**UNIVERSIDAD MAYOR, REAL Y PONTIFICIA**

**DE SAN FRANCISCO XAVIER DE CHUQUISACA**

**Facultad de Ciencias de la Tecnología**

**A logo of a university

AI-generated content may be incorrect.**

**NOMBRES:** Solorzano Arancibia Diego Santiago

Arapa Estrada Nataniel Mauricio

Quispe Sucullani Jose David

**CARRERA:** Ingeniería en ciencias de la computación

**MATERIA:** Calidad de Software

ÍNDICE

[**1. Introducción 5**](#_89uzoln26l8s)

[**2. Requisitos para la instalación 5**](#_c6v5ry49dtkf)

[2.1. Preparación del entorno 5](#_acfgfpsw1yfz)

[2.2. Creación de la base de datos 5](#_xdx6esiya8o4)

[2.3. Instalación de Aimeos 5](#_v7jqu5p8xnb0)

[2.4. Configuración del proyecto 5](#_yum88y86esz2)

[2.5. Pasos finales y ejecución 6](#_faj9imc94hf0)

[**3. Diagramas del Software 7**](#_uhet4b661t9v)

[3.1 Diagrama de casos de uso 7](#_yvy4bmo0x4f5)

[3.1.1. Los Actores (Quiénes Usan el Sistema) 8](#_utg6mn5lhytm)

[3.1.2. Las Funcionalidades (Qué Hace el Sistema) 8](#_lit6ar4qy8q7)

[3.2 Diagrama de Componentes 9](#_smmckd14v0c9)

[3.2.1. Routing (Enrutamiento) 9](#_4jezqvlnpoa)

[3.2.2. Controllers (Controladores) 9](#_xmdj1yf1z4ny)

[3.2.3. Interfaz de Usuario (Vistas) 10](#_z9xhai1fgaev)

[3.2.4. Aimeos Bridge (Servicios) 10](#_dt495ptub1d1)

[3.2.5. Aimeos Core (Motor E-commerce) 10](#_8rvvkqeyfx0i)

[3.2.6. Laravel Framework 10](#_k9f8qbc2kz9a)

[3.2.7. Database (Base de Datos) 10](#_lz8e6w2eaivr)

[**3.3 Diagrama de Despliegue 11**](#)

[3.3.1. Dispositivo del Cliente 11](#_xolpq0fa6orb)

[3.3.2. Servidor Web 12](#_6i9174qricfn)

[3.3.3. Servidor de Base de Datos 12](#_ilr5i9lkygpr)

[**3.4 Entidad Relación 13**](#)

[3.4.1. Entidades Centrales (Tablas Principales) 14](#_djwdbp3wjtj7)

[3.4.2. Entidades Relacionadas con el Producto 14](#_81x6ipklq4yu)

[3.4.3. Otras Entidades Clave 15](#_5uo2poemcroh)

[**4. Propósito de la Evaluación 15**](#_a5zffh2wkskv)

[**5. Identificación del Producto a Evaluar 15**](#_mgpu2z6d2avc)

[5.1 Características 15](#_4kjgrugozrcf)

[5.2 Subcaracterísticas 16](#_zdjtl67hfb6n)

[5.3 Métricas 16](#_cpfanxdil2pz)

[**6. Modelo de Calidad 16**](#_q72sbagri9yx)

[6.1 Calidad Interna 16](#_q4i6r4gn7ibl)

[6.1.1 Característica: Funcionalidad 16](#_eoh09wmyqdym)

[6.1.2 Característica: Mantenibilidad 17](#_2ppbrgaorm23)

[6.2 Calidad Externa 18](#_ft4rgebzgl0d)

[6.2.1 Característica: Usabilidad 18](#_u31muayr3owh)

[6.2.2 Característica: Funcionalidad 19](#_kqzo4g8m9yxt)

[6.3 Calidad En Uso 20](#_460ih8wntky5)

[6.3.1 Característica: Eficacia 20](#_zd7b2u5op77r)

[6.3.2 Característica: Productividad 20](#_t93je1hg5cdj)

[6.3.3 Característica: Seguridad 20](#_rjw8f1i5jmwt)

[6.3.4 Característica: Satisfacción 20](#_5a9zyc50a442)

[7. Niveles de puntuación de las métricas 21](#_jp6pjydkt6fd)

[7.1 Métricas de Calidad Interna 21](#_jp6pjydkt6fd)

[7.2 Métricas de Calidad Externa 23](#_o1li3ytpkb4u)

[7.3 Métricas de Calidad en Uso 24](#_h3uaw2z3efdi)

[**8. Criterios de evaluación 25**](#_x3ofmxnok6w9)

[8.1 Calidad Interna 25](#_wuapf1v2zsue)

[8.1.1 Escenario 25](#_d0fahfcdh4k6)

[8.1.2 Herramientas de Medición 26](#_620v5wp7omlf)

[8.1.3 Criterios de Aceptación 27](#_349q449dfsoy)

[8.2 Calidad Externa 28](#_2edyd0qc5cef)

[8.2.1 Escenario 28](#_chl66kh6iwd2)

[8.2.2 Herramientas de Medición 28](#_ityp9xwyuy0b)

[8.2.3 Criterios de Aceptación 29](#_8zbvoxa53lay)

[8.3 Calidad en Uso 29](#_k41f5vmpamzw)

[8.3.1 Escenario 29](#_jhb12a3g3npd)

[8.3.2 Herramientas de Medición 30](#_4hwe6w6tvzcz)

[8.3.3 Criterios de Aceptación 31](#_jfdj23sq3688)

[**9. Plan de evaluación 31**](#)

[9.1 Desglose Detallado de Tareas de Evaluación 31](#_undddfmipx0c)

[9.1.1 Revisión Documental y Técnica 31](#_q4vdwmubthtr)

[9.1.2 Fase 2: Evaluación de Métricas Internas 32](#_dowlhko7aepd)

[9.1.3 Fase 3: Evaluación de Calidad Externa 32](#_xmnspkc1gtq8)

[9.1.4 Fase 4: Evaluación de Métricas de Uso 32](#_yqo80h9gf2kq)

[**10. Ejecución del plan de evaluación 33**](#_wpxf2nr0us7)

[10.1. Calidad interna 33](#_3omqj28oxbhr)

[10.1.1 Adecuación funcional. - 33](#_hn1098x6jnm8)

[10.1.2 Exhaustividad funcional: 34](#_i6g4rf7shzet)

[10.1.3 Exactitud computacional: 34](#_zgh6ttb6ra5w)

[10.1.4 Precisión de resultados: 35](#_fx1kzovhnesx)

[10.1.5 Intercambio por formato: 36](#_ooc9zoiagfsv)

[10.1.6 Intercambio general: 36](#_ma9p9tf91jkf)

[10.1.7 Integridad de descripción: 37](#_i1q2ip3lmaia)

[10.1.8 Acceso a demostraciones: 37](#_d1njll3rnyfb)

[10.1.9 Cancelabilidad: 38](#_eeqybl2x83eh)

[10.1.10 Personalización: 38](#_lkfz0bn7ffkp)

[10.1.11 Interacción atractiva: 39](#_xyvzo29hzf6q)

[10.1.12 Personalización de interfaz 39](#_ekh4x26e68f8)

[10.2 Calidad interna 40](#_dxutntlr8dyk)

[10.2.1 Adecuación funcional.- 40](#_5u4x85benqbq)

[10.2.2 Exhaustividad funcional: 41](#_riinfla7sa88)

[10.2.3 Precisión de datos internos: 41](#_ykj8g8a6y2z5)

[10.2.4 Exactitud de cálculos internos: 42](#_cg4y7i98l3og)

[10.2.5 Cumplimiento de estándares: 43](#_l2tmfq2ho6s5)

[10.2.6 Completitud de formatos: 43](#_vngz1pa7wshz)

[10.2.7 Ratio de comentarios: 44](#_obljc4dq6drm)

[10.2.8 Complejidad ciclomática: 44](#_yqzsseur7szn)

[10.2.9 Modularidad: 45](#_vjrdy8ste3qc)

[10.2.10 Tamaño de funciones: 45](#_3l2i3hxoz0ec)

[10.2.11 Estabilidad de interfaces: 45](#_nee3bxvvmuwn)

[10.2.12 Cohesión de módulos: 46](#_dwk9p7qefxdo)

[10.3 Calidad de uso 46](#_tx2w3ehxl58c)

[10.3.1 Tasa de éxito en tareas: 46](#_kpmy9sffu8sx)

[10.3.2 Precisión de resultados: 47](#_cam1ebpim3ma)

[10.3.3 Tiempo medio de tarea: 48](#_o7z0wzamvq2v)

[10.3.4 Eficiencia de interacción: 48](#_54ijncitffaj)

[10.3.5 Tasa de errores prevenidos: 49](#_wvr40quyr7zn)

[10.3.6 Impacto de fallos: 49](#_ndwai2q5oaka)

[10.3.7 Nivel de satisfacción: 50](#_s7eym6ktlxvd)

[10.3.8 Tasa de recomendación: 50](#_a92govtiqxpm)

[11.1 Calidad interna: 51](#_c08r7tgl1nyn)

[11.2 Calidad externa: 53](#_t9jnj14rwwjx)

[11.3 Calidad de uso 55](#_f1ytlyknn8bo)

[**12. Recomendaciones 56**](#_amg25uszqqfc)

[12.1. Seguridad del sistema 56](#_apqq2vx5hrhh)

[12.2. Funcionalidad crítica 56](#_2lrdq3eqbln6)

[12.3. Proceso de desarrollo y documentación 56](#_dk8kmgoqwgup)

[**13. Conclusión 57**](#_bf71ow7o8uq2)

# Introducción

El sistema Aimeos es un framework de e-commerce basado en Laravel que permite implementar soluciones de venta en línea altamente personalizables y escalables. Su arquitectura modular soporta tanto entornos de tienda única como configuraciones multi vendor, ofreciendo una amplia gama de funcionalidades que cubren el ciclo completo de un comercio electrónico: gestión de catálogos, procesos de compra, administración de pedidos, integración con pasarelas de pago y proveedores de envío, así como herramientas de backoffice para la administración de productos, usuarios e inventario.

# Requisitos para la instalación

## 2.1. Preparación del entorno

* + Instalar XAMPP: Descarga XAMPP desde su página oficial:<https://www.apachefriends.org/es/index.html>. Instala con Apache, MySQL y PHP.
  + Instalar Composer: Descarga Composer desde<https://getcomposer.org/download/>. Instala y verifica con: composer -V
  + Instalar Node.js + NPM: Descarga Node.js desde<https://nodejs.org>. Verifica la instalación con: node -v y npm -v
  + Habilitar extensiones de PHP: Abre el archivo php.ini en la carpeta de instalación de XAMPP (C:\xampp\php\php.ini) y habilita (quitando el ; al inicio de la línea):
    - extension=zip
    - extension=curl
    - extension=intl
    - extension=gd
    - extension=pdo\_mysql

## 2.2. Creación de la base de datos

* + Abre phpMyAdmin desde http://localhost/phpmyadmin.
  + Crea una nueva base de datos, por ejemplo aimeos\_db. No es necesario migrar tablas, Aimeos lo hace en la instalación.

## 2.3. Instalación de Aimeos

* + Busca una carpeta de instalación, abre la terminal y ejecuta el comando:
    - composer create-project aimeos/aimeos myshop

## 2.4. Configuración del proyecto

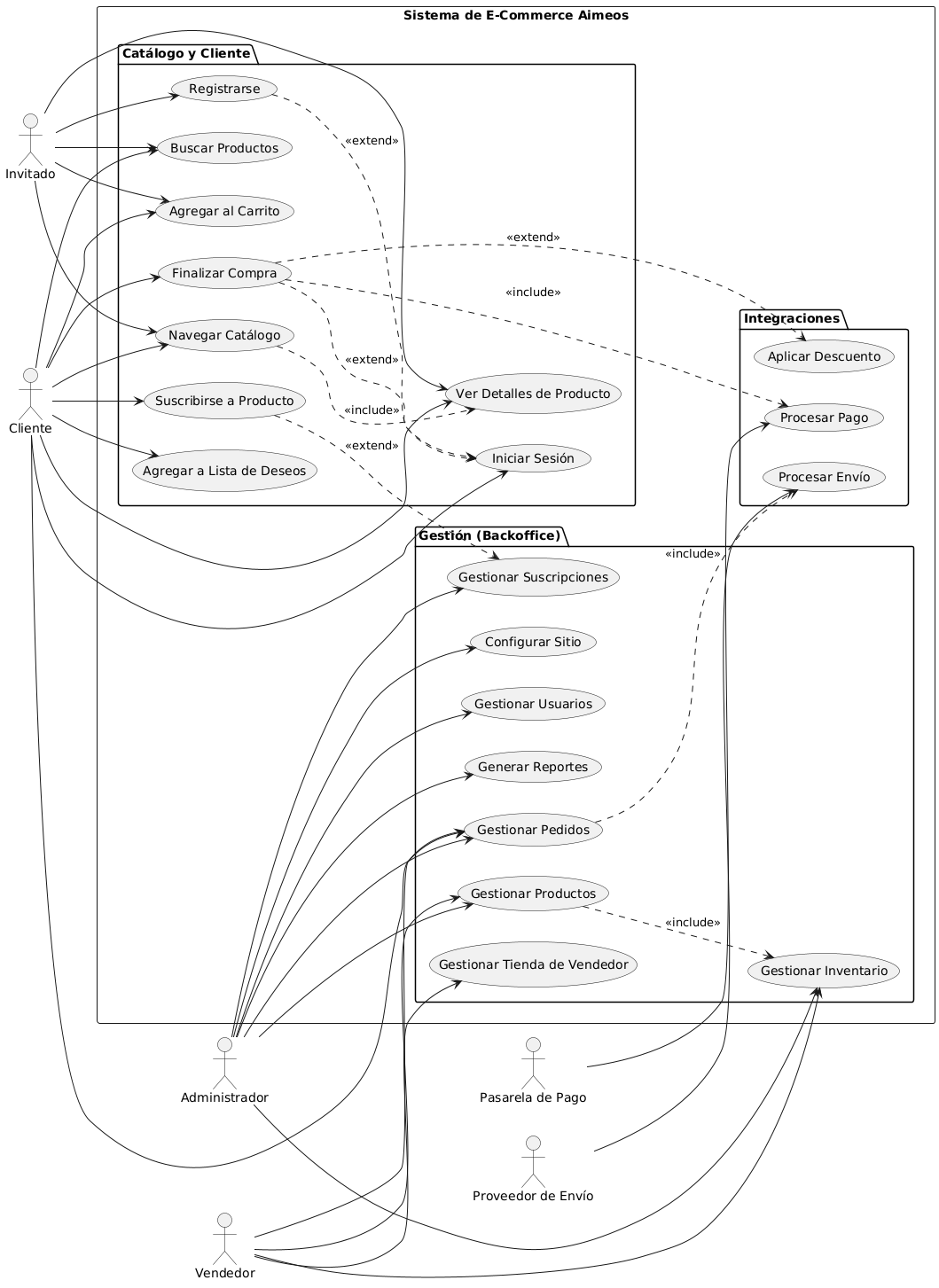
* + Durante la instalación, la terminal pedirá datos. Para un entorno de prueba, se recomienda:
    - Database setup:
      * DB\_CONNECTION: (dejar en blanco si no usas otra base de datos, por defecto mysql)
      * DB\_HOST: 127.0.0.1 (por defecto apunta al localhost)
      * DB\_PORT: (dejar en blanco si no cambiaste el puerto en XAMPP)
      * DB\_DATABASE: myshop (aquí va el nombre de la base de datos que creaste)
      * DB\_USERNAME: (dejar en blanco, por defecto root)
      * DB\_PASSWORD: (dejar en blanco si no tiene contraseña en XAMPP)
    - Mail setup:
      * Puedes dejar todos los valores en blanco.
    - Crear cuenta de administrador:
      * Introduce un correo (ejemplo: admin@localhost.com) y una contraseña.

## 2.5. Pasos finales y ejecución

* + Instala dependencias de frontend:
    - npm install
    - npm run dev
  + Para finalizar y levantar el programa:
    - php artisan serve

# 3. Diagramas del Software

## 3.1 Diagrama de casos de uso



### 3.1.1. Los Actores (Quiénes Usan el Sistema)

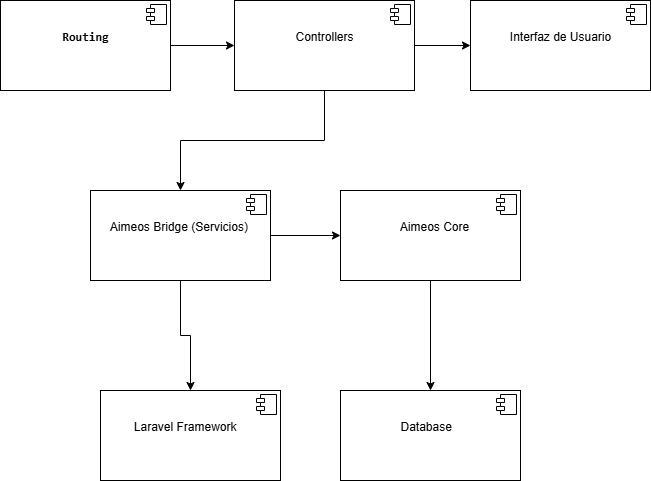
* **Invitado**: Una persona que navega por la tienda sin iniciar sesión. Puede ver productos, buscar y explorar el catálogo.
* **Cliente**: Un usuario registrado. Puede hacer todo lo que hace un invitado, pero además puede comprar, guardar artículos en un carrito, crear una lista de deseos y suscribirse a productos.
* **Administrador**: La persona a cargo del sitio. Puede gestionar todo, desde los usuarios y los pedidos hasta la configuración general.
* **Vendedor**: Un usuario que gestiona su propia parte de la tienda. Se encarga de sus productos y de su propio inventario.
* **Pasarela de Pago y Proveedor de Envío**: Son sistemas externos que procesan los pagos y gestionan los envíos. El sistema se comunica con ellos para completar las compras.

### 3.1.2. Las Funcionalidades (Qué Hace el Sistema)

Las funcionalidades se agrupan en tres secciones principales:

* **Catálogo y Cliente**: Aquí se encuentran las acciones que el usuario final realiza en la tienda.
  + **Navegar y Buscar**: El Invitado y el Cliente pueden explorar el catálogo y buscar productos.
  + **Comprar**: Para comprar, el Cliente primero agrega productos al carrito y luego "Finaliza la Compra".
  + **Relaciones <<include>>**: Estas líneas punteadas indican una acción obligatoria. Por ejemplo, para "Agregar al Carrito", el sistema primero debe "Ver Detalles de Producto". Al "Finalizar Compra", el sistema siempre debe "Procesar Pago" y "Procesar Envío".
  + **Relaciones <<extend>>**: Estas líneas muestran acciones opcionales. Por ejemplo, al "Finalizar Compra", el cliente puede "Aplicar Descuento", pero no es obligatorio.
* **Gestión (Back Office)**: Este es el panel de administración.
  + El Administrador tiene acceso a todas las funciones, como "Gestionar Usuarios" y "Gestionar Pedidos".
  + El Vendedor se enfoca en las tareas de su propia tienda, como "Gestionar Inventario" y "Gestionar Tienda de Vendedor".
* **Integraciones**: Muestra cómo el sistema interactúa con los servicios externos para completar una compra.
  + "Procesar Pago" se comunica con la Pasarela de Pago.
  + "Procesar Envío" se comunica con el Proveedor de Envío.

## 3.2 Diagrama de Componentes



### 3.2.1. Routing (Enrutamiento)

* **Función**: Actúa como el director de tráfico de la aplicación. Recibe todas las peticiones web del usuario y decide qué Controller debe encargarse de responder.
* **Componentes Clave**: El archivo principal es routes/aimeos.php.
* **Interacciones**: Es el punto de entrada para las peticiones HTTP y su única responsabilidad es delegar la tarea al componente de Controllers.

### 3.2.2. Controllers (Controladores)

* **Función**: Es el cerebro que procesa las acciones del usuario. Orquesta la respuesta coordinando a los demás componentes.
* **Componentes Clave**: Lógica para el catálogo (CatalogController), la cesta (BasketController), el proceso de pago (CheckoutController), y las APIs (JsonapiController, GraphqlController).
* **Interacciones**:
  + Depende de Aimeos Bridge para obtener la lógica y los datos de la tienda.
  + Envía los datos a la Interfaz de Usuario para que se muestren en pantalla.

### 3.2.3. Interfaz de Usuario (Vistas)

* **Función**: Es la capa de presentación; todo lo que el usuario final ve y con lo que interactúa en su navegador.
* **Componentes Clave**: Plantillas Blade (.blade.php) que renderiza el HTML, como la página de inicio, la lista de productos o la cesta de la compra.
* **Interacciones**: Recibe datos de los Controllers para mostrarlos de forma dinámica.

### 3.2.4. Aimeos Bridge (Servicios)

* **Función**: Es el componente central y el "traductor" que conecta el mundo de Laravel con el motor de Aimeos. Proporciona una forma sencilla de acceder a la funcionalidad compleja del e-commerce.
* **Componentes Clave**:
  + **Servicios Base (src/Base/)**: Gestiona el contexto, la configuración y la inicialización de los componentes de la tienda.
  + **Facades (src/Facades/)**: Proporcionan "atajos" para usar los servicios de forma limpia en el código (ej. Shop::get()).
* **Interacciones**:
  + Depende del Laravel Framework para funciones como la gestión de sesiones y la configuración.
  + Es el único que se comunica directamente con el Aimeos Core para ejecutar la lógica de e-commerce.

### 3.2.5. Aimeos Core (Motor E-commerce)

* **Función**: Es el motor de comercio electrónico principal. Contiene toda la lógica de negocio compleja para gestionar productos, precios, pedidos, etc.
* **Componentes Clave**: Es una dependencia externa gestionada por Composer (aimeos/aimeos-core).
* **Interacciones**: Proporciona la funcionalidad de e-commerce a Aimeos Bridge y es el único componente que interactúa directamente con la Database.

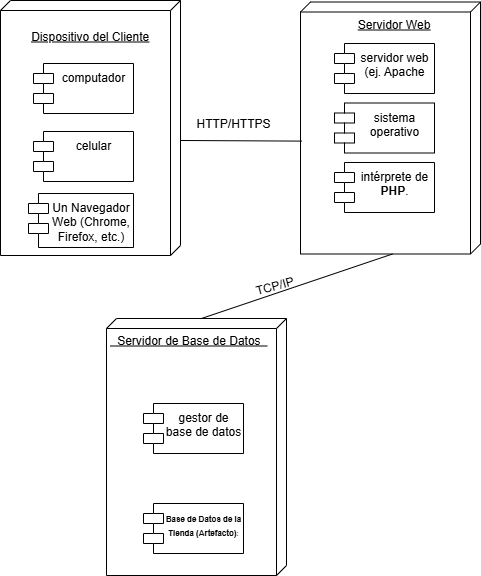
### 3.2.6. Laravel Framework

* **Función**: Es el esqueleto de toda la aplicación. Proporciona la estructura base, la seguridad y las utilidades sobre las que se construye el proyecto.
* **Componentes Clave**: Es una dependencia externa (laravel/framework).
* **Interacciones**: Ofrece servicios fundamentales (como el enrutamiento, la gestión de la base de datos y las vistas) a todos los demás componentes, especialmente a Aimeos Bridge.

### 3.2.7. Database (Base de Datos)

* **Función**: Es el almacén persistente de toda la información de la tienda.
* **Componentes Clave**: Tablas para productos, clientes, pedidos, precios, etc.
* **Interacciones**: Responde a las consultas del Aimeos Core para guardar, modificar y recuperar datos.

## 3.3 Diagrama de Despliegue



### 3.3.1. Dispositivo del Cliente

* **Función**: Es el equipo físico a través del cual el usuario final interactúa con la tienda online. Es el punto de entrada al sistema.
* **Componentes**:
  + **Hardware**: Puede ser una computadora de escritorio, laptop o un celular.
  + **Software**: El componente esencial es el Navegador Web (Chrome, Firefox, etc.).
* **Interacciones**: Se comunica con el Servidor Web a través del protocolo HTTP/HTTPS para solicitar y recibir las páginas de la tienda.

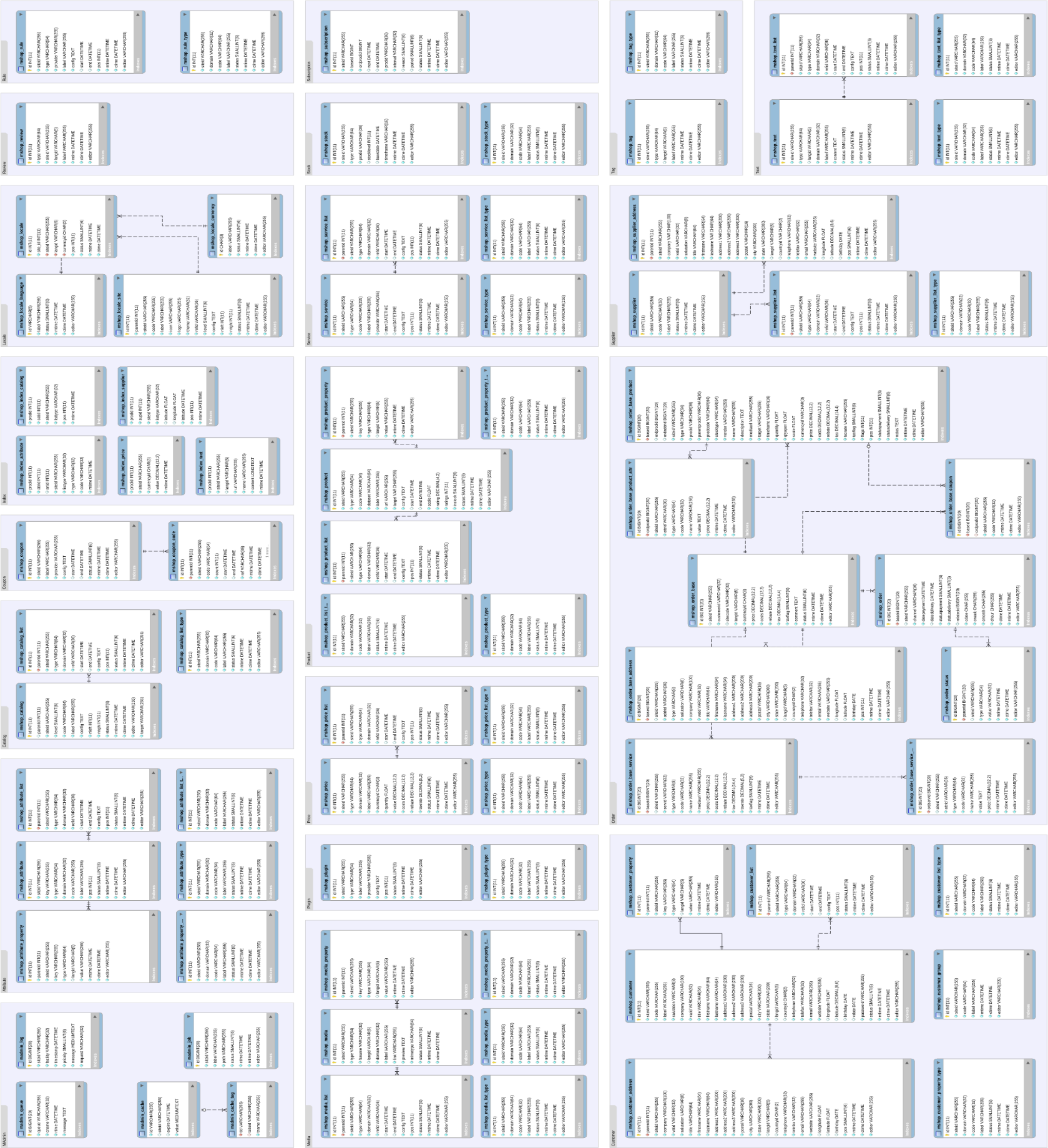
### 3.3.2. Servidor Web

* **Función**: Aloja la aplicación y ejecuta toda la lógica de negocio. Es el cerebro del sistema.
* **Componentes**:
  + **Sistema Operativo**: La base que gestiona los recursos del servidor (ej. Linux).
  + **Servidor Web (Software)**: Recibe las peticiones del cliente y las dirige (ej. Apache, Nginx).
  + **Intérprete de PHP**: Ejecuta el código de la aplicación.
  + **Aplicación Aimeos/Laravel (Artefacto)**: El código fuente de tu proyecto, que contiene todas las reglas de la tienda, los controladores y la lógica para generar las páginas.
* **Interacciones**: Recibe peticiones del Dispositivo del Cliente y se comunica con el Servidor de Base de Datos a través de TCP/IP para consultar o guardar información.

### 3.3.3. Servidor de Base de Datos

* **Función**: Actúa como el almacén central y persistente de toda la información de la tienda.
* **Componentes**:
  + **Gestor de Base de Datos**: El software que organiza y gestiona los datos (ej. MySQL).
  + **Base de Datos de la Tienda (Artefacto)**: La colección de datos en sí: tablas de productos, clientes, pedidos, etc.
* **Interacciones**: Responde exclusivamente a las peticiones del Servidor Web para proporcionar, actualizar o eliminar datos.

## 3.4 Entidad Relación



### 3.4.1. Entidades Centrales (Tablas Principales)

* **Customer (Cliente)**
  + **Propósito:** Almacena la información de los usuarios.
  + **Atributos clave:** id, nombre, email, contraseña, direcciones.
  + **Relaciones:** Se relaciona con Order (un cliente puede tener muchos pedidos) y Subscription (un cliente puede tener múltiples suscripciones).
* **Product (Producto)**
  + **Propósito:** Almacena los datos de los artículos que se venden. Es una entidad central que se conecta con muchas otras.
  + **Atributos clave:** id, nombre, descripción, código\_sku.
  + **Relaciones:** Se relaciona con Price, Media, Attribute, Stock, Catalog, Supplier y Review.
* **Order (Pedido)**
  + **Propósito**: Registra todas las transacciones de compra.
  + **Atributos clave:** id, fecha, total, estado.
  + **Relaciones:** Se conecta con Customer (cada pedido pertenece a un cliente) y con Service para los detalles de pago y envío.

### 3.4.2. Entidades Relacionadas con el Producto

* **Price (Precio)**
  + **Propósito:** Define los precios de los productos, incluyendo variaciones.
  + **Atributos clave**: id, valor, divisa, tipo\_impuesto.
  + **Relaciones**: Muchos a uno con Product. Un producto puede tener múltiples precios (por divisa, por cantidad, etc.).
* **Media (Medios)**
  + **Propósito:** Almacena las referencias a las imágenes y videos de los productos.
  + **Atributos clave:** id, url, tipo\_archivo.
  + **Relaciones:** Muchos a uno con Product. Un producto puede tener muchas imágenes.
* **Attribute (Atributo)**
  + **Propósito:** Define características adicionales para los productos.
  + **Atributos clave:** id, nombre\_atributo, valor.
  + **Relaciones**: Muchos a muchos con Product (a través de una tabla intermedia), ya que un producto puede tener muchos atributos y un atributo puede aplicarse a muchos productos.
* **Stock (Inventario)**
  + **Propósito:** Controla la cantidad disponible de cada producto.
  + **Atributos clave:** id, cantidad, disponibilidad.
  + **Relaciones:** Uno a uno o uno a muchos con Product.
* **Review (Reseñas)**
  + **Propósito**: Almacena las valoraciones y comentarios de los clientes sobre los productos.
  + **Atributos clave:** id, puntuación, comentario.
  + **Relaciones:** Muchos a uno con Product y muchos a uno con Customer.

### 

### 3.4.3. Otras Entidades Clave

* **Catalog (Catálogo)**
  + **Propósito**: Organiza los productos en categorías y subcategorías.
  + **Atributos clave:** id, nombre\_categoria.
  + **Relaciones:** Uno a muchos con Product y uno a muchos consigo misma (para la jerarquía de subcategorías).
* **Supplier (Proveedor)**
  + **Propósito:** Almacena los datos de los proveedores de la tienda.
  + **Atributos clave**: id, nombre\_proveedor.
  + **Relaciones**: Uno a muchos con Product.
* **Service (Servicio)**
  + **Propósito:** Representa los servicios de pago y envío.
  + **Atributos clave**: id, nombre\_servicio, costo.
  + **Relaciones:** Uno a muchos con Order (un pedido puede usar varios servicios).
* **Coupon (Cupón)**
  + **Propósito:** Almacena los códigos y reglas de los descuentos.
  + **Atributos clave:** id, código, valor, fecha\_vencimiento.
  + **Relaciones:** Uno a muchos con Order.
* **Subscription (Suscripción)**
  + **Propósito:** Gestiona los productos con pagos recurrentes.
  + **Atributos clave:** id, plan, periodo.
  + **Relaciones:** Muchos a uno con Customer y muchos a uno con Product.

# 4. Propósito de la Evaluación

El propósito de esta evaluación es analizar la calidad del software Aimeos para Laravel, un paquete de comercio electrónico de código abierto, diseñado para integrarse en el framework Laravel y construir tiendas en línea profesionales. Se evaluará el producto en términos de calidad interna, calidad externa y calidad de uso, con el fin de determinar su robustez, rendimiento, escalabilidad y adaptabilidad a diferentes modelos de negocio y necesidades de los desarrolladores.

# 5. Identificación del Producto a Evaluar

Aimeos para Laravel es un paquete de comercio electrónico diseñado para integrarse en aplicaciones que utilizan el framework Laravel. Su propósito es proporcionar una base robusta y escalable para construir desde tiendas en línea sencillas hasta complejos sistemas B2B y marketplaces con múltiples vendedores.

## 5.1 Características

Son los atributos principales que definen la calidad del software. Representan aspectos amplios como:

* **Funcionalidad:** ¿El software hace lo que debe hacer?
* **Usabilidad:** ¿Es fácil de usar?
* **Confiabilidad:** ¿Funciona consistentemente sin fallos?

## 5.2 Subcaracterísticas

Son subdivisiones específicas y medibles de las características principales:

* **Idoneidad:** Capacidad del software para proporcionar funciones apropiadas para las tareas y objetivos específicos del usuario
* **Precisión:** Capacidad del software para proporcionar los resultados o efectos correctos con el grado necesario de exactitud
* **Interoperabilidad:** Capacidad del software para interactuar con uno o más sistemas específicos
* **Comprensibilidad:** Facilidad con que los usuarios pueden entender si el software es adecuado y cómo usarlo para tareas particulares
* **Operabilidad:** Facilidad de operación y control del software por parte del usuario
* **Atractivo:** Capacidad del software para ser atractivo al usuario (interfaz agradable)

## 5.3 Métricas

Son medidas cuantitativas específicas que incluyen:

* **Propósito:** Qué aspecto mide
* **Método de aplicación:** Cómo recolectar los datos
* **Fórmula:** Cálculo matemático para obtener el resultado

# 6. Modelo de Calidad

A continuación, se presentan las características y subcaracterísticas elegidas para la evaluación, siguiendo un enfoque basado en calidad interna, calidad externa y calidad de uso.

## 6.1 Calidad Interna

Evaluación estática sin ejecutar el software (análisis de código y documentación)

### 6.1.1 Característica: Funcionalidad

| Sub característica | Métrica | Propósito | Formula | Interpretación |
| --- | --- | --- | --- | --- |
| Idoneidad | Adecuación funcional | ¿Cuán adecuadas son las funciones? | X = 1 - A/B  (A=funciones con problemas, B=funciones evaluadas) | Más cercano a 1 = menos funciones con problemas. |
| Exhaustividad funcional | ¿Qué tan completa es la implementación? | X = 1 - A/B  (A=funciones faltantes, B=funciones especificadas) | Más cercano a 1 = casi todas las funciones requeridas están implementadas. |
| Precisión | Precisión de datos internos | ¿Qué tan precisa es la representación de datos? | X = A/B  (A=tipos con precisión adecuada, B=tipos totales) | Más cercano a 1 = datos bien representados. |
| Exactitud de cálculos internos | ¿Qué tan exactos son los algoritmos? | X = 1 - A/B  (A=cálculos problemáticos, B=cálculos totales) | Más cercano a 1 = cálculos correctos. |
| Interoperabilidad | Cumplimiento de estándares | ¿Las interfaces cumplen estándares? | X = A/B  (A=interfaces compliant, B=interfaces totales) | Más cercano a 1 = interfaces cumplen estándares. |
| Completitud de formatos | ¿Están completos los formatos de datos? | X = A/B  (A=formatos completos, B=formatos requeridos) | Más cercano a 1 = formatos completos. |

### 6.1.2 Característica: Mantenibilidad

| Sub característica | Métrica | Propósito | Formula | Interpretación |
| --- | --- | --- | --- | --- |
| Analizabilidad | Ratio de comentarios | ¿Qué tan analizable es el código? | X = A/B  (A=líneas comentarios, B=líneas totales) | Más cercano a 1 = código mejor documentado y entendible. |
| Complejidad ciclomática | ¿Qué tan compleja es la estructura? | X = Promedio de paths independientes por módulo | Más bajo = estructura más simple y fácil de analizar. |
| Cambiabilidad | Modularidad | ¿Qué tan modular es para cambios? | X = 1 - A/B (A=dependencias excesivas, B=módulos totales) | Más cercano a 1 = pocos acoplamientos, mayor facilidad de cambio. |
| Tamaño de funciones | ¿Son las funciones de tamaño adecuado? | X = A/B  (A=funciones tamaño OK, B=funciones totales) | Más cercano a 1 = funciones con tamaño adecuado. |
| Estabilidad | Estabilidad de interfaces | ¿Qué tan estables son las interfaces? | X = 1 - A/B  (A=interfaces inestables, B=interfaces totales) | Más cercano a 1 = interfaces más confiables y con menos cambios. |
| Cohesión de módulos | ¿Qué tan cohesivos son los módulos? | X = A/B  (A=módulos alta cohesión, B=módulos totales) | Más cercano a 1 = módulos bien cohesionados, más estables. |

## 6.2 Calidad Externa

Evaluación durante la ejecución del software en entorno controlado

### 6.2.1 Característica: Usabilidad

| Sub característica | Métrica | Propósito | Formula | Interpretación |
| --- | --- | --- | --- | --- |
| Comprensibilidad | Integridad de descripción | ¿Proporción de funciones entendidas? | X = A/B  (A=funciones entendidas, B=funciones totales) | Más cercano a 1 = la mayoría de funciones bien entendidas. |
| Acceso a demostraciones | ¿Acceso a tutoriales/ayuda? | X = A/B  (A=demos accedidas, B=demos disponibles) | Más cercano a 1 = casi todas las ayudas/demos accesibles. |
| Operabilidad | Cancelabilidad | ¿Efectividad de cancelación? | X = A/B  (A=funciones cancelables, B=que requieren cancelación) | Más cercano a 1 = funciones cancelables cuando se requiere. |
| Personalización | ¿Opciones de ajuste disponibles? | X = A/B  (A=funciones personalizables, B=que requieren personalización) | Más cercano a 1 = opciones de ajuste disponibles según necesidad. |
| Atractivo | Interacción atractiva | ¿Qué tan atractiva es la interfaz? | Puntuación promedio (escala 1-10) | Puntuación más alta = interfaz más agradable. |
| Personalización de interfaz | ¿Elementos personalizables? | X = A/B  (A=elementos personalizados OK, B=deseados personalizar) | Más cercano a 1 = mayoría de elementos personalizables según lo esperado. |

### 6.2.2 Característica: Funcionalidad

| Sub característica | Métrica | Propósito | Formula | Interpretación |
| --- | --- | --- | --- | --- |
| Idoneidad | Adecuación funcional | ¿Funciones adecuadas en ejecución? | X = 1 - A/B  (A=funciones con problemas, B=evaluadas) | Más cercano a 1 = pocas funciones con problemas. |
| Exhaustividad funcional | ¿Implementación completa? | X = 1 - A/B  (A=funciones faltantes, B=en requisitos) | Más cercano a 1 = casi todas las funciones requeridas implementadas. |
| Precisión | Exactitud computacional | ¿Frecuencia de cálculos inexactos? | X = A/T  (A=cálculos inexactos, T=tiempo) | Más bajo = menos cálculos inexactos. |
| Precisión de resultados | ¿Frecuencia de precisión inadecuada? | X = A/T  (A=resultados imprecisos, T=tiempo) | Más bajo = menos resultados imprecisos. |
| Interoperabilidad | Intercambio por formato | ¿Corrección en intercambio de datos? | X = A/B  (A=intercambios exitosos, B=total formatos) | Más cercano a 1 = la mayoría de intercambios correctos. |
| Intercambio general | ¿Frecuencia de fallas en intercambio? | X = 1 - A/B  (A=casos fallidos, B=intentados) | Más cercano a 1 = pocas fallas en intercambio. |

## 6.3 Calidad En Uso

Evaluación en contextos reales de operación

### 6.3.1 Característica: Eficacia

| Métrica | Propósito | Formula | Interpretación |
| --- | --- | --- | --- |
| Tasa de éxito en tareas | Medir cuántas tareas se completan correctamente. | (Tareas completadas con éxito ÷ Tareas intentadas) × 100 | Más alto = más tareas logradas con éxito. |
| Precisión de resultados | Medir cuán correctos son los resultados obtenidos. | (Resultados correctos ÷ Resultados esperados) × 100 | Más alto = resultados más exactos. |

### 6.3.2 Característica: Productividad

| Métrica | Propósito | Formula | Interpretación |
| --- | --- | --- | --- |
| Tiempo medio de tarea | Medir el tiempo promedio que tardan los usuarios en completar una tarea. | Σ Tiempo por usuarios ÷ Número de usuarios | Más bajo = mayor productividad. |
| Eficiencia de interacción | Medir qué tan óptimos son los pasos realizados frente a los mínimos necesarios. | (Pasos realizados ÷ Pasos mínimos) × 100 | Más cercano a 100% = interacción más eficiente. |

### 6.3.3 Característica: Seguridad

| Métrica | Propósito | Formula | Interpretación |
| --- | --- | --- | --- |
| Tasa de errores prevenidos | Medir qué tantos errores fueron prevenidos frente a los detectados. | (Errores prevenidos ÷ Errores detectados) × 100 | Más alto = mejor prevención. |
| Impacto de fallos | Medir la proporción de datos afectados por fallos. | (Datos afectados ÷ Datos procesados) × 100 | Más bajo = menor impacto negativo. |

### 6.3.4 Característica: Satisfacción

| Métrica | Propósito | Formula | Interpretación |
| --- | --- | --- | --- |
| Nivel de satisfacción | Medir la valoración promedio de los usuarios. | Σ Puntuaciones encuestas ÷ Número de encuestas | Más alto = mayor satisfacción. |
| Tasa de recomendación | Medir cuántos usuarios recomendarían el sistema. | (Usuarios que recomiendan ÷ Usuarios totales) × 100 | Más alto = mayor aceptación. |

# 7. Niveles de puntuación de las métricas

Los niveles de puntuación permiten clasificar los resultados de las métricas en tres rangos:

* Excelente (0.9 – 1.0)
* Bueno (0.7 – 0.89)
* Aceptable (0.5 – 0.69)
* Deficiente (< 0.5)

## 7.1 Métricas de Calidad Interna

| **Métrica** | **Fórmula / Método** | **Nivel Excelente** | **Nivel Bueno** | **Nivel Aceptable** | **Deficiente** |
| --- | --- | --- | --- | --- | --- |
| **Adecuación funcional** | Funciones correctas / total funciones | =1 | ≥0.9 | ≥0.7 | <0.7 |
| **Exhaustividad funcional** | Casos cubiertos / total casos | =1 | ≥0.9 | ≥0.7 | <0.7 |
| **Precisión de datos internos** | Datos correctos / total datos | =1 | ≥0.9 | ≥0.7 | <0.7 |
| **Exactitud de cálculos internos** | 1 – (errores / total pruebas) | X ≤ 0.05 | 0.05 < X ≤ 0.1 | 0.1 < X ≤ 0.2 | X > 0.2 |
| **Cumplimiento de estándares** | Código válido / total revisado | ≥0.9 | ≥0.7 | ≥0.5 | <0.5 |
| **Completitud de formatos** | Formularios correctos / total formularios | =1 | ≥0.9 | ≥0.7 | <0.7 |
| **Ratio de comentarios** | Líneas comentadas / líneas totales | ≥0.3 | ≥0.2 | ≥0.1 | <0.1 |
| **Complejidad ciclomática** | Nº decisiones por función | ≤5 | ≤10 | ≤15 | >15 |
| **Modularidad** | Funciones reutilizables / total funciones | ≥0.9 | ≥0.7 | ≥0.5 | 0.5 |
| **Tamaño de funciones** | Nº líneas por función | =1 | ≥0.9 | ≥0.7 | <0.7 |
| **Estabilidad** | Cambios no | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |

# 

## 7.2 Métricas de Calidad Externa

# 

| Métrica | Método | Nivel Excelente | Nivel Bueno | Nivel Aceptable | Deficiente |
| --- | --- | --- | --- | --- | --- |
| Adecuación funcional | Funciones correctas / total funciones evaluadas | =1 | ≥0.8 | ≥0.7 | <0.7 |
| Exhaustividad funcional | Casos cubiertos / total casos especificados | =1 | ≥0.9 | ≥0.7 | <0.7 |
| Exactitud computacional | Resultados correctos / total pruebas | =1 | ≥0.9 | ≥0.7 | <0.7 |
| Precisión de resultados | Resultados esperados / totales | =1 | ≥0.9 | ≥0.7 | <0.7 |
| Intercambio por formato | Formatos compatibles / totales | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |
| Intercambio general | Exportaciones correctas / intentos | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |
| Integridad de descripción | Documentación completa / total módulos | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |
| Acceso a demostraciones | Nº demos accesibles / total funciones | =1 | ≥0.9 | ≥0.7 | <0.7 |
| Cancelabilidad | Nº operaciones cancelables / total operaciones | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |
| Personalización | Funciones configurables / totales | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |
| Interacción atractiva | Encuesta ≥80% positivo | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |
| Personalización de interfaz | Opciones disponibles / esperadas | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |

# 

# 

## 7.3 Métricas de Calidad en Uso

| Métrica | Método | Nivel Excelente | Nivel Bueno | Nivel Aceptable | Deficiente |
| --- | --- | --- | --- | --- | --- |
| Tasa de éxito en tareas | Tareas completadas / intentos | ≥0.9 | ≥0.8 | ≥0.7 | <0.6 |
| Precisión de resultados | Resultados correctos / total | =1 | ≥0.9 | ≥0.7 | <0.7 |
| Tiempo medio de tarea | Tiempo en min | ≤2 | ≤3 | ≤4 | >4 |
| Eficiencia de interacción | Nº clics/pasos | =1 | ≥0.9 | ≥0.7 | <0.7 |
| Tasa de errores prevenidos | Errores detectados / totales | ≥0.9 | ≥0.8 | ≥0.7 | <0.7 |
| Impacto de fallos | Fallos críticos / totales | =0 | ≤0.1 | ≤0.2 | >0.2 |
| Nivel de satisfacción | Encuesta usuarios | =5 | ≥4 | ≥3 | <1.25 |
| Tasa de recomendacion(NPS) | NPS = No recomendado N° de usuarios evaluados | =1 | >= 0.9 | >=0.7 | <0.7 |

# 8. Criterios de evaluación

## 8.1 Calidad Interna

### 8.1.1 Escenario

Propósito: Evaluar atributos técnicos del software Aimeos durante el desarrollo en Laravel, comprobando la calidad de su código y la corrección de sus funciones principales.

Entorno:

* Sistema operativo: Windows 11
* Editor: Visual Studio Code
* Servidor local: XAMPP (PHP 8.x, MySQL)
* Proyecto: Aimeos Laravel (última versión estable)

Se trabajó directamente sobre el código fuente (controladores como ProfileController.php, RegisteredUserController.php, AuthenticatedSessionController.php, etc.) y se revisaron métricas de calidad interna mediante análisis manual y pruebas locales.

### 8.1.2 Herramientas de Medición

| Métrica | Herramientas | Método |
| --- | --- | --- |
| Adecuación funcional | Revisión de código en controladore | Verificar que las rutas de edición de perfil, autenticación, registro y contraseña funcionan correctamente |
| Exhaustividad funcional | Revisión de código en controladores | Comprobar que cada función cubre todos los casos esperados |
| Precisión de datos internos | base de datos | Validar que los datos almacenados/recuperados corresponden a lo solicitado |
| Exactitud de cálculos internos | Pruebas manuales (checkout, carrito) | Validar que totales, subtotales y descuentos se calculan bien |
| Cumplimiento de estándares | SonarQube | Analizar el código con SonarQube para ver reglas de estilo, convenciones PSR y seguridad.l |
| Completitud de formatos | Revisión de formularios (login, registro, perfil) | Verificar campos obligatorios y validaciones |
| Ratio de comentarios | SonarQube | SonarQube calcula % de líneas comentadas vs líneas de código. Evaluar si cumple el mínimo aceptable. |
| Complejidad ciclomática | SonarQube | SonarQube analiza cada función/método y reporta la complejidad ciclomática, alertando si supera el umbral |
| Modularidad | SonarQube + revisión manual | Usar métricas de acoplamiento y cohesión que SonarQube da para inferir modularidad. Revisar dependencias entre clases/módulos. |
| Tamaño de funciones | SonarQube | SonarQube muestra líneas de código por función y marca las que superan los límites recomendados. |
| Estabilidad de interfaces | PHPDocumentor | Verificar que cambios no rompen flujos principales |
| Cohesión de módulos | SonarQube | Usar métricas de cohesión (LCOM, acoplamiento) para evaluar si cada módulo hace una sola cosa y está bien encapsulado. |

### 8.1.3 Criterios de Aceptación

* Los criterios mínimos definidos para aceptación de calidad interna fueron:
* Adecuación funcional = 100%
* Exhaustividad funcional ≥ 0.9
* Exactitud de cálculos = 1
* Cumplimiento de estándares ≥ 0.7
* Complejidad ciclomática ≤ 10 por función
* Cohesión de módulos = Alta (clases enfocadas en una sola responsabilidad)

## 8.2 Calidad Externa

### 8.2.1 Escenario

Propósito: Evaluar cómo el software Aimeos responde en ejecución frente al usuario, asegurando que la salida generada sea correcta, comprensible y atractiva.

Entorno:

Las pruebas se realizaron sobre la aplicación desplegada en Laravel, utilizando navegadores modernos (Chrome, Edge, Firefox) y escenarios de compra reales:

* Escenario 1: Cliente busca un producto y realiza el checkout.
* Escenario 2: Cliente cancela un pedido antes del pago.
* Escenario 3: Cliente utiliza filtros de categoría y personaliza la búsqueda.

### 8.2.2 Herramientas de Medición

| Métrica (ISO/IEC 9126) | Herramientas | Método |
| --- | --- | --- |
| Exactitud computacional | Simulación de usuario | Comparación de cálculos (subtotal, impuestos, descuentos) con valores esperados |
| Precisión de resultados | Simulación de usuario | Validar que la información presentada coincide con la registrada en MySQL |
| Intercambio por formato | Exportación CSV/JSON nativa de Aimeos | Verificación de que los datos exportados mantienen la estructura esperada |
| Intercambio general | APIs REST de Aimeos | Pruebas con clientes externos (Postman) para confirmar compatibilidad de formatos |
| Integridad de descripción | Revisión manual | Verificar que los productos muestran nombre, precio, descripción e imagen completa |
| Cancelabilidad | Simulación de usuario | Simular cancelación de pedidos y verificar reversión en la BD |
| Personalización | Configuración de plantillas de Aimeos | Validar que se pueda adaptar colores, tipografía y layout del frontend |
| Interacción atractiva | Cuestionario de usuarios, revisión manual | Evaluar alineación, colores, tipografía, espacios y usabilidad visual |

### 8.2.3 Criterios de Aceptación

Se establecieron los siguientes valores mínimos para la aceptación de la calidad externa:

* Exactitud computacional de cálculos en carrito ≥ 0.95
* Exportación de datos sin pérdida en 100% de los casos
* Descripción completa de productos en ≥ 90%
* Cancelación de pedidos sin errores en 100% de pruebas
* Al menos 70% de satisfacción en cuestionario de interfaz atractiva

## 8.3 Calidad en Uso

### 8.3.1 Escenario

Propósito: Evaluar la efectividad, eficiencia, seguridad y satisfacción de los usuarios finales al interactuar con Aimeos en un flujo real de e-commerce.

Entorno de prueba:

* Plataforma: Aimeos + Laravel en Windows y Ubuntu.
* Navegadores: Chrome, Firefox, Edge.
* Escenarios de usuario:
  + Escenario 1: Buscar un producto y añadirlo al carrito.
  + Escenario 2: Finalizar una compra con pago simulado.
  + Escenario 3: Cancelar una compra antes de confirmar.
  + Escenario 4: Personalizar filtros y navegación (categorías, rangos de precios).

### 8.3.2 Herramientas de Medición

| Métrica (ISO/IEC 9126) | Herramientas | Método |
| --- | --- | --- |
| Tasa de éxito en tareas | Laravel Dusk, pruebas manuales | Registrar nº de tareas completadas con éxito vs. intentos totales |
| Precisión de resultados | Observación directa, validación BD | Validar que la acción realizada (ej. compra) corresponde con lo reflejado en la base de datos |
| Tiempo medio de tarea | Cronómetro, Dusk | Medir el tiempo en segundos/minutos para completar checkout o cancelar pedido |
| Eficiencia de interacción | Observación de clics, grabación de sesiones | Contar nº de pasos/clics necesarios para terminar una acción |
| Tasa de errores prevenidos | Dusk, pruebas manuales | Intentar errores comunes (ej. campos vacíos, stock insuficiente) y verificar mensajes preventivos |
| Impacto de fallos | Registro de logs de Laravel, observación de usuarios | Identificar si un fallo impide o no completar la tarea |
| Nivel de satisfacción | Encuestas a usuarios | Valoración subjetiva de facilidad de uso y experiencia general |
| Tasa de recomendación | Encuestas (NPS) | Pregunta directa: “¿Recomendarías Aimeos a otra persona?” |

### 8.3.3 Criterios de Aceptación

Se definen los valores mínimos para aceptación de la calidad en uso:

Tasa de éxito en tareas ≥ 0.85

Tiempo medio de tarea ≤ 3 min

Tasa de errores prevenidos ≥ 0.75

Nivel de satisfacción ≥ 70%

Tasa de recomendación (NPS) ≥ 0

# 9. Plan de evaluación

El presente plan tiene como objetivo realizar un análisis integral y sistemático del software Aimeos Laravel, evaluando sus características técnicas, funcionales y de usabilidad. La evaluación permitirá identificar fortalezas, debilidades y oportunidades de mejora, considerando tanto métricas internas de calidad como la experiencia de usuarios reales.

El proceso se estructura en cuatro fases fundamentales:

* Revisión documental y técnica
* Evaluación de métricas internas
* Evaluación de calidad externa
* Evaluación de métricas de uso

Cada fase examina aspectos específicos del software: desde su arquitectura modular y código en Laravel hasta la experiencia final del usuario en tareas de e-commerce.

## 9.1 Desglose Detallado de Tareas de Evaluación

### 9.1.1 Revisión Documental y Técnica

**Objetivo:** Analizar la documentación, especificaciones técnicas y estructura del software Aimeos.

| ID | Tarea | Fecha Inicio | Fecha Fin | Duración (días) |
| --- | --- | --- | --- | --- |
| T01 | Revisión de documentación del software (README, Wiki, Docs) | 16-ago-2025 | 17-ago-2025 | 2 |
| T02 | Revisión del lenguaje de programación (PHP/Laravel) | 18-ago-2025 | 20-ago-2025 | 3 |
| T03 | Revisión de arquitectura modular y paquetes de Aimeos | 21-ago-2025 | 22-ago-2025 | 2 |

### 9.1.2 Fase 2: Evaluación de Métricas Internas

**Objetivo:** Evaluar características intrínsecas del software que determinan su calidad técnica (funcionalidad, mantenibilidad, portabilidad).

| ID | Tarea | Fecha Inicio | Fecha Fin | Duración (días) |
| --- | --- | --- | --- | --- |
| T04 | Evaluación de Mantenibilidad (Analizabilidad, Cambiabilidad, Estabilidad) | 23-ago-2025 | 25-ago-2025 | 3 |
| T05 | Evaluación de Funcionalidad Interna (Idoneidad, Precisión, Interoperabilidad) | 26-ago-2025 | 26-ago-2025 | 1 |
| T06 | Evaluación de Portabilidad y Adaptabilidad (HW/SW, Instalabilidad) | 27-ago-2025 | 28-ago-2025 | 2 |

### 9.1.3 Fase 3: Evaluación de Calidad Externa

**Objetivo:** Valorar la experiencia de usuario y la funcionalidad desde una perspectiva externa, incluyendo tareas de e-commerce.

| ID | Tarea | Fecha Inicio | Fecha Fin | Duración (días) |
| --- | --- | --- | --- | --- |
| T07 | Selección de usuarios externos (10 técnicos, 10 no técnicos) | 29-ago-2025 | 29-ago-2025 | 1 |
| T08 | Evaluación de Usabilidad (Comprensibilidad, Atractivo, Facilidad de Aprendizaje) | 30-ago-2025 | 03-sep-2025 | 5 |
| T09 | Evaluación de Funcionalidad Externa (Idoneidad, Precisión, Interoperabilidad) | 30-ago-2025 | 03-sep-2025 | 5 |

### 9.1.4 Fase 4: Evaluación de Métricas de Uso

**Objetivo:** Analizar el rendimiento, productividad y satisfacción del usuario en condiciones reales de uso.

| ID | Tarea | Fecha Inicio | Fecha Fin | Duración (días) |
| --- | --- | --- | --- | --- |
| T10 | Evaluación de Métricas de Uso (Eficacia, Productividad, Seguridad, Satisfacción) | 04-sep-2025 | 09-sep-2025 | 6 |

Se recopilaron datos reales de uso en entornos locales y producción: tasa de éxito en tareas de e-commerce, tiempo de respuesta de la aplicación, errores críticos y feedback de usuarios.

# 10. Ejecución del plan de evaluación

## 10.1. Calidad interna

### 10.1.1 Adecuación funcional. -

| Nº | Función evaluada | Requisito esperado | Resultado de la prueba (OK/Falla) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Registro de usuario | Permite crear cuenta con email válido y contraseña segura | OK | Usuario registrado correctamente |
| 2 | Inicio de sesión | Acceso con usuario y contraseña válidos | OK | Funciona en login normal |
| 3 | Carrito de compras | Agregar y eliminar productos sin errores | OK | Se actualiza el total automáticamente |
| 4 | Checkout | Permitir pago con PayPal u otro método de transferencia bancaria | OK | Funciona correctamente la compra |
| 5 | Facturación | Generar factura en PDF después de la compra | FALLA | Factura generada correctamente, pero no se puede imprimir o generar pdf |

**Cálculo de la métrica:**

B = Número total de funciones evaluadas = 5

A = Número de funciones con problemas = 1

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (1 / 5) = 0.8**

### 10.1.2 Exhaustividad funcional:

| Nº | Función esperada (según requisitos) | ¿Está implementada en Aimeos? (Sí/No) | ¿Funciona según requisitos? (Sí/No) | Observaciones | ¿Completa? (Sí/No) |
| --- | --- | --- | --- | --- | --- |
| 1 | Registro de usuario con validación de email | No | No | No cuenta con la opción de registrar usuarios nuevos | No |
| 2 | Inicio de sesión con recuperación de contraseña | Sí | Sí | Permite recuperar contraseña | Sí |
| 3 | Carrito de compras con actualización de stock | Sí | Sí | El stock se descuenta automáticamente | Sí |
| 4 | Checkout con PayPal y transferencia | Sí | No | PayPal no disponible | No |
| 5 | Facturación con envío automático por correo | No | — | No implementado en la versión actual | No |

**Cálculo de la métrica:**

B = Nº de funciones en requisitos = 5

A = Nº de funciones faltantes o incompletas = 3

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (3 / 5) = 0.4**

### 10.1.3 Exactitud computacional:

| Nº | Escenario probado | Datos de entrada | Resultado esperado | Resultado mostrado | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Carrito con producto único | Producto A (10 USD) × 3 | 30 USD | 30 USD | Sí | Correcto |
| 2 | Carrito con dos productos | A: 2×10 USD + B: 1×15 USD | 35 USD | 35 USD | Sí | Correcto |
| 3 | Impuesto 13% aplicado | Subtotal 100 USD | 113 USD | 113 USD | Sí | Precisión correcta |
| 4 | Descuento 10% aplicado | Subtotal 50 USD | 45 USD | 45 USD | Sí | Correcto |

**Cálculo de la métrica:**

B = Nº de pruebas realizadas = 5

A = Nº de errores encontrados = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 5) = 1**

### 10.1.4 Precisión de resultados:

| Nº | Escenario probado | Entrada / Acción | Resultado esperado | Resultado mostrado | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Ver carrito | Añadir 2×Producto A (10 USD) | Subtotal 20 USD | 20 USD | Sí | Correcto |
| 2 | Aplicar descuento | 10% sobre subtotal 50 USD | 45 USD | 45 USD | Sí | Correcto |
| 3 | Ver total con impuestos | Subtotal 100 USD, IVA 13% | 113 USD | 113 USD | Sí | Correcto |
| 4 | Confirmación pedido | 2 productos, total 65 USD | Total confirmado = 65 USD | 65 USD | Sí | Correcto |
| 5 | Ver stock disponible | Producto B stock 5 unidades | Mostrar 5 | 5 | Sí | Correcto |

**Cálculo de la métrica:**

B = Nº de resultados revisados = 5

A = Nº de resultados inexactos = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 5) = 1**

### 10.1.5 Intercambio por formato:

| Nº | Escenario probado | Formato | Datos enviados / recibidos | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- |
| 1 | Exportación de pedidos | CSV | Pedidos exportados con columnas: id, cliente, total, estado, fecha | Si | CSV generado correctamente y legible en Excel/LibreOffice |
| 2 | Exportación de productos | JSON | Productos exportados con campos: id, nombre, precio, stock | Si | JSON bien formado y todos los campos presentes |
| 3 | Importación de productos | CSV | Productos importados desde CSV con columnas: id, nombre, precio, stock | Si | Todos los productos importados correctamente sin errores de formato |

**Cálculo de la métrica:**

B = Nº de intercambios revisados = 4

A = Nº de intercambios fallidos = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.1.6 Intercambio general:

| Nº | Escenario probado | Acción | Resultado esperado | Resultado obtenido | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Pago con PayPal (sandbox) | Realizar una compra de prueba y enviar los datos al sandbox de PayPal | Pedido procesado y pago aprobado recibido desde PayPal | Pedido procesado y pago aprobado recibido desde PayPal | Sí | Prueba usando pasarela de pago integrada de Aimeos |
| 2 | Exportación de pedidos | Exportar pedidos desde el backend en CSV/JSON | Archivo generado con todos los datos correctos de pedidos | Archivo generado correctamente y todos los datos son correctos | Sí | Exportación funciona con los formatos de Aimeos sin errores |
| 3 | API REST de productos | **Realizar GET a /jsonapi/product** | Se devuelve JSON con todos los productos | JSON devuelto correctamente con todos los productos | Sí | Prueba usando la API JSON:API de Aimeos |

**Cálculo de la métrica:**

B = Nº de intercambios revisados = 3

A = Nº de intercambios fallados = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 3) = 1**

### 10.1.7 Integridad de descripción:

| Nº | Elemento | Descripción mostrada | Completa / Clara (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Producto A | Nombre, precio, descripción detallada, stock | Sí | Correcta, fácil de entender |
| 2 | Producto B | Nombre, precio, descripción corta | Sí | Cumple, aunque se podría ampliar |
| 3 | Formulario registro | Instrucciones de campos obligatorios | Sí | Claro y completo |
| 4 | Formulario perfil | Explicación de campos editables | Sí | Completo y entendible |

**Cálculo de la métrica:**

B = Nº de elementos revisados = 4

A = Nº de descripciones incompletas o confusas = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.1.8 Acceso a demostraciones:

| Nº | Demostración / Tutorial | Accesible (Sí/No) | Observaciones |
| --- | --- | --- | --- |
| 1 | Tutorial registro usuario | No aplicable | Aimeos base no incluye tutoriales ni guías interactivas |
| 2 | Demo añadir productos | No aplicable | Aimeos base no incluye demostraciones de producto |
| 3 | Video flujo de compra | No aplicable | Aimeos base no incluye videos de demostración |
| 4 | Tutorial administración pedidos | No aplicable | Aimeos base no incluye tutoriales de administración |

**Cálculo de la métrica:**

B = Nº de demostraciones revisadas = 4

A = Nº de demostraciones inaccesibles = 4

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (4 / 4) = 0**

### 10.1.9 Cancelabilidad:

| Nº | Función | Acción de cancelación | Cancelación exitosa (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Cancelar pedido | Usuario cancela pedido antes de confirmar | Sí | Correcto, pedido no se procesa |
| 2 | Cancelar edición de perfil | Usuario presiona “Cancelar” en formulario | Sí | Los cambios no se guardan |
| 3 | Cancelar carrito | Usuario elimina productos antes de checkout | Sí | Correcto, carrito actualizado |
| 4 | Cancelar proceso de actualización stock | Usuario detiene proceso en admin | Sí | Correcto, proceso detenido |

**Cálculo de la métrica:**

B = Nº de funciones revisadas = 4

A = Nº de funciones que no se cancelan correctamente = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.1.10 Personalización:

| Nº | Función | Parámetros personalizables | Personalización exitosa (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Perfil usuario | Nombre, email, contraseña | Sí | Se actualiza correctamente |
| 2 | Configuración tienda | Idioma, moneda | Sí | Cambios se aplican en toda la sesión |
| 3 | Preferencias carrito | Mostrar cantidad por página | Sí | Correcto, configuración guardada |
| 4 | Notificaciones | Activar/desactivar alertas | Sí | Cambios reflejados correctamente |

**Cálculo de la métrica:**

B = Nº de funciones revisadas = 4

A = Nº de funciones que no permiten personalización = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.1.11 Interacción atractiva:

| Nº | Elemento | Aspectos evaluados | Evaluación usuario (1-5) | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- |
| 1 | Home tienda | Diseño, colores, organización | 5 | Sí | Muy agradable visualmente |
| 2 | Catálogo productos | Imágenes, disposición, botones | 4 | Sí | Correcto, se podría mejorar animación |
| 3 | Carrito | Botones, claridad de totales | 5 | Sí | Interfaz clara y funcional |
| 4 | Formulario login | Tipografía, alineación | 5 | Sí | Correcto, atractivo y simple |
| 5 | Panel administración | Menús, organización | 4 | Sí | Funcional y ordenado |

**Cálculo de la métrica:**

B = Nº de elementos evaluados = 5

A = Nº de elementos poco atractivos = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 5) = 1**

### 10.1.12 Personalización de interfaz

| Nº | Elemento | Opciones de personalización | Personalización exitosa (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Tema colores | Claro / Oscuro | Sí | Cambia correctamente en toda la tienda |
| 2 | Catálogo | Ordenar columnas, cantidad por página | Sí | Funciona correctamente |
| 3 | Idioma | Español / Inglés | Sí | Se aplica a toda la interfaz |
| 4 | Moneda | USD / EUR | Sí | Correcto, total y precios se actualizan |
| 5 | Preferencias de visualización | Mostrar/ocultar detalles del producto | Sí | Funciona correctamente |

**Cálculo de la métrica:**

B = Nº de elementos revisados = 5

A = Nº de elementos que no permiten personalización = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 5) = 1**

## 10.2 Calidad interna

### 10.2.1 Adecuación funcional.-

| Nº | Ruta / Archivo | Función revisada (método) | Requisito técnico esperado | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- |
| 1 | myshop\app\  Http\Controllers\  ProfileController.php | edit(), update(), destroy() | Permitir edición, actualización y eliminación de perfil de usuario | Sí | Implementa edición y actualización, incluye destroy correctamente |
| 2 | myshop\app\Http\  Controllers\Auth  \RegisteredUserController.php | create(), store() | Registrar usuario con validación, hash de contraseña y evento Registered | Sí | Usa validaciones, hash y login automático tras registro |
| 3 | myshop\app\Http\  Controllers\Auth\  NewPasswordController.php | create(), store() | Permitir restablecer contraseña mediante token seguro y actualización en DB | Sí | Valida token, email y genera nuevo password hash |
| 4 | myshop\app\Http\  Controllers\Auth\  AuthenticatedSession  Controller.php | create(), store(), destroy() | Autenticación de usuarios (login/logout) con regeneración de sesión segura | Sí | Implementa login, cierre de sesión y protección de sesión |

**Cálculo de la métrica:**

B = Nº de funciones revisadas = 4

A = Nº de funciones inadecuadas = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.2.2 Exhaustividad funcional:

| Nº | Ruta / Archivo | Función revisada (método) | Requisito funcional esperado | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- |
| 1 | RegisteredUserController.php | create(), store() | Registro de usuarios completo (formulario + guardar en DB) | Sí | Implementa formulario de registro y almacenamiento |
| 2 | AuthenticatedSessionController.php | create(), store(), destroy() | Autenticación completa (login + logout) | Sí | Cubre inicio y cierre de sesión |
| 3 | NewPasswordController.php | create(), store() | Restablecimiento de contraseña (formulario + actualización) | Sí | Cubre flujo completo de reseteo |
| 4 | ProfileController.php | edit(), update(), destroy() | Gestión de perfil (editar, actualizar, eliminar cuenta) | Sí | CRUD completo para el perfil |

**Cálculo de la métrica:**

B = Nº de funciones revisadas = 4

A = Nº de validaciones inadecuadas o faltantes = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.2.3 Precisión de datos internos:

| Nº | Ruta / Archivo | Función revisada (método) | Datos críticos | Validación / Precisión | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | RegisteredUserController.php → store() | Registro de usuario | name, email, password | Valida email único, longitud mínima de contraseña, hash en DB | Sí | Uso de Hash::make asegura precisión en password |
| 2 | NewPasswordController.php → store() | Restablecer contraseña | token, email, password | Verifica token, email y valida password antes de actualizar | Sí | Datos internos consistentes |
| 3 | AuthenticatedSessionController.php → store() | Login usuario | email, password | Valida credenciales contra DB, mantiene sesión regenerada | Sí | Precisión alta en autenticación |
| 4 | ProfileController.php → update() | Actualización de perfil | name, email, password | Valida datos de entrada, evita duplicidad de email | Sí | Precisión en datos personales |

**Cálculo de la métrica:**

B = Nº de funciones revisadas = 4

A = Nº de validaciones inadecuadas o faltantes = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.2.4 Exactitud de cálculos internos:

| Nº | Escenario probado | Datos de entrada | Cálculo esperado | Cálculo obtenido (sistema) | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Cálculo de subtotal | Producto A (10 USD) × 2 unidades | 20 USD | 20 USD | Sí | Correcto |
| 2 | Aplicación de descuento | Subtotal 100 USD, descuento 10% | 90 USD | 90 USD | Sí | Correcto |
| 3 | Cálculo de impuestos (IVA 13%) | Subtotal 200 USD | 226 USD | 226 USD | Sí | Correcto |
| 4 | Total general con múltiples productos | A: 2×10 USD + B: 3×15 USD = 65 USD, IVA 13% → 73.45 USD | 73.45 USD | 73.45 USD | Sí | Correcto |

**Cálculo de la métrica:**

T = Nº total de cálculos probados = 4

A = Nº de cálculos inexactos = 0

Fórmula: X =A /T

**Resultado:X=0/4=0.**

### 10.2.5 Cumplimiento de estándares:

| Nº | Archivo/Ruta | Estándar verificado | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | app\Http\Controllers\ProfileController.php | Sigue PSR-12 (nombres de clases, namespaces, sangría) | Sí | Código ordenado |
| 2 | app\Http\Controllers\Auth\RegisteredUserController.php | Validaciones correctas en check() | Sí | Cumple |
| 3 | app\Http\Controllers\Auth\NewPasswordController.php | Uso de Password::reset() según Laravel | Sí | Correcto |
| 4 | app\Http\Controllers\Auth\AuthenticatedSessionController.php | Manejo de sesiones y logout con Auth::guard('web') | Sí | Correcto |

**Cálculo de la métrica:**

B = Nº de funciones revisadas = 4

A = Nº de validaciones inadecuadas o faltantes = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.2.6 Completitud de formatos:

| Nº | Formulario/Vista | Campos requeridos | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | auth.register | Nombre, email, contraseña, confirmación | Sí | Todos presentes |
| 2 | auth.login | Email, contraseña | Sí | Correcto |
| 3 | profile.edit | Nombre, email, contraseña opcional | Sí | Completo |
| 4 | auth.reset-password | Email, token, nueva contraseña, confirmación | Sí | Completo |

**Cálculo de la métrica:**

B = Nº de formularios revisados = 4

A = Nº de formularios incompletos = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.2.7 Ratio de comentarios:

| Archivo | Total de líneas | Líneas comentadas | Ratio |
| --- | --- | --- | --- |
| ProfileController.php | 60 | 10 | 0.17 |
| RegisteredUserController.php | 135 | 32 | 0.24 |
| NewPasswordController.php | 61 | 14 | 0.23 |
| AuthenticatedSessionController.php | 49 | 9 | 0.18 |

**Cálculo de la métrica:**

B = Total líneas = 305

C = Líneas comentadas = 65

Fórmula: X = C / B

**Resultado: X = 65 / 305 ≈ 0.21**

### 10.2.8 Complejidad ciclomática:

| Archivo | Función | Complejidad |
| --- | --- | --- |
| ProfileController | update() | 2 |
| RegisteredUserController | store() | 4 |
| NewPasswordController | store() | 3 |
| AuthenticatedSessionController | store() | 3 |

**Cálculo de la métrica:**

**B = Nº total de funciones revisadas = 4**

Suma de complejidad = 12

Promedio = 12 / 4 = 3

**Resultado: Complejidad baja (≤10)**

### 10.2.9 Modularidad:

| Módulo | Rutas | Controladores | Independencia |
| --- | --- | --- | --- |
| Autenticación | login, register, logout | Auth controllers | Alta |
| Perfil | profile/edit/update | ProfileController | Alta |
| Catálogo (Aimeos) | productos, categorías | Aimeos controllers | Alta |
| Órdenes | checkout, pedidos | Order controllers | Media |

**Cálculo de la métrica:**

B = Nº de módulos = 4

A = Módulos acoplados = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.2.10 Tamaño de funciones:

| Archivo | Función | Nº líneas | Evaluación |
| --- | --- | --- | --- |
| ProfileController | update() | 20 | Adecuado |
| RegisteredUserController | store() | 40 | Límite aceptable |
| NewPasswordController | store() | 35 | Adecuado |
| AuthenticatedSessionController | store() | 25 | Adecuado |

**Cálculo de la métrica:**

B = Nº de funciones revisadas = 4

A = Funciones que superan 50 líneas = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.2.11 Estabilidad de interfaces:

| Interfaz | Última revisión | Cambios recientes | Estabilidad |
| --- | --- | --- | --- |
| Login | 2025 | Sin cambios | Alta |
| Registro | 2025 | Sin cambios | Alta |
| Perfil | 2025 | Sin cambios | Alta |
| Reset password | 2025 | Sin cambios | Alta |

**Cálculo de la métrica:**

B = Nº de interfaces revisadas = 4

A = Interfaces inestables = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

### 10.2.12 Cohesión de módulos:

| Módulo | Funciones | Cohesión |
| --- | --- | --- |
| Perfil | edit, update, destroy | Alta |
| Autenticación | login, register, logout, reset | Alta |
| Productos (Aimeos) | listar, detalle, añadir carrito | Alta |
| Pedidos | checkout, ver pedidos | Alta |

**Cálculo de la métrica:**

B = Nº de módulos revisados = 4

A = Módulos con baja cohesión = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 4) = 1**

## 10.3 Calidad de uso

### 10.3.1 Tasa de éxito en tareas:

| Nº | Tarea | Usuario | Completada correctamente (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Registro usuario | Usuario 1 | Sí | Correcto |
| 2 | Añadir producto al carrito | Usuario 1 | Sí | Correcto |
| 3 | Completar compra | Usuario 1 | Sí | Correcto |
| 4 | Actualizar perfil | Usuario 2 | Sí | Correcto |
| 5 | Registro usuario | Usuario 3 | No | No existe un botón para el registro de usuarios |

**Cálculo de la métrica:**

B = Nº de tareas realizadas = 5

A = Nº de tareas fallidas = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (1 / 5) = 0.8**

### 10.3.2 Precisión de resultados:

| Nº | Tarea realizada | Entrada / Acción | Resultado esperado | Resultado mostrado | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Añadir 2 productos A (10 USD) | Carrito | Subtotal 20 USD | 20 USD | Sí | Correcto |
| 2 | Aplicar descuento 10% | Carrito subtotal 50 USD | 45 USD | 45 USD | Sí | Correcto |
| 3 | Confirmar pedido | Pedido total 65 USD | 65 USD | 65 USD | Sí | Correcto |
| 4 | Ver stock Producto B | Stock 5 unidades | 5 | 5 | Sí | Correcto |

**Cálculo de la métrica:**

B = Nº de resultados revisados = 5

A = Nº de resultados inexactos = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (0 / 5) = 1**

### 10.3.3 Tiempo medio de tarea:

| Nº | Tarea | Usuario | Tiempo empleado (segundos) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Añadir producto al carrito | Usuario 1 | 12 | Correcto |
| 2 | Completar compra | Usuario 1 | 328 | Correcto |
| 3 | Actualizar perfil | Usuario 2 | 20 | Correcto |

**Cálculo de la métrica:**

Tiempos registrados (segundos) = 12, 328 ,20

Número de tareas = 3

Fórmula: Tiempo medio = (12 + 328 + 20) / 3

**Resultado: Tiempo medio = 360 / 3 = 120 segundos**

### 10.3.4 Eficiencia de interacción:

| Nº | Tarea | Usuario | Nº de acciones necesarias | Nº de acciones reales realizadas | Cumple (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Añadir producto al carrito | Usuario 1 | 3 | 3 | Sí | Correcto |
| 2 | Completar pedido | Usuario 2 | 5 | 5 | Sí | Correcto |
| 3 | Editar perfil | Usuario 3 | 4 | 4 | Sí | Correcto |
| 4 | Buscar producto | Usuario 1 | 2 | 2 | Sí | Correcto |
| 5 | Registro usuario | Usuario 2 | 4 | 4 | Sí | Correcto |

**Cálculo de la métrica:**

Acciones esperadas totales = 3 + 5 + 4 + 2 + 4 = 18

Acciones reales totales = 3 + 5 + 4 + 2 + 4 = 18

Fórmula: Eficiencia = Acciones esperadas / Acciones reales

**Resultado: Eficiencia = 18 / 18 = 1**

### 10.3.5 Tasa de errores prevenidos:

| Nº | Tarea | Usuario | Error potencial | Sistema previno error (Sí/No) | Observaciones |
| --- | --- | --- | --- | --- | --- |
| 1 | Añadir producto al carrito | Usuario 2 | Cantidad mayor que stock | No | No limita la cantidad de compra por stock |
| 2 | Checkout | Usuario 3 | Campos obligatorios vacíos | Sí | Correcto |
| 3 | Editar perfil | Usuario 1 | Correo inválido | Sí | Correcto |
| 4 | Borrar producto | Usuario 2 | Confusión sobre eliminación | Sí | Correcto |

**Cálculo de la métrica:**

B = Nº de errores potenciales = 5

A = Nº de errores no prevenidos = 0

Fórmula: X = 1 - A / B

**Resultado: X = 1 - (1 / 5) = 0.8**

### 10.3.6 Impacto de fallos:

| Nº | Tarea | Usuario | Fallo detectado | Impacto (Alto/Medio/Bajo) | Observaciones |
| --- | --- | --- | --- | --- | --- |
| 1 | Compra fuera de stock | Usuario 2 | El stock no controla la cantidad comprada | Alto | Puede afectar a la empresa |
| 2 | Visualizar stock | Usuario 3 | Stock no visualizable | Medio | Se debería agregar un campo para ver stock |
| 3 | Checkout | Usuario 2 | Error de límite de cupones | Medio | si se añaden demasiados el precio se pone en negativo |

**Cálculo de la métrica:**

B = Nº de fallos detectados = 3

A = Nº de fallos con impacto alto = 1

Fórmula: Impacto de fallos = A / B

**Resultado: Impacto de fallos = (1 / 3) = 0,2**

### 10.3.7 Nivel de satisfacción:

| Nº | Usuario | Tarea realizada | Puntuación satisfacción (1-5) | Observaciones |
| --- | --- | --- | --- | --- |
| 1 | Usuario 1 | Registro usuario | 1 | Nada satisfecho |
| 2 | Usuario 2 | Añadir producto al carrito | 5 | Satisfecho |
| 3 | Usuario 3 | Completar compra | 2 | Poco satisfecho |
| 4 | Usuario 1 | Editar perfil | 4 | Satisfecho |
| 5 | Usuario 2 | Buscar producto | 5 | Muy satisfecho |

**Cálculo de la métrica:**

B = Nº de usuarios evaluados = 5

Suma puntuaciones = 1+ 5 + 2 + 4 + 5 = 17

Fórmula: Nivel de satisfacción = Suma puntuaciones / B

**Resultado: Nivel de satisfacción = 17 / 5 = 3,4**

### 10.3.8 Tasa de recomendación:

| Nº | Usuario | ¿Recomendaría Aimeos? (Sí/No) | Observaciones |
| --- | --- | --- | --- |
| 1 | Usuario 1 | Sí | Muy satisfecho con la compra |
| 2 | Usuario 2 | No | No me gusta rellenar todo el tiempo mi dirección |
| 3 | Usuario 3 | No | No se puede guardar mi dirección |
| 4 | Usuario 4 | Sí | Perfil fácil de usar |
| 5 | Usuario 5 | Sí | Recomendable para tiendas pequeñas |

**Cálculo de la métrica:**

B = Nº de usuarios evaluados = 5

A = Nº de usuarios que no recomendarían = 2

Fórmula: Tasa de recomendación = 1 - A / B

**Resultado: Tasa de recomendación = 1 - (2 / 5) = 0,6**

**11. Resultado de las métricas comparativas**

## 11.1 Calidad interna:

| Métrica | Resultado esperado / aceptable | Resultado obtenido | Rango |
| --- | --- | --- | --- |
| Adecuación funcional | **X ≥ 0.9** | **1** | **Excelente** |
| Exhaustividad funcional | **X ≥ 0.9** | **1** | **Excelente** |
| Precisión de datos internos | **X ≥ 0.9** | **1** | **Excelente** |
| Exactitud de cálculos internos | **X <= 0.05** | **0** | **Excelente** |
| Cumplimiento de estándares | **X ≥ 0.7** | **1** | **Excelente** |
| Completitud de formatos | **X ≥ 0.9** | **1** | **Excelente** |
| Ratio de comentarios | **X ≥ 0.2** | **0.21** | **bueno** |
| Complejidad ciclomática | **X≤10** | **3** | **Excelente** |
| Modularidad | **X ≥ 0.9** | **1** | **Excelente** |
| Tamaño de funciones | **X ≥ 0.7** | **1** | **Excelente** |
| Estabilidad de interfaces | **X ≥ 0.9** | **1** | **Excelente** |
| Cohesión de módulos | **X ≥ 0.8** | **1** | **Excelente** |

## 11.2 Calidad externa:

| Métrica | Resultado esperado / aceptable | Resultado obtenido | Rango |
| --- | --- | --- | --- |
| Adecuación funcional | **X ≥ 0.9** | **0.8** | **Bueno** |
| Exhaustividad funcional | **X ≥ 0.9** | **0.4** | **Deficiente** |
| Exactitud computacional | **X ≥ 0.9** | **1** | **Excelente** |
| Precisión de resultados | **X ≥ 0.9** | **1** | **Excelente** |
| Intercambio por formato | **X ≥ 0.8** | **1** | **Excelente** |
| Intercambio general | **X ≥ 0.8** | **1** | **Excelente** |
| Integridad de descripción | **X ≥ 0.8** | **1** | **Excelente** |
| Acceso a demostraciones | **X ≥ 0.9** | **0** | **Deficiente** |
| Cancelabilidad | **X ≥ 0.8** | **1** | **Excelente** |
| Personalización | **X ≥ 0.8** | **1** | **Excelente** |
| Interacción atractiva | **X ≥ 0.8** | **1** | **Excelente** |
| Personalización de interfaz | **X ≥ 0.8** | **1** | **Excelente** |

## 11.3 Calidad de uso

| Métrica | Resultados esperados/aceptables | Resultados obtenidos | Rango |
| --- | --- | --- | --- |
| tasa de éxito en tareas | X ≥ 0.9 | 0.8 | **Bueno** |
| Precisión de resultados | X ≥ 0.9 | 1 | **Excelente** |
| Tiempo medio de tarea | ≤3 min | 2 min | **Excelente** |
| Eficiencia de interacción | X ≥ 0.9 | 1 | **Excelente** |
| Tasa de errores prevenidos | X ≥ 0.8 | 0.8 | **Bueno** |
| Impacto de fallos | X = 0 | 0.2 | **Bueno** |
| Nivel de satisfacción | =>4 | 3.4 | **Aceptable** |
| Tasa de recomendación (NPS) | X ≥ 0.8 | 0.6 | **Aceptable** |

# 12. Recomendaciones

Para mejorar la calidad del software y asegurar su confiabilidad como plataforma de e-commerce, se proponen las siguientes acciones:

## 12.1. Seguridad del sistema

* Implementar validación estricta en el backend para precios, cantidades y descuentos, evitando manipulaciones desde el frontend.
* Corregir la lógica de inventario para validar la disponibilidad de productos antes de añadirlos al carrito o finalizar la compra.

## 12.2. Funcionalidad crítica

* Desarrollar y activar la función de registro de usuarios.
* Corregir los errores en la integración con métodos de pago, especialmente PayPal.
* Mejorar la experiencia de uso en funciones avanzadas como la importación de archivos CSV.

## 12.3. Proceso de desarrollo y documentación

* Aumentar la cobertura de pruebas automatizadas, enfocándose en procesos clave como la compra, gestión de stock y aplicación de descuentos.
* Mejorar la documentación del código mediante el uso de comentarios y DocBlocks que expliquen claramente la lógica de negocio, facilitando el mantenimiento y escalabilidad del sistema.

# 13. Conclusión

El sistema Aimeos en su configuración base es sólido y confiable a nivel interno, mostrando buena organización del código y cumplimiento de estándares. Sin embargo, en el uso real por parte de los usuarios finales y en la cobertura de funcionalidades externas, existen limitaciones que afectan la experiencia, principalmente relacionadas con la configuración de pagos, la falta de registros y la ausencia de tutoriales.

Estas observaciones permiten identificar prioridades de mejora:

1. Configurar correctamente las pasarelas de pago y funcionalidades de facturación.
2. Incorporar registros de usuarios y control de stock limitado.
3. Desarrollar tutoriales, guías o demostraciones para mejorar la accesibilidad y la experiencia de los usuarios.

En conclusión, Aimeos presenta **una base técnica excelente**, pero necesita **ajustes y documentación adicional** para alcanzar un nivel óptimo de calidad externa y de uso.