Enseñanza del desarrollo e-commerce centrado en el usuario mediante un proyecto basado en Go: caso de estudio exploratorio en educación en ingeniería

Teaching user-centered e-commerce development through a Go-based project: an exploratory case study in engineering education

Edgar Manuel Cano Cruz ✉
Departamento de Ingeniería en Desarrollo de Software, Universidad del Istmo, Oaxaca, México
edgarcano@unistmo.edu.mx
ORCID: 0000-0002-3394-1902

https://doi.org/10.36825/RITI.15.36.002

Recibido: Julio 08, 2026
Aceptado: Noviembre 16, 2026

Resumen: Los cursos de ingeniería de software suelen abordar de manera independiente la implementación técnica y la evaluación de la experiencia de usuario, lo que limita la integración de ambas dimensiones durante el proceso de aprendizaje. Ante esta problemática, el presente artículo documenta un caso de estudio educativo exploratorio desarrollado en un curso de Interacción Humano-Computadora, utilizando como eje central el proyecto Go-Plant, una plataforma de comercio electrónico implementada con Go. La experiencia educativa integró el Aprendizaje Basado en Proyectos (ABP), el modelo ADDIE como estructura de diseño instruccional y el Diseño Centrado en el Usuario (DCU). En el proyecto participaron 30 estudiantes organizados en seis equipos y diez usuarios externos que realizaron pruebas formativas de usabilidad. La evaluación inicial identificó dificultades relacionadas con la comprensión de los métodos de pago (9/10 usuarios), el filtrado de productos (5/10) y la localización de información relevante (5/10). Estos hallazgos orientaron el rediseño de la navegación, el catálogo, el carrito y el proceso de compra. Posteriormente, los mismos diez usuarios participaron en una segunda evaluación de aceptación visual, en la que la puntuación media aumentó de 2.54 a 4.28 puntos en una escala de cinco niveles. La prueba de rangos con signo de Wilcoxon para muestras relacionadas mostró una diferencia estadísticamente significativa entre ambas versiones (W=0.00; p=0.001953, bilateral exacta). Los resultados evidencian una mayor aceptación visual de la versión rediseñada y muestran cómo la integración de ABP, ADDIE y DCU permite vincular el desarrollo técnico con la evaluación formativa y la mejora iterativa de interfaces. El estudio ofrece una referencia metodológica para incorporar prácticas de desarrollo centrado en el usuario en la formación en ingeniería, sin establecer efectos causales sobre el aprendizaje estudiantil.

Palabras clave: *Educación en Ingeniería, Aprendizaje Basado en Proyectos, ADDIE, Diseño Centrado en el Usuario, Pruebas de Usabilidad.*

Abstract: Software engineering courses often address technical implementation and user experience evaluation independently, limiting the integration of these dimensions throughout the learning process. To address this challenge, this article presents an exploratory educational case study conducted in a Human-Computer Interaction course, centered on Go-Plant, an e-commerce platform developed using Go. The educational experience integrated Project-Based Learning (PBL), the ADDIE model as an instructional design framework, and User-Centered

© RITI. CC BY-NC

Design (UCD). The project involved 30 students organized into six teams and ten external users who participated in formative usability testing. The initial evaluation identified difficulties in understanding payment methods (9/10 users), filtering products (5/10), and locating relevant information (5/10). These findings guided the redesign of navigation, the product catalog, shopping cart, and checkout process. Subsequently, the same ten users participated in a second visual acceptance evaluation, in which the mean score increased from 2.54 to 4.28 points on a five-point scale. The Wilcoxon signed-rank test for paired samples revealed a statistically significant difference between the two versions (W=0.00; exact two-sided p=0.001953). The findings indicate greater visual acceptance of the redesigned interface and illustrate how integrating PBL, ADDIE, and UCD can connect technical development with formative evaluation and iterative interface improvement. This study provides a methodological reference for incorporating user-centered development practices into engineering education without establishing causal effects on student learning.

Keywords: *Engineering Education, Project-Based Learning, ADDIE, User-Centered Design, Usability Testing.*

1. Introducción

El desarrollo de plataformas web modernas exige que los ingenieros integren competencias que a menudo se enseñan por separado: análisis de requisitos, arquitectura de software, diseño de bases de datos, programación segura, diseño de interfaces, evaluación de usabilidad y mejora iterativa. Esta integración es especialmente importante en sistemas de comercio electrónico, donde el rendimiento técnico y la experiencia de usuario influyen directamente en la navegación, la confianza, la toma de decisiones y la finalización de compras. Para la educación en ingeniería, esto plantea un reto: los estudiantes deben aprender no solo a implementar software funcional, sino también a evaluar cómo los usuarios reales comprenden y utilizan los sistemas que construyen [1], [2], [3].

Los enfoques tradicionales de enseñanza del desarrollo de software suelen enfatizar la sintaxis de programación, módulos aislados o ejercicios predefinidos. Aunque estas actividades son útiles para adquirir habilidades fundamentales, no siempre exponen a los estudiantes a la complejidad de los proyectos auténticos, donde las decisiones técnicas interactúan con necesidades de usuario, restricciones de seguridad, lógica de negocio, accesibilidad y usabilidad de la interfaz. El Aprendizaje Basado en Proyectos (ABP) ofrece una alternativa pertinente porque permite trabajar con problemas realistas, desarrollar artefactos, colaborar en equipos y evaluar soluciones en contextos cercanos a la práctica profesional [4], [5], [6].

Este artículo presenta Go-Plant, una plataforma web de comercio electrónico para la venta de plantas en línea, como proyecto instruccional para enseñar desarrollo web centrado en el usuario. La plataforma se implementó con el lenguaje Go para el *backend*, MySQL para la gestión de datos y HTML, CSS y JavaScript para el *frontend*. A diferencia de tareas convencionales de desarrollo web que concluyen cuando el sistema funciona técnicamente, la experiencia Go-Plant exigió conectar la implementación con investigación de usuarios, prototipado de interfaces, pruebas de usabilidad y rediseño basado en evidencia.

El manuscrito se plantea como un caso de estudio educativo exploratorio. Esta delimitación es importante porque la implementación actual aporta evidencia sólida mediante artefactos producidos por estudiantes y pruebas de usabilidad con usuarios externos, pero no permite formular afirmaciones causales sobre ganancias de aprendizaje. Por ello, el artículo separa tres tipos de evidencia: el modelo instruccional utilizado para organizar la experiencia, los artefactos del proyecto generados por los estudiantes y la evidencia de usabilidad obtenida con usuarios externos. Además, se propone un marco de evaluación por competencias para fortalecer la medición del aprendizaje en iteraciones futuras.

El estudio realiza cuatro contribuciones. Primero, presenta un diseño instruccional replicable que integra ABP, ADDIE y Diseño Centrado en el Usuario (DCU) en un curso de ingeniería de software. Segundo, muestra cómo una plataforma *e-commerce* basada en Go puede funcionar como artefacto auténtico de aprendizaje para conectar desarrollo *backend*, diseño de interfaz, seguridad y evaluación de usabilidad. Tercero, aporta evidencia formativa de usabilidad que muestra cómo la retroalimentación de usuarios externos puede traducirse en decisiones concretas de rediseño. Cuarto, propone un marco de evaluación del aprendizaje estudiantil que puede aplicarse en futuras cohortes para medir con mayor rigor competencias técnicas, de UX, colaborativas y reflexivas. La Figura 1 presenta el marco instruccional que integra ABP, ADDIE y DCU en la experiencia Go-Plant.

© RITI. CC BY-NC

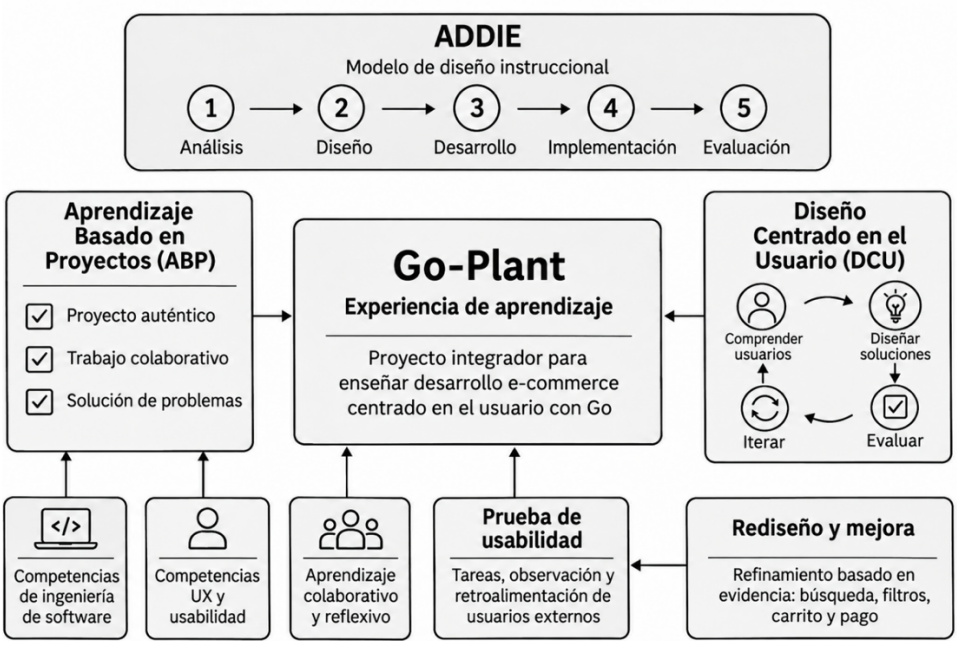


Figura 1. Marco instruccional de la experiencia de aprendizaje Go-Plant.

2. Antecedentes y trabajos relacionados

2.1. Aprendizaje Basado en Proyectos en educación en ingeniería de software

El ABP se ha utilizado ampliamente en educación en ingeniería para promover aprendizaje activo, colaboración, solución de problemas y transferencia de conocimiento. En ingeniería de software es particularmente relevante porque la práctica profesional exige gestionar requisitos, arquitectura, implementación, pruebas, documentación y retroalimentación de interesados. Cuando los estudiantes trabajan en proyectos realistas, pueden comprender mejor la relación entre conceptos teóricos e implementación práctica [4], [5], [6].

En cursos de desarrollo web, el ABP ayuda a superar ejercicios aislados de programación y favorece el desarrollo de sistemas completos compuestos por módulos interactuantes. Estos proyectos exponen a los estudiantes a servicios *backend*, interacción *frontend*, bases de datos, seguridad, decisiones de despliegue y atributos de calidad. No obstante, muchos proyectos educativos siguen enfatizando la terminación técnica más que la evaluación de experiencia de usuario. Como resultado, los estudiantes pueden construir sistemas funcionales, pero no necesariamente fáciles, confiables o eficientes para los usuarios [5], [7], [8].

2.2. Experiencia de usuario como competencia de ingeniería de software

La experiencia de usuario constituye una dimensión crítica de la calidad del software moderno. En plataformas de comercio electrónico, la UX afecta navegación, descubrimiento de productos, confianza, interacción con el carrito, comprensión del pago y confirmación de compra. Para estudiantes de ingeniería, aprender UX no se reduce al diseño visual; implica comprender metas de usuario, observar interacciones, identificar fricciones y transformar evidencia en mejoras del sistema [3], [8], [9], [10], [11], [12].

Integrar UX en la educación en ingeniería de software puede ayudar a los estudiantes a construir una comprensión más completa de la calidad del sistema. Una funcionalidad puede estar correctamente implementada desde la perspectiva de programación y, aun así, fallar si los usuarios no pueden localizarla, interpretarla o confiar en ella. Por ello, la evaluación de usabilidad actúa como puente entre el desarrollo técnico y la efectividad real del sistema [8], [10], [13].

El esquema de la Figura 1 sintetiza tres funciones distintas: ABP define el reto auténtico y el trabajo colaborativo; ADDIE articula el diseño educativo en cinco fases; y DCU introduce la comprensión de usuarios, prototipado, evaluación y mejora. Las correspondencias concretas entre fases y entregables se detallan en la Sección 3.3 y en la Tabla 2.

© RITI. CC BY-NC

2.3. Desarrollo web con Go en educación en ingeniería

El lenguaje Go se utiliza para construir aplicaciones web eficientes, escalables y concurrentes. Su simplicidad, rendimiento y soporte para desarrollo del lado del servidor lo hacen adecuado para enseñar conceptos de ingeniería *backend*, como enrutamiento, manejo de peticiones, acceso a bases de datos, autenticación, sesiones, diseño modular y organización de servicios [14], [15].

Go es menos común en cursos introductorios de desarrollo web que lenguajes y marcos como JavaScript, Python, PHP o Java. Esto abre una oportunidad para diseñar experiencias educativas que expongan a los estudiantes a tecnologías *backend* alternativas, enfatizando arquitectura, mantenibilidad e integración. En Go-Plant, Go no se trató solamente como lenguaje de programación, sino como vehículo para enseñar cómo las decisiones *backend* sostienen servicios digitales centrados en el usuario.

2.4. Plataformas e-commerce como ambientes auténticos de aprendizaje

Los sistemas *e-commerce* son artefactos educativos ricos porque integran múltiples preocupaciones de ingeniería de software en un contexto familiar para los usuarios. Una plataforma típica incluye catálogos de productos, cuentas de usuario, autenticación, carritos de compra, flujo de pago, confirmación de pedidos, administración y diseños responsivos. Estos componentes hacen que los proyectos *e-commerce* sean adecuados para enseñar competencias técnicas y humanocentradas [11], [12].

A diferencia de tareas abstractas de programación, las plataformas *e-commerce* permiten evaluar si los usuarios pueden realizar tareas significativas: explorar, seleccionar, comparar, añadir productos al carrito, modificar cantidades, avanzar al pago y confirmar un pedido. Cada acción ofrece evidencia observable de usabilidad y efectividad del sistema, por lo que este tipo de proyecto resulta adecuado para combinar ABP, ingeniería de software y evaluación UX.

2.5. Modelo ADDIE como estructura del diseño instruccional

El modelo ADDIE proporciona una estructura sistemática para el diseño instruccional mediante cinco fases: Análisis, Diseño, Desarrollo, Implementación y Evaluación. Su aplicación permite articular necesidades de aprendizaje, objetivos educativos, actividades formativas y mecanismos de evaluación, con flexibilidad para adaptarse a diferentes contextos educativos [16].

En Go-Plant, ADDIE se seleccionó para organizar progresivamente las actividades y los entregables de un proyecto de comercio electrónico. Su función fue estructurar la intervención educativa, mientras que el Aprendizaje Basado en Proyectos (ABP) proporcionó el contexto de trabajo auténtico y colaborativo, y el Diseño Centrado en el Usuario (DCU) orientó el análisis de necesidades, prototipado, evaluación de usabilidad y rediseño [4], [9], [10]. La aplicación conjunta de estos enfoques se describe en la Sección 3.3.

3. Intervención instruccional y contexto educativo

3.1. Contexto del curso

El proyecto Go-Plant se implementó en un curso universitario de Interacción Humano-Computadora de un programa de ingeniería orientado al desarrollo de software y sistemas inteligentes. La experiencia involucró a 30 estudiantes de licenciatura organizados en seis equipos de cinco integrantes. Esta estructura permitió distribuir responsabilidades vinculadas con análisis de requisitos, diseño de interfaces, desarrollo *backend*, gestión de base de datos, pruebas de usabilidad, documentación y presentación final.

El proyecto se desarrolló durante un semestre académico como actividad integradora del curso. Go-Plant fue concebido como un reto auténtico de ingeniería de software: diseñar, implementar y evaluar una plataforma *e-commerce* que equilibrara funcionalidad técnica y experiencia de usuario. La plataforma funcionó como artefacto central mediante el cual los estudiantes conectaron ingeniería de requisitos, modelado UML, desarrollo *backend*, diseño *frontend*, arquitectura de base de datos, ciberseguridad, diseño responsivo y pruebas de usabilidad. La Figura 2 muestra el cronograma de trabajo establecido para el desarrollo y la evaluación del proyecto durante el semestre.

© RITI. CC BY-NC

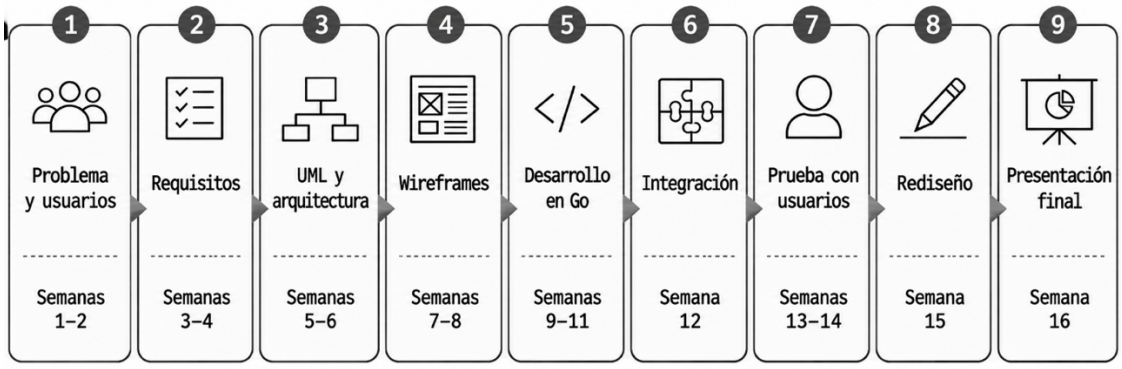


Figura 2. Cronograma de implementación del proyecto en el curso.

3.2. Objetivos de aprendizaje y evidencia esperada

La experiencia instruccional se organizó alrededor de nueve objetivos de aprendizaje: analizar requisitos funcionales y no funcionales; modelar el comportamiento del sistema mediante UML; diseñar una arquitectura web modular; implementar funcionalidad *backend* con Go y MySQL; diseñar interfaces responsivas; aplicar mecanismos básicos de seguridad; realizar pruebas de usabilidad con usuarios externos; rediseñar componentes de interfaz a partir de retroalimentación empírica; y reflexionar sobre la relación entre calidad del software, UX y toma de decisiones en ingeniería.

Para hacer observables estos objetivos, cada competencia se asoció con un artefacto concreto del proyecto. Esta alineación guio la instrucción, la retroalimentación y la documentación durante el semestre. La Tabla 1 relaciona las competencias previstas con los entregables y su propósito educativo.

Tabla 1. Alineación entre competencias instruccionales y evidencia del proyecto.

Competencia	Evidencia del proyecto	Propósito educativo
Ingeniería de requisitos	Requisitos funcionales y no funcionales, casos de uso, criterios de validación	Traducir necesidades de usuario y del sistema en especificaciones explícitas
Modelado del sistema	Diagramas de requisitos, casos de uso, flujos y secuencia	Conectar diseño conceptual con comportamiento esperado del sistema
Arquitectura de software	Estructura por capas basada en rutas, controladores, modelos y configuración	Organizar responsabilidades y dependencias en un sistema mantenible
Desarrollo *backend*	Módulos en Go para productos, inicio de sesión, carrito, direcciones y pago	Practicar implementación modular del lado del servidor
Diseño de base de datos	Modelo entidad-relación y funciones de consulta a base de datos	Sostener operaciones persistentes para usuarios, productos, carritos y pedidos
Seguridad	Autenticación, acceso restringido, protección CSRF, validación de *tokens*	Relacionar mecanismos de seguridad con confianza y operación segura
Diseño UX	*Wireframes* de baja y alta fidelidad, interfaz responsiva, navegación	Planificar interacción antes y durante la implementación
Evaluación de usabilidad	Protocolo de tareas, observaciones, frecuencias cuantitativas, comentarios cualitativos	Identificar fricciones mediante evidencia de usuarios externos
Mejora iterativa	Rediseño de filtros, búsqueda, tarjetas, carrito, flujo de pago y confirmación	Transformar retroalimentación empírica en mejora del sistema

3.3. Integración de ABP, ADDIE y DCU

La intervención integró ABP, ADDIE y DCU para articular el desarrollo técnico con la evaluación centrada en el usuario [4], [6], [9], [16], [17]. El ABP situó el aprendizaje en un proyecto auténtico y colaborativo; ADDIE organizó la secuencia instruccional; y el DCU orientó las actividades de experiencia de usuario. Durante la fase de Análisis se identificaron necesidades y requisitos mediante el pretest de usuarios; en Diseño se elaboraron modelos UML y *wireframes*; en Desarrollo se implementaron los módulos de Go-Plant; en Implementación se

© RITI. CC BY-NC

integró la plataforma y se realizaron pruebas con usuarios externos; y en Evaluación se analizaron los hallazgos de usabilidad y se justificaron las decisiones de rediseño. La Figura 3 y la Tabla 2 muestran la correspondencia entre las fases, las actividades y los entregables del proyecto.

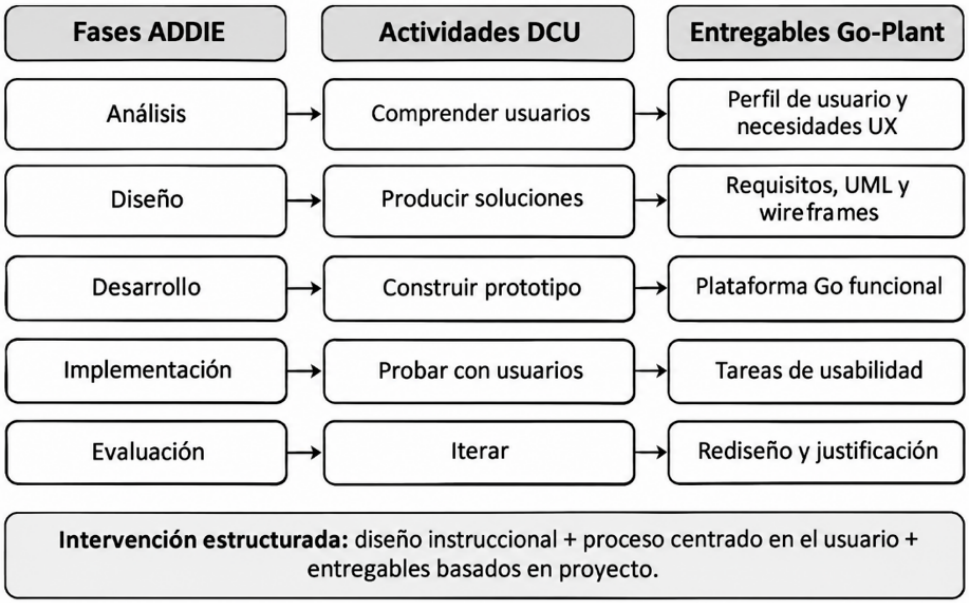


Figura 3. Integración del modelo ADDIE con actividades de DCU y entregables generados.

Tabla 2. Operacionalización de ADDIE en la experiencia Go-Plant.

Fase ADDIE	Propósito educativo	Actividad Go-Plant	Entregable estudiantil
Análisis	Identificar el problema, las características de los estudiantes, los usuarios y las necesidades de aprendizaje	Analizar problemas UX de *e-commerce*, expectativas de usuarios y requisitos técnicos	Planteamiento del problema, perfil de usuario, requisitos iniciales
Diseño	Planificar la solución instruccional y técnica	Definir requisitos del sistema, diagramas UML, arquitectura y prototipos de interfaz	Requisitos, diagramas UML, *wireframes*
Desarrollo	Construir el artefacto de aprendizaje y sus componentes	Implementar módulos *backend*, acceso a datos, plantillas frontend, carrito, login, productos, direcciones y pago	Prototipo funcional y documentación técnica
Implementación	Usar el prototipo en tareas realistas	Integrar módulos y aplicar tareas de usabilidad con usuarios externos	Plataforma integrada y protocolo de prueba
Evaluación	Valorar usabilidad del producto y reflexionar sobre evidencia del proceso de aprendizaje mediante artefactos	Analizar observaciones, identificar fricciones y rediseñar la interfaz	Resultados de usabilidad, justificación de rediseño y reflexión

El DCU guio el componente de experiencia de usuario. Los estudiantes analizaron primero la experiencia previa de usuarios con compras en línea y sus expectativas sobre entornos digitales de compra. Posteriormente especificaron requisitos relacionados con UX, crearon prototipos de baja y alta fidelidad, evaluaron la plataforma con usuarios y rediseñaron la interfaz con base en los resultados [8], [9], [10]. La Tabla 3 detalla las actividades y evidencias del proceso de DCU implementado en Go-Plant, mientras que la Figura 4 representa gráficamente su ciclo iterativo.

© RITI. CC BY-NC

Tabla 3. Actividades de Diseño Centrado en el Usuario implementadas en Go-Plant.

Actividad DCU	Implementación en Go-Plant	Evidencia generada
Comprender usuarios y contexto	Pretest sobre experiencia de compra en línea, familiaridad tecnológica y expectativas	Perfil de usuario y cuestionario Likert
Especificar requisitos de usuario y sistema	Definir requisitos funcionales, no funcionales y relacionados con UX	Tablas de requisitos y casos de uso
Producir soluciones de diseño	Crear *wireframes*, pantallas de alta fidelidad, diseños responsivos y flujos de interacción	Prototipos de baja y alta fidelidad
Evaluar con usuarios	Realizar tareas de navegación, descubrimiento de productos, carrito, pago y retroalimentación	Observaciones, frecuencias cuantitativas y comentarios cualitativos
Iterar y rediseñar	Mejorar búsqueda, filtros, tarjetas de producto, carrito, flujo de pago, *chatbot* y confirmación	Pantallas rediseñadas y justificación de diseño

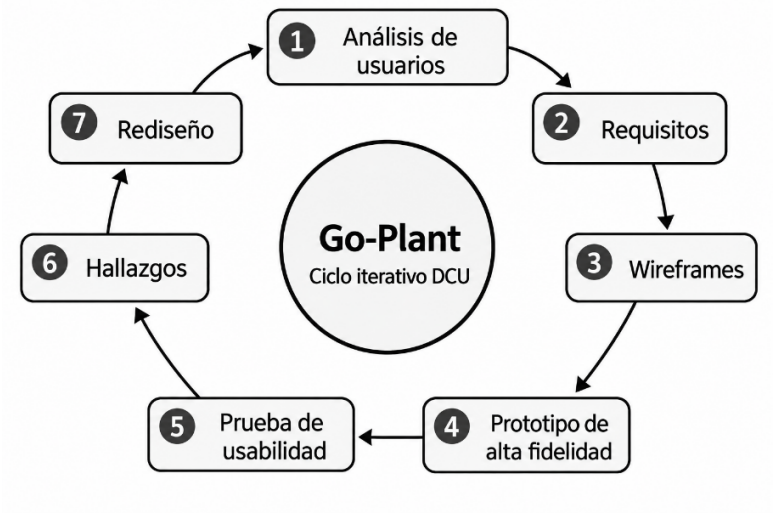


Figura 4. Ciclo de DCU aplicado a Go-Plant.

4. Plataforma Go-Plant como artefacto de aprendizaje

4.1. Alcance funcional

Go-Plant fue diseñada como una plataforma *e-commerce* responsiva para la venta de plantas en línea. El sistema incluyó dos roles principales: clientes y administradores. Los clientes podían navegar por el catálogo, consultar información detallada de productos, registrarse, iniciar sesión, agregar productos al carrito, modificar cantidades, gestionar direcciones de envío, avanzar por el flujo de pago y recibir confirmación de compra. Los administradores podían gestionar usuarios, productos y categorías.

- Exploración y filtrado del catálogo de productos.
- Visualización de detalles de producto.
- Registro e inicio de sesión de usuarios.
- Gestión del carrito de compras.
- Gestión de direcciones de envío.
- Flujo de pago seguro y simulado.
- Confirmación de pedido.
- Administración de productos y categorías.
- Interfaz responsiva.

© RITI. CC BY-NC

4.2. Requisitos no funcionales

La plataforma se diseñó para ser accesible desde navegadores web modernos, intuitiva para usuarios y administradores, segura en el manejo de operaciones sensibles, responsiva en distintos dispositivos y eficiente en la navegación y gestión del catálogo. Los requisitos de seguridad incluyeron acceso restringido, autenticación, protección CSRF y validación basada en *tokens* [1], [2].

Desde la perspectiva educativa, estos requisitos no funcionales fueron importantes porque permitieron a los estudiantes comprender que la calidad del sistema no se limita al cumplimiento de funcionalidades. Rendimiento, seguridad, usabilidad, mantenibilidad y responsividad se trataron como preocupaciones de ingeniería que afectan tanto la confiabilidad técnica como la satisfacción del usuario.

4.3. Arquitectura

La arquitectura se organizó en rutas, controladores, componentes de modelo/acceso a datos y módulos de configuración. Las rutas definieron puntos de acceso para productos, login, carrito, direcciones y acciones de pago. Los controladores gestionaron la lógica de negocio y procesaron solicitudes. Los componentes de modelo se encargaron de operaciones de base de datos, incluyendo recuperación de productos, actualización del carrito, autenticación de usuarios y administración de direcciones. Los módulos de configuración apoyaron sesiones, mensajes flash y parámetros del sistema [1], [14], [15].

Esta arquitectura modular permitió conectar diseño conceptual con implementación. Por ejemplo, el módulo de productos requirió una ruta para navegación, un controlador para filtrar y organizar datos, una función de modelo para recuperar información de la base de datos y una plantilla para renderizar la interfaz. De modo similar, el registro de usuarios integró validación de formularios, manejo JSON, lógica de controlador, *hashing* de contraseñas, gestión transaccional y creación de sesión. La Figura 5 representa la arquitectura modular implementada en la plataforma Go-Plant.

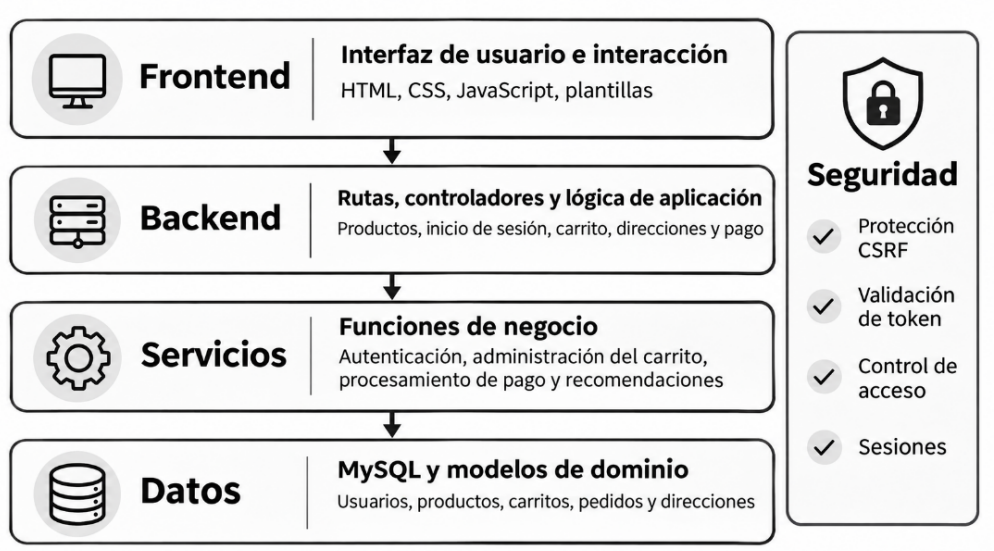


Figura 5. Arquitectura simplificada de la plataforma Go-Plant.

4.4. Diseño de interfaz

El proceso de diseño de interfaz incluyó *wireframes* de baja fidelidad y prototipos de alta fidelidad. Los *wireframes* iniciales representaron página de inicio, catálogo, página de detalle de producto, carrito, pasarela de pago y paneles administrativos. Estos artefactos ayudaron a planificar estructura y navegación antes de la implementación [8], [9], [10], [11].

Los diseños de alta fidelidad refinaron la presentación visual mediante distribución, tipografía, colores, iconografía, botones, tarjetas de producto, filtros y comportamiento responsivo. El catálogo se rediseñó con filtros interactivos y presentación más clara de productos. El carrito se mejoró con controles más accesibles, modificación

© RITI. CC BY-NC

de cantidades, opciones de eliminación y resumen de compra. La sección de pago incorporó opciones más visibles, validación visual y un flujo más guiado.

5. Método de evaluación

La evaluación empírica de Go-Plant se centró en las pruebas formativas de usabilidad y aceptación visual realizadas con diez usuarios externos. De manera complementaria, se presenta un marco de evaluación del aprendizaje estudiantil para futuras implementaciones, sin que forme parte del análisis empírico de este estudio.

5.1. Marco de evaluación del aprendizaje estudiantil

Para fortalecer la evaluación educativa de Go-Plant, se diseñó un marco de evaluación por competencias alineado con los enfoques ABP, ADDIE y DCU. Este marco contempla cuatro dimensiones: competencias técnicas de ingeniería de software, experiencia de usuario (UX), colaboración y aprendizaje reflexivo. Cada dimensión se vincula con criterios observables y evidencias generadas durante el proyecto, como requisitos, diagramas UML, módulos en Go, prototipos, registros de usabilidad y reportes reflexivos [1], [4], [9], [17].

La Tabla 4 presenta estos instrumentos como herramientas complementarias para fortalecer la evaluación del aprendizaje en futuras implementaciones de Go-Plant. Su aplicación permitiría valorar sistemáticamente las competencias técnicas, de UX, colaborativas y reflexivas desarrolladas durante la experiencia educativa, previa revisión y validación de los instrumentos.

Tabla 4. Marco propuesto de evaluación por competencias y percepción estudiantil.

Dimensión	Criterios de evaluación	Evidencias e indicadores propuestos
Competencias técnicas	Requisitos, UML, arquitectura, desarrollo *backend* en Go, bases de datos, seguridad e integración.	Especificaciones, diagramas, módulos funcionales, consultas, mecanismos de seguridad y prototipo integrado.
Competencias UX	Análisis de usuarios, prototipado, pruebas de usabilidad, interpretación de resultados, rediseño y justificación.	Perfiles, *wireframes*, prototipos, registros de prueba, resultados y decisiones de mejora.
Colaboración	Comunicación, liderazgo, distribución de tareas, participación y responsabilidad individual.	Evaluación de pares, cronogramas, registros de trabajo, código y entregables.
Aprendizaje reflexivo	Identificación de aprendizajes, áreas de mejora y relación entre decisiones técnicas y necesidades de usuario.	Reportes reflexivos, discusión final y justificación del rediseño.
Percepción estudiantil	Confianza técnica, comprensión de arquitectura, conciencia UX y seguridad, motivación, colaboración y transferibilidad.	Cuestionario pre/post de autopercepción del aprendizaje.

Nota. Se propone una escala ordinal de 1 a 5 para las rúbricas y una escala Likert de cinco niveles para el cuestionario de percepción. Los instrumentos deberán adaptarse y validarse antes de su aplicación sistemática.

5.2. Usuarios externos participantes

La evaluación formativa incluyó diez participantes externos de 18 a 22 años, todos mexicanos, identificados en los registros originales como U01-U10. La experiencia previa en compras en línea fue media en seis casos, baja en tres y alta en uno; la familiaridad tecnológica fue media en seis casos, alta en tres y baja en uno. Se trató de una muestra pequeña y homogénea que no permite extrapolaciones poblacionales.

5.3. Pretest de usuarios

Antes de utilizar Go-Plant, los diez participantes respondieron ocho preguntas de autoinforme en una escala de 1 a 5 sobre frecuencia de compras en línea, uso de *marketplaces*, facilidad de navegación, malas experiencias, relevancia de la experiencia visual, interés en recomendaciones, abandono por confusión e intención condicional de comprar plantas. El instrumento fue diseñado para la evaluación formativa del proyecto; no se cuenta con evidencia de validación previa por expertos o de propiedades psicométricas.

© RITI. CC BY-NC

El *pretest* registró: frecuencia de compras (media 3,2; DE 0,79); uso de *marketplaces* (3,7; 0,95); facilidad general de navegación (4,1; 0,73); experiencias negativas anteriores (2,9; 1,02); importancia de una buena experiencia visual e interactiva (4,6; 0,52); interés en recomendaciones personalizadas (4,2; 0,71); abandono anterior por confusión (3,0; 1,10); e interés en comprar plantas en línea bajo condiciones de sencillez y confianza (4,5; 0,66). Son resultados descriptivos de percepciones previas; no constituyen una evaluación de la plataforma rediseñada.

5.4. Tareas de usabilidad

Los participantes completaron cinco grupos de tareas:

- Entrar a la plataforma desde un navegador y explorar categorías de productos.
- Leer la descripción de un producto y observar recomendaciones.
- Agregar uno o más productos al carrito, revisar el carrito e intentar modificar o eliminar productos.
- Avanzar al proceso de compra, seleccionar un método de pago y agregar una o más direcciones de envío.
- Identificar aspectos positivos y negativos de la interfaz y sugerir mejoras.

Estas tareas se seleccionaron porque representan el flujo principal de una plataforma *e-commerce*: navegación, descubrimiento de productos, gestión del carrito, pago y retroalimentación.

5.5. Recolección y análisis de datos

Los diez participantes realizaron los cinco grupos de tareas enumerados en la sección 5.4. Después de interactuar con la plataforma se registraron once indicadores binarios (presencia = 1; ausencia = 0) relacionados con navegación, flujo de compra, filtros, tiempos de carga, métodos de pago, carrito, información del producto, recomendaciones y búsqueda. El reporte de origen incluye una matriz por participante (A-J), conteos agregados y comentarios cualitativos. Estos indicadores son observaciones formativas específicas del prototipo, no una escala estandarizada o rúbrica validada de UX. No consta participación documentada de un experto externo en el diseño del instrumento.

Se calcularon frecuencias sobre diez participantes; cada ocurrencia representa diez puntos porcentuales. Para evitar interpretaciones incorrectas, se distinguen indicadores positivos (comodidad y comentarios favorables) de incidencias negativas (dudas, confusión, problemas y frustración). La ausencia de un problema observado no equivale automáticamente a una valoración positiva de ese aspecto. La matriz original es la fuente primaria de los conteos; en el documento de origen existe una discrepancia narrativa de 8/10 frente a 9/10 en métodos de pago, resuelta aquí mediante el conteo de los registros individuales. El análisis cualitativo se utiliza como explicación contextual, sin inferencia estadística ni generalización.

5.6. Segunda actividad de aceptación visual y comparación pareada entre versiones

Como segunda actividad de evaluación, los mismos diez usuarios externos (U01–U10) que participaron en la prueba inicial de usabilidad valoraron visualmente las interfaces inicial y rediseñada de Go-Plant mediante cinco indicadores comunes. Este procedimiento permitió obtener puntuaciones emparejadas por participante y comparar la aceptación visual de ambas versiones. Las pantallas evaluadas incluyeron inicio, catálogo, carrito, envío/pago y confirmación de compra.

Se emplearon cinco ítems en escala Likert de 1 (totalmente en desacuerdo) a 5 (totalmente de acuerdo): I1, identificación de funciones principales; I2, reconocimiento de controles de navegación; I3, claridad de la información de productos; I4, comprensión visual de opciones de envío y pago; e I5, claridad y confianza transmitidas por la organización de la interfaz. Para cada usuario y versión se calculó la media aritmética de los cinco ítems. El instrumento recogió percepción visual y no constituye por sí mismo una medida conductual de eficacia o tiempo de ejecución.

Los registros individuales de los diez participantes se presentan en la Tabla 5. Para cada par se calculó Δi=media rediseñada − media inicial. Se describieron las medias de cada versión, la media y mediana de las diferencias, su desviación estándar y su intervalo observado. Se aplicó una prueba bilateral exacta de rangos con signo de Wilcoxon para muestras relacionadas (α=0.05), con rangos promedio ante empates de magnitud absoluta.

© RITI. CC BY-NC

Se reporta el estadístico W y el valor p exacto; la prueba evalúa las diferencias pareadas y no debe interpretarse como una medición de aprendizaje estudiantil.

Tabla 5. Registros individuales reales de aceptación visual: versión inicial y rediseñada (*n*=10).

Usuario	Versión	I1	I2	I3	I4	I5	Media
U01	Inicial	2	4	3	3	2	2.8
U01	Rediseñada	4	3	5	5	5	4.4
U02	Inicial	1	4	3	2	2	2.4
U02	Rediseñada	4	4	5	4	4	4.2
U03	Inicial	3	2	2	2	2	2.2
U03	Rediseñada	5	4	5	5	3	4.4
U04	Inicial	3	2	1	4	4	2.8
U04	Rediseñada	5	4	3	5	4	4.2
U05	Inicial	2	2	1	4	2	2.2
U05	Rediseñada	5	4	5	5	4	4.6
U06	Inicial	4	3	4	3	3	3.4
U06	Rediseñada	5	3	4	3	4	3.8
U07	Inicial	2	2	3	2	2	2.2
U07	Rediseñada	5	4	5	3	5	4.4
U08	Inicial	3	2	1	3	3	2.4
U08	Rediseñada	5	5	3	4	4	4.2
U09	Inicial	3	3	2	1	2	2.2
U09	Rediseñada	4	5	5	5	4	4.6
U10	Inicial	2	3	3	3	3	2.8
U10	Rediseñada	4	5	4	3	4	4.0

6. Resultados

6.1. Evidencia de aprendizaje basada en artefactos

La versión actual del estudio aporta evidencia del proceso de aprendizaje principalmente a través de los artefactos producidos por los estudiantes, no mediante mediciones cuantitativas pre/post de aprendizaje. Los estudiantes generaron requisitos funcionales y no funcionales, modelos UML, *wireframes* de baja y alta fidelidad, módulos *backend*, arquitectura de base de datos, mecanismos de seguridad, instrumentos de prueba de usabilidad, tablas de observación y rediseños basados en evidencia.

Estos artefactos indican que el proyecto apoyó la integración de varias competencias de ingeniería de software. Las tablas de requisitos y casos de uso evidenciaron la capacidad de traducir necesidades de la plataforma en especificaciones. Los diagramas de arquitectura y la implementación modular mostraron la conexión entre diseño conceptual y estructura de software. Los módulos implementados para productos, login, carrito, gestión de direcciones y pago ofrecieron práctica de desarrollo *backend* y *full-stack*. La prueba de usabilidad y el rediseño aportaron evidencia de pensamiento orientado a UX, especialmente porque los estudiantes identificaron que funcionalidades técnicamente disponibles, como opciones de pago o filtros, no eran necesariamente comprensibles para los usuarios.

6.2. Hallazgos del pretest de usuarios

Las ocho preguntas del pretest muestran que los usuarios otorgaron alta importancia a la calidad visual e interactiva (media=4,6; DE=0,52) y manifestaron interés condicional por comprar plantas en línea si el entorno era sencillo y confiable (4,5; 0,66). También reportaron interés en recomendaciones personalizadas (4,2; 0,71) y una percepción favorable sobre la facilidad general de navegación en tiendas digitales (4,1; 0,73). El uso de *marketplaces* alcanzó 3,7 (DE = 0,95) y la frecuencia de compras en línea, 3,2 (DE=0,79). Estas cifras describen actitudes y experiencias previas de los diez participantes; no prueban efectos atribuibles al proyecto.

© RITI. CC BY-NC

Estos resultados sirvieron como insumo formativo para priorizar la claridad visual, los componentes de búsqueda y las recomendaciones durante el desarrollo del prototipo. No se interpretan como validación empírica de la efectividad de tales componentes.

6.3. Resultados de observación de usabilidad

La prueba de usabilidad identificó fortalezas y dificultades en la interacción con Go-Plant. Entre los aspectos positivos destacaron los comentarios favorables sobre navegación y la comodidad con las recomendaciones (8/10 usuarios en ambos indicadores), así como la interacción con el carrito (7/10). La principal dificultad correspondió a los métodos de pago (9/10), seguida de los filtros y la localización de información de productos (5/10 cada uno). También se registraron problemas de búsqueda (3/10) y confusión durante el proceso de compra (2/10). La Figura 6 presenta las frecuencias observadas y la Tabla 6 sintetiza su interpretación, diferenciando incidencias negativas de valoraciones positivas.

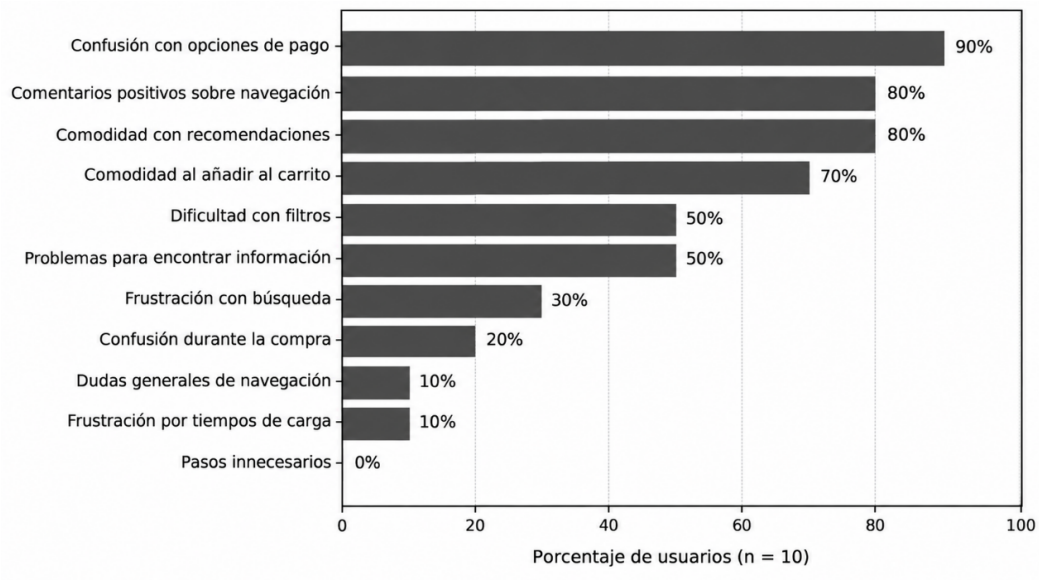


Figura 6. Frecuencia de observaciones de usabilidad identificadas durante la prueba con usuarios externos.

Tabla 6. Síntesis de hallazgos principales de usabilidad.

Aspecto evaluado	Resultado	Interpretación
Comentarios positivos sobre navegación	8/10	La mayoría expresó una valoración favorable de la navegación.
Comodidad al agregar productos al carrito	7/10	Siete participantes manifestaron comodidad al utilizar el carrito.
Comodidad usando recomendaciones	8/10	La mayoría manifestó comodidad al utilizar las recomendaciones.
Confusión con opciones de pago	9/10	La selección de métodos de pago constituyó la principal dificultad observada.
Dificultad para filtrar productos	5/10	La mitad de los participantes presentó dificultades al utilizar los filtros.
Problemas para encontrar información de producto	5/10	La localización de información relevante presentó dificultades para la mitad de los participantes.
Frustración con búsqueda	3/10	Tres participantes manifestaron frustración durante la búsqueda de productos.
Confusión durante el proceso de compra	2/10	Se identificaron dificultades de comprensión del flujo de compra en dos participantes.
Dudas generales de navegación	1/10	Un participante manifestó dudas generales de navegación.

© RITI. CC BY-NC

Frustración por tiempos de carga	1/10	Un participante manifestó frustración relacionada con los tiempos de carga.
Dificultad por pasos innecesarios	0/10	No se registraron dificultades atribuidas a pasos innecesarios durante la prueba.

6.4. Hallazgos cualitativos

La retroalimentación cualitativa fue consistente con las frecuencias observadas en la evaluación de usabilidad. Los usuarios describieron la plataforma como generalmente intuitiva, organizada y visualmente clara. Los comentarios positivos enfatizaron que la navegación era comprensible y que el diseño transmitía orden. No obstante, varios participantes consideraron confusas las opciones de pago y no sabían cuál seleccionar. Otros indicaron que los filtros eran difíciles de localizar o no apoyaban claramente sus preferencias. Algunos usuarios también reportaron que la información del producto era incompleta o no suficientemente visible.

Estos hallazgos ayudaron a los estudiantes a distinguir entre disponibilidad de una funcionalidad y usabilidad de esa funcionalidad. El módulo de pago existía y operaba, pero los usuarios seguían experimentando incertidumbre. El módulo de filtros estaba presente, pero su visibilidad y organización eran insuficientes. La información del producto estaba disponible, pero su distribución no siempre apoyaba la toma rápida de decisiones. Esta distinción constituyó un aspecto formativo relevante del proyecto, al vincular las decisiones de implementación con los problemas de usabilidad identificados durante las pruebas.

6.5. Rediseño iterativo basado en retroalimentación de usuarios

Los resultados de la primera prueba de usabilidad se utilizaron para fundamentar ajustes de diseño. Las prioridades fueron: aclarar opciones de pago (9/10 casos de confusión), aumentar la visibilidad de filtros y de información del producto (5/10 cada uno) y facilitar la búsqueda (3/10 casos de frustración). La Tabla 7 documenta la trazabilidad entre problema observado y decisión de diseño.

Como referencia para analizar y justificar las decisiones de interfaz, se consideraron las heurísticas de usabilidad de [18] y las recomendaciones especializadas para interfaces de comercio electrónico de [19]. Estas referencias proporcionan criterios para organizar elementos de navegación, presentar alternativas de pago, facilitar el reconocimiento de las opciones disponibles y mantener informado al usuario durante los procesos de interacción.

Los principios de diseño considerados para fundamentar las modificaciones de Go-Plant comprenden los siguientes elementos:

1. Visibilidad del estado del sistema: presentación de indicadores que permiten identificar la etapa actual del proceso de compra y recibir retroalimentación sobre las acciones realizadas.
2. Consistencia y estándares: utilización de convenciones reconocibles para organizar controles, categorías, filtros y mecanismos de navegación, reduciendo la necesidad de aprendizaje por parte del usuario.
3. Reconocimiento antes que recuerdo: presentación visible de opciones, características y acciones disponibles, evitando que el usuario deba recordar información proporcionada en pantallas anteriores.
4. Jerarquía visual y diseño minimalista: organización de información mediante agrupaciones, títulos, elementos destacados y reducción de contenido visual innecesario, facilitando la identificación de información relevante.
5. Control y libertad del usuario: disponibilidad de mecanismos comprensibles para modificar productos, cantidades y selecciones durante el proceso de compra.
6. Prevención de errores: presentación de instrucciones, validaciones y mensajes contextuales que contribuyen a disminuir equivocaciones durante la captura de información y la selección de métodos de pago.

La trazabilidad del rediseño se estableció relacionando los problemas identificados durante la prueba de usabilidad con las modificaciones propuestas y los principios de diseño aplicados. Como se muestra en la Tabla 7, las intervenciones priorizaron la comprensión de los métodos de pago, la visibilidad de los filtros, la organización de la información de productos y la disponibilidad de funciones de búsqueda. Asimismo, se incorporaron controles para gestionar el carrito y elementos de confirmación que permiten reconocer el estado del proceso de compra.

© RITI. CC BY-NC

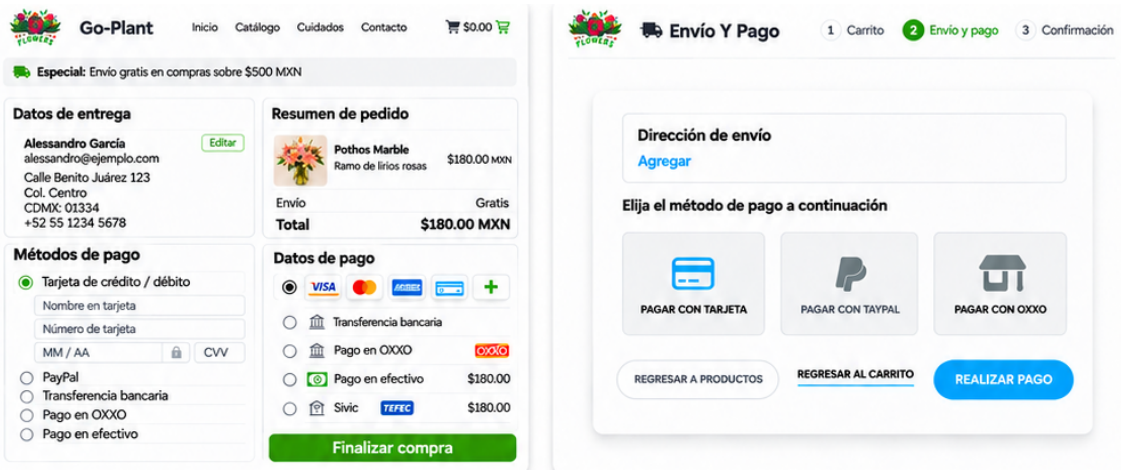
Estas decisiones se fundamentaron en criterios de visibilidad, consistencia, reconocimiento y control del usuario [18], así como en recomendaciones para interfaces de pago en comercio electrónico [19].

Tabla 7. Trazabilidad entre problemas de usabilidad y decisiones de rediseño de Go-Plant.

Hallazgo	Evidencia	Rediseño y principio UX
Pago confuso	9/10 usuarios	Flujo guiado y opciones visibles; reconocimiento y prevención de errores.
Filtros poco visibles	5/10 usuarios	Filtros laterales y etiquetas claras; consistencia y reconocimiento.
Información difícil de localizar	5/10 usuarios	Tarjetas y detalles reorganizados; jerarquía visual.
Búsqueda limitada	3/10 usuarios	Acceso destacado a búsqueda; visibilidad y accesibilidad de funciones.
Controles del carrito mejorables	Observación cualitativa	Cantidad, eliminación y resumen; control del usuario.
Cierre de compra poco claro	Observación cualitativa	Confirmación e indicador de progreso; visibilidad del estado.

La pantalla de productos se rediseñó para atender dificultades con filtros e interacción. Se mejoró la visualización de productos, se incorporó un botón de agregar al carrito más claro y se añadió retroalimentación visual para confirmar que un producto se agregó correctamente. El carrito se rediseñó como una interfaz estructurada de gestión de compra, con imágenes, precio, controles de cantidad, opciones de eliminación, subtotal, métodos de envío, código promocional y botón prominente para proceder al pago.

La interfaz de envío y pago se rediseñó mediante la reorganización de los métodos de pago en tarjetas visuales con iconos, la separación de la dirección de envío y la incorporación de un indicador de progreso (carrito, envío y pago, confirmación). Además, se añadió una pantalla de confirmación con el estado, los datos y el resumen del pedido. La Figura 7 compara las interfaces de pago inicial y rediseñada, componente que presentó mayor confusión durante la evaluación inicial (9/10 usuarios). Las demás modificaciones se documentan en la Tabla 7, mientras que la sección 6.6 presenta el contraste de aceptación visual realizado con los mismos diez participantes, sin atribuir mejoras comprobadas al desempeño funcional.



a) **b)**

Figura 7. Comparación de la interfaz de pago de Go-Plant a) antes y b) después del rediseño, destacando la reorganización de opciones, la guía visual del flujo y la ruta de selección del método de pago.

6.6. Resultados de aceptación visual y comparación estadística entre versiones

La segunda actividad de aceptación visual permitió comparar las interfaces inicial y rediseñada mediante las valoraciones de los mismos diez participantes. Como se presenta en la Tabla 8, la versión inicial obtuvo una media de 2.54 puntos, frente a 4.28 de la rediseñada, con una diferencia estadísticamente significativa según la prueba de rangos con signo de Wilcoxon (W=0.00; p=0.001953). Estos resultados indican una mayor aceptación visual de la interfaz rediseñada, sin constituir una medición directa de mejoras en el desempeño funcional.

© RITI. CC BY-NC

Tabla 8. Síntesis de aceptación visual y comparación pareada entre versiones (n=10).

Indicador	Resultado
Media inicial	2.54 / 5
Media rediseñada	4.28 / 5
Incremento medio	+1.74 puntos
Mediana de las diferencias	1.80 puntos
Desviación estándar de diferencias	0.626
Wilcoxon exacto bilateral	W=0.00; p=0.001953
Diferencias positivas	10 de 10

7. Discusión

La experiencia Go-Plant ilustra cómo un proyecto *e-commerce* realista puede apoyar la integración de ingeniería de software y educación en experiencia de usuario. Al combinar ABP, ADDIE y DCU, los estudiantes avanzaron por un ciclo completo de desarrollo que incluyó requisitos, diseño, implementación, evaluación y rediseño. Esta estructura les permitió conectar artefactos técnicos con resultados centrados en el usuario.

La contribución pedagógica más importante es la conexión explícita entre implementación y evidencia empírica de uso. En muchos cursos de software, el proyecto concluye cuando el sistema funciona técnicamente. En Go-Plant, la funcionalidad técnica fue solo una etapa. Los estudiantes también debieron observar usuarios, interpretar problemas de usabilidad y justificar decisiones de rediseño. Esta experiencia permitió abordar la calidad del software desde una perspectiva integral, considerando funcionalidad, usabilidad, seguridad, mantenibilidad y confianza del usuario [1], [2], [3], [10].

El proyecto también muestra el valor de las plataformas *e-commerce* como ambientes educativos. Las tareas de compra son suficientemente familiares para que usuarios externos participen sin entrenamiento extenso, pero suficientemente complejas para involucrar preocupaciones significativas de ingeniería de software. Catálogos, filtros, gestión del carrito, flujo de pago y pantallas de confirmación exigen integrar lógica *backend*, operaciones de base de datos, interacción *frontend* y decisiones UX. Esto convierte al *e-commerce* en un dominio adecuado para ABP auténtico en educación en ingeniería de software [11], [12].

El marco ABP-ADDIE-DCU funcionó como andamiaje instruccional. El ABP situó el aprendizaje en un problema auténtico; ADDIE ayudó a organizar fases, entregables y evaluación; y el DCU aseguró que el artefacto técnico se analizara desde la perspectiva de usuarios. Esta combinación evitó tratar UX como un paso decorativo final. En cambio, UX se integró en requisitos, prototipado, validación y rediseño [4], [6], [9], [16].

El uso de Go contribuyó al valor educativo al exponer a los estudiantes a conceptos de ingeniería *backend* en un lenguaje asociado con desarrollo web modular y escalable. Los estudiantes conectaron rutas, controladores, modelos, acceso a datos y plantillas con interacciones visibles para el usuario. Esta actividad permitió relacionar las decisiones de implementación con el comportamiento percibido de la plataforma desde la perspectiva del usuario [14], [15].

El caso debe interpretarse como exploratorio. Los registros individuales y la comparación pareada entre versiones aportan evidencia cuantitativa de una mayor aceptación visual de la interfaz rediseñada entre los diez usuarios participantes. Sin embargo, estos datos no miden ganancias de aprendizaje estudiantil ni establecen un efecto causal del diseño instruccional. En futuras cohortes será importante aplicar rúbricas de aprendizaje, mediciones educativas pre/post y pruebas funcionales con tiempos, errores y finalización de tareas.

8. Limitaciones y amenazas a la validez

Deben reconocerse varias limitaciones. Primero, el número de usuarios externos fue pequeño, lo que limita la generalización de los hallazgos de usabilidad. La muestra fue adecuada para una prueba formativa dentro de un proyecto de curso, pero no para formular afirmaciones amplias sobre todos los usuarios de plataformas *e-commerce*. Segundo, los participantes fueron adultos jóvenes con características demográficas relativamente similares; por ello, futuros estudios deben incluir usuarios más diversos en edad, alfabetización digital, necesidades de accesibilidad y experiencia de compra en línea.

© RITI. CC BY-NC

Tercero, la observación inicial se basó principalmente en autoinformes e indicadores binarios, sin registros comparables de tiempos de finalización, errores por tarea o abandono efectivo; la medición posterior documenta aceptación visual, no desempeño conductual. Cuarto, no hubo grupo de control ni mediciones educativas pre/post, por lo que no se atribuyen mejoras de aprendizaje al diseño instruccional. Quinto, el estudio se realizó en una institución y curso, por lo que el contexto, la guía docente, los conocimientos previos y el orden de exposición a las versiones pudieron influir en las valoraciones.

La comparación visual pareada se realizó con los mismos diez participantes, con una muestra pequeña y homogénea. Aunque se identificó una diferencia estadísticamente significativa en aceptación visual, el instrumento específico de cinco ítems no dispone de validación psicométrica documentada. Los resultados deben distinguirse de la primera prueba de usabilidad y no demuestran por sí solos una reducción objetiva de incidencias de pago, errores o tiempos de ejecución.

9. Consideraciones éticas, disponibilidad de datos y conflicto de interés

Los usuarios externos participaron en actividades formativas y los resultados se presentan de forma agregada. El proceso de pago corresponde a una simulación educativa y no se documentan pagos reales.

Los datos agregados y los instrumentos de evaluación pueden ponerse a disposición bajo solicitud razonable, siempre que no comprometan la privacidad de estudiantes o participantes externos. Los autores declaran no tener conflictos de interés relacionados con este estudio.

10. Conclusiones

Este estudio presentó Go-Plant como experiencia de Aprendizaje Basado en Proyectos para enseñar desarrollo *e-commerce* centrado en el usuario en educación en ingeniería. La plataforma permitió integrar ingeniería de requisitos, modelado UML, desarrollo *backend* con Go, diseño de bases de datos, interfaces responsivas, mecanismos de seguridad, pruebas de usabilidad y rediseño iterativo.

La integración de ABP, ADDIE y DCU organizó las fases del proyecto, desde los requisitos y el prototipado hasta las pruebas con usuarios y el rediseño. La primera prueba con diez usuarios identificó confusión con los métodos de pago (9/10), dificultades de filtrado (5/10), problemas para localizar información de producto (5/10) y frustración con la búsqueda (3/10). Las pantallas modificadas de inicio, catálogo, carrito, pago y confirmación permiten documentar las decisiones de interfaz. En la segunda actividad, los mismos diez usuarios evaluaron ambas versiones: la puntuación media de aceptación visual pasó de 2.54 a 4.28, con diferencia pareada significativa (Wilcoxon bilateral exacta: W=0.00; p=0.001953).

La principal contribución es documentar un caso de aplicación del diseño instruccional ABP-ADDIE-DCU utilizando Go-Plant como artefacto integrador de ingeniería de software y evaluación centrada en el usuario. El estudio aporta trazabilidad entre problemas observados, decisiones concretas de rediseño y aceptación visual posterior. Su alcance sigue siendo exploratorio: la evidencia no permite atribuir ganancias de aprendizaje de los estudiantes ni mejoras funcionales a partir del cuestionario visual; futuras investigaciones deben ampliar las muestras y realizar mediciones educativas y conductuales complementarias.

11. Referencias

[1] IEEE Computer Society. (2024). *Software engineering body of knowledge (SWEBOK Guide) (Versión 4.0)*. https://www.computer.org/education/bodies-of-knowledge/software-engineering

[2] International Organization for Standardization & International Electrotechnical Commission. (2023). *Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Product quality model (ISO/IEC 25010:2023)*. https://www.iso.org/standard/78176.html

[3] International Organization for Standardization. (2018). *Ergonomics of human-system interaction—Part 11: Usability: Definitions and concepts (ISO 9241-11:2018)*. https://www.iso.org/standard/63500.html

[4] Lavado-Anguera, S., Velasco-Quintana, P.-J., Terrón-López, M.-J. (2024). Project-based learning (PBL) as an experiential pedagogical methodology in engineering education: A review of the literature. *Education Sciences, 14* (6), 1-20. https://doi.org/10.3390/educsci14060617

© RITI. CC BY-NC

[5] Ramírez de Dampierre, M., Gaya-López, M. C., Lara-Bercial, P. J. (2024). Evaluation of the implementation of project-based-learning in engineering programs: A review of the literature. *Education Sciences, 14* (10), 1-10. https://doi.org/10.3390/educsci14101107

[6] Ingale, S., Kulkarni, V., Patil, A., Vibhute, S. (2024). Implementing project-based learning (PBL) in engineering education: An analytical study of student engagement and learning outcomes with statistical insights. *Educational Administration: Theory and Practice, 30* (1), 4333–4342. https://doi.org/10.53555/kuey.v30i1.8004

[7] Escobar-Castillejos, D., Sigüenza-Noriega, I., Noguez, J., Escobar-Castillejos, D., Berumen-Glinz, L. A. (2024). Enhancing methods engineering education with a digital platform: Usability and educational impact on industrial engineering students. *Frontiers in Education*, *9*, 1-15. https://doi.org/10.3389/feduc.2024.1438882

[8] Menzi Çetin, N. (2024). Integrating usability into software engineering course projects. *Journal of Computer and Education Research, 12* (23), 209–240. https://doi.org/10.18009/jcer.1417245

[9] Hakobyan, L., Sowriraghavan, A. (2025). Enhancing engineering education through user-centred design integration. En B. K. Smith, M. Borge (Eds.), *Learning and collaboration technologies* (Lecture Notes in Computer Science, Vol. 15806, pp. 17–36). Springer. https://doi.org/10.1007/978-3-031-93564-0_2

[10] International Organization for Standardization. (2019). *Ergonomics of human-system interaction—Part 210: Human-centred design for interactive systems (ISO 9241-210:2019)*. https://www.iso.org/standard/77520.html

[11] Hasan, T. I., Silalahi, C. I., Rumagit, R. Y., Pratama, G. D. (2024). UI/UX design impact on e-commerce attracting users. *Procedia Computer Science, 245*, 1075–1082. https://doi.org/10.1016/j.procs.2024.10.336

[12] Scott, E. (2024). *2024 E-Commerce Checkout: Expanded and Updated Checkout Research Findings*. Baymard Institute. https://baymard.com/research-articles/checkout-2024-launch

[13] McCloskey, M. (2014). *Turn user goals into task scenarios for usability testing.* Nielsen Norman Group. https://www.nngroup.com/articles/task-scenarios-usability-testing/

[14] The Go Authors. (2025). *The Go programming language specification*. https://go.dev/ref/spec

[15] The Go Authors. (2024). *Effective Go*. https://go.dev/doc/effective_go

[16] Spatioti, A. G., Kazanidis, I., Pange, J. (2022). A comparative study of the ADDIE instructional design model in distance education. *Information, 13* (9), 1-20. https://doi.org/10.3390/info13090402

[17] Cico, O., Jaccheri, L., Nguyen-Duc, A., Zhang, H. (2021). Exploring the intersection between software industry and software engineering education: A systematic mapping of software engineering trends. *Journal of Systems and Software, 172*, 1-28. https://doi.org/10.1016/j.jss.2020.110736

[18] Nielsen, J. (1994). *10 usability heuristics for user interface design*. Nielsen Norman Group. https://www.nngroup.com/articles/ten-usability-heuristics/

[19] Scott, E. (2023). *Payment method UX: Designing payment selection*. Baymard Institute. https://baymard.com/research-articles/payment-method-selection

© RITI. CC BY-NC