A logo with blue letters

AI-generated content may be incorrect.

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD DE TECNOLOGÍAS EDUCATIVAS**

**Trabajo de Curso de Ingeniería de Software I con la integración de Proyecto de Investigación y Desarrollo 3 y Metodología de la Investigación Científica**

Título: Tienda online para la venta de videojuegos desarrollados por el centro VERTEX

Autores:

Alvaro Alfonso Rodríguez

Dainel Pérez Peñate

Rolando Samuel Cardet Navarro

Tutores:

Ailec Granda Dihigo

Reina Victoria Estrada Nelson

**La Habana, septiembre 2025**

**“Año 67 de la Revolución”**

## Resumen

*[En un solo párrafo, y hasta 250 palabras] (Arial, 12 puntos, interlineado 1.5, texto justificado)*

**Palabras clave:** *palabras 1; palabras 2; palabras 3; palabras 4; palabras 5 (Arial, 12 puntos, interlineado 1.5, texto justificado)*

*Entre 4 y 5 palabras clave, las cuales deben reflejar el contenido central del trabajo y ayudar a indizar el artículo.*

## Índice

Contenido

[Resumen 2](#_Toc216387938)

[Índice 3](#_Toc216387939)

[Introducción 5](#_Toc216387940)

[Capítulo I. Estudio del estado actual del proceso de planificación de cargas académicas en la UCI 9](#_Toc216387941)

[Introducción 9](#_Toc216387942)

[1.1 Contextualización del problema 9](#_Toc216387943)

[1.2 Análisis de mercado 11](#_Toc216387944)

[1.3 Fundamentación del proceso de desarrollo de software 14](#_Toc216387945)

[1.4 Herramientas y tecnologías 19](#_Toc216387946)

[Capítulo II. Modelado del Contexto 25](#_Toc216387947)

[Introducción 25](#_Toc216387948)

[2.1 Reglas del negocio 26](#_Toc216387949)

[2.2 Modelo Conceptual 30](#_Toc216387950)

[2.3 Modelo del Negocio 32](#_Toc216387951)

[Capítulo III. Modelado del Sistema 37](#_Toc216387952)

[Introducción 37](#_Toc216387953)

[3.1 Requisitos funcionales 37](#_Toc216387954)

[3.2 Requisitos no funcionales 40](#_Toc216387955)

[3.3 Documentación de requisitos 42](#_Toc216387956)

[Requisito: Registrar Solicitud de Devolución 44](#_Toc216387957)

[3.4 Validación de requisitos 45](#_Toc216387958)

[Conclusiones parciales 48](#_Toc216387959)

[Capítulo IV. Modelado del Comportamiento e Interacción 48](#_Toc216387960)

[Introducción 48](#_Toc216387961)

[4.1 Diagrama de Paquetes 49](#_Toc216387962)

[4.2 Diagramas de clases 49](#_Toc216387963)

[4.3 Diagramas de interacción 51](#_Toc216387964)

[Conclusiones parciales 52](#_Toc216387965)

[Bibliografía 1](#_Toc216387966)

[Anexos 2](#_Toc216387967)

## Introducción

El desarrollo de la industria de los videojuegos constituye uno de los fenómenos tecnológicos, culturales y económicos más relevantes de las últimas décadas. De acuerdo con Bain & Company (2024), el sector generó ingresos superiores a los 196 mil millones de dólares en 2023 y continúa expandiéndose a un ritmo anual del 6 %, impulsado por la digitalización, la consolidación de nuevas plataformas de distribución y la diversificación de los públicos consumidores. La expansión de este mercado no solo ha transformado el entretenimiento interactivo, sino que también ha abierto oportunidades en ámbitos educativos, científicos y de innovación tecnológica.

En este contexto global, las universidades tecnológicas juegan un papel clave en la generación de conocimiento y la producción de software creativo. La Universidad de las Ciencias Informáticas (UCI), desde su fundación en 2002, ha desarrollado un modelo de formación basado en la integración de la docencia, la investigación y la producción de software, donde los estudiantes participan activamente en proyectos reales que tributan al desarrollo del país. Entre estos proyectos, los videojuegos educativos y recreativos creados por los equipos de desarrollo del centro representan un producto de alto valor agregado, resultado de la aplicación práctica de los conocimientos adquiridos y de la creatividad tecnológica de la comunidad universitaria.

Sin embargo, a pesar del potencial creativo y técnico de los desarrolladores de la UCI, la comercialización y distribución de los videojuegos producidos enfrenta limitaciones significativas. Actualmente, la difusión de estos productos se realiza de forma manual, mediante redes sociales, ferias tecnológicas o enlaces directos, sin una plataforma institucional que garantice la venta segura, la visibilidad y la sostenibilidad de las producciones digitales del centro. Esta carencia limita la proyección externa de los videojuegos desarrollados, restringe sus posibilidades de generar ingresos y obstaculiza la consolidación de un ecosistema de innovación autosostenible dentro de la universidad.

La creciente relevancia del comercio electrónico y las plataformas digitales de distribución —como Steam, Epic Games Store o Itch.io— ha transformado los modelos tradicionales de venta de videojuegos. Según Meristation (2024), el 95 % de los ingresos mundiales del sector provienen ya de ventas digitales, tendencia que evidencia la obsolescencia de los canales físicos y la necesidad de infraestructuras propias para la comercialización en línea. No obstante, la mayoría de las soluciones disponibles en el mercado son de carácter internacional y no se ajustan a las condiciones técnicas, económicas ni legales del entorno cubano, donde las pasarelas de pago nacionales como EnZona y Transfermóvil constituyen alternativas viables para la implementación de sistemas locales de e-commerce.

En consecuencia, se hace necesario desarrollar una tienda online institucional, diseñada específicamente para la venta de videojuegos producidos por la UCI, que garantice seguridad, escalabilidad y una experiencia de usuario moderna. Esta plataforma no solo permitirá a los desarrolladores del centro comercializar sus productos de forma directa, sino que también contribuirá a visibilizar el talento nacional, fortalecer la economía del conocimiento y fomentar el emprendimiento universitario.

Además, la creación de un sistema propio responde a los lineamientos del Programa Nacional de Transformación Digital, que promueve la implementación de soluciones informáticas innovadoras desde las universidades cubanas. El proyecto propuesto se enmarca en la intersección de la Ingeniería de Software, el comercio electrónico y la industria cultural digital, lo que le confiere un carácter interdisciplinario y estratégico.

A partir del diagnóstico anterior, se formula la siguiente pregunta científica:

¿Cómo diseñar e implementar un sistema informático que permita la comercialización digital de los videojuegos desarrollados por la UCI, garantizando seguridad, eficiencia y accesibilidad en la gestión integral de los procesos de venta y promoción?

**Problema científico**

La ausencia de una infraestructura tecnológica que centralice la promoción y comercialización de los videojuegos desarrollados por la UCI limita la visibilidad, el control de licencias y la generación de ingresos, afectando el aprovechamiento del potencial innovador del centro y su contribución al desarrollo económico del país.

**Objeto de estudio**

El proceso de comercialización digital de productos informáticos desarrollados en entornos universitarios.

**Campo de acción**

El diseño e implementación de una tienda online accesible y segura para la venta de videojuegos desarrollados por la Universidad de las Ciencias Informáticas (UCI).

**Objetivo general**

Desarrollar un sistema informático que permita la gestión integral de una tienda online para la venta de videojuegos producidos por la UCI, garantizando la seguridad de las transacciones, la eficiencia operativa y la accesibilidad de los usuarios.

**Objetivos específicos**

Caracterizar el estado actual del proceso de comercialización digital de videojuegos en la UCI mediante el uso de métodos de ingeniería de software.

Analizar las tendencias del mercado de videojuegos y las plataformas internacionales de distribución digital.

Determinar los requisitos funcionales y no funcionales del sistema propuesto mediante técnicas de levantamiento y validación de requisitos.

Diseñar la arquitectura de software de la tienda online, asegurando su modularidad, escalabilidad y seguridad.

Modelar la base de datos que sustente la gestión de productos, usuarios, pedidos y transacciones.

Desarrollar e implementar los módulos de gestión de catálogos, usuarios, pasarelas de pago y analítica de ventas.

Evaluar la usabilidad, rendimiento y seguridad del sistema mediante pruebas funcionales y de aceptación.

Proponer estrategias de despliegue y sostenibilidad a largo plazo para la plataforma desarrollada.

**Tareas de investigación**

Revisión bibliográfica y análisis del estado del arte en plataformas de comercio electrónico de videojuegos.

Diagnóstico del proceso actual de distribución digital en la UCI.

Identificación de actores, procesos y requerimientos del sistema propuesto.

Elaboración del modelo conceptual y diseño arquitectónico de la solución.

Selección de tecnologías, frameworks y herramientas para el desarrollo.

Implementación incremental del sistema informático.

Aplicación de pruebas funcionales, de rendimiento y seguridad.

Evaluación de resultados y validación de los objetivos planteados.

**Delimitación del estudio**

El proyecto se circunscribe al contexto institucional de la Universidad de las Ciencias Informáticas (UCI), específicamente a los videojuegos desarrollados por sus centros de producción y laboratorios académicos. No aborda la venta de productos externos ni la implementación de pasarelas de pago internacionales, concentrándose en la integración con servicios nacionales de pago electrónico y en la difusión digital de contenido propio.

**Importancia del estudio**

El desarrollo de la tienda online propuesta constituye un paso esencial para consolidar el ecosistema digital de la UCI, fortaleciendo el vínculo entre la producción tecnológica, la formación profesional y la innovación económica. A su vez, este sistema permitirá a la universidad posicionarse como pionera en la comercialización de software interactivo dentro del contexto cubano, contribuyendo a la soberanía tecnológica y al fomento del emprendimiento estudiantil.

## Capítulo I. Estudio del estado actual del proceso de planificación de cargas académicas en la UCI

### Introducción

El avance del comercio electrónico ha impulsado un cambio sustancial en los modelos de producción y distribución del software. Las empresas, instituciones y centros de investigación han adoptado plataformas digitales no solo como medios de venta, sino también como espacios de interacción con los usuarios, análisis de comportamiento de consumo y control de procesos comerciales. En este contexto, la gestión de productos digitales se ha convertido en un factor determinante para la sostenibilidad de los proyectos tecnológicos.

En la Universidad de las Ciencias Informáticas (UCI), donde la producción de software forma parte integral del proceso docente, la comercialización digital adquiere un valor estratégico. La institución dispone de un capital humano altamente calificado y de una infraestructura técnica que la posicionan favorablemente para emprender proyectos de distribución digital de sus productos informáticos. Sin embargo, no existen actualmente sistemas específicos que permitan gestionar de forma eficiente la venta, seguimiento y promoción de los videojuegos desarrollados por el centro.

Este capítulo tiene como propósito analizar el estado actual del proceso de comercialización digital dentro de la UCI, identificando las limitaciones, oportunidades y fundamentos teóricos que sustentan el desarrollo de una tienda online institucional. A partir de la revisión de referentes internacionales, nacionales y académicos, se establecen las bases conceptuales y metodológicas que orientarán el diseño de la solución informática propuesta.

### 1.1 Contextualización del problema

La gestión de la comercialización digital de software en instituciones universitarias enfrenta desafíos que difieren de los entornos empresariales tradicionales. En el caso de la UCI, los productos generados —como aplicaciones móviles, sistemas web o videojuegos— surgen de proyectos académicos y productivos que, una vez concluidos, carecen de mecanismos sistematizados para su promoción y distribución. Este vacío provoca que muchas de las soluciones desarrolladas queden restringidas a usos internos o a una difusión limitada, desaprovechando su potencial para alcanzar públicos más amplios.

En el caso particular de los videojuegos desarrollados por el centro, los procesos de difusión y acceso se realizan mediante canales no especializados, tales como intercambios directos, redes sociales o eventos presenciales. Esta práctica dificulta la trazabilidad del producto, la gestión de actualizaciones, el control de licencias y la obtención de estadísticas de uso. Además, la falta de una plataforma unificada impide una relación directa y estable con los usuarios, limitando las oportunidades de retroalimentación y mejora continua.

El escenario internacional evidencia una realidad contrastante. El modelo de distribución digital directa al consumidor (D2C) domina actualmente el mercado de los videojuegos, favoreciendo la autonomía de los desarrolladores y la diversificación de los modelos de negocio. Según Bain & Company (2025), la expansión de plataformas en línea ha permitido a pequeñas y medianas entidades comercializar sus productos sin necesidad de intermediarios, incrementando la eficiencia y el alcance de sus operaciones.

La inexistencia de una tienda institucional en la UCI no solo representa una limitación tecnológica, sino también una contradicción estratégica con el modelo de formación-producción que caracteriza a la universidad. La integración de una plataforma de venta digital permitiría cerrar el ciclo completo del desarrollo de software: desde la ideación y diseño hasta la distribución y análisis postventa, fortaleciendo el vínculo entre la academia y la práctica profesional.

Por tanto, el proceso de comercialización digital en la UCI requiere de una transformación estructural que incorpore herramientas tecnológicas modernas y metodologías de gestión que garanticen eficiencia, seguridad y sostenibilidad. El diseño de una tienda online institucional se presenta como una respuesta coherente a esta necesidad.

Comercialización Digital de Productos Informáticos: Es el proceso integral de promoción, venta, distribución y licenciamiento de software, videojuegos u otras creaciones digitales a través de canales electrónicos. En el contexto de la UCI, se centra en la venta directa al consumidor (D2C).

Tienda Online (E-commerce): Es una plataforma informática diseñada para la venta de bienes o servicios a través de Internet. En el presente proyecto, es el componente central para centralizar la promoción y venta de los videojuegos del centro VERTEX.

Videojuego (Producto Digital): El activo digital principal desarrollado por equipos de la UCI y comercializado a través de la tienda. La venta se refiere a la Licencia digital de uso y no a la transferencia de la propiedad intelectual.

Licenciamiento Digital: El mecanismo de control y gestión que otorga al cliente el derecho de uso del software tras una compra exitosa. La clave de licencia debe ser única e intransferible (RN-06).

Pasarela de Pago Nacional: Sistemas financieros electrónicos (como EnZona y Transfermóvil) que permiten la liquidación de transacciones dentro del marco legal y económico cubano (RN-02).

Seguridad de las Transacciones y Activos: El conjunto de medidas tecnológicas y operativas que garantizan la integridad de los pagos (RnF-04), la protección de los datos sensibles (RnF-02) y la prevención del uso indebido de las licencias (RN-12).

### 1.2 Análisis de mercado

El análisis del mercado de videojuegos permite contextualizar la pertinencia de desarrollar una plataforma de distribución digital propia. En los últimos años, la industria global ha experimentado un crecimiento sostenido, convirtiéndose en uno de los sectores más rentables del entretenimiento interactivo. De acuerdo con Bain & Company (2024), los ingresos mundiales del sector alcanzaron los 196 mil millones de dólares en 2023, impulsados por el incremento de jugadores en dispositivos móviles y el auge de los servicios de distribución digital.

El desplazamiento casi total del formato físico por el digital constituye un hito en la evolución de esta industria. Reportes de *GamesIndustry.biz* y *Meristation* (2024) estiman que más del 95 % de los videojuegos vendidos durante 2023 se comercializaron a través de plataformas en línea, consolidando la hegemonía del comercio electrónico como principal canal de distribución.

En este escenario, la creación de plataformas locales o institucionales se ha convertido en una tendencia emergente, especialmente en universidades y centros de investigación tecnológica. Estas iniciativas no solo permiten difundir los productos generados internamente, sino que también actúan como entornos de aprendizaje para el desarrollo de competencias en comercio digital, diseño de interfaces y gestión de datos.

A nivel nacional, la infraestructura para el comercio electrónico ha mostrado avances significativos con la consolidación de pasarelas como **EnZona** y **Transfermóvil**, que ofrecen entornos seguros para la gestión de transacciones financieras electrónicas. Estas herramientas constituyen una base sólida sobre la cual implementar soluciones locales de venta digital, adaptadas a las condiciones tecnológicas y económicas del país.

El análisis comparativo de homólogos internacionales y nacionales permite identificar brechas relevantes. Plataformas como **Steam**, **Epic Games Store** o **Itch.io** ofrecen experiencias de usuario avanzadas y modelos de monetización diversificados; sin embargo, su integración con sistemas de pago cubanos es inviable por las restricciones de conectividad y los marcos regulatorios. En contraste, la propuesta de la UCI debe orientarse hacia una **plataforma ligera, segura y autosuficiente**, que combine la funcionalidad esencial del comercio digital con la independencia tecnológica y la soberanía sobre los datos y licencias.

Por tanto, el estudio de mercado confirma la necesidad de desarrollar una tienda online propia que permita a la UCI insertarse de forma competitiva en la economía digital y aprovechar el valor agregado de su producción académica y tecnológica.

| **Plataforma Global Analizada** | **Característica Clave Estudiada** | **Restricción/No Usabilidad en Cuba** | **Impacto en el Proyecto VERTEX** |
| --- | --- | --- | --- |
| **Steam, Epic Games, Nintendo eShop, etc.** | **Integración con Pasarelas de Pago Internacionales (Visa/Mastercard, PayPal)** | Requieren cuentas bancarias, pasarelas de pago y sistemas de liquidación que operan exclusivamente bajo regulaciones financieras de EE. UU. o la UE, las cuales imponen **restricciones comerciales** a Cuba. | La UCI no puede recibir los fondos de la venta de sus videojuegos a través de estas plataformas de manera directa, legal y sistemática. |
| **Steam, Google Play, Apple App Store** | **Políticas de Distribución Geográfica (Geo-Blocking)** | Las plataformas globales suelen aplicar bloqueos geográficos para evitar la venta de activos digitales en territorios sujetos a sanciones (incluido Cuba), o tienen procesos de verificación de desarrolladores que son inaccesibles. | Los videojuegos de VERTEX, desarrollados en Cuba, serían **invisibles** o **bloqueados** para el público internacional o incluso el nacional que acceda con IP extranjera. |
| **Epic Games Store, GOG** | **Modelo de Monetización y Comisiones (%)** | Las plataformas retienen un porcentaje significativo de las ventas (típicamente entre 12% y 30%). Aunque este es un modelo de negocio, la imposibilidad de recibir el 100% restante se mantiene. | El modelo de negocio del proyecto (financiación de investigación y desarrollo) requiere una **gestión total de los ingresos**, lo que se logra mejor sin intermediarios extranjeros. |
| **PlayStation Store, Xbox Store** | **Requisitos de Certificación de Consolas (Kits de Desarrollo)** | La distribución en estas plataformas exige licencias, kits de desarrollo y procesos de certificación muy rigurosos, a menudo controlados por regulaciones de exportación de EE. UU. | **Impide la distribución** de videojuegos desarrollados en la UCI para consolas, lo que limita el proyecto a la venta de activos digitales para PC o móvil a través de una plataforma propia. |
| **Todas las Plataformas Globales** | **Uso de Criptomonedas o Métodos de Pago Locales (e.g., Alipay, India UPI)** | Si bien algunas aceptan nuevos métodos, ninguna plataforma global tiene integración con los **sistemas de pago cubanos** legalmente establecidos (**EnZona** y **Transfermóvil**). | La tienda de VERTEX **debe ser compatible** con los mecanismos de pago electrónicos nacionales como requisito fundamental para el mercado cubano (RN-02). |

### 1.3 Fundamentación del proceso de desarrollo de software

El proceso de desarrollo de la tienda online para la venta de videojuegos desarrollados por la UCI exige una fundamentación metodológica sólida que contemple no solo la construcción de software, sino también la integración de políticas institucionales, requisitos de seguridad y criterios de sostenibilidad. Este apartado expone con mayor detenimiento el enfoque de ingeniería de software adoptado, los modelos de proceso seleccionados y el método de trabajo que se ejecutará durante las fases de diseño e implementación.

#### 1.3.1 Enfoque de ingeniería de software

El enfoque de ingeniería de software propuesto para este proyecto se justifica rigurosamente mediante la aplicación de criterios formales de selección de metodologías, tal como lo requiere el marco académico. Para ello, se utiliza el **Modelo de Evaluación de Metodologías de Boehm & Turner**, representado gráficamente mediante un diagrama estelar, que evalúa las características del proyecto y del equipo en cinco dimensiones críticas: Criticidad, Dinamismo, Tamaño del Equipo, Cultura Organizacional y Habilidades del Personal.

**Justificación basada en el Modelo de Boehm & Turner (Tienda Online):**

1. **Criticidad (Media-Alta):** El sistema maneja transacciones financieras reales (Pasarelas Nacionales) y activos intelectuales (Licencias Digitales). Un fallo podría resultar en pérdidas monetarias directas, daño a la **reputación institucional** en el comercio electrónico, y conflictos de Propiedad Intelectual. Esto exige un rigor metodológico alto para la verificación y validación de componentes sensibles.
2. **Dinamismo (Medio-Alto):** El proyecto se inserta en un mercado de comercio electrónico que evoluciona rápidamente (tendencias de UX, pasarelas de pago sujetas a actualizaciones) y sus requisitos (RN-09, RN-12) están sujetos a ajustes basados en políticas de uso y normativas. Se estima que la tasa de cambio de requisitos puede ser del 40-50%.
3. **Tamaño del Equipo (Pequeño):** El equipo es reducido (típico de un trabajo de curso, 3 desarrolladores). Esto favorece la comunicación ágil, pero requiere una estructura que maximice la eficiencia del esfuerzo individual.
4. **Cultura Organizacional (Mixta/Media):** La Universidad de las Ciencias Informáticas (UCI) promueve la innovación (cultura ágil), pero mantiene requisitos institucionales de **documentación formal** y **trazabilidad** (cultura tradicional) para la transferencia de software.
5. **Habilidades del Personal (Media-Baja):** El equipo posee conocimientos técnicos sólidos, pero experiencia limitada en la gestión de proyectos de alta criticidad (pagos, seguridad, licenciamiento). Se requiere una metodología que provea orientación sin inhibir la curva de aprendizaje.

La ubicación de esta evaluación en el diagrama estelar de Boehm & Turner apunta claramente hacia la necesidad de un enfoque que balancee el **control** y la **rapidez**. Por lo tanto, se confirma la adopción de un **Modelo Híbrido** que combina elementos de **Scrum** (para iteraciones rápidas) y **RUP** (para rigor en la arquitectura y documentación formal).

**Estructura del Enfoque Híbrido:**

Este enfoque concilia las exigencias mediante las siguientes decisiones concretas, asegurando agilidad sin sacrificar la gobernanza:

* **Entrega Incremental orientada al valor:** Cada iteración (*sprint*) entregará funcionalidades que aporten valor directo (registro y autenticación, catálogo básico, carga de productos, carrito y proceso de compra simulado y luego real con pasarela nacional). Esto permite evaluar la aceptación y realizar correcciones tempranas.
* **Control de Calidad en Capas Críticas:** Módulos sensibles (pasarelas, encriptación, auditoría) estarán sujetos a **revisiones técnicas formales** (elemento RUP), pruebas de seguridad y auditorías de código antes de su puesta en producción.
* **Participación Continua de Stakeholders:** Se establecerán sesiones periódicas de validación con representantes del centro y usuarios piloto para validar usabilidad, políticas de licenciamiento y procesos de soporte (elemento Scrum).
* **Documentación y Configuración Gestionadas:** Todos los artefactos se versionarán y documentarán (*specs*, diagramas, *scripts* de despliegue) para cumplir con requerimientos institucionales y facilitar la transferencia y mantenimiento.

Este enfoque equilibra agilidad y gobernanza, y está pensado para un equipo académico-pequeño que requiere orientación metodológica sin sofocar la creatividad y el aprendizaje.

#### 1.3.2 Modelo(s) de proceso de software

A partir del enfoque descrito se define el modelo de proceso híbrido que orientará la ejecución. El modelo combina elementos de **Scrum** (para la operacionalización iterativa) y de **RUP** (para actividades de arquitectura, calidad y consolidación documental), estructurado en las siguientes capas:

* **Macrofases (RUP-like):**
  + *Iniciación y Visión:* definición de alcance, identificación de actores clave, contexto legal/administrativo y decisión arquitectónica de alto nivel.
  + *Elaboración de Arquitectura:* diseño de la arquitectura técnica (capas, servicios, seguridad) y establecimiento de criterios de aceptación para componentes críticos.
  + *Construcción Iterativa:* ciclos de implementación guiados por prioridades de negocio.
  + *Transición/Despliegue:* preparación del entorno productivo, pruebas finales y capacitación.
* **Microciclos (Scrum):**
  + *Sprints de 2 semanas* con backlog priorizado (historias de usuario, tareas técnicas, bugs).
  + *Ceremonias ágiles:* planning, daily stand-up, demo de sprint, retrospectiva.
  + *Artefactos:* product backlog, sprint backlog, incremento potencialmente entregable.
* **Prácticas complementarias:**
  + *Integración continua (CI):* pipelines automatizados para build, pruebas unitarias y pruebas de integración.
  + *Despliegue por contenedores:* imágenes Docker y orquestación mínima para entornos de prueba y producción.
  + *Pruebas de seguridad y revisiones formales:* escáneres estáticos y dinámicos, pruebas de penetración (scope limitado) para módulos de pagos y autenticación.

Este diseño de proceso reduce riesgos al garantizar entregas tempranas y controles formales periódicos sobre los componentes más sensibles del sistema.

#### 1.3.3 Método de ingeniería de software

El método práctico aplicado en el ciclo de vida del proyecto integra actividades genéricas (según Pressman) y un conjunto de actividades sombrilla que garantizarán la calidad y continuidad. Se describen los pasos y responsabilidades principales:

1. **Comunicación y adquisición de requisitos**
   * Técnicas: entrevistas semiestructuradas, talleres con grupos focales, encuestas a potenciales usuarios (estudiantes, profesores, personal técnico).
   * Entregables: visión del producto, lista priorizada de historias de usuario, mapa de actores y restricciones legales (licencias, derechos de autor, protección de datos).
2. **Planificación**
   * Definición de releases y sprints, estimación de esfuerzo (puntos de historia), asignación de roles (Scrum Master/Tutor, Product Owner representante del centro, desarrolladores).
   * Elaboración del plan de pruebas y criterios mínimos de aceptación.
3. **Modelado y diseño**
   * Modelado del dominio con UML (casos de uso, diagrama de clases inicial, ER para la BD).
   * Diseño arquitectónico: separación en capas (API, lógica de negocio, persistencia), definición de contratos REST y esquema de seguridad.
   * Documentación de políticas de licenciamiento y versiones del producto (demo, free, premium).
4. **Construcción**
   * Implementación incremental con integración continua.
   * Cobertura de pruebas unitarias y pruebas automatizadas de UI para los flujos críticos (registro, compra, descarga).
   * Revisión de código y control de configuración (GitLab con merge requests y pipelines).
5. **Despliegue y operación**
   * Despliegue en infraestructura institucional (servidores UCI o nube privada), configuración de backups y planes de recuperación.
   * Monitoreo inicial de uso, logs y métricas (tiempo de respuesta, errores, transacciones).
6. **Actividades sombrilla**
   * Gestión de la calidad (QA), gestión de la configuración, gestión de riesgos (listado con mitigaciones), aseguramiento de la accesibilidad y usabilidad.
   * Plan de transferencia de conocimiento y capacitación para el personal de soporte.

El método garantiza que el equipo académico aprenda práctica profesional mientras produce un sistema acorde con los estándares institucionales y los requisitos del dominio e-commerce.

### 1.4 Herramientas y tecnologías

La selección de herramientas y tecnologías no es solo técnica sino estratégica: debe facilitar la operación en la infraestructura de la UCI, maximizar la reutilización de conocimiento local y permitir futuras extensiones. A continuación se detallan los componentes y su justificación.

#### 1.4.1 Herramienta CASE

Para el modelado y documentación se empleará **Visual Paradigm 17.3** (o versión institucional equivalente). Esta herramienta permitirá la generación de diagramas UML (casos de uso, clases, secuencia, actividades), la elaboración de modelos ER y la exportación de artefactos que sirven de referencia para la implementación. El uso de un CASE facilita además la trazabilidad entre requisitos y diseño, y produce documentación estandarizada que cumple con los criterios académicos exigidos.

#### 1.4.2 Lenguaje de modelado

Se adoptará **UML 2.5.x** como lenguaje de modelado para representar de manera formal las estructuras y comportamientos del sistema. UML será utilizado para:

* Diagrama de casos de uso (definir interacciones actor-sistema).
* Diagrama de clases conceptuales y de diseño (entidades: videojuego, producto, licencia, usuario, pedido, transacción).
* Diagramas de secuencia para flujos críticos (proceso de compra, emisión de licencia, actualización de producto).
* Diagrama de actividades para procesos de negocio (gestión de devoluciones, soporte).

El empleo consistente de UML facilita la comunicación entre el equipo técnico y los tutores/actores institucionales.

#### 1.4.3 Marco de trabajo para el desarrollo de la solución informática

La solución técnica se sustentará en una **arquitectura basada en servicios** con separación de responsabilidades: frontend UI, backend API RESTful, y capa de persistencia. Las decisiones clave incluyen:

* **API-first:** todas las funcionalidades del backend se exponen mediante API REST con contratos bien definidos, lo que permite una evolución independiente del cliente (web o app).
* **Autenticación y autorización centralizadas:** modelo de roles (administrador, desarrollador, comprador, soporte) y uso de JWT para sesiones stateless.
* **Modularidad para expansión:** componentes desacoplados (catálogo, pagos, licencias, analítica) que pueden ser desplegados y escalados independientemente.
* **Estrategia de integraciones:** conectores con servicios nacionales de pago (EnZona/Transfermóvil) y mecanismos para futuras integraciones (por ejemplo, CDN para descargas seguras).

Este marco favorece mantenibilidad y la posibilidad de reutilizar módulos en otros proyectos institucionales.

#### 1.4.4 Entorno de desarrollo integrado

El entorno preferido será **Visual Studio Code** por su versatilidad para proyectos full-stack (Python/Django y TypeScript/Angular). Se configurará con extensiones para linting, depuración, test runner y Docker. Además, la integración con GitLab permite pipelines CI/CD que automaticen build, pruebas y despliegue. Para tareas específicas (modelado, bases de datos) se utilizarán herramientas complementarias (pgAdmin para PostgreSQL, clientes REST para pruebas).

#### 1.4.5 Lenguaje de programación

La pila tecnológica propuesta prioriza lenguajes y frameworks con amplia adopción académica y empresarial:

* **Python (Django)** para el backend: ofrece rapidez de desarrollo, ORM robusto y mecanismos de seguridad integrados.
* **TypeScript (Angular)** para el frontend: garantiza tipado estático, escalabilidad y consistencia en aplicaciones SPA.
* **JavaScript/HTML/CSS** para las partes de interfaz y microinteracciones, apoyadas por librerías de componentes accesibles.

Esta combinación facilita la formación de equipo (estudiantes ya familiarizados con Python y JS) y asegura compatibilidad con herramientas de la UCI.

#### 1.4.6 Gestor de base de datos

Se opta por **PostgreSQL** como gestor relacional por su solidez, cumplimiento ACID y características avanzadas (vistas materializadas, indexación, procedimientos almacenados cuando se requiera). El diseño de la BD considerará:

* Entidades principales: **Usuario**, **Desarrollador**, **Videojuego (Producto)**, **Versión**, **Licencia**, **Pedido**, **Transacción**, **Historial de descargas**, **Soporte/Incidencia**.
* Modelado de licencias y claves de descarga: tablas relacionadas que permitan revocar licencias, emitir actualizaciones y gestionar versiones.
* Auditoría y trazabilidad: tablas y triggers para registro de acciones críticas (compras, cambios en precios, emisión de reembolsos).

Se definirá además la estrategia de backups, métricas de crecimiento y particionamiento/archivado en función de la carga esperada.

#### 1.4.7 Herramientas

Para apoyar el proceso de desarrollo, gestión y aseguramiento de la calidad del proyecto se selecciona un conjunto de herramientas complementarias que facilitan la colaboración, el control de tareas, la automatización de pruebas y la documentación del ciclo de vida del software. La elección responde a criterios de accesibilidad para el equipo académico, integración con la infraestructura institucional y capacidad para soportar prácticas profesionales durante la ejecución del trabajo de curso.

* **Gestión de proyectos y seguimiento de tareas:** *ClickUp* (o una alternativa institucional equivalente) será utilizada como herramienta principal para la planificación de releases, gestión del backlog, asignación de tareas y seguimiento de incidencias. ClickUp permite configurar tableros ágiles (Kanban/Scrum), establecer prioridades, vincular tareas con documentos y generar reportes de progreso, lo que facilita la coordinación entre los miembros del equipo, los tutores y los stakeholders de la UCI. En caso de restricciones de acceso a soluciones comerciales, se contempla la utilización de *GitLab Issues* o *Taiga* como alternativas libres y compatibles con GitLab UCI.
* **Automatización de pruebas funcionales y de interfaz:** *Selenium 4* se empleará para la automatización de pruebas end-to-end sobre la interfaz web, cubriendo flujos críticos como registro, autenticación, proceso de compra, emisión de licencias y descarga segura de productos. La automatización con Selenium permitirá ejecutar pruebas de regresión en las pipelines CI/CD, reducir la frecuencia de errores en producción y documentar comportamientos esperados del sistema. Complementariamente, se utilizarán frameworks de testing unitario (pytest para Python, Karma/Jasmine para Angular) para asegurar la cobertura de componentes.
* **Gestión de repositorios y CI/CD:** *GitLab* (instancia institucional) se empleará para alojar el código fuente, gestionar control de versiones y orquestar pipelines de integración continua y despliegue. Las pipelines configuradas automatizarán la construcción de imágenes Docker, ejecución de pruebas automatizadas y despliegue a entornos de staging, contribuyendo a un ciclo de entrega confiable y repetible.
* **Monitorización y analítica básica:** Para el seguimiento operativo inicial se integrarán herramientas ligeras de monitorización y registro (por ejemplo, *Prometheus* + *Grafana* o soluciones institucionales equivalentes) que recojan métricas de rendimiento (latencia, tasa de errores, uso de CPU/memoria) y eventos de negocio (número de transacciones, tasa de conversión). Además, se contemplará la instrumentación de eventos analíticos (p. ej., eventos de compra, abandono de carrito) que permitan realizar análisis posteriores de comportamiento de usuarios.
* **Gestión de incidencias y soporte al usuario:** Se definirá un sistema de tickets (puede usarse *GitLab Issues* o un gestor de helpdesk ligero) para el manejo de consultas, reclamaciones y solicitudes de soporte técnico asociadas a la tienda. Esto facilitará la trazabilidad de problemas y la mejora continua del servicio.
* **Documentación y colaboración:** Para la documentación técnica y de usuario se empleará un repositorio wiki integrado (por ejemplo, *GitLab Wiki* o *Confluence* si está disponible institucionalmente). Se documentarán manuales de usuario, guías de despliegue, políticas de licenciamiento y procedimientos de operación.
* **Herramientas de diseño y prototipado:** Para la definición temprana de interfaces de usuario y validaciones de usabilidad se utilizarán herramientas de prototipado (Figma, Adobe XD o alternativas locales), que permitan construir prototipos interactivos y realizar pruebas de usabilidad con usuarios reales del centro.
* **Herramientas para seguridad y análisis de código:** Se incorporarán escáneres estáticos de seguridad (SAST) y linters en las pipelines (por ejemplo, *Bandit* para Python, *ESLint* para TypeScript) y herramientas de análisis de dependencias para detectar vulnerabilidades conocidas. Estas herramientas soportarán las políticas de revisión técnica y auditoría definidas en el método de trabajo.

La combinación de estas herramientas permitirá al equipo no solo desarrollar el producto con prácticas profesionales, sino también obtener evidencias tangibles de calidad, trazabilidad y sostenibilidad operativa. Además, su integración con la infraestructura de la UCI asegurará que los artefactos producidos —código, documentación, decisiones de arquitectura y registros de pruebas— queden debidamente almacenados y disponibles para futuras iteraciones o para la adopción institucional del servicio.

#### Conclusiones parciales

El análisis realizado en este capítulo permitió caracterizar el estado actual del proceso de comercialización digital en la UCI, identificando carencias estructurales y oportunidades de desarrollo. Se constató la ausencia de herramientas especializadas para la venta y distribución de videojuegos, lo cual limita la visibilidad, la rentabilidad y el aprovechamiento de los productos informáticos generados por el centro.

El estudio del mercado evidenció una tendencia global irreversible hacia la digitalización del comercio de videojuegos, sustentada en plataformas que priorizan la experiencia de usuario, la seguridad y la accesibilidad. En este contexto, la implementación de una tienda online institucional se presenta como una alternativa viable y necesaria, que permitirá a la UCI posicionarse como referente nacional en la comercialización de software educativo y recreativo.

La fundamentación metodológica basada en modelos híbridos de ingeniería de software y el uso de tecnologías modernas de código abierto confirman la viabilidad técnica del proyecto y su alineación con los objetivos estratégicos de la universidad. En consecuencia, el desarrollo del sistema propuesto representa un paso significativo hacia la consolidación de un ecosistema digital autosostenible dentro de la UCI, capaz de articular producción, innovación y emprendimiento.

## Capítulo II. Modelado del Contexto

### Introducción

El presente capítulo establece los fundamentos conceptuales y operativos que rigen el proceso de comercialización digital de videojuegos en el centro VERTEX de la Universidad de las Ciencias Informáticas (UCI). La formalización del negocio es crucial para garantizar que el sistema informático propuesto automatice y optimice las operaciones comerciales de manera precisa, particularmente en la gestión de activos digitales y la integración con sistemas financieros nacionales.

Mediante la aplicación de técnicas de modelado de negocios, se procede a la identificación rigurosa de las Reglas del Negocio que definen las políticas operacionales y las restricciones legales inherentes al comercio electrónico cubano y la distribución de licencias digitales. Posteriormente, se elabora el Modelo Conceptual, que organiza la terminología del dominio en entidades clave, y finalmente, el Modelo del Negocio, que formaliza los roles y los flujos de interacción mediante Casos de Uso del Negocio. Este enfoque estructurado sienta las bases semánticas y procesales necesarias para la posterior especificación de los requisitos del software.

### 2.1 Reglas del negocio

Las reglas del negocio son declaraciones estructuradas que definen o restringen algún aspecto del funcionamiento de la organización, en este caso, el centro VERTEX y su operación de venta online. Su especificación es vital para traducir las políticas institucionales de la UCI, las restricciones operativas, y las características únicas de la venta de software digital a criterios verificables que el sistema debe hacer cumplir.

La naturaleza del producto, al ser software (un videojuego), impone reglas que van más allá de un e-commerce de bienes físicos. Por ejemplo, el activo principal que se vende no es la copia física, sino una licencia digital de uso. Esta distinción fundamental se formaliza en la regla RN-01 para asegurar la protección de la propiedad intelectual de la universidad y su equipo de desarrolladores.

Otro conjunto de reglas se deriva de la restricción operativa del entorno nacional. La tienda debe operar exclusivamente con pasarelas de pago cubanas como EnZona o Transfermóvil. Esta limitación introduce una capa de complejidad en la gestión de la confiabilidad, ya que el sistema no puede depender de procesos de *rollback* o conciliación automáticos estándar de plataformas internacionales. En consecuencia, la regla de Inferencia RN-04 se vuelve esencial, ya que el sistema debe ser capaz de detectar fallos y alertar al personal para una conciliación manual en caso de transacciones en estado indefinido, garantizando la integridad financiera.

La regla RN-08 asegura la trazabilidad de la propiedad intelectual, asociando cada producto al centro VERTEX. Esto permite realizar auditorías futuras y gestionar cualquier modelo de regalías o distribución de ingresos que pueda establecer la UCI, reforzando el ciclo de producción e innovación. La RN-06 (Clave Única) y la RN-12 (Bloqueo por Uso Indebido) están directamente relacionadas con la RN-01, protegiendo el activo digital contra la piratería y el uso compartido excesivo, un control necesario dado que el producto es una licencia limitada.

Reglas del Negocio de la Tienda Online VERTEX

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Clasificación** | **Nombre** | **Descripción** |
| RN-01 | Hecho | Naturaleza del Producto | El producto vendido es una licencia digital de uso de software (videojuego), no la transferencia de la propiedad intelectual ni del código fuente. |
| RN-02 | Restricción | Pasarelas de Pago Habilitadas | Solo se procesarán transacciones mediante la integración con las pasarelas de pago nacionales EnZona o Transfermóvil. |
| RN-03 | Restricción | Control de Acceso por Licencia | Solo los Clientes con una licencia activa pueden acceder al enlace de descarga o a las actualizaciones. |
| RN-04 | Inferencia | Alerta de Fallo de Transacción | Si la Pasarela de Pago devuelve un código de error durante la compra, el sistema debe registrar el incidente y notificar al cliente y al Administrador de Tienda para conciliación manual. |
| RN-05 | Facilitador de Acción | Emisión de Licencia Automática | Una vez confirmada la recepción del pago, el sistema debe generar automáticamente la clave de licencia y habilitar el enlace de descarga temporal. |
| RN-06 | Restricción | Clave de Licencia Única | Toda transacción exitosa debe generar una clave de licencia única e intransferible, la cual es la base del derecho de uso. |
| RN-07 | Computacional | Cálculo de Precio Final | El sistema calcula el precio final sumando el precio base del producto más impuestos o comisiones aplicables, según la política institucional. |
| RN-08 | Hecho | Trazabilidad del Desarrollador | Cada Videojuego debe estar asociado a su Centro de Producción (VERTEX) y al equipo de desarrollo responsable para efectos de gestión de la IP y auditoría. |
| RN-09 | Restricción | Devoluciones Limitadas | Las devoluciones solo se permiten si la licencia no ha sido utilizada para iniciar la descarga del software o si se notifica un defecto crítico dentro de las 72 horas posteriores a la compra. |
| RN-10 | Computacional | Detección de Desviación de Precio | El sistema debe comparar el precio de venta actual con el precio promedio histórico. Si la desviación supera el 15%, genera una alerta al Gestor de Catálogo. |
| RN-11 | Inferencia | Detección de Transacción Duplicada | Si se recibe un *callback* de pago con un ID de transacción ya registrado, el sistema debe inferir un intento de duplicidad y marcar la transacción como potencialmente fraudulenta. |
| RN-12 | Facilitador de Acción | Bloqueo por Uso Indebido | Si se supera el límite de descargas o se detecta acceso desde múltiples IPs, el sistema debe suspender temporalmente la licencia hasta la revisión por el Gestor de Soporte (RN-05). |

### 2.2 Modelo Conceptual

El Modelo Conceptual define la estructura semántica del dominio de la Tienda Online, identificando los conceptos fundamentales y sus interrelaciones clave. En un e-commerce de productos digitales, la abstracción debe diferenciar claramente el contenido intelectual (Videojuego) del derecho de uso (Licencia) y del registro financiero (Transacción).

La distinción entre el concepto de Videojuego (la propiedad intelectual y metadatos) y Versión (el archivo binario específico) es crucial. El Cliente adquiere la Licencia para utilizar una o varias Versiones de un Videojuego. Esta separación soporta la capacidad de actualizar el software a través de parches o mejoras sin requerir una nueva compra, siempre que la Licencia siga activa.

La Transacción y la Licencia están interconectadas a través del Pedido. El Pedido centraliza la intención de compra. La Transacción es el registro del movimiento de dinero y la Licencia es el derecho de acceso que se genera como resultado del éxito de la Transacción. Este vínculo es fundamental para la auditoría, la gestión de devoluciones y el cumplimiento de la RN-06.

Descripción de las Clases del Modelo Conceptual

|  |  |  |
| --- | --- | --- |
| **No.** | **Nombre de la Clase** | **Descripción** |
| 1 | Cliente | Entidad que representa al comprador o usuario registrado que interactúa con la tienda, realiza pedidos y consume los productos. |
| 2 | Desarrollador | Entidad de la UCI que publica el videojuego (VERTEX) y gestiona sus actualizaciones. |
| 3 | Videojuego (Producto) | El activo digital principal (metadatos, descripción) comercializado. |
| 4 | Versión (Software) | Iteración específica y descargable del binario de software. |
| 5 | Pedido | Agrupa la intención de compra, los ítems seleccionados y el monto total, sirviendo como contenedor de la venta. |
| 6 | Transacción | Registra el estado, la fecha y el identificador único de la operación financiera con la Pasarela de Pago. |
| 7 | Licencia | El derecho de uso otorgado al Cliente tras el pago exitoso, con clave única y estado (Activa, Revocada). |
| 8 | Descarga | Registro de los eventos de acceso al archivo binario por el Cliente, incluyendo fecha, hora e IP, esencial para el monitoreo de la RN-12. |

### 2.3 Modelo del Negocio

El Modelado del Negocio se centra en la identificación de los roles que participan e interactúan con la Tienda Online, distinguiendo entre actores externos y trabajadores internos.

#### 2.3.1 Actores del negocio

Los actores del negocio son roles externos a la organización que interactúan con el sistema para lograr sus propios objetivos.

Actores del Negocio

|  |  |
| --- | --- |
| **Actor** | **Descripción** |
| Cliente | Usuario final que consume el servicio y el producto, e inicia el proceso de compra. |
| Desarrollador (Equipos VERTEX) | Rol externo que provee el contenido, gestiona la subida de archivos binarios y actualiza los metadatos del producto. |

#### 2.3.2 Trabajadores del negocio

Los trabajadores del negocio son roles internos que ejecutan las tareas y operaciones esenciales para el funcionamiento del e-commerce.

Trabajadores del Negocio

|  |  |
| --- | --- |
| **Trabajador** | **Descripción** |
| Administrador de Tienda | Responsable de la supervisión general del sistema, la configuración de la Pasarela de Pago y el monitoreo de la seguridad. |
| Gestor de Catálogo/Producto | Encargado de la gestión de inventario digital, precios, descuentos y la aprobación de nuevas versiones de videojuegos. |
| Gestor de Soporte/Devoluciones | Maneja las incidencias post-venta, las solicitudes de reembolso, y autoriza las revocaciones o reactivaciones de licencias, incluyendo la conciliación manual de pagos fallidos (RN-04). |
| Analista de Negocio | Monitorea métricas comerciales, utiliza la información generada por el sistema para proponer estrategias de precios (RN-10) y optimización del flujo de ventas. |

#### 2.3.3 Descripción Textual de Casos de Uso del Negocio

Se seleccionan flujos críticos que demuestran la funcionalidad de e-commerce y la integración con las restricciones del entorno cubano. Se utilizan las plantillas detalladas del modelo.

Caso de Uso del Negocio

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso del Negocio** | **Procesar Compra Digital** | | |
| **Actores** | - Cliente (Iniciador)  - Pasarela de Pago (Participante) | | |
| **Resumen** | Este caso de uso se inicia cuando el Cliente confirma los productos en el carrito y selecciona la Pasarela de Pago Nacional (RN-02). El proceso gestiona la comunicación con la plataforma de pago (ej. EnZona), verifica la confirmación de la transacción, y culmina con la emisión automática de la Licencia Digital (RN-05). Es el caso de uso más crítico para la sostenibilidad económica del centro. | | |
| **Casos de Uso relacionados** | - Gestionar Catálogo (Inclusión)  - Gestionar Licenciamiento y Descarga Segura (Dependencia) | | |
| **Acción del actor** | | **Respuesta del proceso de negocio** | |
| 1. El Cliente selecciona los videojuegos y procede al *Checkout*.  3. El Cliente es redirigido a la interfaz de la Pasarela de Pago.  5. La Pasarela de Pago devuelve un *callback* (respuesta asíncrona) al sistema de la Tienda. | | 2. El sistema calcula el precio final (RN-07), registra el Pedido en estado "Pendiente" y genera el *link* de pago.  4. La Pasarela de Pago procesa la autenticación del Cliente y la deducción del monto.  6. El sistema valida el *callback*, cambia el estado del Pedido a "Completado" y a la Transacción a "Exitosa" (RF-26).  7. El sistema genera la Licencia Única (RN-06) y notifica al Cliente (RN-05). | |
| **Flujos alternativos** | | | |
| Fallo de Transacción (No Confirmación) | | | |
| **Acción del actor** | | | **Respuesta del proceso de negocio** |
| 1. La Pasarela de Pago devuelve un código de error o no responde al *callback* en el tiempo límite.  3. El sistema notifica al Gestor de Soporte y al Analista de Negocio sobre la transacción en estado indefinido. | | | 2. El sistema marca la Transacción como "Fallida/Pendiente de Conciliación" (RN-04).  4. El sistema mantiene el Pedido en "Pendiente", no genera la Licencia y notifica al Cliente. |

Caso de Uso del Negocio

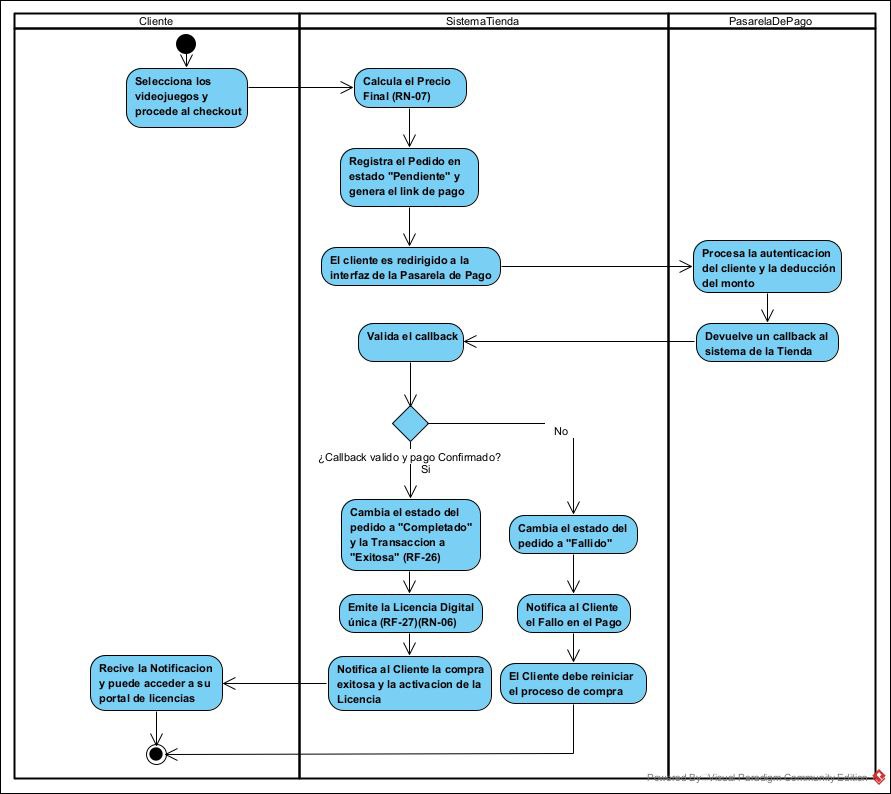
|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso del Negocio** | **Gestionar Licenciamiento y Descarga Segura** | | |
| **Actores** | - Cliente (Iniciador)  - Desarrollador (Participante)  - Administrador de Tienda (Supervisor) | | |
| **Resumen** | Este caso de uso asegura que, una vez pagado el producto, el Cliente reciba su derecho de uso (Licencia) y que el acceso al software sea trazable y restringido (RN-03, RN-06, RN-12). Incluye la gestión de nuevas versiones del software. Este control es crucial para la protección de la propiedad intelectual. | | |
| **Casos de Uso relacionados** | - Procesar Compra Digital (Dependencia) - Manejar Soporte y Devoluciones (Extensión: Revocación) | | |
| **Acción del actor** | | **Respuesta del proceso de negocio** | |
| 1. El Cliente accede a su portal de licencias activas.  3. El Cliente solicita la descarga de la Versión actual del videojuego.  5. El Desarrollador sube una nueva Versión del videojuego (parche).  7. El Cliente intenta una descarga repetida del software. | | 2. El sistema valida el estado de la Licencia (RN-03).  4. El sistema genera un enlace de descarga temporal y único, registrando el evento de Descarga (RN-12, RF-28).  6. El sistema vincula automáticamente la nueva Versión a todas las Licencias activas (RF-15).  8. El sistema monitorea el límite de Descargas. Si se supera (RN-12), se emite una alerta al Gestor de Soporte. | |
| **Flujos alternativos** | | | |
| Bloqueo por Límite Excedido | | | |
| **Acción del actor** | | | **Respuesta del proceso de negocio** |
| 1. El Cliente intenta descargar el software por sexta vez (límite hipotético excedido).  3. El Cliente contacta al soporte para la reactivación. | | | 2. El sistema suspende temporalmente la Licencia (RN-12) y notifica al Cliente.  4. El Gestor de Soporte revisa la traza de Descargas (IPs) y reactiva la Licencia si no hay indicio de uso indebido. |

Caso de Uso del Negocio

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso del Negocio** | **Manejar Soporte y Devoluciones** | | |
| **Actores** | - Cliente (Iniciador)  - Gestor de Soporte/Devoluciones (Ejecutor) | | |
| **Resumen** | Se activa cuando un Cliente reporta un problema (fallo de descarga, error de transacción o solicitud de reembolso). El proceso implica verificar la Licencia y Transacción, aplicar la política de devolución (RN-09) y, si se aprueba, revocar la Licencia y notificar para la gestión del *rollback* financiero. | | |
| **Casos de Uso relacionados** | - Procesar Compra Digital (Dependencia) - Gestionar Licenciamiento (Inclusión) | | |
| **Acción del actor** | | **Respuesta del proceso de negocio** | |
| 1. El Cliente envía una solicitud de soporte o devolución (RF-34).  3. El Gestor de Soporte revisa el Historial de Descargas y el estado de la Licencia (RN-09).  5. Si la devolución es aprobada, el Gestor de Soporte ejecuta la revocación. | | 2. El sistema crea un ticket de soporte, asociado al Cliente, la Licencia y la Transacción.  4. El sistema proporciona un dashboard con el estado completo de la transacción para el análisis del Gestor.  6. El sistema **revoca la Licencia** (RF-29), inhabilita el acceso a la descarga y notifica a la Pasarela de Pago para la gestión del reembolso (si el proceso lo permite). | |
| **Flujos alternativos** | | | |
| Problema Técnico (Fallo de Descarga) | | | |
| **Acción del actor** | | | **Respuesta del proceso de negocio** |
| 1. El Cliente reporta un fallo recurrente al iniciar la descarga.  3. El Gestor de Soporte puede restablecer el contador de descargas o proporcionar un nuevo token temporal. | | | 2. El sistema verifica el estado del archivo (Versión) y el límite de descargas del Cliente (RN-12).  4. El sistema genera un nuevo token de descarga con validez extendida. |

Diagrama de Actividades del Caso de Uso del Negocio

Los diagramas de actividades, siguiendo la notación UML, modelarán el flujo de trabajo del proceso de venta digital. El diagrama para el caso de uso Procesar Compra Digital debe mostrar las particiones de nado para el Cliente, el Sistema Tienda y la Pasarela de Pago, enfatizando el flujo de control, la decisión de redirección y el punto de sincronización donde el sistema espera el callback asíncrono de la Pasarela de Pago, marcando el punto de alta criticidad en la lógica del negocio.



### Conclusiones parciales

El desarrollo del presente capítulo, centrado en el Modelado del Contexto, permitió la aplicación sistemática de técnicas de la Ingeniería de Requisitos, logrando una comprensión profunda y una especificación formal del dominio del problema. La categorización de doce reglas del negocio transformó los lineamientos institucionales y las restricciones operativas de las pasarelas de pago nacionales en un conjunto de criterios accionables y verificables. Posteriormente, la elaboración del Modelo Conceptual facilitó la abstracción y organización de los conceptos fundamentales, estableciendo la diferencia crítica entre Videojuego, Versión, Licencia y Transacción. Finalmente, el Modelado del Negocio, mediante la identificación de Actores y Trabajadores y la descripción detallada de Casos de Uso críticos, delimitó con precisión los límites del sistema y documentó exhaustivamente las necesidades operativas de compra, licenciamiento y soporte. La descripción detallada de cada caso de uso constituye el insumo principal para la derivación formal de los requisitos funcionales y no funcionales del sistema.

## Capítulo III. Modelado del Sistema

### Introducción

El presente capítulo tiene como objetivo fundamental definir las capacidades y cualidades que regirán el desarrollo del sistema informático de la Tienda Online. A través de la aplicación de técnicas de la ingeniería de requisitos, se procede a la especificación detallada de los requisitos funcionales (RF), que definen el comportamiento específico del sistema, y los requisitos no funcionales (RnF), que establecen los criterios de calidad y las restricciones técnicas para su operación. Esta especificación constituye el contrato técnico esencial que guiará las fases subsiguientes de diseño, implementación y validación de la solución.

### 3.1 Requisitos funcionales

El listado de Requisitos Funcionales es una derivación directa y atómica de los Casos de Uso del Negocio y las Reglas del Negocio. Se identificaron un total de 14 Requisitos Funcionales (RF), que cubren desde la gestión básica de cuentas hasta la compleja gestión de activos digitales y transacciones financieras nacionales.

La función de **Generar Licencia Única (RF-08)** es fundamental, ya que operacionaliza la RN-06. Su correcta implementación es lo que diferencia la plataforma de una simple descarga de archivos, asegurando la trazabilidad y la protección intelectual. De igual manera, la **Conciliación Manual de Transacciones (RF-07)** es un requisito administrativo crucial, requerido por la RN-04, que permite a los Gestores de Soporte resolver transacciones que quedaron en estado indefinido tras la comunicación asíncrona con la Pasarela de Pago.

A continuación, se presentan los Requisitos Funcionales identificados, clasificados por áreas de funcionalidad crítica, con la complejidad y prioridad estimadas:

Requisitos Funcionales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Nombre** | **Descripción** | **Prioridad** | **Complejidad** |
| RF-01 | Gestionar Roles y Permisos | Asignar accesos basados en el perfil (RBAC) para Administradores, Gestores y Desarrolladores. | Alta | Media |
| RF-02 | Cargar Metadatos del Videojuego | Registro de descripción, requisitos, imágenes, categorías y centro desarrollador (RN-08). | Alta | Media |
| RF-03 | Cargar Versión de Software | Subir el archivo binario asociado a una Versión específica al almacenamiento seguro. | Media | Media |
| RF-04 | Vincular Licencia a Versión Actualizada | Asegurar que licencias activas puedan acceder a nuevas versiones si aplica. | Alta | Media |
| RF-05 | Calcular Subtotal y Total | Aplicar descuentos y comisiones, según la RN-07. | Alta | Baja |
| RF-06 | Procesar Pago vía Pasarela Nacional | Iniciar y completar la transacción con EnZona/Transfermóvil. (Crítica) | Crítica | Alta |
| RF-07 | Registrar Transacción Externa | Almacenar ID, estado y respuesta del *callback* de la pasarela. | Alta | Media |
| RF-08 | Generar Licencia Única | Creación criptográfica de la clave de uso y registro de su estado (RN-06). | Crítica | Alta |
| RF-09 | Habilitar Descarga Segura | Proveer enlace temporal para descarga con validación de Licencia (RN-05). | Alta | Alta |
| RF-10 | Revocar Licencia | Anular el derecho de uso por el Gestor de Soporte. | Media | Media |
| RF-11 | Registrar Solicitud de Devolución | Flujo de apertura y seguimiento de reembolsos (RN-09). | Media | Baja |
| RF-12 | Monitorear Límite de Descargas | Verificar que la Licencia no exceda los límites de uso (RN-12). | Alta | Media |
| RF-13 | Generar Reporte de Ventas por Producto | Reporte financiero básico y de distribución de ingresos. | Alta | Media |
| RF-14 | Monitorear Embudo de Compra | Trazabilidad de conversión (vistas > carrito > pago). | Media | Alta |

### 3.2 Requisitos no funcionales

Los requisitos no funcionales definen las cualidades de calidad y las restricciones técnicas bajo las cuales debe operar el sistema, clasificadas según la norma IEEE 29148.

La Seguridad (RnF-02) es crítica y debe incluir la protección del contenido descargable. Esto implica que el sistema debe prevenir el *hotlinking* (enlaces directos al binario) y asegurar que solo las Licencias activas y válidas puedan iniciar una descarga, mediante el uso de tokens seguros. Esta medida de seguridad físico-lógica es una implementación directa de la RN-03.

El Rendimiento (RnF-01) y la Usabilidad (RnF-03) deben priorizar la ligereza del sistema para mitigar los posibles problemas de conectividad que pueden existir en el contexto operativo. El diseño debe ser *responsive* y el tiempo de respuesta muy bajo, especialmente en el flujo de compra.

Requisitos No Funcionales (RnF) (IEEE 29148)

|  |  |  |
| --- | --- | --- |
| **No.** | **Categoría (IEEE 29148)** | **Descripción Detallada** |
| RnF-01 | Rendimiento | El tiempo de carga del Catálogo principal no debe exceder los 2 segundos para el 90% de las peticiones. Los procesos de generación de Licencia deben completarse en menos de 500 ms. |
| RnF-02 | Seguridad | Implementación de RBAC, uso obligatorio de SSL/TLS para todas las rutas y cifrado de datos sensibles. El sistema debe proteger el acceso al binario descargable mediante autenticación y tokens temporales. |
| RnF-03 | Usabilidad | Diseño *responsive* que garantice la accesibilidad y operatividad del flujo de compra completo en dispositivos móviles. El tiempo de aprendizaje para Gestores no debe superar los 30 minutos. |
| RnF-04 | Fiabilidad | El módulo de Transacción debe garantizar el registro correcto de los pagos, incluso en caso de fallo del *callback* asíncrono, con mecanismos de consulta de estado de respaldo. Disponibilidad operativa del 99.9%. |
| RnF-05 | Mantenibilidad | Arquitectura desacoplada (Django/Angular) que permita la actualización de dependencias o SDKs de pago sin modificar las capas de negocio esenciales. |
| RnF-06 | Escalabilidad | La arquitectura debe permitir un crecimiento del 100% en el volumen de Videojuegos y Clientes manteniendo los niveles de rendimiento especificados. |
| RnF-07 | Portabilidad | Compatibilidad con las últimas dos versiones estables de navegadores web y funcionamiento óptimo sobre la infraestructura de red de la UCI. |
| RnF-08 | Capacidad | Almacenamiento y gestión eficiente de hasta 100,000 registros históricos de Descargas y Transacciones. |

### 3.3 Documentación de requisitos

La documentación se realiza utilizando técnicas de ingeniería de requisitos para asegurar la trazabilidad y la comprensión precisa, siguiendo el modelo académico. Se aplica la plantilla detallada para el requisito de máxima complejidad, el Procesar Pago vía Pasarela Nacional (RF-06).

#### Aplicación de Técnica de Plantilla Detallada: Procesar Pago vía Pasarela Nacional (RF-06)

**Requisito: Procesar Pago vía Pasarela Nacional (RF-06)**

1. Información de Identificación y Trazabilidad:  
   Identificador: RF-06  
   Nombre: Procesar Pago vía Pasarela Nacional  
   Versión: 1.0  
   Estado: Aprobado  
   Fuente: Caso de Uso del Negocio "Procesar Compra Digital", Regla de Negocio RN-02  
   Prioridad: Crítica
2. Descripción Funcional Detallada:  
   Descripción: El sistema debe orquestar el flujo de pago electrónico iniciando una transacción con la pasarela nacional seleccionada (EnZona o Transfermóvil), esperando la confirmación de pago (callback) y gestionando el registro del estado.  
   Precondiciones:

* El Cliente ha finalizado el *checkout* y el Pedido está en estado "Pendiente".
* Las credenciales de la Pasarela de Pago están configuradas y activas.  
  Secuencia de Acciones:

1. El Cliente selecciona la Pasarela de Pago.
2. El sistema construye la solicitud de pago con el monto total (RN-07) y el ID de referencia.
3. El sistema redirige al Cliente a la URL de la Pasarela de Pago para autenticación y confirmación.
4. El sistema inicia la escucha del *callback* asíncrono.
5. Si el *callback* es "Éxito", el sistema procede a registrar la Transacción como Exitosa (RF-07).
6. Si el *callback* es "Fallo", el sistema notifica al Cliente y marca el Pedido como fallido.
7. Si el callback no se recibe en 60 minutos, el sistema marca la Transacción como "Pendiente de Conciliación" (RN-04).  
   Postcondiciones:

* Transacción registrada y vinculada al Pedido.
* Si exitosa: Se activa la generación de la Licencia Única (RF-08).  
  Excepciones:
* Fallo de Conexión con la Pasarela: El sistema registra un error de red y permite al Cliente reintentar la solicitud.
* Transacción Duplicada (RN-11): Si el ID de transacción ya existe, el sistema rechaza la operación y genera una alerta de fraude.

1. Criterios de Validación y Aceptación

* El sistema redirige correctamente al Cliente a la interfaz de EnZona/Transfermóvil.
* El sistema registra el ID de Transacción proporcionado por la Pasarela Externa (RF-07).
* Tras un *callback* de éxito, la Licencia se genera y el estado del Pedido es "Completado".
* Si no hay *callback* en el tiempo límite, el estado se marca como "Pendiente de Conciliación", activando una alerta interna.

**Historia de Usuario del requisito funcional RF-11: Registrar Solicitud de Devolución**

|  |  |  |
| --- | --- | --- |
|  | | |
| **Número:** 1 | Requisito: Registrar Solicitud de Devolución | |
| **Programador:**  Alvaro Alfonso Rodriguez | | **Iteración Asignada:** 1 |
| **Prioridad:** Media | | **Tiempo Estimado:**5 hora |
| **Riesgo en Desarrollo:** N/A | | **Tiempo Real:** 7 horas |
| **Descripción:**  El Cliente solo puede solicitar la devolución si han transcurrido menos de 72 horas desde la compra y la Licencia no ha sido utilizada para iniciar la descarga del software (RN-09). El sistema presenta un formulario que requiere seleccionar un motivo de devolución válido de una lista predefinida. | | |
| **Observaciones:**  Se muestra una ventana de devolución donde el motivo de la solicitud es obligatorio | | |
| **Prototipo elemental de interfaz gráfica de usuario** | | |

**Historia de Usuario del requisito funcional RF-05: Calcular Subtotal y total**

|  |  |  |
| --- | --- | --- |
|  | | |
| **Número:** 1 | Requisito: Calcular Subtotal y Total | |
| **Programador:**  Dainel Pérez Peñate | | **Iteración Asignada:** 2 |
| **Prioridad:** Alta | | **Tiempo Estimado:**3 hora |
| **Riesgo en Desarrollo:** Bajo | | **Tiempo Real:** 7 horas |
| **Descripción:**  El sistema debe computar automáticamente el costo acumulado de los videojuegos añadidos al carrito. Debe sumar los precios base, aplicar los descuentos vigentes y añadir las comisiones institucionales o de pasarela según la **RN-07**, mostrando el desglose claro al usuario antes del pago. | | |
| **Observaciones:**  El cálculo debe actualizarse en tiempo real cada vez que el usuario modifique la cantidad de productos en el carrito. | | |
| **Prototipo elemental de interfaz gráfica de usuario** | | |

### 3.4 Validación de requisitos

La validación es la fase esencial para asegurar que los requisitos especificados son **correctos, completos, consistentes y viables**. Para un sistema de e-commerce bajo restricciones específicas, la viabilidad técnica de la integración de pagos y la seguridad del licenciamiento son puntos de enfoque críticos.

La validación se llevó a cabo mediante la aplicación de técnicas complementarias:

1. **Listas de Chequeo y Consistencia Lógica:** Se verificó sistemáticamente que cada requisito fuese atómico, no ambiguo y consistente con las Reglas del Negocio (RN). Esta verificación se centró en la lógica de licenciamiento (RN-06) para evitar fallas en la emisión de activos digitales.
2. **Prototipado de Media Fidelidad:** Se crearon maquetas interactivas de los flujos de *Checkout* y *Descarga de Licencia*. La validación con usuarios potenciales verificó la **Usabilidad (RnF-03)**, garantizando que el diseño ligero y *responsive* cumpla con las necesidades del Cliente.
3. **Matriz de Trazabilidad:** Se documentó formalmente el mapeo que vincula cada RF a su origen (Caso de Uso de Negocio y Regla del Negocio). Esta técnica garantiza la **Completitud** del ERS, confirmando que todos los flujos de negocio definidos en el Capítulo II tienen un requisito de software asociado.
4. **Revisión Técnica Formal (RTF) y Verificación de Viabilidad:** Las sesiones de RTF se enfocaron en los requisitos de alta complejidad, especialmente la **Integración de Pagos (RF-25)**. Se confirmó con expertos en la materia que el flujo de transacción propuesto es técnicamente realizable con las APIs de EnZona/Transfermóvil disponibles. Se validó que el sistema permite la gestión de la Transacción Externa (RF-26) y la conciliación manual (RN-04) como pasos de mitigación esenciales, cumpliendo con la Fiabilidad (RnF-04) exigida para el contexto cubano.

La aplicación rigurosa de estas técnicas de validación asegura que el ERS es un documento sólido que minimiza los riesgos de ambigüedad durante la construcción del software.

La elección de la siguiente arquitectura física y lógica se fundamenta en los requisitos no funcionales de rendimiento, seguridad y mantenibilidad del sistema:

| **Capa** | **Nodos (Dispositivos)** | **Argumentación** |
| --- | --- | --- |
| **Capa de Presentación (Frontend)** | **Navegadores Web de Clientes** (Computadoras, Dispositivos Móviles) | Es la capa de interacción directa con el usuario. El sistema debe ser **accesible** desde cualquier dispositivo moderno, por lo que se utiliza una tecnología basada en **HTML5, CSS y JavaScript** (posiblemente un *framework* como Vue.js o React), permitiendo la **independencia** del sistema operativo del cliente. |
| **Capa de Lógica de Negocio (Backend)** | **Servidor de Aplicaciones (Middleware)** | Este nodo es el núcleo del sistema, donde reside la lógica de negocio (por ejemplo, la gestión de inventario de licencias, la validación de pagos y la autenticación de usuarios). Se justifica su uso para **centralizar el procesamiento**, manejar la seguridad de las peticiones (a través de **HTTPS**) y **desacoplar** la lógica de la presentación. Se recomienda un *framework* robusto como **Node.js con Express** o **Python con Django/Flask** para garantizar la eficiencia en el manejo de peticiones concurrentes. |
| **Capa de Datos** | **Servidor de Bases de Datos** | Este nodo está dedicado exclusivamente al almacenamiento persistente de la información crítica (catálogo de videojuegos, licencias, registros de usuarios, historial de transacciones). Se justifica mantenerlo **separado** del Servidor de Aplicaciones para aumentar la **seguridad** (no es accesible directamente desde el exterior) y el **rendimiento** (el servidor se enfoca solo en la gestión de las consultas). Se propone **PostgreSQL** o **MySQL** por su robustez, licencia abierta y capacidad de manejar datos transaccionales. |

### Conclusiones parciales

El desarrollo del presente capítulo permitió cumplir satisfactoriamente con el objetivo de especificar los requisitos del sistema de gestión de cargas académicas. Mediante la aplicación sistemática de técnicas de ingeniería de requisitos se definió un conjunto completo de 48 requisitos funcionales atómicos y 8 categorías de requisitos no funcionales, estableciendo así las bases técnicas fundamentales para el diseño detallado del sistema. La especificación se realizó mediante técnicas complementarias que garantizaron la calidad y completitud de los requisitos, tales como las historias de usuario y las plantillas estructuradas de requisitos detallados. Este abordaje integral aseguró que los requisitos definidos respondieran completamente a las necesidades del negocio y a las particularidades del modelo de comercialización digital de la UCI. Como resultado del capítulo, se dispone de una especificación técnica completa y validada que constituye el contrato formal para las fases subsiguientes de diseño e implementación.

## Capítulo IV. Modelado del Comportamiento e Interacción

### Introducción

El presente capítulo transforma los requisitos de software (RF y RnF) en un diseño arquitectónico y de detalle ejecutable. Esta fase es la transición esencial del "qué" al "cómo" se construirá el sistema, garantizando la modularidad y la mantenibilidad requeridas por los requisitos no funcionales (RnF-05 y RnF-06).

Introducción al Diseño de software. Conceptos asociados al diseño del software. Principios del modelado del diseño de software. Diseño de interfaces de usuario. Principios de diseño de interfaces. Proceso de Diseño de Interfaces (análisis de usuarios y contexto de uso, definición de requisitos de interacción, creación de prototipos)

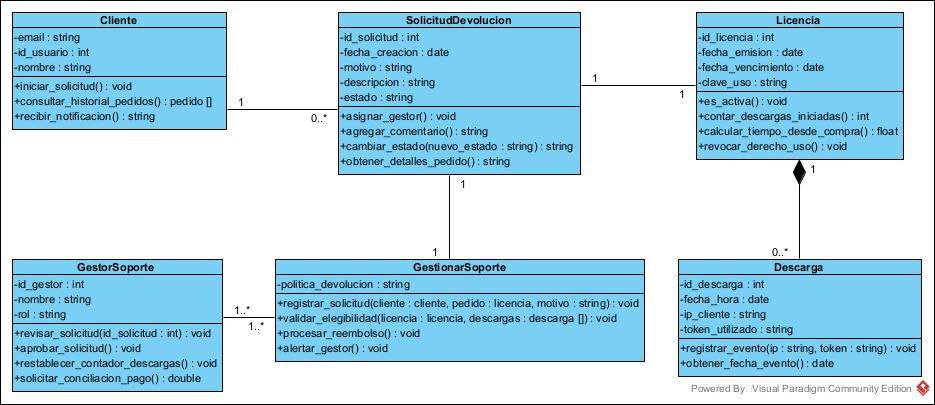
El diseño de software define la arquitectura, los componentes y las interfaces del sistema, priorizando la modularidad, la **cohesión** (responsabilidades relacionadas) y el **bajo acoplamiento** (mínima dependencia entre módulos) para la calidad. Los principios de modelado exigen crear el modelo más sencillo posible con un propósito explícito. El diseño de interfaces se rige por principios de **consistencia** y **concisión** para asegurar la usabilidad (RnF-03). El proceso de diseño de interfaces incluye el análisis de usuarios, la definición de requisitos de interacción y la creación de prototipos.

### 4.1 Diagrama de Paquetes

[Opcional]

### 4.2 Diagramas de clases

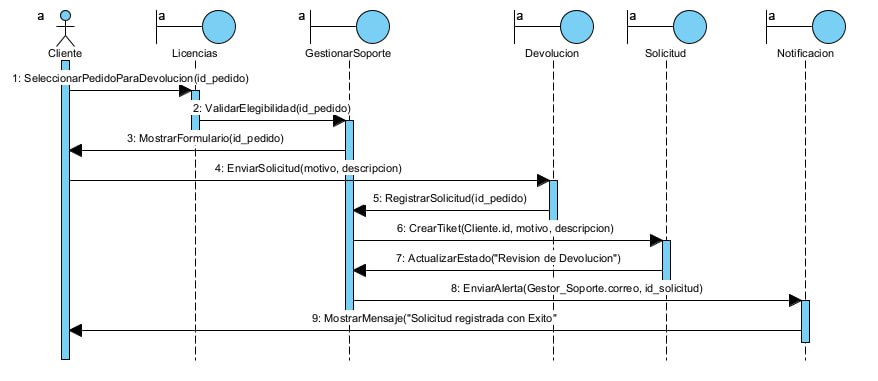
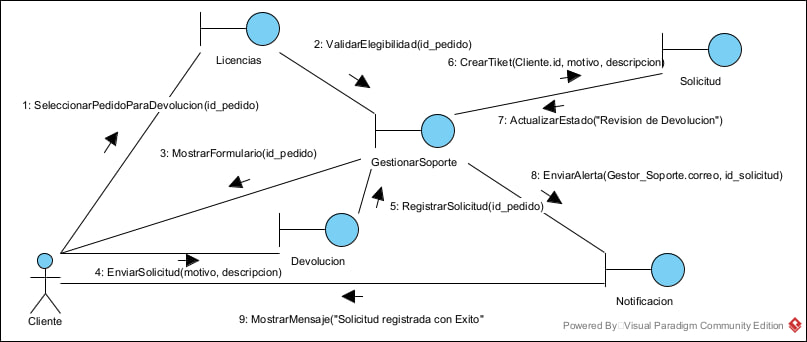
#### 4.2.1 Diagrama de clases del Caso de uso Manejar Soporte y Devoluciones.

**

|  |  |  |
| --- | --- | --- |
| **No.** | **Nombre de la Clase** | **Descripción** |
| 1 | Cliente | Representa al usuario comprador. Sus métodos están relacionados con la interacción del cliente con su portal. **Funcionalidad:** Iniciar y dar seguimiento a solicitudes, consultar historial de pedidos. |
| 2 | SolicitudDevolucion | Entidad central que registra el reclamo post-venta (RF-34). **Funcionalidad:** Almacena el estado, motivo y descripción del reclamo. Permite la asignación a un gestor y el registro de comentarios y cambios de estado durante el proceso de revisión. |
| 3 | Licencia | Clase de persistencia (Entidad) que contiene el derecho de uso del software (RN-06). **Funcionalidad:** Valida el estado de la licencia (Activa/Revocada), calcula el tiempo transcurrido desde la compra y expone métodos para revocar el derecho de uso (RF-29). Es experto en su estado de validez. |
| 4 | Descarga | Entidad que registra cada evento de acceso al binario por parte del Cliente. **Funcionalidad:** Registra la IP y la hora de acceso, y cuenta el total de descargas. Es esencial para verificar que no se haya iniciado la descarga, como exige la **RN-09** (Devoluciones Limitadas). |
| 5 | GestionarSoporte | Controlador del módulo de Soporte (Patrón Controlador). **Funcionalidad:** Orquesta el flujo de ticket, ejecuta la validación de elegibilidad (RF-31, aplicando RN-09), procesa la aprobación del reembolso, y coordina la notificación al Gestor de Soporte. |
| 6 | GestorSoporte | Representa al trabajador interno responsable de la gestión de incidencias. **Funcionalidad:** Contiene métodos para revisar solicitudes, aprobar o rechazar devoluciones, restablecer contadores de descarga (RN-12) y manejar la conciliación manual de pagos (RF-32). |

### 4.3 Diagramas de interacción

#### 4.3.1 Diagrama de secuencia/colaboración del Caso de uso Manejar Soporte y Devoluciones. Tienda Virtual.



### Conclusiones parciales

La elaboración del Capítulo IV ha permitido alcanzar el objetivo de diseñar la estructura lógica del sistema informático, asegurando la trazabilidad desde los requisitos hasta el diseño detallado. Se definieron los modelos estáticos y dinámicos del sistema, incluyendo la arquitectura modularizada por paquetes, lo que garantiza el cumplimiento de los principios de diseño de bajo acoplamiento y alta cohesión.

La aplicación rigurosa de estos principios se verificó en las clases para los flujos críticos, como Procesar Compra Digital y Manejar Soporte y Devoluciones. Este diseño garantiza que la lógica implemente directamente las Reglas de Negocio sensibles (RN-06: Clave Única; RN-09: Devoluciones Limitadas) y cumpla con la Mantenibilidad (RnF-05) y Escalabilidad (RnF-06) requeridas para la plataforma de la UCI.

## Bibliografía

Adom, D. (2025). (PDF) Review: Academic Stress among Faculty and Students in Higher Institutions. *ResearchGate*.

Alvarez, B. M. G., & Cárdenas, J. A. Q. (2010). *ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO PARA EL APOYO AL PROCESO DE ASIGNACIÓN DE LA CARGA ACADÉMICA USANDO PROGRAMACIÓN CON RESTRICCIONES*. *2010*.

Arroyo, V. J., Ávila, B. M., & Baltazar, M. I. H. (2023). Revisión narrativa de carga académica y calidad de vida en estudiantes universitarios. *Janaskakua*, *5*(11).

Bernal Peralta, J. (2021). *La carga académica de estudiantes de una Facultad de Administración y Economía (Chile). Un aporte a la gestión de docencia universitaria | Iberoamerican Business Journal*.

Gómez, M. C. S., & Muñoz-Repiso, A. G.-V. (2002). Formación y profesionalización docente del profesorado universitario. *Revista de Investigación Educativa*, *20*(1), 153-171.

Pérez Pino, M. T., Ciudad Ricardo, F. Á., Puentes Puentes, Ú., Estrada Molina, O., Piñero Gómez, Y., & López Velázquez, A. (2015). Indicadores para la evaluación de la calidad de la formación del ingeniero en Ciencias Informáticas. *Revista Cubana de Ciencias Informáticas*, *9*(4), 171-186.

*Por un Modelo de formación Humanista | Universidad de las Ciencias Informáticas*. (2012, mayo 7).

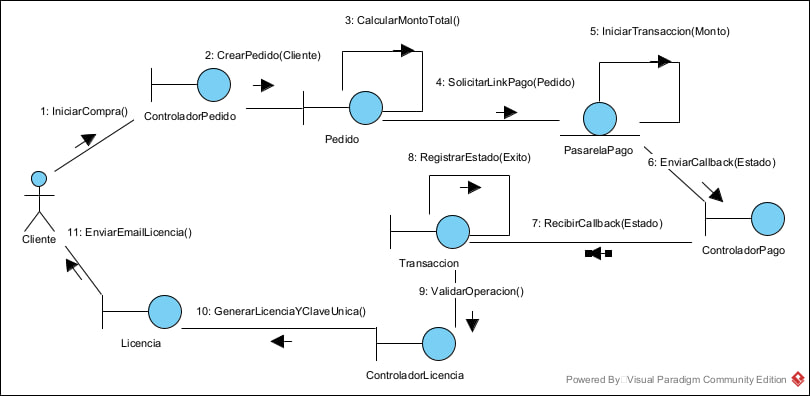
*Universidad de las Ciencias Informáticas*. (2025, septiembre 25). https://www.uci.cu/

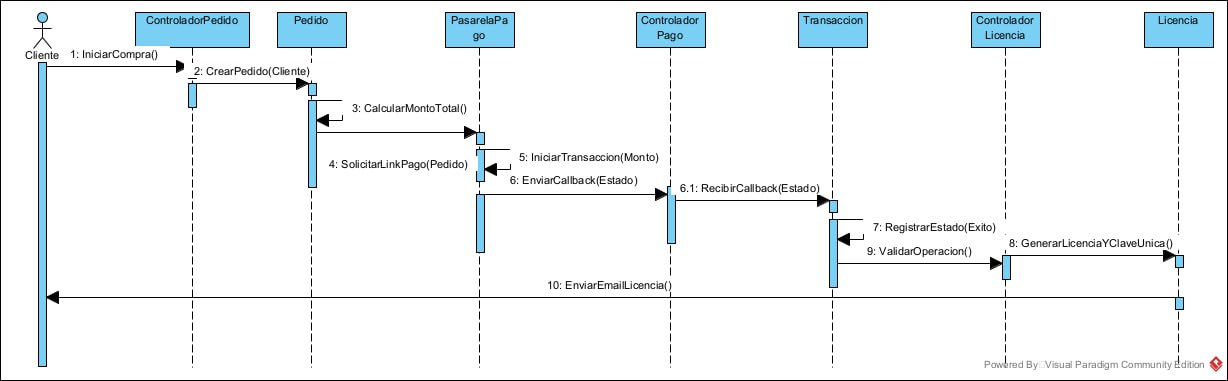
Valenzuela García, R. (2016). *Resolución No. 66/14*

Valle, P. E. R., Aguilar, L. C., & Herrera, J. F. L. (2025). Formación y desarrollo de la fuerza de trabajo calificada de nivel medio y superior desde el enfoque Ciencia, Tecnología y Sociedad más Innovación. *Pedagogía*

## Anexos

**Diagrama de secuencia/colaboración del Caso de uso Procesar Compra Digital**





**Diagrama de secuencia/colaboración del Caso de Uso Gestionar Licenciamiento y Descarga Segura**

