# DOCUMENTACIÓN TÉCNICA: PROYECTO HARDINFINITY

Proyecto: Plataforma de Comercio Electrónico de Hardware

Versión: 1.0

Fecha: Diciembre 2025

## ÍNDICE

1. [Resumen Ejecutivo](https://www.google.com/search?q=%23i-resumen-ejecutivo)
2. [Introducción](https://www.google.com/search?q=%23ii-introducci%C3%B3n)
3. [Justificación del Proyecto](https://www.google.com/search?q=%23iii-justificaci%C3%B3n-del-proyecto)
4. [Objetivos](https://www.google.com/search?q=%23iv-objetivos)
5. [Alcance del Proyecto](https://www.google.com/search?q=%23v-alcance-del-proyecto)
6. [Identificación de Stakeholders](https://www.google.com/search?q=%23vi-identificaci%C3%B3n-de-stakeholders)
7. [Requerimientos Funcionales (Detallados)](https://www.google.com/search?q=%23vii-requerimientos-funcionales)
8. [Requerimientos No Funcionales](https://www.google.com/search?q=%23viii-requerimientos-no-funcionales)
9. [Historias de Usuario](https://www.google.com/search?q=%23ix-historias-de-usuario)
10. [Modelo de Datos (Diccionario Completo)](https://www.google.com/search?q=%23x-modelo-de-datos)
11. [Arquitectura del Sistema](https://www.google.com/search?q=%23xi-arquitectura-del-sistema)
12. [Plan de Implementación](https://www.google.com/search?q=%23xii-plan-de-implementaci%C3%B3n)

## I. RESUMEN EJECUTIVO

**HardInfinity** es una plataforma web de comercio electrónico diseñada específicamente para la comercialización de equipos de cómputo, componentes de hardware de alto rendimiento, periféricos y accesorios tecnológicos. A diferencia de las tiendas generalistas, HardInfinity se centra en ofrecer una experiencia técnica detallada, permitiendo a los usuarios filtrar productos por especificaciones precisas y verificar compatibilidades.

El sistema contempla un catálogo organizado jerárquicamente, un motor de búsqueda eficiente, un carrito de compras persistente, gestión de pedidos en tiempo real y un panel administrativo integral para la gestión de inventario, usuarios y transacciones. Además, como valor agregado innovador, integra un módulo de inteligencia artificial básico para recomendaciones personalizadas.

El presente documento detalla el análisis, diseño, requerimientos y arquitectura del software, sirviendo como hoja de ruta para el desarrollo del sistema utilizando tecnologías modernas como **Laravel** en el backend y **Vue.js** en el frontend.

## II. INTRODUCCIÓN

En la última década, el comercio electrónico se ha consolidado como el canal principal para la adquisición de tecnología. Sin embargo, el nicho de mercado enfocado en el ensamblaje de computadoras y hardware de alto rendimiento (Gaming, Workstations) enfrenta un problema común: la falta de información técnica detallada y la dificultad para encontrar componentes compatibles en plataformas de venta masiva.

Muchas tiendas online actuales presentan problemas de organización, lentitud en la carga, falta de claridad en las especificaciones técnicas o procesos de compra engorrosos. Ante esta problemática, nace **HardInfinity**, una propuesta de plataforma moderna, veloz y especializada.

El objetivo del proyecto no es solo crear una tienda online, sino desarrollar un sistema web robusto, mantenible y escalable que aplique buenas prácticas de ingeniería de software, patrones de diseño y una arquitectura limpia que garantice la seguridad y la eficiencia en cada transacción.

## III. JUSTIFICACIÓN DEL PROYECTO

### 3.1. Justificación Técnica

El desarrollo de HardInfinity permite la implementación de una arquitectura de **Monolito Modular Moderno** utilizando **Inertia.js**, lo que elimina la complejidad de gestionar una API REST separada mientras se mantiene la experiencia fluida de una *Single Page Application (SPA)*. Esto representa una oportunidad para aplicar conocimientos avanzados en frameworks PHP (Laravel) y JavaScript (Vue.js), así como diseño de bases de datos relacionales complejas.

### 3.2. Justificación Comercial

La demanda de hardware en Perú y Latinoamérica sigue en crecimiento. Los usuarios buscan confianza y especialización. Una plataforma que ofrezca fichas técnicas exactas, fotos reales y recomendaciones de compatibilidad (IA) tiene una ventaja competitiva significativa sobre marketplaces genéricos donde la información suele ser errónea o incompleta.

## IV. OBJETIVOS

### 4.1. Objetivo General

Diseñar y desarrollar una plataforma web de comercio electrónico ("HardInfinity") orientada a la venta especializada de equipos de cómputo y hardware, garantizando una experiencia de compra rápida, segura y eficiente mediante una arquitectura de software moderna y escalable.

### 4.2. Objetivos Específicos

1. **Catálogo Especializado:** Implementar un buscador avanzado y filtros dinámicos basados en especificaciones técnicas (ej: filtrar RAM por frecuencia y latencia).
2. **Gestión de Ventas:** Desarrollar un flujo de *checkout* seguro e intuitivo, integrando gestión de múltiples direcciones de envío y simulación de pasarelas de pago.
3. **Administración Total:** Construir un panel administrativo (Backend) que permita el control total del inventario (CRUD de productos), gestión de stock y trazabilidad de pedidos.
4. **Innovación con IA:** Integrar un sistema de recomendaciones basado en reglas o aprendizaje automático básico para sugerir productos complementarios (Cross-selling).
5. **Seguridad y Roles:** Garantizar la integridad de los datos mediante autenticación robusta, encriptación de contraseñas y gestión estricta de roles (Administrador vs Cliente).

## V. ALCANCE DEL PROYECTO

### 5.1. Alcance Funcional (Lo que SÍ incluye)

El sistema HardInfinity abarcará los siguientes módulos en su versión 1.0:

* **Módulo de Usuarios:** Registro de clientes, inicio de sesión seguro, recuperación de contraseñas, gestión de perfil y libreta de direcciones.
* **Módulo de Catálogo:** Categorización multinivel (Categorías y Subcategorías), gestión de marcas, galería de imágenes por producto y fichas técnicas.
* **Módulo de Transacciones:** Carrito de compras editable, resumen de orden, selección de método de pago y generación de pedidos.
* **Módulo Administrativo:** Dashboard de métricas básicas, gestión de productos, control de stock manual y cambio de estados de pedidos (Pendiente -> Pagado -> Enviado).
* **Módulo de IA:** Motor de recomendaciones en la vista de detalle del producto ("Productos similares" o "Comprados juntos habitualmente").

### 5.2. Exclusiones (Fuera del Alcance)

Las siguientes características no se desarrollarán en esta fase:

* Desarrollo de aplicaciones móviles nativas (Android/iOS).
* Integración en tiempo real con facturación electrónica SUNAT.
* Integración logística automatizada (API) con empresas de courier.
* Chatbots de soporte en tiempo real.

## VI. IDENTIFICACIÓN DE STAKEHOLDERS

| **Stakeholder** | **Rol** | **Descripción** |
| --- | --- | --- |
| **Usuario Final (Cliente)** | Comprador | Persona que navega el catálogo, se registra y realiza compras. |
| **Administrador** | Gestor | Encargado de mantener el catálogo, revisar el stock y gestionar los pedidos. |
| **Equipo de Desarrollo** | Desarrollador | Responsables de la arquitectura, codificación, pruebas y despliegue. |
| **Proveedor de Hosting** | Infraestructura | Servicio donde se alojará la base de datos y la aplicación web. |

## VII. REQUERIMIENTOS FUNCIONALES

### 7.1. Módulo de Autenticación y Usuarios

* **RF-01:** El sistema permitirá el registro de nuevos usuarios solicitando nombre, email y contraseña.
* **RF-02:** El sistema permitirá el inicio de sesión validando credenciales encriptadas.
* **RF-03:** El sistema permitirá la recuperación de contraseña vía enlace al correo electrónico.
* **RF-04:** El usuario podrá gestionar múltiples direcciones de envío en su perfil.

### 7.2. Módulo de Catálogo

* **RF-05:** El sistema mostrará un listado de productos con paginación y ordenamiento.
* **RF-06:** El sistema permitirá filtrar productos por categoría, marca, rango de precios y atributos técnicos.
* **RF-07:** El sistema mostrará el stock disponible en tiempo real para cada producto.
* **RF-08:** La ficha de producto incluirá una galería de imágenes interactiva.

### 7.3. Módulo de Ventas (Carrito y Checkout)

* **RF-09:** El usuario podrá agregar productos al carrito y modificar cantidades.
* **RF-10:** El sistema validará que la cantidad solicitada no exceda el stock disponible.
* **RF-11:** El proceso de checkout solicitará confirmación de dirección y método de pago.
* **RF-12:** El sistema generará un número de pedido único al finalizar la compra exitosamente.

### 7.4. Módulo Administrativo

* **RF-13:** El administrador podrá Crear, Leer, Actualizar y Eliminar (CRUD) productos.
* **RF-14:** El administrador podrá gestionar las categorías y marcas del sistema.
* **RF-15:** El administrador podrá ver el listado de todos los pedidos y filtrar por estado.
* **RF-16:** El administrador podrá cambiar el estado de un pedido (ej: de "Pagado" a "Enviado").

### 7.5. Módulo de Inteligencia Artificial

* **RF-17:** El sistema registrará las interacciones del usuario (productos vistos).
* **RF-18:** El sistema mostrará una sección de "Recomendados para ti" basada en la categoría o historial de navegación.

## VIII. REQUERIMIENTOS NO FUNCIONALES

* **RNF-01 Seguridad:** Las contraseñas se almacenarán usando algoritmos de hash robustos (Bcrypt).
* **RNF-02 Rendimiento:** El tiempo de carga de la página de inicio no debe exceder los 2 segundos.
* **RNF-03 Usabilidad:** La interfaz debe ser completamente responsiva (adaptable a Móviles, Tablets y Desktop).
* **RNF-04 Escalabilidad:** La arquitectura debe permitir agregar nuevos módulos sin reescribir el núcleo del sistema.
* **RNF-05 Infraestructura:** El sistema debe funcionar sobre servidores Linux con soporte para PHP 8.2+ y MySQL 8.0.

## IX. HISTORIAS DE USUARIO

| **ID** | **Rol** | **Historia** | **Criterio de Aceptación** |
| --- | --- | --- | --- |
| **HU-01** | Cliente | Como cliente, quiero filtrar las tarjetas de video por memoria VRAM para encontrar la que necesito. | El filtro debe actualizar la lista sin recargar la página completa. |
| **HU-02** | Cliente | Como cliente, quiero ver fotos detalladas del producto para asegurar su calidad. | Debe existir una galería con zoom o modal. |
| **HU-03** | Admin | Como administrador, quiero desactivar productos sin stock para que no se vendan por error. | Debe existir un switch de "Activo/Inactivo" en el panel. |
| **HU-04** | Admin | Como administrador, quiero ver los pedidos pendientes de envío para despacharlos hoy. | Debe haber una vista filtrada solo con estado "Pagado". |

## X. MODELO DE DATOS (DICCIONARIO DE DATOS)

El sistema utiliza una base de datos relacional MySQL con 13 tablas normalizadas.

### A. Gestión de Usuarios

1. **users**: Almacena credenciales y datos básicos.
   * id, name, email, password, role\_id, created\_at.
2. **roles**: Define los niveles de acceso.
   * id, name (Admin, Customer).
3. **user\_addresses**: Direcciones de envío del usuario.
   * id, user\_id, address\_line, city, zip\_code, is\_default.

### B. Inventario y Catálogo

1. **categories**: Jerarquía de productos.
   * id, name, slug, parent\_id.
2. **brands**: Fabricantes de hardware.
   * id, name, logo\_url.
3. **products**: Tabla principal del inventario.
   * id, brand\_id, category\_id, name, slug, description, price, stock, is\_active.
4. **product\_images**: Galería multimedia.
   * id, product\_id, image\_path, is\_primary.
5. **specifications**: Atributos técnicos dinámicos.
   * id, product\_id, spec\_name (ej: "Socket"), spec\_value (ej: "LGA1700").

### C. Ventas y Transacciones

1. **orders**: Cabecera del pedido.
   * id, user\_id, status (pending, paid, shipped, cancelled), total\_amount, shipping\_address, created\_at.
2. **order\_items**: Detalle de productos por pedido.
   * id, order\_id, product\_id, quantity, unit\_price.
3. **payment\_methods**: Métodos aceptados.
   * id, name, details.
4. **transactions**: Registro de intentos de pago.
   * id, order\_id, transaction\_code, status.

### D. Inteligencia Artificial

1. **user\_interactions**: Registro de comportamiento.
   * id, user\_id, product\_id, type (view, cart\_add), created\_at.

## XI. ARQUITECTURA DEL SISTEMA

### 11.1. Stack Tecnológico

* **Backend:** Laravel Framework (PHP 8.2+). Manejo de rutas, controladores, modelos, migraciones y seguridad.
* **Frontend:** Vue.js 3 (Composition API). Interfaz reactiva y moderna.
* **Estilos:** TailwindCSS. Diseño rápido y responsivo "Mobile First".
* **Capa de Enlace:** Inertia.js. Permite construir una SPA (Single Page Application) utilizando el enrutamiento y controladores clásicos de Laravel, eliminando la necesidad de una API REST compleja para el consumo interno.
* **Base de Datos:** MySQL 8.0.
* **Servidor Web:** Apache o Nginx.

### 11.2. Patrón de Diseño

Se utiliza el patrón **MVC (Modelo-Vista-Controlador)**:

* **Modelo:** Clases Eloquent ORM que representan las tablas de la BD.
* **Vista:** Componentes Vue (.vue) que renderizan la interfaz.
* **Controlador:** Lógica de negocio que recibe la petición HTTP, consulta al modelo y retorna la respuesta Inertia.

## XII. PLAN DE IMPLEMENTACIÓN

Para garantizar el éxito del proyecto, se seguirá el siguiente cronograma de fases:

1. **Fase 1: Configuración del Entorno**
   * Instalación de Laravel, configuración de .env, instalación de dependencias Node (Vue, Tailwind).
2. **Fase 2: Diseño de Base de Datos**
   * Creación de migraciones para las 13 tablas y seeders para datos de prueba.
3. **Fase 3: Desarrollo del Backend Base**
   * Definición de Modelos, Relaciones y Lógica de Autenticación.
4. **Fase 4: Panel Administrativo**
   * Desarrollo de las vistas CRUD para productos y categorías (necesario para llenar la tienda).
5. **Fase 5: Frontend Público (Tienda)**
   * Desarrollo de la Home, Catálogo, Filtros y Ficha de Producto.
6. **Fase 6: Carrito y Checkout**
   * Implementación de la lógica de compra y registro de pedidos.
7. **Fase 7: Pruebas y Despliegue**
   * Pruebas de usuario, corrección de bugs y despliegue en servidor de producción.