
Fortalecimiento de la gestión de pedidos a través de la digitalización del área mayorista



Lucas Cumsille Fernández
Ingeniería Civil Informática
Universidad Adolfo Ibáñez

Índice

Resumen ejecutivo	3
Executive Summary	4
1. Introducción	5
1.1 Contexto.....	5
1.2 Problema en la empresa	6
1.3 Oportunidades	7
2. Objetivos.....	7
2.1 Objetivo general	7
2.2 Objetivos específicos.....	8
3. Metodologías.....	8
4. Medidas de desempeño	9
5. Planificación	10
6. Estado de Arte	10
7. Propuesta de Solución	13
8. Análisis.....	15
8.1 Análisis de Riesgo	15
8.2 Evaluación Económica.....	16
8.3 Plan de implementación	17
9. Desarrollo.....	18
10. Resultados	25
10.1 Resultados Obtenidos	25
11. Conclusiones	26
12. Anexos	26

Resumen ejecutivo

Veggo es una Startup que se dedica a la venta y distribución de frutas y verduras. Como toda Startup se encuentra en constante cambio y crecimiento. Por ende, para no verse estancada en un punto se tiene que innovar. Ante esto se analizaron los puntos críticos y principales dolores en la venta y la gestión de los pedidos. Dentro del análisis se encontró que en el área de ventas los procesos, a largo plazo, no eran los adecuados ya que, muchos de estos constataban en procedimientos manuales. Dentro de estos se encontraba el ingreso de pedidos, que era realizado por el equipo de ventas y esto causaba una carga innecesaria en los vendedores. Si bien, era parte de su trabajo, esto le requería varias horas al día, destinando su tiempo en algo que a largo plazo ocasionaría una necesidad de aumentar los recursos a medida que los pedidos aumentasen.

Se aprecia una gran oportunidad al poder integrar herramientas digitales para facilitar el flujo de venta y poder liberar el tiempo y carga de los vendedores. Sin embargo, esto no solo está limitado para el área de ventas, sino también para la empresa en general. Esta oportunidad permite a la empresa a digitalizar las demás áreas para mejorar la comunicación de datos y encargados, al ser una empresa más bien multidisciplinaria la comunicación entre los departamentos es de suma importancia. Para ello se decidió por levantar un Framework de Ecommerce para el manejo del catálogo de productos y desarrollar una aplicación externa que permita al cliente ingresar su pedido y su posterior gestión. Para medir el impacto de la nueva plataforma de ingreso se estudiará la reacción del cliente a esta nueva herramienta, cuanto disminuye la carga al vendedor y la repercusión ante el flujo de datos.

Para esto se decidió implementar una metodología SCRUM para el avance de desarrollo, este se escogió debido a que a lo largo del desarrollo se tiene que tener retroalimentación por parte del cliente para definir las funcionalidades adheridas. Con ello se realizó la planificación de desarrollo donde el tiempo de implementación fue de 3 meses en total, contando el levantamiento y configuración de las diferentes plataformas. A raíz de las tareas definidas, en el mes de septiembre se realizó el levantamiento de un Framework Ecommerce de código abierto y su configuración destinada principalmente al despliegue de los productos en una página web y el manejo del catálogo de productos. Posteriormente, en el mes de octubre se

diseñó, desarrolló e integro una plataforma web externa que funciona como un tomador de pedidos, este se conecta con el BackOffice de la empresa para el ingreso de pedidos. Ya teniendo desarrollada la base del tomador de pedidos procede el desarrollo de funcionalidades particulares para mejorar la experiencia de usuario, como es el crear un catálogo personal para cada cliente que corresponde a los productos más comprados. Por ende, para que el cliente haga uso de esta plataforma es un requisito contar con su catálogo personal para estar habilitado.

Si bien, la totalidad de clientes no se encuentran habilitados, quienes están han utilizado esta plataforma sin la necesidad de volver al sistema anterior. Esto ha permitido al vendedor alivianar su carga y disponer de tiempo que se ha designado para tareas que permiten a la empresa crecer, como es la captación de nuevos clientes y fortalecer la relación con los existentes. Cabe destacar, que el flujo de venta cambió considerablemente respecto a meses anteriores, dejando menos espacio para errores en el ingreso de pedidos. A futuro, se incita a la empresa a utilizar este canal digital para desarrollar e instaurar una infraestructura de sistemas que permita escalar haciendo uso de herramientas digitales adecuadas a los requerimientos de la empresa.

Executive Summary

Veggo is a startup dedicated to the sale and distribution of fruits and vegetables. Like any startup, it is in constant change and growth. Therefore, in order not to become stagnant, innovation is necessary. In light of this, critical points and main pain points in sales and order management were analyzed. Within the analysis, it was found that in the sales area, at long-term processes were not adequate, as many of them relied on manual procedures. Among these was the order entry, which was carried out by the sales team and caused an unnecessary burden on the sellers. While this was part of their job, it required several hours a day, allocating their time to something that in the long run would necessitate increasing resources as orders grew.

There is a significant opportunity to integrate digital tools to streamline the sales flow and free up time and burden on the sellers. However, this opportunity is not only limited to the sales area but also for the company as a whole. This opportunity allows the company to digitize other areas to improve data communication and coordination, given that it's a multidisciplinary company where interdepartmental communication is crucial. To achieve this, a decision was made to establish an E-commerce Framework for managing the product catalog and to develop an external application allowing customers to place orders and manage them subsequently. To measure the impact of the new entry platform, the reaction of the customer to this new tool, the decrease in burden on the seller, and the repercussion on data flow will be studied.

For this purpose, it was decided to implement a SCRUM methodology for development progress. This was chosen because throughout the development, feedback from the client is essential to define the attached functionalities. Thus, a development plan was made with a total implementation time of 3 months, including the development and configuration of the different platforms. As part of the defined tasks, in September, an open-source E-commerce

Framework was established and configured mainly for deploying products on a web page and managing the product catalog. Subsequently, in October, an external web platform was designed, developed, and integrated as an order taker, which connects to the company's BackOffice for order entry. Once the base for the order taker was developed, additional features were created to enhance user experience, such as creating a personalized catalog for each customer based on their most purchased products. Therefore, it's a requirement for the customer to have their personalized catalog to access this platform.

Although not all customers are currently enabled, those who are enabled have used this platform without needing to come back to the previous system. This has allowed the seller to alleviate their burden and have time allocated to tasks that enable company growth, such as acquiring new customers and strengthening relationships with existing ones. It's worth noting that the sales flow has significantly changed compared to previous months, leaving less room for errors in order entry. In the future, the company is encouraged to use this digital channel to develop and implement a systems architecture that allows scaling using digital tools tailored to the company's requirements.

1. Introducción

1.1 Contexto

Veggo es una Startup fundada a fines del 2019 que se dedica a la venta y distribución de frutas y verduras desde el mercado mayorista Lo Valledor. A lo largo de los años Veggo ha probado distintos modelos de negocios aplicados en la venta minorista y mayorista. A pesar de que en estricto rigor ambos tipos de compra llegan al mismo sistema BackOffice¹, ambas ventas eran distintas principalmente por el tipo de cliente. El canal minorista contaba con una estrategia de negocio B2C por ende a este tipo de cliente se le denomina *cliente hogar*, una persona individual. Por otro lado, el canal mayorista cuenta con una estrategia de negocio B2C donde el cliente se conforma de una empresa por ende es un tipo de cliente que realiza pedidos a gran escala y con alta frecuencia, se le es denominado como *cliente mayorista*. Al ser un Startup la operación es 24/5. Cuenta con turnos de día y de noche. Donde, en el día se llevan a cabo los trabajos administrativos, como la contabilidad, desarrollo, logística y ventas. En la noche, el proceso de compra, calidad de productos, empaquetado y armado de pedidos. De esta manera el proceso de compra de productos depende de las ventas realizadas dentro del día y son estos despachados en la mañana del día hábil siguiente.

Por el lado del negocio B2C se tenía construida una infraestructura tecnológica que constaba de una aplicación web y una móvil como Frontend y un Backend enlazado a un servicio OMS².

¹ BackOffice: El BackOffice es el conjunto de sectores y funciones administrativas que están detrás de la operación de una empresa. (Sydle) <https://www.sydle.com/es/blog/back-office-60ede32b5378ca509b146da5>

² OMS: Un OMS es un sistema de software informático que automatiza el seguimiento del número de ventas, los pedidos, el inventario y el proceso de entrega. (Microsoft) <https://dynamics.microsoft.com/es-es/intelligent-order-management/order-management-system-oms/#:~:text=%C2%BFQu%C3%A9%20es%20un%20OMS%3F,supervisar%20personas%2C%20procesos%20y%20asociaciones.>

Para alimentar al sistema OMS desde el Backend se tenía levantado Framework de Ecommerce Magento, el cual proporciona una base para montar un comercio en línea. Provee de herramientas que ayudan a gestionar el catálogo de productos al igual que una base de datos relacional MySQL y un panel de administración.

En el caso del canal de ventas mayorista, al ser un negocio B2B, las principales ventas eran a locales comerciales tales como locales gastronómicos, cadenas de restaurantes y negocios de venta minorista. Por ende, el tipo de cliente se diferencia con el cliente hogar en la forma de comprar, los volúmenes difieren en gran escala al igual que el valor de cada pedido. Hoy en día el proceso de compra por parte de los clientes mayoristas consta de hacer llegar al vendedor una lista de productos que desea comprar y sus respectivas cantidades todo esto a través de un medio no adherido a la infraestructura de la empresa.

En consecuencia, se decidió diseñar un sistema parecido al modelo B2C donde a diferencia de usar el Framework de Magento se optó por Shopify resultando en el levantamiento de una página web donde los clientes contaban con el catálogo de productos con precio para mayorista y podían realizar sus pedidos. No obstante, esta nueva plataforma no fue bien recibida por los clientes debido a que complejizaba su proceso de compra y le tomaba más tiempo del que contaba para realizar esta tarea.

1.2 Problema en la empresa

Actualmente la empresa cuenta únicamente con un canal de ventas mayorista y como fue mencionado anteriormente, el proceso actual presenta varias tareas que requieren del vendedor para llevarlas a cabo, como es así el ingreso de cada pedido, el cual requiere de un ingreso manual. Esto se debe a que no existe una infraestructura tecnológica que gestione los pedidos como era en el caso de ventas a hogar. Debido a que, el proceso de venta en el área mayorista necesita de la acción por parte del vendedor de ingresar cada pedido en una plantilla Google Sheets que cuenta con funcionalidades a partir de GAS (Google App Script, lenguaje propio de Google derivado del lenguaje JavaScript que permite crear funciones para el manejo de información entre los documentos del ecosistema de Google). Para ello el vendedor tiene que seleccionar el cliente al cual se le va a vender, esto carga la información del cliente a los campos de la plantilla. Posteriormente, tiene que seleccionar cada producto a vender seguido del formato requerido. Luego ingresar la cantidad de cada producto y, por último, el cobro de despacho. Para finalizar, dentro las funcionalidades de la plantilla, existe un botón que extrae los datos de cada campo para luego ingresar la información en las diferentes hojas de trabajo.

Esto conlleva una carga dentro de las tareas del vendedor que le toma, en palabras del jefe de ventas, entre 4 a 6 horas en realizar. Esto también depende de la hora en la que el cliente le envía la lista de los productos a comprar a su vendedor asignado. Por ende, implica un aumento en la carga administrativa del vendedor en tareas reiterativas dentro de la semana, destinando su tiempo en tareas que no aportan al crecimiento de la empresa como es el captar nuevos potenciales clientes y establecer una relación entre la empresa y este.

Dado a que la empresa se encuentra en esta situación desde hace ya unos meses, ha llegado a un punto donde es difícil escalar ya que se necesitaría de más trabajo por parte de los vendedores y esto se traduce en un aumento de los recursos de la empresa, como es la contratación de nuevo vendedores para que realicen el ingreso de los pedidos cada día.

1.3 Oportunidades

Esto no solo presenta una oportunidad en la mejora del flujo de compra por parte del cliente a la empresa, sino que, también dar el primer paso para instaurar una infraestructura tecnológica para la gestión de los pedidos no solo en el canal mayorista, si no que, a nivel empresa con un apropiado control de inventario, distribución y contabilidad.

Sin embargo, a pesar de que el nuevo sistema de gestión de pedidos es una integración al BackOffice actual el éxito de este permitirá instaurar, a futuro, una arquitectura digital de sistemas diferente a la actual y robusta para la empresa. Esto se verá reflejado en un mejor flujo de datos, en la reducción de la carga administrativa a los trabajadores y parte de los costos administrativos.

Flujo ingreso pedido anterior

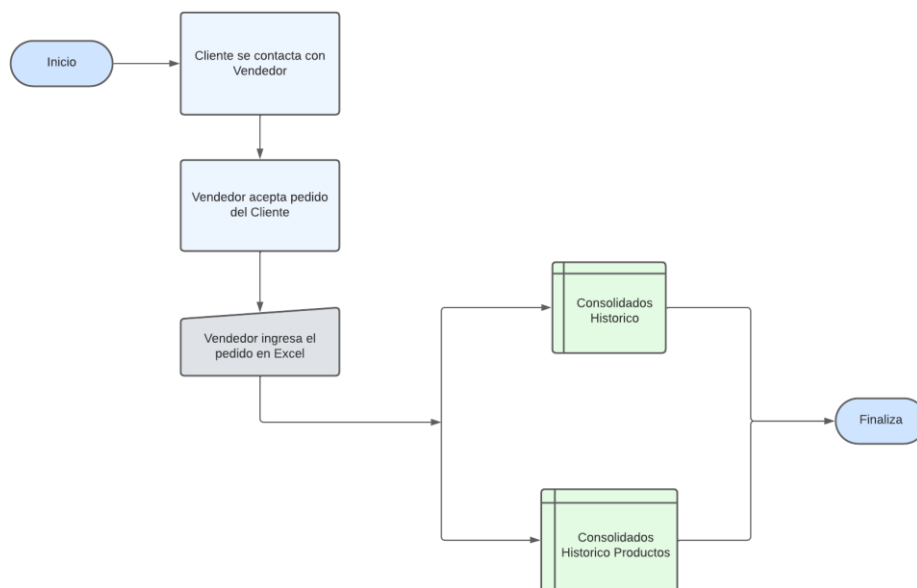


Diagrama de flujo 1: Proceso del ingreso de un pedido antes de implementar la solución.

2. Objetivos

2.1 Objetivo general

El objetivo principal es establecer un sistema que permita gestionar los pedidos ingresados al BackOffice. Estos tienen que ser ingresados por el cliente y administrados por los vendedores. Debe tener una interfaz amigable para el usuario y de uso fácil.

Esto se puede llevar a cabo replicando aspectos y tecnologías utilizadas en el negocio B2C que se tenía instaurado en el canal minorista adaptado a las necesidades y requerimientos del cliente mayorista.

2.2 Objetivos específicos

1. *Levantar Framework Ecommerce.*

Instalar Framework Magento CE (Community Edition) en una nueva instancia EC2 (Elastic Compute Cloud) en AWS (Amazon Web Service) con Ubuntu Server 22.04.

2. *Desarrollo aplicación web.*

Desarrollar y levantar MVP de la aplicación web ya integrada con el BackOffice de la empresa. Esta debe de estar instalada en el mismo servidores donde se encuentra el Framework utilizado.

3. *Configurar servidor DNS y servidor Apache2*

Para que la aplicación web quede habilitada para el cliente es necesario configurar el Virtual Hosting de Apache. Como ambas aplicaciones, web y Framework se encuentran en el mismo servidor el configurar el Virtual Hosting permite al usuario acceder a ambas plataformas mediante el IP del servidor y el puerto que queda designado para cada una. Esto también, requiere que el proveedor DNS, en el caso de Veggo es Cloudflare, se modifiquen los registros de este para establecer las URLs designadas a partir del dominio de Veggo.

En el caso de la página principal de Veggo (<https://www.veggo.cl>) se habilitará el puerto 80 de la instancia para el Framework de Magento. En este caso, la web principal se encarga de desplegar los productos que la empresa tiene habilitados (esto cambia según la temporada del año).

Por otra parte, la aplicación web integrada se designa el puerto 8080 con el subdominio de veggo.cl (<https://form.veggo.cl>).

4. *Analizar la respuesta del usuario/cliente.*

Al ya tener el MVP se tiene que realizar pruebas con clientes pequeños, los que cuenten con solo un local de despacho y hablar directamente con ellos para recibir retroalimentación de su parte, esto para conocer nuevas funcionalidades que faciliten la experiencia del usuario y/o se deben de hacer cambios a lo previamente desarrollado.

5. *Cumplir con los nuevos requerimientos del cliente.*

A partir de la retroalimentación de los clientes se deben de crear las nuevas tareas para el desarrollo de la aplicación web. Esta tarea es reiterativa debido a que la experiencia del cliente siempre es mejorable.

3. Metodologías

Para dar con la causa del por qué los clientes mayoristas rechazaron el uso de la plataforma para hacer pedidos se establecerán reuniones con los clientes frecuentes para ver su experiencia con la plataforma y que no les acomode de esta.

Para ello se utilizará la metodología SCRUM para el progreso semanal, teniendo reuniones con los clientes sobre su experiencia con la herramienta desarrollada, mejoras y si esta cumple con las expectativas de este.

Se ha escogido por la metodología SCRUM debido a que el desarrollo de la aplicación web y las futuras funcionalidades se llevarán a cabo por parte del practicante. La interacción con los clientes, al igual que las reuniones y las supervisiones de los sprint con el practicante serán designadas al jefe de operaciones de la empresa.

Las tareas del practicante pueden ser alteradas según las funcionalidades requeridas para el uso de la aplicación por parte del usuario final. Esto queda a criterio del supervisor.

4. Medidas de desempeño

Para medir el impacto del proyecto se enfocará principalmente en la satisfacción del cliente, los tiempos del vendedor y el incremento en captación de nuevos clientes.

En cuanto a la satisfacción por parte del cliente se centrará en la retroalimentación recibida de su parte, si este opta por utilizarla y si se inclina por volver al método “tradicional”. Se espera una migración de clientes del método tradicional a la digital de al menos un 33%, un tercio de los clientes habilitados para el uso del canal digital al final del proyecto.

Por otro lado, los tiempos del vendedor para que sea significativo su impacto se considera una disminución de su tiempo laboral de al menos 1 hora diaria. Al tener un porcentaje de clientes habilitados, la cantidad de pedidos ingresados de manualmente disminuye y a partir del tiempo promedio de ingreso, este tendrá tiempo disponible para realizar otras tareas.

Esto se medirá a partir de:

$$\left(\frac{T * \gamma}{\eta}\right) * \theta = \text{Tiempo disponible}$$

Donde:

- T: Promedio pedidos diarios.
- γ : Clientes habilitados.
- η : Total de clientes.
- θ : Tiempo promedio ingreso pedido.

Por ende, permitirá enfocar este tiempo en captar nuevos clientes al igual que mejorar la relación con los ya existentes.

A partir del tiempo disponible, la visita a nuevos potenciales clientes establece una relación de comercio, por ende, una posible nueva venta.

Para ello se medirá de la siguiente manera:

$$E * \left(\frac{\mu}{30}\right) = \text{Nuevas ventas}$$

Donde:

- E: Cantidad potenciales clientes.
- μ : Promedio ventas últimos 4 meses.
- 30: Periodo de 30 días.

5. Planificación

La planificación del proyecto se divide en tareas generales (macro) y las tareas específicas (micro) para ello se ha creado una carta Gantt, esta contiene tanto las tareas macro y como las tareas específicas a realizar por el encargado de desarrollo. El tiempo estimado para llevar a cabo el proyecto es de 480 horas aproximadamente, lo cual equivale a 11 semanas.

Carta Gantt proyecto

Calendario												
Mes	Septiembre				Octubre				Noviembre			
Semanas	1	2	3	4	5	6	7	8	9	10	11	12
Framework Ecommerce												
Acondicionar ambiente de producción												
Levantamiento de Framework en ambiente de producción												
Configurar Magento												
Montar frontend												
Cargar catálogo de productos												
Configurar DNS												
Tomador de pedidos												
Diseño												
Desarrollo de tareas												
Realizar pruebas internas												
Levantar prototipo en ambiente de producción												
Conectar con BackOffice actual												
Cientes												
Presentar tomador de pedidos												
Analizar retroalimentación de clientes												
Definir próximo avance												

Carta Gantt 1: Planificación a 3 meses con las tareas para la implementación del proyecto.

6. Estado de Arte

Para el Estado del Arte se estudió el contexto cultural y tecnológico relacionado a la evolución del comercio digital de los últimos años. Esto nos ayuda a entender de mejor manera los

beneficios y los desafíos que implica instaurar un negocio electrónico; qué herramientas se utilizan en la industria y cómo son integradas al modelo de negocios. De esta manera, se pueden mencionar casos de empresas que se dedican al mismo rubro que Veggo, como lo es Frubana y Frest. Si bien ambas empresas se dedican principalmente a la venta de clientes minoristas estas hacen uso de tecnologías que permiten la venta de productos por medios digitales a través de la red virtual que tienen instaurada.

Un negocio electrónico, también conocido como Ecommerce, consiste principalmente tanto en la compra y venta de productos, físicos o digitales, como el de servicios a través de internet. Esto como cualquier otra actividad económica trae consigo beneficios comparado con el medio tradicional al igual que nuevos desafíos para la industria.

Dentro de los principales beneficios de un Ecommerce se encuentra la reducción de costos en la puesta en marcha, en mantenimiento y de establecimiento. El levantamiento de un comercio digital es más sencillo, rápido y barato que uno convencional ya que, la instalación y configuración del entorno puede ser realizada por un proveedor de servicio o bien por un encargado dentro de la empresa en un servidor local o remoto. Esto también, conlleva al mantenimiento de dicho entorno que al ser uno digital se puede realizar parches y actualizaciones al sistema sin tener de manera remota. También, cabe destacar, que al ser un comercio digital no es necesario disponer de un local físico, permitiendo un ahorro en gastos fijos asociados como es el pago de arriendo de un local y de suministros básicos (Agua, Luz, ente otros.). A su vez, al ser un comercio online este no presenta limitaciones convencionales como lo es la geografía y la accesibilidad del cliente. Al no estar limitada a la geografía esta se puede expandir de manera indefinida si los medios se dan, por ejemplo, que preste un servicio digital (soporte, capacitaciones, etc.) o productos digitales como lo son las licencias de software. También, al estar alojada en la red global, la disponibilidad del comercio puede estar habilitada las 24 horas del día, los 7 días a la semana. Sin embargo, a pesar de tener una amplia variedad de beneficios no es un sistema carente de deficiencias. Esta nueva actividad económica conlleva nuevos desafíos para las empresas como son los plazos de entrega, la distribución, captación de nuevos clientes, la definición de los precios según la competencia, la calidad del producto y el servicio al cliente. Todas estas son aristas que son necesarias para entregar un buen servicio al cliente.

Hoy en día el comercio digital se encuentra implementado en la gran mayoría de las empresas y dentro de la última década ha incrementado de manera considerable.³ Logrando instaurar un nuevo concepto 'Comercio electrónico' o bien, conocido como 'ecommerce'. Este consiste en la compra y venta a través de Internet como su nombre lo indica y, como el crecimiento en las ventas por internet ha escalado considerablemente empresas de tecnologías, como Adobe, WordPress han creado un Framework de ecommerce para empresas proporcionando una base tecnológica para montar su propio negocio en línea.

Por ejemplo, Adobe es propietario de Magento Enterprise Edition y Community Edition, esto al igual que varias empresas, ofrecen dos tipos de productos, uno de código cerrado y con soporte técnico 24 horas y otro de código abierto donde no hay un soporte por parte del

³ StackScale. (2023). *Crecimiento y estadísticas del eCommerce (2023)*. StackScale.
<https://www.stackscale.com/es/blog/crecimiento-estadisticas-ecommerce/>

proveedor. Por su parte, WordPress es dueño de WooCommerce. Ambos Frameworks ofrecen servicios parecidos, sin embargo, la principal diferencia es el cobro por este sistema.⁴

Por una parte, tenemos los Frameworks open source que no tienen un cobro adicional dado que la instalación, el alojamiento y la configuración las debe de hacer el usuario. En cambio, sí se opta por un proveedor como Shopify este tiene cobro mensual dependiendo de plan que se opte⁵, al igual que si se opta por Magento EE (Enterprise edition) este tiene un cobro fijo mensual por cada servicio que entregue al igual que un pago inicial por la licencia de Magento.

En base a estudios realizados por la empresa de tecnologías e infraestructura en la nube StackScale en el año 2020 debido al confinamiento mundial por el virus COVID-19 el comercio electrónico aumentó en un 25% en orden mundial, y el año 2021 creció un 15%. Si bien los datos más relevantes en este estudio son realizados en base a datos de la región de Europa, nos permite generar una estimación del crecimiento del consumo electrónico por la población Latino Americana. (StackScale, 2023).⁶

Se han estudiado dos empresas, Frubana y Frest, empresas que también se dedican a la venta y distribución de frutas y verduras, entre otros productos para el hogar. Esto nos permite orientarnos para comparar las tecnologías que ambas empresas han utilizado para montar su Ecommerce. Cabe mencionar que, si ambas empresas se dedican al mismo rubro que Veggo, Frubana es una empresa extranjera con tiendas en Colombia, Brasil y México⁷. En cuanto a Frest, es una empresa nacional fundada el año 2015.⁸

Para un análisis más a fondo de la competencia se ha utilizado la herramienta de reconocimiento de tecnologías Wappalyzer de la empresa del mismo nombre, la versión que se utilizó consiste en una extensión de navegador para su uso rápido⁹, esta permite exportar la información en formato CSV para luego generar tablas. Las tecnologías que nos interesa investigar son, el lenguaje de programación, si es que utiliza un Framework de trabajo, si utiliza un CDN (Content Delivery Network) y las librerías de programación. Se hace caso omiso a las tecnologías utilizadas, por ejemplo, en el ámbito de Advertising, Analytics, entre otros.

Tecnologías utilizadas por Frest y Frubana

URL	Miscellaneous	Widgets	Analytics
https://co.frubana.com	LottieFiles		CleverTap; Google Ads Conversion Tracking; Facebook Pixel; Google Analytics
https://www.frest.cl			TikTok Pixel;

⁴ Adobe Commerce 2023. Magento 2023. <https://business.adobe.com/ar/products/magento/magento-commerce.html>

⁵ Shopify (2023). Shopify pricing (2023). Shopify. <https://www.shopify.com/pricing>

⁶ StackScale. (2023). *Crecimiento y estadísticas del eCommerce (2023)*. StackScale. <https://www.stackscale.com/es/blog/crecimiento-estadisticas-ecommerce/>

⁷ Frubana LinkedIn. <https://www.linkedin.com/company/frubana>

⁸ Frest LinkedIn. <https://www.linkedin.com/company/frest-mercado-saludable>

⁹ Wappalyzer. <https://www.wappalyzer.com>

			Pinterest Conversion Tag; Google Analytics ; Facebook Pixel
Advertising	Tag managers	Live chat	JavaScript libraries
Google Ads; Microsoft Advertising	Google Tag Manager		Moment.js; lit-html; lit-element; jQuery UI; jQuery
		WhatsApp Business Chat	
Maps	Programming languages	Issue trackers	Ecommerce
Google Maps	Java		Cart Functionality
Google Maps		Sentry	
Security	Font scripts	CDN	Marketing automation
HSTS	Google Font API	jsDelivr; Amazon CloudFront	CleverTap
HSTS		jsDelivr	
JavaScript frameworks	PaaS	Load balancers	UI frameworks
	Amazon Web Services	Amazon ALB	Semantic UI
Vue.js; Astro	Vercel		Tailwind CSS

Tabla 1: Resumen de las tecnologías utilizadas por Frubana y Frest. Frubana se representa de color naranja, Frest de amarillo y los títulos de color verde.

A partir de la tabla anterior, con Frubana vemos que el lenguaje utilizado es Java con librerías para JavaScript como jQuery, Moment.js y Lit (HTML, element), también cuenta con el servicio CDN de jsDelivr que se relaciona muy bien con las librerías de JavaScript. Todo esto montado en un servidor AWS con un modelo de servicio PaaS, donde los recursos utilizados fueron integrados a este ambiente. Por otro lado, en el caso de Frest vemos, al igual que en Frubana, el uso de jsDelivr para el CDN en conjunto con Vue.js y Astro como Frameworks de JavaScript todo montado en un servidor de Vercel, proveedora de servicios Cloud, en un modelo PaaS.

En base a lo anterior, el mejor camino para la empresa es replicar el modelo de la competencia y adaptarlo a las necesidades de esta y de los clientes, teniendo presente lo que ya previamente se había instaurado en el canal de ventas minorista.

7. Propuesta de Solución

De esta manera, se presenta la oportunidad de impactar directamente en las ventas de la empresa realizando un cambio en la forma en que los pedidos son ingresados al sistema. Para ello es importante conocer y entender los requerimientos del usuario en su proceso de compra, planificar el sistema cómo se integrará con el BackOffice de la empresa, y cómo debe de ser su diseño para que sea una solución adaptable y escalable.

Por otro lado, es importante mencionar que se debe de actuar sin generar repercusiones en los demás departamentos, excluyendo el de ventas. Ya que, al cambiar la manera en que los

pedidos son ingresados sin un previo análisis de cómo este entra al sistema de la empresa se puede generar una pérdida de información, lo cual es crítico. Esto puede ser tal y como sería la facturación y en el despacho del pedido. A esto viene el punto mencionado anteriormente "cómo el sistema se integrará con el BackOffice".

De esta manera, a partir de los ejemplos revisados dentro del análisis del Estado del Arte, aparecen las siguientes posibles soluciones para mejorar y superar la situación actual de la empresa.

Solución 1:

Instaurar un Framework de Ecommerce open source en el servidor de la empresa. Fijando el entorno de trabajo del Framework, instalar las dependencias necesarias, crear Frontend para la visualización a los clientes y la configuración de los registros DNS con el dominio de la empresa.

El beneficio de esta opción es la versatilidad del Framework, al ser open source las integraciones que sean requeridas el desarrollador puede llevarlas a cabo en gracias a la extensa documentación. Por otro lado, al ser open source el tiempo de levantamiento y configuración puede ser extenso comparado con contratar el servicio del proveedor o de un externo.

Solución 2:

Levantamiento de un sistema OMS para la gestión de pedidos. Esto conlleva a la instalación del ambiente y enlace con aplicación web o móvil.

En cuanto a esta solución, lo que es atractivo es tener un sistema aislado donde los pedidos son gestionados. Sin embargo, esto conlleva a problemas con el desarrollo de aplicaciones específicas, como puede ser, por ejemplo, un software destinado al ingreso de los pedidos, el cual tiene que tener una comunicación con el OMS y luego con el BackOffice. En consecuencia, un mayor tiempo de desarrollo. También cabe destacar que, existen Frameworks Ecommerce que contienen funcionalidades parecidas al OMS.

Solución 3:

Contratar servicio de WhatsApp Business API mediante un proveedor autorizado y crear un asistente virtual que ayude a gestionar los pedidos de los clientes a través de la plataforma de comunicación orientada a empresas y/o emprendimientos.

El principal beneficio de implementar esta solución es la relación y la experiencia con el cliente. Esto se ve reflejado en un sistema novedoso para el usuario con integraciones con Inteligencia Artificial lo cual es llamativo hoy en día. No obstante, esta propuesta se ve limitada a las funciones de los proveedores de servicio WhatsApp Business API y del costo que tiene asociado.

Como se ha mencionado, la empresa en el año 2022 contrató el servicio de Wingsoft, empresa especializada en desarrollo de Software, con el motivo de crear el canal digital para venta minorista, cuando este estaba vigente. El trabajo consistía en levantar un servicio OMS en un servidor remoto y posteriormente un desarrollo de una aplicación para dispositivos móviles. Cuya aplicación estaría enlazada con el catálogo de, en ese entonces, Magento y este se encargaría en el procesamiento de los pedidos realizados la App. Ayudando así con los pedidos realizados por los clientes minoristas.

El sistema gestor de pedidos para el canal minorista resultó ser un éxito. Los pedidos realizados por la aplicación móvil entraban sin problemas. Estos luego eran procesados y aprobados por el equipo de ventas y derivados al área de contabilidad para generar los documentos tributarios correspondientes. Esto constituye la parte administrativa de la venta.

A partir de esto, y de lo analizado en el Estado del Arte la solución que más se acomoda a los requerimientos de la empresa es derivada de la primera solución expuesta, ya que resuelve puntos críticos para un canal digital, un Frontend simple y un Backend ya esquematizado. Al escoger un Framework de código abierto las herramientas del nuevo canal digital pueden ser moldeadas para cumplir con los objetivos específicos que se tienen ideados para completar el objetivo general. A esto se tiene que añadir un sistema aparte que permita al cliente acceder a su información y realizar un pedido. El pedido tiene que ser enviado al BackOffice para su posterior procesamiento. Más bien, el Framework será utilizado para desplegar al público la página de Veggo y los productos que tiene habilitado para la compra.

8. Análisis

8.1 Análisis de Riesgo

Dentro el proceso de implementación de la solución es necesario tener en cuenta los riesgos que esta conlleva, los obstáculos que pueden ocurrir en el desarrollo de esta y los posibles errores que implica, en este caso, levantar un nuevo sistema digital en la empresa.

Los principales riesgos o peligros que podemos definir al término de la implementación son:

1. El cliente no hace uso de la aplicación para ingresar pedidos.
2. El servidor sufre una interrupción debido a la falta de control en los pedidos realizados.
3. La disponibilidad del servicio se ve afectada.
4. La comunicación entre la aplicación y el sistema BackOffice de la empresa presenta problemas.

A continuación, se presenta la matriz de riesgo para dar mejor a entender el impacto de cada falla y cuál es la ocurrencia de esta, siendo clasificado según la gravedad de este, clasificado en *Insignificante*, *Menor*, *Moderado*, *Importante* y *Catastrófico*. A su vez, es necesario tener en presente de la probabilidad de ocurrencia de cada uno, siendo así estimado en base a 5 escenarios, *Muy improbable*, *No es probable*, *Posible*, *Probable* y *Muy probable*.

Matriz de Riesgo

	1 Insignificante	2 Menor	3 Moderada	4 Importante	5 Catastrófica
5 Muy probable	5	10	15	20	25
4 Probable	4	8	12	16	20

3 Posible	3	6	9	12	15
				La disponibilidad del servicio se ve afectada.	
2 No es probable	2	4	6	8	10
				La comunicación entre la aplicación y el sistema BackOffice de la empresa presenta problemas.	El cliente no hace uso de la aplicación para ingresar pedidos.
1 Muy improbable	1	2	3	4	5
			El servidor sufre una interrupción debido a la falta de control en los pedidos realizados.		

Matriz de Riesgo: Riesgos asociados a la solución escogida en escala Insignificante, Menor, Moderada, Importante y Catastrófica.

8.2 Evaluación Económica

Antes de definir el plan de la puesta en marcha es debido realizar un análisis económico para definir bien los costos asociados a la implementación, con tal de mantener privada la información sensible de la empresa se ha decidido escalar valores, monetarios principalmente, por un factor común conocido por un encargado de la empresa.

Lo primero a considerar, es que los softwares de terceros que se han decidido a utilizar son de código abierto, por ende, no implica en un costo inicial o mensual para su uso. Esto conlleva que el mantenimiento, configuración e implementación se tiene que llevar por parte del encargado.

Por ende, el costo de implementación reside en el tiempo que tome el levantamiento, configuración de los softwares externos y el desarrollo de la aplicación web. Este es equivalente al costo a partir de las horas hombre que son necesarias para implementarlo.

El costo de inversión es calculado de la siguiente manera:

$$\left(\frac{\Phi}{180} * \eta\right) * \Lambda = \text{Costo proyecto}$$

Donde:

- Φ : Sueldo encargado.
- η : Cantidad de encargados.
- 180: Horas trabajadas al mes.
- Λ : Horas de implementación para el proyecto.

Por ende, tenemos:

$$\left(\frac{420560}{180} * 1\right) * 480 = 1121494$$

Un costo directo aproximado de \$1.13 millones de pesos chilenos para llevar a cabo el proyecto. Este valor no considera los costos de suministros básicos y arriendo de oficina.

Cabe mencionar, los costos asociados al realizar el ingreso de manera manual, estos se calculan de la misma manera que el anterior, derivado del tiempo que le toma al vendedor dentro de su jornada laboral.

Por ende, tenemos:

1. El tiempo promedio de ingreso de un pedido es de 10 minutos.
2. Se tiene un promedio de tickets 30 diarios.
3. El sueldo del vendedor es de \$938.720

Ente esto, obtenemos:

$$\left(\frac{938720}{180}\right) * (30 * 10) = 1564534$$

El costo aproximado al ingresar los pedidos manualmente es de \$1.6 millones de pesos chilenos. Como fue mencionado, este costo es diario.

8.3 Plan de implementación

Con el fin de tener organizado los tiempos y tareas a implementar para llevar a cabo el proyecto se entrará en detalle sobre los pasos a seguir. A su vez cada uno de estos se encuentra declarado y asignado a su respectiva semana en la carta Gantt sobre planificación del proyecto.

Las tareas a realizar se clasifican en 3 categorías: *Framework Ecommerce*, *Tomador de pedidos* y *Clientes*. Cada una de estas categorías abarcan subtareas que deben de llevar a cabo para completar el objetivo general.

Las subtareas para el levantamiento del Framework constan en:

1. Acondicionar ambiente de producción.

2. Levantamiento de Framework en ambiente de producción.
3. Configurar Magento.
4. Montar frontend.
5. Cargar catálogo de productos.
6. Configurar DNS.

Dentro de la categoría de la aplicación para tomar pedidos estás las tareas:

1. El diseño de la aplicación.
2. El desarrollo de la aplicación.
3. Realizar pruebas internas.
4. Levantar prototipo en ambiente de producción.
5. Conectar con el BackOffice actual.

Por último, quedan las tareas relacionadas con la comunicación y experiencia del cliente:

1. Presentar tomador de pedidos.
2. Analizar retroalimentación de los clientes.
3. Definir el próximo avance.

9. Desarrollo

En el mes de agosto se realizó un análisis en el proceso de venta en la empresa, en ello se encontró que empleados del área de ventas realizaban el ingreso de los pedidos de manera manual utilizando una plantilla de Google sheets acondicionada. Dentro de esta hoja de trabajo se encontraban las hojas individuales con la información de los clientes y el catálogo de los productos habilitados con sus respectivos precios actualizados. La plantilla para el ingreso de pedidos contaba con funciones nativas de Google para extraer la información del catálogo clientes e ingresar los campos requeridos para la venta, Rut del cliente, su razón social, dirección de facturación, dirección de despacho, entre otros. También, esta tenía integradas funcionalidades particulares de GAS (Google App Script) que permitían la transferencia de información a través de los diferentes documentos del ecosistema de Google que eran un pilar en el BackOffice de la empresa. Si bien esto permitía un ingreso más rápido al BackOffice esta tarea de ingresar el pedido no le corresponde al vendedor si no que al cliente. El proceso de venta consistía en, por parte del cliente el hacer llegar una lista de los productos seleccionados a su vendedor a cargo por cualquier medio electrónico y este procedía a ingresarlo utilizando la plantilla. Esto también implicaba una comunicación con el cliente para dar aviso de los productos disponible para la venta sin tener una imagen de referencia.

Ante esto se decidió en el mes de septiembre el levantar un Ecommerce de código abierto destinado a la visualización del catálogo al público general y un Frontend que puede ser encontrado en internet. Este se encargaría solamente de posicionar a Veggo en la red global, ya que, los clientes no harían uso de esta plataforma para realizar sus pedidos dado el tiempo que le tomaría el ingreso y debido a los pasos que no son necesarios para un cliente mayorista. Para el Framework Ecommerce se escogió Magento CE (Community Edition) ya que no requiere el pago de una licencia para su uso al igual que no tiene adherido un cobro por las transacciones realizadas, sin embargo, esto significa que no cuenta con un soporte

por parte de Adobe y todos los cambios y configuraciones deben ser realizados por un encargado de la empresa o bien por un tercero. Esta plataforma fue instalada en una instancia AWS (Amazon Web Service) donde se utilizó un Stack tecnológico LAMP (Linux, Apache, MySQL, PHP) para el ambiente del servidor. Para el levantamiento del Framework no se hizo uso de máquinas virtuales o de contenedores, si no bien, fue instalado en sistema global. Se decidió hacerlo de esta manera dado que al momento de configurar el Virtual Hosting del servidor Apache2 (HTTP) pudiese presentar problemas. También, para poner a veggo en línea fue necesario editar los registros del proveedor DNS (Cloudflare) para que el redireccionamiento de la url del dominio global de Veggo ("veggo.cl") fuese dirigido al Frontend de la nueva instancia. Cabe mencionar que, AWS provee de un servicio llamado "IP elástica" el cual consiste en entregar una IP estática que puede ser asignada a cualquier instancia que el usuario tenga, con esto se ha designado una IP elástica a la instancia para la configuración de los registros DNS.

Frontend Veggo

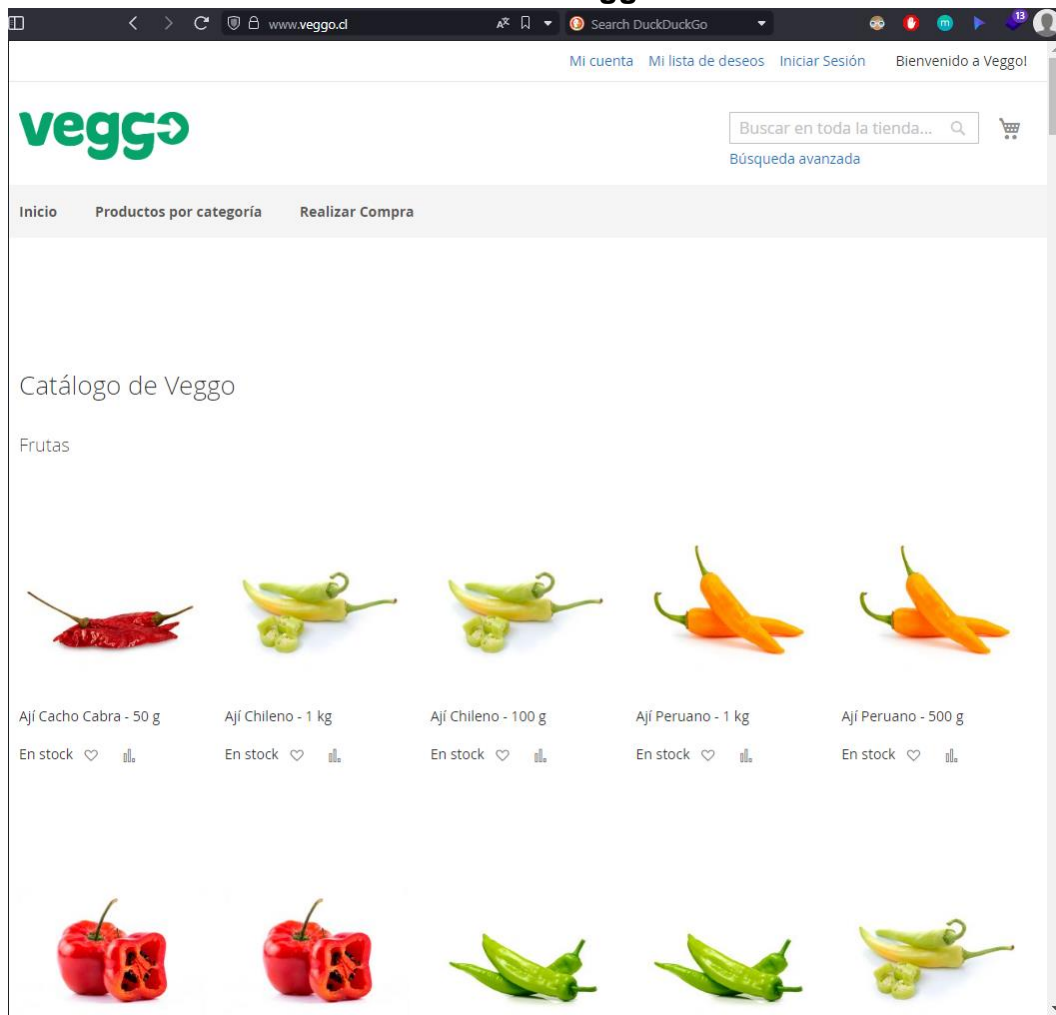


Imagen 1: Página de inicio de Veggo, este fue creado utilizando el motor web nativo de Magento CE.

Ya levantado el Framework, en el mes de octubre se procedió al desarrollo de la aplicación web conectada al BackOffice destinada al ingreso de los pedidos por parte del cliente. Esta fue desarrollada en Python 3.11 utilizando las librerías Flask para el desarrollo web y

Bootstrap 4 para el interfaz de usuario. Al igual que todo desarrollo de aplicaciones esta se encuentra sujeta a los requerimientos del usuario, estos son:

1. Interfaz de usuario amigable.
2. De uso fácil para el usuario.
3. Que el proceso no le tome más de 5 minutos en realizar.
4. El diseño tiene que ser coherente al manual de marca de la empresa.

En base a esto, el desarrollo del MVP (Producto Mínimo Viable) se enfocó principalmente en la interacción del usuario con la plataforma, permitiéndole, con solo ingresar su Rut de cliente, acceder a su catálogo personalizado desplegado como un formulario donde solo se necesita que ingresar el campo de la cantidad requerida por cada producto para luego ser ingresado utilizando un solo botón. Con esto, la información del pedido se envía al BackOffice en formato JSON para después ser procesado e ingresado a las principales hojas de trabajo. Esto permitió aliviar la carga del vendedor el cual solo se enfocaría en la administración de los pedidos entrantes y la comunicación con el cliente. También se creó un subdominio de la empresa para el acceso a la aplicación, este es “from.veggo.cl” y se agregó a los registros DNS.

Tomador de Pedidos preliminar

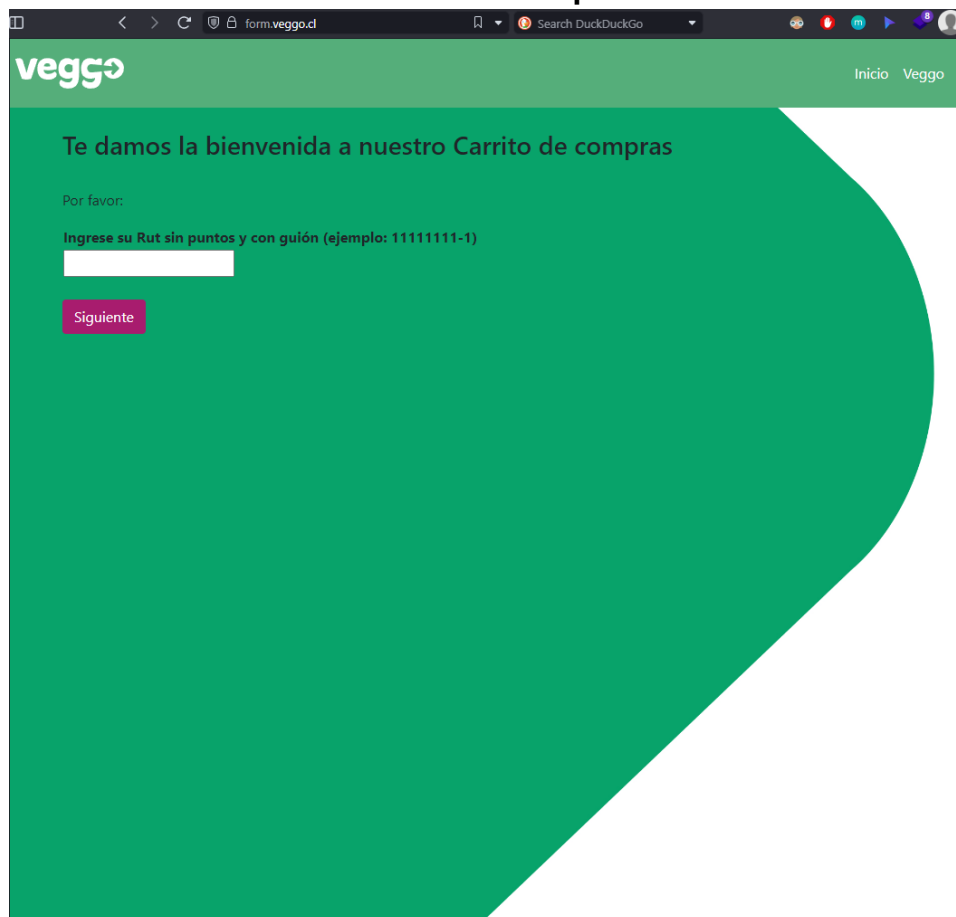


Imagen 2: MVP aplicación web para el ingreso de pedidos. Página de inicio donde el cliente ingresando su Rut puede acceder su información y catálogo personal.

Estructura Tomador de Pedidos

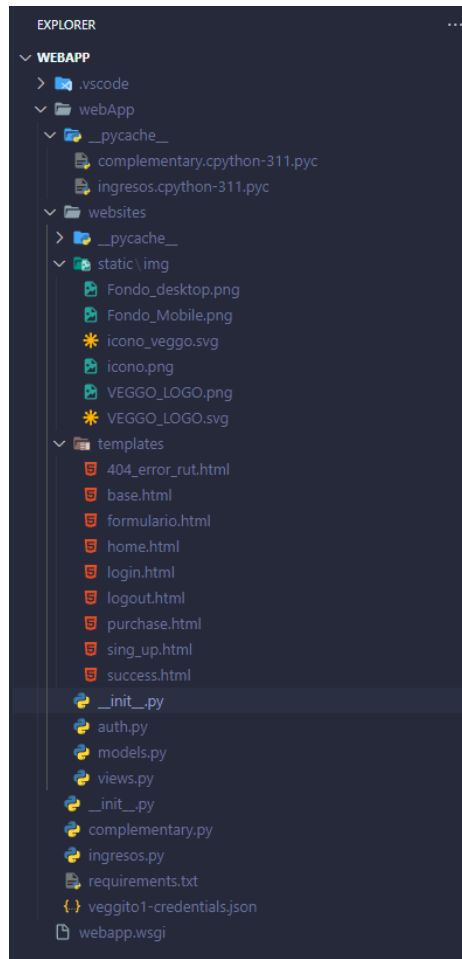


Imagen 3: Estructura ficheros tomador de pedidos. La carpeta “/websites” es donde se alojan las principales funciones para las páginas web. Dentro de la subcarpeta “/templates” se encuentran las plantillas de las páginas web. En la subcarpeta “/static/img” es donde se guardan las imágenes del proyecto. Por último, la carpeta global aloja el archivo iniciador de la aplicación Flask al igual que archivos con funciones adicionales.

Iniciador aplicación Flask

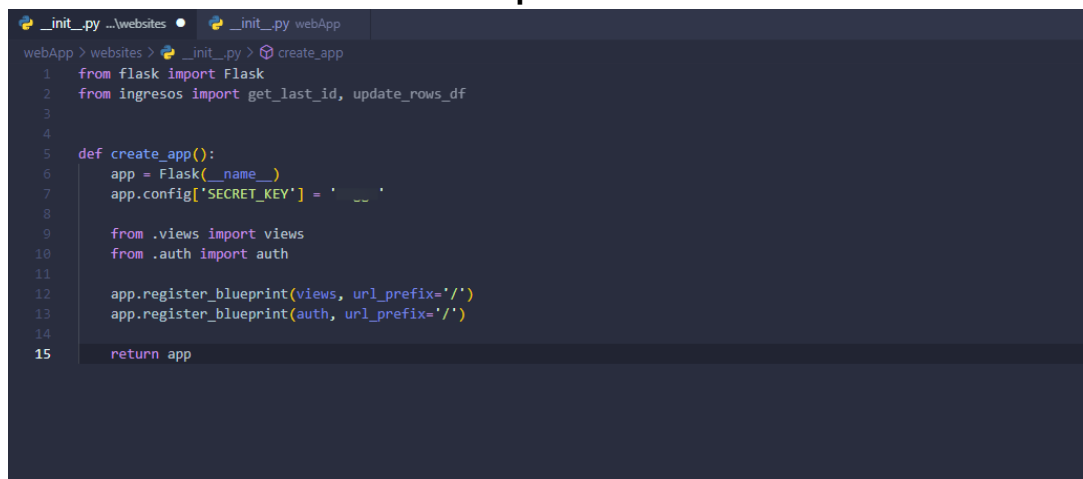


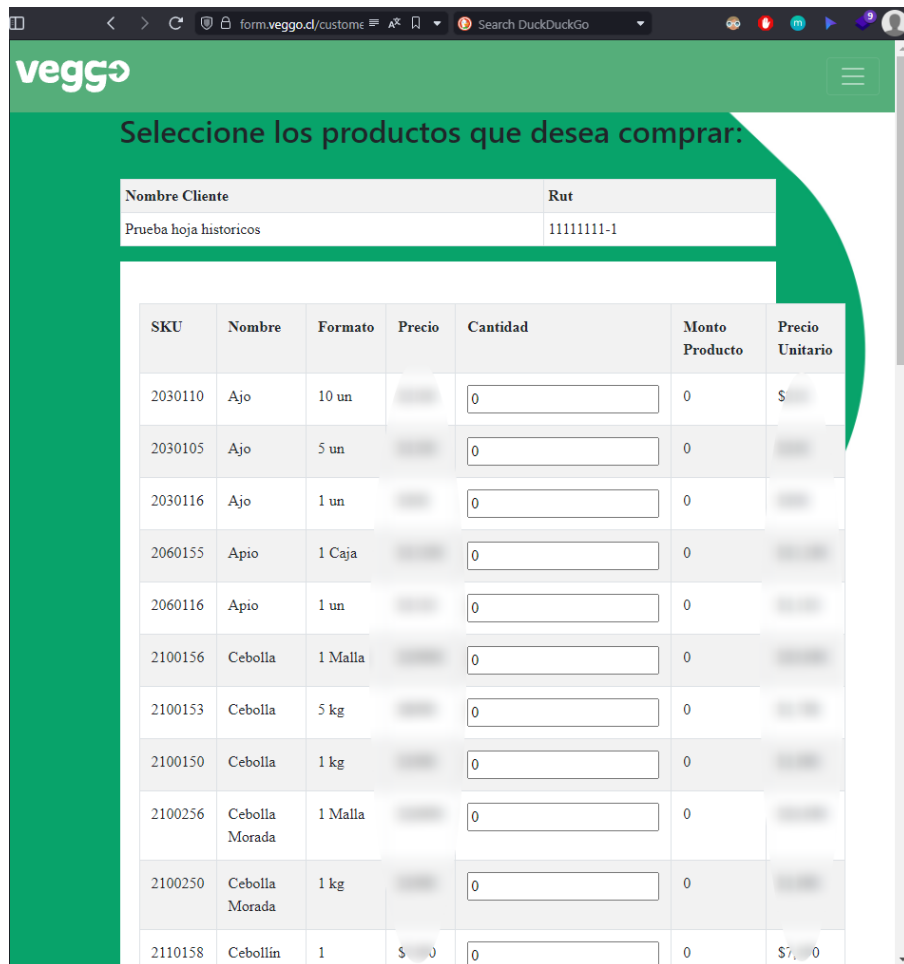
Imagen 4: Archivo destinado a la iniciación de la aplicación web y su configuración. Se importan los esquemas requeridos para el funcionamiento de las rutas de la aplicación (páginas web y funciones Backend).

Extracto plantillas archivo views.py

```
views.py X
webApp > websites > views.py > guardar_cliente
1 from flask import Blueprint, render_template, request
2
3 import json
4 import gspread as gs
5 import pandas as pd
6 from ingresos import *
7 from complementary import *
8
9 # Credenciales Google Workspace
10 client = gs.service_account(filename='webApp/veggito1-credentials.json')
11
12 # Declaracion Blueprint Views
13 views = Blueprint('views', __name__)
14
15
16 @views.route('/')
17 def home():
18     return render_template("home.html")
19
20
21 @views.route('/customer', methods=['POST'])
22 def guardar_cliente():
23     try:
24
25         rut = request.form['rut']
26
27         spreadsheet = client.open('Orden de compra - Almacenes')
28         sheet = spreadsheet.worksheet('Catalogo Clientes')
29
30         clientes = pd.DataFrame(sheet.get_all_records())
31         info_cliente = clientes.loc[clientes['Rut'] == rut ]
32         print(f'Catalogo Cliente:\n{info_cliente["Tipo Catalogo"]}')
33
34         # Catalogo Cliente
35         tipo_catalogo = str(info_cliente['Tipo Catalogo'].values[0])
36
37         sheet_productos = spreadsheet.worksheet(tipo_catalogo)
38
39         productos = pd.DataFrame(sheet_productos.get_all_records())
40         productos = productos.rename(columns={
41             'Productos': 'Nombre',
42             'Precio Venta': 'Precio'
43         })
44
45         productos['Precio'] = productos['Precio'].str.replace('$', '').str.replace(',','').astype(int)
46         productos = productos.sort_values(by='Nombre')
47
48         sheet_cobro_despachos = spreadsheet.worksheet('CobroDespachos')
49         cobro_despachos = pd.DataFrame(sheet_cobro_despachos.get_all_records())
50
51         return render_template('purchase.html', productos=productos.to_dict(orient='records'),
52                                info_cliente=info_cliente.to_dict(orient='records'))
53     except Exception as e:
54         return render_template('404_error_rut.html')
```

Imagen 5: Extracto de código destinado al manejo de peticiones y datos por parte del usuario. En este extracto se muestra la página de inicio donde el cliente ingresa su Rut y es redirigido a la ruta "/customer" donde se procesa los datos del cliente al igual que su catálogo personal.

Información de cliente



vegg

Seleccione los productos que desea comprar:

Nombre Cliente	Rut
Prueba hoja historicos	11111111-1

SKU	Nombre	Formato	Precio	Cantidad	Monto Producto	Precio Unitario
2030110	Ajo	10 un		<input type="text" value="0"/>	0	\$
2030105	Ajo	5 un		<input type="text" value="0"/>	0	
2030116	Ajo	1 un		<input type="text" value="0"/>	0	
2060155	Apio	1 Caja		<input type="text" value="0"/>	0	
2060116	Apio	1 un		<input type="text" value="0"/>	0	
2100156	Cebolla	1 Malla		<input type="text" value="0"/>	0	
2100153	Cebolla	5 kg		<input type="text" value="0"/>	0	
2100150	Cebolla	1 kg		<input type="text" value="0"/>	0	
2100256	Cebolla Morada	1 Malla		<input type="text" value="0"/>	0	
2100250	Cebolla Morada	1 kg		<input type="text" value="0"/>	0	
2110158	Cebollín	1	\$	<input type="text" value="0"/>	0	\$7,00

Imagen 6: Página “customer”, es esta se despliega el catálogo del cliente al igual que su información. Los precios no son mostrados en la imagen por confidencialidad.

A raíz de la instauración de la aplicación web el flujo de compra cambió considerablemente al cómo era antes de este proyecto.

Flujo de compra actual

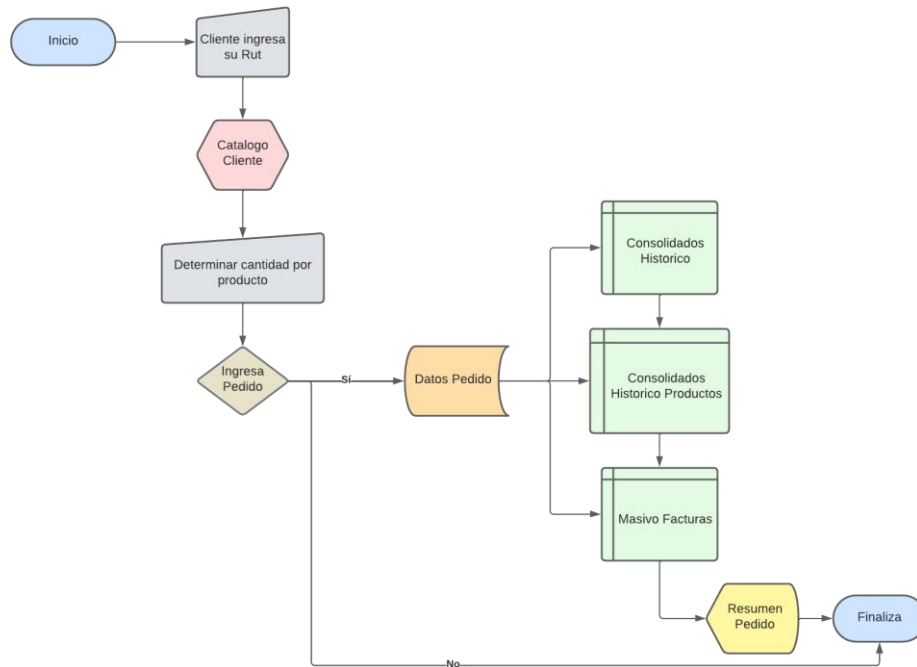
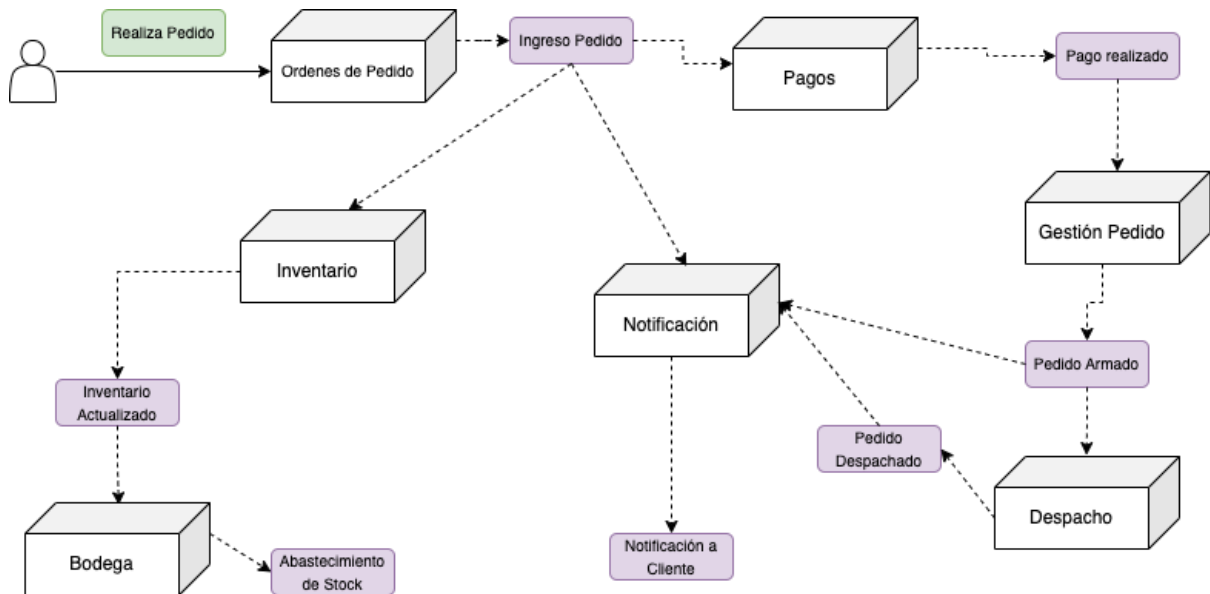


Diagrama de flujo 2: Este diagrama muestra cómo se comporta el sistema para el de ingreso de un pedido al BackOffice. Para revisar el diagrama de flujo anterior ir a apartado de oportunidad.

En vista del buen funcionamiento de la aplicación web se ha optado por escalar este proyecto, como es mencionado en el apartado de oportunidades lo realizado estos últimos meses es el hincapié para un futuro desarrollo e instauración de una arquitectura de sistemas para la empresa. En noviembre se centró en la investigación y desarrollo de un prototipo de infraestructura digital empresarial. No obstante, lo señalado anteriormente corresponde a recomendaciones para la empresa con el fin de establecer una arquitectura más robusta a futuro. A partir de esto, se sugiere optar por una arquitectura basada en eventos o EDA (Event Driven Architecture) donde el evento que gatilla el procesamiento de los datos es el ingreso de un pedido. Se escoge esta opción dado que el evento afecta a los diferentes departamentos como es Contabilidad, Ventas, Compra y Operaciones, donde dentro de Operaciones se encuentra Inventariado y Bodega principalmente. Esto permitirá a futuro el poder automatizar la gran mayoría de las tareas de la empresa.

Event Driven Architecture



EDA: Arquitectura basada en Eventos, patrón de arquitectura de software donde acciones son realizadas en reacción a un evento.

10. Resultados

10.1 Resultados Obtenidos

A partir de las métricas, se ha obtenido un resultado positivo. Por una parte, tenemos que el 22% de los clientes que han sido habilitados para el uso de la plataforma web. Así, se obtuvo una reducción en el tiempo laboral destinado al ingreso de pedidos por el equipo de ventas de 1 hora y 12 minutos. Si bien, no son los resultados esperados, la satisfacción por parte del cliente es positiva, no teniendo la necesidad de volver al sistema tradicional.

Por otro lado, al tener más tiempo libre, el vendedor lo ha designado para captar potenciales nuevos clientes. Donde al principio de este periodo se tenía asignado un promedio de 5 visitas semanales a potenciales clientes, este aumentó a 8 visitas semanales. Por ende, con una venta de \$286.430 promedio mensual por cliente, permite aumentar los ingresos dependiendo de la cantidad de nuevos clientes.

Cabe destacar que, a causa de realizar el ingreso de pedidos de forma manual, el margen de error era más alto comparado con el uso de la plataforma. Por ejemplo, cuando el vendedor ingresa un producto a la orden de venta este era propenso a equivocarse en el formato del producto, llevando a un error que provocaba un gasto adicional a la empresa, como era la reposición de productos y el costo por el despacho. Esto disminuyó debido a que el cliente era el que realizaba el pedido y sin estar presionado por el tiempo este verifica que los productos son los requeridos.

11. Conclusiones

Se puede concluir que, si bien no se logró habilitar la plataforma a todos los clientes debido a sus restricciones, se logró completar el objetivo principal que era aligerar la carga puesta en los vendedores. Esto da la oportunidad de que la empresa escale sin tener que aumentar los recursos actuales, ya que, se están utilizando herramientas digitales en los procesos internos de la empresa.

Como resultado se obtuvo el primer paso para la digitalización global de la empresa, que era crear el canal digital para el área de ventas con la finalidad de facilitar la gestión de los pedidos, validando así, la oportunidad de construir una infraestructura nueva y completamente digital.

12. Anexos