

# MyMed: tu droguería de confianza

Emmanuel Quintero, Juan Buitrago, Juan Burbano, Manuela Mayorga, María Amú y Samuel Escalante.

Presentado a: Oscar Hernan Mondragón Martinez.

**Resumen** - El proyecto utiliza un conjunto de datos de la página "datos.gob" que incluye información sobre productos farmacéuticos, con columnas adicionales como precio, año de compra y año de caducidad. La arquitectura del sistema se basa en una aplicación web llamada "MyMed" con un backend en Node.js y un frontend en PHP. Consta de tres microservicios: "microMedic" para buscar y gestionar medicamentos, "microUsuarios" para autenticación y gestión de usuarios, y "microCompras" para la gestión de compras. Se evaluaron varias opciones de empaquetado y despliegue en contenedores y clusters de procesamiento distribuido, y se seleccionó Docker y Docker Swarm como las soluciones más adecuadas.

**Índice de Términos** - Aplicación web, Datos farmacéuticos, Microservicios, Docker.

## I. INTRODUCCIÓN

La creación de una droguería virtual busca ofrecer a los usuarios una experiencia de compra farmacéutica conveniente y accesible. Esta aplicación web brinda la posibilidad de explorar un amplio catálogo de productos, desde medicamentos recetados hasta suplementos vitamínicos, a través de una interfaz intuitiva. Basada en una arquitectura sólida con Node.js en el backend y PHP en el frontend, su despliegue eficiente y escalable se realiza mediante el uso de Docker. Esta herramienta de contenedores garantiza una gestión óptima del sistema al permitir un empaquetado y despliegue consistente en diferentes entornos. Los usuarios pueden acceder a información detallada sobre los productos, como precios y disponibilidad en el inventario, lo que les brinda una experiencia de compra cómoda y confiable con una amplia variedad de productos farmacéuticos de alta calidad y precios competitivos.

## II. ANÁLISIS

### A. Selección del dataset

Hemos obtenido un conjunto de datos de la página "datos.gob", que es un repositorio de datos abiertos a nivel mundial. Este conjunto de datos cuenta con "2510" registros de información centradas en la nómina de productos de venta directa en el ámbito farmacéutico. Inicialmente, el conjunto de datos constaba únicamente de cinco columnas: ID, descripción, laboratorio y condición de venta. Sin embargo, para llevar a cabo un análisis adecuado, hemos decidido añadir las siguientes columnas: precio, año de compra, año de caducidad, porcentaje de efectividad e inventario. Estas adiciones se realizaron teniendo en cuenta los requisitos específicos de cada columna y garantizando una lógica coherente para facilitar el posterior análisis.

A continuación, se podrá encontrar el dataset solo con seis filas, mostrando como ejemplo qué contiene cada columna.

TABLE I  
Dataset repartido en dos tablas.

ID	Descripción	Laboratorio
1	SPERTI PREPARATION H UNGÜENTO TÓPICO	LABORATORIOS WYETH LLC.
2	INFLUVAC VACUNA ANTI - INFLUENZA SUSPENSIÓN INYECTABLE	ABBOTT LABORATORIES DE CHILE LTDA.
3	BACITOPIC UNGÜENTO DÉRMICO	LABORATORIO CHILE S.A.
4	FLUAD VACUNA ANTIGRIPIAL SUSPENSIÓN INYECTABLE	NOVARTIS CHILE S.A.
5	GC-FLU VACUNA ANTIINFLUENZA INACTIVADA. FRACCIONADA SUSPENSIÓN INYECTABLE	LABORATORIO BIOSANO S.A.
6	GC-FLU PEDIÁTRICA VACUNA ANTIINFLUENZA INACTIVADA FRACCIONADA SUSPENSIÓN INYECTABLE	LABORATORIO BIOSANO S.A.

Condición Venta	Precio	Year Compra	Year Caducidad	Porcentaje de efectividad	Inventario
Indirecta	29.417	2022	2023	85%	100
Directa	87.013	2019	2025	87%	86
Directa	11.605	2022	2024	86%	83

Indirecta	78.297	2020	2024	84%	80
Directa	73.108	2021	2025	95%	73
Indirecta	11.707	2021	2023	99%	82

1) *ID*: Esta columna contiene el ID único de cada producto, lo cual nos permite identificar de manera única cada medicamento en el conjunto de datos.

2) *Descripción*: Esta columna nos proporciona el nombre del medicamento, lo cual es fundamental para identificar cada uno de ellos de manera clara y precisa.

3) *Laboratorio*: Aquí se indica el nombre del laboratorio o empresa que es titular del medicamento, es decir, el fabricante o responsable legal del producto.

4) *Condición Venta*: En esta columna encontramos la condición de venta del medicamento. En este caso, se trata de una condición de venta directa, lo que implica que puede ser adquirido sin necesidad de receta médica u otra restricción, y también la condición de venta indirecta

5) *Precio*: Esta columna indica el precio de cada medicamento. Es un dato relevante para analizar la variación de precios y realizar comparaciones entre diferentes productos.

6) *Año de compra y de Caducidad*: Estas columnas contienen el año en el que se compró cada medicamento y el año de caducidad correspondiente. Estos datos son de suma importancia, ya que nos permiten realizar análisis temporales y evaluar la frescura y vigencia de los productos.

7) *Porcentaje de Efectividad*: Esta columna nos brinda información sobre el grado de efectividad de cada medicamento. Este dato es valioso para evaluar la calidad y eficacia de los productos.

8) *Inventario*: Esta columna contiene información sobre la disponibilidad de los medicamentos en la tienda.

#### B. Generación y selección de alternativas de solución

Para MyMed, existen múltiples alternativas para empaquetarla en contenedores y desplegarla en un cluster de procesamiento de datos distribuido. En las opciones, podemos evaluar:

1) *Docker*: esta es una de las herramientas más populares para poder empaquetar aplicaciones en contenedores. Docker nos permitiría crear, desplegar y ejecutar nuestra aplicación en contenedores de manera rápida y fácil. Además, nos ofrece una gran cantidad de herramientas y servicios para la gestión de contenedores y la creación de imágenes.

2) *Apache Mesos*: esta plataforma de gestión de recursos distribuidos nos permitiría la ejecución de nuestra aplicación en contenedores en un cluster de procesamiento de datos distribuido. Mesos ofrece una gran flexibilidad para la gestión de recursos y la configuración de los nodos de procesamiento.

3) *Kubernetes*: esta plataforma de orquestación de contenedores nos permitirá automatizar el despliegue, la gestión y el escalado de MyMed en contenedores. Kubernetes se ha convertido en la herramienta de facto para la gestión de contenedores en clusters de procesamiento de datos distribuidos.

4) *OpenShift*: esta plataforma de contenedores de código abierto nos podría permitir la gestión de MyMed en contenedores en un cluster de procesamiento de datos distribuido. OpenShift ofrece una gran flexibilidad y personalización, así como una integración total con herramientas de DevOps.

5) *Docker Swarm*: la herramienta de orquestación de contenedores integrada en Docker nos permitiría desplegar y gestionar nuestra aplicación en contenedores en un cluster de procesamiento de datos distribuido. Docker Swarm es una solución sencilla y fácil de usar para la gestión de contenedores.

6) *Amazon ECS*: este servicio de contenedores totalmente administrado nos podría permitir ejecutar “MyMed” en contenedores en la nube de Amazon Web Services. ECS ofrece una gran escalabilidad y disponibilidad, así como una integración total con otros servicios de AWS.

7) *Google Kubernetes Engine*: este es un servicio de orquestación de contenedores completamente administrado que nos podría facilitar el ejecutamiento de nuestra aplicación en contenedores en la nube de Google Cloud. GKE ofrece una gran escalabilidad, disponibilidad y seguridad, así como una integración total con otros servicios de Google Cloud.

8) *Microsoft Azure Kubernetes Service (AKS)*: este es un servicio administrado de kubernetes que nos permitiría hacer el empaquetado de una aplicación en contenedores y el despliegue en clusters de procesamiento de datos distribuido en Microsoft Azure. Esta plataforma funciona en la nube de Microsoft donde quedan guardados los registros.

Después de considerar cuidadosamente las herramientas disponibles y los objetivos de nuestro proyecto, hemos determinado que Docker y Docker Swarm son las opciones más adecuadas para nuestro escenario.

Al elegir Docker y Docker Swarm, podemos aprovechar los beneficios de la portabilidad, escalabilidad y aislamiento que Docker proporciona, mientras utilizamos Docker Swarm para administrar y orquestar nuestros contenedores en un entorno distribuido. Esto nos permite desplegar nuestra aplicación de manera eficiente y satisfacer nuestras necesidades de manera efectiva, manteniendo la consistencia y la flexibilidad en todo el ciclo de la aplicación.

### C. Definición de la Arquitectura

La arquitectura completa del sistema se basa en la aplicación web "MyMed" desarrollada utilizando Node.js en el backend y PHP en el frontend. Para permitir el empaquetado y despliegue de la aplicación basada en microservicios y API Rest en un cluster de contenedores, se utilizará Docker y Docker Swarm como tecnologías principales.

Los componentes necesarios para esta arquitectura incluyen:

1) *Backend*: El backend de la aplicación consta de tres microservicios: "microMedic", "microUsuarios" y

"microCompras". Cada microservicio tiene sus propios archivos de models y controllers y desempeña una función específica dentro de la aplicación.

2) *Microservicio de Usuarios (microUsuarios)*: Este microservicio se encarga de la autenticación y gestión de usuarios. Permite a los usuarios registrarse, iniciar sesión y actualizar sus perfiles.

3) *Microservicio de Medicamentos (microMedic)*: Este microservicio proporciona funcionalidad para buscar medicamentos, ver su información y detalles, y administrar la lista de medicamentos de los usuarios. Permite crear nuevas entradas de medicamentos y actualizar las existentes.

4) *Microservicio de Compras (microCompras)*: Este microservicio gestiona las compras de los usuarios. Permite agregar medicamentos al carrito de compras, realizar compras y registra las compras realizadas en la base de datos. También proporciona el historial de compras a los usuarios.

5) *Docker*: Se utilizará Docker para empaquetar cada uno de los microservicios y crear contenedores independientes para cada uno de ellos. Esto garantiza la portabilidad y aislamiento de los microservicios, lo que facilita su despliegue en diferentes entornos.

6) *Docker Swarm*: Docker Swarm se utilizará como orquestador de contenedores para gestionar y coordinar los contenedores en el cluster. Permite escalar los microservicios, mantener su disponibilidad y equilibrar la carga de manera eficiente.

Con esta arquitectura, los microservicios de "MyMed" se empaquetarán como contenedores utilizando Docker y se desplegarán en un cluster de contenedores administrado por Docker Swarm. Esto proporciona un entorno escalable y flexible para la ejecución de los microservicios. Los usuarios podrán interactuar con la aplicación a través de la interfaz frontend en PHP, mientras que los microservicios trabajarán en conjunto para brindar la funcionalidad completa de la aplicación, incluyendo la autenticación de usuarios, la gestión de medicamentos y la administración de compras.

#### D. Pipeline, componentes y algoritmos a utilizar:

A)

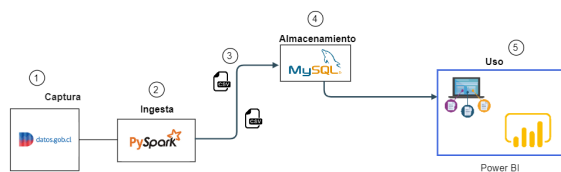


Fig 1. Pipeline del procesamiento de los datos

##### 1) Captura desde "datos.gob":

Para obtener el dataset de la nómina de productos de venta directa en el ámbito farmacéutico, se accedió a la página datos.gob, específicamente a la sección que contenía el conjunto de datos de interés. Se realizó una búsqueda detallada para identificar el dataset correcto y se procedió a descargarlo y acceder a los datos necesarios. Se verificó que se obtuviera la totalidad del dataset y además se le añadieron unas columnas como se había mencionado anteriormente para su posterior análisis.

##### 2) Ingesta con PySpark:

Se configuró el entorno de PySpark en el sistema y se importaron las bibliotecas necesarias. Se utilizó PySpark para cargar los datos descargados previamente. Usamos el formato del archivo en CSV, se utilizó la función correspondiente de lectura de datos de PySpark para crear un DataFrame de Spark. Esto permitió realizar operaciones eficientes y escalables en los datos.

##### 3) Almacenamiento en MySQL:

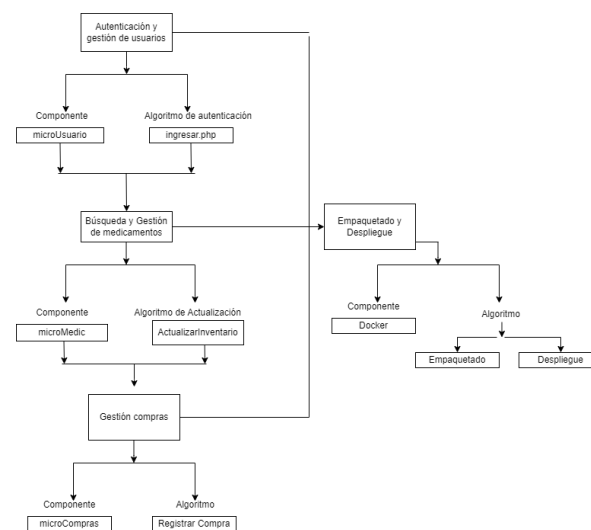
Se configuró la conexión con una base de datos MySQL, asegurándose de tener los controladores adecuados y la información de acceso correcta. Se utilizó PySpark para transformar y preparar los datos según fuera necesario para su almacenamiento en MySQL. Posteriormente, se utilizó la función de escritura de datos de PySpark para guardar el DataFrame de Spark en una tabla de la base de datos MySQL.

##### 4) Uso

Después de lanzar la aplicación a través de Docker y almacenar los datos en MySQL, pudimos aprovecharla para realizar análisis y obtener información analítica relevante. A continuación, se utilizó Power BI como herramienta de visualización de datos para generar un

informe completo que mostrará la información analítica obtenida. Este informe, presentado a la farmacia, brindará una visión clara de la demanda de medicamentos a lo largo del tiempo y les permitirá tomar las medidas correspondientes en la gestión de su inventario. Con esta solución integral, la farmacia podrá ajustar su inventario de manera inteligente, aumentando la disponibilidad de medicamentos durante los años con mayor demanda y evitando situaciones de escasez.

#### B) Componentes - servicios y algoritmos:



**B.1) Componente:** Microservicio de Usuarios (microUsuarios).

- **Algoritmo de autenticación:** Este algoritmo se implementó en el código como "ingresar.php", donde se lleva a cabo el proceso correspondiente para el inicio de sesión, teniendo en cuenta si es usuario jefe o usuario normal.

**B.2) Componente:** Microservicio de Medicamentos (microMedic).

- **Algoritmo de actualización:** Para permitir la creación y actualización de medicamentos.

**B.3) Componente:** Microservicio de Compras (microCompras).

- **Algoritmo de registro de compras:** Para registrar las compras realizadas por los usuarios

**B.4) Componente:** Docker.

- **Algoritmo de empaquetado:** Docker permitió empaquetar cada microservicio en un contenedor independiente. El empaquetado se realizó utilizando

un archivo Dockerfile que definió las dependencias y configuraciones necesarias para el contenedor.

- Algoritmo de despliegue: Docker Swarm se encargó del despliegue de los contenedores en el cluster, utilizando algoritmos de programación distribuida y equilibrado de carga para garantizar la disponibilidad y rendimiento de los microservicios.

### III. DISEÑO

#### A) Diagrama de los componentes

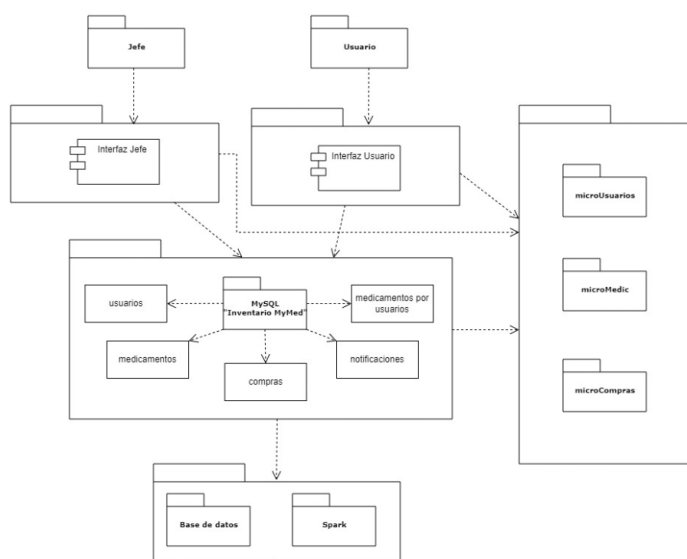


Fig 2. Diagrama de Componentes

#### B) Relación y flujo de trabajo entre los componentes:

En el sistema MyMed, se han desarrollado tres microservicios principales: microMedic, microUsuarios y microCompras. Cada uno de estos microservicios desempeña un papel específico dentro de la aplicación y maneja diferentes conjuntos de datos.

Cada microservicio tiene sus propios archivos de modelos (models) y controladores (controllers) para gestionar los datos y la lógica de negocio asociada.

Estos archivos son responsables de interactuar con la base de datos y proporcionar las funcionalidades requeridas por cada microservicio.

En conjunto, estos tres microservicios colaboran para ofrecer la funcionalidad completa de MyMed, permitiendo a los usuarios autenticarse, buscar medicamentos, agregarlos a su lista personal, realizar compras y acceder a su historial de compras.

#### C) Descripción de los componentes:

##### C.1) Microservicio de Usuarios (microUsuarios):

Funciones principales:

- Registro de usuarios: Permite a un usuario jefe registrar en la aplicación a un nuevo usuario proporcionando información básica como nombre, usuario, contraseña y rol.

- Inicio de sesión de usuarios: Permite a los usuarios iniciar sesión en la aplicación utilizando sus credenciales registradas

- Validación de credenciales: Permite a cada usuario ingresar a la interfaz de usuario mediante una validación previa de que su usuario si exista en la base de datos.

##### C.2) Microservicio de Medicamentos (microMedic):

Funciones principales:

- Visualización de detalles de medicamentos: Proporciona información detallada sobre un medicamento específico, como su descripción, precio por unidad y la cantidad que hay en el inventario

- Creación y actualización de entradas de medicamentos: Permite al usuario jefe de MyMed agregar nuevos medicamentos a la base de datos o actualizar la información de los medicamentos existentes.

##### C.3) Microservicio de Compras (microCompras):

Funciones principales:

- Gestión del carrito de compras: Permite a los usuarios agregar medicamentos al carrito de compras y realizar cambios en las cantidades de medicamentos seleccionados.

- Procesamiento de compras: Realiza la lógica de negocio para procesar las compras de medicamentos, calcular el costo total y la fecha en que fue realizada la compra

- Registro de historial de compras: Guarda registros de las compras realizadas por los usuarios, incluyendo detalles como los medicamentos comprados, el costo total, la fecha de compra, etc.

- Acceso al historial de compras: Permite a los usuarios acceder a su historial de compras y al usuario jefe ver un resumen de las compras anteriores realizadas por todos los usuarios en MyMed.

##### C.4) Base de datos MyMed (MySQL):

Esta base de datos contiene 5 tablas (usuarios, compras, medicamentos, medicamentos por usuarios, notificaciones), estas almacenan los datos que se van obteniendo de la aplicación)

C.5) *Spark*: Este componente se encarga de la analítica realizada apartir de la base de datos que fue utilizada para crear la solución (la aplicación)

#### D) Diagrama de despliegue

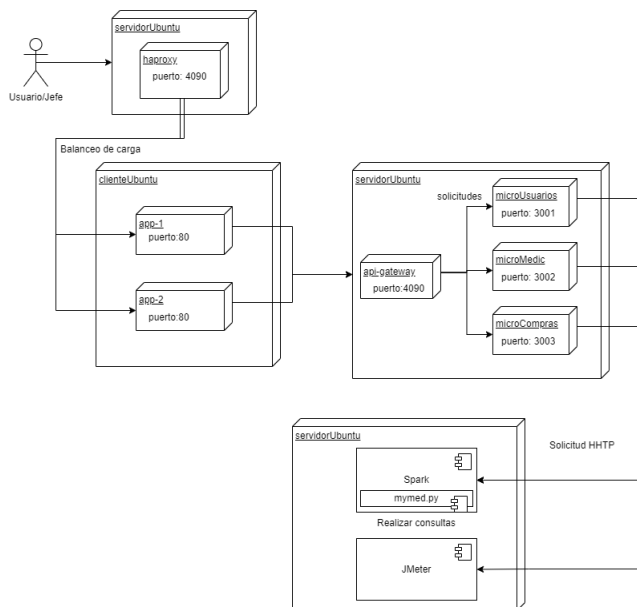


Fig 3. Diagrama de Despliegue.

## IV. IMPLEMENTACIÓN Y PRUEBAS

### A) Solución diseñada

Para abordar eficazmente la gestión de inventario en farmacias, es crucial contar con información precisa sobre la demanda de medicamentos. En este contexto, se implementó un entorno de contenedores utilizando Docker, lo que permitió una configuración y despliegue sencillos de la solución. Posteriormente, se utilizó PySpark para llevar a cabo el análisis de datos, procesando las fechas de compra de los medicamentos. A través de este proceso, se determinó el año en el que se registraron la mayor cantidad de compras y, por lo tanto, se experimentó una mayor demanda de medicamentos.

A continuación, se muestran los pasos para realizar el análisis de los datos obtenidos:

#### Análisis descriptivo:

Utilizando PySpark, se aplicó el método describe() a las columnas "precio\_unitario", "porcentaje\_efectividad" y "Ventas\_Totales\_Year". Esto proporcionó estadísticas descriptivas como recuento, promedio, desviación estándar, mínimo y máximo para cada una de las columnas mencionadas.

#### Análisis de tendencias:

Los datos se agruparon por el atributo "year\_compra" y se calculó el promedio de "precio\_unitario" y la suma de "Ventas\_Totales\_Year" para cada año de compra. Luego, los resultados se mostraron ordenados por el atributo "year\_compra", lo que permitió identificar las tendencias en relación con los precios unitarios y las ventas totales en cada año.

#### Segmentación:

Se realizó una segmentación de datos utilizando los atributos "laboratorio" y "condicion\_venta". Para el atributo "laboratorio", se calculó la suma de "Ventas\_Totales\_Year" para cada laboratorio y se mostraron los resultados ordenados por la suma de ventas totales, lo que permitió identificar los laboratorios con mayores ventas. Para el atributo "condicion\_venta", se calculó el promedio de "precio\_unitario" para cada condición de venta y se mostraron los resultados ordenados alfabéticamente por la condición de venta, lo que brindó información sobre los precios promedio asociados a cada condición de venta.

#### Análisis de caducidad:

Se calculó el tiempo restante de caducidad en años, tomando la diferencia entre los atributos "year\_caducidad" y "year\_compra". Luego, se realizó un conteo de medicamentos para cada período de tiempo restante de caducidad y se mostraron los resultados ordenados por el tiempo restante de caducidad, lo que proporcionó una visión de la distribución de los medicamentos en función de su tiempo de caducidad.

#### Análisis de precios:

Los datos se ordenaron en función del atributo "precio\_unitario" de forma descendente y se seleccionaron los cinco medicamentos con los precios unitarios más altos. A continuación, se realizó una segmentación por el atributo "laboratorio" y se calculó el promedio de "precio\_unitario" para cada laboratorio, mostrando los resultados ordenados por el promedio de precios unitarios de forma descendente. Este análisis permitió identificar los medicamentos más costosos y los laboratorios con precios promedio más altos.

### B) Pruebas de funcionamiento de los componentes

- Página de inicio de sesión:

**MYMED**  
**Venta y gestión de medicamentos**  
Aquí podrá comprar y gestionar sus medicamentos comprados.

**Iniciar sesión**  
 Usuario:  
  
 Contraseña:

## -Página de inicio de “user” (usuario normal):

**YOU MED**
[Comprar Medicamentos](#)
[Mis Compras](#)
Bienvenido, User

**Bienvenido a YOU MED**  
 Seleccione una opción en la barra de navegación para comenzar.

## -Página de comprar medicamentos de “user”:

**YOU MED**
[Comprar Medicamentos](#)
[Mis Compras](#)
Bienvenido, User

**Lista de Medicamentos**

Nombre	Precio	Inventario	Cantidad
SPIRIT PREPARACIÓN H UNIGÉNTICO TÓPICO	\$22,869.00	75	0
INFLUAVAC VACUNA ANTI - INFLUENZA SUSPENSIÓN INYECTABLE	\$24,400.00	89	0
BACTOPIC UNIGÉNTICO DÉRMICO	\$70,116.00	11	0
FLUAD VACUNA ANTIGRIPIAL SUSPENSIÓN INYECTABLE	\$51,304.00	35	0
GC-FLU VACUNA ANTINFLUENZA INACTIVADA, FRACCIONADA SUSPENSIÓN INYECTABLE	\$28,904.00	89	0
GC-FLU PEDIÁTRICA VACUNA ANTINFLUENZA INACTIVADA, FRACCIONADA SUSPENSIÓN INYECTABLE	\$67,232.00	94	0
FLUAZUR PEDIÁTRICO VACUNA ANTINFLUENZA INACTIVADA SUSPENSIÓN INYECTABLE	\$56,317.00	50	0
BIOWAC-FLU VACUNA ANTINFLUENZA INACTIVADA SUSPENSIÓN INYECTABLE	\$18,825.00	26	0
VAXOGRIP VACUNA ANTINFLUENZA INACTIVADA SUSPENSIÓN INYECTABLE	\$25,213.00	4	0
VAXOGRIP PEDIÁTRICO VACUNA ANTINFLUENZA INACTIVADA SUSPENSIÓN INYECTABLE	\$91,111.00	21	0
...			
EUROGISC INFANTE POLVO PARA SUSPENSIÓN ORAL 125 mg/5 mL CON SOLVENTE	\$62,737.00	2	0
DICLOFENACO DIETILAMONIO CREMA TÓPICA 1.15%	\$28,756.00	45	0
DICLOFENACO DIETILAMONIO CREMA TÓPICA 1.15%	\$67,413.00	86	0
ASAACOL SOLUCIÓN TÓPICA 0.1%	\$54,096.00	33	0
CEBION FRESCA SOLUCIÓN ORAL PARA GELAS 100 mg/mL	\$78,488.00	48	0
FOSFOLAMP SOLUCIÓN NASAL PARA NEBULIZACIÓN 0.1%	\$47,671.00	2	0
NAPRESEN GEL 10%	\$15,698.00	41	0
IPSON SUSPENSIÓN ORAL 100 mg/5 mL	\$39,867.00	66	0
OCUCORTECT FLUIDO SOLUCIÓN OFTÁLMICA 1%	\$11,247.00	68	0
HEADOL FLU DÍA POLVO PARA SOLUCIÓN ORAL EN SOBRES	\$62,709.00	10	0
SALONIPAS NUEVA FÓRMULA PARCHES TÓPICO	\$39,331.00	88	0
ZOREX COMPRIMIDOS MASTICABLES 5 mg	\$16,064.00	49	0

## -Página de “mis compras” de user:

**YOU MED**
[Comprar Medicamentos](#)
[Mis Compras](#)
Logout User

Usuario	Medicamento ID	Medicamento Nombre	Cantidad	Precio Total	Total de la Cuenta	Fecha de Compra
User	1	SPIRIT PREPARACIÓN H UNIGÉNTICO TÓPICO	#3	\$68,267	236267	May 19, 2021
	2	INFLUAVAC VACUNA ANTI - INFLUENZA SUSPENSIÓN INYECTABLE	#2	\$48840		
	3	BACTOPIC UNIGÉNTICO DÉRMICO	#1	\$70116		
	4	FLUAD VACUNA ANTIGRIPIAL SUSPENSIÓN INYECTABLE	#1	\$51104		

## -Ahora, iniciamos sesión como usuario “jefe”.

## -Página de inventario de “jefe”:

MY MED				INVENTARIO DE MEDICAMENTOS	EDITAR MEDICAMENTO	AGREGAR NUEVO USUARIO	COMPRAS	NOTIFICACIÓN DE MEDICAMENTOS	LOGOUT JEFE
ID MEDICAMENTO	DESCRIPCIÓN	PRECIO (UNITARIO)	INVENTARIO						
1	SPIRIT PREPARACIÓN H UNIGÉNTICO TÓPICO	\$22,869.00	75						
2	INFLUAVAC VACUNA ANTI - INFLUENZA SUSPENSIÓN INYECTABLE	\$24,400.00	89						
3	BACTOPIC UNIGÉNTICO DÉRMICO	\$70,116.00	11						
4	FLUAD VACUNA ANTIGRIPIAL SUSPENSIÓN INYECTABLE	\$51,304.00	35						
5	GC-FLU VACUNA ANTINFLUENZA INACTIVADA, FRACCIONADA SUSPENSIÓN INYECTABLE	\$28,904.00	89						
6	GC-FLU PEDIÁTRICA VACUNA ANTINFLUENZA INACTIVADA, FRACCIONADA SUSPENSIÓN INYECTABLE	\$67,232.00	94						
7	FLUAZUR PEDIÁTRICO VACUNA ANTINFLUENZA INACTIVADA SUSPENSIÓN INYECTABLE	\$56,317.00	50						
8	BIOWAC-FLU VACUNA ANTINFLUENZA INACTIVADA SUSPENSIÓN INYECTABLE	\$18,825.00	26						
9	VAXOGRIP VACUNA ANTINFLUENZA INACTIVADA SUSPENSIÓN INYECTABLE	\$25,213.00	4						
10	VAXOGRIP PEDIÁTRICO VACUNA ANTINFLUENZA INACTIVADA SUSPENSIÓN INYECTABLE	\$91,111.00	21						
11	EUROGISC INFANTE POLVO PARA SUSPENSIÓN ORAL 125 mg/5 mL CON SOLVENTE	\$62,737.00	2						
12	DICLOFENACO DIETILAMONIO CREMA TÓPICA 1.15%	\$28,756.00	45						
13	DICLOFENACO DIETILAMONIO CREMA TÓPICA 1.15%	\$67,413.00	86						
14	ASAACOL SOLUCIÓN TÓPICA 0.1%	\$54,096.00	33						
15	CEBION FRESCA SOLUCIÓN ORAL PARA GELAS 100 mg/mL	\$78,488.00	48						
16	FOSFOLAMP SOLUCIÓN NASAL PARA NEBULIZACIÓN 0.1%	\$47,671.00	2						
17	NAPRESEN GEL 10%	\$15,698.00	41						
18	IPSON SUSPENSIÓN ORAL 100 mg/5 mL	\$39,867.00	66						
19	OCUCORTECT FLUIDO SOLUCIÓN OFTÁLMICA 1%	\$11,247.00	68						
20	HEADOL FLU DÍA POLVO PARA SOLUCIÓN ORAL EN SOBRES	\$62,709.00	10						
21	SALONIPAS NUEVA FÓRMULA PARCHES TÓPICO	\$39,331.00	88						
22	ZOREX COMPRIMIDOS MASTICABLES 5 mg	\$16,064.00	49						

## -Agregar medicamento como “jefe”:

**AGREGAR MEDICAMENTO**

Descripción del Medicamento

Precio Unitario

Inventario

## -Editar medicamento como “jefe”:

**MY MED**
[INVENTARIO DE MEDICAMENTOS](#)
[EDITAR MEDICAMENTO](#)
[AGREGAR NUEVO USUARIO](#)
[COMPRAS](#)
[NOTIFICACIÓN DE MEDICAMENTOS](#)
[LOGOUT JEFE](#)

ID Medicamento	Nuevo Inventario
1	72
2	67
3	10
4	34
5	89
6	94
7	50
8	26
9	4
10	21

**EDITAR MEDICAMENTO**

ID del medicamento

Nuevo Inventario

## -Crear usuario como “jefe”:



MY MED INVENTARIO DE MEDICAMENTOS EDITAR MEDICAMENTO AGREGAR NUEVO USUARIO COMPRAS NOTIFICACIÓN DE MEDICAMENTOS LOGOUT JEFE

Nombre	Usuario	Password	Rol
UsuarioComprador	User	1234	0
UsuarioJefe	Jefe	1234	1

CREAR USUARIO

MY MED INVENTARIO DE MEDICAMENTOS EDITAR MEDICAMENTO AGREGAR NUEVO USUARIO COMPRAS NOTIFICACIÓN DE MEDICAMENTOS LOGOUT JEFE

CREAR USUARIO

Nombre

UsuarioComprador

UsuarioJefe

Nombre

Usuario

Password

¿Es jefe?

Si

No

Crear Usuario

-Historial de las compras que puede ver el “jefe”:

MY MED Medicamentos Agregar Nuevo Usuario Compras Logout JEFE

Usuario	ID Medicamento	Nombre Medicamento	Cantidad	Precio Total Med.	Total Compra	Fecha Compra
User	3	BACTROPC UNGUENTO DERMICO	#1	\$70116	23607	May 19 2023
	4	FILUAD VACUNA ANTIGRIPIAL SUSPENSIÓN INYECTABLE	#1	\$51104		
	1	SPIRETI PREPARACIÓN UNGUENTO TÓRICO	#3	\$66207		
	2	INFLUNAC VACUNA INDI - INFLUENZA SUSPENSIÓN INYECTABLE	#2	\$48940		

-Notificación de medicamentos que están por debajo de once unidades, para poder actualizar inventario; desde usuario “jefe”:

MY MED INVENTARIO DE MEDICAMENTOS EDITAR MEDICAMENTO AGREGAR NUEVO USUARIO COMPRAS NOTIFICACIÓN DE MEDICAMENTOS LOGOUT JEFE

ID	INFORMACIÓN MEDICAMENTO	FECHA DE NOTIFICACIÓN
1	El medicamento (ID 3) BACTROPC UNGUENTO DERMICO está por debajo de 11 unidades (10)	2023-05-18T19:18:55.000Z

### C) Pruebas de escalabilidad y desempeño

Se llevaron a cabo pruebas de balanceo de carga utilizando HAProxy y JMeter en la aplicación Dockerizada. Se realizaron cuatro pruebas distintas, cada una con un total de 20,000 peticiones:

- Prueba 1: Aplicación sin escalar en Docker Swarm

En esta prueba, se evaluó el rendimiento de la aplicación sin realizar ningún escalado. Se utilizó HAProxy para distribuir la carga entre los contenedores de la aplicación y se enviaron 20,000 peticiones para medir el tiempo de respuesta y la capacidad de procesamiento.

- Prueba 2: Escalado a 3 réplicas del frontend de la aplicación

En esta prueba, se realizó un escalado del frontend de la aplicación a 3 réplicas utilizando Docker Swarm. Se configuró HAProxy para balancear la carga entre estas réplicas y se enviaron 20,000 peticiones para evaluar el impacto del escalado en el rendimiento y la capacidad de respuesta de la aplicación.

- Prueba 3: Escalado a 6 réplicas del frontend de la aplicación

En esta prueba, se aumentó aún más el escalado del frontend de la aplicación a 6 réplicas. HAProxy se

configuró para distribuir la carga de manera equitativa entre estas réplicas y se enviaron 20,000 peticiones para analizar el comportamiento del sistema con un mayor número de réplicas.

- Prueba 4: Escalado del backend y frontend de la aplicación

En esta prueba final, se realizó un escalado tanto del backend como del frontend de la aplicación. Se aumentó el número de réplicas tanto en el frontend como en el backend, y se utilizó HAProxy para balancear la carga entre todas las réplicas. Se enviaron 20,000 peticiones para evaluar el rendimiento global del sistema después de este escalado completo.

Cada prueba permitió evaluar el rendimiento y la capacidad de respuesta de la aplicación bajo diferentes configuraciones de escalado. Los resultados obtenidos ayudaron a determinar cómo el escalado afecta el rendimiento y la capacidad de la aplicación para manejar una mayor carga de trabajo.

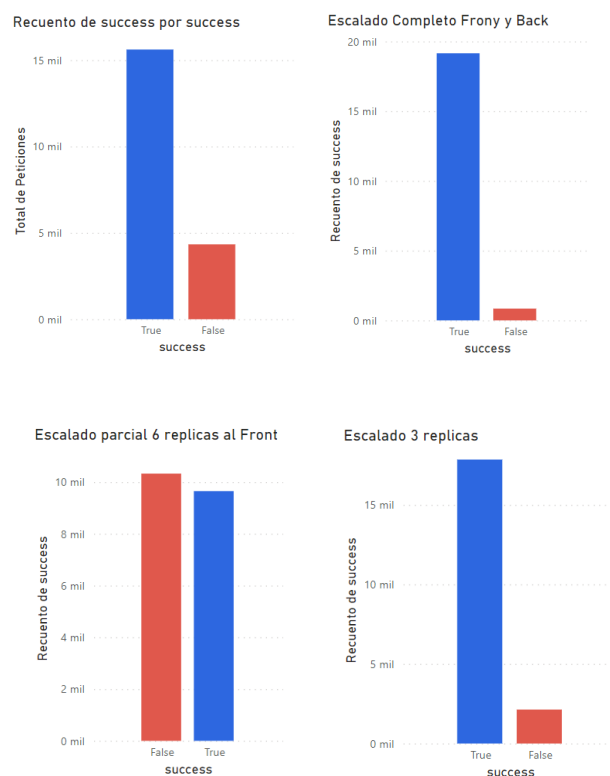


Fig 4. Diagramas de barras sobre las pruebas de escalabilidad.

6,98 mil	30,18 mil
Promedio	Promedio_EscalaCompleta





## V. ANALÍTICA

En la siguiente tabla, realizamos un análisis estadístico exhaustivo del conjunto de datos, obteniendo el recuento (count), el valor máximo (max), la media (mean) y el valor mínimo (min) de las variables porcentaje\_efectividad, precio\_unitario y suma de Ventas\_Totales\_Year.

summary	porcentaje_efectividad	precio_unitario	Suma de Ventas_Totales_Year
count	2500	2500	2500
max	100	25000	548000
mean	900332	141844	4370644
min	80	3000	328000
stddev	599447005401574	6694798336761749	6399703196774264
<b>Total</b>			<b>63997032023408</b>

Fig 5. Tabla de estadísticas generales.

Además, llevamos a cabo un análisis adicional utilizando el dataset correspondiente, centrándonos en el promedio del precio unitario por año. En este análisis, se evidenció que el año 2020 tuvo un promedio de 3,29%, el cual fue el porcentaje más bajo en comparación con los demás años. Como equipo, llegamos a la conclusión de que esta disminución en los precios de los medicamentos se debió a la situación de la pandemia por COVID-19 que se estaba desarrollando en ese año. Los laboratorios correspondientes tomaron la decisión de reducir los precios para que las personas pudieran acceder a los medicamentos y así combatir eficazmente dicho virus.

Promedio de avg(precio\_unitario) por Año

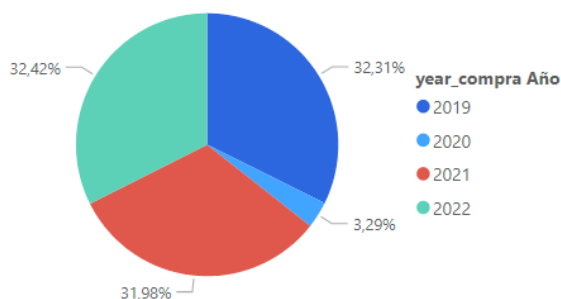


Fig 6. Gráfico circular sobre el promedio de precio unitario por año.

También nos propusimos analizar un gráfico de barras que mostrara la suma de ventas totales por laboratorio.

A partir de esta gráfica, se pudo observar que el laboratorio líder en ventas fue "KNOP LABORATORIOS S.A." con un total de 149 millones en ventas totales.

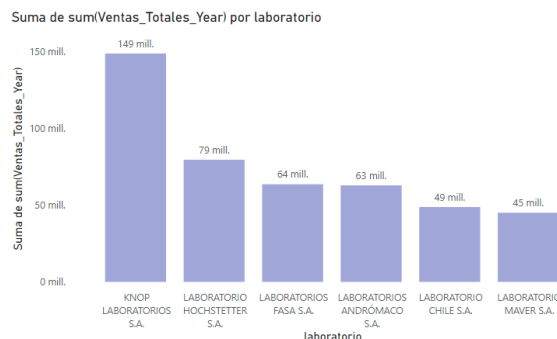


Fig 7. Diagrama de barras sobre la suma de ventas totales por laboratorio.

## VI. CONCLUSIONES

1. La arquitectura sólida de la aplicación, con Node.js en el backend y PHP en el frontend, sugirió un rendimiento eficiente y una respuesta rápida de la plataforma. Esto significa que los usuarios pueden esperar una experiencia de navegación fluida y sin demoras.
2. El uso de Docker para el despliegue eficiente y escalable de la aplicación fue una estrategia inteligente. Docker permitió un empaquetado y despliegue consistente en diferentes entornos, lo que facilitó la gestión y escalabilidad del sistema.
3. Al utilizar HAProxy como un balanceador de carga, se puede distribuir eficientemente el tráfico de usuarios hacia múltiples instancias de la aplicación, garantizando una alta disponibilidad y una respuesta rápida.

## VII. REFERENCIAS

- [1] Ministerio de Ciencia, Tecnología, Conocimiento e Innovación de Chile. (Año). Datos Abiertos de Chile. Recuperado de <https://datos.gob.cl/en/dataset/10071>. Fecha de acceso: 11 Mayo 2023.
- [2] Escalante, S. (2021). ProyectoRedes\_MyMed [Repositorio de código]. GitHub. Disponible en: [https://github.com/SamuelEscalante/ProyectoRedes\\_MyMed](https://github.com/SamuelEscalante/ProyectoRedes_MyMed). Accedido: 19 de mayo de 2023.