

# Servicio Nacional de aprendizaje

## ANÁLISIS Y DESARROLLO DE SOFTWARE

Diagrama de despliegue para caso de estudio y proyecto de software. GA4-220501095-AA3-EV03

Autores:

Frank Sebastian Villa Lobo

Jhojam Camilo Arteaga

Jorge Totena

Julio 2023

# Introducción

En el actual mundo digital, el comercio electrónico ha experimentado un crecimiento exponencial, abriendo nuevas oportunidades para diversos sectores. Del mismo modo, el ámbito veterinario no es la excepción, ya que cada vez más clínicas y hospitales veterinarios buscan expandir su alcance y mejorar la experiencia de sus clientes a través de diferentes plataformas de compras o redes sociales. En este sentido, el desarrollo de un proyecto de software de comercio electrónico específico para un negocio de este sector se ha convertido en una necesidad imprescindible para mantenerse competitivo y proporcionar un servicio de calidad.

Este trabajo tiene como objetivo presentar el diseño e implementación de un diagrama de despliegue para un proyecto de desarrollo de software de e-commerce, destinado a una clínica veterinaria. Este tipo de diagramas son herramientas de suma importancia, ya que permiten visualizar la estructura y la disposición física de los componentes de un sistema con su respectivo tiempo de ejecución. Su importancia está relacionada a su capacidad para representar cómo los diversos elementos del software interactúan entre sí y con el hardware subyacente, facilitando la identificación de posibles cuellos de botella, vulnerabilidades y mejoras de rendimiento. Por otra parte, proporcionan una visión panorámica de la infraestructura necesaria para la puesta en marcha del proyecto, los diagramas de despliegue se convierten en una herramienta esencial para la planificación, la asignación de recursos y la toma de decisiones informadas.

En cuanto a los requisitos funcionales, el software de e-commerce para veterinarias debe ofrecer una plataforma intuitiva y amigable que permita a los clientes realizar compras de productos y servicios relacionados con el cuidado y bienestar de sus mascotas. Del mismo modo, los requisitos no funcionales, el sistema debe ser altamente seguro y confiable, garantizando la privacidad de los datos personales y financieros de los clientes.

En resumen, este trabajo se enfoca en la importancia de los diagramas de despliegue en el desarrollo de software de e-commerce para una veterinaria. Además, se detallarán los requisitos funcionales y no funcionales clave que guiarán la implementación de la plataforma, buscando brindar una solución completa y eficiente que contribuya al éxito y la expansión del negocio veterinario en el mundo digital.

## Requisitos funcionales y no funcionales

Los requisitos funcionales y no funcionales son elementos fundamentales en los proyectos de software, ya que definen tanto las acciones y funcionalidades específicas que el sistema debe cumplir para adaptarse a las necesidades planteadas por el cliente durante el proceso de levantamiento de información.

Se hace un desglose de las restricciones y características que determinan la calidad y rendimiento de la solución tecnológica que se proponga. Los requisitos funcionales aseguran que el software cumpla con las necesidades y expectativas de los usuarios, brindando una experiencia satisfactoria y eficiente. Por otro lado, los requisitos no funcionales garantizan que el sistema sea seguro, escalable, rápido y confiable, lo que contribuye a la estabilidad y el éxito a largo plazo del software. La consideración y equilibrio entre ambos tipos de requisitos resulta crucial para el desarrollo de un software completo y exitoso que cumpla con los objetivos y requerimientos del proyecto.

Para este proyecto en específico se plantearon los diferentes requisitos funcionales y no funcionales, correspondientes al proyecto de desarrollo de software asignado y según las necesidades que se plantearon de un posible cliente.

### Requisitos funcionales

A continuación, se plantean los requisitos funcionales de la plataforma de compras de artículos, accesorios y comida para mascotas que permitirá a la empresa para la cual se hace este servicio alcanzar nuevos paradigmas en términos de ventas, de alcance a nuevos clientes y analítica de datos.

RF-0001	Gestión de productos y carrito de compras
Propósito	Permitir a los usuarios explorar y seleccionar productos de manera fácil y eficiente desde un catálogo completo. Además, busca facilitar el proceso de compra proporcionando un carrito de compras virtual donde los usuarios pueden administrar y revisar sus selecciones antes de finalizar el pedido
Entrada	<ul style="list-style-type: none"> <li>• Selección del usuario en el catálogo de productos.</li> <li>• Cantidad de productos a agregar al carrito de compras.</li> <li>• Detalles de envío y método de pago seleccionado durante el proceso de checkout.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Navegación en el catálogo:</b> El usuario accede al catálogo de productos desde la página principal del sitio web de ecommerce. Allí, puede explorar diferentes categorías de productos y ver los detalles descriptivos de cada artículo, como nombre, imagen, precio, descripción, calificaciones y reseñas de clientes anteriores.</li> <li>• <b>Agregar al carrito de compras:</b> Una vez que el usuario ha encontrado un producto deseado, puede agregarlo al carrito de compras haciendo clic en un botón o icono correspondiente. El producto seleccionado se guarda en el carrito virtual.</li> <li>• <b>Administración del carrito:</b> El usuario puede acceder al carrito de compras en cualquier momento para revisar los productos seleccionados. Dentro del carrito, puede modificar la cantidad de productos o eliminar artículos que ya no desee comprar.</li> <li>• <b>Proceso de checkout:</b> Cuando el usuario está satisfecho con su selección en el carrito de compras, puede proceder al proceso de checkout. Durante este paso, el usuario ingresa la dirección de envío y selecciona su método de pago preferido (tarjeta de crédito, PayPal, etc.).</li> <li>• <b>Confirmación de compra:</b> Antes de finalizar la compra, el usuario tiene la oportunidad de revisar el pedido completo, incluida la dirección de envío, los detalles del producto y el método de pago. Una vez que todo esté en orden, el usuario confirma la compra.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Confirmación del producto agregado al carrito de compras.</li> <li>• Confirmación de la modificación o eliminación de un producto en el carrito.</li> <li>• Confirmación del pedido completo antes de la compra.</li> <li>• Confirmación final de la compra realizada.</li> </ul>

RF-0002	Notificación de transacción
Propósito	Mantener a los clientes informados y actualizados sobre las transacciones importantes que se realizan en el sitio web de ecommerce. Al generar automáticamente correos electrónicos de notificación, se brinda a los clientes una confirmación detallada de cada etapa crucial del proceso de compra, lo que mejora la confianza en la empresa y la satisfacción del cliente.
Entrada	<ul style="list-style-type: none"> <li>• Información de pedido del cliente, incluidos los detalles de los artículos seleccionados y la dirección de envío.</li> <li>• Información de seguimiento de paquetes para el despacho de un pedido.</li> <li>• Información de facturación y detalles del pago del cliente.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Realización de un pedido:</b> Cuando un cliente realiza un pedido en el sitio web de ecommerce, el sistema genera automáticamente un correo electrónico de notificación. Este correo electrónico contiene los detalles relevantes del pedido, como los artículos seleccionados, la cantidad, el precio, la dirección de envío y la fecha estimada de entrega.</li> <li>• <b>Despacho de un pedido:</b> Cuando el pedido ha sido empacado y despachado para su envío, el sistema envía automáticamente un correo electrónico de notificación al cliente. Este correo electrónico incluye la información de seguimiento del paquete y la confirmación de que el pedido ha sido enviado.</li> <li>• <b>Emisión de una factura:</b> Una vez que el pedido ha sido entregado con éxito al cliente, el sistema genera automáticamente una factura correspondiente a la compra realizada. El cliente recibe un correo electrónico de notificación con la factura adjunta y los detalles de la transacción.</li> <li>• <b>Registro de un pago:</b> Cuando el cliente realiza el pago de la factura, ya sea mediante tarjeta de crédito, transferencia bancaria u otro método de pago, el sistema envía automáticamente un correo electrónico de notificación al cliente. Este correo electrónico sirve como confirmación de que el pago ha sido recibido y procesado.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Correo electrónico de notificación para confirmar el pedido realizado por el cliente.</li> <li>• Correo electrónico de notificación para informar al cliente sobre el despacho y envío del pedido.</li> <li>• Correo electrónico de notificación con la factura adjunta para confirmar la emisión de la factura.</li> <li>• Correo electrónico de notificación para confirmar el registro y procesamiento del pago realizado por el cliente.</li> </ul>

<b>RF-0003</b>	<b>Compatibilidad del sistema operativo</b>
Propósito	Asegurar que el software pueda funcionar de manera óptima y sin problemas tanto en sistemas operativos Windows como Android. Al garantizar la compatibilidad, se busca brindar a los usuarios una experiencia consistente y de alta calidad, independientemente del sistema operativo que estén utilizando. Esto mejora la satisfacción del usuario y aumenta la adopción del software en diferentes plataformas.
Entrada	<ul style="list-style-type: none"> <li>• Especificaciones y requisitos del diseño del software.</li> <li>• Plataformas y versiones específicas de los sistemas operativos Windows y Android.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Diseño y desarrollo:</b> Durante la etapa de diseño y desarrollo del software, los desarrolladores se aseguran de tener en cuenta las características y limitaciones específicas de los sistemas operativos Windows y Android. Se utilizan diferentes herramientas y prácticas adecuadas para garantizar la compatibilidad.</li> <li>• <b>Pruebas exhaustivas:</b> Después de desarrollar el software, se llevan a cabo pruebas exhaustivas en ambos sistemas operativos. Esto incluye pruebas funcionales y de rendimiento para identificar cualquier error o problema que pueda surgir en una plataforma específica.</li> <li>• <b>Optimización:</b> Si es necesario, se realizan ajustes y optimizaciones para garantizar un rendimiento óptimo en cada sistema operativo. Se busca maximizar la eficiencia y minimizar el consumo de recursos para ofrecer una experiencia fluida al usuario.</li> <li>• <b>Liberación y actualización:</b> El software compatible con ambos sistemas operativos se lanza al mercado o a los usuarios. Además, se monitorea constantemente el rendimiento y la retroalimentación de los usuarios para abordar cualquier problema potencial y proporcionar actualizaciones periódicas para mejorar la compatibilidad.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Software diseñado y desarrollado para funcionar de manera óptima en sistemas operativos Windows y Android.</li> <li>• Pruebas exhaustivas y correcciones de errores para garantizar la compatibilidad.</li> <li>• Rendimiento y experiencia del usuario mejorados en ambas plataformas.</li> <li>• Actualizaciones periódicas para mantener la compatibilidad y abordar problemas potenciales.</li> </ul>

RF-0004	Gestión de usuarios
Propósito	Proporcionar una gestión segura y efectiva de usuarios para el software. Permite a los usuarios crear cuentas únicas y seguras mediante la verificación de credenciales en la base de datos. Además, el sistema garantiza un acceso controlado y protege la seguridad de las cuentas, bloqueando a usuarios no autorizados o aquellos que ingresen credenciales incorrectas
Entrada	<ul style="list-style-type: none"> <li>• Nombre de usuario y contraseña para la creación de una cuenta.</li> <li>• Nombre de usuario y contraseña para el inicio de sesión.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Creación de cuenta:</b> Los usuarios pueden acceder a la opción de crear una cuenta única proporcionando un nombre de usuario y una contraseña. El sistema verifica la disponibilidad del nombre de usuario y asegura que las contraseñas sean seguras antes de crear la cuenta en la base de datos.</li> <li>• <b>Verificación de credenciales:</b> Cuando un usuario intenta iniciar sesión, el sistema verifica las credenciales ingresadas (nombre de usuario y contraseña) con las almacenadas en la base de datos. Se asegura de que las credenciales sean válidas y correspondan a una cuenta activa y autorizada.</li> <li>• <b>Control de acceso:</b> Una vez que las credenciales son verificadas con éxito, el sistema otorga acceso a la cuenta del usuario, lo que les permite navegar y utilizar las funcionalidades específicas del software.</li> <li>• <b>Bloqueo de usuarios no autorizados:</b> Si un usuario intenta acceder a una cuenta con credenciales incorrectas o inactivas repetidamente, el sistema bloquea temporalmente el acceso para proteger la seguridad de la cuenta. Se puede requerir una verificación adicional o el restablecimiento de la contraseña para desbloquear la cuenta.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Creación exitosa de la cuenta única en la base de datos.</li> <li>• Verificación exitosa de las credenciales para permitir el acceso a la cuenta.</li> <li>• Bloqueo temporal de usuarios no autorizados o con credenciales incorrectas para proteger la seguridad de la cuenta.</li> <li>• Acceso controlado y seguro a las funcionalidades específicas del software para usuarios autorizados.</li> </ul>

## Requisitos no funcionales

A continuación, se plantean los requisitos no funcionales de la plataforma de compras que se desea desarrollar. Esto permitirá moldear la estructura del software a la medida de las necesidades del cliente y de sus futuros compradores.

RNF-0001	Rendimiento
Propósito	Garantizar que el sistema ofrezca un alto rendimiento y eficiencia en términos de tiempo de respuesta y consumo de recursos. Busca proporcionar una experiencia de usuario fluida y evitar retrasos significativos en la interacción con el software.
Entrada	<ul style="list-style-type: none"><li>• Solicitudes de los usuarios para acceder a diferentes funcionalidades del software.</li></ul>
Proceso	<ul style="list-style-type: none"><li>• <b>Tiempo de respuesta rápido:</b> El sistema está diseñado y optimizado para procesar las solicitudes de los usuarios de manera eficiente, asegurando que las respuestas se entreguen en menos de 5 segundos.</li><li>• <b>Eficiencia en el consumo de recursos:</b> Eficiencia en el consumo de recursos: Se han implementado prácticas de programación y optimización para minimizar el consumo de recursos del sistema. El software opera dentro de límites específicos, utilizando menos de 400 MB de memoria RAM y manteniendo el tamaño del programa instalado por debajo de 600 MB en el disco duro.</li></ul>
Respuesta	<ul style="list-style-type: none"><li>• Tiempo de respuesta rápido, con las solicitudes de los usuarios procesadas en menos de 5 segundos.</li><li>• Consumo eficiente de recursos, no superando los límites establecidos de 400 MB de memoria RAM y 600 MB en el disco duro.</li></ul>



<b>RNF-0002</b>	<b>Seguridad</b>
Propósito	Garantizar la máxima seguridad de los datos del cliente. Se busca proteger la información almacenada en la base de datos mediante restricciones de acceso, encriptación de datos sensibles y una gestión cuidadosa de los permisos para evitar accesos no autorizados.
Entrada	<ul style="list-style-type: none"> <li>• Acceso a la base de datos de clientes y datos sensibles almacenados.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Restricción de acceso a la base de datos:</b> Solo el administrador del sistema tiene acceso a la base de datos de clientes. Se implementan mecanismos de autenticación y autorización para asegurar que solo el administrador autorizado pueda acceder y administrar la base de datos.</li> <li>• <b>Encriptación de datos sensibles:</b> Los datos sensibles almacenados en la base de datos, como contraseñas o información personal, se encriptan para protegerlos contra posibles amenazas de seguridad. Esto garantiza que incluso si hay un acceso no autorizado a la base de datos, los datos permanecen ilegibles y seguros.</li> <li>• <b>Gestión cuidadosa de permisos de acceso:</b> La autoridad para cambiar los permisos de acceso al sistema se restringe al administrador de acceso a datos. Se implementan controles estrictos para evitar cambios no autorizados en los permisos, lo que reduce el riesgo de accesos no deseados.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Acceso restringido a la base de datos solo para el administrador del sistema.</li> <li>• Datos sensibles encriptados y seguros en la base de datos.</li> <li>• Gestión cuidadosa de permisos de acceso para evitar accesos no autorizados.</li> </ul>

<b>RNF-0003</b>	<b>Fiabilidad</b>
Propósito	Garantizar la fiabilidad y disponibilidad del sistema. Se busca que el sistema esté disponible el 95,99% del tiempo para que los usuarios puedan acceder a él de manera constante. Además, se busca mantener una probabilidad de fallo muy baja, no superior al 2%, para asegurar que los usuarios confíen en la estabilidad y calidad del servicio que ofrece el sistema, tanto para la interacción del usuario como para el procesamiento preciso y seguro de las transacciones
Entrada	<ul style="list-style-type: none"> <li>• Interacción de usuarios con el sistema de comercio electrónico.</li> <li>• Transacciones procesadas a través del sistema.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Disponibilidad del sistema:</b> Se implementan medidas y prácticas para garantizar que el sistema esté disponible el 95,99% del tiempo. Esto incluye redundancia en servidores, monitoreo pro activo, mantenimiento programado y rápida recuperación en caso de fallo.</li> <li>• <b>Minimización de probabilidad de fallo:</b> Se realizan pruebas exhaustivas y análisis de calidad del software para mantener la probabilidad de fallo del sistema por debajo del 2%. Se busca identificar y corregir posibles errores y vulnerabilidades para mejorar la confiabilidad del sistema.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Disponibilidad del sistema garantizada en al menos el 95,99% del tiempo.</li> <li>• Probabilidad de fallo del sistema no superior al 2%, lo que asegura la confianza y calidad del servicio para los usuarios.</li> </ul>

<b>RNF-0004</b>	<b>Disponibilidad</b>
Propósito	Garantizar la disponibilidad y correcto funcionamiento del sistema al realizar una compra. Se busca verificar que el usuario tenga suficientes fondos para completar la transacción y, en caso de insuficiencia de fondos, detener la transacción y mostrar mensajes de error claros e informativos para guiar al usuario sobre cómo resolver el problema
Entrada	<ul style="list-style-type: none"> <li>• Transacción de compra realizada por el usuario.</li> <li>• Saldo de fondos en la cuenta o método de pago seleccionado por el usuario.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Verificación de fondos:</b> Cuando un usuario realiza una compra, el sistema realiza una verificación para asegurarse de que el usuario tenga suficientes fondos en su cuenta o método de pago seleccionado.</li> <li>• <b>Detención de la transacción:</b> Si los fondos son insuficientes, el sistema detiene la transacción y muestra un mensaje de error al usuario, informando que no se puede completar la compra debido a la falta de fondos suficientes.</li> <li>• <b>Mensajes de error claros e informativos:</b> Todos los mensajes de error generados por el sistema son diseñados para ser claros y comprensibles para los usuarios. Proporcionan una explicación del problema y cómo pueden resolverlo, como agregar más fondos a su cuenta o seleccionar un método de pago válido.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Verificación de fondos y confirmación de suficiencia para completar la transacción.</li> <li>• Detención de la transacción y muestra de mensaje de error en caso de fondos insuficientes.</li> <li>• Mensajes de error claros e informativos para guiar al usuario sobre cómo resolver el problema.</li> </ul>

<b>RNF-0005</b>	<b>Mantenibilidad</b>
Propósito	Garantizar que el software sea fácilmente mantenible y actualizable por el equipo de desarrollo. Se busca facilitar la corrección de errores y la implementación de mejoras al utilizar código limpio, bien comentado, una arquitectura modular y documentación detallada del sistema
Entrada	<ul style="list-style-type: none"> <li>• Código fuente del software con comentarios y documentación detallada.</li> <li>• Componentes y módulos del sistema identificados.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Código limpio y bien comentado:</b> El equipo de desarrollo se asegura de escribir código limpio y estructurado, con comentarios claros que expliquen la funcionalidad de cada sección para facilitar la comprensión. Código limpio y bien comentado: El equipo de desarrollo se asegura de escribir código limpio y estructurado, con comentarios claros que expliquen la funcionalidad de cada sección para facilitar la comprensión.</li> <li>• <b>Arquitectura modular:</b> El software se divide en módulos independientes y cohesivos, lo que facilita la identificación y corrección de errores y permite actualizar componentes específicos sin afectar otras partes del sistema.</li> <li>• <b>Documentación detallada:</b> Se crea una documentación exhaustiva del sistema, que incluye descripciones de la arquitectura, funcionalidades y flujos de trabajo. Esto permite a cualquier miembro del equipo de desarrollo comprender rápidamente cómo funciona el software.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Facilidad para el equipo de desarrollo para realizar actualizaciones y correcciones de errores.</li> <li>• Mayor eficiencia en el mantenimiento del software debido a la comprensión rápida de la estructura y funcionalidades del sistema.</li> <li>• Reducción del tiempo y esfuerzo necesarios para implementar cambios y mejoras en el software.</li> </ul>

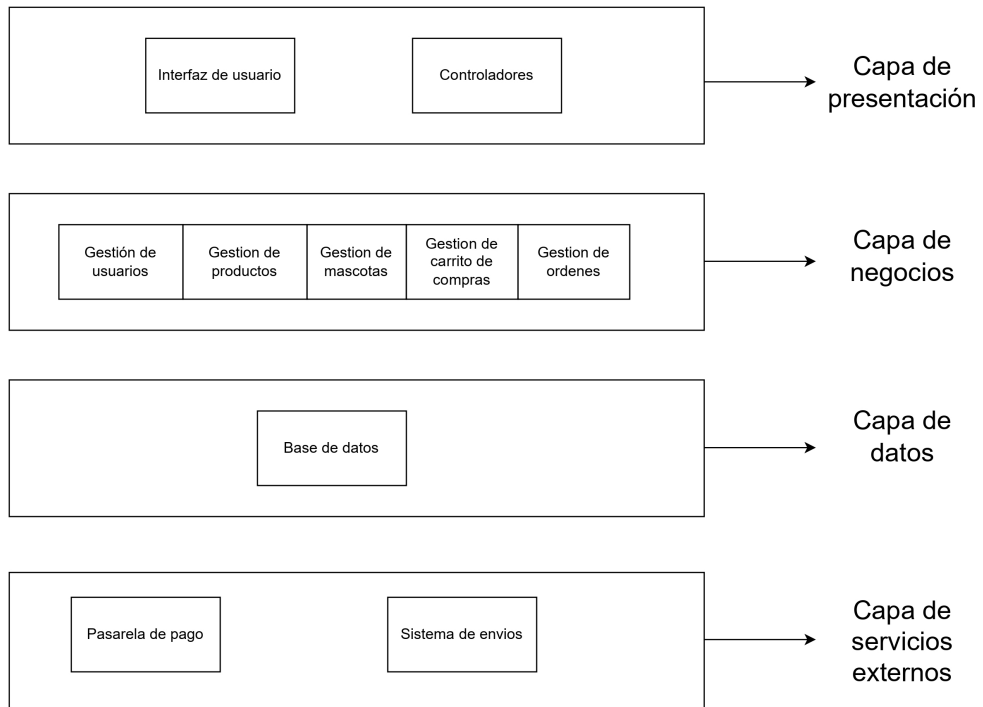
<b>RNF-0006</b>	<b>Portabilidad</b>
Propósito	asegurar que el software sea portátil y accesible en una variedad de dispositivos. Se busca proporcionar una experiencia de usuario consistente y atractiva a través de diferentes tamaños y orientaciones de pantalla, desde computadoras de escritorio y laptops hasta tablets y smartphones. Las interfaces gráficas deben ser intuitivas, atractivas y diseñadas siguiendo principios de experiencia de usuario para facilitar la navegación y la compra
Entrada	<ul style="list-style-type: none"> <li>• Requisitos de diseño responsive y adaptabilidad a diferentes dispositivos.</li> <li>• Principios de diseño de experiencia de usuario para crear interfaces intuitivas y atractivas.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>• <b>Diseño "Responsive":</b> El software se diseña y desarrolla con un diseño "responsive", lo que permite que se adapte automáticamente a diferentes tamaños de pantalla y dispositivos, brindando una experiencia óptima al usuario en todas las plataformas.</li> <li>• <b>Adaptabilidad a diferentes dispositivos:</b> Se implementan ajustes en la interfaz gráfica y la disposición de elementos para garantizar que el software sea funcional y se vea bien en computadoras de escritorio, laptops, tablets y smartphones.</li> <li>• <b>Experiencia de usuario intuitiva:</b> Se aplican principios de diseño de experiencia de usuario para crear interfaces gráficas intuitivas y atractivas. Se busca guiar a los usuarios de manera efectiva a través del proceso de navegación y compra.</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Software portátil y accesible en una variedad de dispositivos.</li> <li>• Experiencia de usuario consistente y atractiva en diferentes plataformas.</li> <li>• Interfaces gráficas intuitivas que facilitan la navegación y la compra para los usuarios.</li> </ul>

## Arquitectura por capas

La arquitectura de software por capas es un enfoque de diseño en el que un sistema de software se organiza en diferentes capas o niveles funcionales, en donde cada capa se encarga de una tarea específica dentro del conjunto de tareas que conforman un software. Cada capa se comunica con las capas adyacentes a través de interfaces bien definidas, lo que permite una separación clara de responsabilidades y facilita el desarrollo, mantenimiento y escalabilidad del software. La importancia de la arquitectura de software por capas radica en varios aspectos. En primer lugar, proporciona una estructura modular que facilita la reutilización de componentes. Cada capa puede ser desarrollada de manera independiente, lo que acelera el desarrollo, al permitir trabajar con diferentes equipos de trabajo en cada uno de los frentes que conforman el proyecto y reduce los errores y redundancias al no tener que implementar funcionalidades repetitivas.

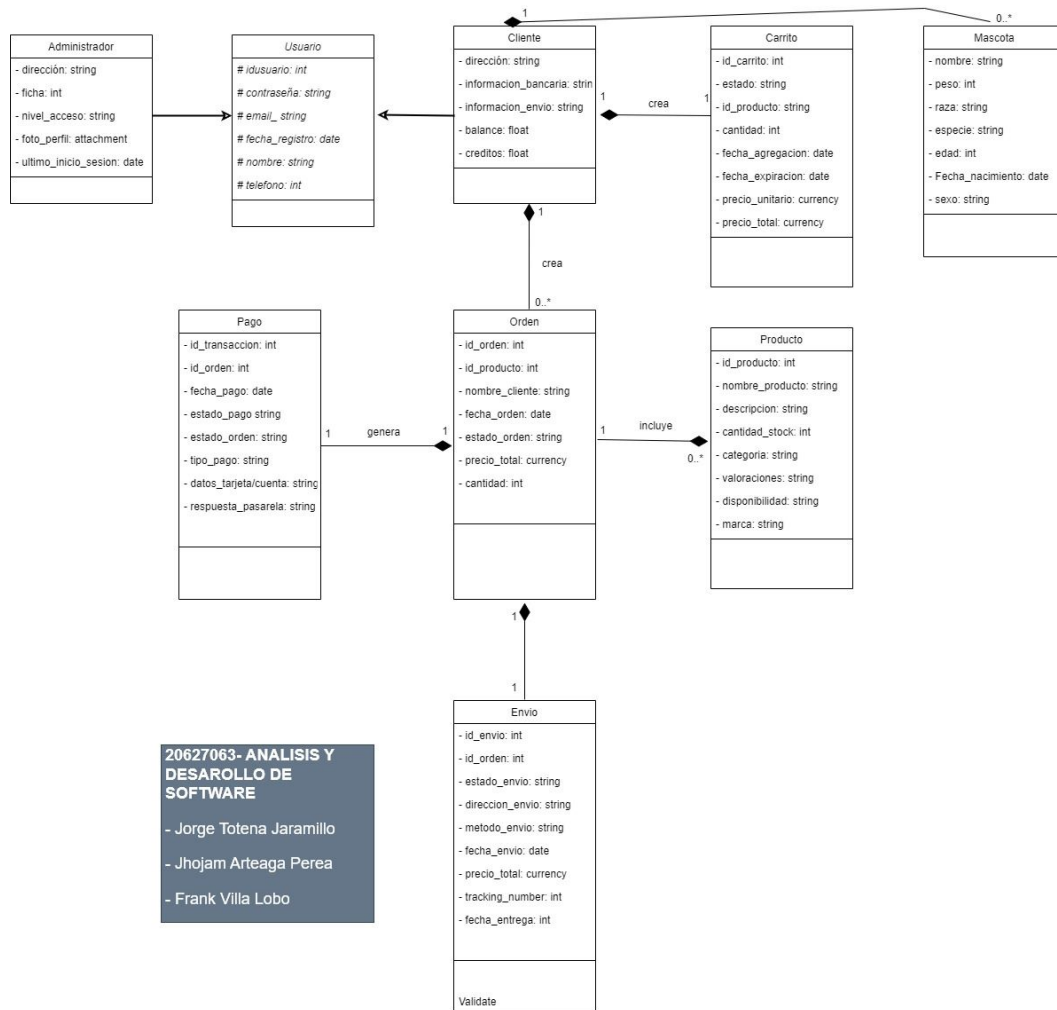
Del mismo modo, la arquitectura por capas mejora la claridad y comprensión del sistema. Al separar las funcionalidades en capas distintas, se facilita la comunicación entre los miembros del equipo de desarrollo y asegura una comprensión común del sistema. Otra ventaja es la facilidad de mantenimiento, en caso de ser necesario, realizar cambios o mejoras en el software, sería mucho más fácil, ya que la arquitectura por capas permite identificar rápidamente la capa afectada y realizar las modificaciones sin afectar el resto del sistema.

En resumen, la arquitectura de software por capas es una herramienta poderosa para crear sistemas eficientes, flexibles y fáciles de mantener. Facilita la reutilización de componentes, mejora la comunicación y comprensión del equipo de desarrollo, y permite un desarrollo y mantenimiento más sencillo y escalable del software. A continuación, presentamos la arquitectura por capas planteada para este proyecto.



## Diagrama de clases

El diagrama de clases es una representación gráfica en el desarrollo de software que muestra las clases y sus relaciones en un sistema orientado a objetos. En este diagrama, las clases representan objetos o entidades del sistema, mientras que las relaciones establecen cómo se relacionan entre sí. Es una herramienta fundamental para visualizar y entender la estructura y las interacciones del software, facilitando la comunicación entre los miembros del equipo de desarrollo y permitiendo una planificación y diseño más efectivos del sistema. El diagrama de clases planteado para este proyecto se muestra a continuación.





## Diagrama de despliegue

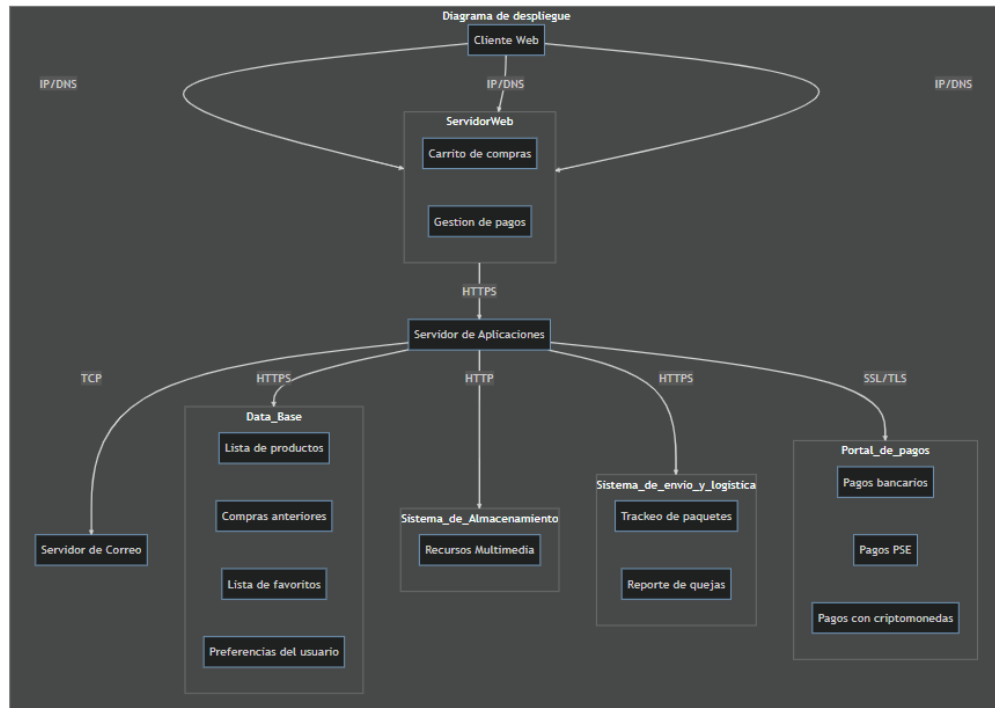
El diagrama de despliegue es una representación gráfica que muestra cómo se organizan y conectan físicamente los componentes de un sistema de software en un entorno de ejecución. En otras palabras, es una visualización de la infraestructura y la arquitectura física del software, incluyendo servidores, dispositivos, redes y otros recursos necesarios para que el sistema funcione correctamente.

La importancia del diagrama de despliegue radica en que proporciona una visión clara y completa de la distribución física del software, lo que permite a los desarrolladores y arquitectos comprender cómo interactúan los componentes y cómo se comunican entre sí. Al comprender esta estructura física, los equipos pueden identificar posibles cuellos de botella, puntos de fallo y problemas de rendimiento antes de implementar el sistema en un entorno de producción.

En resumen, el diagrama de despliegue es una herramienta esencial en proyectos de desarrollo de software, ya que proporciona una representación gráfica de la infraestructura física del sistema, permitiendo una mejor comprensión, planificación y optimización del software antes de su implementación. A continuación, se presenta el diagrama de despliegue utilizando como herramienta TIC el lenguaje MERMAID, especializado para la graficación de diagramas.

```
diagramaDespliegue.md 2
ama de despliegue para caso de estudio y proyecto de software. GA4-220501095-AA3-EV03 > diagramaDespliegue.md > # Diagrama de despliegue
1  # Diagrama de despliegue
2  mermaid
3  %%{init: {'theme': 'dark'}}%%
4  %%{sequence: {'wrap': false}}%%
5  graph LR
6      subgraph Diagrama_de_despliegue
7          subgraph ServidorWeb
8              Carrito[Carrito de compras]
9              Ppagos[Gestion de pagos]
10          end
11          C[Cliente Web]
12          SA[Servidor de Aplicaciones]
13          subgraph Data_Base
14              productos[Lista de productos]
15              compras[Compras anteriores]
16              favoritos[Lista de favoritos]
17              informacion[Preferencias del usuario]
18          end
19          SP[Servidor de Correo]
20          subgraph Sistema_de_envio_y_logistica
21              tracking[Trackeo de paquetes]
22              quejas[Reporte de quejas]
23          end
24          subgraph Sistema_de_Almacenamiento
25              resources[Recursos Multimedia]
26          end
27          subgraph Portal_de_pagos
28              pagosB[Pagos bancarios]
29              pagosPSE[Pagos PSE]
30              pagosCripto[Pagos con criptomonedas]
31          end
32      end
33
34      C -->|IP/DNS| ServidorWeb
35      C -->|IP/DNS| ServidorWeb
36      C -->|IP/DNS| ServidorWeb
37      ServidorWeb -->|HTTPS| SA
38      SA -->|HTTPS| Data_Base
39      SA -->|TCP| SP
40      SA -->|HTTP| Sistema_de_Almacenamiento
41      SA -->|HTTPS| Sistema_de_envio_y_logistica
42      SA -->|SSL/TLS| Portal_de_pagos
43  
```

## Diagrama de despliegue



## Conclusiones

El proyecto de ecommerce para una veterinaria ha sido cuidadosamente planificado y diseñado, considerando tanto los requisitos funcionales como los no funcionales para asegurar un sistema completo y eficiente. Los requisitos funcionales definen las funcionalidades clave del software, garantizando una experiencia de compra fluida y notificaciones efectivas para los clientes. Por otro lado, los requisitos no funcionales como el rendimiento, disponibilidad, portabilidad y seguridad son fundamentales para garantizar un sistema rápido, confiable, adaptable y seguro.

El diagrama de despliegue muestra cómo el software se distribuye en diferentes entornos, asegurando una correcta asignación de recursos. La arquitectura por capas ha sido una elección acertada, permitiendo una fácil reutilización de componentes y un desarrollo, mantenimiento y escalabilidad simplificados. En general, el proyecto refleja un enfoque integral y cuidadoso en el diseño y desarrollo del software, preparándolo para adaptarse y crecer exitosamente en el mercado de ecommerce.