

**Sistema Unificado de Medicamentos - Everdose  
Software Architecture Document**

**Version 2.0**

**Software Architecture Documentation**



**Everdose**  
**Servicio Unificado de Medicamentos**



**Prepared By:**  
**Karol Valentina Yate**  
**Gustavo Mendez**  
**Juan Diego Osorio**  
**Juan Sebastián Valderrama**

**11 de mayo de 2020**  
**Pontificia Universidad Javeriana**

Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

## Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms and Abbreviations	5
2. Architectural Representation	6
3. Architectural Goals and Constraints	6
3.1 Scalability	6
3.2 Interoperability	6
3.3 Persistence	6
3.4 Reliability/Availability (failover)	6
3.5 Performance	6
4. Use-Case View	7
5. Logical View	8
6. Deployment View	9
7. Implementation View	10
8. Data View	11

Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

# Software Architecture Document

## 1. Introduction

En el presente documento es creado por el grupo Arquisofttech para describir el sistema de Everdose, un servicio unificado de medicamentos para el manejo desde la petición, compra y despacho de medicamentos para todo tipo de clientes existentes. Se describe en el mismo documento la vista arquitectónica que se va a utilizar para cumplir los objetivos de calidad propuestos por el grupo, junto con un conjunto de vistas que; dada su naturaleza, permitirán una explicación mucho más formal sobre el cómo se realizó la plataforma de Everdose y sus funcionalidades definitivas.

### 1.1 Purpose

En Colombia existe una gran variedad de distribuidores de productos farmacéuticos; inclusive, el listado de las droguerías llega alrededor de 28 entidades operando únicamente en Bogotá. Sin embargo, la mayoría no cuenta con un catálogo online para la comercialización, y aún menos son las farmacias o laboratorios que poseen una tienda en línea para la venta y distribución de estos productos de primera necesidad.

Ante esta problemática que poseen las personas para obtener medicamentos, sobre todo quienes no se pueden desplazar de su domicilio, y aún más, frente a la actual pandemia que crece a pasos agigantados de COVID-19, se hace necesaria una plataforma que permita la compra de productos online y como valor agregado, brinde las herramientas necesarias para localizar los fármacos independientemente de la droguería que se encuentre, ya que el desabastecimiento de los productos más especializados en las tiendas es bastante común hoy en día, gracias a la situación actual que está viviendo el mundo. Con esta inspiración, el grupo Arquisofttech desea brindar una solución de software que permita suplir estas necesidades que pueda permitir la mejora en la calidad de vida de los ciudadanos con ayuda de la tecnología y la informática para la adquisición de estos recursos de manera digital.

### 1.2 Scope

Este documento se desarrollará en el marco de la segunda entrega del proyecto **Servicio unificado de medicamentos** (SUM). Es basado sobre todo en la arquitectura base que fue implementada en el desarrollo de software, utilizando dos herramientas diferentes (JavaEE y Vue).

El alcance de este proyecto se limitará a la implementación de la arquitectura y el sistema mencionado anteriormente utilizando las nuevas tecnologías manejadas hoy en día en el sector de la industria (micro-services). Particularmente se implementarán las siguientes funcionalidades:

- Creación del perfil de usuario a partir de un correo e información personal y financiera. Podrá ser modificado y eliminado según el criterio del usuario.
- Creación del perfil de los proveedores que permitirán la venta de fármacos registrados en el inventario propio. El sistema almacenará el historial de compras del usuario. Podrá ser modificado y eliminado según las alianzas hechas con los laboratorios.
- Inicio de sesión desde cualquier dispositivo con conexión a internet
- Realizar consultas por parte de los usuarios de los medicamentos existentes.
- Se debe brindar la entrega del medicamento con el menor precio existente en el mercado, que cumpla con las condiciones de entrega menor a 45 minutos
- Debe permitir la solicitud de uno (1) o más medicamentos en un solo pedido, así vengan de diferentes sucursales.
- Si no existe disponibilidad local del medicamento, se enviará la solicitud a otra sucursal para el envío.

Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

- Se debe realizar la solicitud a los proveedores para el aprovisionamiento del inventario, cuando llega al umbral mínimo de petición.
- Permitir el seguimiento del envío del medicamento. El usuario puede ver en donde se encuentra el medicamento en tiempo real, mientras llega a su domicilio.
- Se debe realizar la validación de la tarjeta de crédito. Si es local o internacional y hacer la enrutación hacia el ente adecuado.
- Realizar todas las operaciones de canje hechas en el día.
- Se debe realizar un reporte on-line de las transacciones hechas al día.
- Página de FAQ para los usuarios.

Cabe aclarar, que en el alcance de este proyecto no se contempla la posibilidad de envíos a otros países o ciudades diferentes de Bogotá. De igual manera, este proyecto no abarca ninguna funcionalidad adicional a las mencionadas anteriormente.

### 1.3 Definitions, Acronyms and Abbreviations

- **Proveedores:** Distribuidores farmacéuticos seleccionados para la provisión de productos, un API estándar es supuesto.
- **Droguería:** Local comercial físico de venta y distribución de medicamentos con o sin prescripción médica.
- **Fármaco / medicamento:** Productos farmacéuticos que permiten la cura de una enfermedad de un usuario. Estos productos son producidos por los laboratorios farmacéuticos y vendidos a través de las droguerías.
- **Repartidor / despachador:** Persona o empresa dedicada al envío de productos. Son empresas vinculadas a la organización (Rappi, Domicilios, etc.)
- **Inventario:** La cantidad de medicamentos de cada una de las referencias que posee una droguería o sucursal. Cada una de ellas posee su propio inventario y lo gestiona de manera personal.
- **Alianza:** Son las distintas droguerías asociadas al sistema. En caso de no existir disponibilidad del medicamento en el sistema central, se le reenvía la solicitud a la droguería aliada más cercana y con menor precio.
- **Sucursal:** Es una sede aparte de la droguería principal. Se encuentran distribuidas en todo el territorio de Bogotá. Cada una de las droguerías tienen sus propias sucursales.
- **Administrador:** Corresponde a un tipo de usuario en el sistema que se encarga de desempeñar funciones de administración del inventario.
- **Transacción:** Una transacción es la realización de un pago a los proveedores, las droguerías o a los repartidores, que es necesaria para el funcionamiento del negocio; o un pago de un cliente al negocio a cambio de los servicios ofrecidos por este último.
- **Reporte:** Es una recopilación de información ordenada que describe las transacciones realizadas por el sistema.
- **Pedido:** Un pedido es una colección de medicamentos solicitados por un cliente, los cuales deben ser entregados al mismo.

Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

## 2. Architectural Representation

El sistema fue diseñado utilizando una arquitectura estándar de microservicios, en la cual se realizó una división completamente diferente de los diferentes actores en los cuales se divide el proyecto, partiendo desde los clientes, pasando por el manejo del stock y finalizando con el despachador. A nivel de Frontend, fue completamente implementado utilizando Vue.js, ya que un miembro del equipo tenía conocimiento previo de la herramienta y gracias a la aprobación del ing. Cesar Bustacara por haber trabajado utilizando esta arquitectura. Sin embargo, el resto del contenido del proyecto fue desarrollado completamente con las herramientas obtenidas en el curso utilizando Java Versión Empresarial (JavaEE) y manejando como base de datos relacional MySQL.

## 3 Architectural Goals and Constraints

Ya que deseábamos cumplir con algunos atributos de calidad desde la parte del diseño, utilizamos la arquitectura de microservicios para poder alcanzar con esta meta, ya que este estándar de diseño nos permitió tener desde el momento de su implementación, algunos valores agregados, los cuales serán descritos a continuación:

### 3.1 Scalability

- La arquitectura de microservicios permitió que cada servicio se ejecute como un componente independiente; es decir, cada componente de sí mismo, lo único que requiere de los demás es la comunicación y el paso de información entre los servicios de los que consume.

### 3.2 Interoperability

- Para garantizar la transparencia en el paso de información y evitar la pérdida o modificación sensible en el sistema, se optó por utilizar un formato tipo .JSON para este objetivo. Con ello, cada uno de los servicios se envían información y no hay problemas en la transformación de la misma. De igual forma, la arquitectura de microservicios implementada tiene un componente para paso de información (EJB) que permiten la comunicación de manera eficiente y única de cada servicio.

### 3.3 Persistence

- La persistencia fue implementada con lo aprendido en el curso de Arquitectura de Software. Esta fue implementada utilizando MySQL y las herramientas propias de JavaEE (JPA). La ventaja en este caso fue que cada microservicio que lo necesitara tenía un JPA propio, lo que le permitía el acceso a la base de datos en el momento que lo necesitara.

### 3.4 Reliability (failover)

- Para evitar los daños en la información si más de un usuario consumía recursos, se determinó un orden en el que los microservicios pueden utilizar los datos o la forma en la que estos consumían información unos de otro. De esta forma, se evita que más de un usuario esté en el mismo microservicio y pueda modificar la información hasta que el último de estos componentes realice su transacción final o retorne al último estado deseable de la transacción. Fue basado en los principios para las transacciones ACID.

### 3.5 Performance

- Esta es una de las ventajas más notorias en el uso de microservicios. Ya que cada uno se encuentra desplegado en un servidor distinto, cada uno tiene un nivel de rendimiento muy superior a comparación de una aplicación robusta que se despliega completamente en un único servidor o en dos, como era la naturaleza propia del proyecto. Además, las distintas operaciones únicamente consumen recursos en el microservicio que se está ejecutando. De esta forma, el proyecto posee un mejor desempeño que la arquitectura base implementada en la mayoría de proyectos del curso.

Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

## 4. Use-Case View

En esta sección, se identificarán los actores en el desarrollo del proyecto Everdose, así como los casos de uso que se decidieron implementar para cumplir los objetivos previamente establecidos desde la entrega pasada. En este caso, no se realizaron modificaciones al diagrama anterior, brindando una trazabilidad esperada en la implementación y el diseño.

La vista de casos de uso es la siguiente:

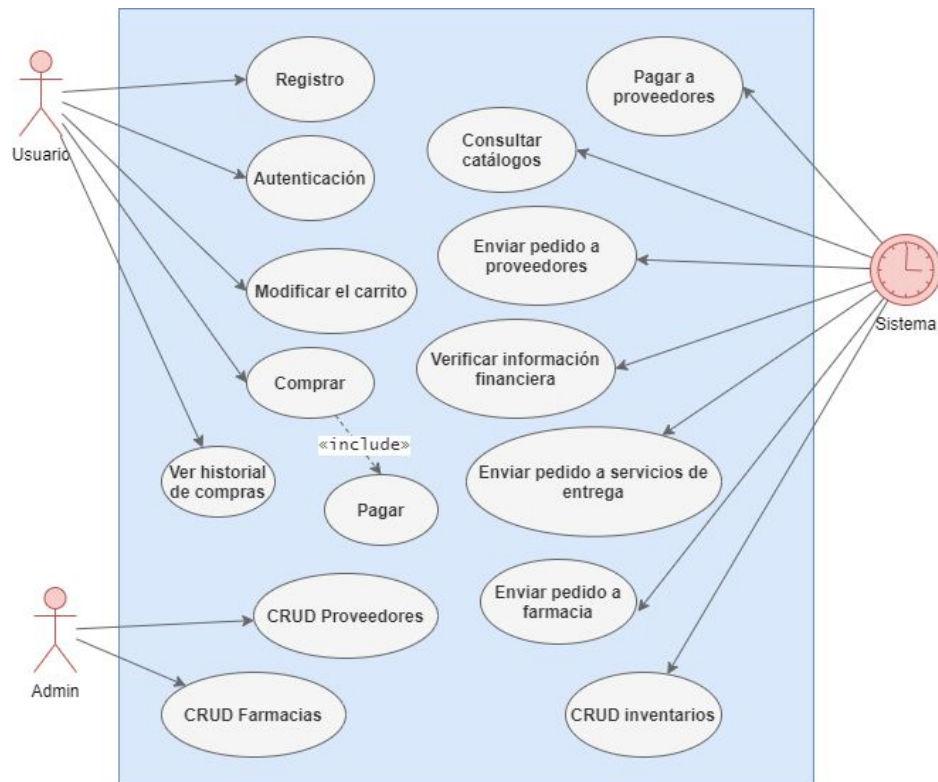


Diagrama No.1 Vista de Casos de Uso

En esta vista, es posible ver la existencia de 3 actores fundamentales

- Sistema:** Es el actor principal, comprendido a su vez como Everdose. El será el encargado de realizar los procesos de compra y venta de medicamentos. Se decidió dejar a parte a los proveedores en este diagrama, ya que su única función en el sistema es restablecer los inventarios que están a punto de acabarse y recibir las peticiones de Everdose. Igualmente, con las farmacias externas, al ser parte del BPO (Business Process Outsourcing) de la compañía; es decir, ser una especie de subcontratación de los procesos de envío de los medicamentos, ya que cada uno se encarga de su propio manejo de inventarios y envíos de manera externa, no se tiene en cuenta en este diagrama.

Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

- **Usuario:** Es la entidad que va a comprar en el sistema. Puede ser una persona natural o jurídica que busca algún recurso que Everdose tiene disponible. Tiene la facultad de realizar una autenticación en el sistema, realizar las compras, modificar su carrito y realizar el pago por dichos productos que desea.
- **Administrador:** O conocido también como la persona a cargo de Everdose. Es la persona capaz de gestionar los diferentes componentes que tiene el sistema para el público en general. Se encarga sobre todo del manejo del inventario, poder verificar cuales son los proveedores y las farmacias que están asociadas con Everdose y poder eliminarlos o agregarlos al sistema cuando desee, siempre y cuando sea beneficioso para la compañía estas decisiones.

## 5. Logical View

En la vista lógica se desea hacer una explicación del funcionamiento de Everdose, basándose exclusivamente desde el punto de vista del usuario, el cual comprende todos los tipos de actores existentes en el sistema Everdose. Sin embargo, es necesario explicar que el funcionamiento del usuario va a permitir mostrar todos los componentes del sistema sin tener que basarse en muchas perspectivas. De esta forma, el diagrama para la vista lógica es el siguiente:

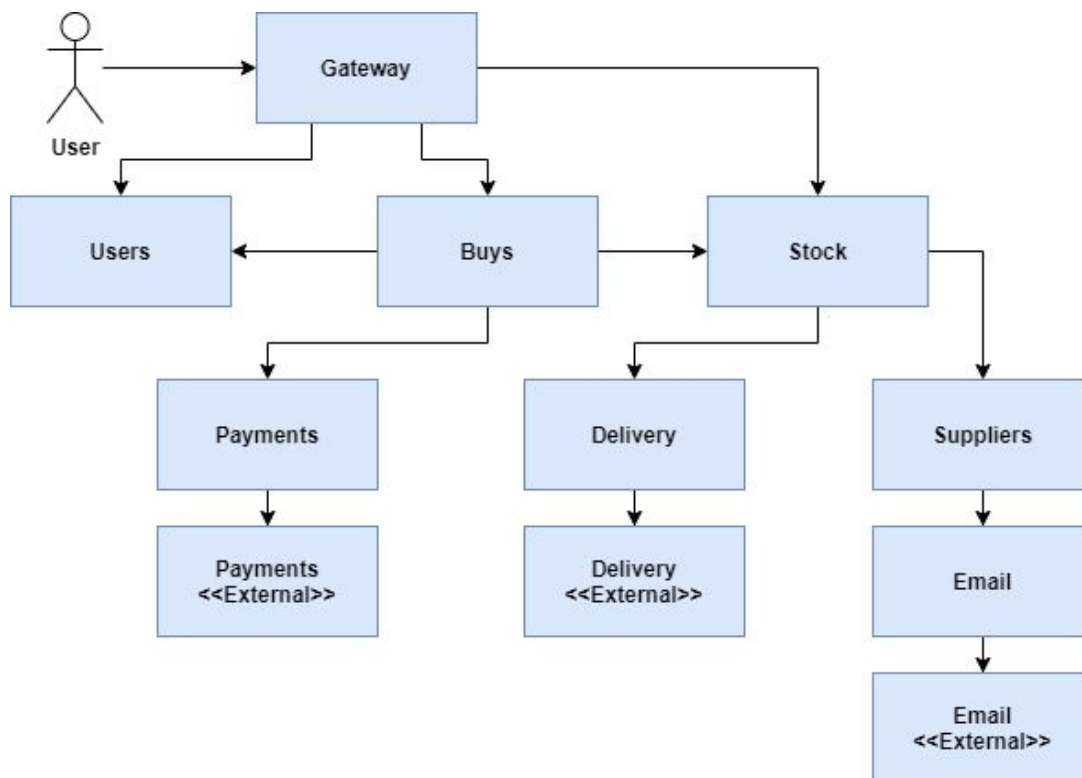


Diagrama No. 2 Vista lógica

En este diagrama podemos ver la interacción entre los diferentes microservicios con los que el usuario, o actor principal, va a utilizar a medida que se desenvuelva con la aplicación. Pasando desde un Gateway inicial, y alcanzando a llegar a todos los puntos del diagrama. Es necesario mencionar que no todos los usuarios van a usar el sistema en su totalidad, ya que el proceso para proveedores es diferente que el de un cliente normal.



Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

## 6. Deployment View

En la vista de despliegue, es posible identificar los elementos y las formas en las que cada uno de los componentes de los microservicios van a ser desplegados en la infraestructura física que se desea utilizar. La vista asociada a esta posee el modo en el que cada uno interactúa y sus conexiones, esto junto con las herramientas

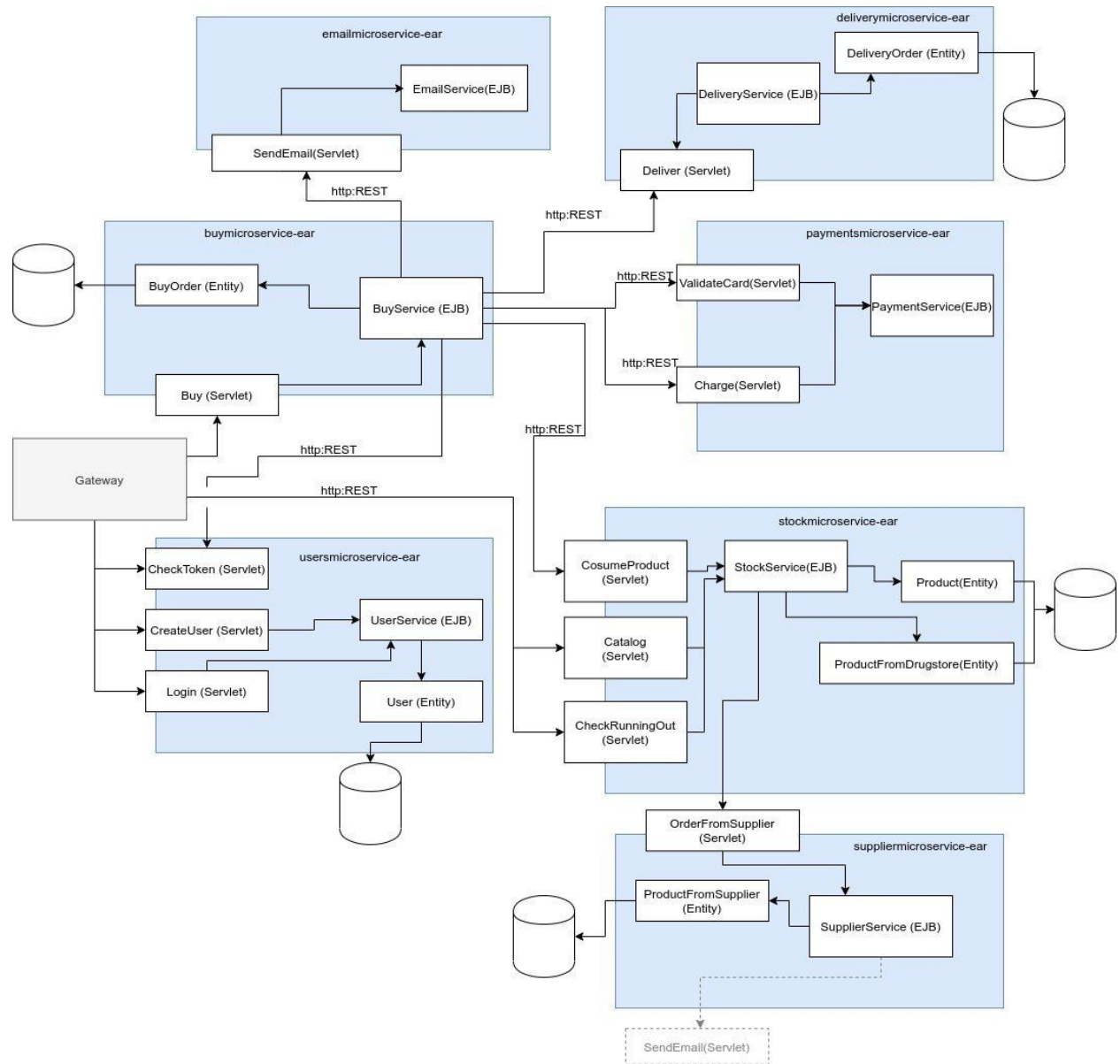


Diagrama No. 3 Vista de despliegue.

Como podemos ver en el diagrama, se verifican las herramientas brindadas por Java, uso de entidades, servlets y EAR's, estos a su vez conectados con dos bases de datos de las cuales consumen información. De esta forma, los componentes como los usuarios o el stock poseen características muy similares a este diagrama que, por cuestión de facilidad visual, no se presentan en el diagrama.

Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

## 7. Implementation View

Como parte del desarrollo, el siguiente diagrama permitirá explicar de manera similar el funcionamiento del sistema, con la diferencia que se muestran todos los actores que interactúan con el sistema y el modo en el que acceden cada uno a los distintos microservicios ofrecidos por Everdose. El diagrama es el siguiente:

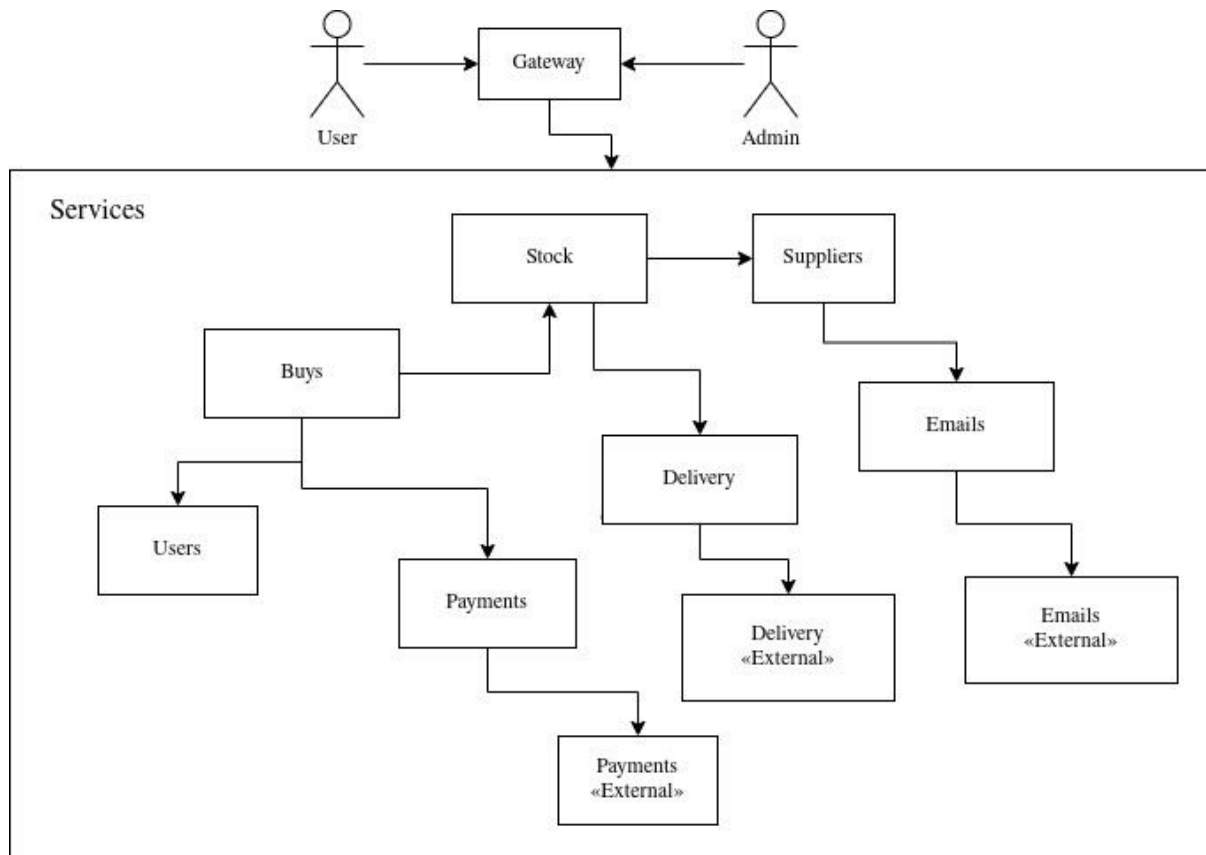


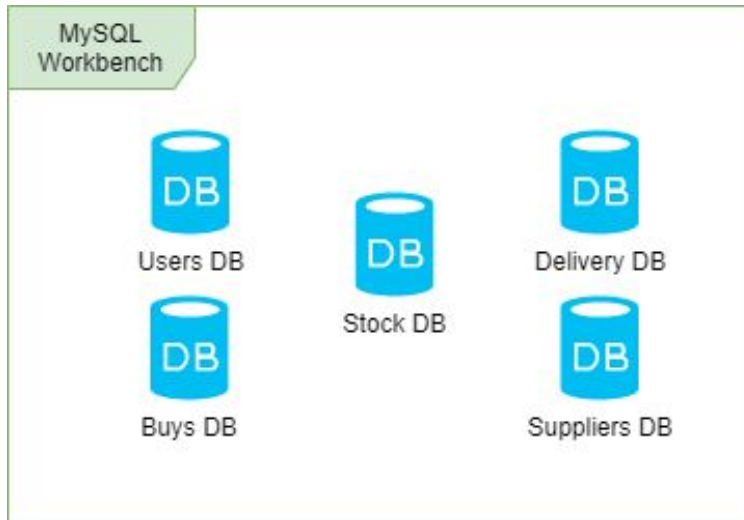
Diagrama No. 3 Vista de desarrollo

A pesar de que es muy parecido a la vista lógica, en este diagrama es posible detallar que son agrupados cada uno de los microservicios que ofrece la plataforma de Everdose. Sin embargo, cada uno de estos microservicios pueden estar desplegados desde máquinas o contenedores completamente diferentes. Además, aparece un nuevo tipo de usuario llamado “Admin”, el cual se divide del tipo de usuario normal. A pesar de tener ambos una misma entrada al sistema, las funcionalidades propias del administrador difieren del cliente frecuente, dado que el primero se encarga únicamente de la información existente en la página, mientras que el usuario normal entra con intenciones de ser un futuro comprador en la tienda, más no puede modificar nada a menos que haga una compra. En este caso, lo único que llegaría a afectar sería el stock del producto.

Una de las ventajas existentes en la implementación fue el manejo de payments. Esto es debido a que fue externalizada esta implementación utilizando Stripe,

Everdose - Servicio Unificado de Medicamentos	Version: 2.0
Software Architecture Document	Fecha: 11/05/2020

## 8. Data View



Como parte de la información requerida para el correcto funcionamiento de Everdose, y para evitar la volatilidad de un único punto de entrada para hacer la persistencia de la información. Archisofttech decidió utilizar 7 bases de datos en las cuales la información sería almacenada y utilizada, dependiendo de la seguridad que se necesite en cada caso, de tokens para la información.

Además de esto, el grupo de trabajo decidió utilizar un formato especial para el envío de mensajes para tener un marco de referencia único para la comunicación entre microservicios, lo que serviría para evitar problemas de configuración desde el back al frontend;

y claramente, también entre los diferentes componentes implementados por Archisofttech.

Las características más específicas, como lo son las tablas utilizadas en las bases de datos y la configuración para la instalación y correcto uso del proyecto se puede encontrar en el Anexo [1. Repositorio Everdose](#)

## 9. Anexos

- [Anexo 1. Repositorio proyecto Everdose - Arquisofttech](#)