

Descripción corta de Arquitectura

Descripción corta de Arquitectura de la Tienda Virtual

Preparado por:
Geovanny Astorga López,
Luis Pablo Monge Angulo,
Josué Vargas Hernández

Índice

1. Introducción	2
1.1. Alcance	2
1.2. Propósito	2
1.3. Usuarios Interesados	2
1.4. Recomendaciones de conformidad con esta práctica	2
2. Referencias	2
3. Definiciones	4
4. Prácticas de descripción arquitectónica	4
4.1. Documentación de Arquitectura	4
4.2. Identificación de stakeholders y sus responsabilidades	5
4.3. Selección de puntos de vista de la arquitectura	5
4.4. Vistas de la Arquitectura	6
4.4.1. Escenarios - Casos de Uso	6
4.4.2. Lógica - Diagrama de Clases	7
4.4.3. Diagrama de Base de Datos	8
4.5. Consistencia en la cantidad de vistas de la arquitectura	10
4.6. Arquitectura Lógica	10
4.6.1. Calidad	10
4.7. Ejemplos de Uso	10
4.8. Detalles de implementación	10
4.8.1. Lenguajes y plataformas	10

1. Introducción

1.1. Alcance

En el presente documento está contenido una breve descripción del diseño de la arquitectura elaborado para la **Tienda Virtual**, el cual es producto del análisis hecho a partir de los requerimientos solicitados por el cliente para el respectivo sistema, estos requerimientos pueden ser satisfechos con las tecnologías y características evaluadas juntamente con el cliente y futuros usuarios.

El documento está organizado de la siguiente manera:

1. Las características generales del diseño.
2. Los modelos y vistas que o detallan.

1.2. Propósito

El propósito del presente documento es proporcionar una breve descripción arquitectónica de la Tienda Virtual, usando unos pocos diagramas de algunas vistas diferentes a partir del modelo 4 + 1, con el cual se espera dar una pincelada de lo que se requiere para el sistema

1.3. Usuarios Interesados

El presente documento que hace referencia a la arquitectura de Software utilizado en la Tienda Virtual, puede ser usado por todos las personas que desean comprender el diseño y construcción de dicho sistema. Para aquellos futuros usuarios que lo utilizaran tales como el cliente, usuarios finales, operadores.

También, servirá como base para que los desarrolladores de software, testers, ingenieros, diseñadores, aseguradores de calidad que puedan construir, implementar, revisar, probar, dicho sistema con los requerimientos definidos en el documento *SRS Tienda Virtual*.

Además el documento podrá ser utilizado por auditores y asesores independientes que cuenten con los permisos necesarios.

1.4. Recomendaciones de conformidad con esta práctica

NA.

2. Referencias

Las referencias utilizadas para la elaboración del presente documento son las siguientes:

- IEEE Std 1471-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
- IEEE/EIA Std 12207.0-1996, IEEE/EIA Standard-Industry Implementation of ISO/IEC 12207: 1995, Information Technology-Software life cycle processes.
- ISO/IEC/IEEE 42020, Architecture description template for use with ISO/IEC/IEEE 42010:2011.

3. Definiciones

- **Arquitecto:** Persona, equipo u organización responsable de la arquitectura del sistema.
- **Arquitectura:** La organización fundamental de un sistema encarnado en sus componentes, sus relaciones entre sí y con el medio ambiente, y los principios que guían su diseño y evolución.
- **Arquitectura de Software:** Base de un sistema de software que es construida pensando tanto en satisfacer las necesidades actuales, como en proporcionar al software las capacidades necesarias para permitir su mantenimiento y evolución de acuerdo a las necesidades del negocio y las solicitudes de los clientes.
- **Componente:** Cualquier elemento abstracto, visible, externo, de alto nivel, analizable, que pueda construir una funcionalidad de la solución del sistema.
- **Escenario:** Especifica el comportamiento y limita el interés de un área específica del sistema para uno o varios stakeholders.
- **HTTP:** Protocolo de Transferencia de Hipertexto.
- **Modulo:** Ver definición de componente.
- **Sistema:** Una colección de componentes organizados para cumplir una función específica o varias funciones.
- **SRS:** Documento de requerimientos de software.
- **Stakeholder:** Un individuo, equipo u organización con intereses, preocupaciones o afectados por un sistema.
- **TCP:** Protocolo de control de transmisión.
- **Tipos de vista:** Especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.
- **Vista:** Una representación de un sistema completo desde la perspectiva de un conjunto relacionado de preocupaciones.
- **Ziari:** Palabra eslovaca (raíz: *žiar*) que significa brillar, calor, ferviente, resplandor.

4. Prácticas de descripción arquitectónica

4.1. Documentación de Arquitectura

NA.

4.2. Identificación de stakeholders y sus responsabilidades

Stakeholder	Descripción
Administrador	Usuario encargado de manejar el sistema Ziari, creación de productos, revisión de stock, entre otras tareas
Usuario	Persona que utiliza el sistema con más frecuencia, encargado de hacer compras
Cliente	Persona a la que se le desarrolla el sistema Tienda Virtual, dueño del proyecto

4.3. Selección de puntos de vista de la arquitectura

Vistas	UML
Escenario	<ul style="list-style-type: none">■ Casos de Uso
Lógica	<ul style="list-style-type: none">■ Diagrama de Clases
Extra	<ul style="list-style-type: none">■ Diagrama de Base de Datos

4.4. Vistas de la Arquitectura

A continuación se mostrarán las diferentes vistas con sus respectivos diagramas.

4.4.1. Escenarios - Casos de Uso

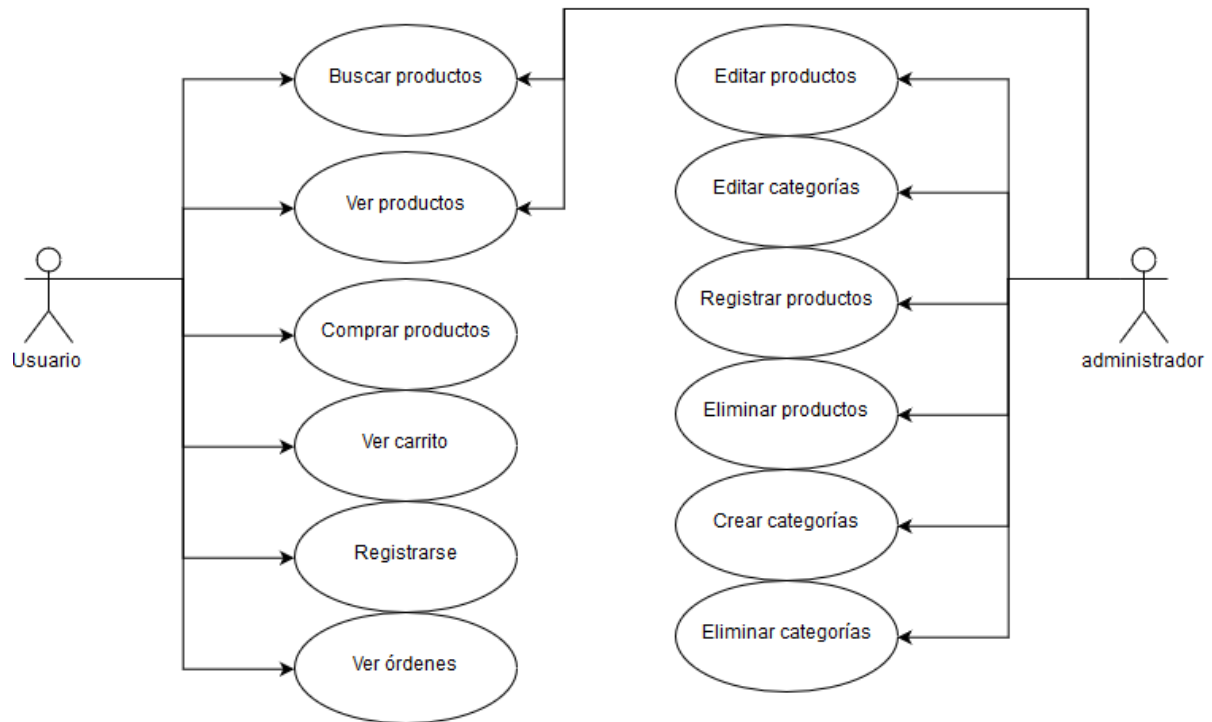


Figura 1: Diagrama de Casos de Uso

4.4.2. Lógica - Diagrama de Clases

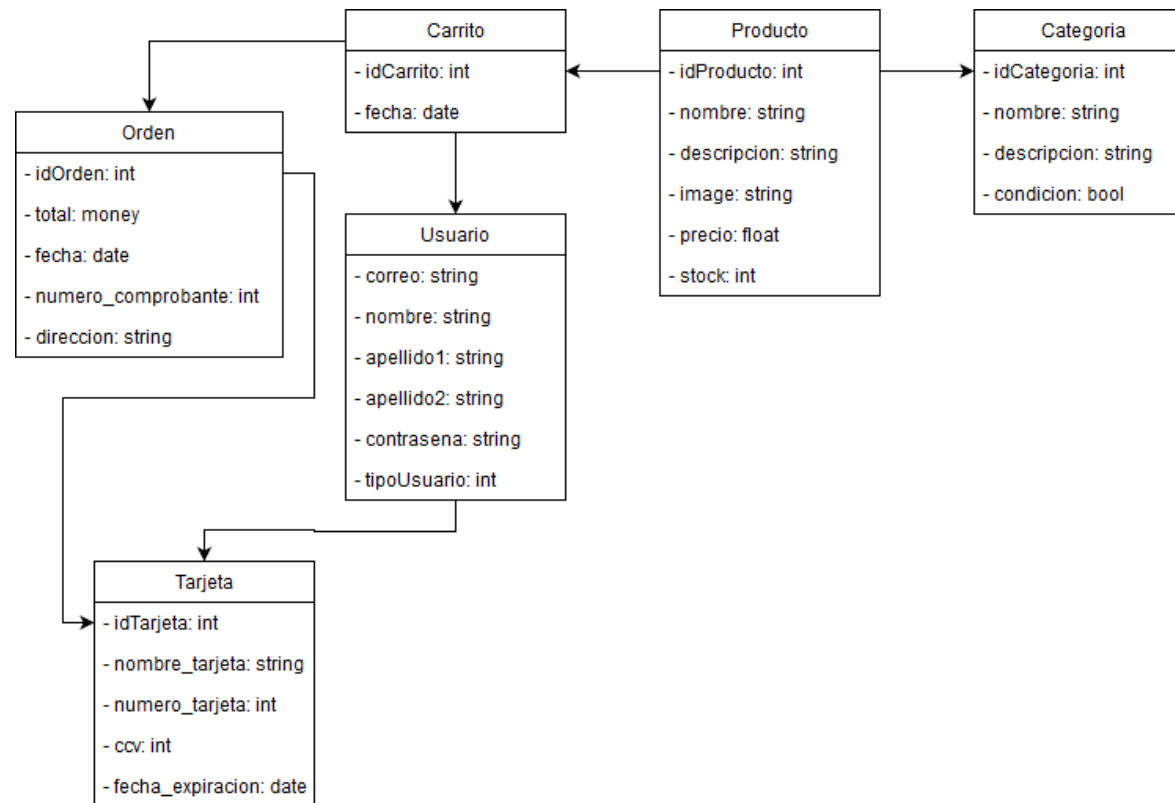


Figura 2: Diagrama de Clases

4.4.3. Diagrama de Base de Datos

4.5. Consistencia en la cantidad de vistas de la arquitectura

NA.

4.6. Arquitectura Lógica

4.6.1. Calidad

1. El sistema se desarrollo de forma web responsive.
2. Se tendrá un plan de pruebas para el desarrollo del sistema.

4.7. Ejemplos de Uso

NA.

4.8. Detalles de implementación

La especificación de un sistema intensivo en software tiene como última representación al código fuente de los componentes. Este código indica los detalles del software, por medio de un lenguaje, capaz de ser traducido automáticamente a instrucciones de la máquina. Acompaña al código las llamadas *previsiones de compilación*, constituidos por todos los elementos de soporte necesarios para realizar la construcción de los componentes a partir del conjunto de códigos. Esta sección detalla la obtención y uso del paquete de código fuente para el proyecto. Para facilitar el uso de este, para las futuras ampliaciones o correcciones del sistema.

4.8.1. Lenguajes y plataformas

La lógica de diseño arquitectónico aplicada en este documento, abre la posibilidad de que la implementación de la lógica sea efectuada con lenguajes de paradigma Orientado a Objetos, en este caso php con el framework Laravel. De este paradigma va a depender directamente lo que los desarrolladores podrán implementar, su curva de aprendizaje y en varios casos opciones propias de la empresa para la cual se efectúa el diseño.

Si la implementación se desea desarrollar bajo lenguajes que no sean del paradigma Orientado a Objetos, se deberá realizar una nueva vista que cumpla con los requerimientos funcionales y no funcionales de los stakeholders que lo solicitan.