

Práctica Profesional Supervisada:  
Seguimiento de notas de pedido, estado y  
recorrido de los camiones de la flota  
mediante una app.



**Alumno:** Gramuglia, Lautaro. LU: 248868

**Email:** gramuglialautaro@gmail.com

## Resumen del proyecto

Uno de los principales problemas de las empresas que optan por adoptar las nuevas tecnologías y desembarcar en el eCommerce, es resolver correctamente el envío de los pedidos. El cliente final es el eslabón más importante en cualquier procedimiento logístico de entrega, ya que está depositando su confianza en el servicio solicitado. La reputación de la marca se pone bajo la lupa, tanto para bien como para mal. Un factor clave es la satisfacción del cliente, que dependerá directamente de la correcta entrega de los pedidos en los plazos acordados.

Para cumplir con estas premisas se necesita una comunicación interna clara, fluida y transparente, sobre todo entre el vendedor y los operarios. El desarrollo de un sistema capaz de comunicar las diferentes áreas descentralizadas encargadas de elaborar la entrega final es el problema que busca resolver este proyecto. Este sistema permitirá agilizar la labor por parte de los operarios y mejorará la transparencia del estado del pedido en cada sector de la empresa. Además, el estado de los pedidos será sincronizado en tiempo real, y el cliente recibirá mediante casilla de correo electrónico de forma transparente un mensaje con los últimos movimientos cada vez que se actualice esta información.

El proyecto consiste en el desarrollo de dos sistemas independientes. El primero, cuyo objetivo es comunicar los distintos sectores internos de la empresa, así como también informar al cliente el paso a paso de su pedido solicitado. Este sistema está compuesto por el desarrollo de una aplicación web para monitorear el estado de los pedidos (hora, fecha, lugar, sector en el que se encuentra, etc), una aplicación web que permita visualizar el historial de pedidos finalizados (completados, cancelados, filtrar por un cliente en específico, etc); y una aplicación móvil capaz de escanear códigos QR con la finalidad de actualizar automáticamente los estados del pedido en los distintos sectores de la empresa.

El segundo sistema, tiene como objetivo realizar un seguimiento de los viajes realizados por cada chofer/camión, además de que su estado pueda ser consultado en cualquier instante de tiempo, por el vendedor o cualquier otra entidad dentro de la empresa. Está compuesto por el desarrollo de una aplicación web, para monitorear el estado de los camiones (si se encuentra en viaje; si está esperando a ser cargado; o está cargado y listo para viajar), otra aplicación web que permita realizar alta, baja y modificación de camiones, así como también de los viajes asociados a cada camión; y por último, una aplicación móvil capaz de escanear códigos QR

con la finalidad de actualizar automáticamente los estados de los camiones (comenzó/finalizó el viaje, el camión se encuentra ocioso/disponible, fuera de servicio, actualmente cargando). También se implementaron las APIs necesarias para dar soporte a las aplicaciones de ambos sistemas.

La empresa se encargó de proveer los recursos necesarios para una correcta ejecución del proyecto. En resumen, se adquirieron un total de diez smartphones, se delegó la creación de los códigos QR a una empresa que trabaja con ellos en otros proyectos, y se adquirió un servidor para desplegar las aplicaciones.

## Introducción

La adquisición de un sistema de seguimiento en tiempo real con el monitoreo de rutas/estado, plazos de entrega, ayudará a combatir la inseguridad y mejorar la calidad del servicio al cliente final, por la transparencia de la operación logística que el sistema ofrece. Además, mejorará la comunicación interna de la empresa, ya que cada etapa por la que atraviesa el pedido es independiente en cuanto al lugar físico donde se ubica dentro de la misma. Cada operario debe comunicar al sector encargado de controlar la logística del envío el estado actual del mismo, por lo que, gracias a las aplicaciones desarrolladas, el trabajo se vuelve menos tedioso, más ágil y transparente.

## Desarrollo

El primer paso para desarrollar el proyecto fue contactar con el cliente, realizar reuniones para analizar el problema a resolver y comenzar a modelar una posible solución, capturando los requerimientos fundamentales.

Tal como se estudió en la cátedra “Metodologías de Desarrollo de Software I” y en la optativa “Métodos Ágiles para el Desarrollo de Software”, una de las claves para llevar a cabo de manera correcta, ordenada y eficiente un proyecto, es utilizar un ciclo de vida estructurado durante el desarrollo. En este se especifican las etapas del desarrollo de software y el orden en que se deben ejecutar. En resumen, las etapas que se llevan a cabo son:

- ❖ **Captura de requerimientos:** Se realizan entrevistas con el cliente identificando los requerimientos y necesidades del usuario. Se comprende el problema en lenguaje natural.
- ❖ **Análisis:** Se modelan los requerimientos del usuario. Consiste en llevar el lenguaje natural del cliente a un lenguaje formal/terminología del desarrollador.
- ❖ **Diseño:** Se modela la solución del sistema, teniendo en cuenta el ambiente de implementación a utilizar, los recursos disponibles tales como base de datos, equipos; se define el lenguaje de programación, performance deseada, etc.
- ❖ **Implementación:** En base a los requerimientos y tecnologías definidas, se comienza con la implementación.
- ❖ **Testeo:** En esta etapa se verifica y valida el sistema teniendo en cuenta algunos criterios determinados por el grupo correspondiente
- ❖ **Mantenimiento:** Consiste en actualizar y modificar el sistema si surgen nuevos requerimientos.

En particular, Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto, utilizando un modelo de ciclo de vida iterativo e incremental. Ciertos actores son fundamentales para el proyecto. El Product Owner de este proyecto es a su vez, el cliente, quien decide qué funciones construir y el orden; gestiona los recursos disponibles; y define los criterios de aceptación. El Scrum Master, es quien lidera el proyecto y a quien se debe consultar ante cualquier eventual inconveniente, duda, o incluso para coordinar meetings.

Para el desarrollo de las aplicaciones, comenzamos definiendo un product backlog en trello organizando todas las tareas por hacer y asignándolas a cada uno del equipo. Estas tareas eventualmente se modificaban de acuerdo a los requerimientos del cliente o incluso surgían nuevos cambios o tareas, durante las reuniones semanales. Se utilizó *Git* para manejar el versionado del proyecto, que junto con *github* permitía trabajar en equipo de forma simultánea sin perjudicar los cambios hechos por cualquier miembro del equipo.

En la Tabla 1 se muestran las tareas ejecutadas de cada objetivo y sus correspondientes subtareas:

Etapa	Sub-etapa	Sub-tarea	Horas invertidas
<b>Reuniones</b>	Empresa encargada de generar QR		5
	Cliente		30
<b>Desarrollo de servidores</b>	RESTful API	Documentación	4
		Creación de puntos de acceso (endpoints) y controladores	40
		Creación de modelos	8
		Testing manual	10
	PostgreSQL	Modelar el problema	8
<b>Desarrollo de aplicaciones de usuario</b>	Aplicaciones Móviles	Diseño y creación de actividades	6
		Conexión con servidor y manejo de datos	15
		Flujo de actividades	10
		Creación y uso de escaner QR	4
		Creación logos y aplicación	3

		Testing Unidad	15
<b>Documentación general</b>	Diagrama de flujo, Diagrama de casos de uso, Diagrama de despliegue, Guía de endpoints y Guía de instalación		32
<b>Instalación y configuración del entorno de producción</b>	Pruebas en Windows Server		12

**Tabla 1**

## 1. Reuniones

- a. **Empresa encargada de código QR:** Para la creación de los códigos QR propios de cada pedido, la empresa subcontrata a otra. Fue necesario contactar y acordar con la misma esta operación mediante intercambio de mails y videollamadas. La empresa encargada de la creación de los códigos QR define el formato en que la información será entregada a través de estos. (Horas invertidas: 5hs).
- b. **Cliente:** Una parte fundamental del proyecto es una comunicación fluida con el cliente. Para esto se realizaron reuniones cada dos semanas en promedio, en las cuales se mostraban avances del proyecto, se solicitaban cambios, mejoras, y se intercambiaban puntos de vista. (Horas invertidas: 30hs)<sup>1</sup>.

## 2. Desarrollo de servidores

- a. **RESTful API:** El objetivo es consultar y manipular los datos desde diferentes entornos (aplicaciones móviles, web apps, entre otras).
  - i. **Documentación:** Se realizó una documentación del código fuente programado para posteriores upgrades, mejoras o cambios. (Horas invertidas: 4hs).

---

<sup>1</sup> No se contemplan las horas de reuniones intra diarias con el equipo

- ii. **Creación de endpoints y controladores:** Se definen las rutas por medio de las cuales las aplicaciones de usuario se conectan al servidor para recibir y enviar los datos. Se utiliza una arquitectura y patrón de diseño de tipo MVC (Model View Controller) para estructurar el proyecto y delegar funcionalidad a los distintos componentes. (Horas invertidas: 40hs).
- iii. **Creación de modelos:** Se crean las clases necesarias para mapear los objetos relacionales y restricciones de las distintas tablas pertenecientes a la base de datos. Se utiliza la librería 'sequelize' que cumple con este objetivo. (Horas invertidas: 8hs).
- iv. **Testing manual:** Se utilizó Postman para probar los diferentes endpoints implementados. Se constató el correcto funcionamiento de los mismos, la respuesta entregada (código y datos) tanto en llamadas válidas como inválidas. (Horas invertidas: 10hs).

**b. Base de datos**

- i. **Modelar el problema:** A partir de la captura de requerimientos, se pensó como organizar los datos en diferentes tablas y cómo se relacionan estas entre sí. (Horas invertidas: 8hs)<sup>2</sup>.

**3. Desarrollo de aplicaciones de usuario**

**a. Aplicación móvil:**

- i. **Diseño y creación de actividades:** Se modelan las vistas de usuario y se diseña la UI. Se utilizan tanto librerías provistas por el propio sistema android jetpack como librerías de terceros open source con el fin de implementar una interfaz amigable con el usuario, fácil de comprender y minimalista. (Horas invertidas: 6 hs).
- ii. **Conexión con servidor y manejo de datos:** El propósito de esta tarea específica es lograr una comunicación con el servidor, tanto para recibir

---

<sup>2</sup> No se tienen en cuenta las horas de entrevistas/reunión con el cliente o cambios realizados durante el desarrollo del proyecto.

datos en el login de un usuario, como para enviarlos al utilizar la función principal de la aplicación (escáner). Se utiliza retrofit para realizar conexiones, parsear datos, entre otros. Se crearon en tiempo de desarrollo vistas para introducir manualmente parámetros necesarios como la dirección ip, puerto, etc. (Horas invertidas: 15hs).

- iii. **Flujo de actividades:** Consiste en una correcta navegación dentro de la aplicación, una transición fluida y sin errores. Para esto se debe prestar principal atención en el ciclo de vida de las actividades declaradas y el estado de ejecución. (Horas invertidas: 10hs)
- iv. **Creación y uso de scanner.** Se incluye una dependencia que permite acceder a la cámara, ejecutar una actividad propia, y realizar la función de escáner con el objetivo de procesar los datos provenientes de un código QR. (Horas invertidas: 4 hs).
- v. **Creación logos y apk embebida:** Creación y configuración de logos, imágenes en distintas resoluciones utilizando Image Assets Studio. Se utiliza para desarrollar y compilar la aplicación el kit de desarrollo SDK versión 29, siendo retrocompatible hasta Android 6 (Marshmallow o también conocido como API 23). (Horas invertidas: 3 hs)
- vi. **Testing de unidad:** Creación de test para comprobar el funcionamiento de los distintos endpoints y los correspondientes llamados a través de Retrofit, utilizando JUnit. Comprobación de parseo de datos y flujo de actividades. (Horas invertidas: 15hs)

4. **Instalación y configuración del entorno de producción:** El objetivo es permitir la creación de paquetes estándar para el despliegue de las aplicaciones que incluyan todo lo necesario para que estas funcionen (dependencias) y que estén aisladas del sistema subyacente para lograr que siempre funcionen exactamente igual.

- a. **Pruebas en Windows Server.** Se realizaron pruebas para poder instalar docker y docker-compose sobre el sistema operativo Windows Server 2016. Se utilizó un entorno virtual en el cual se logró instalar ambos componentes. Se detectó un problema de permisos al intentar crear los contenedores propios de la aplicación.



Debido a los cambios necesarios a realizar para que las aplicaciones corran en este servidor, la empresa decidió migrar el proyecto a un entorno open source GNU/Linux. (Horas invertidas: 12hs).

**5. Documentación General:** Se desarrollaron diversos diagramas con el fin de documentar el trabajo realizado. Algunos de ellos fueron realizados para que futuros desarrolladores puedan continuar con el proyecto o implementar mejoras, actualizaciones, mantenimiento, y que a la hora de retomar el trabajo se reduzca la dificultad de llevar a cabo el requerimiento solicitado. Cada uno de estos informes son de vital importancia para comprender las distintas partes del proyecto, ya sea perteneciente al área de hardware, conexiones, desarrollo. Entre la documentación desarrollada se encuentra (Horas invertidas 32 hs):

- a. **Diagrama de Flujo:** este tipo de diagramas tiene como fin explicar gráficamente el proceso. En el mismo, se muestra como funciona el servicio empezando desde la aplicación móvil manipulada por el usuario, indicando que es lo que sucede en la api hasta finalizar en el front-end del usuario.
- b. **Diagrama de casos de usos:** el diagrama de casos de uso es utilizado para explicar la comunicación y comportamiento del sistema entre los distintos actores (usuario, servidor, aplicación, etc). En este caso, se utilizó para explicar la relación entre cada uno de los actores y los casos de uso.
- c. **Diagrama de despliegue:** el diagrama de despliegue muestra la arquitectura del sistema, el hardware, software y middlewares, y cómo se conectan estos nodos entre sí. A través de este diagrama es posible visualizar cómo el sistema se desplegará físicamente.
- d. **Guía de instalación:** esta guía está orientada a desarrolladores. Por un lado, se realizaron las acciones necesarias para que con un simple comando se instalen los distintos servicios sin más complicaciones para empezar a desarrollar lo antes posible. Además, se explica el paso a paso para instalar estos servicios de una forma extendida, ejecutando los comandos necesarios para desplegar cada una de las partes del proyecto por separado, como por ejemplo, la base de datos, las tablas de usuario, etc.

- e. Guía de endpoints:** se desarrolló una guía en la cuál se explica cada uno de los endpoints implementados con su correspondiente estructura (método, parámetros de consulta, cuerpo de la solicitud, parámetros de ruta, etc).

En cuanto al desarrollo de las aplicaciones, diversas tecnologías fueron utilizadas, algunas vistas durante la carrera y otra gran parte surge de la investigación que hicimos para realizar dicho trabajo:

1. **Creación del servidor:** Se creó una REST API utilizando Node.js junto con Express (una para cada proyecto), y algunas otras librerías tales como:
  - **Express:** Es un marco de aplicación web back-end para Node.js. Está diseñado para crear aplicaciones web y API.
  - **Dotenv:** Es un módulo sin dependencias que permite cargar las variables de entorno de un proyecto node desde un archivo.
  - **Http-errors:** Sirve para crear errores HTTP para Express muy fácilmente.
  - **Morgan:** Es un middleware de registrador de solicitudes HTTP para node.js
  - **Websocket:** Es un protocolo de red basado en TCP. Define cómo se intercambian los datos entre redes. Debido a que es muy confiable y eficiente, es utilizado por casi todos los clientes. TCP establece la comunicación entre dos puntos finales, que se denominan sockets. Esto permite que los datos se transfieran en ambas direcciones.
2. **Conexión con la base de datos:** Se utilizó el motor de base de datos relacional PostgreSQL, el cual requiere utilizar una librería que permite conectar con Node.js. Esta librería se llama *pg*, que es un cliente de PostgreSQL sin bloqueo para Node.js. Por otro lado, se utiliza *sequelize*, una herramienta ORM de Node.js basada en promesas para Postgres, MySQL, MariaDB, SQLite y Microsoft SQL Server, y que cuenta con un sólido soporte de transacciones, relaciones, carga ansiosa y perezosa, replicación de lectura y más.
3. **Aplicaciones móviles:** La primer aplicación móvil, tiene como objetivo, escanear el QR de cada pedido y mandar sus datos al servidor, ya sea que se

quiera crear un pedido, actualizar su estado, cancelarlo o despacharlo; mientras que la restante tiene por objetivo escanear el QR de cada camión y mandar sus datos al servidor, con el fin de actualizar el estado del mismo y asignarle un viaje correspondiente según sea el caso. Estas aplicaciones hacen uso de las siguientes librerías para su correcto funcionamiento:

- ✓ **Retrofit:** es un cliente de servidores REST para Android. Permite realizar peticiones GET, POST, PUT, PATCH, DELETE y HEAD; además de gestionar parámetros, parsear datos automáticamente, entre otros.
  - ✓ **gson:** es una librería escrita en Java que permite convertir objetos Java a su correspondiente representación JSON. Del mismo modo, permite convertir un “Json string” a su equivalente objeto Java/Kotlin.
  - ✓ **okhttp:** es un cliente HTTP eficiente, permite el intercambio de datos de manera segura, rápida, y ahorrando ancho de banda.
  - ✓ **zxing:** “zebra crossing” es una librería de código abierto enfocada en procesar imágenes de código de barra y permite escanear códigos QR.
  - ✓ **circularAnimatedDrawable:** es una librería de terceros que permite realizar animaciones en las vistas. Permite configurar las diferentes propiedades de los objetos como color, opacidad, animaciones al presionar determinado botón, etc.
  - ✓ **jetpack:** es un conjunto de librerías provista por android que ayuda a los desarrolladores a seguir las prácticas recomendadas, reducir el código estándar y escribir código que funcione de manera coherente en los dispositivos y las distintas versiones de Android.
4. **Aplicaciones Webs:** Para el primer sistema se crearon 2 aplicaciones webs, las cuales tienen como objetivo mostrar el estado actual de los pedidos. La primera aplicación, consta de una tabla con la información de los pedidos que están actualmente en producción, con el fin de tener una vista global de los pedidos. Y la segunda, es un buscador de pedidos, en la cual se puede filtrar por número de pedido (para consultar por un pedido en particular) o por email o nombre de cliente (para consultar todos los pedidos que realizó este cliente). Para el

sistema restante se crearon dos aplicaciones web de usuario. La primera enfocada en presentar como *cardviews* el estado de los distintos camiones, mientras que la restante actúa como un gestor de viajes, en la que pueden darse de alta nuevos viajes, camiones, y controlar los existentes.

5. **Instalación y configuración del entorno de producción:** Etapa final del proceso, es aquí donde se prepara todo el proyecto para llevarlo a producción y que se ejecute sin inconvenientes. Se hace uso de varias tecnologías, tales como:

- ✓ **Docker:** Es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas librerías, herramientas de sistema, código y tiempo de ejecución. Con Docker, se puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que su código se ejecutará.
- ✓ **Docker-compose:** Es una herramienta para definir y ejecutar aplicaciones multi-contenedores de Docker.

Estas tecnologías, se instalaron en una máquina física, a la cual se le instaló el sistema operativo Debian 10, corriendo en modo servidor (sin entorno gráfico, para un menor consumo de recursos):

- ✓ **Debian 10:** Es un sistema operativo de computadora popular y de libre acceso que usa el kernel de Linux y otros componentes del programa obtenidos del proyecto GNU.

## Resultados

### Resultados Sistema Pedidos

Una vez finalizado el proyecto se obtuvieron las aplicaciones y servidores mencionados a lo largo de este informe. A continuación se muestran las vistas de usuario de las mismas.

## Aplicación móvil

En la Figura 1 puede verse el logo desarrollado para la aplicación móvil.



Figura 1

La aplicación móvil cuenta con una sección para loguear los distintos operarios de la empresa (Figura 2 izquierda). Para esto se requiere conexión a internet, seleccionar el sector correspondiente y la contraseña propia. Luego según el usuario logueado, se muestra la sección actividades. El procedimiento consiste en escanear el pedido una vez que el paquete llega al sector y nuevamente se debe escanear cuando finaliza el proceso en dicho lugar. También es posible cancelar un pedido en caso de fallas durante el proceso, siendo el sector de carga y despacho los únicos que cuentan con esta cualidad (Figura 2 centro). Una vez seleccionada la acción a realizar, se muestra una nueva vista, en la cual se habilita el escáner y resume los datos obtenidos al escanear (Figura 2 derecha).

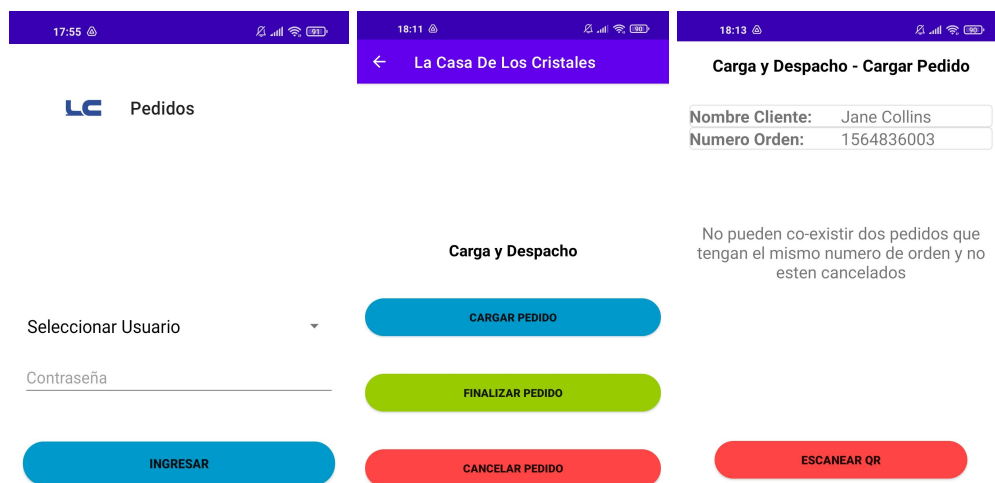


Figura 2: Izquierda

En la Figura 3 puede verse la ejecución de una nueva vista que, luego de solicitar los permisos de cámara necesarios, se ejecuta la funcionalidad de la cámara escáner.

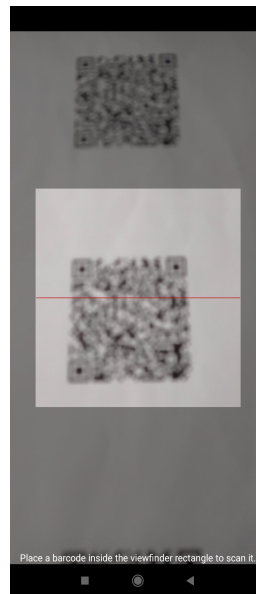


Figura 3

## Aplicación Web - Tabla

LC

LA CASA DE LOS CRISTALES

Estado actual de los pedidos

En proceso

Proceso finalizado

Cancelado

Corte

NP:  
1,564,836,005  
Cliente: Jane Collins

Figura 4

		Estado actual de los pedidos		En proceso	Proceso finalizado	Cancelado
Corte	DVH	Pulido y Entrantes		Cancelados		
<div> NP: 1,564,836,005  Cliente: Jane Collins </div>	<div> NP: 1,564,836,004  Cliente: Jane Collins </div>	<div> NP: 1,564,836,000  Cliente: Jane Collins </div>	<div> NP: 1,564,836,001  Cliente: Jane Collins </div>	<div> NP: 1,564,836,002  Cliente: Jane Collins </div>		

Figura 5

En las Figuras 4 y 5, se muestran los estados de los diferentes pedidos que están en producción, que han sido cancelados o despachados. Estos se muestran en rojo cuando recién comienzan una tarea estando en producción, y pasan a color verde una vez finalizan la misma. Las tareas por las que puede pasar un pedido son: Corte, Fábrica de DVH, Pulido y Entrantes, Templado o Laminado Eva. En ocasiones, un pedido puede ser cancelado en caso de fallas

durante el proceso, cuando esto ocurre el pedido se mueve a la columna “Cancelados” resaltada en color gris al igual que cambia de color la tarjeta.

### Aplicación Web - Historial

La Figura 6 muestra el historial de pedidos realizados. Se pueden ver todos los datos asociados al pedido y el estado del mismo.



Figura 6

Por otra parte, el mail que recibe el usuario se auto-completa con los datos del pedido según su estado actual. Estas aplicaciones, como ya se mencionó, fueron desplegadas en un servidor bajo el sistema operativo linux. Se realizaron pruebas en la empresa para constatar el correcto funcionamiento y sincronización de todas las herramientas entregadas.

### Resultados Sistema Camiones

Una vez finalizado el proyecto se obtuvieron las aplicaciones y servidores mencionados a lo largo de este informe. A continuación se muestran las vistas de usuario de las mismas.

### Aplicación Móvil

En la Figura 7 (izquierda) puede verse el logo desarrollado para la aplicación móvil.



Figura 7

En la Figura 7 (centro) puede verse las distintas funcionalidades de la aplicación móvil. Se solicitó que cada sector cuente con un acceso directo para escanear el código QR correspondiente al camión. En la Figura 7 (derecha) se ilustra la información previa a escanear el código QR, y una vez escaneado mediante la opción que se ve en pantalla, los campos se autocompletan.

#### Aplicación Web - Tabla de camiones

Estado de los camiones		Programa de salida de camiones	
 <b>Camión: 3</b> Chofer: Destino: Carga:	 <b>Camión: 2</b> Chofer: Destino: Carga:	<b>Lunes</b> 12/07	
	 <b>Camión: 1</b> Chofer: Destino: Carga:	Camión N° 1 Destino: Buenos Aires	Chofer: Juan Cruz Mendizabal Carga: Vidrio y Procesos
		Camión N° 2 Destino: Buenos Aires	Chofer: Juan Carlos Carga: Procesos
		<b>Martes</b> 13/07	
		<b>Miércoles</b> 14/07	
		<b>Jueves</b> 15/07	
		<b>Viernes</b> 16/07	
		Camión N° 3 Destino: Buenos Aires	Chofer: Maximiliano Carga: Vidrio
		<b>Sábado</b> 17/07	

Figura 8

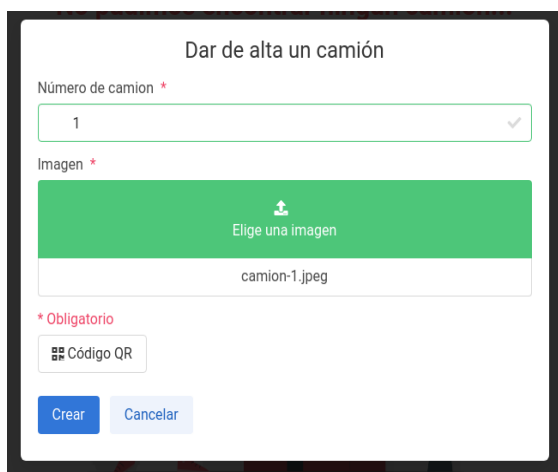
En la Figura 8, se muestran los estados de los diferentes camiones indicando en rojo cuando están “fuera de servicio”, en verde cuando están “en viaje” y en gris cuando están “preparados para salir de viaje”. La columna de la derecha, muestra el programa de salida de



camiones, donde se muestra cada viaje con su respectivo camión asignado, el chofer, su destino y el tipo de carga (pudiendo existir más de un viaje para un mismo día).

### Aplicación Web - Gestor de camiones

En la Figura 9, se muestra el formulario para poder cargar un camión, donde se coloca el número de identificación del camión y su imagen. Una vez asignado el número de camión, es posible descargar su QR asociado, y por último, los botones para crear el camión o bien para cancelarlo.



Formulario "Dar de alta un camión" con los siguientes campos:

- Número de camión \*:
- Imagen \*: 

Elige una imagen

camion-1.jpeg
- \* Obligatorio:
- Botones:

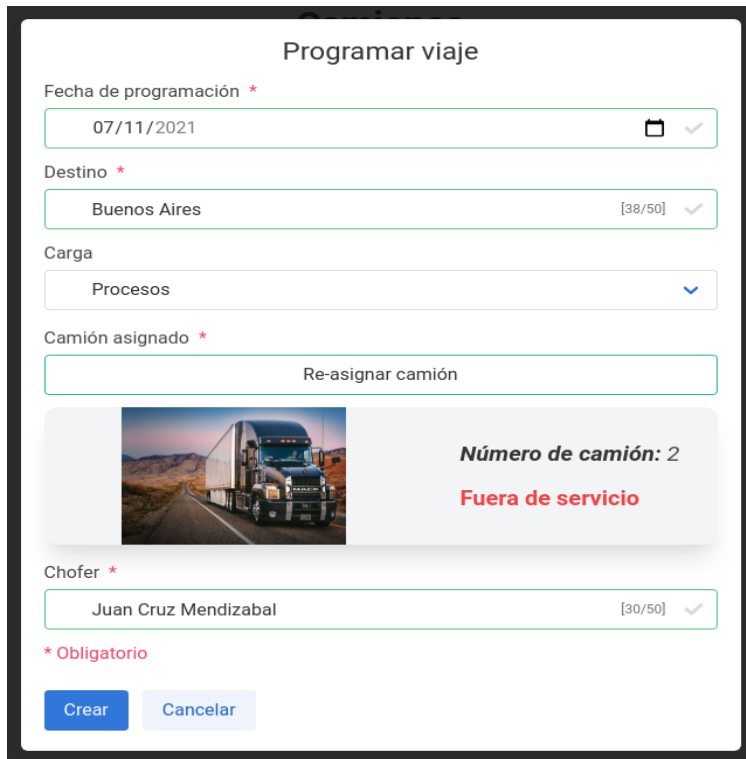
Figura 9

En la Figura 10, se muestra la lista de camiones previamente cargados, con la opción de ocultarlos, para no ocupar demasiado espacio.



Figura 10

En la Figura 11, se muestra el formulario para dar de alta un viaje donde se debe colocar la fecha de programación del viaje, su destino, el tipo de carga que lleva, el camión asignado y el nombre del chofer. Además, cuenta con dos botones, uno para crear el viaje en sí y otro para cancelarlo.



The screenshot shows a web form titled "Programar viaje". It contains several input fields: "Fecha de programación" with the date "07/11/2021", "Destino" with "Buenos Aires", "Carga" with a dropdown menu showing "Procesos", and "Camión asignado" with a button "Re-asignar camión". Below this is a section for the truck, featuring a photo of a truck and the text "Número de camión: 2" and "Fuera de servicio" in red. The "Chofer" field contains "Juan Cruz Mendizabal". At the bottom, there are two buttons: "Crear" (blue) and "Cancelar" (light blue). A red asterisk indicates mandatory fields.


Programar viaje

Fecha de programación \*  
07/11/2021

Destino \*  
Buenos Aires [38/50]

Carga  
Procesos

Camión asignado \*  
Re-asignar camión

 **Número de camión: 2**  
**Fuera de servicio**

Chofer \*  
Juan Cruz Mendizabal [30/50]

\* Obligatorio

Crear Cancelar

Figura 11

Por último, en la Figura 12, se muestra la lista de viajes programados junto con un resumen del mismo, donde se incluye la fecha de programado del viaje, el destino, el tipo de carga que lleva, el chofer asignado, la foto del camión y la fecha de creación del mismo.



Figura 12

Estas aplicaciones, fueron desplegadas en un servidor bajo el sistema operativo linux. Se realizaron pruebas en la empresa para constatar el correcto funcionamiento y sincronización de todas las herramientas entregadas.

## Conclusiones

Se logró cumplir con los objetivos planteados, entregando las aplicaciones finalizadas, desplegar los servidores con los que se conectan, e integrar las distintas herramientas para que funcionen sincronizadas y sin generar errores.

Haciendo un paralelismo con la carrera, se aplicaron diversos conocimientos adquiridos durante el transcurso de los años, desde conceptos básicos de programación, hasta la utilización de tecnologías, frameworks, metodologías de trabajo, sistemas operativos, bases de datos, patrones de diseño, arquitecturas, etc. Se reforzaron y afianzaron estos conocimientos integrándolos en un mismo proyecto, lo cual permite trasladar lo aprendido a un caso real y experimentar/solucionar problemas que en la teoría no existen.

Por último destacar que se aprendieron nuevas tecnologías, herramientas, frameworks, etc; pero lo más importante fue relacionarse con personas, con conocimientos muy variados, algunas de las cuales no poseen los mismos conocimientos técnicos, por lo que la comunicación es uno de los pilares fundamentales para completar el proyecto de manera satisfactoria. Además, trabajar en un proyecto que utilizarán de manera permanente conlleva una mayor responsabilidad, presión y compromiso de parte de todo el grupo de trabajo.