Plataforma E-Commerce Multi-Tenant

Buyee

Versión 1.0

Realizado por:

Daniel Tamara Rivera 91509

David Alejandro Medina Ruiz 91769

Samuel Santiago Falla Alfaro 90180

**Fecha 28/05/2025**

**Tabla de Contenido**

[1. Visión General del Proyecto 4](#_Toc198844236)

[1.1. Contexto 4](#_Toc198844237)

[1.2. Interesados Claves 4](#_Toc198844238)

[2. Listado de Requerimientos y Atributos 6](#_Toc198844239)

[2.1. Requerimientos Funcionales 6](#_Toc198844240)

[2.2. Requerimientos No Funcionales 10](#_Toc198844241)

[2.3. Atributos de Calidad 13](#_Toc198844242)

[2.4. Ponderación en Método de Selección por Pesos 13](#_Toc198844243)

[2.5. Drivers Prioritarios 14](#_Toc198844244)

[3. Tácticas de Arquitectura y Estilos Arquitectónicos 15](#_Toc198844245)

[3.1. Tácticas 15](#_Toc198844246)

[3.1.1. Fiabilidad 15](#_Toc198844247)

[3.1.2. Rendimiento 16](#_Toc198844248)

[3.1.3. Seguridad 16](#_Toc198844249)

[3.1.4. Flexibilidad 17](#_Toc198844250)

[3.1.5. Usabilidad 17](#_Toc198844251)

[3.2. Patrones Arquitectónicos 18](#_Toc198844252)

[3.3. Estrategia 19](#_Toc198844253)

[4. Modelo de Arquitectura 4 + 1 21](#_Toc198844254)

[4.1. Escenarios Identificados 21](#_Toc198844255)

[4.1.1. Casos de Uso de Comprador 21](#_Toc198844256)

[4.1.2. Casos de Uso de Proveedor 22](#_Toc198844257)

[4.1.3. Casos de Uso de Administrador 23](#_Toc198844258)

[4.2. Vista Lógica 23](#_Toc198844259)

[4.3. Vista de Implementación 25](#_Toc198844260)

[4.4. Vista de Procesos 27](#_Toc198844261)

[4.5. Vista Física 33](#_Toc198844262)

# Visión General del Proyecto

## Contexto

El proyecto consiste en desarrollar una plataforma SaaS (Software as a Service) que será administrada por una empresa a fin de ofrecer una herramienta que permita a pequeños y medianos negocios crear y gestionar sus propios portales de comercio electrónico de forma sencilla y con mínima administración técnica, accediendo a todas las funcionalidades usuales de un paquete de comercio electrónico. Esto incluiría intercambio de monedas e internacionalización, manejo de usuarios y confiabilidad.

El objetivo es ofrecer una solución robusta, escalable y fácil de usar que democratice el acceso al comercio electrónico para negocios que no cuentan con grandes recursos técnicos o de inversión inicial para desarrollar sus propias soluciones y conectar con los clientes que tengan. Al ser puntos de comercio internacional por su naturaleza virtual, es posible contar con millones de estos clientes por parte de los negocios y, asimismo, miles de negocios que a través de una configuración simplificada podrán poner a funcionar sus plataformas de venta en diferentes idiomas y divisas. Cabe resaltar que de acuerdo al modelo de negocio empleado la herramienta es agnóstica al proceso de logística efectuado por cada proveedor involucrado, solamente se enfoca en brindar funcionalidades base para suplir las operaciones básicas de cada entidad comercial.

## Interesados Claves

**Proveedores**

Los proveedores son comprendidos como los responsables que harán uso de la herramienta*, g*eneralmente son pequeños o medianos negocios sin gran capacidad técnica, por lo que valoran la simplicidad, flexibilidad y autonomía que ofrece la plataforma para crear los sitios donde alojarán y gestionarán los productos, información de utilidad comercial de sus usuarios y la presentación misma de sus servicios de cara al público.

**Compradores**

Los compradores son todos aquellos que son finalmente quienes consultarán y harán uso del sitio establecido por quien(es) administra(n) el negocio. Este grupo ejecuta sus acciones comunes de compras acorde a lo que haya definido la tienda, incluye tanto visitantes como usuarios registrados que interactúan con los productos, completan compras, dejan reseñas y acceden a su historial de pedidos, entre otros aspectos básicos de comercio. Hacen parte de los interesados ya que son los que definen el objetivo de estos negocios, es decir, sin los clientes de los denominados proveedores, no hay valor para el uso de esta herramienta parte de ellos.

**Administrador de la herramienta**

Comprende la representación que hará uso final del proyecto. Sin una especificación de las cualificaciones que describen quien(es) hará(n) uso de la administración de la herramienta, será un rol el cual hará parte de moderar la solución que ofrece la empresa que desea construir este proyecto de software.

**Equipo de TI**

El equipo de TI es el responsable del desarrollo, mantenimiento y evolución de la plataforma. Este grupo incluye desarrolladores, arquitectos de software, diseñadores UX/UI, DevOps e ingenieros de calidad. Sus funciones incluyen: implementar nuevas funcionalidades, garantizar la disponibilidad del sistema, gestionar bases de datos, monitorear el rendimiento y resolver incidencias técnicas.

**Equipo de Seguridad**

Este equipo vela por la integridad, confidencialidad y disponibilidad de la plataforma y de los datos almacenados. Sus responsabilidades incluyen la implementación de mecanismos de autenticación y autorización, el cumplimiento de normativas de protección de datos y privacidad (como GDPR o equivalentes), el monitoreo de amenazas, la respuesta ante incidentes y la protección frente a amenazas a la seguridad de la herramienta.

# Listado de Requerimientos y Atributos

## Requerimientos Funcionales

Acorde a lo establecido de las partes interesadas y el enfoque del proyecto de software a desarrollar, se dividen los requerimientos funcionales acorde a los usuarios principales que se identifican como actores del sistema. Estos consisten en los clientes que son quienes manejan el negocio, proveedores; los clientes de ellos, los compradores; quien(es) administrará(n) la herramienta en la empresa, administrador.

**Requerimientos funcionales para compradores.**

* RF001: Registrar cuenta.
* RF002: Iniciar sesión.
* RF003: Recuperar contraseña.
* RF004: Editar perfil.
* RF005: Navegar y buscar productos en el catálogo del negocio.
* RF006: Ver detalles del producto.
* RF007: Añadir/eliminar productos al carrito de compras.
* RF008: Ver y modificar el carrito de compras.
* RF009: Realizar el proceso de checkout.
* RF010: Aplicar cupones de descuento.
* RF011: Seleccionar método de pago.
* RF012: Ver historial de pedidos o compras realizadas.
* RF013: Seleccionar idioma y moneda de preferencia.
* RF014: Solicitar devolución/reembolso de productos.
* RF015: Dejar reseñas y calificaciones de productos.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **DESCRIPCIÓN** | **DETALLES** | **STAKEHOLDERS** |
| RF001 | Registrar cuenta | El sistema debe permitir a los compradores crear una cuenta proporcionando datos personales y credenciales como email, nombre de usuario, país, fecha de nacimiento y contraseña. | Compradores, equipo de TI y seguridad. |
| RF002 | Iniciar sesión | El sistema debe permitir al comprador autenticarse con sus credenciales. | Compradores, equipo de TI y seguridad. |
| RF003 | Recuperar contraseña | El sistema debe permitir al comprador restablecer la contraseña mediante verificación por email. | Compradores, equipo de TI y seguridad. |
| RF004 | Editar perfil | El sistema debe permitir al comprador editar información relacionada a su cuenta como nombre, contraseña, email, dirección de envío y métodos de pago guardados. | Compradores, equipo de TI y seguridad. |
| RF005 | Navegar y buscar productos en el catálogo del negocio | El sistema debe permitir al comprador explorar y buscar productos usando filtros, búsqueda por nombre y categoría. | Compradores, equipo de TI. |
| RF006 | Ver detalles del producto | El sistema debe mostrar la información detallada del producto como nombre, descripción, imágenes, precio, y variantes. | Compradores, proveedores, equipo de TI. |
| RF007 | Añadir/eliminar productos al carrito de compras | El sistema debe permitir al comprador añadir o eliminar productos del carrito de compras. | Compradores, equipo de TI. |
| RF008 | Ver y modificar el carrito de compras | El sistema debe permitir al comprador consultar y editar cantidades del producto antes de proceder al pago. | Compradores, equipo de TI. |
| RF009 | Realizar el proceso de checkout | El comprador puede proporcionar la dirección de envío y datos de facturación para completar la compra. | Compradores, equipo de TI. |
| RF010 | Aplicar cupones de descuento | El sistema debe permitir al comprador acceder a un apartado para digitar el código de promoción proveniente de la tienda o de otras fuentes. | Compradores, proveedores, equipo de TI. |
| RF011 | Seleccionar método de pago | El sistema debe ofrecer múltiples métodos de pago con validación segura de la transacción. | Compradores, proveedores, equipo de TI y seguridad. |
| RF012 | Ver historial de pedidos o compras realizadas | El sistema debe permitir consultar compras anteriores, estado del pedido, fecha y detalles. | Compradores, equipo de TI. |
| RF013 | Seleccionar idioma y moneda de preferencia | El sistema debe permitir personalizar idioma y moneda disponibles en la tienda. | Compradores, proveedores, equipo de TI. |
| RF014 | Solicitar devolución/reembolso de productos | El sistema debe permitir al comprador solicitar el reembolso o devolución del producto comprado. | Compradores, equipo de TI. |
| RF015 | Dejar reseñas y calificaciones de productos | El sistema debe permitir valorar productos comprados con puntuación y/o comentarios. | Compradores, equipo de TI. |

**Requerimientos funcionales para proveedores**

* RF101: Registrarse como proveedor.
* RF102: Crear tienda.
* RF103: Personalizar la apariencia básica de la tienda.
* RF104: Gestionar el catálogo de productos.
* RF105: Gestionar información de los productos.
* RF106: Gestionar el inventario.
* RF107: Crear y gestionar promociones y cupones de descuento.
* RF108: Configurar métodos de pago.
* RF109: Ver y gestionar compras recibidas.
* RF110: Acceder a reportes básicos de ventas y actividad.
* RF111: Configurar idiomas y monedas presentadas en su tienda.
* RF112: Gestionar políticas de la tienda.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **DESCRIPCIÓN** | **DETALLES** | **STAKEHOLDERS** |
| RF101 | Registrar como proveedor | El proveedor puede registrar su cuenta como dueño de tienda, ingresando nombre, correo, contraseña, teléfono, dirección (opcional). | Proveedores, equipo de TI y seguridad. |
| RF102 | Crear tienda | El sistema debe permitir crear tienda a un proveedor ingresando datos comerciales y de contacto. | Proveedores, equipo de TI. |
| RF103 | Personalizar apariencia de su tienda | El proveedor puede modificar tema visual, logo y colores principales. | Proveedores, equipo de TI. |
| RF104 | Gestionar el catálogo de productos | El proveedor puede crear, actualizar, eliminar y consultar productos publicados en la tienda. | Proveedores, equipo de TI. |
| RF105 | Gestionar información de los productos | El sistema debe permitir al proveedor modificar la información de cada producto, indicando su descripción, talla, colores, precio y otras características relacionadas. | Proveedores, equipo de TI y seguridad. |
| RF106 | Gestionar el inventario | Permite definir el stock disponible y recibir alertas por bajo inventario. | Proveedores, equipo de TI. |
| RF107 | Crear y gestionar promociones y cupones de descuento | El sistema debe permitir al proveedor crear descuentos por volumen, cupones y promociones por tiempo limitado a través de un código que puedan registrar en su tienda. | Proveedores, equipo de TI. |
| RF108 | Configurar métodos de pago | El sistema debe permitir al cliente definir las pasarelas de pago integrables para sus ventas. | Proveedores, equipo de TI y seguridad. |
| RF109 | Ver y gestionar compras recibidas | El sistema debe permitir al proveedor ver los pedidos de los compradores y poder actualizar su estado (enviado, entregado, etc.). | Proveedores, equipo de TI. |
| RF110 | Acceder a reportes de ventas y actividad | El sistema debe permitir al proveedor acceder a las estadísticas de ventas como productos más vendidos, zonas de compra, reseñas, calificaciones, etc. | Proveedores, equipo de TI. |
| RF111 | Configurar idiomas y monedas presentadas en su tienda | El sistema debe permitir al proveedor habilitar diferentes idiomas en la tienda y monedas en las que mostrará los productos. | Proveedores, equipo de TI. |
| RF112 | Gestionar políticas de la tienda | El sistema debe permitir al proveedor modificar información legal de la tienda como sus políticas de reembolso o de compra presente en los procesos de checkout. | Proveedores, equipo de TI y seguridad. |

**Requerimientos funcionales para administrador**

* RF201: Gestionar cuentas de proveedores.
* RF202: Monitorear la salud y el rendimiento de la plataforma.
* RF203: Gestionar plantillas visuales para tiendas.
* RF204: Acceder a métricas y analíticas globales.
* RF205: Gestionar la facturación de proveedores.
* RF206: Gestionar las integraciones globales.
* RF207: Gestionar contenido informativo y soporte.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **DESCRIPCIÓN** | **DETALLES** | **STAKEHOLDERS** |
| RF201 | Gestionar cuentas de proveedores | El sistema debe permitir al administrador aprobar cuentas, suspender, eliminar o consultar información de los proveedores. | Administrador, equipo de TI y seguridad. |
| RF202 | Monitorear la salud y el rendimiento de la plataforma | El sistema debe permitir visualizar métricas como incidentes, errores y disponibilidad de los servicios integrados. | Administrador, equipo de TI. |
| RF203 | Gestionar plantillas visuales para tiendas | El sistema debe permitir crear, modificar, activar o desactivar plantillas base disponibles para tienda. | Administrador, equipo de TI. |
| RF204 | Acceder a métricas y analíticas globales | El sistema debe permitir al administrador consultar a estadísticas de uso de tiendas, número de usuarios activos o tendencias generales de la plataforma. | Administrador, equipo de TI. |
| RF205 | Gestionar la facturación de proveedores | El sistema debe permitir al administrador establecer precios respectivos, ver historial de pagos y generar facturas automáticas de proveedores. | Administrador, equipo de TI y seguridad. |
| RF206 | Gestionar las integraciones globales | El sistema debe permitir al administrador mantener las integraciones compartidas para todas las tiendas como pasarelas de pago y soporte de idioma. | Administrador, equipo de TI y seguridad. |
| RF207 | Gestionar contenido informativo y soporte | El sistema debe permitir al administrador publicar y actualizar políticas de seguridad, términos y condiciones. | Administrador, equipo de TI. |

## Requerimientos No Funcionales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **DESCRIPCIÓN** | **DETALLES** | **MÉTRICAS** | **STAKEHOLDERS** |
| NFR001 | Escalabilidad global | El sistema debe soportar despliegue en 3 regiones estratégicas de AWS para atender proveedores y compradores en diferentes zonas geográficas. | 3 regiones AWS activas  (ej: us-east-1, eu-west-1, ap-southeast-1) | Proveedores, compradores, administrador de la herramienta, equipo de TI. |
| NFR002 | Latencia de respuesta en transacción | El sistema debe soportar 1000 transacciones por segundo durante picos de carga, a fin de evitar perdida de paquetes. | 1000 transacciones  /seg  (pico) | Compradores y proveedores. |
| NFR003 | Autenticación segura | Protección de acceso no autorizado mediante OAuth2 y MFA (Multi-Factor Authentication). | OAuth2 + MFA habilitado | Equipo de seguridad, proveedores, compradores, y administrador de la herramienta. |
| NFR004 | Alta disponibilidad | El sistema debe tener una disponibilidad del 99.5% mediante replicación, balanceo de carga y zonas de disponibilidad. | 99.5% uptime  (≈ 2 días de downtime/año). | Compradores y proveedores, administrador de la herramienta, equipo de TI y seguridad. |
| NFR005 | Panel de administración sencillo | El sistema debe tener una interfaz intuitiva con procesos guiados y documentación interactiva, lo cual conlleve a una capacitación técnica menor a 30 minutos en completar tareas básicas. | < 30 minutos para completar tareas básicas | Compradores, proveedores y administrador de la herramienta. |
| NFR006 | Rendimiento de APIs | La respuesta de las APIs debe ser inferior a 200ms para el 95% de las solicitudes en lecturas simples y hasta 800 ms para procesos de pago o actualización de inventario. | < 200ms de respuesta en tareas simples,  < 800 ms en tareas más pesadas | Compradores, proveedores, equipo de TI. |
| NFR007 | Escalabilidad horizontal | El sistema debe tener un escalado automático para manejar más de 10000 de compradores activos sin degradar la latencia aceptable. | < 10000 compradores recurrentes | Proveedores, equipo de TI, administrador de la herramienta. |
| NFR008 | Seguridad OWASP Top 10 | El sistema debe cubrir un 99% en la protección contra ataques comunes (inyección SQL, XSS, CSRF), mediante pruebas automatizadas. | 99% de cobertura de OWASP Top 10, mediante pruebas automatizadas | Equipo de seguridad. |
| NFR009 | Mantenibilidad ágil | El sistema debe adoptar una arquitectura modular y un pipeline CI/CD que permita realizar actualizaciones y correcciones de forma rápida y segura.  Los despliegues deberán ejecutarse mediante estrategias de zero-downtime (como blue/green deployments o rolling updates), garantizando que el tiempo de inactividad por despliegue no supere los 5 segundos bajo condiciones normales de operación | Tiempo máximo de inactividad por despliegue: ≤ 5 segundos | Equipo de TI, equipo de seguridad. |
| NFR010 | Internacionalización | Soporte para 5 idiomas, divisas y regulaciones locales inicialmente (ej: GDPR, PCI-DSS). | Soporte para 5 idiomas  /divisas iniciales | Compradores y proveedores. |
| NFR011 | Hosting "invisible" | Infraestructura completamente autogestionada (AWS Fargate, RDS, Lambda) sin necesidad de intervención del proveedor. | 100% de operaciones automatizadas | Proveedores, administrador de la herramienta, equipo de TI. |

## Atributos de Calidad

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STAKEHOLDERS** | **Flexibilidad** | **Rendimiento** | **Seguridad** | **Fiabilidad** | **Usabilidad** | **Mantenibilidad** | **Adecuación funcional** | **Capacidad de interacción** | **TOTAL** |
| Proveedor | 20 | 25 | 5 | 25 | 10 |  | 10 | 5 | 100 |
| Comprador |  | 30 | 5 | 30 | 30 |  | 5 |  | 100 |
| Administrador de la herramienta | 40 |  | 10 | 20 | 15 |  |  | 15 |  |
| Equipo de TI | 30 | 25 |  | 15 |  | 15 |  | 15 | 100 |
| Equipo de seguridad |  |  | 60 | 25 |  | 15 |  |  | 100 |
|  | 18 | 16 | 16 | 23 | 11 | 6 | 3 | 7 | 100 |

## Ponderación en Método de Selección por Pesos

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ATRIBUTO** | **SUBCATEGORÍA** | **DESCRIPCIÓN** | **MÉTRICA** | **IMPACTO** | **DIFICULTAD** | **% PESO** | **VALOR** |
| **Flexibilidad** | *Escalabilidad* | Escalabilidad global | 3 regiones AWS activas | 2 | 2 | 18 | 63 |
| Escalabilidad horizontal | < 10000 compradores recurrentes | 2 | 1 |
| **Rendimiento** | *Comportamiento temporal* | Rendimiento de APIs | < 200ms de respuesta en tareas simples,  < 800 ms en tareas más pesadas | 2 | 3 | 16 | 72 |
| *Utilización de recursos* | Latencia de respuesta en transacción | 1000 transacciones  /seg (pico) | 2 | 2 |
| **Seguridad** | *Autenticidad* | Autenticación segura | OAuth2 + MFA habilitado | 2 | 2 | 16 | 72 |
| *Resistencia* | Seguridad OWASP Top 10 | 99% de cobertura de OWASP Top 10, mediante pruebas automatizadas | 3 | 2 |
| **Fiabilidad** | *Disponibilidad* | Alta disponibilidad | 99.5% uptime  (≈ 2 días de downtime/año). | 3 | 2 | 23 | 115 |
| **Usabilidad** | *Adaptabilidad* | Panel de administración sencillo | < 30 minutos para completar tareas básicas | 2 | 1 | 11 | 33 |
| **Mantenibilidad** | *Modularidad* | Mantenibilidad ágil | Tiempo máximo de inactividad por despliegue: ≤ 5 segundos | 2 | 2 | 6 | 24 |
| **Adecuación Funcional** | *Completitud funcional* | Internacionalización | Soporte para 5 idiomas/divisas iniciales | 2 | 2 | 3 | 12 |
| **Capacidad de Interacción** | *Asistencia al usuario* | Hosting "invisible" | 100% de operaciones automatizadas | 3 | 1 | 7 | 28 |

## Drivers Prioritarios

|  |  |
| --- | --- |
| **ATRIBUTO** | **VALOR** |
| Fiabilidad | 115 |
| Rendimiento | 72 |
| Seguridad | 72 |
| Flexibilidad | 63 |
| Usabilidad | 33 |
| Mantenibilidad | 24 |
| Capacidad de Interacción | 28 |
| Adecuación Funcional | 12 |

# Tácticas de Arquitectura y Estilos Arquitectónicos

De acuerdo con el método de selección por pesos efectuado en la anterior sección, se logró definir el orden de prioridad en atributos de calidad a enfocarse en el desarrollo e implementación de la herramienta; por ello se indican los siguientes cinco atributos de calidad claves para considerar en las tácticas de arquitectura:

1. Fiabilidad
2. Rendimiento
3. Seguridad
4. Flexibilidad
5. Usabilidad

## Tácticas

### Fiabilidad

* **Manejo de fallos:** A fin de evitar que los fallos de un servicio logren afectar el sistema completo, se aplica el patrón *Circuit Breaker*. Este patrón detecta las fallas recurrentes y aísla temporalmente el componente que genera un problema, previniendo así efectos en cascada. Además, se definen políticas de *retry* para manejar fallos transitorios, y se implementan mecanismos de *fallback* como respuestas por defecto o colas de procesamiento que pueden diferir en el proceso continuo del software, de esta manera es posible operar las funcionalidades críticas del sistema ante alguna contingencia.
* **Persistencia y recuperación de datos:** El sistema implementa una política robusta de protección de datos que incluye *backups* automáticos programados, complementados con pruebas de restauración periódicas para validar su integridad. Además, la funcionalidad de *Point-in-Time Recovery* (PITR) para bases de datos es importante implementarla debido a la restauración precisa de datos que hacer, logra hasta un segundo específico dentro del período de retención configurado, y esto es esencial para corregir errores humanos o restaurar estados consistentes.
* **Monitoreo proactivo:** Para garantizar un sistema altamente confiable y resiliente, se establecerá una estrategia de observabilidad integral basada en el monitoreo proactivo de todos los componentes críticos del sistema. Esto incluye métricas, logs y trazas distribuidas que permitan detectar, diagnosticar y anticipar fallos antes de que afecten al usuario final. Para esto se piensa utilizar *Prometheus* como recolector de métricas y *Grafana* para su visualización.

### Rendimiento

* **Almacenamiento de datos:** El uso de sistemas como Redis o CDNs permite almacenar y servir recursos estáticos de forma eficiente, mejorando el rendimiento, la escalabilidad y la disponibilidad en aplicaciones. Además, posibilita la minimización de llamadas entre servidores y base de datos externas, esto es crucial debido a que mantiene la eficiencia y velocidad en un entorno distribuido.
* **Reducir la sobrecarga computacional:** La demanda de acceso en recursos en línea entrega picos de botellas en el sistema. El balanceo de carga permite distribuir las solicitudes entrantes entre múltiples servidores o nodos, lo que mejora significativamente el rendimiento del sistema. Asegura mayor disponibilidad del servicio y garantiza una experiencia de usuario fluida, incluso en momentos de alta demanda. Distribución de peticiones entre múltiples servidores o nodos para mejorar la disponibilidad y el tiempo de respuesta.
* **Cantidad de peticiones limitadas:** Con ayuda de *Rate Limiting* es posible balancear disponibilidad, seguridad y eficiencia en una herramienta, ya que controla el número de solicitudes que un usuario o servicio puede hacer al sistema en un periodo de tiempo determinado.

### Seguridad

* **Autenticación de usuarios:** Con ayuda de OAuth 2.0, protocolo de autorización permite a las aplicaciones de terceros acceder a recursos protegidos en nombre de un usuario, sin exponer sus credenciales. Este opera mediante el uso de tokens de acceso temporales, que definen el alcance (*scopes*), duración y permisos del acceso otorgado, lo que permite un control fino y seguro sobre los recursos compartidos. Además, es posible añadir una capa extra de seguridad integrando la autenticación multifactor (MFA) al requerir múltiples métodos de verificación (como claves sincronizadas o dispositivos físicos) para validar la identidad del usuario al realizar accesos o compras en línea. Y, adicionalmente, a través de un Firewall será posible filtrar conexiones enfocado desde el lado administrativo.
* **Límite de exposición:** La implementación de una pasarela de pagos certificada permite procesar transacciones financieras de forma segura, asegurando la confidencialidad, integridad y protección de datos sensibles como los de tarjetas de débito o crédito. Estas plataformas están diseñadas para cumplir con estándares internacionales de seguridad (como PCI-DSS) y ofrecen mecanismos de cifrado y validación antifraude, minimizando el riesgo de filtración o manipulación de la información.
* **Mantener la integridad:** Mediante los certificados SSL (Secure Socket Layer) se logra establecer una conexión segura y cifrada entre el cliente y el servidor, protegiendo la confidencialidad e integridad de los datos transmitidos. Esta táctica garantiza que la información sensible, como credenciales o datos financieros, no pueda ser interceptada ni alterada por terceros (ataques man-in-the-middle). La implementación de SSL/TLS se realiza mediante el protocolo HTTPS y certificados emitidos por una autoridad certificadora (CA) confiable.

### Flexibilidad

* **Escalado horizontal:** El escalado horizontal consiste en aumentar el número de servidores (o instancias) para manejar una mayor carga de trabajo. Esta estrategia permite distribuir eficientemente la demanda entre múltiples nodos, evitando cuellos de botella y manteniendo un rendimiento estable y sostenible.
* **Lanzamientos graduales:** Para reducir riesgos operativos y aumentar la calidad de entrega de valor, se propone adoptar prácticas de integración y despliegue continuo (CI/CD), automatizando todas las fases del ciclo de vida del software. En este apartado, se implementa el despliegue *Zero Downtime*, el cual puede usar estrategias *Blue/Green* (servicio duplicado, con transferencia del tráfico de usuarios desde la anterior hacia la nueva versión) o *Canary Releases* (despliegue gradual a un subconjunto de usuarios) para mitigar riesgos y evitar degradar la latencia aceptada.

### Usabilidad

* **Diseño centrado en el usuario:** El sistema permite a los pequeños negocios crear apartados de tienda altamente personalizables, a través de un portal modular basado en *templates* reutilizables. Estos *templates* incluyen componentes UI prediseñados y configurables, con soporte para *wizards* que guíen al cliente en la configuración inicial del portal (por ejemplo, selección de pasarelas de pago, estructura del catálogo). Cada proveedor podrá contar con un *dashboard* interactivo, enfocado en métricas clave como ventas, comportamiento de sus compradores y gestión de inventario, con previsualizaciones que facilitan la toma de decisiones.
* **Retroalimentación al usuario:** Para mantener al usuario informado de forma oportuna, es pertinente implementar un sistema de notificaciones en tiempo real utilizando tecnologías como *WebSockets*. Estas notificaciones abarcarán eventos clave como pedidos confirmados, pagos rechazados, cambios en el inventario, entre otros; donde los usuarios podrán personalizar qué tipo de alertas desean recibir y cómo.

## Patrones Arquitectónicos

En base a las distintas tácticas identificadas para tratar los posibles escenarios que son potenciales a presentarse en la interacción del sistema, se decide abordar esta organización a través de un patrón de arquitectura microservicios. Además, se tiene en cuenta también la experiencia y acercamiento general del equipo hacia este patrón arquitectónico. Asimismo, a fin de apoyar la fiabilidad y seguridad de asegurar de que las demás funcionalidades repartidas en cada uno de los microservicios no necesariamente se vean afectados por problemas en otro y, del mismo modo, lograr dar una escalabilidad a nivel individual según se requiera. Por tal razón patrones como el monolítico, debido a la cantidad esperada de usuarios proveedores como compradores, no son tomadas como candidatas a plantear una solución acorde ya que presentan una escalabilidad menos conveniente para el cumplimiento de los atributos de calidad. En contraste, arquitecturas como microkernel, si bien proponía una solución apta para la personalización de cada tienda, bajo los requerimientos dados, su enfoque se consideró no viable por su centralización de procesamiento que no convencía en comparación a otras maneras de conectar con los servicios desde un único punto tal como un patrón API Gateway. Este último ofrece otras ventajas para la integración del sistema como la posibilidad de integrar un balanceador de cargas, monitoreo, logging, manejo de errores, y aplicar módulos de seguridad para lograr conectar al cliente identificado bajo cierto acceso a las rutas correspondientes. A comparación de su posible alternativa con Backend For Frontend (BFF), no se identificó una especial atención a un cliente y con la guía de rutas adecuada del API Gateway se concluyó los patrones a implementar junto con su compatibilidad con microservicios.

Al elegir microservicios con API Gateway como el más adecuado, se halló que gracias al diseño topológico del primero, es compatible con otros patrones que presentan una integración de interés apoyando las comunicaciones entre componentes del sistema a una escala más grande como un sistema de mensajería entre los servicios basado en eventos para coordinar cambios necesarios que involucren los respectivos con Event-Driven, desacoplando estas llamadas en servicios que ejecuten una cadena de peticiones entre los involucrados.

## Estrategia

Para este punto, se partirá de la arquitectura selecta de microservicios que implementará un API Gateway por donde se interceptarán todas las peticiones de los clientes, una vez capturados acá, se plantean dos módulos enfocados a monitoreo y seguridad para asegurar las tácticas descritas en monitoreo proactivo, autenticación de usuario e incluso aspectos técnicos como manejo de fallos en caso uno de los servicios a los que se dirige la petición del cliente falle. Gracias a este patrón, se considera el uso del patrón Circuit Breaker y Retry para el manejo de errores en caso tal uno de los servicios no se encuentre disponible al intentar acceder a este.

Por otro lado, a nivel de servicios se hará disposición de 3 tipos de almacenamiento, contando con caché para agilizar información recurrente como la información del usuario, su historial, etc. Estos harán uso de una base de datos compartida, para el enfoque Multi-tenant que se realiza divididos por *schemas*. Servicios especializados para guardar información multimedia también son incluidos en esta capa de componentes. Además, la definición de un Event Bus para el patrón Event-Driven, es crucial en procesos como la generación de un pago para desacoplar la llamada de este respecto a los demás servicios y mantener un mejor sistema modular con las responsabilidades únicas de los servicios.

Por último, en la condición de mayor permiso como lo es el usuario administrador, se opta por definir un cliente distinto a los comunes que pueda ser verificado mientras el módulo de seguridad y un firewall que permita filtrar la conexión autorizada a unos servicios exclusivos para este rol en particular. Así como los demás servicios pueden escalar, estos no se esperan que cuenten mayor variabilidad en sus conexiones. Tal implementación mantiene la mantenibilidad y conexión con los demás servicios.

# Modelo de Arquitectura 4 + 1

La arquitectura de la herramienta será proyectada mediante el Modelo de Vista 4+1 debido a que proporciona una estructura coherente y clara en la organización de la información del negocio, lo cual ayuda a visualizar y comunicar la estructura que maneja siendo influenciada por los requisitos del usuario.

## Escenarios Identificados

### Casos de Uso de Comprador

Diagrama

El contenido generado por IA puede ser incorrecto.

Mediante este diagrama se proyecta los diversos y principales comportamientos que tiene la herramienta frente al perfil de un comprador. En aquel, se evidencia que acciones como la búsqueda de un producto, modificación del carrito y uso de cupones de descuento no necesitan tener una sesión iniciada por el actor (comprador) para su respectivo uso, al igual que consultar las diferentes tiendas que se encuentran activas. Sin embargo, si el comprador desea hacer efectiva la compra de los productos almacenados en el carrito debería encontrarse autenticado en la herramienta; esto mismo sucede para las acciones de dejar una reseña de un producto, ver su historial de compras y solicitar devolución o reembolso de algún producto que haya comprado, ya sea por razones de expectativa o en calidad de llegada.

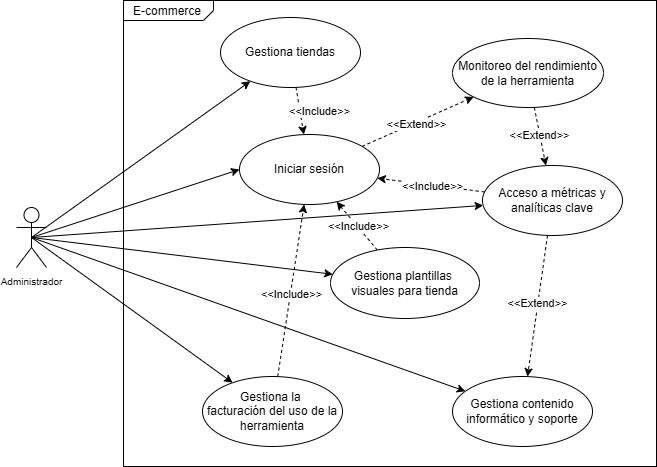
### Casos de Uso de Proveedor

Diagrama

El contenido generado por IA puede ser incorrecto.

El proveedor, encargado de gestionar su tienda hacia su público comprador, está encargado inicialmente de crear y configurar la tienda mediante la cual venderá sus respectivos productos o servicios. En este apartado el proveedor logra personalizar a su estilo el negocio del cual estará encargado de administrar, como ejemplo sería la portada de presentación, logotipo de su tienda, activación de pasarela de pago, idioma de presentación de sus productos, entre otras funcionalidades ya descritas con anterioridad. Además, el proveedor estará encargado de gestionar su inventario, catálogo, información detallada de los productos, promociones y cupones, y un acceso a reportes básicos de venta, como se presencia en el diagrama. Todos estos servicios de la herramienta se encontrarán habilitados cuando se encuentre al día y realizando de manera activa los pagos por el uso de la misma, lo cual se podrá realizar en el mismo perfil del proveedor cuando tenga una o más tiendas activas a su nombre.

### Casos de Uso de Administrador



Como se puede observar en el diagrama, el administrador no solo tendrá la opción de monitorear el uso de la herramienta por parte de los proveedores y gestionar el contenido informático y de soporte, sino que también será capaz de evaluar la creación de tiendas que soliciten los proveedores a fin de tener una regulación del comercio que desee estar en la herramienta e implementar la respectiva facturación a cada negocio involucrado; además, podrá adjuntar y modificar plantillas de presentación visual dispuesto a todas las tiendas registradas en el sistema. Todo esto será respaldado si el administrador se encuentra con su sesión iniciada.

## Vista Lógica

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Este diagrama busca presentar las características y funcionalidades de una plataforma global de comercio electrónico (*E-Commerce*). Dirigida a dos tipos de usuarios principales:

* **Usuarios**, interesados en descubrir nuevos comercios, realizar compras y crear su tienda dentro de la herramienta.
* **Administradores**, individuos encargados de administrar el uso de todo el sistema e-commerce.

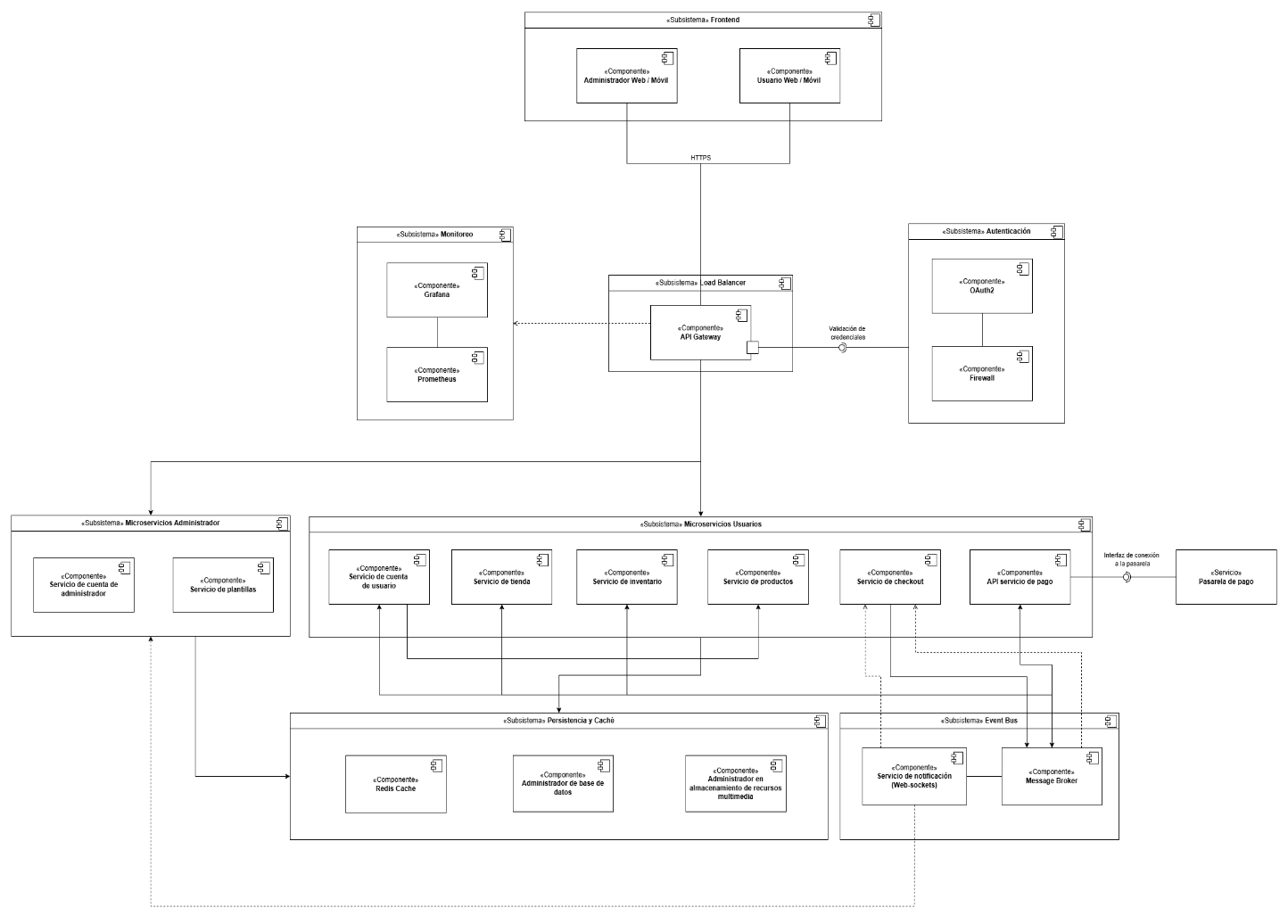
El software dispondrá de acceso rápido bajo dispositivos mayormente demandados en la actualidad (computadores y teléfonos). Para la protección de privacidad y confidencialidad de los usuarios, se implementarán protocolos de seguridad como autenticación (OAuth2) y verificación Multi-Factor (MFA).

El acceso a las distintas funcionalidades dependerá del perfil y necesidades del usuario, ya sea para comprar productos o para construir y gestionar una tienda en línea. Entre las funciones esenciales se incluyen:

* Gestión de productos
* Carrito de compras
* Registro de usuarios
* Creación de tiendas virtuales
* Métodos de pago seguros

Se integran servicios tecnológicos para garantizar la eficiencia en los procesos y la seguridad en el almacenamiento de datos. Plataformas de computación en la nube permiten una infraestructura escalable, flexible y altamente disponible, lo que asegura un funcionamiento continuo y una gestión eficiente de recursos.  
Asimismo, el uso de pasarelas de pago reconocidas (PayPal o Mercado Pago) aseguran transacciones seguras, rápidas y compatibles con múltiples monedas y métodos de pago, cumpliendo con estándares internacionales de protección de datos financieros.

## Vista de Implementación



Para obtener un vistazo a la implementación del sistema se optó por un diagrama de componentes que describen las partes importantes del diseño del software bajo una división de cada responsabilidad del negocio y que reflejan sus propósitos acordes a lo definido en la estrategia lo cual permite el manejo de cumplimiento de los atributos de calidad principales identificados en previas secciones.

Siguiendo con un flujo común de interacción, los **clientes** están divididos en dos componentes con el objetivo de distinguir el acceso al sistema. Como se expresaba en la estrategia, el rol de administración presenta un acceso exclusivo a funcionalidades reservadas del sistema de software. Para los dos tipos de usuarios tanto del proveedor (o dueño(s) del negocio), que abre una tienda gracias a la herramienta, como del cliente de este o el denominado comprador, se expresa un acceso bajo un mismo componente de cliente donde cambiará la información presentada para cada uno.

Seguidamente se define un punto de acceso único hacia los servicios que se encargará de procesar las conexiones. Gracias a este patrón **API Gateway**, es posible anexar otros módulos que sirven tanto monitoreo de métricas técnicas del sistema con **Grafana** y **Prometheus,** también para la verificación de acceso mediante la autorización procesada por el componente **OAuth2 y** la filtración de conexiones exclusivamente autorizadas para ciertas rutas con una configuración de **Firewall** enfocada a validar la petición del cliente definido para la administración.

Por parte de los microservicios, se hace una distinción entre aquellos reservados para las tiendas y una vez más, los servicios exclusivos de administración. A continuación, se define la responsabilidad de cada componente de servicio para los usuarios comunes (proveedor y comprador).

Una vez realizada la conexión, se definen los siguientes microservicios en su capa respectiva:

* **Servicio de cuenta de usuario:** Contiene la lógica de manejar los datos de los usuarios proveedor y comprador constando la información de los perfiles de sus cuentas.
* **Servicio de tienda:** Gestiona la información de la tienda tanto la básica como la legal de la misma modificada por el proveedor dueño de esta, incluyendo la personalización, facturación u ordenes creadas por compras de clientes, gestión de promociones, reembolsos solicitados, etc.
* **Servicio de inventario:** Se enfoca en el manejo del inventario registrado por la tienda y operaciones de consulta de los productos publicados en la respectiva.
* **Servicio de productos:** Se encarga de la manipulación de información de productos como su categorización, búsqueda, reseñas registradas y tienda asociada a este.
* **Servicio de checkout:** Maneja el proceso de pago del comprador, desde modificar el carrito, hasta el formulario y su elección de pago.
* **API servicio de pago:** Servicio que integra servicios de terceros de pasarelas de pago, desacoplando la responsabilidad del **servicio de checkout**, permite la conexión para el procesamiento del pago.
* **Servicio de cuenta de administrador:** Contiene la lógica de negocio acorde a los requerimientos del administrador centralizando el monitoreo de la plataforma y gestionando la visualización de tiendas de los proveedores.
* **Servicio de plantillas:** Gestiona las modificaciones globales que comparten la herramienta a los proveedores que personalizan sus tiendas.

Cabe resaltar de la misma manera, el subsistema de bus de eventos. Este se encarga de contener los componentes que implementarán el patrón Event-Driven y a los que se suscribe el dashboard respectivo para tener datos actualizados de la plataforma como servicios de la capa acceso común.

* **Message Broker:** Desacopla y mantiene los mensajes en un orden específico para este contexto, utilizado cuando un comprador efectúa un pago por un producto, permitiendo una sola petición de **servicio checkout**. Este procesa en orden las peticiones para **API servicio de pago**, **Servicio de inventario**, **Servicio de tienda** y **Servicio de cuenta de usuario.**
* **Servicio de notificación:** Implementa mensajes en tiempo real para componentes suscritos a este donde se envían las métricas que suceden en los eventos de las tiendas.

Por último, se diseña una capa de almacenamiento compartida donde se define componentes de **manejo de caché con Redis**, para guardar información de constante consulta que se realizan a algunos servicios como el servicio de cuenta, productos y tienda. Adicionalmente, se definen componentes básicos como la **base de datos** para almacenar los datos relacionales de las cuentas de los usuarios, productos, historiales, pagos, etc.; para posibles datos multimedia en la definición de tiendas, se involucra un componente especializado que conecte con el sistema de almacenamiento especializado para este tipo de datos.

## Vista de Procesos

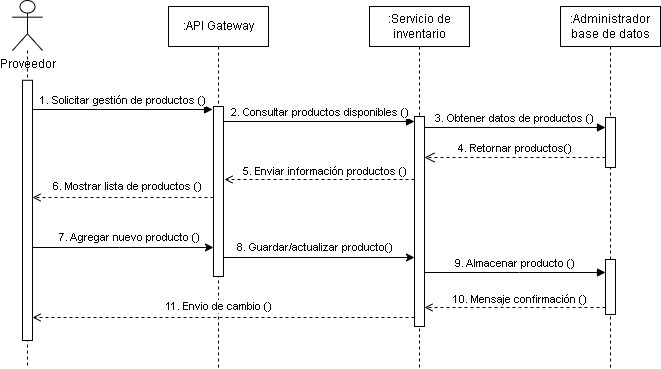
* Proceso 1: Creación de tienda

Diagrama

El contenido generado por IA puede ser incorrecto.

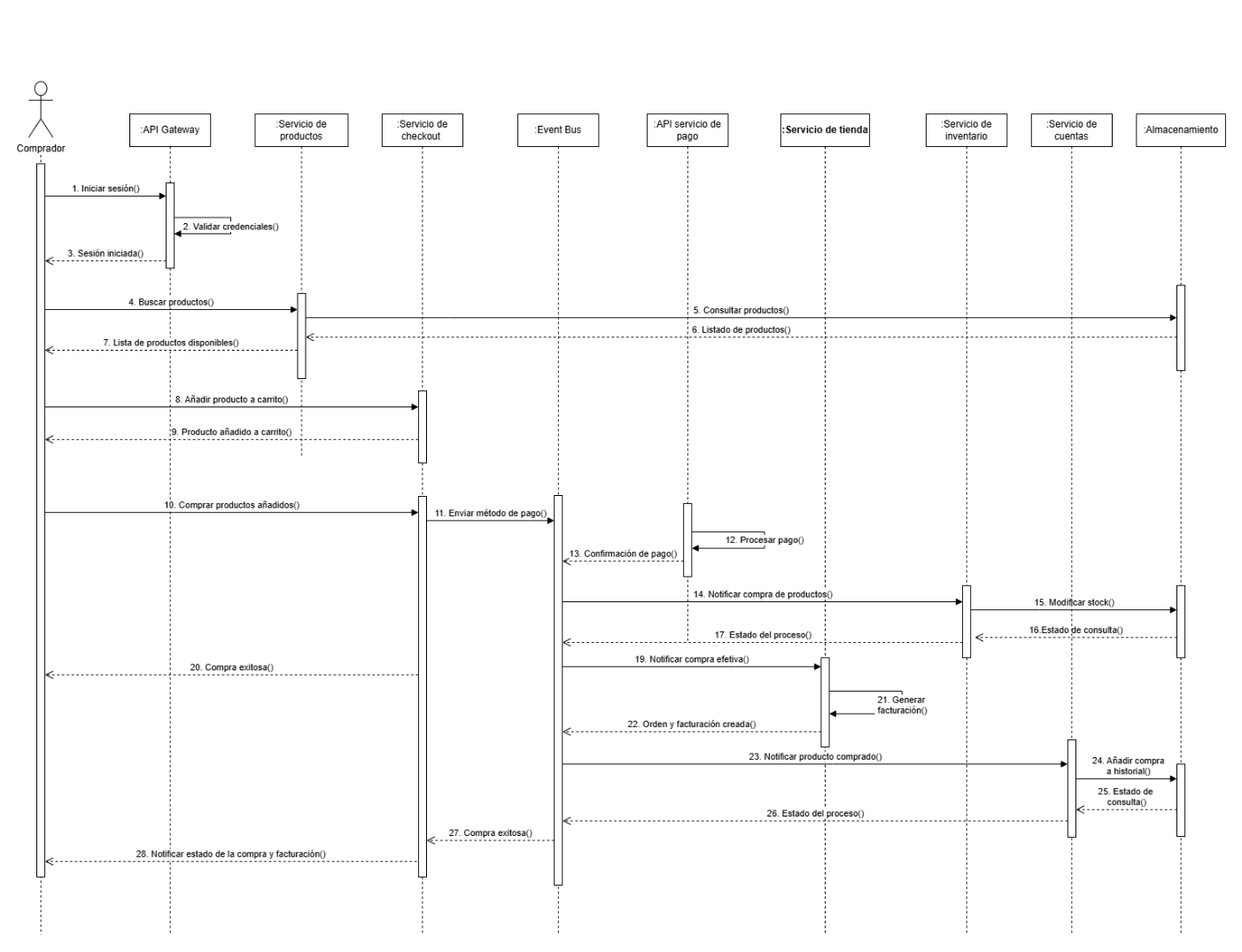
La secuencia describe el proceso del actor proveedor al crear una tienda online, por lo tanto el sistema solicita estar registrado previamente para poder acceder a las funcionalidades del sistema, bajo la conexión del **API Gateway** se realiza los procesos de validación, para luego proceder a su inicialización. Ya internamente el sistema **Persistencia y cache** aloja los recursos multimedia para no entorpecer los datos relevantes de la **Base de datos** y mantener alejados información estática de la tienda. Una vez digitados los parámetros para la configuración de la página como (moneda, idioma, país, pasarela de pagos, etc.), se llevará la información a los correspondientes servicios ajustando la configuración y confirmación con **servicio de tienda.** Finalmente, una vez llevado el proceso, la herramienta hará los cambios y acto seguido se le notificará al usuario por pantalla.

* Proceso 2: Gestión de productos



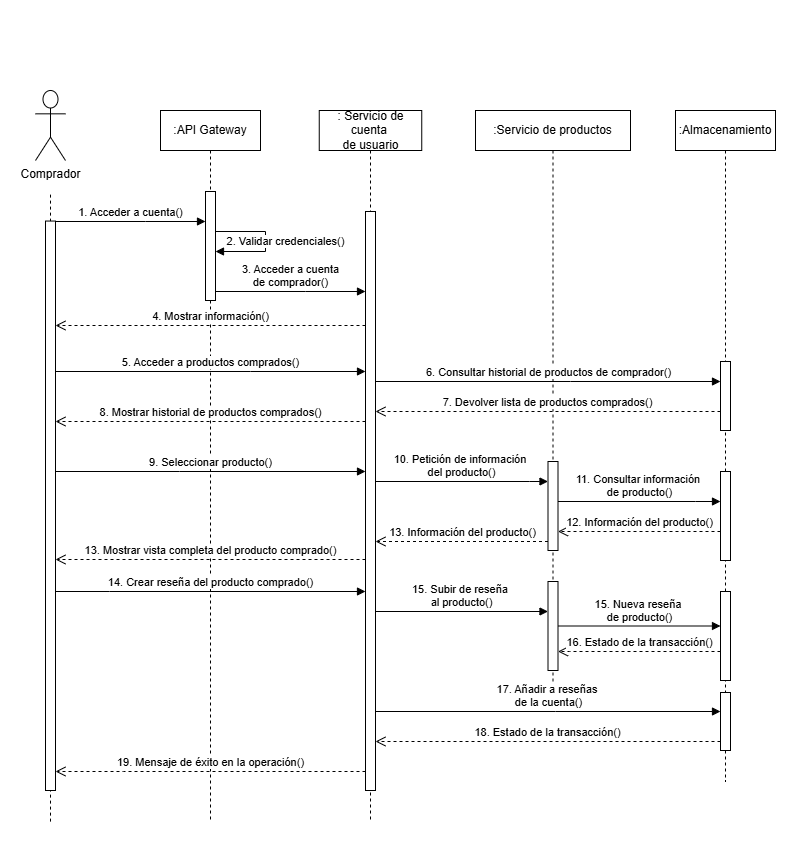
La secuencia describe el proceso de gestión de productos del actor proveedor para la gestión de productos ya estando en la tienda online. Bajo esta secuencia, el actor solicita los servicios del sistema, siendo el **Servicio de Inventario** el encargado de la consulta de los productos almacenados. Luego, con el **Servicio Administrador de Base de Datos**, se reintegra la información al usuario. Ya con la información, el actor agrega los cambios y sucesivamente almacena la información mediante la base de datos del administrador, retornando el cambio al proveedor.

* Proceso 3: Realización de compra por parte de un cliente de un negocio.



La secuencia describe el proceso del actor comprador desde su inicio de sesión, es previamente autenticado por el subsistema de seguridad respectivo conectado al **API Gateway**, acá ocurre la validación y redirige al usuario comprador a la ruta especificada; donde **Servicio de productos** permite la búsqueda por categorías o consultas específicas (filtros). Avanzando al momento de hacer el pago, una vez el usuario ha añadido y llenado el formulario respectivo de la tienda en particular, entonces **Servicio checkout** realiza una única petición capturada por el **Message Broker** dentro del **Event Bus**, el cual controlará el flujo del mensaje que primero irá hacía **API servicio de pago**, únicamente recibido el estado de un pago exitoso, este seguirá el control hacia **Servicio de inventario** donde se hará el proceso de control del stock o modificaciones necesarias para el tipo de producto que fue realizada la compra. Seguidamente, al realizarse esta acción en este servicio, se comunicará con el **Servicio de tienda** para procesar la facturación o generar una orden, así como otros datos relacionados a la compra del cliente. Finalmente, de ser un proceso exitoso, como resultado se añadirá el producto en el historial de compras del cliente gracias a **Servicio de cuenta de usuario.** Al final de la secuencia, el comprador es notificado con la facturación respectiva generada posterior del orden de llamados de servicio.

* Proceso 4: Cliente del negocio realiza una reseña a un producto registrado en su historial de compras.



Esta secuencia involucra servicios como **Servicio de cuenta de usuario, Servicio de productos** y los componentes de almacenamiento. El actor, como comprador, es validado para acceder a esta ruta a lo largo del proceso por el **API Gateway**, pero es indicado únicamente al inicio. Posterior, se indica la manera en la que la cuenta recupera datos del usuario y sus compras, pero este **Servicio de cuenta de usuario** debe hacer una petición para poder obtener la información completa del producto donde se agregará una reseña por parte de un comprador. Esta modificación llega al **Servicio de productos,** el servicio que contiene la lógica para hacer estas acciones en particular.

* Proceso 5: Administrador en gestión de facturación de proveedores

Diagrama

El contenido generado por IA puede ser incorrecto.

Este diagrama de secuencia ilustra el proceso completo mediante el cual un administrador de la plataforma visualiza qué proveedores (propietarios de tiendas) han hecho el pago de acuerdo al uso que esté ejerciendo sobre el sistema (membresía). El flujo abarca desde la solicitud de datos contables por proveedor hasta la visualización de los mismos, incluyendo interacciones clave con múltiples servicios y módulos permitiendo que el proceso de facturación automática presente en la plataforma sea almacenada y visualizada.

## Vista Física

Diagrama

El contenido generado por IA puede ser incorrecto.

Para esta vista física representada con un diagrama de despliegue y utilizando servicios de AWS se procura brindar una solución escalable, segura, resiliente y de fácil mantenimiento. Esta arquitectura sigue un enfoque de microservicios desacoplados y orientados a eventos, permitiendo así una rápida evolución funcional y una alta disponibilidad. Los usuarios y el administrador acceden al sistema mediante interfaces web y móviles distribuidas por CloudFront, mientras que API Gateway gestiona las solicitudes, integrándose con Amazon Cognito que maneja los tokens JWT para la autenticación y, WAF (Web Application Firewall) para la protección contra amenazas y restringir acceso a rutas no autorizadas de acuerdo con el rol que tengan los usuarios. Por otro lado, los servicios de la lógica del negocio están contenida en múltiples microservicios desplegados en Fargate dentro de un clúster ECS (Elastic Container Service), organizados por funcionalidades y obteniendo un escalado automático, mientras que los otros microservicios relacionados al administrador no tendrán esta característica de escalado. Cada uno de los microservicios estarán interconectados con un Retry y Circuit Breaker, por lo cual, si un servicio llega a fallar temporalmente, el sistema podrá seguir funcionando sin colapsar por completo

Mientras tanto, el almacenamiento y la persistencia de datos se manejan con RDS (Relational Database Service), ElastiCache para el caché de recursos y S3(Simple Storage Service) para archivos multimedia, optimizando así el rendimiento y la disponibilidad de la información. Además, se incluye una comunicación entre el Servicio checkout y la función Lambda especializada para el manejo de pagos a través de múltiples pasarelas como PayPal, Stripe, PayU y/o MercadoPago, esto lo realiza de forma asincrónica mediante EventBridge y SNS (Simple Notification Service) para el envío de notificación a los servicios correspondientes para su respectiva actualización de información en el inventario y catálogo. Finalmente, se garantiza la observabilidad del sistema mediante CloudWatch, Prometheus y Grafana, facilitando el monitoreo, la trazabilidad y la toma de decisiones operativas de acuerdo con los datos del sistema. Por lo anterior, esta arquitectura posibilita que la herramienta sea flexible, mantenible y preparada para enfrentar demandas de tráfico variables acompañada de una evolución funcional constante.

Cabe resaltar que esta arquitectura está enfocada para ser desplegada en una sola región, por lo tanto, si se piensa abarcar en diferentes regiones a nivel global solo será necesario replicar este diagrama e interconectar las diversas bases de datos y almacenamiento del Data Store con cada una de las regiones.