipo de usuario, ya sea *Administrador* u *Operario*, se procederá a analizar cada vista respectivamente.

Vista Principal Administrador

Esta vista cuenta principalmente con 5 estados:

- ***DATA***: El cual es un arreglo de objetos en formato JSON, donde cada uno tiene la información de cada contenedor.
- ***STATUS***: Estado del contenedor por el cual se quiera filtrar.
- ***ID_ORDER***: Número de orden por el cual se desea filtrar.
- ***ID_CONTAINER***: Id del contenedor por el cual se quiere filtrar.
- ***DATA_AUX***: Arreglo en formato JSON, en el cual se agregan todos los contenedores que cumplen la regla del filtrado.

Y las siguientes funcionalidades:

- Despliegue de tabla con los datos de los contenedores.
- Filtrado de contenedores según *ID ORDER*, *ID CONTAINER* o *STATUS*.
- Botón para ir a la vista del despliegue de mapa.
- Subir archivo Excel con la información de los contenedores.
- Subir la información de un solo contenedor.
- Crear usuario.
- Cerrar sesión.

Despliegue de tabla

En este módulo, se hace uso del estado *DATA* almacenado en el componente creado en React, llamado *TABLA*.

En el estado *DATA*, se almacena toda la información de los contenedores, donde utilizando nuevamente *Fetch API*, se realiza una *Request* del tipo *GET* a la URL *"http://localhost:4000/api/v1/cargos/"*, en el *Header* se agrega *'Content-Type':'application/json'* nuevamente para indicar que la respuesta será un formato JSON y además se debe añadir el *Token* almacenado al iniciar la sesión. En esta ocasión, el *body* va vacío, ya que, al ser de tipo GET no debemos mandar información, sino, solo esperar la respuesta. Luego con la respuesta de la API, se almacena el nuevo Token recibido y también se actualiza el estado *DATA*.

El componente *TABLA*, primero revisa si es que estado *DATA* tiene información, si es que no tiene, solo se retornan los encabezados de la tabla. Si el estado *DATA* tiene la información ya cargada de la API, se procede a recorrer el arreglo y rellenar el cuerpo de la tabla con la información de cada contenedor.

La información desplegada corresponde a:

- Estado de Alerta: El cual se representa de color verde, rojo o amarillo según sea el tiempo de retraso.
- ID ORDER: Número de la orden.
- ETA: Fecha inicial estimada en llegar al puerto de destino.
- ETD: Fecha en que sale del puerto origen.
- CLIENT: Nombre del cliente del contenedor.
- DESTINATION PORT: Puerto de destino.

- ID CONTAINER: Número del contenedor.
- SHIPPING LINE: Nombre de la naviera.
- VESSEL: Modelo del buque.
- PO: Puerto de Origen.
- STATUS: Estado de la carga, que puede ser:
    - ALL
    - IN PRODUCTION
    - PROGRAMMED
    - DISPATCHED
    - SHIPPED
- ETA UPDATE: Fecha actualizada en llegar al puerto destino.

En la siguiente figura se puede observar el flujo del funcionamiento básico del componente *TABLA*:

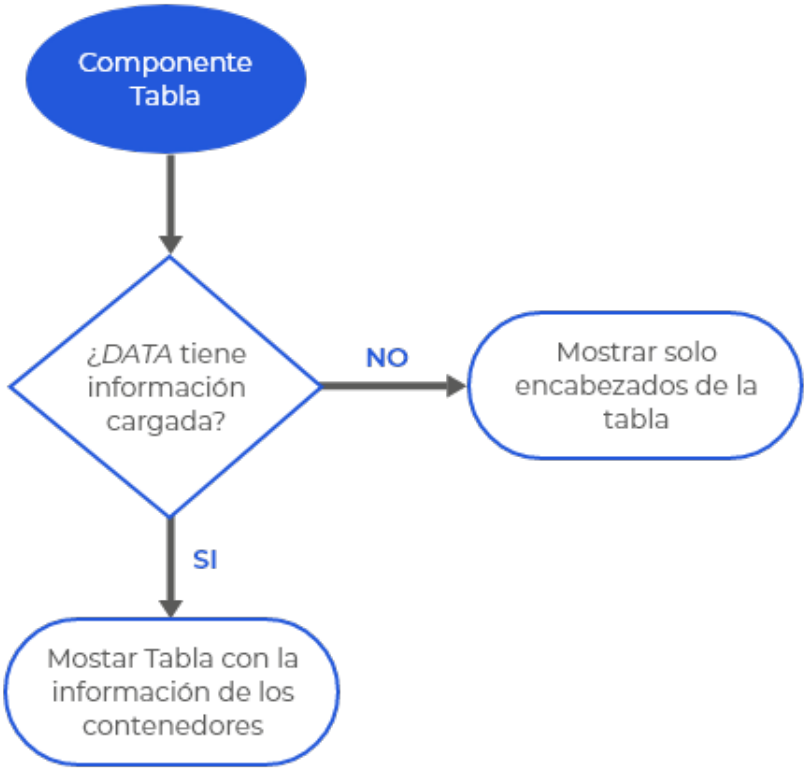


Figura 4.8: Diagrama de flujo del componente Tabla.

Filtrado de Contenedores

En este módulo se utilizan los estados restantes *STATUS*, *ID_ORDER*, *ID_CONTAINER* y *DATA_AUX*.

Al momento en que el usuario presiona el botón "*Search Container*", se procede a recorrer el arreglo del estado *DATA*. Durante este proceso se hace la comparación de

cada contenedor con los estados *STATUS, ID_ORDER e ID_CONTAINER*, y si es que son iguales, se agrega al estado *DATA_AUX*. Luego, aprovechando la característica de React, se le entrega el estado *DATA_AUX* al componente *TABLA*, para que solo muestre los contenedores filtrados. En el caso que no se encuentre, el componente *TABLA* estará vacío desplegando solo los encabezados de la tabla, como se explicó anteriormente.

El siguiente diagrama representa los eventos que se van gatillando para realizar el filtrado de contenedores:

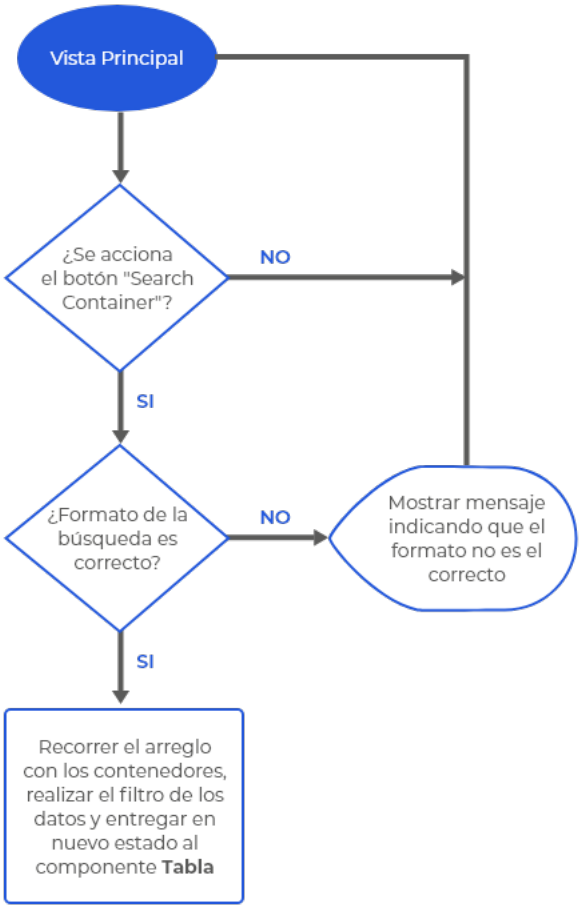


Figura 4.9: Diagrama de flujo del filtrado de contenedores.

Subir archivo Excel

Para este funcionamiento, bajo el perfil *Administrador*, se procede a seleccionar el botón *Upload Info*, el cual abre una ventana emergente, donde se puede seleccionar la posibilidad de cargar un archivo Excel. Para realizar esta acción, se crea un nuevo componente en React, llamado "***ExcelComponent***", el cual posee los siguientes estados:

- ***selectFile***: es el archivo Excel que se subirá.
- ***isFormInvalid***: variable que indica si es que el archivo seleccionado efectivamente es de formato Excel.

Si es que al momento de seleccionar el archivo, este no es de formato Excel, se procederá a mostrar un mensaje indicando que el formato del archivo es incorrecto. En el caso contrario, se podrá seleccionar el botón *Upload*, lo cual gatillará la función que se conectará con la API que permite subir el archivo.

La URL para poder subir el archivo es *"http://localhost:4000/api/v1/cargos/upload_information"*, esta vez para usar la API, se utiliza la biblioteca de React *Axios*, pensada para facilitar el consumo de servicios web que devuelvan datos JSON además de ser muy sencilla de utilizar, Axios funciona como un objeto, que para seleccionar el tipo de *Request*, se llama a un método, en este caso como queremos subir un archivo, se utiliza *POST*, entregando como parámetros la URL de la API, el archivo a subir y un *Headers* que contiene el tipo de contenido *'Content-Type': 'application/json'* de igual manera que usando *API Fetch* más el respectivo Token de seguridad.
Si la respuesta de la API es correcta, se procede a guardar el nuevo Token y desplegar un aviso con la biblioteca *SweetAlert* que el archivo se subió de forma correcta.
En el caso contrario, se desplegará un aviso con SweetAlert, informando que se produjo un problema a subir el archivo.

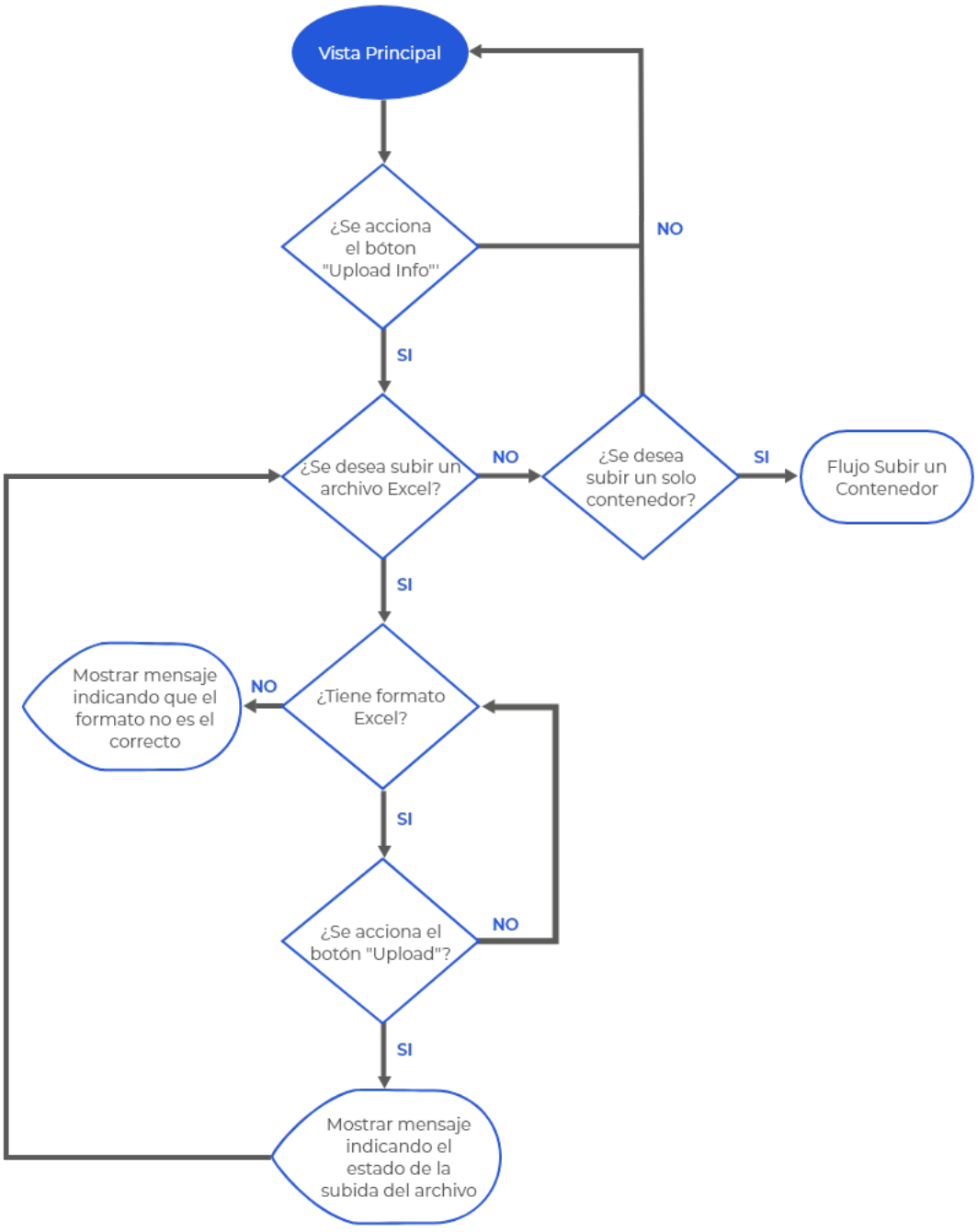


Figura 4.10: Diagrama de flujo de la subida del archivo Excel.

Subir un Contenedor

Al seleccionar el botón *Upload Info*, también existe la opción de subir la información de un solo contenedor, al seleccionar el botón *ADD ROW* se procede a abrir un nuevo componente de React *CONTAINERFORM* en otra ventana emergente, el cual contiene los siguientes estados que el mismo usuario deberá agregar, llenando el formulario desplegado:

- ***ID ORDER***: Número de la orden.
- ***ETA***: Fecha inicial estimada en llegar al puerto destino.
- ***ETD***: Fecha que sale del puerto origen.
- ***CLIENT***: Nombre del cliente del contenedor.
- ***DESTINATION PORT***: Puerto de destino.
- ***ID CONTAINER***: Número del contenedor.
- ***SHIPPING LINE***: Nombre de la naviera.
- ***VESSEL***: Modelo del buque.
- ***PO***: Puerto de Origen.
- ***STATUS***: Estado de la carga, que puede ser:
    - ALL
    - IN PRODUCTION
    - PROGRAMMED
    - DISPATCHED
    - SHIPPED

Luego de llenar todo el formulario y apretar el botón *ADD CONTAINER*, se utiliza API Fetch, para realizar una Request del tipo POST a la URL *"http://localhost:4000/api/v1/cargos/"*, agregando en el header el Token y que la respuesta será del tipo JSON, y por último en el Body se añaden todos los estados anteriormente descritos.

Con la respuesta de la API, se actualiza el Token y despliega un aviso con la ayuda de SweetAlert, que la información ha sido subida correctamente. En el caso contrario, se muestra un aviso que hubo un problema.

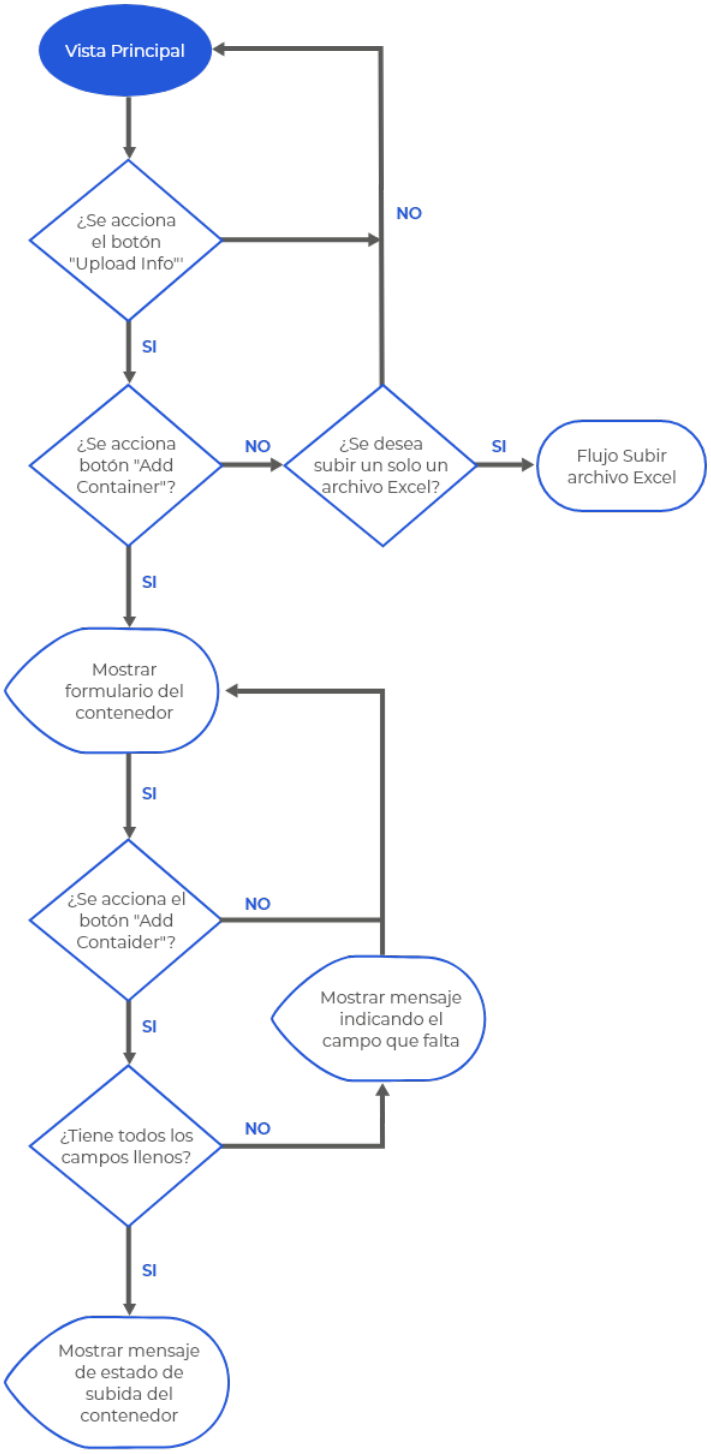


Figura 4.11: Diagrama de flujo de la subida de un contenedor.

Crear un nuevo usuario

Al seleccionar el botón *Crear Usuario*, se procede a abrir un nuevo componente de React *SignUpForm* en otra ventana emergente, el cual contiene los siguientes estados que el mismo administrador deberá agregar, llenando el formulario desplegado:

- ***Name***: Nombre del usuario.
- ***LastName***: Apellido del usuario.
- ***Email***: Email de ingreso para el usuario.
- ***Password***: Contraseña del usuario.
- ***isAdmin***: El cual indica si es que es Administrador u operario el nuevo usuario.

Luego de llenar todo el formulario y apretar el botón *Sign Up*, se utiliza API Fetch, para realizar una Request del tipo POST a la URL *"http://localhost:4000/api/v1/users"*, agregando en el header el Token y que la respuesta será del tipo JSON, y por último en el Body se añaden todos los estados anteriormente descritos.
Con la respuesta de la API, se actualiza el Token y despliega un aviso con la ayuda de SweetAlert, que el usuario ha sido añadido correctamente. En el caso contrario, se muestra un aviso que hubo un problema.

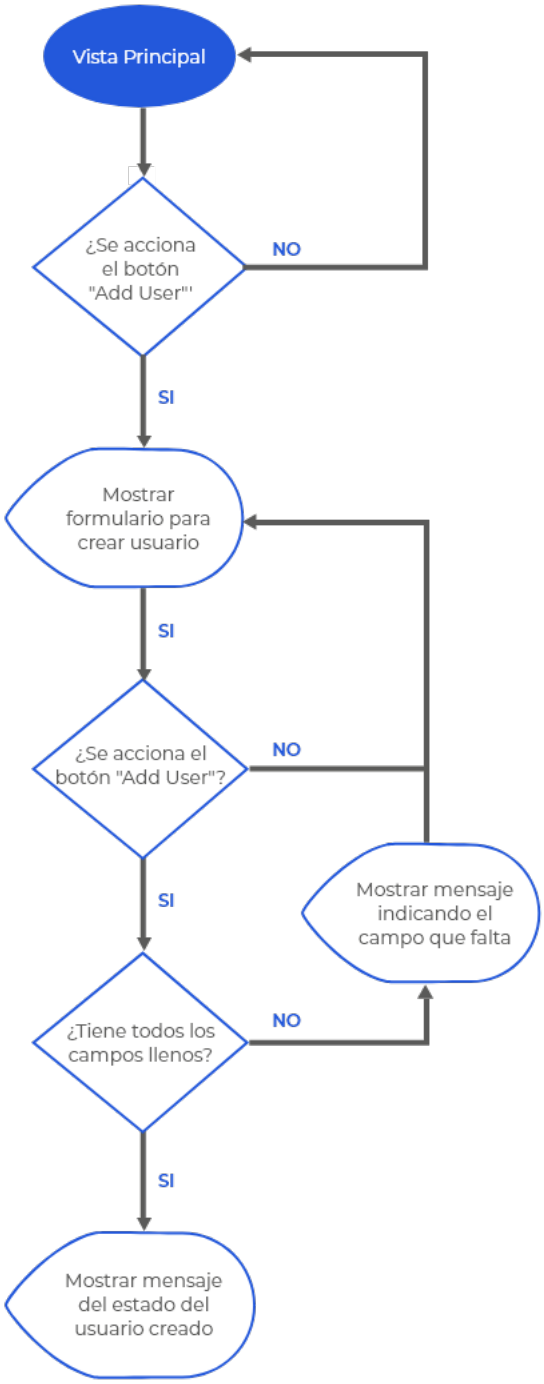


Figura 4.12: Diagrama de flujo de la creación de un usuario.

Cerrar Sesión

En este módulo, al seleccionar la opción *cerrar sesión*, se procede a eliminar los datos de sesión almacenados, redirigiéndose a la vista del Login, como se puede ver en la Fig. 4.13 .

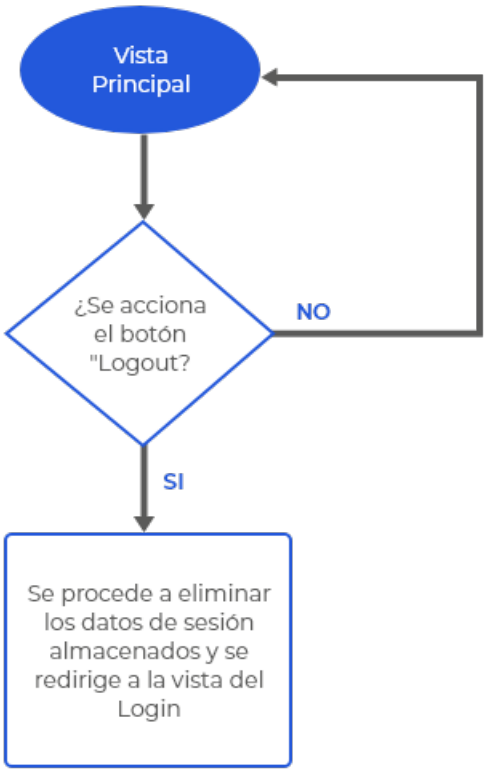


Figura 4.13: Diagrama de flujo de cerrar sesión.

Despliegue del mapa

Al presionar el botón con del mundo, se procede a redirigir a la vista del mapa, el que se explica en la siguiente sección, por ultimo como se puede ver en la Fig. 4.14.

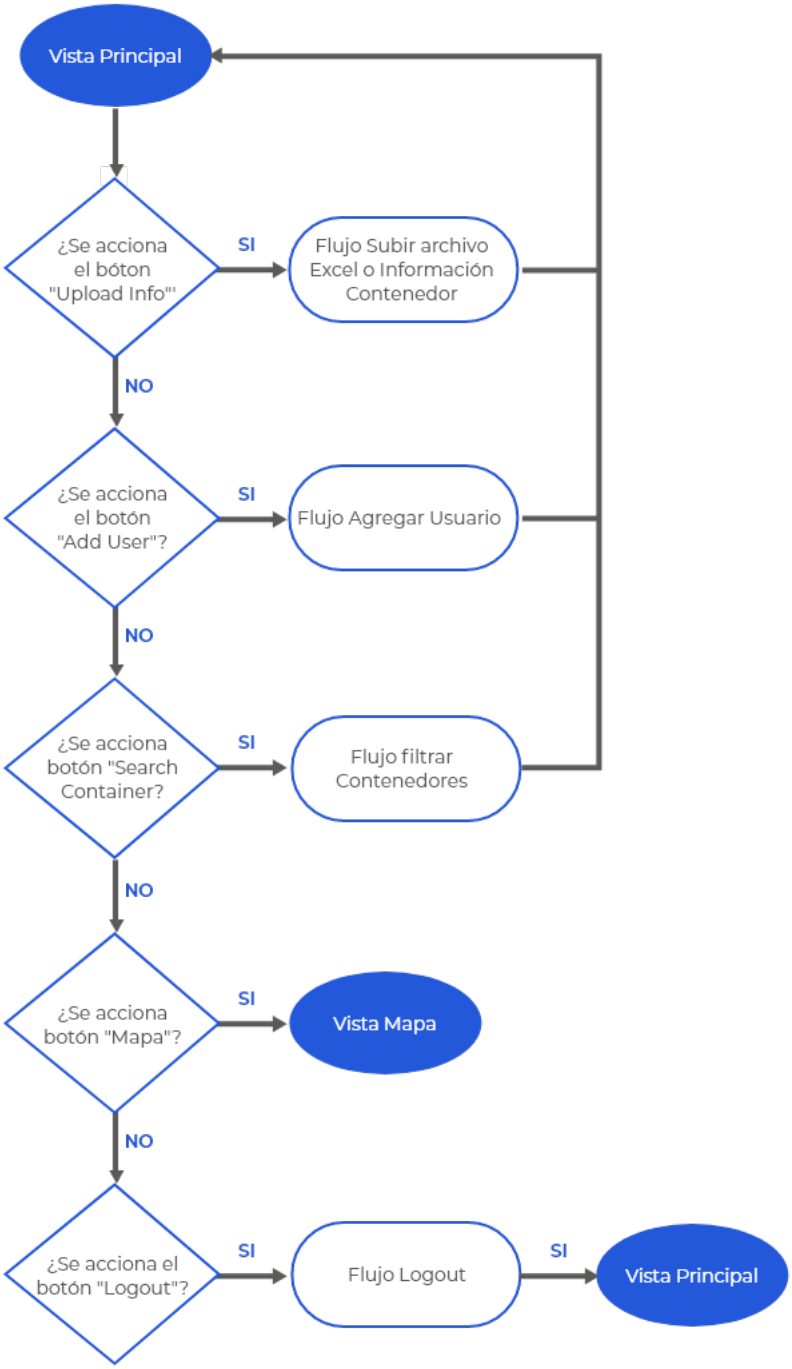


Figura 4.14: Diagrama flujo de la vista principal Administrador.

Vista Principal Operario

Esta vista es igual que la vista del Administrador con los mismos estados, pero con solo 4 funcionalidades:

- Despliegue de tabla con los datos de los contenedores.
- Filtrado de contenedores según *ID ORDER*, *ID CONTAINER* o *STATUS*.
- Botón para ir a la vista del despliegue del mapa.
- Cerrar sesión.

Las cuales funcionan de la misma forma anteriormente descrita, la única diferencia con la Vista principal del Administrador, es que esta vista no posee los botones para subir información ni para crear un nuevo usuario, en la Fig. 4.15 se muestra el flujo de esta vista .

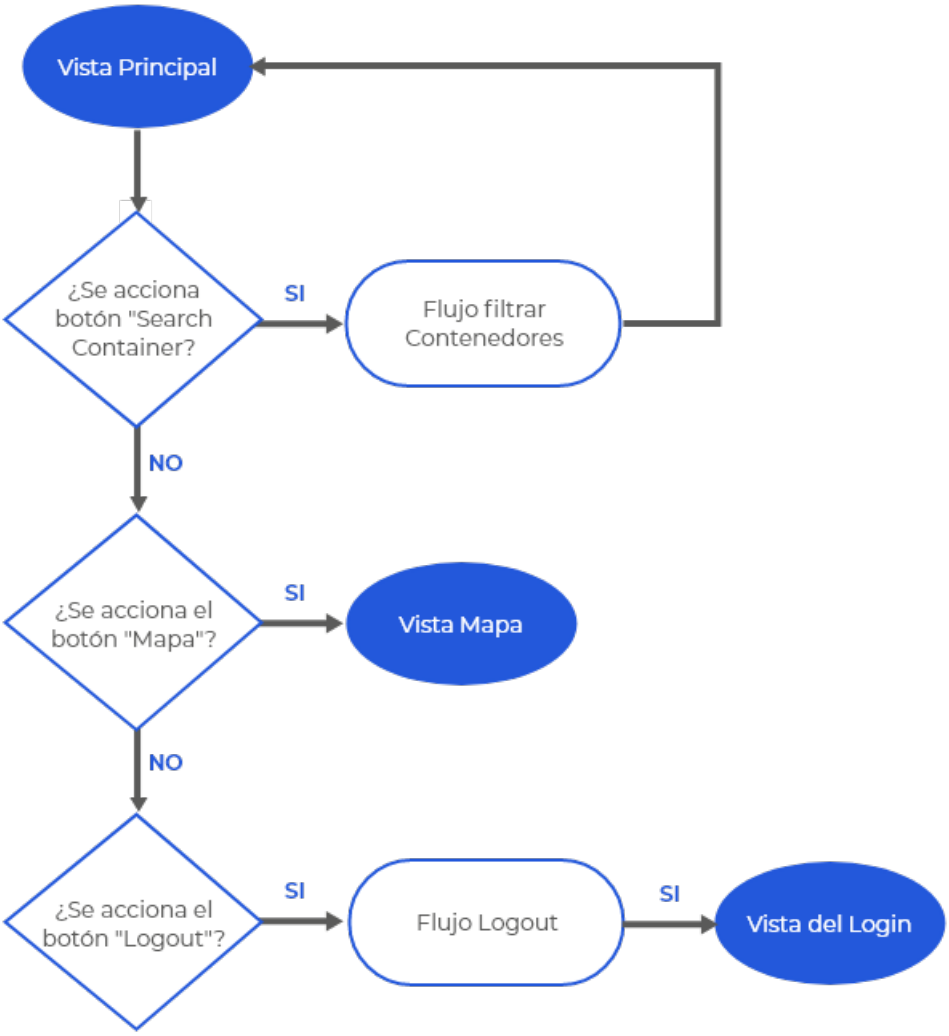


Figura 4.15: Diagrama flujo de la vista principal Operario.

4.9.3. Vista Mapa

Este modulo, es la vista al seleccionar el botón con el icono del mundo, la vista desplegada es la misma tanto en el operador como en el administrador, por lo que el análisis de los componentes será la misma en ambas, esta vista principal consta de 3 componentes principales, 9 variables de estados y 4 funcionalidades.

Las variables de estados son:

- Data: El cual es un arreglo de objetos en formato JSON, en el cual cada uno tiene la información de cada contenedor.
- STATUS: Estado del contenedor por el cual se quiera filtrar.
- ID_ORDER: Número de orden por el cual se quiere filtrar.
- ID_CONTAINER: Número de orden por el cual se quiera filtrar.
- DATA_AUX: Arreglo en formato JSON, en el cual se agregan todos los contenedores que cumplen la regla del filtrado.
- CONTAINER_DISPLAY: Información del contenedor que se está desplegando.
- DELAY: Cantidad de contenedores con atraso.
- ALMOST_TIME: Cantidad de contenedores que están con atraso pero aún no es grave.
- INTIME: Cantidad de contenedores que están a tiempo.

En esta vista se evidencian 4 funcionalidades:

- Despliegue de los contenedores en el mapa.
- Filtrado de contenedores según *ID ORDER*, *ID CONTAINER* o *STATUS*.
- Despliegue de contador para los estados de atrasos de los contenedores.
- Mostrar información al seleccionar un contenedor.

Despliegue de los contenedores en el mapa

Para realizar esta funcionalidad se utilizó la biblioteca de react *google-maps-react*, gracias a esta biblioteca podemos utilizar *Google Maps API* y así poder desplegar el mapa en la vista. Con esta biblioteca se realizó uno de los tres componentes que se utilizan en esta vista, llamado *Container_map*, este componente es el encargado de:

- Mostrar los contenedores en el mapa.
- Enviar la información sobre el contenedor que se selecciona.
- Pintar los contenedores según el grado de atraso.

Al momento de iniciar la Vista Mapa, en el estado *DATA*, se almacena toda la información de los contenedores, donde se utiliza nuevamente *Fetch API*, se realiza un Request de tipo *GET* a la URL *"http://localhost:4000/api/v1/cargos/"*, en el *Headers* se debe agregar *'Content-Type':'application/json'* nuevamente para indicar que la respuesta será un formato JSON y además se debe añadir el *Token*. En esta ocasión, el *body* va vacío.
Luego con la respuesta de la API, se almacena el nuevo Token recibido y también se actualiza el estado *DATA* que es entregado como parámetro al componente *Containers-Map*.
En el componente *Containers-Map*, se agrega la biblioteca de Google Maps API que posee un componente llamado *Marker*, que despliega un marcador en la *LATITUD* y *LONGITUD* entregadas como parámetros, los cuales están disponibles en el estado *DATA*, el cual de además de tener la información de los contenedores, contiene las coordenadas correspondiente a cada uno.
Se realiza una función *displayMarkers*, que usando el estado *DATA*, se procede a recorrerlo, a cada contenedor se procede a comparar la variable *ESTADO_ALARMA*, dependiendo del valor si es *Rojo*,*Amarillo* o *Verde*, se procede a colocar el *Marker* con una imagen de tipo *.SVG*, cuya cualidad es mantener sus proporciones de tamaño sin perder la calidad de la imagen, esta imagen solo varia el color según corresponda.

Para concluir, se utiliza el componente *Map* que necesita como parámetro y el zoom que poseerá el mapa, coordenadas iniciales y se agrega la función *displayMarkers* que retorna los marcadores que representan la posición de los contenedores, comos e ve en la Fig.4.16 .

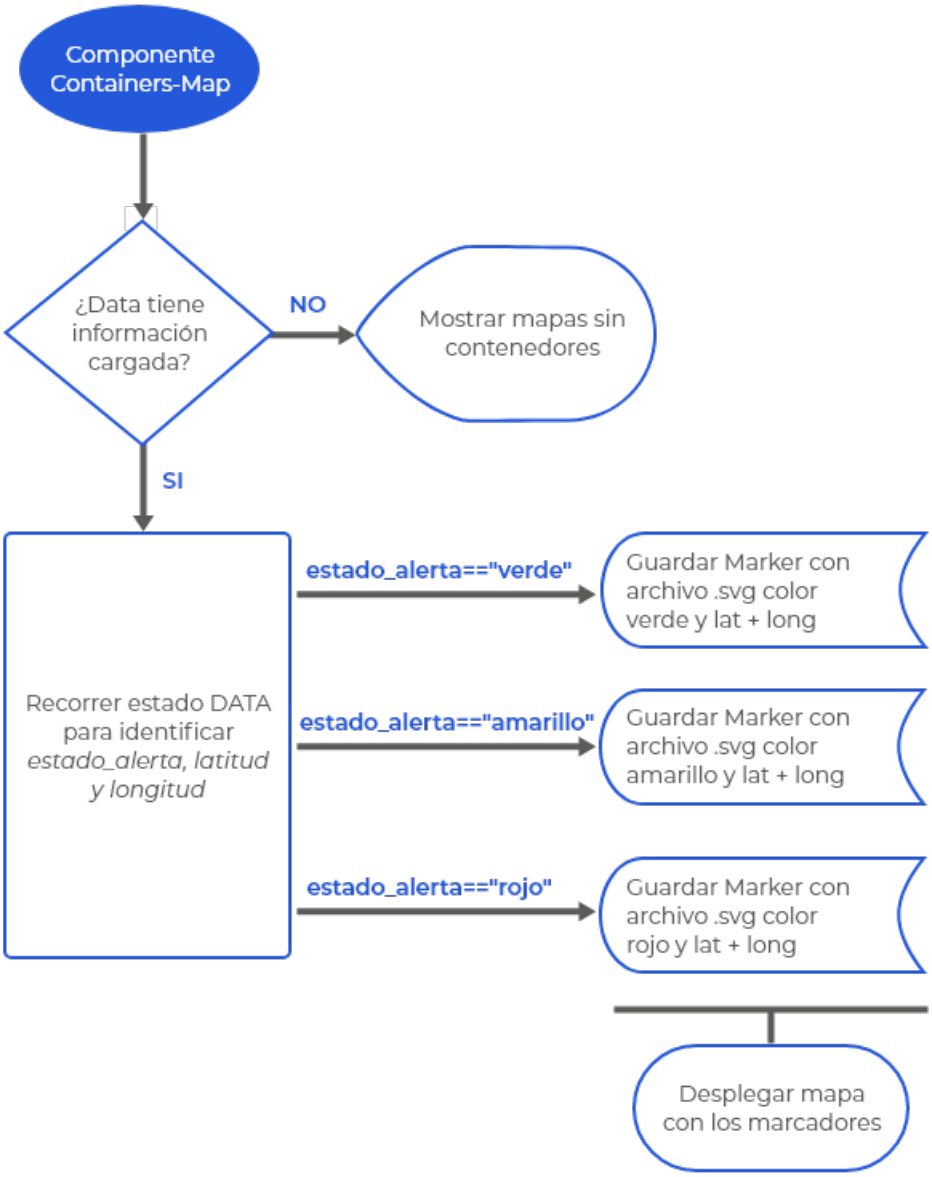


Figura 4.16: Diagrama de flujo para desplegar los contenedores.

Filtrado de contenedores

En esta funcionalidad, aprovechamos la principal característica de React, la reutilización de componentes, por lo que se vuelve a usar el componente creado en la vista principal, de la misma forma, utilizando los estados *STATUS*,*ID_CONTAINER*, *ID_ORDER*, *DATA_AUX* con la misma lógica de la figura 4.9.

Contador de atrasos de contenedores

En esta vista, se decidió agregar tres contadores, que corresponden a los estados *DELAY*, *ALMOST_TIME*, *INTIME* los cuales indican la cantidad de contenedores que están atrasados, con un leve atraso y los que aún están a tiempo.
Para llevar acabo esta funcionalidad se creó un componente, *displayInfo*, el cual dentro de este contiene 2 componentes más:

- *INFODELAY*: Contiene los contadores de los estados de retraso de los contenedores.

- *INFOCONTAINER*: Contiene la información del contenedor seleccionado en el mapa, este se explicará después con mayor profundidad.

Al momento de iniciar la vista mapa, en los estados *DELAY*, *ALMOST_TIME* y *INTIME*, se almacena la información correspondiente al estado de los contenedores, donde, utilizando nuevamente *Fetch API*, se realiza un Request de tipo *GET* a la URL *"http://localhost:4000/api/v1/cargos/estado_alarmas/"*, en el *Headers* se debe agregar *'Content-Type':'application/json'* nuevamente para indicar que la respuesta será un formato JSON y además se debe añadir el *Token*, nuevamente el *body* va vacío.
Luego con la respuesta de la API, se almacena el nuevo Token recibido y también se actualiza los estados *DELAY*, *ALMOST_TIME* y *INTIME* que son entregados como parámetros al componente *INFODELAY*.

Figura 4.17: Diagrama de flujo de contador de atrasos de contenedores.

Información del contenedor seleccionado

Para esta funcionalidad se utiliza el estado *CONTAINER_DISPLAY*, el cual almacena la información del contenedor seleccionado en el mapa.

En el componente *Container_map*, se hace un llamado a una función creada en Vista mapa llamada *UPDATE_INFOCONTAINER*, esta función es la encargada de almacenar la información del contenedor seleccionado, la cual recibe como parámetro un contenedor y luego actualiza el estado *CONTAINER_DISPLAY*.

Con el método *onClick()* de react, la cual sirve para accionar una función al momento de escuchar el evento de hacer *click*, retornando la información que esta alojada en el componente *Container_map*.

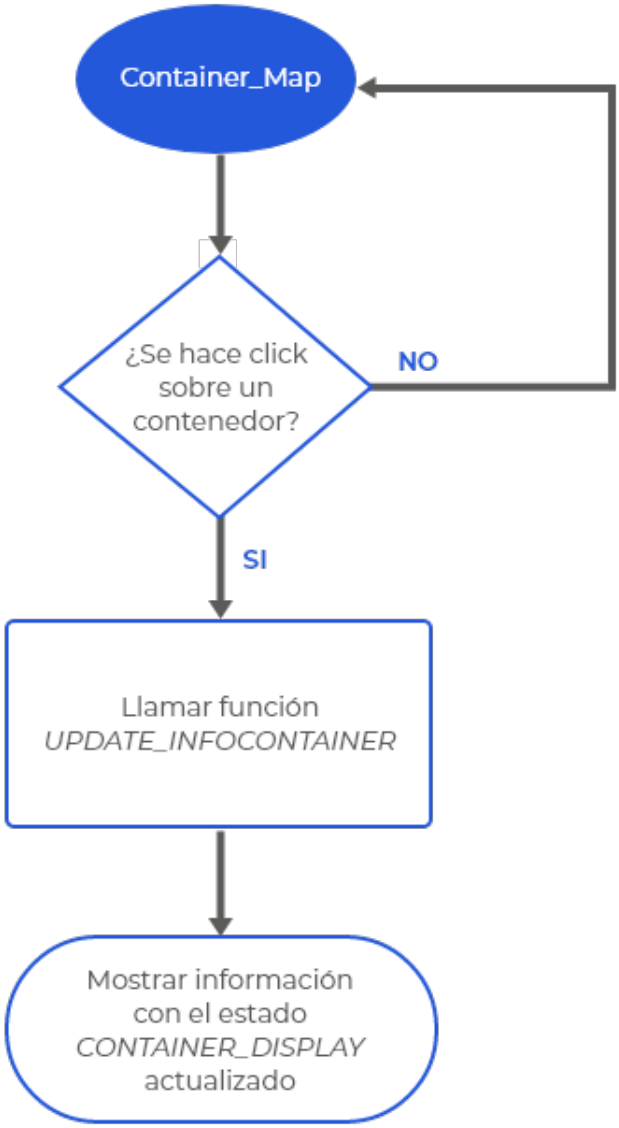


Figura 4.18: Diagrama de flujo de la información del contenedor seleccionado.

Por lo que, el flujo general de la vista del mapa, quedaría de la siguiente forma:

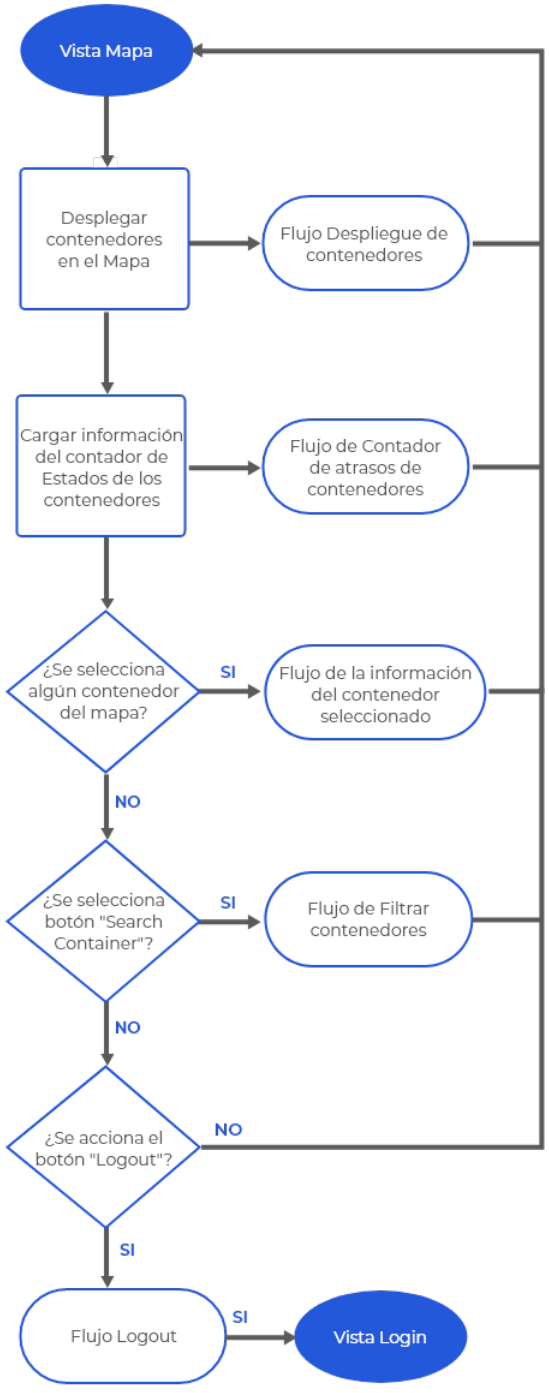


Figura 4.19: Diagrama de flujo de la vista del mapa.

4.10. Resumen

A partir de la tabla 3.1 expuesta en la sección anterior, podemos comparar nuestra propuesta con lo que existe actualmente en el mercado, mostrando las ventajas y desventajas que existen.

	SmartTracking	Searates	Arviem's	Ocean Insights
Permitir busqueda de un contenedor	✓	✓	✓	✓
Dashboard de administración	✓	✓	✓	✓
Despliegue de Mapa	✓	✓	✓	✓
Actualización de ETA	✓	✓	✓	✓
Interfaz amigable	✓	✓	✓	✓
Monitoreo de Temperatura			✓	✓
Aplicación Movil		✓		
Alertas	✓		✓	
Integración con webs navieras	✓		✓	
Multiples perfiles	✓			
Carga de archivos en formato .xlsx	✓			

Cuadro 4.9: Comparación SmartTracking vs otras ofertas del mercado

La principal ventaja de nuestro producto con el de la competencia, es que este fue pensado para adaptarse a las herramientas y necesidades del publico objetivo(empresas que realizan exportaciones de productos), muchas empresas que trabajan con exportaciones guarda su información en sistemas del plantillas excel, por lo que una integración con nosotros seria mucho mas rápida, a su vez el contar con un sistema de alarmas le da al cliente la facilidad de centrarse solo en sus pedidos con algún tipo de problema y poder reaccionar a tiempo.

Capítulo 5

Resultados

En esta sección se muestra los resultados al desarrollar en el Framework ReactJS, el Login en el cual el operario o administrador pueden ingresar a su página correspondiente con sus credenciales personales, luego se les redirigen a sus vistas asignadas para que puedan realizar sus funcionalidades respectivas en donde el administrador puede cargar la información de los contenedores necesarios para que así el operador pueda visualizarlos en su vista propia, en la cual el puede ver el estado de los contenedores.
Además ambos usuarios pueden realizar filtrados, con el cual pueden buscar algún cargamento en específico a partir del ID de la orden, estado de la carga o el ID del contenedor.
También, tanto el operario como el administrador, pueden acceder a la vista del mapa para observar de manera gráfica y aproximada donde se encuentra actualmente el contenedor.
La vista del mapa, posee información del estado de arribo de todos los cargamentos indicando si están atrasados, en su tiempo estimado o a punto de llegar tarde.
Igualmente en esta vista, se puede contemplar la información específica de un solo contenedor gracias a la funcionalidad de filtrado, y al momento de hacer *click* sobre el cargamento en el mapa, se despliega al costado, toda la información de este.

5.1. Login

En la figura 5.1, es la vista inicial al momento de entrar a la aplicación *SmartTracking*, donde deben estar completas sus credenciales con su *Email Adress* y *Password*, dependiendo de estos, se pasará a la siguiente vista según sea *Administrador* u *Operario*, además en esta vista, se agrega una imagen que hace referencia de que se trata la aplicación, para inspirar confianza, seguridad y dar a entender que sabemos los inconvenientes que posee la logística del transporte de contenedores.

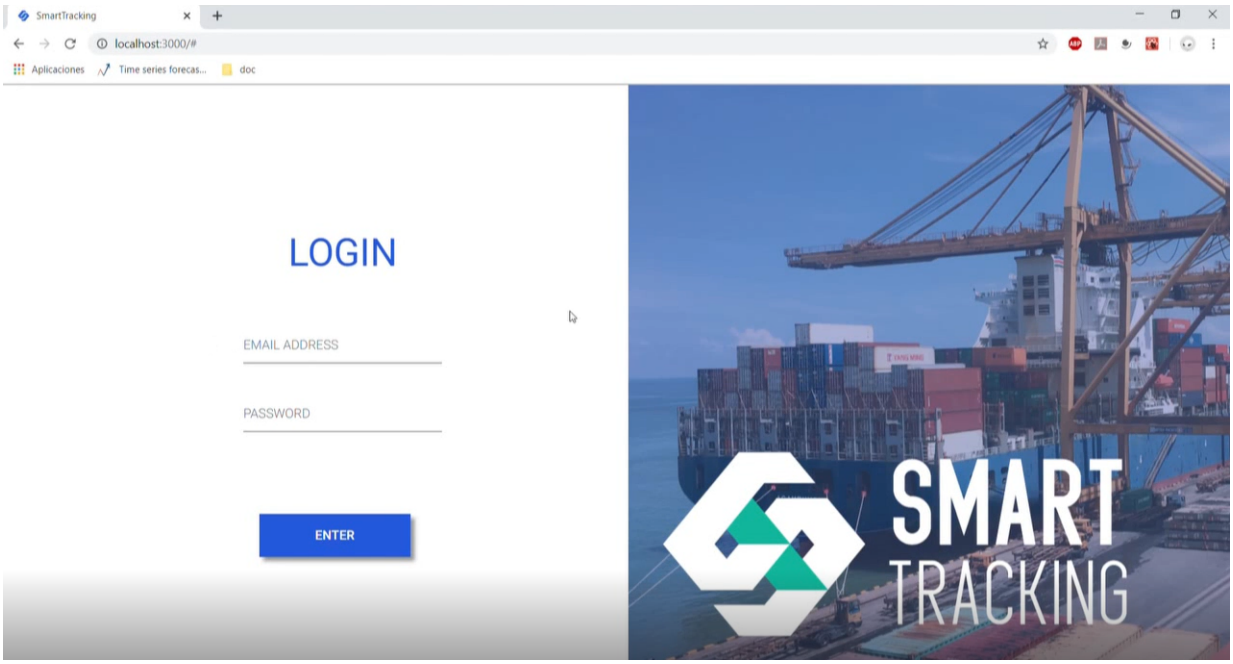


Figura 5.1: Desarrollo del Login.

5.2. Vista Principal

Administrador

Al momento de comprobar que las credenciales son correctas, se hace la redirección a la Figura 5.2, en un comienzo no se despliega ningún contenedor, ya que no hay contenedores cargados, por lo que en la Figura 5.3 se procede a subir un archivo Excel con la información de los contenedores y además en la Figura 5.4 se realiza la subida de un solo contenedor, evidenciando el funcionamiento de cargar información a la plataforma. Luego en la Figura 5.5 se puede comprobar con el despliegue de la información de los contenedores. Por último en la Figura 5.6, se procede a crear un nuevo operario para que pueda ingresar a la plataforma.

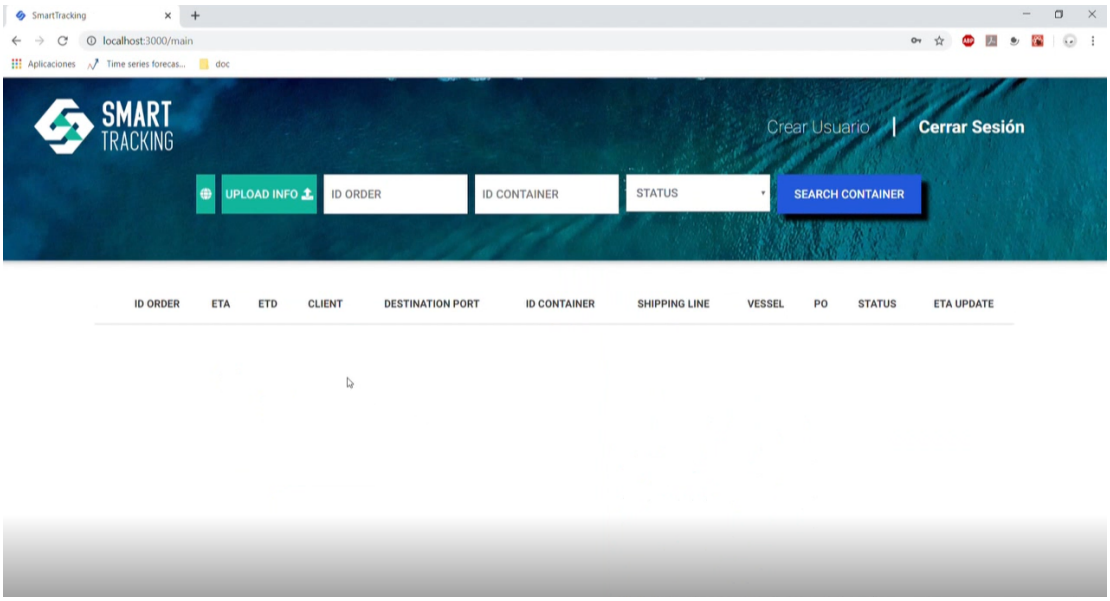


Figura 5.2: Vista principal al iniciar como administrador.

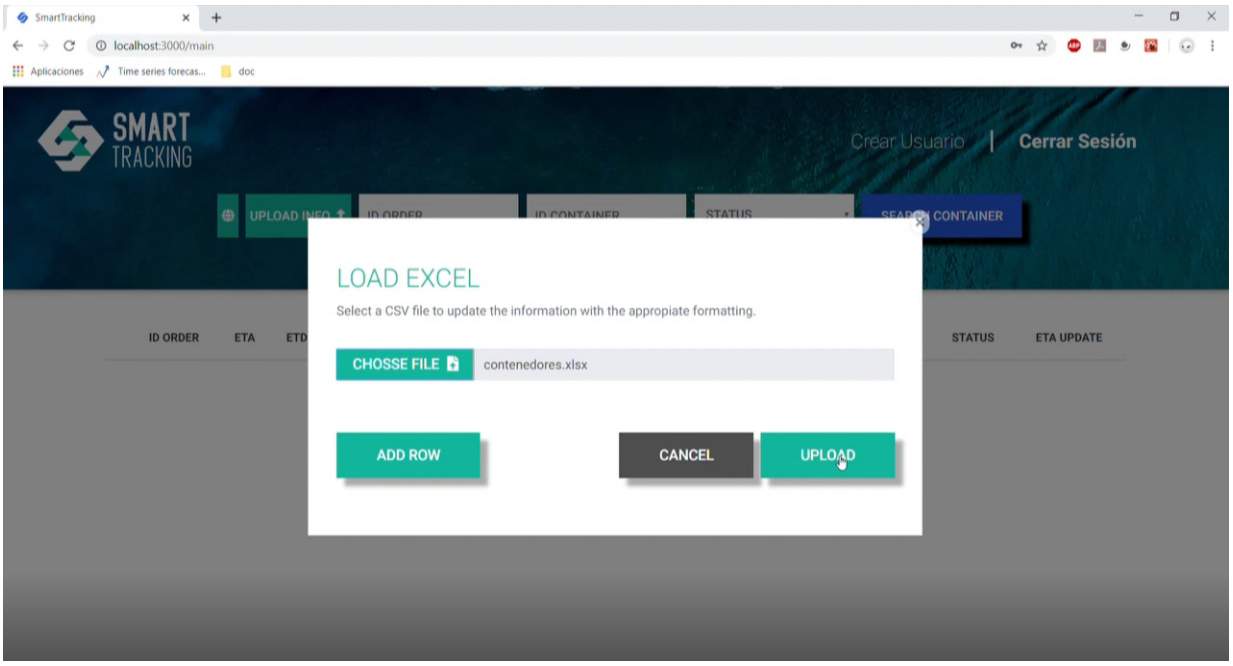


Figura 5.3: Subida archivo Excel en la Vista Principal del Administrador.

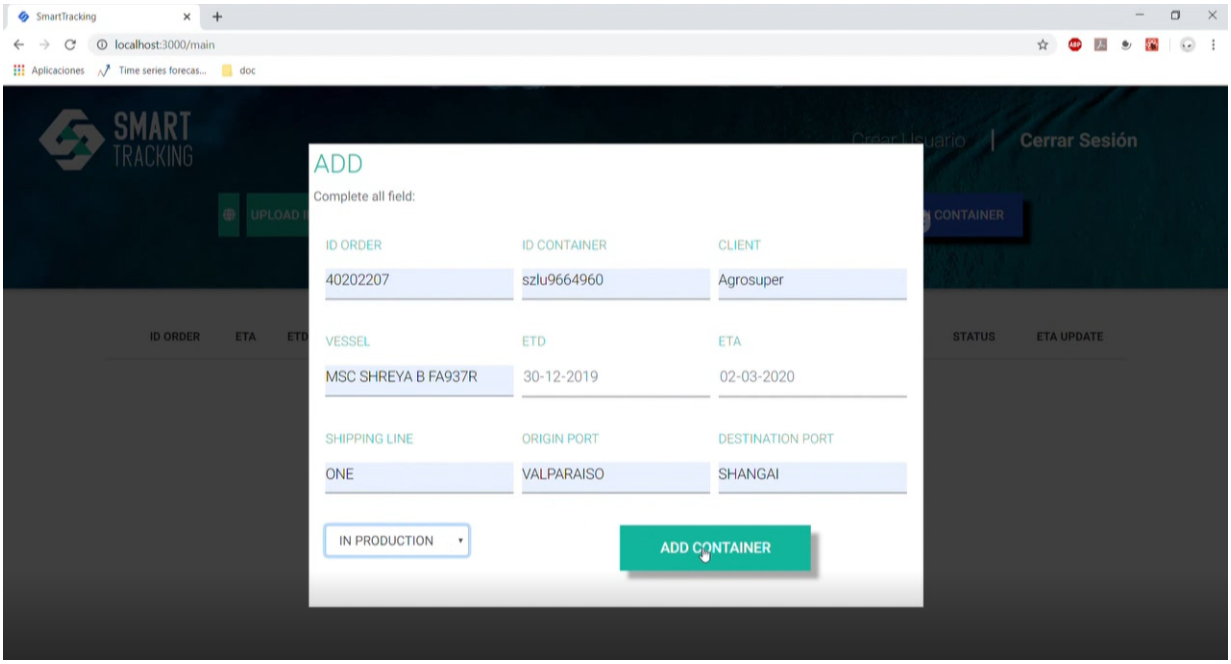


Figura 5.4: Subida de un contenedor en la Vista Principal del Administrador.

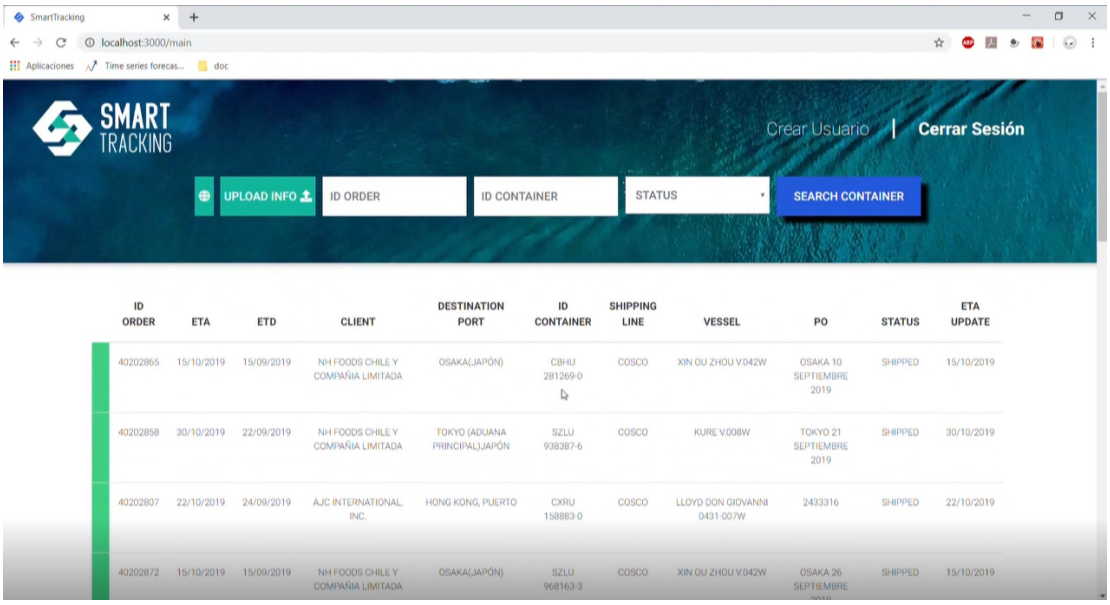


Figura 5.5: Tabla con datos en la Vista Principal.

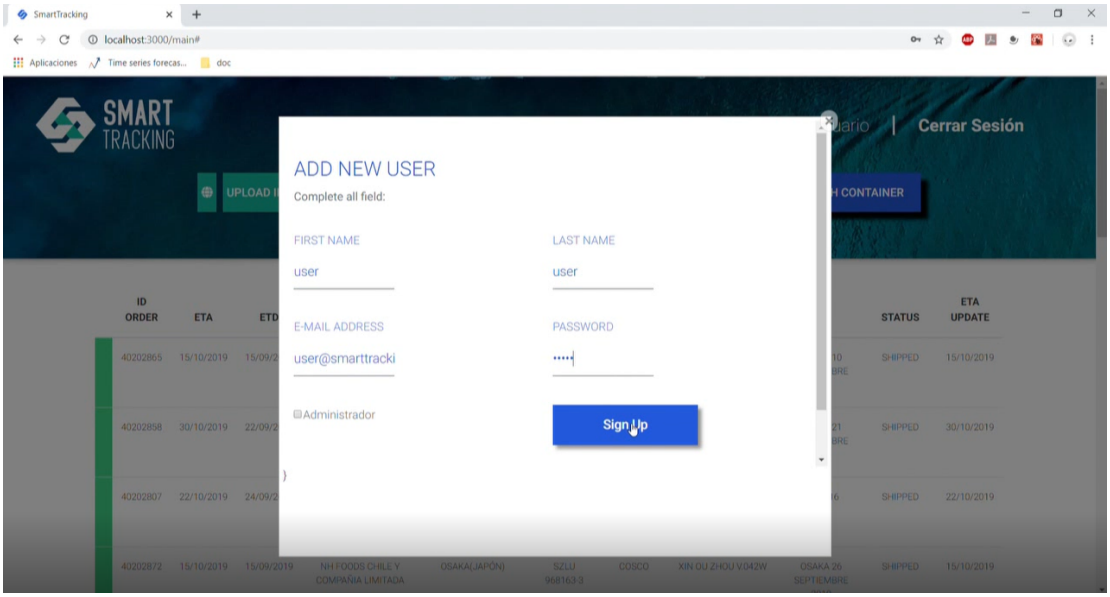


Figura 5.6: Creando a un operario en Vista Principal del Administrador.

Operario

Al momento de seleccionar *Cerrar Sesión*, se redirige al Login, con el fin de poder iniciar sesión como Operario.

Cuando el operario inicia sesión, se redirecciona a la Figura 5.7, en la cual se puede evidenciar la ausencia del botón *UPLOAD INFO*, *Crear Usuario*, ya que por sus privilegios de usuario, no tiene acceso a esta vista.

También se muestra de inmediato la información cargada por el Administrador en la tabla, la cual gracias al botón *SEARCH CONTAINER* se puede filtrar el o los contenedores que se desean ver en la tabla, como se ve en la Figura 5.8.

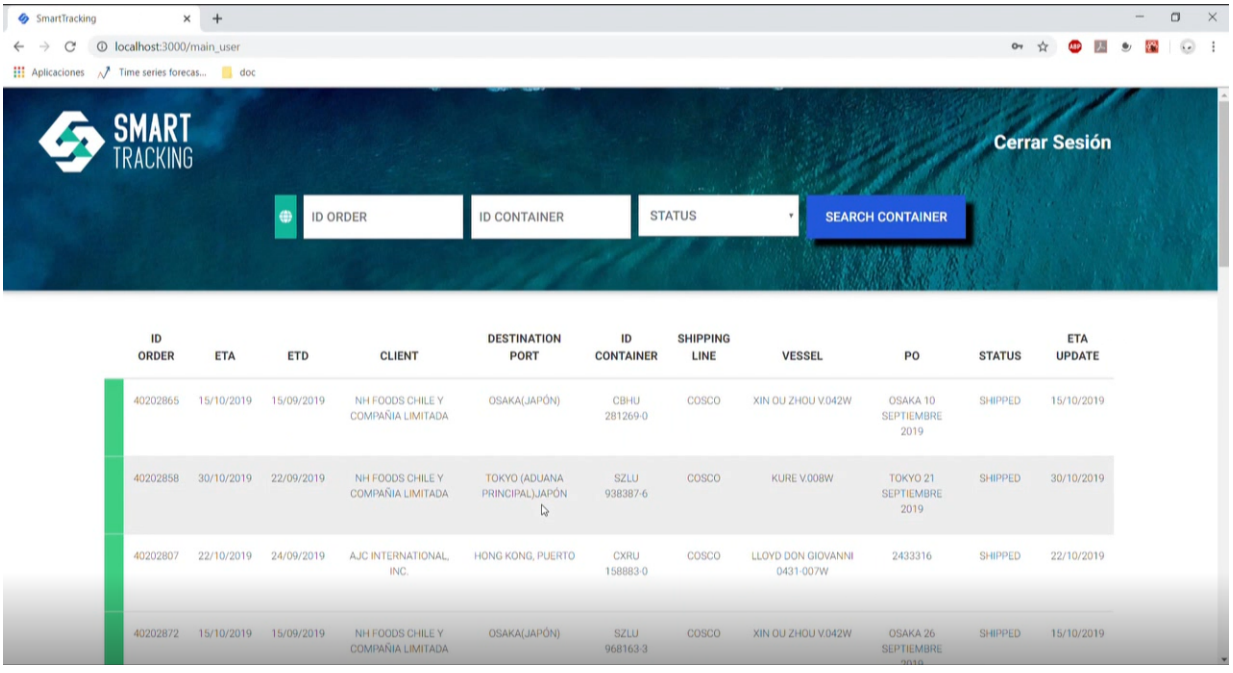


Figura 5.7: Vista principal al iniciar como Operario.

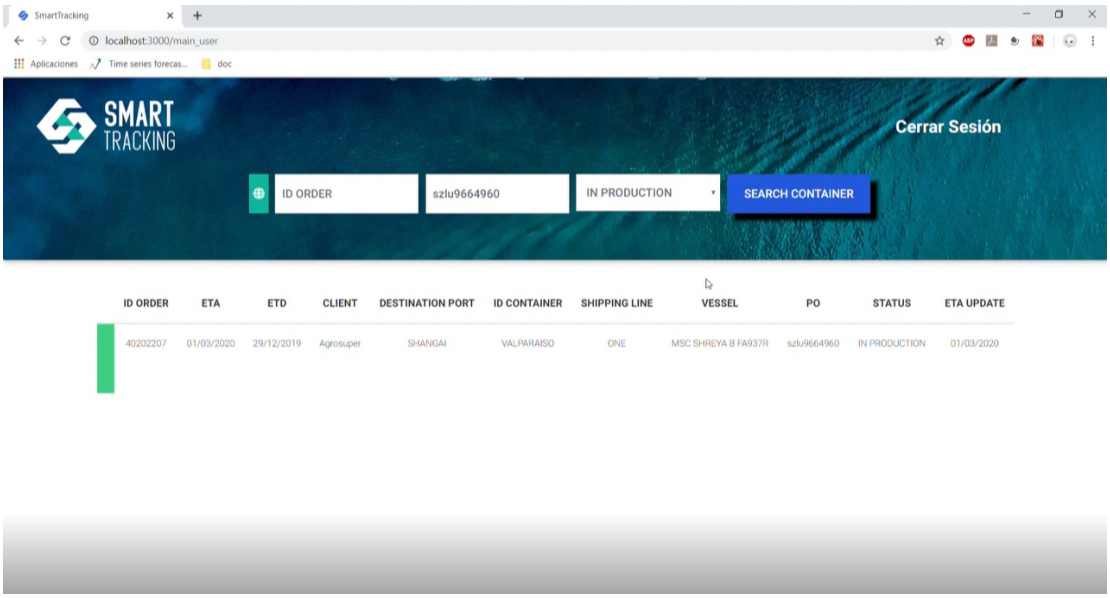


Figura 5.8: Filtrado de contenedor en la Vista Principal.

5.3. Vista Mapa

Esta vista está disponible tanto en la vista principal del operario, como en la vista principal del administrador, la cual tienen acceso gracias al primer botón de izquierda a derecha de color verde con un icono del mundo, al accionarlo, produce la redirección a la Figura 5.9.

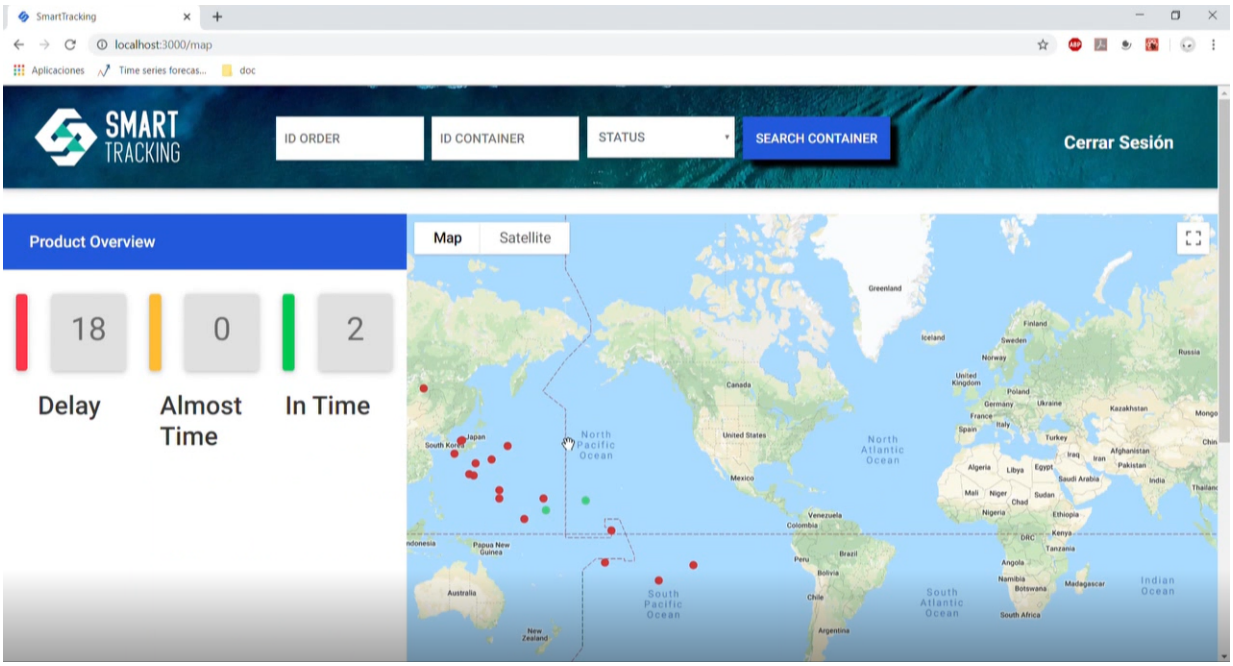


Figura 5.9: Vista Mapa para Administrador y Operario.

En la Figura 5.9, se cargan inmediatamente los contenedores en el mapa, pudiendo ver el estado de atraso, el lugar aproximado donde van y además se cargan los contadores de atraso de los contenedores.

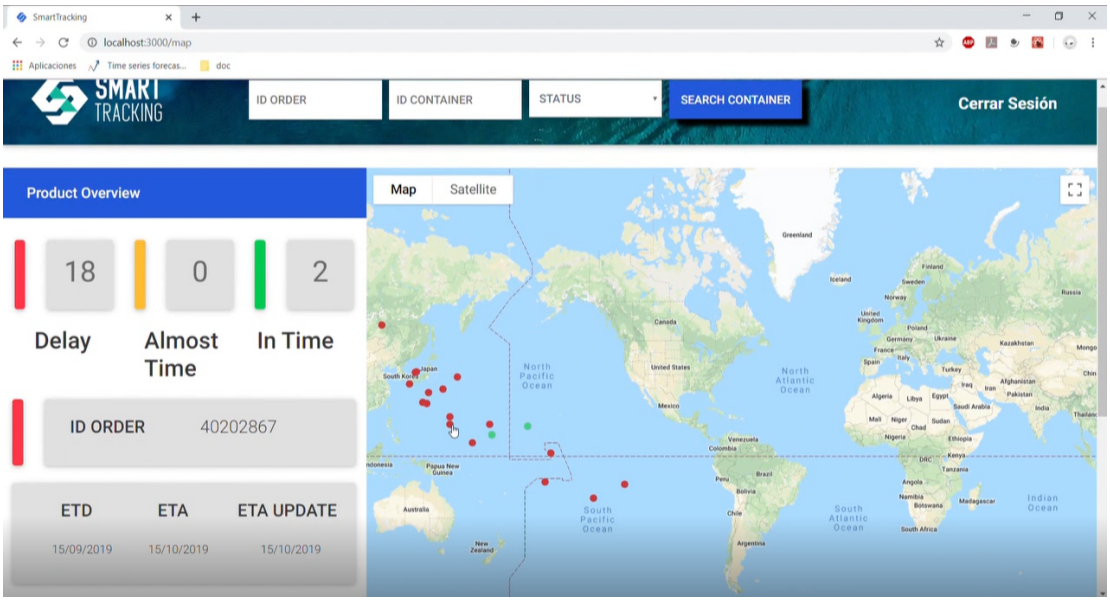


Figura 5.10: Seleccionar un contenedor en el mapa.

Al momento de seleccionar un contenedor dentro del mapa, se cargan los datos de este, desplegándose toda su información, como se ve en la Figura 5.10 y 5.11.

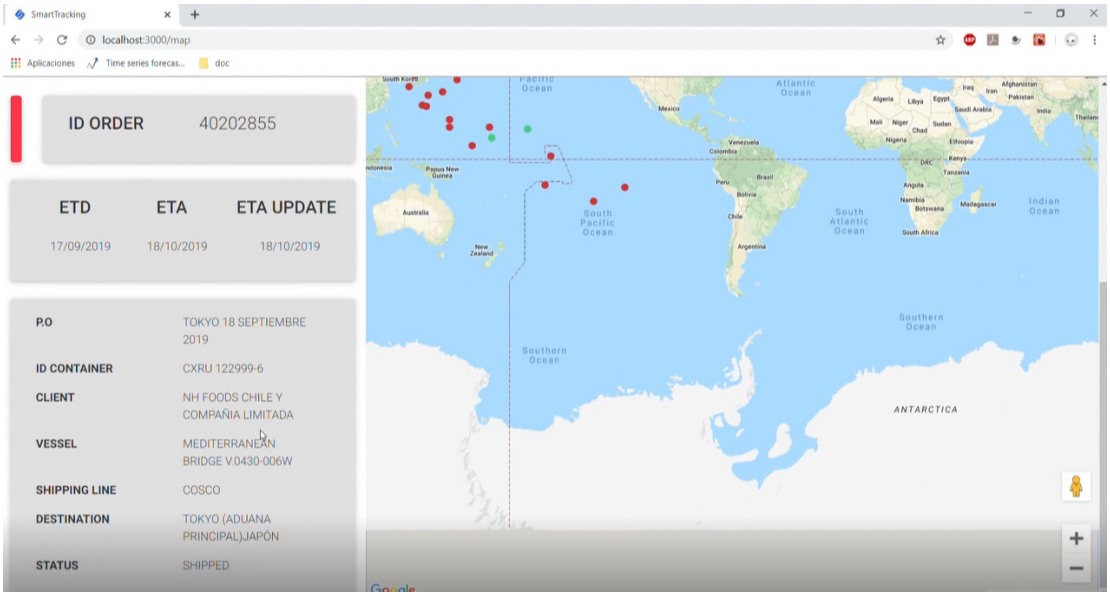


Figura 5.11: Continuación figura anterior.

Por ultimo, en la 5.12, se puede ver el funcionamiento del botón *Search Container*,

el cual filtra por un ID ORDER, desplegando solo a ese contenedor en el mapa, en este caso de color verde.

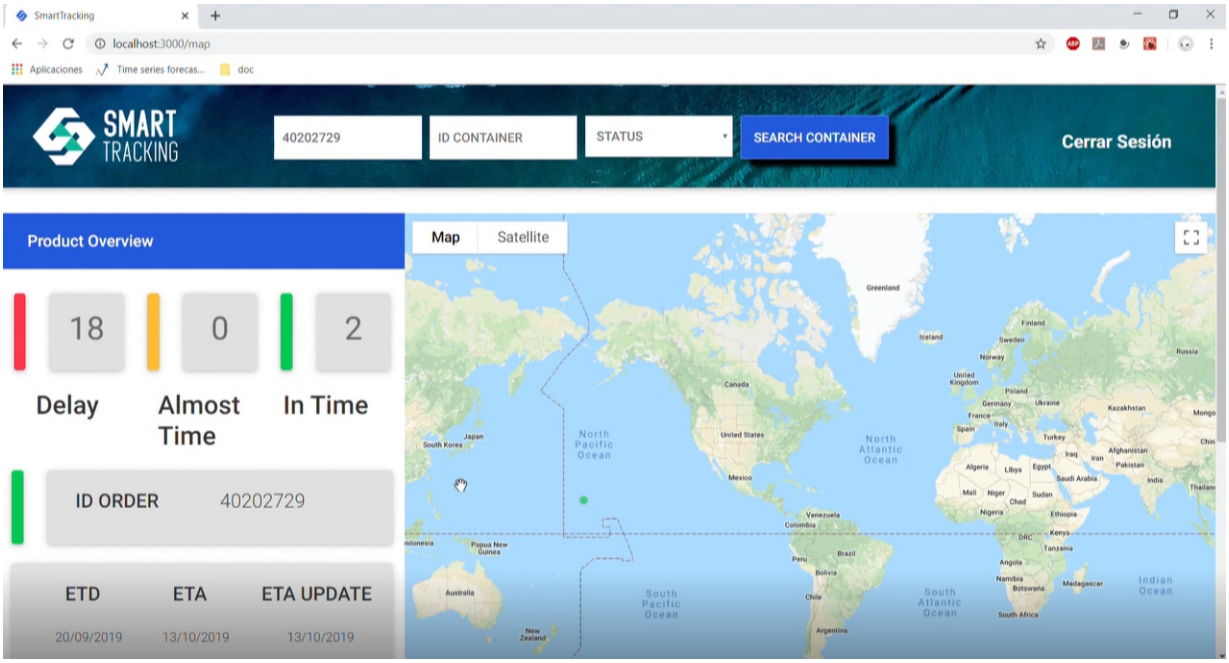


Figura 5.12: Filtrado de contenedor en la Vista Mapa.

Capítulo 6

Conclusiones y Trabajos futuros

Este trabajo presenta una resolución del problema sobre seguimiento de contenedores. El método propuesto es la creación de una aplicación web, el cual mediante *APIs* proporciona información sobre la actualización de sus *ETAs*.

En la aplicación, existen usuarios que pueden subir información de los contenedores y otros usuarios los cuales pueden ver el estado de estos contenedores, tomando decisiones a partir de la información que la aplicación brinda, ya sea poder avisar al cliente cuando debería llegar el contenedor o avisar ante algún plan de contingencia por el atraso de alguno de estos.

El problema de seguimiento de cargas es muy habitual en estos tiempos, por lo que *SmartTracking* es una solución novedosa y a la vanguardia de los problemas del Siglo XXI, el cual puede abordar más desafíos y no solo enfocarse en seguimiento por el mar, sino que también podría ser un seguimiento total en su línea productiva, desde que sale el contenedor de la empresa, hasta que llegue al cliente, pudiendo otorgar un mejor servicio.

Un desafío que tiene la plataforma, es poder poner en producción la aplicación web, encontrar alguna empresa la cual esté dispuesta a poder probar *SmartTracking*, de esta forma poder recibir críticas, ya sea, sobre el diseño de la interfaz, agregar nuevas funcionalidades que solucionen los problemas de la empresa o revelar errores de programación que tenga la aplicación.

Lo que queda como trabajo a futuro para poder llegar a implementar en la plataforma, es lograr recolectar la suficiente información sobre los ETAs iniciales, ETAs reales, ETD y rutas comerciales usadas, luego apartir de esta información poder realizar un modelo predictivo, para así de esta forma tener otra ETA la cual complemente la información entregada por las APIs de las navieras.

Bibliografía

[1] Ministerio de relaciones exteriores. Liderazgo de Chile en las exportaciones mundiales. https://www.prochile.gob.cl/wp-content/uploads/2019/05/liderazgo_de_chile_en_exportaciones_mundiales_2018_IC.pdf, 2018.

[2] Canal de Televisión 24 hrs. Ganancias 2018 del mercado de las carnes y pescados. https://www.24horas.cl/economia/exportaciones_de_carne_chilena_superaron_los_mil_millones_de_dolares_en_2018_3192990, Marzo 2019.

[3] Dolgansky E. ¿por qué se retrasó su carga?. lilly + associates - international transportation logistics. https://www.shiplilly.com/es/blog/por-que-se-retraso-su-carga/, Agosto 2016.

[4] Agrosuper. Agrosuper. https://www.agrosuper.cl/, 2017.

[5] Dimabe. Supertracking. https://aresweb.dimabe.cl/, Marzo 2020.

[6] Andrés Eslava. Aspectos logísticos-comerciales en el transporte internacional de contenedores marítimos. https://revistadelogistica.com/transporte-y-distribucion/aspectos_logisticos_comerciales_en_el_transporte_internacional_de_contenedores_maritimos/, Julio 2018.

[7] Alonso Cabrebas Cánovas. *Transporte internacional de mercancias*. ICEX Instituto Español del comercio exterior, Madrid, España, Marzo 2011.

[8] Adam Dunkels Jean-Philippe Vasseur. Container tracking. https://www.researchgate.net/publication/301115099_Container_Tracking, Diciembre 2010.

[9] Hitoshi Hayashi Masashi Shimizu Minoru Katayama, Hiroshi Nakada. Survey of rfid and its application to international ocean/air container tracking, ieice trans. commun., vol. e95.b, no. 3, pp. 773–793, 2012. https://www.researchgate.net/publication/258649745_Survey_of_RFID_and_Its_Application_to_International_OceanAir_Container_Tracking, Febrero 2012.

[10] Seung-Bum AHN. Container tracking and tracing system to enhance global visibility, proceedings of the eastern asia society for transportation studies, vol. 5, pp. 1719 - 1727, 2005. https://www.researchgate.net/publication/242153973_CONTAINER_TRACKING_AND_TRACING_SYSTEM_TO_ENHANCE_GLOBAL_VISIBILITY, Julio 2004.

[11] Li-Yen Shue Sheng-Tun Li. A study of logistics infomediary in air cargo tracking, industrial management and data systems, vol. 103, no. 1-2, 01.01.2003, p. 5-13. https://www.researchgate.net/publication/220672315_A_study_of_logistics_infomediary_in_air_cargo_tracking, Enero 2003.

[12] V.K.Devi,A.D.M.Gururaj,A.Kavya,E.Umamaheswari. Truck tracking and alerts monitoring system, International Journal of Civil Engineering and Technology 9(11):105-111. https://www.researchgate.net/publication/329733160_Truck_tracking_and_alerts_monitoring_system, Octubre 2018.

[13] Ruey Long Cheu, Hao Wang, Der-Horng Lee. Incorporating Telemetry and Car-Following Data for Real-Time Monitoring of Container Trucks, Eighth International Conference on Applications of Advanced Technologies in Transportation Engineering (AATTE) May 26-28, 2004 — Beijing, China. https://www.researchgate.net/publication/268598245_Incorporating_Telemetry_and_Car-Following_Data_for_Real-Time_Monitoring_of_Container_Trucks, 2004.

[14] Chintan Amrit Jos van Hillegersberg Sjoerd van der Spoel. "predictive analytics for truck arrival time estimation: a field study at a european distribution centre," int. j. prod. res., vol. 55, no. 17, pp. 5062–5078, sep. 2017. https://www.tandfonline.com/doi/full/10.1080/00207543.2015.1064183, Julio 2015.

[15] Mahmoud Kharrat Naghmeh Ivaki Abbas Asosheh, Arash Afshinfar. A network model for the intelligent marine container tracking, proceedings of the 8th conference on applied informatics and communicationsaugust 2008 pages 303–308. https://www.researchgate.net/publication/262315275_A_network_model_for_the_intelligent_marine_container_tracking, Agosto 2008.

[16] Nguyen Thai Dung Timo Bretschneider. Container tracking via ais satellites, conference: Asian conference on remote sensing. https://www.researchgate.net/publication/275036073_Container_tracking_via_AIS_satellites, Septiembre 2014.

[17] A. Gagnon R. Schmidt and D. Allcock. "maritime containers tracking trial results," ieee aerosp. electron. syst. mag., vol. 24, no. 9, pp. 10–14. https://www.researchgate.net/publication/245345785_Maritime_Containers_Tracking_Trial_Results, Agosto 2009.

[18] Raymond Hardij. Predicting arrival times for tankers ships using recurrent neural networks. http://arno.uvt.nl/show.cgi?fid=147103, Junio 2018.

[19] Z. Yang N. H. M. Salleh, R. Riahi and J. Wang. "predicting a containership's arrival punctuality in liner operations by using a fuzzy rule-based bayesian network (frbbn)," asian j. shipp. logist., vol. 33, no. 2, pp. 95–104. https://www.researchgate.net/publication/318343401_Predicting_a_Containership's_Arrival_Punctuality_in_Liner_Operations_by_Using_a_Fuzzy_Rule-Based_Bayesian_Network_FRBBN, Junio 2017.

[20] SeaRates LLC. Searates. https://www.searates.com/, Marzo 2020.

[21] ARVIEM AG. Arviem. https://arviem.com/, Marzo 2020.

[22] Ocean Insights Ltd. Ocean insights. https://www.ocean-insights.com/, Marzo 2019.

[23] Hamburg Südamerikanische Dampfschifffahrts-Gesellschaft A/S Co KG. Track trace. https://www.hamburgsud-line.com/liner/es/liner_services/ecommerce/track_trace/index.html, Marzo 2020.

[24] Facebook Inc. React js. https://es.reactjs.org/, Marzo 2020.

[25] Evan You. Vue js. https://vuejs.org/, Marzo 2020.

[26] by Google. Angular js. https://angularjs.org/, Marzo 2020.

[27] Stack Overflow. Stack overflow developer survey 2020. https://insights.stackoverflow.com/survey/2020, 2020.

[28] StrongLoop TJ Holowaychuk et al. Express js. https://expressjs.com, 2010.

[29] Pivotal Software. Spring. https://spring.io/, Octubre 2002.

[30] Django Software Foundation. Django. https://www.djangoproject.com/, Julio 2005.

[31] David Heinemeier Hansson. Ruby on rail (ror). https://rubyonrails.org/, Diciembre 2005.

[32] Ryan Dahl. node js. https://nodejs.org, Mayo 2009.

[33] Isaac Schlueter. Node package manager. https://www.npmjs.com/, febrero 2014.

[34] LearnBoost. Mongoose. https://mongoosejs.com/, febrero 2011.