



Universidad de las Ciencias
Informáticas

**Universidad de las Ciencias Informáticas
Facultad 4**

Tienda virtual de modelos tridimensionales Cosmox

*Trabajo de Diploma para optar por el título de Ingeniero
en Ciencias Informáticas*

Autora: Yipsi Lourdes Sánchez Mesa

Tutor:

Ing. Jose Emilio Badia Valdés

**La Habana, Diciembre 2022
“Año 64 de la Revolución”**

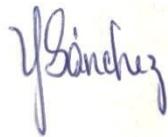
La Ciencia es una tentativa en el sentido de lograr que la caótica diversidad de nuestras experiencias sensoriales corresponda a un sistema de pensamiento lógicamente ordenado.

Albert Einstein

Declaración de autoría

Declaro ser autora de la presente tesis que tiene por título: Tienda virtual de modelos tridimensionales Cosmox y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los 5 días del mes de diciembre del año 2022.



Yipsi Lourdes Sánchez Mesa

Autora



Ing. Jose Emilio Badia Valdés

Tutor

Dedicatoria

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional. A mi padre, a pesar de nuestra distancia física, siento que estás conmigo siempre y aunque nos faltaron muchas cosas por vivir juntos, sé que este momento hubiera sido tan especial para ti como lo es para mí. A mi madre, por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional. A mi novio, gracias por ser no solo mi novio, sino también mi mejor amigo, cómplice, confiar en mí y apoyarme en todos y cada uno de los pasos que doy en mi vida, siempre estas ahí para mí. A mi tía Rebeca, a quien quiero como a una madre, por compartir momentos significativos conmigo y por siempre estar dispuesta a escucharme y ayudarme en cualquier momento. A Tonito, porque te amo infinitamente hermano.

Agradecimientos

A mi papá Antonio Sánchez Pérez, el amigo incondicional que me enseñó a trabajar por mis sueños y quien tuvo fe en mí siempre. Aunque ya no está en este plano, su amor seguirá conmigo todos los días de mi vida.

A mi mamá Odilia Mesa Díaz, por ser la amiga y compañera que me ha ayudado a crecer, gracias por estar siempre conmigo en todo momento. Gracias por la paciencia que has tenido para enseñarme, por el amor que me das, por tus cuidados, por los regaños que me merecía y que no entendía. Gracias mamá por estar al pendiente durante toda esta etapa.

A mi novio por estar siempre para mí, por los ánimos semestre a semestre, por acompañarme durante todo este proceso y por nunca dudar que lo lograría.

A mis abuelos, tíos y tía por estar siempre al tanto de todo lo que acontece en mi vida y esperar lo mejor de mí.

A mi hermano Tonito, por estar siempre pendiente y ayudarme en todo lo que necesité.

A los profesores que me formaron durante estos 5 años.

A mi tutor, Jose Emilio Badía Valdés, por haberme orientado en todos los momentos que necesité sus consejos.

En fin, a todos los que de una forma u otra tuvieron que ver en la obtención de este logro tan grande...

MUCHAS GRACIAS!!!

Resumen

La generación, procesamiento y visualización de modelos en 3D o tridimensionales se ha convertido hoy en día en un estándar en cuanto a diseño o generación de gráficos por computadoras para casi todas las ramas de la industria. La de videojuegos, no escapa a esta realidad, siendo una de las ramas que más utiliza este tipo de herramientas debido al papel fundamental que juegan los modelos tridimensionales para conformar el entorno o escena que ambientan el concepto de los mismos. Internet, actualmente, brinda un sinnúmero de posibilidades y una de ellas son las tiendas virtuales, mediante las cuales se puede comercializar cualquier tipo de bienes y servicios.

En el presente trabajo se aborda el desarrollo de una tienda virtual de modelos tridimensionales para el “Centro de Tecnologías Interactivas (VERTEX)”. Actualmente los desarrolladores de modelos 3D de VERTEX no cuentan con una plataforma que les permita subir sus diseños y que estos puedan ser comprados. Esto formaría parte de otro mecanismo de entrada monetaria a la Universidad y a VERTEX, con lo que se estimularía a sus creadores por el trabajo realizado.

Para el desarrollo de la tienda, se utilizaron como tecnologías: el entorno de desarrollo integrado PyCharm, lenguaje de programación Python, framework de desarrollo Django, el Lenguaje de Base de datos PostgreSQL y el sistema gestor de base de datos PgAdmin4. El proceso de desarrollo de *software* estuvo guiado por la metodología Programación Extrema (XP). Finalmente se obtuvo una tienda que permite publicar los modelos tridimensionales desarrollados en VERTEX.

Palabras clave: modelos 3D, tienda virtual.

Abstract

The generation, processing and visualization of 3D or three-dimensional models has become today a standard in terms of design or generation of computer graphics for almost all branches of the industry. The video game industry does not escape from this reality, being one of the branches that most uses this type of tools due to the fundamental role that three-dimensional models play in shaping the environment or scene that set the concept of the same. Internet, nowadays, offers us endless possibilities and one of them is the virtual stores, through which any type of goods and services can be commercialized.

This paper deals with the development of a virtual store of three-dimensional models for the "Interactive Technology Center (VERTEX)". Currently, the developers of 3D models for VERTEX do not have a platform that allows them to upload their designs and make them available for purchase. This would be part of another mechanism for monetary input to the University and VERTEX, thus stimulating its creators for their work.

For the development of the store, the following technologies were used: PyCharm integrated development environment, Python programming language, Django development framework, PostgreSQL database language and PgAdmin4 database management system. The software development process was guided by the Extreme Programming (XP) methodology. Finally, a store was obtained that allows publishing the three-dimensional models developed in VERTEX.

Keywords: *3D models, virtual store.*

Índice

Introducción	1
Capítulo 1. Fundamentación teórica de la comercialización de modelos tridimensionales sobre tiendas virtuales y los sistemas que sustentan su automatización.....	5
1.1 Comercialización de modelos tridimensionales en tiendas virtuales	5
1.2 Análisis valorativo de sistemas homólogos.....	8
1.3 Metodología de desarrollo de software	10
1.4 Tecnología de <i>software</i> para el desarrollo de una tienda virtual	15
Conclusiones parciales	17
Capítulo 2. Propuesta de solución de la tienda virtual de modelos tridimensionales COSMOX.....	19
2.1. Propuesta de solución	19
2.2 Requerimientos funcionales	19
2.3 Requerimientos no funcionales	21
2.4 Historia de Usuario (HU)	22
2.5 Estimación de esfuerzo por Historias de usuario	26
2.6 Arquitectura de la aplicación.....	27
2.7 Diseño de la base de datos	29
2.8 Tarjetas de Contenido, Responsabilidad y Colaboración.....	30
2.9 Patrones de diseño	32
Conclusiones parciales	34
Capítulo 3. Implementación y pruebas de la tienda virtual de modelos tridimensionales COSMOX.....	35
3.1 Tareas de ingeniería	35
3.2 Interfaces de la tienda virtual.....	39
3.3 Pruebas de software	42
3.3.1 Pruebas unitarias	43
3.3.2 Pruebas de aceptación.....	47
3.3.3 Pruebas de carga y estrés.....	50
Conclusiones parciales	52
Conclusiones generales.....	53
Recomendaciones	54
Bibliografía Consultada:.....	55
Anexos	59

Introducción

En este siglo XXI, el desarrollo del *hardware* de los equipos de cómputo hizo que surgiera una nueva rama de las artes gráficas, la cual se sustenta en *software* especiales de diseño 3D o tridimensional: los gráficos 3D por computadora (3DCG, por sus siglas en inglés), brindan la posibilidad de representar en un espacio de trabajo tridimensional cualquier objeto, lo cual conduce a la obtención de un modelo tridimensional (Izquierdo Pardo, Pardo Gómez, & Izquierdo Lao, 2020).

Los modelos tridimensionales constituyen gráficos 3D generados por computadoras a través del empleo de programas informáticos especiales de trabajo en 3D. Para su obtención se requiere realizar trabajos de artes gráficas en computadoras y el estudio de técnicas y tecnologías relacionadas con estos.

La generación, procesamiento y visualización de modelos en 3D o tridimensionales se ha convertido hoy en día en un estándar en cuanto a diseño o generación de gráficos por computadoras para casi todas las ramas de la industria. La de videojuegos, no escapa a esta realidad, siendo una de las ramas que más utiliza este tipo de herramientas debido al papel fundamental que juegan los modelos tridimensionales para conformar el entorno o escena que ambientan el concepto de los mismos. El creciente desarrollo tecnológico ha propiciado que a nivel mundial la producción y el consumo de videojuegos se posicionen como una manera de narrar, crear y participar en la producción de contenidos que son además preferidos por varios grupos sociales como forma de entretenimiento y aprendizaje.

Las nuevas Tecnologías de la Información y las Comunicaciones (TICs) han revolucionado la forma de vivir, permitiendo la invención de nuevos bienes y servicios y nuevos métodos de comercialización y cobro. Las virtudes de las TICs no son difíciles de enumerar: su mayor velocidad, capacidad y distribución de la información permiten que usuarios de distintas partes del planeta pueden conectarse usando computadores y otros dispositivos especializados, para comunicarse de múltiples maneras y emprender diversas transacciones: comprar y vender objetos e información, compartir datos personales, conversar en tiempo real, incluso jugar videojuegos en línea aún sin hablar el mismo idioma (Etecé, 2022).

Cada vez más servicios y productos se ofrecen a través de Internet o de aplicaciones en dispositivos móviles, permitiendo el desarrollo de toda una rama comercial inmediata e internacional. Esta situación ha pasado de ser una opción, a convertirse en una necesidad, por lo que se puede decir que la humanidad está siendo testigo de una nueva era para los negocios y su relación con los clientes.

En la actualidad, se habla de la necesidad de repensar el negocio de forma transversal para encontrar nuevas oportunidades y ventajas competitivas. Es en este contexto donde nace la oportunidad de plantear una hoja de ruta del negocio en la era de la disrupción tecnológica,

estableciendo estrategias tanto internas como externas para adaptar la compañía a las prestaciones que ofrecen las nuevas tecnologías. Internet, actualmente, brinda un sinfín de posibilidades y una de ellas son las tiendas virtuales, las cuales traen consigo una metamorfosis respecto a nuestra forma de vida (Asesores, 2022). Este medio, forma parte fundamental de lo que se conoce como comercio electrónico y está teniendo un gran apogeo en los últimos años. Además, traen consigo ventajas como: permanece abierta al público las 24h (lo cual **facilita la compra** a aquellas personas que, por incompatibilidad de horarios de comercio, no pueden ir a las tiendas), se puede vender durante todo el año; permite llegar a más clientes y reduce costos, comparada con una tienda física, los costos son muchos menores.

El Centro de Tecnologías Interactivas (VERTEX), perteneciente a la Universidad de las Ciencias Informáticas (UCI), se especializa en la creación de videojuegos. Con el objetivo de integrar la sociedad cubana a los videojuegos, desarrollaron la Plataforma Cosmox. La cual es un espacio para publicar videojuegos exclusivamente cubanos, interactuar con los usuarios y brindar servicios a los desarrolladores, cuyo objetivo es convertirla en una plataforma con un conjunto de servicios para ofrecer contenidos entretenidos de producción nacional (Cuba a Pulso, 2021).

A raíz de la realización del evento *Global Game Jam* y el lanzamiento de dicha plataforma, junto a la gama de videojuegos que la integran y sus servicios; los organismos afines y la comunidad de desarrolladores de videojuegos en Cuba han adquirido un interés importante por ampliar y mejorar los productos de este tipo. Sin embargo, el proceso de desarrollo de videojuegos por lo general requiere de un grupo de “assets” o recursos que en su mayoría son elaborados en 3D y por su complejidad requieren de un gran consumo de tiempo por parte de los diseñadores; característica que afecta el tiempo de desarrollo de los mismos.

La situación anterior ha dado lugar a una necesidad de los desarrolladores de acceder a recursos 3D previamente elaborados, que permitan agilizar y mejorar el proceso de desarrollo de videojuegos y el uso de modelos tridimensionales en industrias que requieran de estos recursos. Actualmente los desarrolladores de modelos 3D del Centro VERTEX no cuentan con una tienda que les permita comercializar sus diseños.

Debido a las limitantes abordadas anteriormente se tiene como **problema de investigación**: ¿Cómo contribuir a la comercialización de los modelos tridimensionales generados por los desarrolladores de videojuegos de VERTEX?

Este problema tiene como **objeto de estudio**: proceso de comercialización de modelos tridimensionales. De esta manera el **campo de acción** queda enmarcado como: comercialización de modelos tridimensionales en tiendas virtuales.

Para dar solución al problema planteado se propone como **objetivo general**: Desarrollar una tienda virtual que contribuya a la comercialización de los modelos tridimensionales generados por los desarrolladores de videojuegos de VERTEX.

Tareas de investigación:

1. Construcción del marco teórico de la investigación, relacionado con la comercialización de modelos tridimensionales a través de tiendas virtuales.
2. Caracterización del proceso de comercialización de modelos tridimensionales.
3. Implementación de la tienda virtual.
4. Validación de la solución propuesta mediante pruebas de software.

Para el cumplimiento del objetivo general planteado se utilizarán los siguientes métodos de investigación:

Métodos Teóricos:

Analítico-Sintético: este método se utiliza para realizar el análisis y estudio de las bibliografías existentes sobre las plataformas web de tiendas virtuales y lograr obtener de manera sintetizada el contenido necesario y suficiente para la ejecución de la investigación.

Métodos empíricos:

Observación: se utiliza como método de referencia al observar distintas tiendas virtuales que comercializan modelos tridimensionales que sirvieron de análisis. Esto permitió definir las características fundamentales de la tienda.

Con el desarrollado de la Plataforma web se esperan como beneficios: gestionar modelos tridimensionales por parte de cualquier usuario o entidad. Permitir a los internautas publicar sus modelos en un muro personal donde otros usuarios puedan comprarlos utilizando los mecanismos de pago establecidos por el Gobierno Cubano. Contribuir a la interacción entre diseñadores y desarrolladores de modelos tridimensionales. Aportar un mecanismo de entrada monetaria a la Universidad y el Centro de Tecnologías Interactivas.

El documento está estructurado en 3 capítulos:

Capítulo 1. Fundamentación teórica de la comercialización de modelos tridimensionales sobre tiendas virtuales y los sistemas que sustentan su automatización: en este capítulo se aborda todo lo referente a la fundamentación teórica de la investigación de tiendas virtuales de modelos tridimensionales, se realiza el estudio del arte de sitios web similares. Para un mejor entendimiento se detallan los conceptos asociados al tema y componentes generales de una tienda virtual. Se adopta la metodología de desarrollo a emplear, se selecciona el lenguaje de programación a utilizar, así como la herramienta y lenguaje para el modelado. Queda plasmado el sistema gestor de base de datos, además de las herramientas para el desarrollo de la tienda virtual.

Capítulo 2. Propuesta de solución de la tienda virtual de modelos tridimensionales COSMOX: en este capítulo se describe la propuesta de solución y se ofrecen detalles de los principales aspectos relacionados con su diseño. Se definen las funcionalidades que deberán tener la tienda virtual y el diseño de la arquitectura que tendrá la misma para una mejor organización de sus componentes. Se definen los patrones de diseño usados, así como el modelo de datos para un mejor entendimiento de cómo funciona la tienda virtual de modelos tridimensionales.

Capítulo 3. Implementación y pruebas de la tienda virtual de modelos tridimensionales COSMOX: en este capítulo se abordan las fases de implementación y pruebas, realizadas a la tienda virtual. Además se describen las pruebas realizadas al sistema para validar la solución.

Capítulo 1. Fundamentación teórica de la comercialización de modelos tridimensionales sobre tiendas virtuales y los sistemas que sustentan su automatización

En el capítulo se describirán los conceptos fundamentales asociados al objeto de estudio, así como los principales componentes que deben conformar una tienda virtual de modelos 3D. Se lleva a cabo un análisis de aplicaciones con funcionalidades semejantes. Por último, se describen las diferentes herramientas y tecnologías a utilizar para su desarrollo y se realiza un estudio valorativo de las metodologías de desarrollo de software.

1.1 Comercialización de modelos tridimensionales en tiendas virtuales

Uno de los procesos que rigen el funcionamiento de la sociedad moderna es la comercialización de bienes y servicios. Establecida desde tiempos antiguos en el que se estableció el intercambio monetario por esos bienes y servicios. El impacto de las TICs a impuesto nuevas formas de comercialización, que tienen su expresión más clara en el comercio electrónico. Desde la década de los 90, con el objetivo de acercar los productos a los consumidores se incorporó el **comercio electrónico** como un fenómeno internacional, el cual se caracteriza como el proceso de compra y venta de productos por medios electrónicos, como aplicaciones móviles e Internet. Se refiere tanto al comercio minorista en línea como a las compras en línea, así como a las transacciones electrónicas (Ecommerce-Platforms, 2022).

Por medio de internet, los clientes pueden acceder a diversas marcas, productos y servicios en todo momento, en cualquier lugar. La relevancia de este tipo de comercio es tal que los negocios lo toman como parte de la estrategia de ventas gracias a su eficiencia. Los establecimientos ya cuentan con páginas web y crean perfiles en redes sociales para conseguir llegar a un mayor rango de público. Vender y comprar ya es una tarea sencilla propiciada, desde luego, por la tecnología, como los dispositivos móviles con acceso a la red (Higuerey, 2019).

El pago de los productos y servicios se lleva a cabo mediante una **plataforma de pago**, que consiste en un portal que conecta una cuenta bancaria con el procesador de pagos correspondiente. Transmite la información de la transacción, ya sea de forma virtual a través de servicios de pago web y APIs o en persona a través de una terminal de pago (Betania V., 2022). Los negocios de comercio electrónico necesitan una pasarela de pago ya que garantiza la seguridad de los pagos y protege los datos sensibles.

La cúspide del comercio electrónico es precisamente el diseño e implementación de **tiendas virtuales o tiendas en línea**, se pueden describir como una plataforma de comercio que se vale de un sitio web para realizar sus ventas y transacciones. Por lo general, las compras en una tienda virtual se pagan con tarjeta de crédito en el mismo sitio web. Sin embargo, se pueden utilizar otros medios de pago como transferencias bancarias, cupones de pago, entre otros. En la mayoría de los

casos la tienda virtual suele requerir que los usuarios se registren ingresando sus datos antes de poder realizar una compra (Headways , 2016).

En la actualidad, existe un gran número de tiendas virtuales con todo tipo de precios y desde el manejo complejo hasta lo más sencillo, a nivel internacional se puede mencionar a Amazon. Esta, es una empresa de tecnología multinacional estadounidense basada en Internet que vende libros, música, películas, artículos para el hogar, productos electrónicos, juguetes y muchos otros productos (Cortés Sánchez, 2022).

En Cuba, como consecuencia de las medidas de aislamiento social para evitar la propagación de la COVID-19, se generó una oportunidad de desarrollo para el *e-commerce*, con el objetivo de acercar a las familias los productos alimenticios y de primera necesidad hasta su domicilio; apoyado por la posibilidad de uso de diferentes canales de pagos electrónicos ya instrumentados en el país. Como cumplimiento a las indicaciones gubernamentales, estimular el *e-commerce* como una de las nuevas formas de comercialización es una de las políticas de la informatización de la sociedad. Se pretende la comercialización de productos a través de las tiendas virtuales mediante las plataformas creadas por las empresas de comercio minorista (Lopes Martínez, y otros, 2021). De las existentes en la actualidad, se encuentra Superfácil, la cual pone a disposición de la población la posibilidad de acceder a servicios y productos cubanos en todo el país.

Las tiendas virtuales a nivel internacional y también en el contexto cubano, no solo comercializan bienes tangibles, sino también intangibles. Dentro de los principales bienes intangibles que en la era moderna se comercializan, están los videojuegos. Un **videojuego** es una aplicación interactiva orientada al entretenimiento que, a través de ciertos mandos o controles, permite simular experiencias en la pantalla de un televisor, una computadora u otro dispositivo electrónico (Pérez Porto & Gardey, 2022). Los videojuegos se han convertido en una industria sofisticada y próspera que es capaz de dinamizar la economía.

Una parte fundamental de los videojuegos, que garantiza su desarrollo acelerado son los modelos tridimensionales. Un **modelo tridimensional** es la representación matemática de cualquier objeto tridimensional (real o imaginario) en un entorno de *software* 3D. Para el proceso de modelado 3D es necesario tener en cuenta las propiedades físicas, como texturas, tamaños, triángulos, vértices, etc. y así poder construir cualquier elemento que se desee, desde personajes a objetos, que pueden o no, tener una animación (Vanegas Prieto, 2020).

Las tiendas virtuales dedicadas a la comercialización de modelos tridimensionales tienen características y componentes muy propias que la literatura las recoge, como:

Diseño

El diseño de la tienda *online* es la carta de presentación de cara al público y la primera impresión que causarás en tus posibles clientes. Por eso, contar con un diseño atractivo, puede tener un

enorme impacto en el número de ventas de tu tienda. Un comercio electrónico bien diseñado, le proporcionará al usuario una experiencia mucho más agradable y hará que sea más fácil navegar, buscar y elegir los productos, para completar el proceso de compra con más agilidad (LanceTalent, 2015).

Usabilidad

El catálogo de modelos 3D debe estar visible desde el primer momento. Fácil acceso mediante una navegación por categorías y subcategorías. El proceso de la compra debe estar enfocado al modelo 3D de una forma clara y rápida. Disponer de un potente buscador que ofrezca la posibilidad de acceder al catálogo por distintos criterios como (precio, fecha u orden alfabético). Facilitar el acceso a apartados de servicio de la tienda como información de contacto, forma de comprar y condiciones generales. El proceso de compra debe estar enfocado al producto de una forma rápida y clara, a su vez debe ser fácil y rápido para tratar de finalizar el mayor número de ventas posibles (Consultec, 2021).

Accesibilidad

El catálogo debe ser accesible por categorías comerciales. Es importante su correcta clasificación utilizando categorías como novedades, más vendidos, en oferta, entre otras. En todos los casos se tendría que escoger el tamaño del texto óptimo y destacando aquella información más relevante: nombre del modelo, botón Comprar y el precio (Consultec, 2021).

Catálogo de modelos 3D

El catálogo de modelos 3D es la otra carta de presentación a los clientes. Se debe prestar atención y cuidado a la hora de seleccionar qué modelos 3D se van a ofrecer, y cómo se van a mostrar y destacar (Consultec, 2021).

Carrito de la compra

La cesta o carrito de la compra es un elemento indispensable en la tienda virtual. Este elemento debe ofrecer la posibilidad de añadir, eliminar o modificar los modelos 3D que durante la navegación se han ido seleccionando e incorporando. Se debe dar la posibilidad, que el cliente pueda visualizar de una forma clara: los productos comprados especificando la cantidad e importe total de la compra (Consultec, 2021).

Motor de búsqueda

Con la idea de facilitar al cliente encontrar los modelos 3D, es indispensable disponer de un potente motor de búsqueda o buscador integrado que permita la búsqueda por diversos criterios (Consultec, 2021).

Registro y área de usuario

El registro de clientes es un aspecto muy importante de una tienda virtual, tanto para que realicen sus pedidos, como a la hora de poder llevar a cabo acciones posteriores de fidelización y captación

de nuevos clientes partiendo de una base de datos de clientes extensa. En la zona de registro de un comercio electrónico los clientes deben poder consultar, al menos: datos personales, estado del pedido realizado e histórico de pedidos y facturación (Consultec, 2021).

1.2 Análisis valorativo de sistemas homólogos.

En la actualidad existen varias tiendas virtuales que comercializan modelos tridimensionales. Como parte de la investigación se llevó a cabo un estudio de las mismas, que cuentan con funcionalidades que puedan satisfacer las necesidades del Centro. Debido a esto, solo se tomaron en cuenta las tiendas que presentaran un sistema de pago.

Cults3D

Forma parte de una de las principales bibliotecas de modelos 3D, clasificando el contenido por categorías y ofreciendo el archivo necesario por precios que varían en función de la complejidad del modelo. Existen archivos gratuitos, y otros de varios miles de dólares, siendo posible ordenar las listas en función de esta y otras variables, todo desde el menú superior. Juguetes, piezas, colgantes, anillos, piezas de decoración, herramientas, lámparas, etcétera; la cantidad es bastante variada, y los detalles de cada objeto muestran exactamente lo que hay disponible, para evitar sorpresas después de la descarga.

Además de ofrecer la oportunidad de vender modelos y de descargar las creaciones de otras personas, ofrecen un servicio de creación de modelos 3D personalizados y 100% imprimibles, siendo necesario rellenar un formulario con los detalles de dicho modelo y con el presupuesto que se tiene para poder encontrar la mejor relación entre coste y beneficio. Los creadores de modelos en 3D pueden subir sus trabajos y establecer el precio, así como interactuar con los compradores para que puedan realizarse ajustes (Diego Polo, 2017). El pago de los diseños adquiridos se realiza a través de la compañía de transferencia de dinero *PayPal*. Los usuarios reciben un 80% de ganancias por cada diseño vendido, mientras que la tienda se queda con el 20% restante.



Figura 1. Tienda Cults3D

Free3D

Es una plataforma *online* impulsada por la comunidad *Maker* para publicar archivos tridimensionales. Destaca entre otras cosas por su interfaz de usuario, que es muy intuitiva y fácil de usar. Como dice su nombre, muchos de los modelos que se encuentran en esta página son de acceso gratuito, pero muchos otros son de pago. Esta segunda opción permite a los creadores y diseñadores exhibir las obras y modelos que han trabajado, y obtener un beneficio de ello. A pesar de ello, es considerada una de las mejores páginas para encontrar archivos 3D. Las transferencias para el pago de los productos comprados se realiza mediante la plataforma de pago *PayPal*.

Principales características

Free3D es conocida en la comunidad por su interfaz de usuario clara y estructurada. Cuenta con más de 420.000 diseños 3D, que son fáciles de encontrar gracias a la forma de ordenarlos por categorías según el tipo de modelo que se busca. Entre ellos hay archivos gratuitos de arquitectura, vehículos, personajes, animales, muebles y figuras. La principal competencia entre los miembros de la comunidad se encuentra debido a la función de votación, que permite a los usuarios valorar positiva o negativamente un diseño. Uno de los puntos más a favor de Free3D es la cantidad de idiomas en los que está disponible (Alicia M., 2021).

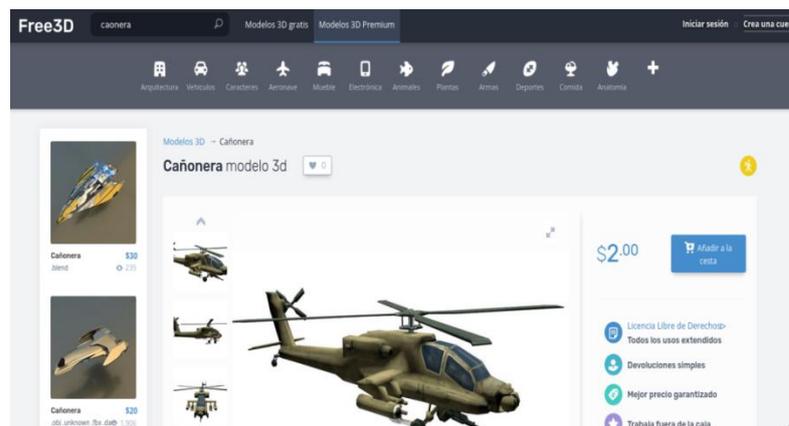


Figura 2. Tienda Free3D

Unity Asset Store

Alberga una creciente biblioteca de *assets* gratuitos y comerciales creados tanto por *Unity Technologies* como por miembros de la comunidad. Hay una gran variedad de *assets* disponibles, que abarcan desde texturas, modelos y animaciones, hasta ejemplos de proyectos completos, tutoriales y *asset* de extensión. El contenido disponible en la Tienda de *Asset* ayuda a mejorar tu proyecto, juego o aplicación y a reducir la carga de trabajo necesaria para crear herramientas o modelos desde cero. Los *assets* se adquieren en el sitio web de la Tienda de *Asset* y pueden añadirse a tus proyectos mediante el Gestor de Paquetes, dentro del Editor (Unity, 2016). La tienda utiliza la plataforma de pago *Paypal*, para pagar los *asset* adquiridos y para hacerles llegar las ganancias a los propietarios.

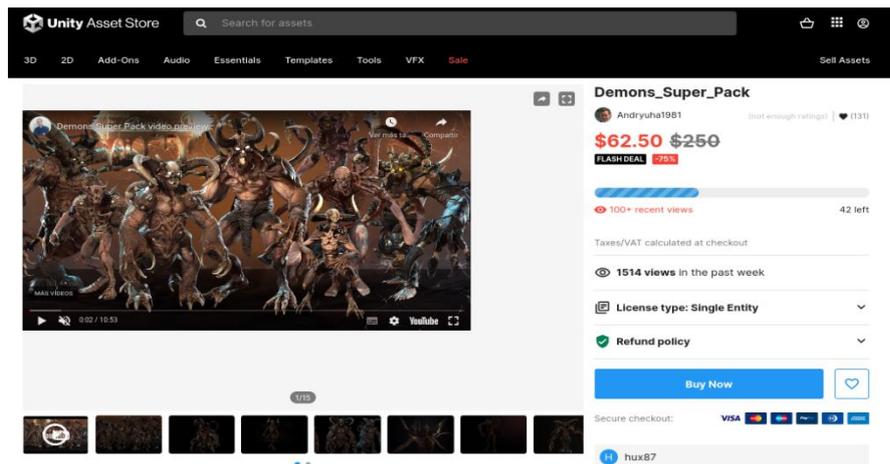


Figura 3. Tienda *Unity Asset Store*

En cuanto a funcionalidades, cualquiera de las tiendas anteriores pudiese satisfacer las necesidades porque permiten subir los diseños de los desarrolladores para su comercialización. Por otra parte, poseen varios inconvenientes que imposibilitan su utilización. Uno de estos es que no permiten la administración de las cuentas de usuario ni de los diseños que se suban a estos sitios, por lo que no se pudiera tener el control del uso de los mismos.

Otro de los rasgos negativos es que por causa del bloqueo, Cuba no puede usar la compañía de transferencia de dinero *PayPal*, por tal motivo, los diseñadores del país se ven limitados a que sus productos no sean comercializados. Por otra parte, el código fuente de estos sitios no está liberado pues pertenecen a organizaciones privadas y una de las políticas en Cuba es fomentar el uso de *software* libre y aplicaciones de código abierto como vía para alcanzar la soberanía tecnológica.

Pero no todo es negativo sobre la investigación realizada, pues a pesar de no poder utilizar estas tiendas, fue posible identificar funcionalidades y características que pueden contribuir con una mayor calidad de un desarrollo propio.

Entre ellas se encuentran:

- Clasificación del contenido por categorías según el tipo de modelo que se busca.
- El buscador, facilita a los clientes a encontrar el producto que necesitan.
- Forma de mostrar las propiedades de cada producto.
- Estilos de diseño.

1.3 Metodología de desarrollo de software

Actualmente las metodologías de ingeniería de *software* pueden considerarse como una base necesaria para la ejecución de cualquier proyecto de desarrollo de *software* que se considere serio, y que necesite sustentarse en algo más que la experiencia y capacidades de sus programadores y equipo. Estas metodologías son necesarias para poder realizar un proyecto profesional, tanto para

poder desarrollar efectiva y eficientemente el *software*, como para que sirvan de documentación y se puedan rendir cuentas de los resultados obtenidos.

Un amplio y buen conocimiento de estas metodologías servirá de base teórica y permitirá comprender completamente todo lo que requiere el análisis, diseño, desarrollo e implantación de un sistema. Además es importante, por la demanda que se tiene hoy en día por parte de muchas empresas, el conocimiento de algunas metodologías de desarrollo de *software* en específico.

Una metodología de desarrollo de *software* es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo han evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo (Maida & Pacienza, 2017).

Las primeras que surgieron fueron las metodologías tradicionales, como su nombre indica, son las que se han usado toda la vida. Buscan imponer disciplina al proceso de desarrollo de *software* y de esa forma volverlo predecible y por ello eficiente (Demera, 2021).

De hecho, estas metodologías tienen un enfoque predictivo, donde se sigue un proceso secuencial en una sola dirección y sin marcha atrás. La estimación/captura de requisitos se realiza una única vez al principio del proyecto y es precisamente por eso que nuestra estimación tendrá mucha importancia ya que de ella dependen todos los recursos que emplearemos en el proyecto. Si se quiere adoptar una metodología tradicional, el desarrollo de un proyecto debe empezar siempre con un riguroso proceso de captura de requisitos, análisis y diseño.

Las metodologías ágiles surgen como alternativa a las tradicionales porque ayudan a reducir la probabilidad de fracaso por sub-estimación de costos, tiempos y funcionalidades en entornos cambiantes. Además, se caracterizan por ser adaptativas y flexibles, esto significa que no son reticentes a los cambios, al revés, los imprevistos son eventos esperados que aprenderás a acoger con gran normalidad (Demera, 2021).

En la tabla que se muestra a continuación aparece una comparativa entre estos dos grupos de metodologías:

Tabla 1. Comparación entre Metodología Ágil y Tradicional

Metodologías ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.

Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo).	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del <i>software</i> .	La arquitectura del <i>software</i> es esencial y se expresa mediante modelos.
Poca documentación.	Documentación exhaustiva.
Muchos ciclos de entrega.	Pocos ciclos de entrega.

Dentro de las metodologías Tradicionales se pueden encontrar:

Waterfall:

También denominada en “cascada”. Consiste en desarrollar un proyecto de forma secuencial, comenzando con las fases de análisis y diseño y terminando con las de testeo y puesta en producción. Cada fase tiene entregables específicos y un proceso sencillo de revisión. No es un modelo ideal para proyectos de gran tamaño. Si los requisitos no están claros al principio, es un método poco efectivo (Deloitte, 2022).

Rational Unified Process (RUP, por sus siglas en inglés):

El proceso unificado de desarrollo de *software* es un conjunto de actividades usadas para transformar los requisitos de un usuario en un *software*; cuenta con tres características clave: dirigido a casos de uso, centrado en la arquitectura, iterativo e incremental. Este proceso involucra cuatro fases (inicio, elaboración, construcción, transición) y nueve disciplinas, las cuales son repetibles en cada fase (modelado del negocio, requisitos, análisis y diseño, implementación, pruebas, despliegue, gestión del cambio y configuraciones, gestión del proyecto, entorno). A las seis primeras disciplinas se les conoce como flujos de trabajo del proceso y a las tres últimas como flujos de trabajo de soporte (García Nieto, Martínez Trinidad, de los Santos Posadas, & Jiménez Casas, 2017).

Dentro de las ágiles están:

Scrum:

Se enfoca en entregar varias iteraciones de un producto para proporcionar a las partes interesadas el mayor valor comercial en la menor cantidad de tiempo. Fomenta que los productos se construyan más rápido, ya que cada conjunto de objetivos debe completarse dentro del marco de tiempo de cada *sprint*, ello requiere una planificación frecuente y el establecimiento de objetivos, lo que ayuda al equipo Scrum a centrarse en los objetivos del *sprint* actual y aumentando así la productividad (Euroforum, 2022).

Programación Extrema (XP):

Es una Metodología basada en prueba y error. Cliente bien definido, los requisitos pueden y van a cambiar. Grupo pequeño y muy integrado (máximo 12 personas). Equipo con formación elevada y capacidad de aprender. Reduce el costo del cambio en todas las etapas del ciclo de vida del sistema. Se basa en la simplicidad, la comunicación y la realimentación del código desarrollado (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2017).

Objetivos de XP

- La Satisfacción del cliente.
- Potenciar el trabajo en grupo.
- Minimizar el riesgo actuando sobre las variables del proyecto: costo, tiempo, calidad, alcance.

Características fundamentales de XP:

- No se planifica o diseña para un futuro distante, estas actividades se realizan durante el proceso de desarrollo.
- Se basa en prueba y error.
 - Pocos artefactos.
 - Pocos roles.
 - Se aplica en grupos pequeños.
 - Pruebas continuas.
 - Código compartido con el equipo.

Artefactos:

Historias de usuario (HU):

Las HU representan una breve descripción del comportamiento del sistema, se realizan por cada característica principal del sistema y son utilizadas para cumplir estimaciones de tiempo y el plan de lanzamientos, así mismo reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2017). Cada HU debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas.

Tareas de ingeniería:

Una HU se descompone en varias tareas de ingeniería, las cuales describen las actividades que se realizarán en cada historia de usuario, así mismo las tareas de ingeniería se vinculan más al desarrollador, ya que permite tener un acercamiento con el código (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2017).

Pruebas de aceptación:

Las Pruebas de aceptación son de vital importancia para el éxito de una iteración y el comienzo de la siguiente, con lo cual el cliente puede conocer el avance en el desarrollo del sistema y a los programadores lo que les resta por hacer. Además, permite una retroalimentación para el desarrollo de las próximas historias de usuarios a ser entregadas. Estas son comúnmente llamadas pruebas del cliente, por lo que son realizadas por el encargado de verificar si las historias de usuarios de cada iteración cumplen con la funcionalidad esperada (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2017).

Tarjetas CRC (Clase/Responsabilidades /Colaboradores):

Permiten conocer que clases componen el sistema y cuales interactúan entre sí.

Fases de XP:

La Programación Extrema consta de 4 fases, las cuales son:



Figura 4. Fases de XP (Bustamante & Rodríguez , 2018)

Planificación: La Metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores. El proyecto comienza recopilando las historias de usuarios, las que constituyen a los tradicionales casos de uso. Una vez obtenidas estas historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

Diseño: La Metodología XP hace especial énfasis en los diseños simples y claros. Un diseño simple se implementa más rápidamente que uno complejo, por lo que propone implementar el diseño más simple posible que funcione.

Codificación: En esta fase se desglosan las operaciones en las que se pueden implementar cada HU mediante las tareas de ingeniería o programación.

Pruebas: Se realizan pruebas unitarias, de aceptación y de carga y estrés.

Luego del estudio realizado sobre las metodologías existentes y las características del proyecto, se escogen las metodologías ágiles. Dentro de ellas se decide utilizar XP, la cual es más factible porque el tiempo de desarrollo es corto, el cliente está bien definido: Vertex, los requisitos de la tienda virtual pueden variar y es un equipo pequeño, compuesto por una persona.

1.4 Tecnología de *software* para el desarrollo de una tienda virtual

El sistema informático a desarrollar, debe ser una solución tecnológica simple, basada en la web, multiplataforma, y que facilita la comercialización de modelos tridimensionales. Para alcanzar este propósito se realizó un análisis de los factores técnicos y ambientales que afectan el desarrollo de las aplicaciones informáticas (Pressman & Maxim, 2015).

Entre los criterios técnicos considerados, en la elección de las herramientas y tecnologías, fue determinante la necesidad de una futura integración con la plataforma de videojuegos COSMOX. Adicionalmente, fueron tenidos en cuenta la portabilidad, el procesamiento complejo interno, la facilidad de la instalación, y la facilidad del cambio. Por otra parte, la familiaridad con el modelo del proyecto utilizado, y la experiencia del equipo de desarrollo con el lenguaje de programación y las tecnologías, constituyeron los criterios ambientales de mayor peso.

Teniendo en cuenta estos elementos, fueron seleccionadas las siguientes tecnologías y herramientas:

Lenguaje de programación Python

El lenguaje seleccionado para el desarrollo de este proyecto de *software* es Python. Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. Es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar *software* sin límites. Con el paso del tiempo, Python ha ido ganando adeptos gracias a su sencillez y a sus amplias posibilidades, sobre todo en los últimos años, ya que facilita trabajar con inteligencia artificial, *big data*, *machine learning* y *data science*, entre muchos otros campos en auge (Santander, 2021). (Versión 3.9.2)

Framework de desarrollo Django

Framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago (MDN Web Docs, 2022). Se empleará en el desarrollo del sistema en su versión 4.0.4.

Lenguaje de Marcado de Hipertexto HTML 5

Es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. HTML no es un lenguaje de programación; es un **lenguaje de marcado** que define la estructura de tu contenido. HTML consiste en una serie de elementos que usarás para encerrar diferentes partes del contenido para que se vean o comporten de una determinada manera. Las etiquetas de encierre pueden hacer de una palabra o una imagen un hipervínculo a otro sitio, se pueden cambiar palabras a cursiva, agrandar o achicar la letra, etc (MDN Web Docs H. , 2022).

Lenguaje de hojas de estilo en cascada CSS 3

Hojas de Estilo en Cascada (CSS, del inglés *Cascading Style Sheets*) un lenguaje de marcas enfocado a definir, crear y mejorar la presentación de un documento basado en HTML. El código CSS hace la vida más fácil al desarrollador *front-end* al separar las estructura de un documento HTML de su presentación. Dicho de otro modo el HTML actuaría como es esqueleto de la web, definiendo su estructura básica, y el CSS añadiría toda la capa de personalización sobre el que la web define su aspecto final (OpenWebinars, 2019).

Django REST Framework

Es un potente y flexible conjunto de herramientas para construir APIs web (Django Rest framework, 2022). (Versión 3.14.0)

Entorno de Desarrollo Integrado PyCharm

Primero se debe tener claro el concepto de IDE (Entorno de Desarrollo Integrado). Un IDE es un *software* que combina todas las herramientas que necesita un desarrollador para hacer su trabajo, sin tener que salir del mismo. Básicamente te permite enfocarte 100% en tu código, mientras que el IDE se encarga de todo lo demás. PyCharm es un IDE **enfocado en Python** que trae varias funcionalidades específicamente para este lenguaje (Manotoa, 2021). (Versión 2019.1.3)

Lenguaje de Base de datos PostgreSQL

Es un sistema para gestionar bases de datos de muy alto nivel. Es un sistema de base de datos relacional de alta disponibilidad. Es capaz de funcionar de manera estable en el servidor y, por lo tanto, resulta robusto, una de las principales características que buscan las empresas. Además, es consistente y tolerante a fallos. Es compatible con el modelo relacional, ya que asegura siempre su integridad referencial.

Algunas de sus principales características son: (Arsys, 2018)

- **Alta concurrencia:** Es capaz de atender a muchos clientes al mismo tiempo y entregar la misma información de sus tablas, sin bloqueos.
- **Soporte para múltiples tipos de datos de manera nativa:** Ofrece los tipos de datos habituales en los sistemas gestores, pero además muchos otros que no están disponibles en

otros competidores, como direcciones IP, direcciones MAC, *Arrays*, números decimales con precisión configurable, figuras geométricas, etc.

- **Soporte a triggers:** Permite definir eventos y generar acciones cuando estos se disparan.
- **Trabajo con vistas:** Esto quiere decir que pueden consultar los datos de manera diferente al modo en el que se almacenan.
- **Objeto-relacional:** Otra de sus principales características, que permite trabajar con sus datos como si fueran objetos y ofrece mecanismos de la orientación a objetos, como herencia de tablas.
- **Soporte para gran cantidad de lenguajes:** PostgreSQL es capaz de trabajar con funciones internas, que se ejecutan en el servidor, escritas en diversos lenguajes como C, C++, Java, PHP, Python o Ruby. Además, ofrece interfaces para ODBC y JDBC, así como interfaces de programación para infinidad de lenguajes de programación.

Gestor de Base de Datos PgAdmin

Es una plataforma de administración y desarrollo de código abierto para PostgreSQL y sus sistemas de administración de bases de datos relacionados. Está escrita en Python y jQuery, y es compatible con todas las características de PostgreSQL. Puede utilizar pgAdmin para tareas que van desde redactar consultas de SQL básicas hasta monitorear sus bases de datos y configurar arquitecturas de bases de datos avanzadas (DigitalOcean, 2020).

Herramienta de modelado: Visual Paradigm

Visual Paradigm for UML (VP-UML) es una herramienta de diseño UML y herramienta CASE del inglés (*Computer Aided Software Engineering*, Ingeniería de *Software* Asistida por Computadora) diseñada para ayudar al desarrollo de software. Soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML), y la notación de Modelado de Procesos de Negocio (BPMN) (Library, 2022).

Ofrece un completo conjunto de herramientas de equipos de desarrollo de *software* necesario para la captura de requisitos, planificación de *software*, planificación de controles, modelado de clases y modelado de datos. Además, permite modelar todos los tipos de diagramas de clases, código inverso y generar código desde los diagramas (Library, 2022).

Conclusiones parciales

En el presente capítulo se realizó un estudio de las principales tiendas virtuales de modelos tridimensionales que cuentan con un sistema de pago, se llegó a la conclusión de que por sus características no pueden utilizarse, pero brindaron funcionalidades para desarrollar una propia. Con la realización del *software* se espera cumplir con la necesidad de comercializar los modelos tridimensionales.

De igual forma fueron analizadas diferentes herramientas y tecnologías que constituyen la propuesta tecnológica para el desarrollo de la Tienda virtual. Se definió como lenguaje de programación Python, *Framework* web Django, de Entorno de Desarrollo Integrado se usará PyCharm que es enfocado en Python, y PostgreSQL y PgAdmin para el diseño y administración de la base de datos.

Debido a las características del proyecto, se hace necesario utilizar una metodología ágil, que permita a los miembros del equipo de desarrollo tener el *software* listo en el menor tiempo posible. Por tal motivo se escogió como metodología de desarrollo XP, la cual permite una estrecha relación con el cliente, facilitando el trabajo del desarrollador.

Capítulo 2. Propuesta de solución de la tienda virtual de modelos tridimensionales COSMOX

En este capítulo se realiza una descripción de la propuesta de solución. Se definirán los requisitos funcionales y no funcionales, así como las Historias de usuario correspondientes. Además, se escoge y describe la arquitectura a utilizar; se evidencian las funcionalidades de cada clase y sus relaciones mediante el uso de tarjetas CRC. También se especifica el uso de los patrones GRASP en cada clase y se escogen los patrones GoF más adecuados para efectuar la programación de la aplicación.

2.1. Propuesta de solución

En la presente investigación se propone realizar una plataforma web de comercio electrónico que permita comercializar los modelos tridimensionales desarrollados en el centro VERTEX. El sistema contará con una interfaz principal en la cual se presentarán todos los productos publicados por los usuarios, los mismos podrán ser filtrados por su categoría, para una mayor facilidad de búsqueda. Cada producto podrá ser agregado a una cesta para su posterior facturación.

Además la tienda contará con una funcionalidad donde el usuario podrá agregar sus productos favoritos para acceder a ellos de forma más rápida. El administrador podrá realizar la gestión de las categorías a la que pertenecen los productos. Como parte de las necesidades del centro se le permite al usuario la creación de un perfil en el cual pueda gestionar sus diseños para su posterior venta, utilizando los mecanismos de pago establecidos por el Gobierno cubano.

Debido a las restricciones legales que existen en nuestro país, se tiene que firmar un contrato con las empresas para que puedan utilizar las plataformas de pago. Actualmente es imposible la integración con una plataforma de pago hasta no culminar el proceso de contratación. Por lo tanto se decidió utilizar un servicio de pago sencillo, cuya única función será informar si se pagó o no.

2.2 Requerimientos funcionales

Los requerimientos funcionales definen las acciones que debe realizar el sistema, son capacidades o condiciones que debe cumplir, cómo debe comportarse en situaciones específicas. En algunos casos también pueden plantear explícitamente qué no debe hacer el sistema (Sommerville, I. , 2011). A continuación, se exponen los diferentes Requerimientos funcionales (RF) planteados por el usuario:

Tabla 2. Requerimientos funcionales

Número	Nombre	Descripción	Prioridad	Complejidad
RF1	Autenticar usuario	El sistema debe permitir al usuario autenticarse en la tienda.	Alta	Alta
RF2	Registrar usuario	El sistema debe permitir registrar un usuario.	Media	Media

RF3	Listar categorías	El sistema debe mostrar las categorías de productos.	Alta	Baja
RF4	Adicionar categoría	El sistema debe permitir al administrador adicionar una nueva categoría.	Alta	Baja
RF5	Modificar categoría	El sistema debe permitir al administrador modificar la información de una categoría.	Alta	Baja
RF6	Eliminar categoría	El sistema debe permitir al administrador eliminar una categoría.	Alta	Baja
RF7	Actualizar perfil	El sistema debe permitir al usuario actualizar su perfil.	Media	Media
RF8	Filtra por categoría	El sistema debe permitir al usuario filtrar los productos por categoría.	Alta	Baja
RF9	Listar productos	El sistema debe mostrar al usuario el listado de los productos disponibles en la tienda.	Alta	Baja
RF10	Listar productos de cliente	El sistema debe mostrar al usuario el listado de sus productos.	Alta	Baja
RF11	Adicionar producto	El sistema debe permitir adicionar un nuevo producto.	Alta	Baja
RF12	Modificar producto	El sistema debe permitir modificar la información de un producto.	Alta	Baja
RF13	Eliminar producto	El sistema debe permitir eliminar un producto.	Alta	Baja
RF14	Buscar producto	El sistema debe permitir buscar productos a partir de un criterio de búsqueda introducido por el usuario.	Media	Media
RF15	Administrar favoritos	El sistema debe permitir agregar y eliminar un producto a favoritos.	Alta	Baja
RF16	Listar productos favoritos	El sistema debe permitir listar los productos favoritos de un cliente.	Alta	Baja
RF17	Adicionar producto a la cesta	El sistema debe permitir adicionar productos a la cesta.	Alta	Alta
RF18	Administrar cesta de productos	El sistema debe permitir eliminar producto y facturar la compra.	Alta	Media

RF19	Ver detalles de producto	El sistema debe permitir visualizar los detalles del producto.	Baja	Baja
RF20	Realizar pago	El sistema debe permitir realizar el proceso de pago.	Alta	Alta
RF21	Listar productos comprados	El sistema debe mostrar al usuario el listado de los productos comprados.	Baja	Baja
RF22	Descargar archivo	El sistema debe permitir al usuario descargar un archivo de un producto comprado.	Alta	Media

2.3 Requerimientos no funcionales

Los requerimientos no funcionales (RnF) son propiedades o cualidades que el producto debe tener. Son limitaciones sobre servicios o funciones que ofrece el sistema; se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema (Sommerville, I. , 2011). Se definieron como RnF los siguientes:

Usabilidad

- RnF1: El sistema podrá ser usado por todo aquel personal que posea un nivel básico en computación y en la navegación por sitios web.

Interfaz

- RnF2: Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La entrada de datos incorrecta será detectada claramente e informada al usuario.

Software

- RnF3: El sistema debe ser multiplataforma.
- RnF4: El sistema debe ser capaz de ejecutarse en los navegadores Mozilla Firefox, Google Chrome, Safari y el Opera.

Seguridad

- RnF5: Asegurar que las funciones del sistema sean visibles según el nivel de acceso que tenga el usuario autenticado.

Eficiencia

- RnF6: El sistema debe permitir que los usuarios interactúen con él de manera concurrente. El tiempo de demora de una petición al servidor debe ser menor a cinco segundos.

Soporte

- RnF7: El sistema tiene la opción de ser mejorado, así como de incorporarle nuevas funcionalidades en caso de ser necesarias.

2.4 Historia de Usuario (HU)

Las HU describen las características principales y el comportamiento del sistema a desarrollar. Se generaron un total de 22 HU, a continuación, se muestran las correspondientes a los requisitos funcionales “Registrar usuario”, “Adicionar producto”, “Adicionar producto a la cesta”, “Ver detalles de producto”, “Realizar pago”, “Listar productos comprados” y “Descargar archivo”, el resto se encuentra en el anexo A.1.

Tabla 3. Historia de Usuario “Registrar usuario”.

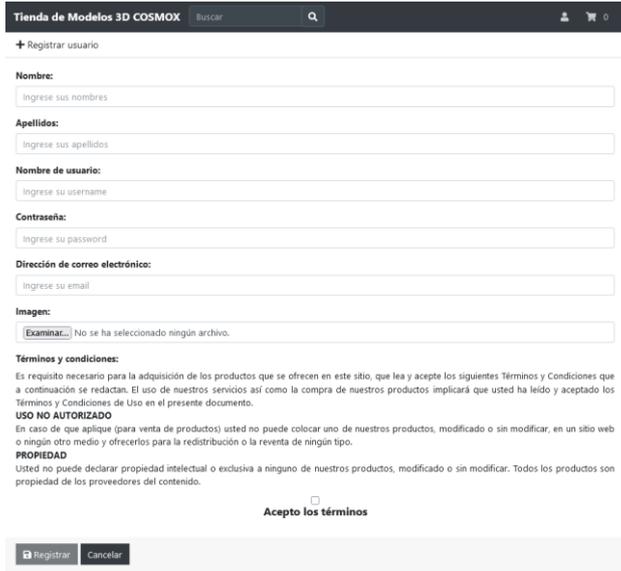
Historia de usuario	
Número: 2	Nombre: Registrar usuario
Usuario: Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1.5	Iteración asignada: 1
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Los usuarios se pueden registrar en el sitio.	
Observaciones:	
<ol style="list-style-type: none"> 1. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. 2. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse. 	
Interfaz:	
	

Tabla 4. Historia de Usuario “Adicionar producto”.

Historia de usuario	
Número: 11	Nombre: Adicionar producto

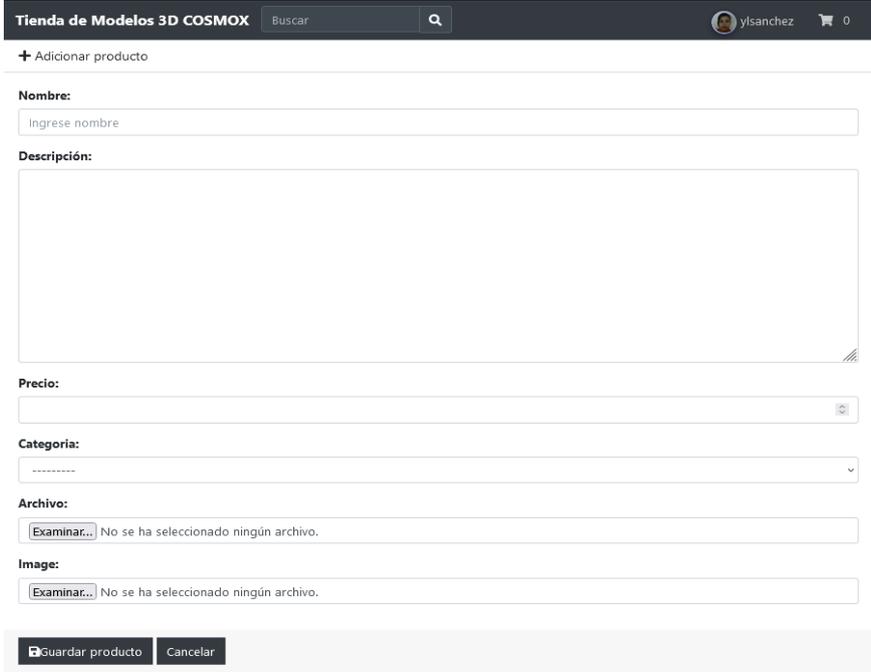
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.5	Iteración asignada: 2
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Cuando un cliente accede a la opción Mis productos, el sistema muestra un listado de sus productos y brinda la posibilidad de adicionar uno nuevo. El sistema muestra la interfaz adicionar producto, al seleccionar el botón Nuevo registro el sistema regresa al listado de sus productos.	
Observaciones: 1. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse. 2. Si el usuario introduce un producto existente el sistema le muestra un mensaje “Producto existente”.	
Interfaz:	
	

Tabla 5. Historia de Usuario “Adicionar producto a la cesta”.

Historia de usuario	
Número: 17	Nombre: Adicionar producto a la cesta
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2.0	Iteración asignada: 3
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema brinda la posibilidad de adicionar el producto a la cesta.	
Observaciones: 1. Si al presionar el botón “Adicionar producto” el usuario no está autenticado el sistema lo	

redirecciona al autenticar.

Interfaz:



Tabla 6. Historia de Usuario “Ver detalles de producto”.

Historia de usuario	
Número: 19	Nombre: Ver detalles de producto
Usuario: Cliente	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema permite ver los detalles del producto seleccionado.	
Observaciones: No aplica	
Interfaz:	
	

Tabla 7. Historia de Usuario “Realizar pago”.

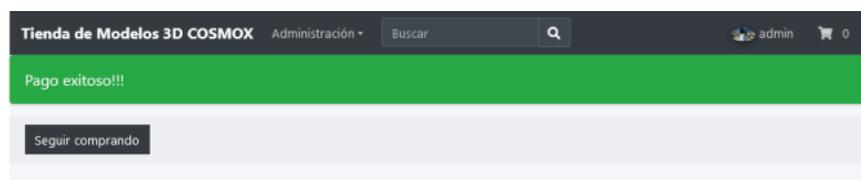
Historia de usuario	
Número: 20	Nombre: Realizar pago
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2.5	Iteración asignada: 4
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema permite realizar el pago de los productos agregados a la cesta.	
Observaciones: No aplica	
Interfaz:	
	

Tabla 8. Historia de Usuario “Listar productos comprados”.

Historia de usuario	
Número: 21	Nombre: Listar productos comprados
Usuario: Cliente	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 4
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema muestra un listado con los productos comprados por el cliente.	
Observaciones: No aplica	
Interfaz:	

Tabla 9. Historia de Usuario “Descargar archivo”.

Historia de usuario	
Número: 22	Nombre: Descargar archivo
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1.0	Iteración asignada: 4
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema permite descargar el archivo del producto comprado.	
Observaciones: No aplica	
Interfaz:	

2.5 Estimación de esfuerzo por Historias de usuario

Se realiza la estimación de esfuerzo por cada HU para lograr un correcto desarrollo del sistema.

Tabla 10. Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Autenticar usuario	1.0
	2	Registrar usuario	1.5
	3	Listar categorías	1.0
	4	Adicionar categoría	1.0
	5	Modificar categoría	1.0
	6	Eliminar categoría	1.0
	7	Actualizar perfil	1.5
	8	Filtra por categoría	1.0
2	9	Listar productos	0.5
	10	Listar productos de cliente	0.5
	11	Adicionar producto	1.5
	12	Modificar producto	1.0
	13	Eliminar producto	1.0
	14	Buscar producto	1.0
	15	Administrar favoritos	1.5
	16	Listar productos favoritos	1.0
3	17	Adicionar producto a la cesta	2.0
	18	Administrar cesta de productos	2.0
	19	Ver detalles de producto	0.5
4	20	Realizar pago	2.5
	21	Listar productos comprados	0.5
	22	Descargar archivo	1.0
Total			25.5

2.5.1. Desarrollo del plan de iteraciones

En esta fase se elabora un plan de iteraciones, en que se especifica la prioridad con la que se implementarán las HU según la iteración correspondiente. La siguiente tabla refleja la duración (en semanas) y el orden en que se deberán implementar las iteraciones:

Tabla 11. Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Autenticar usuario	9.0
	2	Registrar usuario	
	3	Listar categorías	

	4	Adicionar categoría	
	5	Modificar categoría	
	6	Eliminar categoría	
	7	Actualizar perfil	
	8	Filtra por categoría	
2	9	Listar productos	8.0
	10	Listar productos de cliente	
	11	Adicionar producto	
	12	Modificar producto	
	13	Eliminar producto	
	14	Buscar producto	
	15	Admisnistrar favoritos	
3	17	Adicionar producto a la cesta	4.5
	18	Administrar cesta de productos	
	19	Ver detalles de producto	
4	20	Realizar pago	4.0
	21	Listar productos comprados	
	22	Descargar archivo	
Total			25.5

2.5.2 Plan de entregas

En el plan de entregas mostrado a continuación se especifican la fecha de inicio y de fin de cada iteración.

Tabla 12. Plan de entrega de las iteraciones

Iteraciones	Fecha inicio	Fecha fin
1	9 de mayo de 2022	9 de julio de 2022
2	10 de julio de 2022	1 de septiembre de 2022
3	2 de septiembre de 2022	2 de octubre de 2022
4	3 de octubre de 2022	31 de octubre de 2022

2.6 Arquitectura de la aplicación

La Arquitectura de *Software* es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución (Billy Reynoso , 2022). A grandes rasgos es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se le percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. En la creación del sistema se utiliza la arquitectura basada en el patrón Modelo Vista *Template* (MVT), el cual no es más que una variación que implementa Django del

patrón Modelo Vista Controlador (MVC), MVT es un patrón de diseño de *software* para desarrollar una aplicación web.

La estructura MVT tiene las siguientes tres partes:

Modelo: la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos (Uniwebsidad, 2022).

Vista: la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelo y las plantillas (Uniwebsidad, 2022).

Template: la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento (Uniwebsidad, 2022).

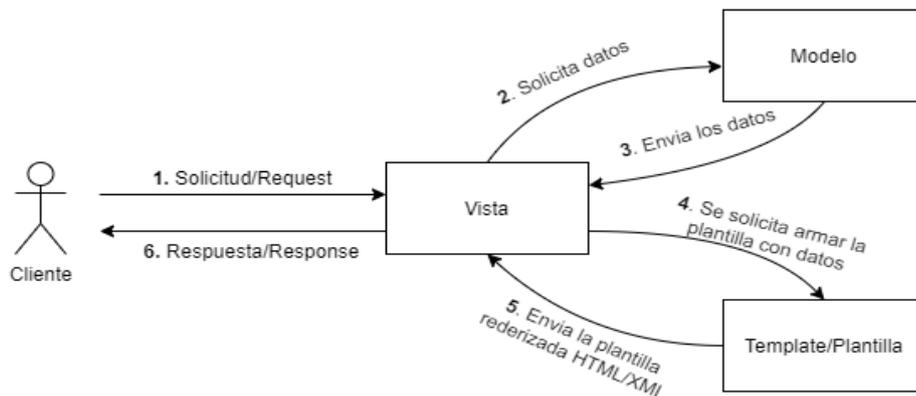


Figura 5. MVT (Caicedo, 2021).

1. El cliente (a través del navegador web) genera una solicitud.
2. La vista recibe esa solicitud y solicita los datos al modelo.
3. El modelo busca y le envía los datos a la vista.
4. La vista necesita mostrar los datos en formato HTML y se la pasa al *template*/plantilla para que arme la plantilla.
5. El *template*/plantilla arma la plantilla HTML y la envía a la vista.
6. La vista muestra el HTML debidamente armado al cliente. Si el cliente es un navegador web, agarra el HTML y lo renderiza para mostrarle la información lo más amigable posible al cliente final.

2.7 Diseño de la base de datos

En el modelo de datos se definen los conceptos que son utilizados en el sistema y que describen la estructura de la base de datos diseñada. En este modelo se representan los datos, sus atributos y tipos, sus relaciones.

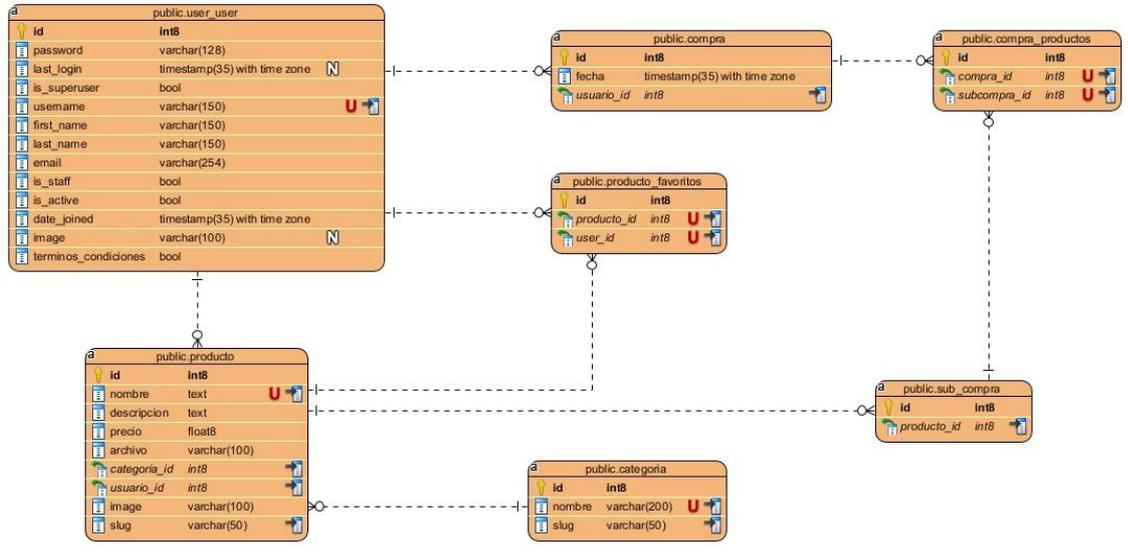


Figura 6. Modelo de datos.

Descripción de las tablas

Se identificaron 7 tablas, a continuación se detallan 2 de ellas, encontrándose el resto en los anexos del 1 al 5.

Tabla 13. Descripción de la tabla producto

Nombre: producto		
Descripción: almacena todos los productos		
Atributo	Tipo	Descripción
id	long	Identificador
nombre	string	Nombre literal del producto
descripcion	string	Descripción del producto
precio	double	Precio del producto
archivo	string	Archivo del producto
categoria_id	long	Llave foránea que hace referencia al identificador de la tabla categoria.
usuario_id	long	Llave foránea que hace referencia al identificador de la tabla user.
image	string	Imagen del producto
slug	string	Atributo por el que se filtra la búsqueda

Tabla 14. Descripción de la tabla categoría

Nombre: categoria		
Descripción: almacena todas las categorías		
Atributo	Tipo	Descripción
id	long	Identificador
nombre	string	Nombre literal de la categoría
slug	string	Atributo por el que se filtra la búsqueda

2.8 Tarjetas de Contenido, Responsabilidad y Colaboración

La metodología XP se utiliza tarjetas CRC para describir las relaciones entre clases.

Tabla 15. Tarjeta CRC Autenticar usuario

Tarjeta CRC	
Clase: LoginFormView	
Responsabilidad	Colaboración
Validar credenciales.	LoginView
Logueo del sistema.	User

Tabla 16. Tarjeta CRC Registrar usuario

Tarjeta CRC	
Clase: UserCreateView	
Responsabilidad	Colaboración
Creación del perfil de un usuario.	CreateView
	User

Tabla 17. Tarjeta CRC Listar categorías

Tarjeta CRC	
Clase: CategoriaListView	
Responsabilidad	Colaboración
Listar las categorías.	ListView
	Categoria

Tabla 18. Tarjeta CRC Adicionar categoría

Tarjeta CRC	
Clase: CategoriaCreateView	
Responsabilidad	Colaboración
Crear categorías.	CreateView
	Categoria

Tabla 19. Tarjeta CRC Modificar categoría

Tarjeta CRC	
-------------	--

Clase: CategoriaUpdateView	
Responsabilidad	Colaboración
Actualizar una categoría.	UpdateView Categoría

Tabla 20. Tarjeta CRC Listar productos

Tarjeta CRC	
Clase: IndexView	
Responsabilidad	Colaboración
Cargar los datos de la interfaz principal. Listar productos y categorías.	TemplateView Producto Categoría

Tabla 21. Tarjeta CRC Listar productos de cliente

Tarjeta CRC	
Clase: ProductoListView	
Responsabilidad	Colaboración
Listar los productos de un cliente.	ListView Producto

Tabla 22. Tarjeta CRC Adicionar producto

Tarjeta CRC	
Clase: ProductCreateView	
Responsabilidad	Colaboración
Crear un producto.	CreateView Producto Categoría

Tabla 23. Tarjeta CRC Modificar producto

Tarjeta CRC	
Clase: ProductUpdateView	
Responsabilidad	Colaboración

Modificar un producto.	UpdateView Producto Categoria
------------------------	-------------------------------------

Tabla 24. Tarjeta CRC Actualizar perfil

Tarjeta CRC	
Clase: UserUpdateView	
Responsabilidad	Colaboración
Actualizar el perfil de un usuario.	UpdateView User

2.9 Patrones de diseño

Los patrones de diseño son modelos de código que resuelven problemas clásicos. Son soluciones a problemas de diseño de *software* que puedes encontrar en una aplicación del mundo real. Un Patrón de Diseño no es un código que esté listo para ser usado en tu aplicación, sino que es un modelo que puedes usar para resolver un problema. Son neutrales en cuanto al lenguaje, por lo que pueden aplicarse a cualquier lenguaje que soporte la orientación a objetos (Siebert Domareski, 2020).

Patrones de Asignación de Responsabilidades (GRASP) Es en sistema orientado a objetos se compone de objetos que envían mensajes a otros objetos para que lleven a cabo las operaciones requeridas. Los diagramas de interacción describen gráficamente estas operaciones, a partir de los objetos en interacción, que se responsabilizan de una actividad determinada (Hardy Demestre & Hernández González, 2019).

Los patrones GRASP utilizados en la solución propuesta fueron:

Experto: Clase que tiene la responsabilidad de ejecutar una acción determinada y cuenta con el acceso a los datos necesarios (Hardy Demestre & Hernández González, 2019). Una buena práctica es que este patrón se aplique a todas las clases que conforman el sistema y un ejemplo de esto es la clase CategoriaCreateView, cuya única función es crear categorías.

Creador: este patrón es quien guía el proceso de asignación de responsabilidades relacionadas con la creación de objetos. Tiene el objetivo de asignarle a la clase B la responsabilidad de crear una instancia de la clase A (Hardy Demestre & Hernández González, 2019). Este patrón se pone de manifiesto en la clase ProductCreateView, al ser esta la encargada de crear todas las instancias de los objetos que se manejan a la hora de crear un nuevo producto.

Bajo acoplamiento: Plantea que se debe poder reutilizar las funcionalidades de las distintas clases, con un nivel de dependencia mínima. Este patrón se evidencia en todas las aplicaciones web que funcionen sobre Django, pues cada pieza de las aplicaciones tiene un propósito clave, que puede modificarse sin afectar otras piezas. Por ejemplo, se puede cambiar la URL de cierta parte de la aplicación sin tener que afectar la implementación subyacente o se puede modificar el HTML de una página sin tener que tocar el código Python que la renderiza (Hardy Demestre & Hernández González, 2019).

Alta cohesión: Asigna responsabilidades de manera tal que la cohesión siga siendo alta, o sea que las funcionalidades de las clases estén altamente relacionadas de forma tal que exista una colaboración entre ellas para compartir el esfuerzo y no caiga todo el peso sobre una única clase. Usar este patrón simplifica el mantenimiento y favorece el bajo acoplamiento (Hardy Demestre & Hernández González, 2019).

Controlador: Permite manejar todos los eventos del sistema, al servir de intermediario entre las interfaces y el algoritmo que las implementa (Hardy Demestre & Hernández González, 2019). Este patrón se evidencia en la clase `ProductCreateView`, pues recibe los datos del producto y los salva en la base de datos.

Patrones GOF: Patrones publicados por Gamma, Helm, Johnson y Vlossodes en 1995: patrones de la banda de los cuatro (del inglés, Gang of Four). Esta serie de patrones permiten ampliar el lenguaje, aprender nuevos estilos de diseño y además introducir más notación UML .

Los patrones de diseño de GoF se dividen en tres categorías: Patrones de Creación para la creación de objetos; Patrones Estructurales para proporcionar relaciones entre los objetos; y finalmente, Patrones de Comportamiento para ayudar a definir cómo interactúan los objetos (Hardy Demestre & Hernández González, 2019). A continuación, se describen los patrones GoF utilizados en la solución propuesta:

Decorador: Patrón estructural que extiende la funcionalidad de un objeto dinámicamente de manera tal que es transparente a sus clientes, utiliza una instancia de una subclase de la clase original que delega las operaciones al objeto original (Hardy Demestre & Hernández González, 2019). Este patrón se evidencia en la clase `LoginRequired` (decorador que trae django por defecto, para acceder a una clase).

Observer: Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos. También conocido como "*spider*" define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se encarga de notificar este cambio a todos los otros dependientes. Este patrón se pone de manifiesto cuando se modifica un concepto Categoría, el observador notifica el cambio a los Productos que la contienen, actualizando el cambio.

Conclusiones parciales

En el presente capítulo, se definió la propuesta de solución para la problemática planteada, detallada a través de los requisitos funcionales y no funcionales y las Historias de Usuario. La utilización del patrón arquitectónico y los patrones de diseño GRASP y GOF contribuyó al diseño de la aplicación, proporcionando una estructura para la misma y posibilitando el empleo de buenas prácticas de programación y la reutilización de código.

Capítulo 3. Implementación y pruebas de la tienda virtual de modelos tridimensionales COSMOX

En el presente capítulo se describen las tareas correspondientes a las fases de implementación y pruebas del software. En la primera, se elaboran las tareas de ingeniería y programación necesarias para la implementación de las HU, las mismas se describen desde el punto de vista del programador. Para la validación del *software*, se realizan las pruebas de aceptación, que son definidas por la metodología XP y pueden ser ejecutadas en cualquier etapa de desarrollo.

3.1 Tareas de ingeniería

La metodología XP, propone, que se describan las tareas de ingeniería necesarias para dar solución a cada HU. Las mismas, son descritas desde el punto de vista del programador, por lo que una HU puede tener más de una tarea de ingeniería asociada. A continuación se muestran las tareas de ingeniería agrupadas por iteración:

Iteración 1

Dentro de la primera iteración se encuentran las funcionalidades “Autenticar usuario”, “Registrar usuario”, “Listar categorías”, “Adicionar categoría”, “Modificar categoría”, “Eliminar categoría”, “Actualizar perfil” y “Filtra por categoría”, correspondientes a las historias de usuario 1, 2, 3, 4, 5, 6, 7 y 8. Las tareas definidas para la iteración son:

- Tarea No.1: Autenticar usuario
- Tarea No 2: Registrar usuario
- Tarea No.3: Listar categorías
- Tarea No.4: Adicionar categoría
- Tarea No.5: Modificar categoría
- Tarea No.6: Eliminar categoría
- Tarea No.7: Actualizar perfil
- Tarea No.8: Filtra por categoría

A continuación, se muestran las correspondientes a las tareas 1, y 2, el resto se encuentra en el anexo A.3:

Tabla 25. Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 1
Nombre de la tarea: Autenticar usuario	

Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 09/05/22	Fecha de fin: 15/05/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de LoginView, que sea capaz de autenticar el usuario en la tienda.	

Tabla 26. Tarea de ingeniería # 2

Tarea	
Número de tarea: 2	Número de Historia de usuario: 2
Nombre de la tarea: Registrar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha de inicio: 16/05/22	Fecha de fin: 26/05/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de CreateView, que sea capaz de registrar un usuario en la tienda.	

Iteración 2

Dentro de la segunda iteración se encuentran las funcionalidades “Listar productos”, “Listar productos de cliente”, “Adicionar producto”, “Modificar producto”, “Eliminar producto”, “Buscar producto”, “Administrar favoritos” y “Listar productos favoritos” correspondientes a las historias de usuario 9, 10, 11, 12, 12, 14, 15 y 16. Las tareas definidas para la iteración son:

- Tarea No.9: Listar productos
- Tarea No.10: Listar productos de cliente
- Tarea No.11: Adicionar producto
- Tarea No.12: Modificar producto
- Tarea No.13: Eliminar producto
- Tarea No.14: Buscar producto
- Tarea No.15: Agregar a favorito
- Tarea No.16: Eliminar favorito
- Tarea No.17: Listar productos favoritos

A continuación, se muestran las correspondientes a las tareas 9 y 10, el resto se encuentra en el anexo A.3:

Tabla 27. Tarea de ingeniería # 9

Tarea	
Número de tarea: 9	Número de Historia de usuario: 9
Nombre de la tarea: Listar productos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 10/07/22	Fecha de fin: 13/07/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de ListView, que sea capaz de listar los productos registrados en la tienda.	

Tabla 28. Tarea de ingeniería # 10

Tarea	
Número de tarea: 10	Número de Historia de usuario: 10
Nombre de la tarea: Listar productos de cliente	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 14/07/22	Fecha de fin: 16/07/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de ListView, que sea capaz de listar los productos del cliente autenticado, registrados en la tienda.	

Iteración 3

Dentro de la tercera iteración se encuentran las funcionalidades “Adicionar producto a la cesta”, “Administrar cesta de productos” y “Ver detalles de producto” correspondientes a las historias de usuario 17, 18 y 19. Las tareas definidas para la iteración son:

- Tarea No.18: Adicionar producto a la cesta
- Tarea No.19: Eliminar productos de la cesta
- Tarea No.20: Limpiar cesta
- Tarea No.21: Ver detalles de producto

A continuación, se muestran las correspondientes a las tareas 18 y 19, el resto se encuentra en el anexo A.3:

Tabla 29. Tarea de ingeniería # 18

Tarea	
Número de tarea: 18	Número de Historia de usuario: 17
Nombre de la tarea: Adicionar producto a la cesta	
Tipo de tarea: Desarrollo	Puntos estimados: 2.0
Fecha de inicio: 02/09/22	Fecha de fin: 15/09/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de adicionar un producto a la cesta del cliente autenticado.	

Tabla 30. Tarea de ingeniería # 19

Tarea	
Número de tarea: 19	Número de Historia de usuario: 18
Nombre de la tarea: Eliminar productos de la cesta	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 16/09/22	Fecha de fin: 22/09/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de eliminar un producto de la cesta del cliente autenticado.	

Iteración 4

Dentro de la cuarta iteración se encuentra la funcionalidad “Realizar pago”, “Listar productos comprados” y “Descargar archivo” correspondiente a las historias de usuario 20, 21 y 22. Las tareas definidas para la iteración son:

- Tarea No.22: Realizar pago
- Tarea No.23: Listar productos comprados
- Tarea No.24: Descargar archivo

Tabla 31. Tarea de ingeniería # 22

Tarea	
Número de tarea: 22	Número de Historia de usuario: 20
Nombre de la tarea: Realizar pago	
Tipo de tarea: Desarrollo	Puntos estimados: 2.5
Fecha de inicio: 03/10/22	Fecha de fin: 19/10/22

Programador responsable: Yipsi Lourdes Sánchez Mesa
Descripción: Efectuar el pago mediante los mecanismos de pago establecidos por el Gobierno cubano.

Tabla 32. Tarea de ingeniería # 23

Tarea	
Número de tarea: 23	Número de Historia de usuario: 21
Nombre de la tarea: Listar productos comprados	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 20/10/22	Fecha de fin: 23/10/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de listar los productos comprados por cliente autenticado.	

Tabla 33. Tarea de ingeniería # 24

Tarea	
Número de tarea: 24	Número de Historia de usuario: 22
Nombre de la tarea: Descargar archivo	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 24/09/22	Fecha de fin: 31/10/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de descargar el archivo asociado a un producto comprado.	

3.2 Interfaces de la tienda virtual

Finalmente se obtuvo una tienda que permite publicar los modelos tridimensionales desarrollados en el Centro de Tecnologías Interactivas (VERTEX). La tienda contribuye además a fomentar el interés de desarrollo de los mismos por parte de la comunidad universitaria, ya que otorga la posibilidad a los desarrolladores de mostrar los modelos que desarrollan. Además, se garantiza la disponibilidad de un sitio propio para la UCI. Teniendo en cuenta que Internet es una herramienta potente sin importar en que ámbito se maneje, esta tienda virtual es otro recurso que facilita a la comunidad no tener que buscar y publicar en tiendas internacionales.

3.2.1. Interfaz “Listar productos”

Cuando el usuario accede a la tienda se muestra un listado de todos los productos registrados en la misma. En los productos se muestra: la imagen del producto, el nombre, el precio, el ícono de añadir a favorito y el botón de adicionar a la cesta, como se muestra a continuación en la Figura 6.

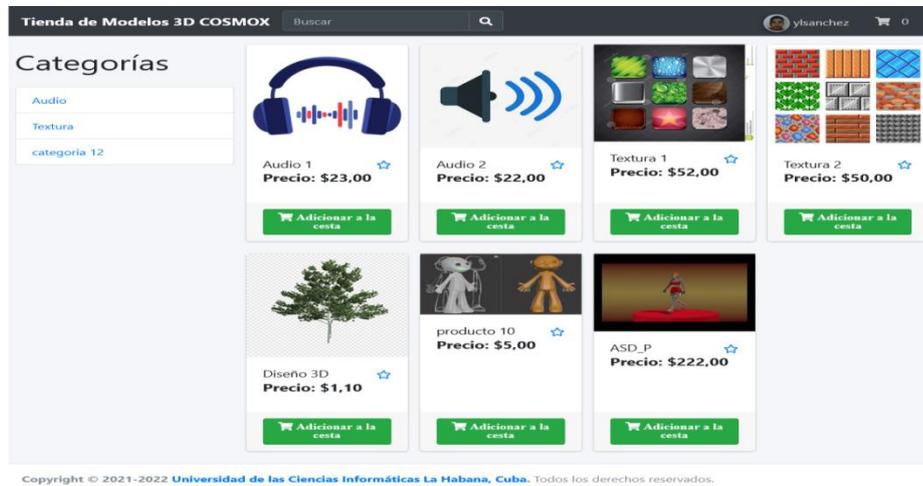


Figura 7. Interfaz “Listar productos”

3.2.2. Interfaz “Listar categorías”

Una vez autenticado un usuario administrador en la tienda tiene la posibilidad de gestionar categorías. Esta acción se lleva a cabo cuando accede a la sección de “Administración”, y escoge la opción “Categorías”. El sistema muestra un listado de todas las categorías registradas. En el listado de categorías se muestra: el nombre, además da la posibilidad de adicionar una nueva categoría; así como, modificar y eliminar las existentes. Por otra parte el usuario tiene la posibilidad de buscar, aumentar o disminuir la cantidad de categorías que se muestran en la lista y navegar por las páginas, como se muestra a continuación en la Figura 7.

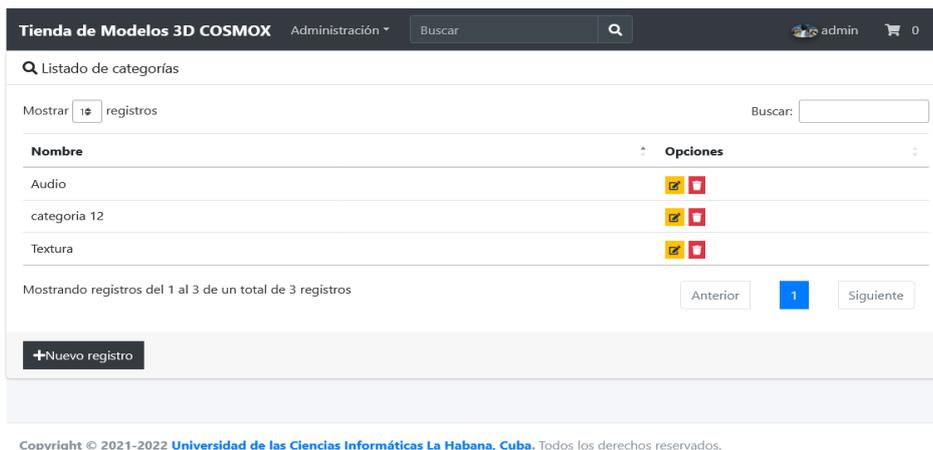


Figura 8. Interfaz “Listar categorías”

3.2.3. Interfaz “Adicionar producto”

Cuando el usuario se autentica y accede a la opción “Mis productos” se muestra un listado de todos los productos del cliente, el sistema le da la posibilidad de adicionar uno nuevo. Para la creación de

un producto se registra: el nombre, descripción, el precio, la categoría a la que pertenece, el archivo del producto y la imagen del mismo, como se muestra a continuación en la Figura 8.

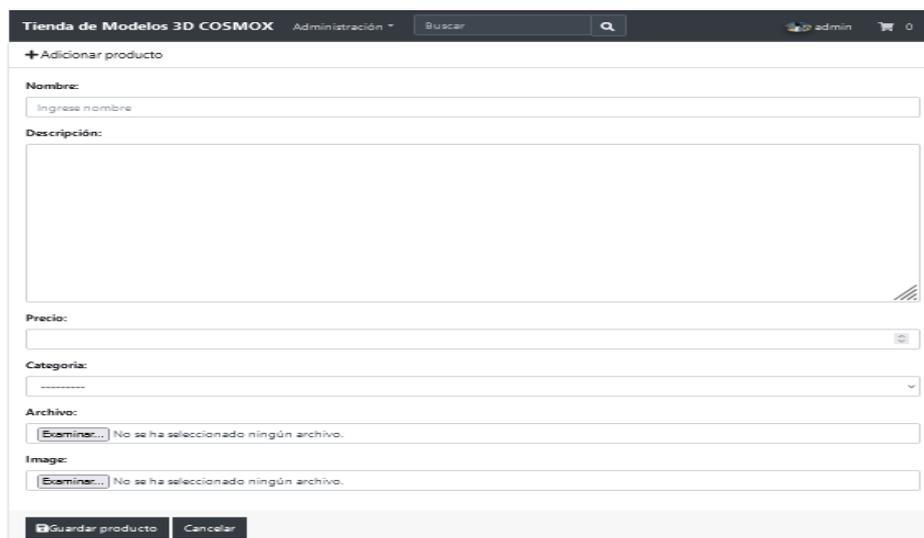


Figura 9. Interfaz “Creación de un producto”

3.2.4. Interfaz “Listar productos favoritos”

Cuando el usuario se autentica y accede a la opción “Favoritos” se muestra un listado de todos los productos que el cliente marcó como favoritos. El sistema brinda la posibilidad de desmarcarlos del listado o adicionarlos a la cesta, como se muestra a continuación en la Figura 9.

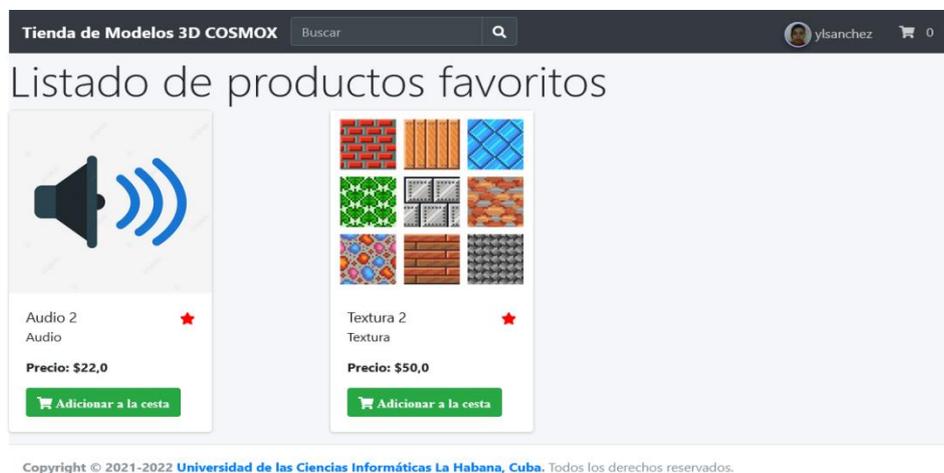
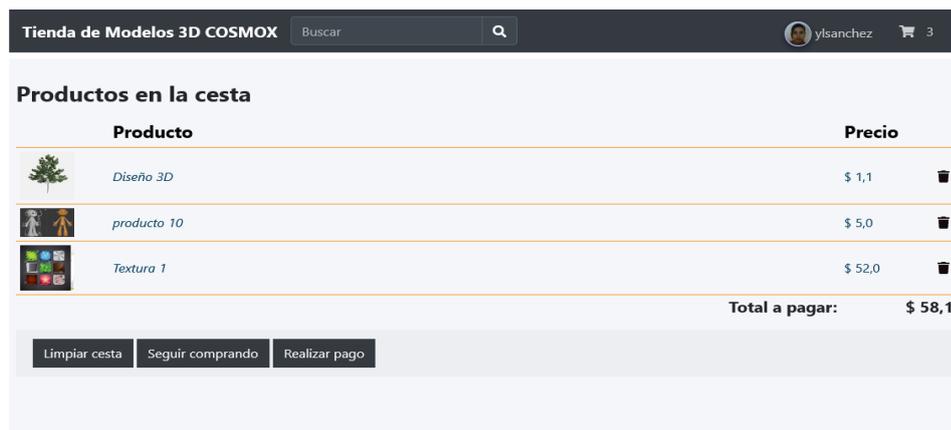


Figura 10. Interfaz “Listar productos favoritos”

3.2.5. Interfaz “Administrar cesta de productos”

Cuando el usuario se autentica y adiciona productos a su cesta, el sistema le brinda la posibilidad de administrar la misma. Esta acción se lleva a cabo cuando accede a la opción “Cesta”, donde se muestra un listado de todos los productos que el cliente adicionó para comprar. El sistema brinda la posibilidad de eliminar un producto del listado o limpiar la cesta, además de poder seguir comprando y de realizar el pago, como se muestra a continuación en la Figura 10.



Copyright © 2021-2022 Universidad de las Ciencias Informáticas La Habana, Cuba. Todos los derechos reservados.

Figura 11. Interfaz “Administrar cesta de productos”

3.3 Pruebas de software

Las pruebas de *software* son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier etapa del desarrollo del sistema, desde su creación hasta su puesta en producción. Lo interesante de las pruebas es que se puedan ejecutar de manera automática, para determinar en cualquier momento si tenemos una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin que nos demos cuenta (Turrado, 2020).

Las pruebas de *software* son los procesos que permiten verificar y revelar la calidad de un producto de *software*. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un sistema informático. Básicamente es una fase en el desarrollo de *software*, consistente en probar las aplicaciones construidas. Para determinar el nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto a las especificaciones iniciales del sistema.

Las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo, agrupadas por niveles de prueba aplicada por etapas (Jacobson, I., Booch, G., & Rumbaugh J. , 2000). Se distinguen los siguientes niveles de pruebas:

- **Prueba de desarrollador:** Es la prueba diseñada e implementada por el equipo de desarrollo.
- **Prueba independiente:** Es la prueba que es diseñada e implementada por alguien independiente del grupo de desarrolladores.
- **Prueba de unidad:** Se centra en el esfuerzo de verificación de la unidad más pequeña del diseño del *software*, el componente o módulo de *software*. El objetivo es comprobar que el módulo, entendido como una unidad funcional, está correctamente codificado.

- **Prueba de integración:** Es una técnica sistemática para construir la arquitectura del *software*, mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. Es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso.
- **Prueba de sistema:** Esta prueba tiene como objetivo verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las operaciones apropiadas funcionando como un todo.
- **Prueba de aceptación:** Es la prueba final antes del despliegue del sistema. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales.

3.3.1 Pruebas unitarias

Una prueba unitaria es la comprobación de un módulo determinado dentro de un sistema. Son llevadas a cabo por los programadores encargados de cada módulo, los cuales aseguran que un determinado módulo cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema.

Los programadores deben realizar estas pruebas cuando la interfaz de un método de la aplicación no es clara, la implementación es complicada, para probar entradas y condiciones inusuales, luego de modificar algo. Éstas deben contemplar cada módulo del sistema que pueda generar fallas. Que todo código liberado pase correctamente las pruebas unitarias habilita que funcione la propiedad colectiva del código (Sommerville, I. , 2011).

Las pruebas unitarias no descubrirán todos los errores del código, por definición, sólo examinan porciones de código por separado. Esto quiere decir que los errores de integración, problemas de rendimiento y otros que afectan al sistema en su conjunto, no serán detectados.

Como parte del desarrollo de la tienda virtual de modelos tridimensionales se realizaron pruebas unitarias una vez finalizada la implementación de cada una de las funcionalidades desarrolladas. Para el desarrollo de las mismas se utilizó como *framework* de pruebas Pytest, el cual es un marco de prueba de Python relativamente maduro y completamente funcional. Proporciona documentación completa en línea, una gran cantidad de complementos de terceros y ayuda integrada, adecuada para muchos proyectos pequeños o grandes. Pytest es flexible y fácil de aprender. La salida estándar se puede capturar durante la depuración de impresión y la ejecución de la prueba, adecuada para pruebas unitarias simples a pruebas funcionales complejas (Programador clic, 2022).

En el siguiente código se refleja los resultados arrojados en el archivo de prueba `unitary_test.py` haciendo uso del marco de prueba Pytest:

Código fuente: Pruebas unitarias.

```
import pytest

from py._path.svnwc import cache
from djangoProject1.models import Categoria, Producto
from user.models import User
from django.contrib.auth import *

@pytest.mark.django_db
def test_autenticar():
    new = User.objects.create_user(username='testuser', password='testuser1*+')
    aux = authenticate(username=new.username, password=new.password)
    user = User.objects.get(id=1)
    assert user.is_active

@pytest.mark.django_db
def test_registrar_usuario():
    user = User.objects.create_user(username='registrar',
        password='regist1*+', is_superuser=True,
        first_name='Registrar', last_name='Usuario')
    assert user.username == 'registrar'
    assert user.first_name == 'Registrar'
    assert user.last_name == 'Usuario'
    assert user.is_superuser

@pytest.mark.django_db
def test_listar_categorias():
    Categoria.objects.create(nombre='New1')
    Categoria.objects.create(nombre='New2')
    resultados = Categoria.objects.all()
    cat1 = Categoria.objects.get(slug='new1')
    cat2 = Categoria.objects.get(slug='new2')
    assert cat1 in resultados
    assert cat2 in resultados

@pytest.mark.django_db
def test_add_categoria():
    categ = Categoria.objects.create(nombre='Categoria1')
    assert categ.nombre == 'Categoria1'

@pytest.mark.django_db
def test_modificar_categoria():
    cat = Categoria.objects.create(nombre='Mapa')
    Categoria.objects.filter(slug=cat.slug).update(nombre='Mares')
    cat.refresh_from_db()
    assert cat.nombre == 'Mares'

@pytest.mark.django_db
def test_eliminar_categoria():
    cat = Categoria.objects.create(nombre='New3')
    catAux = Categoria.objects.get(slug=cat.slug)
    catAux.delete()
    resultados = Categoria.objects.all()
    assert catAux.id not in resultados

@pytest.mark.django_db
def test_actualizar_perfil():
    perfil = User.objects.create(username='actualizar', first_name='Actualizar', last_name='Perfil')
    User.objects.filter(username=perfil.username).update(email='actualizar@uci.cu')
```

```

User.objects.filter(username=perfil.username).update(is_superuser=True)
perfil.refresh_from_db()
assert perfil.email== 'actualizar@uci.cu'
assert perfil.is_superuser

@pytest.mark.django_db
def test_filtrar_categoria():
    catAux = Categoria.objects.create(nombre='Cat')
    user = User.objects.create_user(username='testuser1', password='testuser1')
    prod = Producto.objects.create(nombre='Testprod', usuario=user,
        descripcion='Producto de prueba',
        precio=25.50, categoria=catAux,
        archivo='archivo/2022/06/24/A.jpg',
        image='imagenes/2022/10/26/png-transparent-tree-3d-modeling.png')
    productos = Producto.objects.filter(categoria=catAux)
    prod = Producto.objects.get(slug='testprod')
    assert prod in productos

@pytest.mark.django_db
def test_listar_productos():
    catAux = Categoria.objects.create(nombre='Cat')
    user = User.objects.create_user(username='testuser1', password='testuser1')
    Producto.objects.create(nombre='Testprod1', usuario=user, descripcion='Producto de prueba1',
        precio=25.50, categoria=catAux, archivo="", image="")
    Producto.objects.create(nombre='Testprod2', usuario=user, descripcion='Producto de prueba2',
        precio=25.50, categoria=catAux, archivo="", image="")
    Producto.objects.create(nombre='Testprod3', usuario=user, descripcion='Producto de prueba3',
        precio=25.50, categoria=catAux, archivo="", image="")

    productos = Producto.objects.all()
    prod = Producto.objects.get(slug='testprod3')
    assert prod in productos

@pytest.mark.django_db
def test_listar_productos_cliente():
    catAux = Categoria.objects.create(nombre='Cat')
    user1 = User.objects.create_user(username='testuser1', password='testuser1')
    user2 = User.objects.create_user(username='testuser2', password='testuser2')
    Producto.objects.create(nombre='Testprod1', usuario=user1, descripcion='Producto de prueba1',
        precio=25.50, categoria=catAux, archivo="", image="")
    Producto.objects.create(nombre='Testprod2', usuario=user2, descripcion='Producto de prueba2',
        precio=25.50, categoria=catAux, archivo="", image="")
    Producto.objects.create(nombre='Testprod3', usuario=user2, descripcion='Producto de prueba3',
        precio=25.50, categoria=catAux, archivo="", image="")
    productos = Producto.objects.filter(usuario_id=2)
    prod = Producto.objects.get(slug='testprod1')
    assert prod not in productos

@pytest.mark.django_db
def test_add_producto():
    catAux = Categoria.objects.create(nombre='Cat')
    user1 = User.objects.create_user(username='testuser1', password='testuser1')
    prod = Producto.objects.create(nombre='Testprod3', usuario=user1, descripcion='Producto de prueba3',
        precio=25.50, categoria=catAux, archivo="", image="")
    assert prod.slug == 'testprod3'

@pytest.mark.django_db
def test_modificar_producto():
    catAux = Categoria.objects.create(nombre='Cat')
    user1 = User.objects.create_user(username='testuser1', password='testuser1')

```

```

prod = Producto.objects.create(nombre='Testprod3', usuario=user1, descripcion='Producto de prueba3',
precio=25.50,categoria=catAux, archivo="", image=")
Producto.objects.filter(slug=prod.slug).update(precio=200.0)
prod.refresh_from_db()
assert prod.precio == 200.0

@pytest.mark.django_db
def test_eliminar_producto():
    prodElim = Producto.objects.create(nombre='Testprod3', usuario_id=2, descripcion='Producto de prueba3',
precio=25.50,categoria_id=1, archivo="", image=")
    lista = Producto.objects.all()
    prodElim.delete()
    assert prodElim.id not in lista

@pytest.mark.django_db
def test_buscar_producto():
    catAux = Categoria.objects.create(nombre='Cat')
    user1 = User.objects.create_user(username='testuser1', password='testuser1')
    Producto.objects.create(nombre='Testprod1', usuario=user1, descripcion='Producto de prueba3',
precio=25.50,categoria=catAux, archivo="", image=")
    Producto.objects.create(nombre='Testprod2', usuario=user1, descripcion='Producto de prueba3',
precio=55.00,categoria=catAux, archivo="", image=")
    Producto.objects.create(nombre='Testprod3', usuario=user1, descripcion='Producto de prueba3',
precio=125.50,categoria=catAux, archivo="", image=")
    prod = Producto.objects.get(slug='testprod2')
    lista = Producto.objects.all()
    assert prod in lista

```

En la siguiente figura se muestran los resultados arrojados a través del IDE PyCharm donde fueron ejecutadas las pruebas, evidenciándose la aceptación de las mismas:

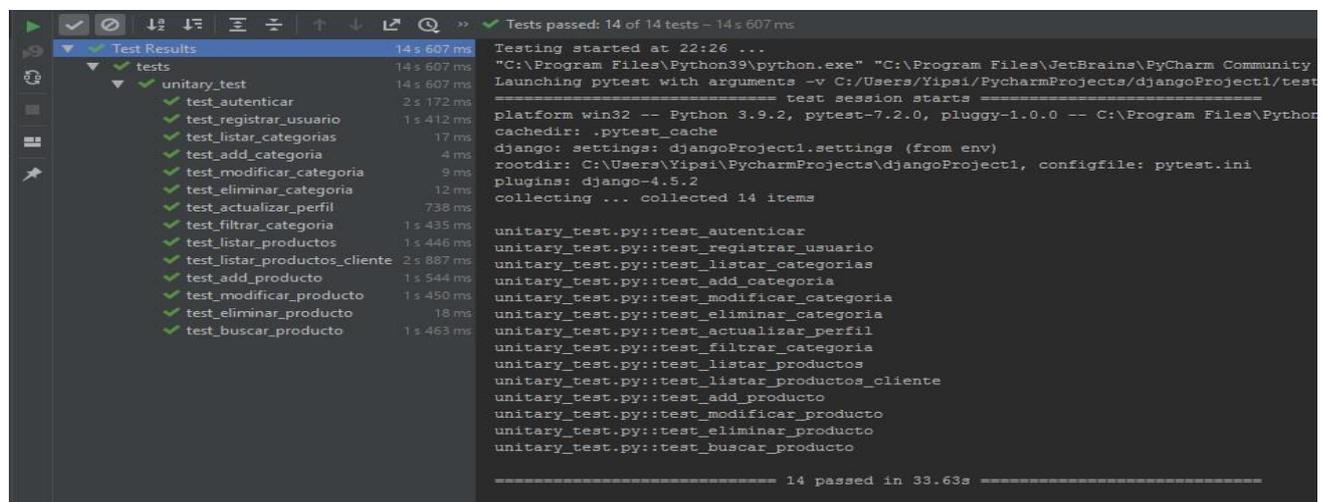


Figura 12. Resultados de las pruebas unitarias.

Se realizaron un total de 14 pruebas unitarias, de ellas 2 resultaron no satisfactorias (al eliminar una categoría, se eliminaban los productos asociados a esta y al crear un producto, aceptaba valores negativos en el precio de este), lo cual representa el 14% del total de prueba unitarias realizadas. Mientras las 12 pruebas restantes resultaron satisfactorias para un 86% del total, ver Figura 13. Los errores detectados por las pruebas no satisfactorias fueron mitigados después de 1 iteración de prueba, ver Figura 13.

Resultados

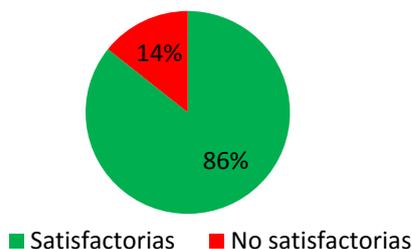


Figura 13. Resultados de las pruebas unitarias

3.3.2 Pruebas de aceptación

Las pruebas de aceptación son pruebas formales, ejecutadas para verificar si un sistema satisface sus requerimientos de negocio. Estas pruebas requieren que el *software* se encuentre en funcionamiento, y se centran en replicar el comportamiento de los usuarios, a fin de rechazar cambios si no se cumplen los objetivos. Estos objetivos pueden ir más allá de obtener una respuesta específica (PYM, 2022).

Las pruebas de aceptación:

- Son usualmente un conjunto de pruebas manuales que se realizan luego de que una fase de desarrollo ha finalizado (de modo que se pueda volver rápidamente e iterar si algo no está correcto).
- Verifican que las características del *software* estén alineadas con todas las especificaciones iniciales y criterios de aceptación.

Para que este tipo de pruebas se lleve a cabo correctamente resulta importante que los responsables del proyecto definan los criterios de aceptación justo antes de empezar a trabajar en el mismo. Así mismo, cualquier requerimiento adicional que surja durante el proceso deberá verse reflejado en tales criterios de aceptación (PYM, 2022).

3.3.2.1 Diseño de casos de prueba de aceptación

Las pruebas de aceptación fueron realizadas al culminar cada una de las 4 iteraciones de implementación. En cada una el equipo se reunió con el cliente para detectar las no conformidades. Se realizaron casos de prueba de aceptación para cada HU, las cuales se dividieron en las cuatro iteraciones de entrega del proyecto:

Iteración 1

A continuación, se muestran las correspondientes a las HU 1 y 2, el resto se encuentra en el anexo A.4:

Tabla 34. Prueba de aceptación # 1

Caso de prueba de aceptación

Código: HU1_P1	Historia de usuario: 1
Nombre: Autenticar usuario	
Descripción: Autenticar usuario en la tienda.	
Condiciones de ejecución: El cliente debe poseer un usuario en la tienda.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Usuario" y escoger la opción "Acceder" e introducir el usuario y la contraseña y presionar el botón "Iniciar sesión".	
Resultados esperados: El cliente se autentica en la tienda.	

Tabla 35. Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: HU2_P1	Historia de usuario: 2
Nombre: Registrar usuario	
Descripción: Se registra un usuario en la tienda.	
Condiciones de ejecución: No aplica	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Usuario" y escoger la opción "Registrarse", introducir los datos y presionar el botón "Registrar".	
Resultados esperados: El cliente adquiere un usuario en la tienda.	

Iteración 2

A continuación, se muestran las correspondientes a las HU 9 y 10, el resto se encuentra en el anexo A.4:

Tabla 36. Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: HU9_P1	Historia de usuario: 9
Nombre: Listar productos	
Descripción: Mostrar el listado de productos existentes en la tienda.	
Condiciones de ejecución: No aplica.	
Pasos de ejecución: Acceder a la página principal.	
Resultados esperados: Muestra el listado de productos existentes en la tienda.	

Tabla 37. Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: HU10_P1	Historia de usuario: 10
Nombre: Listar productos de cliente	
Descripción: Mostrar el listado de productos del usuario logueado.	
Condiciones de ejecución: El usuario debe estar logueado.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Usuario" y escoger la opción "Mis productos".	

Resultados esperados: Muestra el listado de productos del usuario logueado.

Iteración 3

A continuación, se muestran las correspondientes a las HU 17 y 18, el resto se encuentra en el anexo A.4:

Tabla 38. Prueba de aceptación # 18

Caso de prueba de aceptación	
Código: HU17_P1	Historia de usuario: 17
Nombre: Adicionar producto a la cesta	
Descripción: Adicionar el producto seleccionado a la cesta.	
Condiciones de ejecución: El usuario debe estar logueado.	
Pasos de ejecución: Hacer clic en el botón "Adicionar a la cesta" que se encuentra en la parte inferior producto.	
Resultados esperados: Adiciona el producto a la cesta de productos del usuario.	

Tabla 39. Prueba de aceptación # 19

Caso de prueba de aceptación	
Código: HU18_P1	Historia de usuario: 18
Nombre: Eliminar productos de la cesta	
Descripción: Eliminar un producto de la cesta.	
Condiciones de ejecución: El usuario debe estar logueado. Tener productos en la cesta.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Carrito", una vez dentro, hacer clic en el ícono "Eliminar" del producto que desea eliminar de la cesta.	
Resultados esperados: Elimina el producto de la cesta.	

Tabla 40. Prueba de aceptación # 20

Caso de prueba de aceptación	
Código: HU18_P2	Historia de usuario: 18
Nombre: Limpiar cesta.	
Descripción: Limpiar la cesta de productos.	
Condiciones de ejecución: El usuario debe estar logueado. Tener productos en la cesta.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Carrito", una vez dentro, hacer clic en el botón "Limpiar cesta".	
Resultados esperados: Elimina todos los productos de la cesta.	

Iteración 4

A continuación, se muestran las correspondientes a las HU 20 y 21, el resto se encuentra en el anexo A.4:

Tabla 41. Prueba de aceptación # 22

Caso de prueba de aceptación	
Código: HU20_P1	Historia de usuario: 20
Nombre: Realizar pago.	
Descripción: Efectuar el pago de los productos.	
Condiciones de ejecución: El usuario debe estar logueado. Tener productos en la cesta.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Carrito", una vez dentro, hacer clic en el botón "Realizar pago".	
Resultados esperados: Facturar los productos de la cesta.	

Tabla 42. Prueba de aceptación # 23

Caso de prueba de aceptación	
Código: HU21_P1	Historia de usuario: 21
Nombre: Listar productos comprados.	
Descripción: Lista los productos comprados por el cliente.	
Condiciones de ejecución: El usuario debe estar logueado. Tener productos comprados.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Usuario" y escoger la opción "Productos comprados".	
Resultados esperados: Listado de los productos comprados por el cliente.	

Se realizaron un total de 24 pruebas de aceptación, de ellas todas resultaron satisfactorias, lo cual representa el 100% de satisfacción, ver Figura 14.

Resultados



Figura 14. Resultados de las pruebas de aceptación.

3.3.3 Pruebas de carga y estrés.

Las pruebas de cargas tienen como objetivo saber cuánta carga puede soportar un sistema sin empeorar su rendimiento. Estas pruebas sirven para conocer la capacidad máxima de un sistema bajo una carga determinada y poder dar con las causas que puedan condicionar su rendimiento (Sanchez Avalos, 2019).

Las pruebas de estrés son realizadas sobrecargando un sistema más allá de sus especificaciones, para verificar cómo y cuándo fallará. Dentro de informática podemos colocar una gran carga en la

base de datos, entradas (peticiones) continuas al sistema o almacenar información más allá de la capacidad de memoria del sistema (Sanchez Avalos, 2019).

Se utilizó la herramienta *Apache JMeter* para la realización de estas pruebas, a continuación, se describen las variables que miden el resultado de las mismas, realizadas al módulo:

Muestra: Cantidad de peticiones realizadas para cada URL.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mín: Tiempo mínimo que demora un hilo en acceder a una página.

Máx: Tiempo máximo que demora un hilo en acceder a una página.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

Kb/s: Velocidad de carga de las páginas.

El ambiente de prueba estuvo conformado por:

Hardware de prueba:

- Sistema Operativo: Windows v.10
- Microprocesador: AMD A4-7210 APU 1,8 GHz
- Memoria RAM: 4.00 GB
- Disco duro: 1024 GB

Como se muestra en la siguiente tabla, se simularon las peticiones realizadas a la tienda por un total de 50, 75 y 100 usuarios simultáneamente en cada caso, los cuales realizan hasta 4 peticiones por segundo. Se obtuvieron los siguientes resultados:

Tabla 43. Resultados de la prueba carga y estrés

Usuarios	Muestras	Media	Min	Max	%Error	Rend	Kb/s
50	200	1226	82	4232	0.00%	29.5	205.4
75	300	1795	71	6934	2.67%	28.9	190.62
100	400	2187	241	7680	8.00%	32.8	191.65

Las pruebas que fueron realizadas muestran que la tienda es capaz de responder a las peticiones, evidenciando así que esta puede procesar la carga esperada. Para esto se realizaron un total de 200 peticiones de 50 usuarios conectados simultáneamente en un tiempo promedio de 1226 milisegundos (1.2 segundos aproximadamente) con 0 % de error.

También se realizaron 300 peticiones iniciadas por 75 usuarios y en este caso el módulo respondió en 1795 milisegundos (1.8 segundos aproximadamente) como tiempo promedio. Aunque el módulo

no fue capaz de responder correctamente el 2.67% de las peticiones realizadas, se demuestra que el módulo puede procesar la carga esperada.

Con el objetivo de analizar el comportamiento de la tienda en condiciones extremas, se realizó una prueba de estrés para un conjunto de 100 usuarios conectados simultáneamente. En este caso, el módulo responde a las 400 peticiones en un tiempo promedio de 2187 milisegundos (2.1 segundos aproximadamente), pero con un porcentaje de error de 8.00. Este resultado está estrechamente relacionado al entorno donde se realizó la prueba, el cual no es un servidor dedicado sino un cliente habilitado.

Conclusiones parciales

La implementación de la propuesta de solución permitió obtener una aplicación funcional capaz de contribuir con la comercialización de los modelos tridimensionales creados por los desarrolladores de Vertex. Mediante la ejecución de las pruebas de software, se constató la calidad en la implementación realizada.

Conclusiones generales

Considerando los resultados descritos en este informe, la necesidad y el objetivo planteado por la investigación, se arriban a las siguientes conclusiones:

- El planteamiento de los conceptos relacionados con la investigación, permitió definir las características y componentes que debía poseer la solución y se pudo verificar a través de la revisión de la literatura.
- La caracterización del proceso de comercialización de modelos tridimensionales, permitió la identificación de las funcionalidades necesarias para la solución de acuerdo a las necesidades requeridas por el cliente.
- La metodología, tecnologías y herramientas seleccionadas facilitaron el desarrollo de la tienda virtual, asegurando la sostenibilidad de la solución.
- La ejecución de pruebas de *software* y corrección de las no conformidades encontradas durante la etapa de desarrollo, demostró la validez de la solución propuesta y el cumplimiento de los requisitos definidos por el cliente.

Recomendaciones

Para dar continuidad a la presente investigación se recomienda:

- Una vez finalizado el contrato para el uso de una pasarela de pago, sustituir el método de pago mediante la API creada por el correcto.

Bibliografía Consultada:

Django Rest framework. (2022). *Django Rest framework*. Obtenido de <https://www.django-rest-framework.org/>

Alicia M. (2021 de febrero de 2021). *Free3D, la plataforma de archivos 3D para todos los usuarios*. Obtenido de <https://www.3dnatives.com/es/free3d-plataforma-archivos-3d-080220212/#!>

Arsys. (13 de junio de 2018). *¿Qué es PostgreSQL y por que llevarlo a Cloud?* Obtenido de <https://www.arsys.es/blog/postgresql-servidores>

Asesores, E. (2022). *Creación y Desarrollo de Tiendas Virtuales*. Obtenido de <https://www.elg-asesores.com/index.php/servicios/web-marketing/creacion-y-desarrollo-de-tiendas-virtuales>

Betania V. (16 de Septiembre de 2022). *Pasarela de pago: Qué es, cómo funciona, las mejores pasarelas comparadas*. Obtenido de <https://www.hostinger.es/tutoriales/pasarela-de-pago>

Billy Reynoso , C. (2022). *INTRODUCCIÓN A LA ARQUITECTURA DE SOFTWARE*. Obtenido de https://www.ecotec.edu.ec/material/material_2019F1_COM335_02_128427.pdf

Bustamante , D., & Rodríguez , J. (marzo de 2018). *Metodología actual. Metodología XP*. Obtenido de <https://luismejias21.files.wordpress.com/2018/03/metodologia-xp.pdf>

Caicedo, P. (4 de febrero de 2021). *¡Hola Mundo! en Django*. Obtenido de <https://pedrocaicedo.medium.com/hola-mundo-en-django-2363ffb839d1>

Consultec, S. (2021). *Crear un negocio en Internet ¿Por dónde empiezo?* Obtenido de <https://www.vitoria-gasteiz.org/docs/wb021/contenidosEstaticos/adjuntos/es/98/76/39876.pdf>

Cortés Sánchez, P. (18 de enero de 2022). *Qué es Amazon Historia*. Obtenido de <https://paginasweb.tech/amazon/que-es-amazon-historia/>

Cuba a Pulso. (18 de noviembre de 2021). *Cosmox: Plataforma de videojuegos cubana con servicios interesantes*. Obtenido de <https://cubapulso.com/cosmox-plataforma-de-videojuegos-cubana-con-servicios-interesantes/>

Deloitte. (2022). *¿Cuál es la metodología más adecuada para tu proyecto?* Obtenido de <https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html>

Demera, R. (2 de Febrero de 2021). *Metodología ¿Tradicional o Ágil?* Obtenido de https://tech.tribalyte.eu/blog-metodologias-tradicional-vs-agil#Metodologias_agiles

Diego Polo, J. (18 de enero de 2017). *Cults3D, para encontrar modelos 3D que podremos imprimir*. Obtenido de <https://www.whatsnew.com/2017/01/18/cults3d-para-encontrar-modelos-3d-que-podremos-imprimir/>

DigitalOcean. (22 de enero de 2020). *Cómo instalar y configurar pgAdmin 4 en el modo de servidor*. Obtenido de <https://www.digitalocean.com/community/tutorials/how-to-install-configure-pgadmin4-server-mode-es>

Ecommerce-Platforms. (2022). *¿Qué es el comercio electrónico? Definición de comercio electrónico para 2022*. Obtenido de <https://ecommerce-platforms.com/es/glossary/ecommerce>

Etecé, E. (2022). *TICs*. Obtenido de <https://concepto.de/quienes-somos/>

Euroforum. (1 de febrero de 2022). *Metodología Scrum: definición, herramientas y ejemplos de proyectos*. Obtenido de <https://www.euroforum.es/blog/metodologia-scrum-definicion-herramientas-y-ejemplos-de-proyectos/>

García Nieto, M., Martínez Trinidad, T., de los Santos Posadas, H., & Jiménez Casas, M. (2017). *Aplicación del proceso unificado en el desarrollo de un software que estima el inventario y el crecimiento-rendimiento maderable en plantaciones de eucalipto*. Obtenido de https://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-04712017000100163

Hardy Demestre, D., & Hernández González, Y. (20 de marzo de 2019). *Design of a Human Resources Information Management System*. Obtenido de <http://publicaciones.uci.cu>

Headways, m. (2016). *Tienda virtual*. Obtenido de <https://headways.com.mx/glosario-mercadotecnia/palabra/tienda-virtual/>

Higuerey, E. (1 de Junio de 2019). *Comercio electrónico: conoce todo sobre este modelo de negocios y cuáles son sus ventajas*. Obtenido de <https://rockcontent.com/es/blog/comercio-electronico/>

Izquierdo Pardo, J. M., Pardo Gómez, M. E., & Izquierdo Lao, J. M. (14 de septiembre de 2020). *Modelos digitales 3D en el proceso de enseñanza-aprendizaje de las ciencias médicas*. Obtenido de <http://www.medisan.sld.cu/index.php/san/article/view/3134/2102>

Jacobson, I., Booch, G., & Rumbaugh J. . (2000). *El proceso unificado de desarrollo de software (No. 004.41)*. Pearson Educación. . New York, USA.

LanceTalent. (18 de Marzo de 2015). *Diseña tu tienda online*. Obtenido de <https://www.lancetalent.com/blog/como-hacer-el-diseno-de-tu-tienda-online-10-ejemplos-para-inspirarte/>

Library. (2022). *Herramienta CASE Visual Paradigm*. Obtenido de <https://1library.co/article/herramienta-case-visual-paradigm-marco-te%C3%B3rico.qvllerry>

Lopes Martínez, I., Marrero Durán, S. P., Feria Martínez, M. A., Grass Santos, A., Espina Martín, Y., & Lugo Almaguer, A. (2021). Impacto de la Covid-19 en las Cadenas de Suministro Globales: Caso Comercio Electrónico. *Revista Cubana de Administración Pública y Empresarial*. Obtenido de <https://apye.esceg.cu/index.php/apye/article/view/153/109>

Maida, E. G., & Pacienza, J. (2017). *Metodologías de desarrollo de software*. Obtenido de <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Manotoa, A. (2021). *Tutorial de PyCharm: características, instalación y usos*. Obtenido de <https://platzi.com/blog/pycharm/>

MDN Web Docs. (2022). *Introducción a Django*. Obtenido de <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>

MDN Web Docs, H. (2022). *Conceptos básicos de HTML*. Obtenido de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics

Meléndez Valladarez, S. M., Gaitan, M. E., & Pérez Reyes, N. N. (2017). *Metodología Ágil de Desarrollo de Software Programación*.

OpenWebinars. (26 de junio de 2019). *Qué es CSS y para qué sirve*. Obtenido de <https://openwebinars.net/blog/que-es-css/>

Pérez Porto, J., & Gardey, A. (2022). *Definición de videojuego - Qué es, Significado y Concepto*. Obtenido de <https://definicion.de/videojuego/>

Pressman, & Maxim. (2015). *Software Engineering: A Practitioner's Approach. Eighth Edition. 8va ed. CM SIGSOFT Software Engineering Notes. McGraw-Hill Education*.

Programador clic. (2022). *Prueba el código de Python con Pytest*. Obtenido de <https://programmerclick.com/article/3366908761/>

PYM. (2022). *Los diferentes tipos de testing en el desarrollo de software*. Obtenido de <https://programacionymas.com/blog/tipos-de-testing-en-desarrollo-de-software>

Sanchez Avalos, S. (26 de Septiembre de 2019). *Pruebas de carga vs Pruebas de estrés*. Obtenido de <https://sospnt.com/blog/103-load-testing-vs-stress-testing>

Santander. (9 de abril de 2021). *Python: qué es y por qué deberías aprender a utilizarlo*. Obtenido de <https://www.becas-santander.com/es/blog/python-que-es.html>

Siebert Domareski, H. (28 de Julio de 2020). *Design Patterns — Introduction*. Obtenido de <https://henriquesd.medium.com/design-patterns-introduction-220f811db857>

Sommerville, I. . (2011). *Ingeniería de Software. 9na ed.* México: Addison Wesley. Pearson Education, Inc. ISBN 978-607-32-0603-7. 2011. 792 páginas.

Turrado, J. (10 de marzo de 2020). *Qué son las pruebas de software*. Obtenido de <https://www.campusmv.es/recursos/post/que-son-las-pruebas-de-software.aspx>

Unity, D. (2016). *Manual de Unity*. Obtenido de <https://docs.unity3d.com/es/530/Manual/AssetStore.html>

Uniwebsidad. (2022). *El patrón de diseño MTV*. Obtenido de <https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv>

Vanegas Prieto, M. F. (10 de noviembre de 2020). *Modelos 3D* . Obtenido de <https://niixer.com/index.php/2020/11/10/modelos-3d/>

Anexos

A.1. Historias de usuario

Tabla A.1.1. Historia de Usuario "Autenticar usuario".

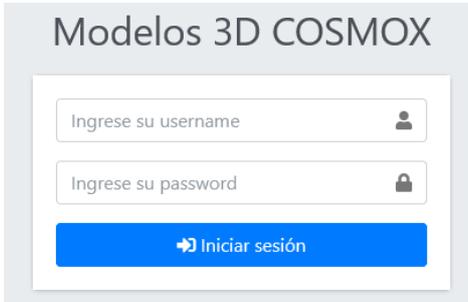
Historia de usuario	
Número: 1	Nombre: Autenticar usuario
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1.0	Iteración asignada: 1
Programador responsable: Yipsi Lourdes Sánchez Mesa	
<p>Descripción: Los usuarios se pueden autenticar en el sitio. Existen dos roles, uno administrador y otro cliente. Para autenticarse deben llenar los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre de usuario: (Obligatorio. Campo de texto. Permite los caracteres: (a-z, A-Z, 0-9) y los restantes caracteres.) • Contraseña: (Obligatorio. Campo de texto. Permite los caracteres: (a-z, A-Z, 0-9) y los restantes caracteres.) 	
<p>Observaciones:</p> <ol style="list-style-type: none"> 1. Si el usuario introduce la información de forma correcta, automáticamente entra a la tienda. 2. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error. 3. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse. 	
<p>Interfaz:</p> 	

Tabla A.1.2. Historia de Usuario "Listar categorías".

Historia de usuario	
Número: 3	Nombre: Listar categorías
Usuario: Sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.0	Iteración asignada: 1
Programador responsable: Yipsi Lourdes Sánchez Mesa	
<p>Descripción: Cuando un usuario administrador se autentica en la tienda y accede en el panel Administrar a la opción Categorías el sistema muestra un listado con las categorías existentes.</p>	
Observaciones:	

1. En caso de no existir ninguna categoría, el sistema muestra un mensaje “No existe información a mostrar”.

Interfaz:

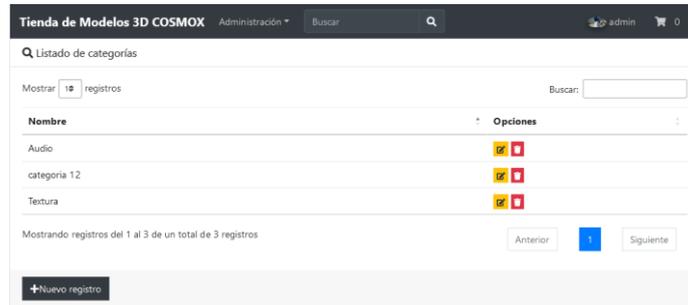


Tabla A.1.3. Historia de Usuario “Adicionar categoría”.

Historia de usuario	
Número: 4	Nombre: Adicionar categoría
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.0	Iteración asignada: 1
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Cuando un usuario administrador se autentica en la tienda y accede en el panel Administración a la opción Categorías, el sistema muestra un listado con las categorías existentes y brinda la posibilidad de adicionar una nueva. El sistema muestra la interfaz adicionar categoría, al seleccionar el botón Nuevo registro el sistema regresa al listado de categorías.	
Observaciones:	
1. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.	
2. Si el usuario introduce una categoría existente el sistema le muestra un mensaje “Categoría existente”.	
Interfaz:	

Tabla A.1.4. Historia de Usuario “Modificar categoría”.

Historia de usuario	
Número: 5	Nombre: Modificar categoría
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja

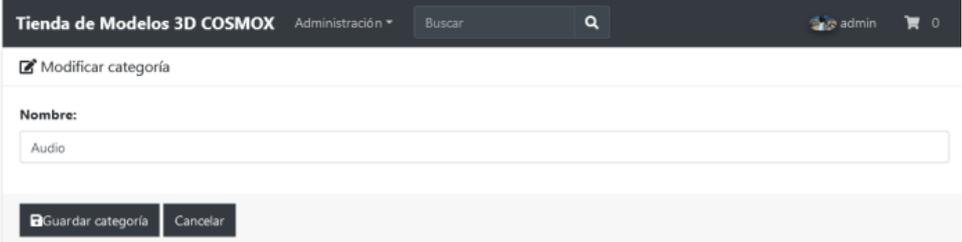
Puntos estimados: 1.0	Iteración asignada: 1
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Cuando un usuario administrador se autentica en la tienda y accede en el panel Administrar a la opción Categorías, el sistema muestra un listado con las categorías existentes y selecciona la opción modificar. El sistema muestra la interfaz modificar categoría, al seleccionar el botón aceptar el sistema regresa al listado de categorías.	
Observaciones: 1. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse. 2. Si el usuario introduce una categoría existente el sistema le muestra un mensaje “Categoría existente”.	
Interfaz: 	

Tabla A.1.5. Historia de Usuario “Eliminar categoría”.

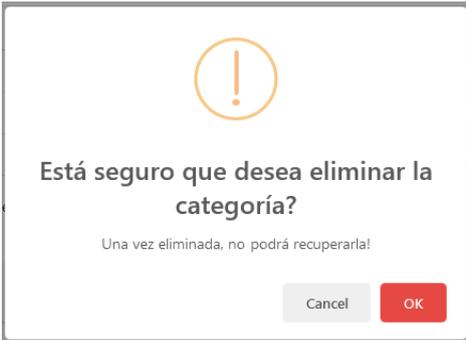
Historia de usuario	
Número: 6	Nombre: Eliminar categoría
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.0	Iteración asignada: 1
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Cuando un usuario administrador se autentica en la tienda y accede en el panel Administrar a la opción Categorías, el sistema muestra un listado con las categorías existentes y selecciona la opción eliminar.	
Observaciones: 1. El sistema muestra un mensaje “¿Está seguro que desea eliminar la categoría?”, al seleccionar el botón OK, se elimina la categoría seleccionada.	
Interfaz: 	

Tabla A.1.6. Historia de Usuario “Actualizar perfil”.

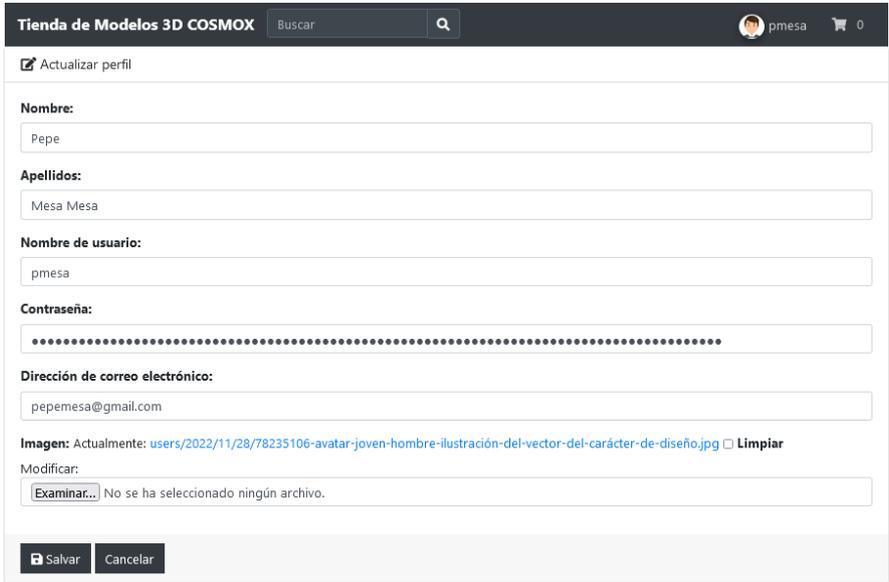
Historia de usuario	
Número: 7	Nombre: Actualizar perfil
Usuario: Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1.5	Iteración asignada: 1
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema permite actualizar los datos del perfil.	
Observaciones:	
<p>1. Si el usuario introduce la información de forma incorrecta, el sistema emite un mensaje notificando el error.</p> <p>2. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse.</p>	
Interfaz:	
	

Tabla A.1.7. Historia de Usuario “Filtra por categoría”.

Historia de usuario	
Número: 8	Nombre: Filtra por categoría
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.0	Iteración asignada: 1
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema debe permitir al usuario filtrar los productos por categoría.	
Observaciones:	
<p>1. Si el usuario escoge una categoría que no tiene productos asociados, el sistema muestra un mensaje notificando que “No se encontraron productos de esa categoría.”</p>	
Interfaz:	

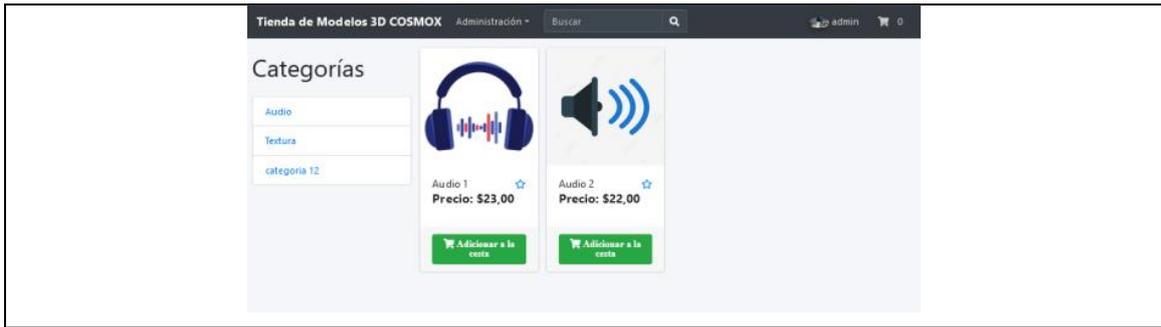


Tabla A.1.8. Historia de Usuario “Listar productos”.

Historia de usuario	
Número: 9	Nombre: Listar productos
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Muestra un listado con los productos disponibles en la tienda.	
Observaciones: No aplica	
Interfaz:	

Tabla A.1.9. Historia de Usuario “Listar productos de cliente”.

Historia de usuario	
Número: 10	Nombre: Listar productos de cliente
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Cuando un cliente se autentica en la tienda y accede a su perfil, el sistema muestra un listado con sus productos.	

Observaciones: No aplica

Interfaz:

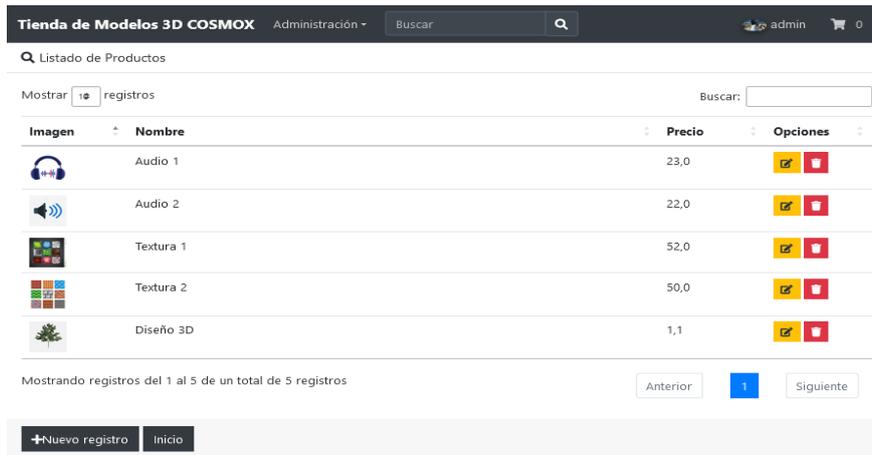


Tabla A.1.10. Historia de Usuario "Modificar producto".

Historia de usuario	
Número: 12	Nombre: Modificar producto
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.0	Iteración asignada: 2
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Cuando un cliente accede a la opción Mis productos, el sistema muestra un listado de sus productos y selecciona la opción modificar. El sistema muestra la interfaz modificar producto, al seleccionar el botón aceptar el sistema regresa al listado de productos.	
Observaciones: 1. Si el usuario introduce la información dejando campos obligatorios vacíos, el sistema emite un mensaje indicándole que los campos obligatorios deben llenarse. 2. Si el usuario introduce un producto existente el sistema le muestra un mensaje "Producto existente".	
Interfaz:	

Tabla A.1.11. Historia de Usuario “Eliminar producto”.

Historia de usuario	
Número: 13	Nombre: Eliminar producto
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.0	Iteración asignada: 2
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Cuando un cliente accede a la opción Mis productos, el sistema muestra un listado de sus productos y selecciona la opción eliminar.	
Observaciones: 1. El sistema muestra un mensaje “Está seguro que desea eliminar el producto?”, al seleccionar el botón OK, se elimina el producto seleccionado.	
Interfaz:	

Tabla A.1.12. Historia de Usuario “Buscar producto”.

Historia de usuario	
Número: 14	Nombre: Buscar producto

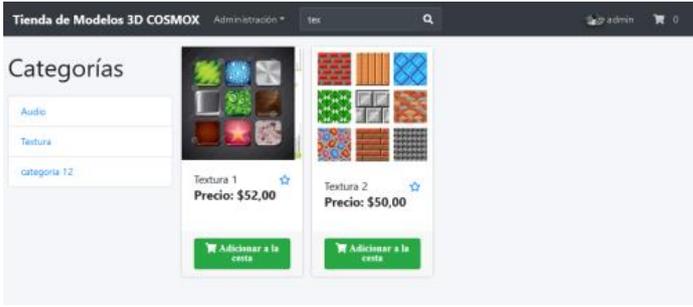
Usuario: Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1.0	Iteración asignada: 2
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema permite buscar productos por nombre.	
Observaciones: No aplica	
Interfaz:	
	

Tabla A.1.13. Historia de Usuario “Administrar favoritos”.

Historia de usuario	
Número: 15	Nombre: Administrar favoritos
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.5	Iteración asignada: 2
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema permite agregar y eliminar un producto al listado de favoritos.	
Observaciones: No aplica	
Interfaz: -	

Tabla A.1.14. Historia de Usuario “Listar productos favoritos”.

Historia de usuario	
Número: 16	Nombre: Listar productos favoritos
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1.0	Iteración asignada: 2
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: El sistema permite listar los productos favoritos de un cliente.	
Observaciones: No aplica	
Interfaz:	

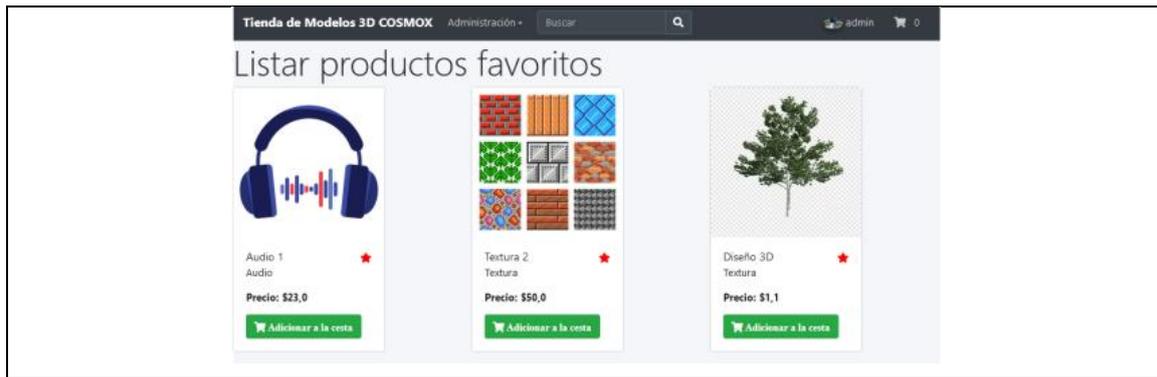


Tabla A.1.15. Historia de Usuario “Administrar cesta de productos”.

Historia de usuario	
Número: 18	Nombre: Administrar cesta de productos
Usuario: Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2.0	Iteración asignada: 3
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Una vez en la cesta de productos el cliente puede eliminar los productos adicionados, facturar la compra o seguir comprando.	
Observaciones: No aplica	
Interfaz:	

A.2. Descripción de las tablas

Tabla A.2.1. Descripción de la tabla producto_favoritos

Nombre: producto_favoritos		
Descripción: almacena todos los productos favoritos		
Atributo	Tipo	Descripción
id	long	Identificador.
producto_id	long	Llave foránea que hace referencia al identificador de la tabla producto.
user_id	long	Llave foránea que hace referencia al identificador de la tabla user.

Tabla A.2.2. Descripción de la tabla compra

Nombre: compra		
Descripción: almacena todas las compras		
Atributo	Tipo	Descripción
id	long	Identificador.
fecha	date	Fecha de efectuada la compra.

usuario_id	long	Llave foránea que hace referencia al identificador de la tabla user.
------------	------	--

Tabla A.2.3. Descripción de la tabla sub_compra

Nombre: sub_compra		
Descripción: almacena todas la subcompra		
Atributo	Tipo	Descripción
id	long	Identificador
producto_id	long	Llave foránea que hace referencia al identificador de la tabla producto.

Tabla A.2.4. Descripción de la tabla compra_productos

Nombre: compra_productos		
Descripción: almacena todas las compras de productos		
Atributo	Tipo	Descripción
id	long	Identificador
compra_id	long	Llave foránea que hace referencia al identificador de la tabla compra.
subcompra_id	long	Llave foránea que hace referencia al identificador de la tabla sub_compra.

Tabla A.2.5. Descripción de la tabla user_user

Nombre: user_user		
Descripción: almacena todos los usuarios		
Atributo	Tipo	Descripción
id	long	Identificador.
password	string	Contraseña del usuario.
last_login	date	Fecha de la última vez que se logueó.
is_superuser	boolean	Tiene todos los permisos sin asignarlos explícitamente.
username	string	Nombre de usuario.
first_name	string	Nombre.
last_name	string	Apellidos.
email	string	Dirección de correo electrónico.
is_staff	boolean	Si puede acceder al sitio de administración.
is_active	boolean	Si está activo el usuario.
date_joined	date	Fecha de creado el usuario.
image	string	Imagen del producto
terminos_condiciones	boolean	Si está de acuerdo con los términos y condiciones.

A.3. Tareas de ingeniería

Tabla A.3.1. Tarea de ingeniería # 3

Tarea	
Número de tarea: 3	Número de Historia de usuario: 3
Nombre de la tarea: Listar categorías	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 27/05/22	Fecha de fin: 02/06/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de ListView, que sea capaz de listar las categorías registradas en la tienda.	

Tabla A.3.2. Tarea de ingeniería # 4

Tarea	
Número de tarea: 4	Número de Historia de usuario: 4
Nombre de la tarea: Adicionar categoría	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 03/06/22	Fecha de fin: 09/06/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de CreateView, que sea capaz de adicionar una nueva categoría.	

Tabla A.3. 4. Tarea de ingeniería # 5

Tarea	
Número de tarea: 5	Número de Historia de usuario: 5
Nombre de la tarea: Modificar categoría	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 10/06/22	Fecha de fin: 16/06/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de UpdateView, que sea capaz de modificar una categoría.	

Tabla A.3.5. Tarea de ingeniería # 6

Tarea	
Número de tarea: 6	Número de Historia de usuario: 6
Nombre de la tarea: Eliminar categoría	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 17/06/22	Fecha de fin: 23/06/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de eliminar una categoría.	

Tabla A.3.6. Tarea de ingeniería # 7

Tarea

Número de tarea: 7	Número de Historia de usuario: 7
Nombre de la tarea: Actualizar perfil	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha de inicio: 24/06/22	Fecha de fin: 03/07/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de UpdateView, que sea capaz de modificar el perfil de un usuario.	

Tabla A.3.7. Tarea de ingeniería # 8

Tarea	
Número de tarea: 8	Número de Historia de usuario: 8
Nombre de la tarea: Filtra por categoría	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 04/07/22	Fecha de fin: 09/07/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de filtrar los productos por la categoría seleccionada.	

Tabla A.3.8. Tarea de ingeniería # 11

Tarea	
Número de tarea: 11	Número de Historia de usuario: 11
Nombre de la tarea: Adicionar producto	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha de inicio: 17/07/22	Fecha de fin: 27/07/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de CreateView, que sea capaz de adicionar un producto a la lista de productos del cliente autenticado en la tienda.	

Tabla A.3.9. Tarea de ingeniería # 12

Tarea	
Número de tarea: 12	Número de Historia de usuario: 12
Nombre de la tarea: Modificar producto	

Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 28/07/22	Fecha de fin: 03/08/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de UpdateView, que sea capaz de modificar un producto de la lista de productos del cliente autenticado en la tienda.	

Tabla A.3.10. Tarea de ingeniería # 13

Tarea	
Número de tarea: 13	Número de Historia de usuario: 13
Nombre de la tarea: Eliminar producto	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 04/08/22	Fecha de fin: 10/08/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de eliminar un producto de la lista de productos del cliente autenticado en la tienda.	

Tabla A.3.11. Tarea de ingeniería # 14

Tarea	
Número de tarea: 14	Número de Historia de usuario: 14
Nombre de la tarea: Buscar producto	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 11/08/22	Fecha de fin: 17/08/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de buscar un producto en la tienda.	

Tabla A.3.12. Tarea de ingeniería # 15

Tarea	
Número de tarea: 15	Número de Historia de usuario: 15
Nombre de la tarea: Agregar a favorito	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 18/08/22	Fecha de fin: 27/08/22

Programador responsable: Yipsi Lourdes Sánchez Mesa
Descripción: Crear una vista basada en función, que sea capaz de adicionar un producto a la lista de productos favoritos del cliente autenticado.

Tabla A.3.13. Tarea de ingeniería # 16

Tarea	
Número de tarea: 16	Número de Historia de usuario: 15
Nombre de la tarea: Eliminar favorito	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 28/08/22	Fecha de fin: 01/09/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de eliminar un producto de la lista de productos favoritos del cliente autenticado.	

Tabla A.3.14. Tarea de ingeniería # 17

Tarea	
Número de tarea: 17	Número de Historia de usuario: 16
Nombre de la tarea: Listar productos favoritos	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 28/08/22	Fecha de fin: 01/09/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista con la ayuda de ListView, que sea capaz de listar los productos favoritos del cliente autenticado, registrados en la tienda.	

Tabla A.3.15. Tarea de ingeniería # 20

Tarea	
Número de tarea: 20	Número de Historia de usuario: 18
Nombre de la tarea: Limpiar cesta	
Tipo de tarea: Desarrollo	Puntos estimados: 1.0
Fecha de inicio: 23/09/22	Fecha de fin: 29/09/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de eliminar todos los	

productos de la cesta del cliente autenticado.

Tabla A.3.16. Tarea de ingeniería # 21

Tarea	
Número de tarea: 21	Número de Historia de usuario: 19
Nombre de la tarea: Ver detalles de producto	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 30/09/22	Fecha de fin: 02/10/22
Programador responsable: Yipsi Lourdes Sánchez Mesa	
Descripción: Crear una vista basada en función, que sea capaz de mostrar los detalles de un producto seleccionado.	

A.4. Diseño de casos de prueba de aceptación

Tabla A.4.1. Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: HU3_P1	Historia de usuario: 3
Nombre: Listar categorías	
Descripción: Se listan todas las categorías de productos de la tienda.	
Condiciones de ejecución: El usuario debe ser administrador.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior de la página en la sesión de "Administración" y escoger la opción "Categorías".	
Resultados esperados: El sistema muestra un listado con todas las categorías de productos de la tienda.	

Tabla A.4.2. Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: HU4_P1	Historia de usuario: 4
Nombre: Adicionar categoría	
Descripción: Adiciona una nueva categoría.	
Condiciones de ejecución: El usuario debe ser administrador.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior de la página en la sesión de "Administración" y escoger la opción "Categorías", luego hacer clic en el botón "Nuevo registro", introducir los datos y presionar el botón "Guardar categoría".	
Resultados esperados: Se adiciona una nueva categoría.	

Tabla A.4.3. Prueba de aceptación # 5

Caso de prueba de aceptación

Código: HU5_P1	Historia de usuario: 5
Nombre: Modificar categoría	
Descripción: Modifica una categoría.	
Condiciones de ejecución: El usuario debe ser administrador.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior de la página en la sesión de “Administración” y escoger la opción “Categorías”, luego hacer clic en la opción “Modificar”, introducir los datos y presionar el botón “Guardar categoría”.	
Resultados esperados: Se modifica la categoría seleccionada.	

Tabla A.4.4. Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: HU6_P1	Historia de usuario: 6
Nombre: Eliminar categoría	
Descripción: Eliminar una categoría.	
Condiciones de ejecución: El usuario debe ser administrador.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior de la página en la sesión de “Administración” y escoger la opción “Categorías”, luego hacer clic en la opción “Eliminar” y en el panel que se muestra, presionar el botón “Ok”.	
Resultados esperados: Se elimina la categoría seleccionada.	

Tabla A.4.5. Prueba de aceptación # 7

Caso de prueba de aceptación	
Código: HU7_P1	Historia de usuario: 7
Nombre: Actualizar perfil	
Descripción: Actualiza los datos del usuario autenticado.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono “Usuario” y escoger la opción “Actualizar perfil”, modificar o adicionar los datos y presionar el botón “Salvar”.	
Resultados esperados: Se actualizan los datos del usuario.	

Tabla A.4.6. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU8_P1	Historia de usuario: 8
Nombre: Filtra por categoría	
Descripción: Actualiza el listado de productos según la categoría seleccionada.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: En la parte izquierda de la página, el cliente debe seleccionar la categoría por la cual desea filtrar los productos.	
Resultados esperados: Se muestran solo los productos de la categoría seleccionada.	

Tabla A.4.7. Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: HU11_P1	Historia de usuario: 11
Nombre: Adicionar producto	
Descripción: Adicionar producto al listado de productos del usuario logueado.	
Condiciones de ejecución: El usuario debe estar logueado.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Usuario" y escoger la opción "Mis productos", hacer clic en el botón "Nuevo registro", introducir los datos y hacer clic en el botón "Guardar producto".	
Resultados esperados: Muestra el listado de productos del usuario logueado.	

Tabla A.4.8. Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: HU12_P1	Historia de usuario: 12
Nombre: Modificar producto	
Descripción: Modificar producto del listado de productos del usuario logueado.	
Condiciones de ejecución: El usuario debe estar logueado.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Usuario" y escoger la opción "Mis productos", luego escoger la opción "Modificar", introducir o modificar los datos y hacer clic en el botón "Guardar producto".	
Resultados esperados: Se modifica el producto seleccionado.	

Tabla A.4.9. Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: HU13_P1	Historia de usuario: 13
Nombre: Eliminar producto	
Descripción: Eliminar producto del listado de productos del usuario logueado.	
Condiciones de ejecución: El usuario debe estar logueado.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Usuario" y escoger la opción "Mis productos", luego hacer clic en la opción "Eliminar" y en el panel que se muestra, presionar el botón "Ok".	
Resultados esperados: Se elimina el producto seleccionado.	

Tabla A.4.10. Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: HU14_P1	Historia de usuario: 14
Nombre: Buscar producto	
Descripción: Buscar producto del listado de productos de la tienda.	
Condiciones de ejecución: No aplica.	
Pasos de ejecución: En la parte superior de la página se encuentra el "Buscar producto", el	

usuario debe introducir el nombre del producto que desea encontrar y presionar el botón “Buscar”.
Resultados esperados: Se muestra un listado con los productos que cumplan con el criterio de búsqueda.

Tabla A.4.11. Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: HU15_P1	Historia de usuario: 15
Nombre: Agregar a favorito	
Descripción: Agregar el producto al listado de favoritos del usuario.	
Condiciones de ejecución: El usuario debe estar logueado.	
Pasos de ejecución: Seleccionar el marcador que se encuentra en la parte superior derecha del producto.	
Resultados esperados: Adiciona el producto al listado de favoritos del usuario.	

Tabla A.4.12. Prueba de aceptación # 16

Caso de prueba de aceptación	
Código: HU15_P2	Historia de usuario: 15
Nombre: Eliminar favorito	
Descripción: Eliminar el producto del listado de favoritos del usuario.	
Condiciones de ejecución: El usuario debe estar logueado.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono “Usuario” y escoger la opción “Favoritos”, en el listado de productos favoritos, hacer clic en el marcador del producto que desea eliminar.	
Resultados esperados: Elimina el producto del listado de favoritos del usuario.	

Tabla A.4.13. Prueba de aceptación # 17

Caso de prueba de aceptación	
Código: HU16_P1	Historia de usuario: 16
Nombre: Listar productos favoritos.	
Descripción: Lista los productos favoritos del cliente.	
Condiciones de ejecución: El usuario debe estar logueado. Tener productos favoritos.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono “Usuario” y escoger la opción “favoritos”.	
Resultados esperados: Listado de los productos favoritos del cliente.	

Tabla A.4.14. Prueba de aceptación # 21

Caso de prueba de aceptación	
Código: HU19_P1	Historia de usuario: 19
Nombre: Ver detalles de producto.	

Descripción: Ver detalles de un producto seleccionado.
Condiciones de ejecución: No aplica
Pasos de ejecución: El cliente debe hacer clic encima de la imagen de un producto.
Resultados esperados: Ver detalles de un producto.

Tabla A.4.15. Prueba de aceptación # 24

Caso de prueba de aceptación	
Código: HU22_P1	Historia de usuario: 22
Nombre: Descargar archivo.	
Descripción: Descargar el archivo asociado a un producto comprado.	
Condiciones de ejecución: El usuario debe estar logueado. Tener productos comprados.	
Pasos de ejecución: El cliente debe hacer clic en la parte superior derecha de la página en el ícono "Usuario" y escoger la opción "Productos comprados".	
Resultados esperados: Descarga el archivo del producto seleccionado.	