

Documento de Propuesta de Diseño de Software I, II y III

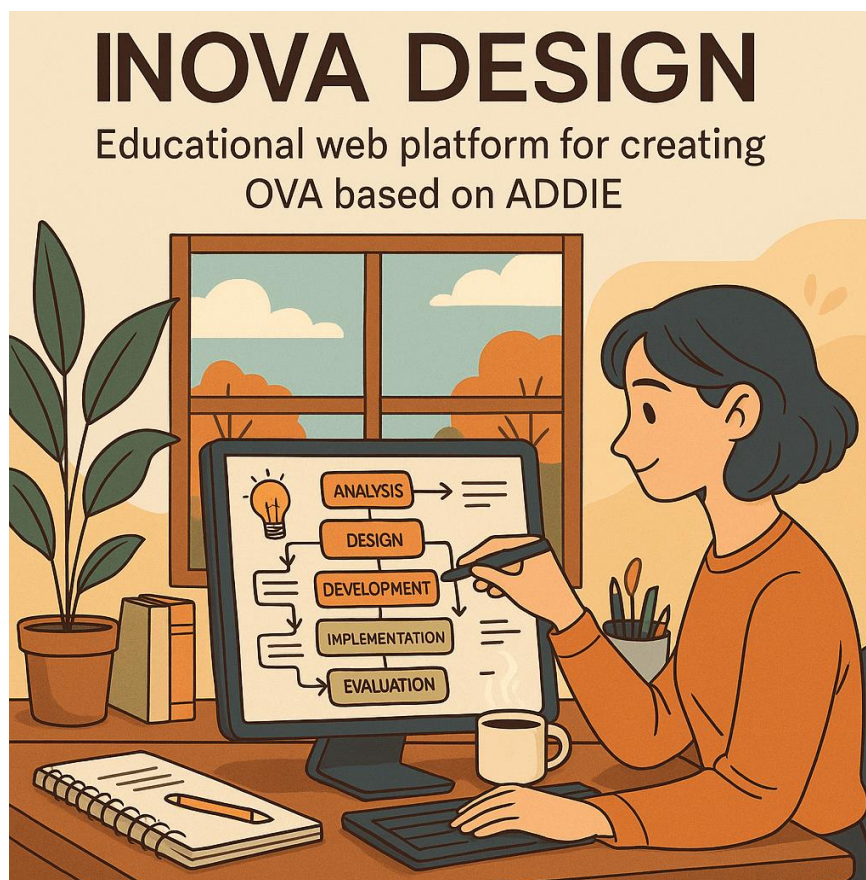
InOva Design

Autores

- Duberney Barrera Ortega dbarreraortega83@correo.unicordoba.edu.co
- Jesús David Ceballos Diaz jceballosdiaz@correo.unicordoba.edu.co
- Gabriela García Gil - ggarciagil@correo.unicordoba.edu.co

Tutor

- Alexander Toscano Ricardo atoscano@correo.unicordoba.edu.co



https://github.com/area-de-informatica/ds1_pa_inovadesign.git

Descripción Del Software

InOva Design es una plataforma web educativa creada con el propósito de facilitar a los estudiantes la creación de Objetos Virtuales de Aprendizaje (OVA) de manera didáctica, estructurada y accesible. Esta herramienta guía a los usuarios mediante un proceso paso a paso, basado en el modelo pedagógico ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación). A lo largo de cada etapa, la plataforma ofrece orientaciones claras, recursos interactivos y plantillas que ayudan a organizar ideas, desarrollar contenidos educativos y diseñar experiencias de aprendizaje efectivas. Todos los OVA creados con esta herramienta son compatibles con el estándar SCORM (Sharable Content Object Reference Model), lo que permite su integración en plataformas de aprendizaje virtual (LMS), asegurando su reutilización, seguimiento y adaptabilidad a diferentes contextos educativos.

Contenido

1. ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS	6
INTRODUCCIÓN.....	6
PROPÓSITO DEL DOCUMENTO.....	6
ALCANCE DEL PROYECTO.....	7
DEFINICIONES Y ACRÓNIMOS	9
2. DESCRIPCIÓN GENERAL.....	11
OBJETIVOS DEL SISTEMA.	11
FUNCIONALIDAD GENERAL.	11
USUARIOS DEL SISTEMA	12
RESTRICCIONES.	13
3. REQUISITOS FUNCIONALES.....	14
MAKUP DE LA INTERFAZ DE USUARIO (UI).....	15
CASOS DE USO – DIAGRAMA DE CASOS DE USO.....	15
DESCRIPCIÓN DETALLADA DE CADA CASO DE USO.....	16
DIAGRAMAS DE FLUJO DE CASOS DE USO 1.....	16
DIAGRAMAS DE FLUJO DE CASOS DE USO 2.....	18
DIAGRAMAS DE FLUJO DE CASOS DE USO 3.....	19
DIAGRAMAS DE FLUJO DE CASOS DE USO 4.....	20
DIAGRAMAS DE FLUJO DE CASOS DE USO 5.....	21
DIAGRAMAS DE FLUJO DE CASOS DE USO 6.....	23
DIAGRAMAS DE FLUJO DE CASOS DE USO 7.....	24
DIAGRAMAS DE FLUJO DE CASOS DE USO 8.....	25
DIAGRAMAS DE FLUJO DE CASOS DE USO 9.....	26
DIAGRAMAS DE FLUJO DE CASOS DE USO 10.....	27
PRIORIDAD DE REQUERIMIENTOS.	28
4. REQUISITOS NO FUNCIONALES.	29
REQUISITOS DE DESEMPEÑO.....	29
REQUISITOS DE SEGURIDAD.....	29
REQUISITOS DE USABILIDAD	29
REQUISITOS DE ESCALABILIDAD	29
REQUISITOS DE DISPONIBILIDAD	30
COMPATIBILIDAD MULTIPLATAFORMA	30
ACCESIBILIDAD	30
MODELADO E/R.....	31
DIAGRAMA DE ENTIDAD-RELACIÓN	38
DIAGRAMA RELACIONAL.....	38
SCRIPT DE MODELO RELACIONAL.....	38
DESCRIPCIÓN DE ENTIDADES Y RELACIONES.....	39
REGLAS DE INTEGRIDAD REFERENCIAL	40
COLECCIONES (NoSLQ).....	40
ANEXOS	40
DIAGRAMAS ADICIONALES.....	40
REFERENCIAS	40
ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND	41
INTRODUCCIÓN	41

PROPÓSITO DE LA ETAPA	41
ALCANCE DE LA ETAPA	41
DEFINICIONES Y ACRÓNIMOS	41
DISEÑO DE LA ARQUITECTURA DE BACKEND.....	41
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA	41
COMPONENTES DEL BACKEND.....	41
DIAGRAMAS DE ARQUITECTURA	41
ELECCIÓN DE LA BASE DE DATOS	41
EVALUACIÓN DE OPCIONES (SQL o NoSQL).....	41
JUSTIFICACIÓN DE LA ELECCIÓN.....	41
DISEÑO DE ESQUEMA DE BASE DE DATOS	41
IMPLEMENTACIÓN DEL BACKEND.....	42
ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN.....	42
CREACIÓN DE LA LÓGICA DE NEGOCIO	42
DESARROLLO DE ENDPOINTS Y APIS.....	42
AUTENTICACIÓN Y AUTORIZACIÓN	42
CONEXIÓN A LA BASE DE DATOS	42
CONFIGURACIÓN DE LA CONEXIÓN	42
DESARROLLO DE OPERACIONES CRUD.....	42
MANEJO DE TRANSACCIONES	42
PRUEBAS DEL BACKEND	42
DISEÑO DE CASOS DE PRUEBA.....	42
EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN.....	43
MANEJO DE ERRORES Y EXCEPCIONES	43
ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND	44
INTRODUCCIÓN.....	44
PROPÓSITO DE LA ETAPA	44
ALCANCE DE LA ETAPA	44
DEFINICIONES Y ACRÓNIMOS	44
CREACIÓN DE LA INTERFAZ DE USUARIO (UI).....	44
DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS	44
CONSIDERACIONES DE USABILIDAD	44
MAQUETACIÓN RESPONSIVA	44
PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS).....	44
DESARROLLO DE LA LÓGICA DEL FRONTEND	44
MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS	44
USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE)	45
CONSUMO DE DATOS DESDE EL BACKEND.....	45
CONFIGURACIÓN DE CONEXIONES AL BACKEND	45
OBTENCIÓN Y PRESENTACIÓN DE DATOS	45
ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE)	45
INTERACCIÓN USUARIO-INTERFAZ	45
MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS	45
IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS	45

MEJORAS EN LA EXPERIENCIA DEL USUARIO	45
PRUEBAS Y DEPURACIÓN DEL FRONTEND	45
DISEÑO DE CASOS DE PRUEBA DE FRONTEND	45
PRUEBAS DE USABILIDAD.....	45
DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO.....	46
IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND	46
MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO)	46
VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND	46
INTEGRACIÓN CON EL BACKEND	46
VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND	46
PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND	46

1.Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Introducción

Propósito del Documento

El presente proyecto tiene como objetivo desarrollar una plataforma web educativa llamada **InOva Design**, orientada a guiar a los usuarios en la creación de Objetos Virtuales de Aprendizaje (OVA) basados en el modelo pedagógico **ADDIE** y compatibles con el estándar **SCORM**. La plataforma permitirá estructurar contenidos educativos, diseñar actividades interactivas y crear evaluaciones formativas, asegurando su correcta integración en plataformas de gestión del aprendizaje (LMS). A través de un enfoque interactivo y progresivo, se busca mejorar la experiencia de aprendizaje y facilitar el proceso de diseño de OVA por parte de los usuarios.

- **Etapa 1: Diseño de la Aplicación y Análisis de Requisitos**

Durante esta primera fase del proyecto se llevó a cabo el análisis detallado de los requerimientos funcionales y no funcionales del sistema InOva Design. Se definió el público objetivo y se establecieron los roles de usuario principales, identificando las necesidades específicas del entorno educativo para la creación de Objetos Virtuales de Aprendizaje (OVA). Posteriormente, se elaboró el diseño conceptual del sistema, incluyendo los casos de uso, el modelo entidad-relación (E/R), la arquitectura general de la aplicación y los primeros bocetos de interfaz de usuario. Esta etapa permitió establecer la base teórica, funcional y pedagógica para el desarrollo del proyecto, alineándose con el modelo instruccional ADDIE y los estándares SCORM, asegurando una estructura clara y escalable para las siguientes fases.

- **Etapa 2: Persistencia de Datos con Backend – Servidor**

En esta etapa se procedió con la implementación de la lógica de negocio del sistema, desarrollando el backend encargado de gestionar la persistencia de datos mediante una base de datos relacional. Se programaron los endpoints necesarios para permitir el registro, autenticación, creación y recuperación de OVAs por parte de los usuarios. A su vez, se garantizaron los principios de seguridad y consistencia en el manejo de la información. Además, se realizó la validación de los formularios y se implementaron controles para proteger los datos sensibles. Esta fase fue esencial para garantizar que la información educativa y las acciones del usuario quedaran almacenadas correctamente y pudieran ser consultadas por la plataforma de forma eficiente y segura.

- **Etapa 3: Consumo de Datos y Desarrollo Frontend – Cliente**

La tercera etapa del proyecto consistió en la construcción de la interfaz gráfica de la plataforma InOva Design, orientada a ofrecer una experiencia interactiva y pedagógica al usuario. Se implementó un diseño web adaptable utilizando tecnologías, permitiendo a los usuarios crear sus OVAs paso a paso siguiendo las fases del modelo ADDIE. Se integraron los servicios desarrollados en el backend para consumir y mostrar los datos de manera

dinámica, brindando acceso a funcionalidades como creación de contenido, vista previa, descarga y evaluación. La plataforma fue sometida a pruebas de usabilidad y funcionamiento, lo que permitió realizar ajustes que mejoraron la navegación, accesibilidad y presentación del contenido educativo. Esta etapa consolidó el sistema como una herramienta educativa funcional y completa.

Alcance del Proyecto

El proyecto **InOva Design** tiene como alcance el desarrollo de una plataforma web educativa que guíe a los usuarios en la creación de Objetos Virtuales de Aprendizaje (OVA) siguiendo las etapas del modelo pedagógico ADDIE. La plataforma permitirá registrar y autenticar usuarios, presentar contenidos teóricos por cada fase, generar formularios interactivos, validar los aportes ingresados de forma manual o mediante inteligencia artificial en el futuro, y ofrecer una vista previa del OVA construido. Además, los avances serán almacenados en una base de datos, garantizando la continuidad del proceso de aprendizaje.

Nº	Funcionalidad	Descripción
1	Registrar usuario	Permitir a los usuarios crear una cuenta para guardar su avance.
2	Iniciar sesión (Login)	Acceso de usuarios registrados para continuar su progreso.
3	Cargar la sección de una fase del módulo ADDIE	Mostrar contenido educativo y formularios sobre una fase.
4	Generar formulario interactivo por etapa	Cada módulo tendrá un formulario para que el usuario aplique lo aprendido.
5	Evaluar aportes del usuario en una sección de la fase del módulo ADDIE	Validación de los análisis creados, manualmente o con IA en el futuro.
6	Generar retroalimentación automática por IA	Sugerencias sobre mejoras en lo que el usuario ha escrito.
7	Guardar avances del usuario	Cada acción importante se almacena en el servidor para seguridad.
8	Visualizar progreso del usuario	Mostrar al usuario cuánto ha completado en cada módulo ADDIE.
9	Generar vista previa del OVA construido	Ver cómo quedaría el OVA final basado en sus respuestas.
10	Reiniciar proceso desde cero	Botón que limpia todo el avance para comenzar de nuevo.

Funcionalidades Futuras

Nº	Funcionalidad Futura	Descripción
1	Exportar OVA en formato SCORM	Permitir que el usuario descargue su OVA creado como archivo SCORM.
2	Validar automáticamente aportes mediante IA	Utilizar inteligencia artificial para evaluar los análisis, diseños, desarrollos, implementaciones y evaluaciones del usuario.
3	Seleccionar plantillas visuales para OVA	Permitir que el usuario elija entre diferentes diseños o estilos de presentación.
4	Subir archivos multimedia al OVA	Dar opción al usuario de adjuntar imágenes, audios o documentos a su proyecto.
5	Generar certificado de finalización	Crear un diploma digital cuando el usuario complete el diseño de su OVA.
6	Mostrar progreso del usuario con barra de avance	Visualizar el avance de módulos completados mediante una barra de progreso.
7	Generar evaluación final del proceso	Calificar el OVA completo del usuario al finalizar todas las etapas.
8	Activar modo accesible en la plataforma	Permitir cambiar la visualización a alto contraste, fuentes grandes y navegación accesible.
9	Generar actividades de tipo arrastrar y soltar	Permitir que los usuarios diseñen actividades interactivas de “drag and drop” para sus OVA.

Definiciones y Acrónimos

API: Interfaz de Programación de Aplicaciones (Application Programming Interface).

DBMS: Sistema de Gestión de Bases de Datos (Database Management System).

SQL: Lenguaje de Consulta Estructurada (Structured Query Language).

HTTP: Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol).

REST: Transferencia de Estado Representacional (Representational State Transfer).

JSON: Notación de Objetos de JavaScript (JavaScript Object Notation).

JWT: Token de Web JSON (JSON Web Token).

CRUD: Crear, Leer, Actualizar y Borrar (Create, Read, Update, Delete).

ORM: Mapeo Objeto-Relacional (Object-Relational Mapping).

MVC: Modelo-Vista-Controlador (Model-View-Controller).

API RESTful: API que sigue los principios de REST.

CI/CD: Integración Continua / Entrega Continua (Continuous Integration / Continuous Delivery).

SaaS: Software como Servicio (Software as a Service).

SSL/TLS: Capa de sockets seguros/Seguridad de la Capa de Transporte (Secure Sockets Layer/Transport Layer Security).

HTML: Lenguaje de Marcado de Hipertexto (Hypertext Markup Language).

CSS: Hojas de Estilo en Cascada (Cascading Style Sheets).

JS: JavaScript.

DOM: Modelo de Objeto del Documento (Document Object Model).

UI: Interfaz de Usuario (User Interface).

UX: Experiencia del Usuario (User Experience).

SPA: Aplicación de Página Única (Single Page Application).

AJAX: Asincrónico JavaScript y XML (Asynchronous JavaScript and XML).

CMS: Sistema de Gestión de Contenido (Content Management System).

CDN: Red de Distribución de Contenido (Content Delivery Network).

SEO: Optimización de Motores de Búsqueda (Search Engine Optimization).

IDE: Entorno de Desarrollo Integrado (Integrated Development Environment).

CLI: Interfaz de Línea de Comandos (Command Line Interface).

PWA: Aplicación Web Progresiva (Progressive Web App).

RU: Registrar usuario

IS: Iniciar sesión (Login)

CSFMA: Cargar la sección de una fase del módulo ADDIE

GFIE: Generar formulario interactivo por etapa

EAUSFMA: Evaluar aportes del usuario en una sección de la fase del módulo ADDIE

GERAI: Generar retroalimentación automática por IA

GAU: Guardar avances del usuario

VPU: Visualizar progreso del usuario

GVPOC: Generar vista previa del OVA construido

RPDC: Reiniciar proceso desde cero

EOFS: Exportar OVA en formato SCORM

VAAMI: Validar automáticamente aportes mediante IA

SPVO: Seleccionar plantillas visuales para OVA

SAMO: Subir archivos multimedia al OVA

GCF: Generar certificado de finalización

MPUBA: Mostrar progreso del usuario con barra de avance

GEFP: Generar evaluación final del proceso

AMAP: Activar modo accesible en la plataforma

GATAS: Generar actividades de tipo arrastrar y soltar

2. DESCRIPCIÓN GENERAL.

Objetivos del sistema.

El objetivo del sistema InOva Design es proporcionar una plataforma web educativa orientada a guiar a los usuarios —principalmente docentes y estudiantes— en la creación estructurada de Objetos Virtuales de Aprendizaje (OVA), mediante el uso del modelo pedagógico ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación). La plataforma busca facilitar el diseño instruccional a través de formularios guiados, retroalimentación automática con inteligencia artificial, y funcionalidades interactivas que permiten transformar ideas pedagógicas en objetos digitales reutilizables y compatibles con estándares SCORM.

El sistema también tiene como propósito fomentar la autonomía del usuario en la producción de recursos didácticos, mediante un entorno intuitivo, modular y alineado con buenas prácticas de accesibilidad, usabilidad y seguimiento del proceso de aprendizaje.

Funcionalidad General.

Registro y Autenticación de Usuarios: Los usuarios pueden registrarse en la plataforma y acceder de forma segura a sus proyectos y avances.

Navegación Guiada por el Modelo ADDIE: La plataforma está estructurada por fases (Análisis, Diseño, Desarrollo, Implementación y Evaluación), cada una con contenidos, formularios y actividades asociadas.

Formularios Interactivos por Etapa: Cada fase incluye un formulario para registrar los aportes del usuario, validables de forma manual o automatizada.

Retroalimentación Automática por IA: Los aportes textuales del usuario pueden ser analizados mediante inteligencia artificial para ofrecer sugerencias de mejora.

Vista Previa del OVA: Permite visualizar el resultado parcial o final del OVA con base en las respuestas del usuario.

Almacenamiento Automático de Avances: El sistema guarda continuamente el progreso del usuario para evitar pérdida de información.

Reinicio de Proceso: Posibilidad de comenzar nuevamente un proyecto desde cero si así lo desea el usuario.

Visualización del Progreso: Se muestra el avance en cada módulo del modelo ADDIE mediante gráficas o porcentajes.

Exportación del OVA (Funcionalidad futura): Posibilidad de descargar el OVA en formato SCORM para su uso en plataformas LMS.

Selección de Plantillas Visuales (Funcionalidad futura): Se podrán elegir estilos de diseño para personalizar la apariencia del OVA.

Carga de Recursos Multimedia (Funcionalidad futura): Integración de imágenes, audios y documentos dentro del contenido del OVA.

Generación de Certificados (Funcionalidad futura): Emisión de un certificado digital al finalizar completamente el diseño del OVA.

Modo Accesible (Funcionalidad futura): Visualización en alto contraste, fuentes ampliadas y navegación adaptada para personas con discapacidad.

Actividades Interactivas (Funcionalidad futura): Creación de recursos de tipo “arrastrar y soltar” y otras dinámicas lúdicas dentro del OVA.

Gestión de Roles y Permisos: Existen distintos niveles de usuario (invitado, registrado, docente administrador) con funcionalidades específicas.

Compatibilidad SCORM y LMS: El sistema garantiza la creación de OVA exportables y reutilizables en múltiples entornos virtuales de aprendizaje.

Usuarios del Sistema

FUNCIONALIDAD	USUARIO INVITADO	USUARIO REGISTRADO	DOCENTE ADMINISTRADOR
Registrar usuario	X		X
Iniciar sesión (Login)		X	X
Cargar la sección de una fase del módulo ADDIE		X	X
Generar formulario interactivo por etapa		X	X
Evaluar aportes del usuario en una sección de la fase del módulo ADDIE			X
Generar retroalimentación automática por IA			X
Guardar avances del usuario		X	
Visualizar progreso del usuario		X	X
Generar vista previa del OVA construido		X	X
Reiniciar proceso desde cero		X	X

Restricciones.

Solo los usuarios registrados podrán acceder al conjunto completo de funcionalidades de la plataforma InOva Design. Los usuarios invitados únicamente podrán visualizar contenido limitado y no tendrán acceso al editor de OVA ni a formularios interactivos.

El acceso a las funcionalidades de evaluación manual, retroalimentación por IA y visualización del progreso de otros usuarios estará restringido al rol Docente Administrador.

No se permitirá el registro de múltiples cuentas con el mismo correo electrónico.

Por razones de compatibilidad, los archivos multimedia que se suban al sistema (en futuras versiones) deberán cumplir con formatos y tamaños establecidos (por ejemplo, imágenes en .jpg o .png menores a 5MB).

Solo se podrá tener un proyecto activo por usuario en la versión básica del sistema; futuras versiones podrían habilitar múltiples proyectos simultáneos.

La exportación del OVA en formato SCORM estará habilitada únicamente cuando se haya completado al menos el 80% del flujo ADDIE.

El sistema requerirá conexión a internet para todas sus funciones, ya que no cuenta con un modo offline.

3. REQUISITOS FUNCIONALES.

Gestión de Usuario

El sistema debe permitir el registro de nuevos usuarios mediante un formulario de creación de cuenta con validación de correo electrónico y contraseña.

Los usuarios registrados podrán iniciar sesión con sus credenciales y acceder a sus proyectos guardados.

Los usuarios podrán reiniciar el proceso de creación de su OVA, eliminando los avances anteriores.

Navegación por el Modelo ADDIE

La plataforma debe estar estructurada en cinco módulos secuenciales: Análisis, Diseño, Desarrollo, Implementación y Evaluación.

El usuario debe poder acceder al contenido educativo y a los formularios correspondientes de cada fase.

La navegación entre módulos debe seguir un orden lógico y permitir guardar el avance por fase.

Creación de Contenido Educativo

Cada módulo debe incluir un formulario interactivo que permita registrar los aportes del usuario respecto a la fase correspondiente.

Los formularios deben almacenar los datos ingresados automáticamente al detectar interacciones.

El sistema debe generar una vista previa del OVA con base en los aportes registrados por el usuario.

Evaluación y Retroalimentación

El sistema debe permitir al Docente Administrador revisar y evaluar los aportes del usuario en cada fase.

El sistema debe mostrar la evaluación como "aprobado" o "reprobado", con observaciones opcionales.

La plataforma debe generar retroalimentación automática utilizando inteligencia artificial basada en los textos ingresados por el usuario.

Progreso y Exportación del OVA

El sistema debe mostrar el porcentaje de avance del usuario en los módulos ADDIE, mediante texto o gráficos.

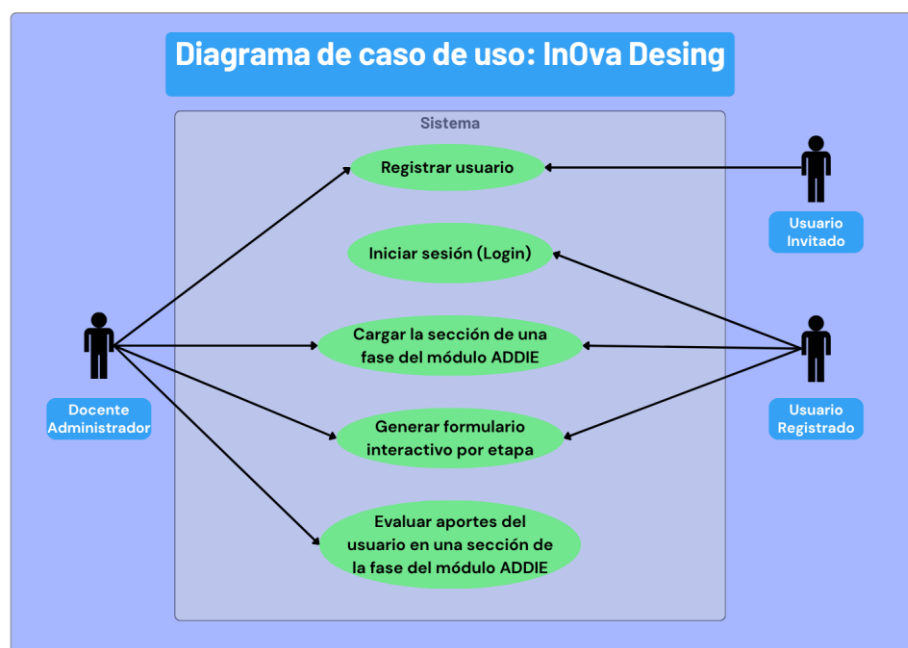
El usuario debe poder ver su progreso completo en cualquier momento desde su perfil.

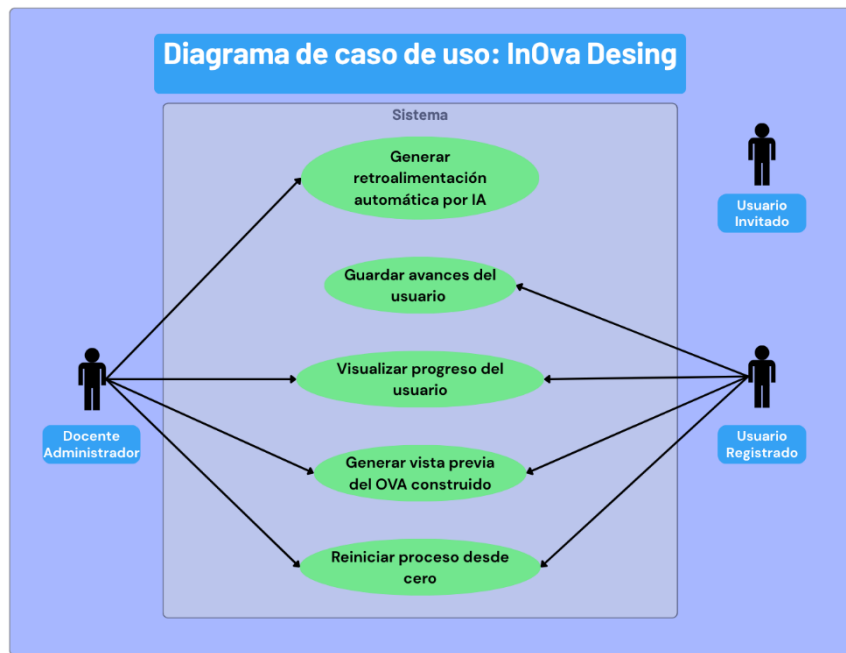
En versiones futuras, el sistema debe permitir exportar el OVA creado en formato SCORM.

También se permitirá la descarga de un certificado de finalización al completar todas las fases del modelo.

Mackup De La Interfaz De Usuario (UI)

Casos de Uso – Diagrama de casos de uso





Descripción detallada de cada caso de uso

Diagramas de Flujo de Casos de Uso 1.

	1. Registrar Usuarios
Urgencia: 5	
Esfuerzo: 5	
AOR: accede a la opción Registrarse	
RFR: redirige al formulario de registro	
ICR: Ingresa correo de registro.	Flujo: AOR, RFR, ICR, VC, GI, MME, ICR, VC, GI, MME.
VC: Valida correo.	

GI: guarda la información.	
MME: Muestra mensaje de éxito.	
ICDR: Ingresa contraseña de registro	

Caso No. 1 Registrar Usuario

ID:	CU-1		
Nombre	Registrar Usuario		
Actores	Usuario Invitado, Docente Administrador		
Objetivo	Este caso debe Permitir a los usuarios crear una cuenta.		
Urgencia	5		
Esfuerzo	5		
Pre-condiciones	El usuario debe tener acceso a internet y no estar previamente registrado con el mismo correo.		
Flujo Normal	Usuario Invitado	Docente Administrador	Sistema
	Accede a la opción “Registrarse”.		
			Redirige al formulario de registro
	Ingresa correo de registro		
			Valida correo.
			Guarda la información.
			Muestra mensaje de éxito.
	Ingresa contraseña de registro		
			Valida contraseña.
			Guarda la información.
			Muestra mensaje de éxito.
Flujo alternativo 1	Ingresa correo.		
			Valida el correo.
			Muestra mensaje de error “el correo ingresado ya está registrado”.

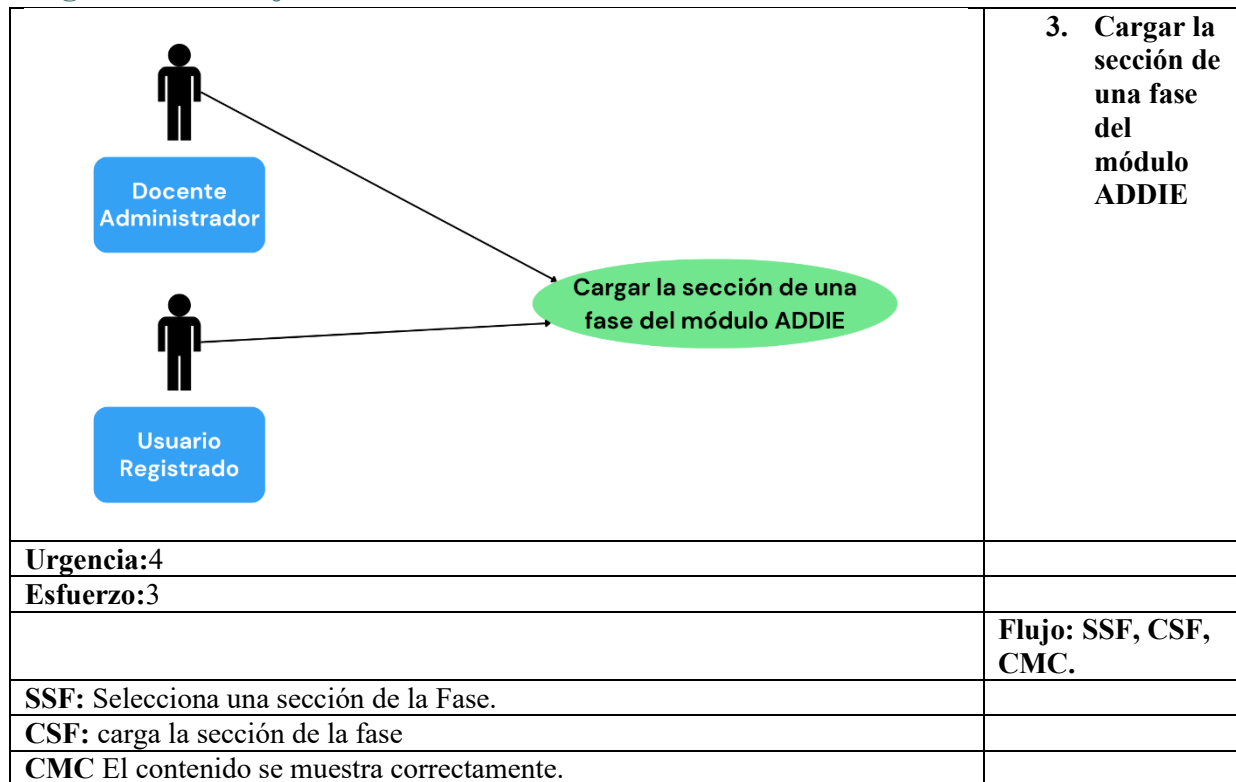
Diagramas de Flujo de Casos de Uso 2.

<pre> graph TD DA[Docente Administrador] --> IS([Iniciar sesión (Login)]) UR[Usuario Registrado] --> IS </pre>	2. Iniciar sesión (Login)
Urgencia: 5	
Esfuerzo: 4	
AOIS: Accede a la opción “Iniciar sesión”.	
RFIS: Redirige al formulario de Iniciar Sesión.	Flujo: AOIS, RFIS, ICC, VC, CIS.
ICC: Ingresa su correo y contraseña.	
VC: Valida las credenciales.	
CIS: correctas, se inicia sesión.	

Caso No. 2 Iniciar sesión (Login)

ID:	CU-2		
Nombre	Iniciar sesión (Login)		
Actores	Usuario Registrado, Docente Administrador		
Objetivo	Este caso debe permitir el acceso de usuarios registrados.		
Urgencia	5		
Esfuerzo	4		
Pre-condiciones	El usuario debe estar previamente registrado y tener acceso a internet.		
Flujo Normal	Usuario Registrado	Docente Administrador	Sistema
	Accede a la opción “Iniciar sesión”.		
			Redirige al formulario de Iniciar Sesión
	Ingresa su correo y contraseña.		
			Valida las credenciales.
			Son correctas, se inicia sesión.
Flujo alternativo 1	Ingresa su correo y contraseña.		
			Valida las credenciales.
			Contraseña incorrecta
			muestra mensaje de error “La contraseña es incorrecto”

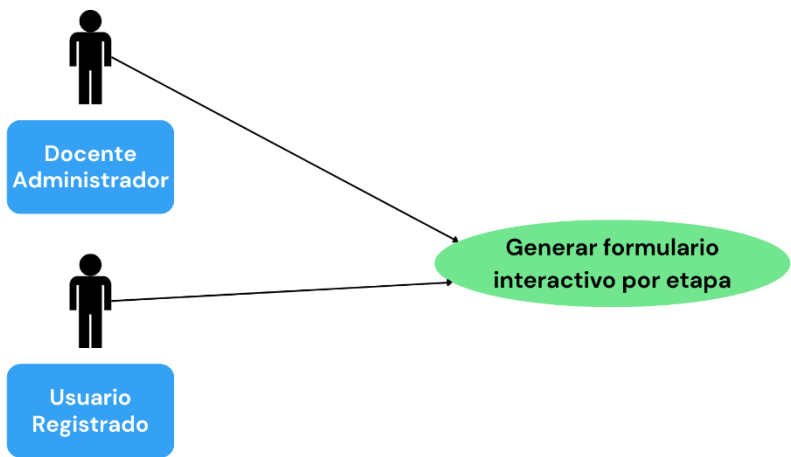
Diagramas de Flujo de Casos de Uso 3.



Caso No. 3 Cargar la sección de una fase del módulo ADDIE

ID:	CU-3		
Nombre	Cargar la sección de una fase del módulo ADDIE		
Actores	Usuario Registrado, Docente Administrador		
Objetivo	Este caso debe mostrar los contenidos educativos y formularios de una fase		
Urgencia	4		
Esfuerzo	3		
Pre-condiciones	El usuario debe haber iniciado sesión.		
Flujo Normal	Usuario Registrado	Docente Administrador	Sistema
	Selecciona una sección de la Fase.		
			carga la sección de la fase.
			El contenido se muestra correctamente.
Flujo alternativo 1			No se encuentra contenido.
			muestra mensaje no hay contenido disponible.

Diagramas de Flujo de Casos de Uso 4.

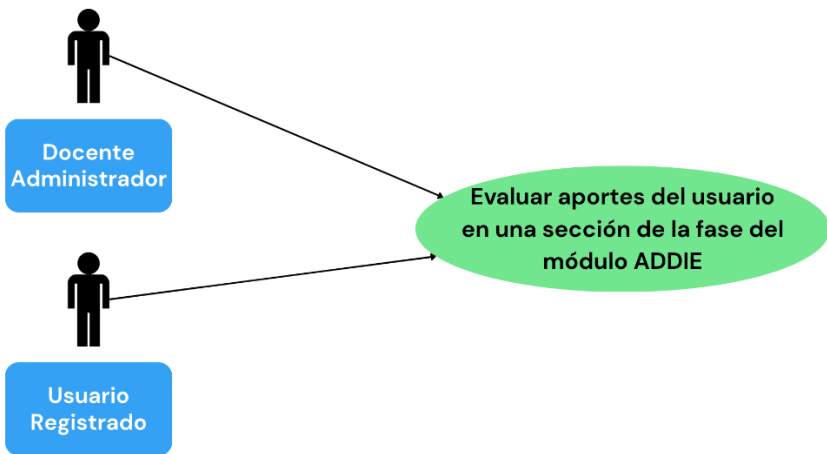
	4. Generar formulario interactivo por etapa
Urgencia:3	
Esfuerzo:3	
	Flujo: AFM, GFSE, RP, EF, GBD, RMEF.
AFM: Accede al formulario desde el módulo	
GFSE: Genera formulario según la etapa.	
RP: Responden las preguntas.	
EF: envía el formulario.	
GBD: Guarda en la base de datos.	
RMEF: Retorna mensaje “Evaluación Finalizada”	

Caso No. 4 Generar formulario interactivo por etapa

ID:	CU-4		
Nombre	Generar formulario interactivo por etapa		
Actores	Usuario Registrado, Docente Administrador		
Objetivo	Este caso debe permitir que cada módulo tenga un formulario para que el usuario aplique lo aprendido.		
Urgencia	3		
Esfuerzo	3		
Pre-condiciones	Haber completado módulos anteriores.		
Flujo Normal	Usuario Registrado	Docente Administrador	Sistema
	Accede al formulario desde el módulo		
			Genera formulario según la etapa.
	Se responden las preguntas.		
	envía el formulario.		
			Guarda en la base de datos.
			Retorna mensaje “Evaluación Finalizada”
Flujo alternativo 1	El formulario no se genera.		

		muestra mensaje de error con reintento.
--	--	---

Diagramas de Flujo de Casos de Uso 5.

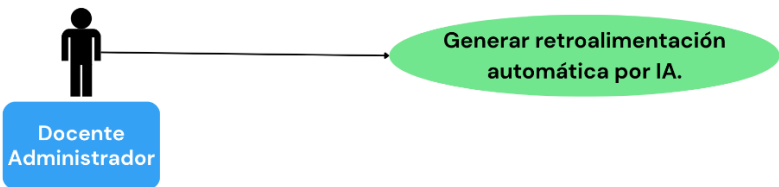
		5. Evaluar aportes del usuario en una sección de la fase del módulo ADDIE
Urgencia:3		
Esfuerzo:2		
		Flujo: SSFE, MAU, SU, RAU, MCAR, RO, COSA, GA, RMAGC.
SSFE: Selecciona la sección de la fase a evaluar.		
MAU: Muestra los aportes de los usuarios.		
SU: Selecciona usuario.		
RAU: Retorna a los aportes del usuario.		
MCAR: Marca como aprobado o Reprobado.		
RO: Realiza observaciones.		
COSA: Da clic a la opción “Subir Aportes”		
GA: Guarda los aportes		
RMAGC: Retorna mensaje “Aporte Guardado correctamente”		

Caso No. 5 Evaluar aportes del usuario en una sección de la fase del módulo ADDIE

ID:	CU-5	
Nombre	Evaluar aportes del usuario en una sección de la fase del módulo ADDIE	
Actores	Docente Administrador	
Objetivo	Este caso debe permitir validar los aportes de una fase de forma manualmente o con IA en el futuro.	
Urgencia	3	
Esfuerzo	2	
Pre-condiciones	El usuario debe haber enviado sus aportes y realizado el formulario de la fase.	
Flujo Normal	Docente Administrador	Sistema

	Selecciona la sección de la fase a evaluar.	
		Muestra los aportes de los usuarios.
	Selecciona usuario.	
		Retorna los aportes del usuario.
	Marca como aprobado o Reprobado.	
	Realiza observaciones (en caso de que dese hacerlo.)	
	Da clic a la opción “Subir Aportes”	
		Guarda los aportes
		Retorna mensaje “Aporte Guardado correctamente”
Flujo alternativo 1	omite validación manual	
		Delega a la IA.
		La IA realiza las correcciones
		La IA aprueba o reprueba
		La IA realiza observaciones
		La IA guarda los aportes

Diagramas de Flujo de Casos de Uso 6.

	6. Generar retroalimentación automática por IA
Urgencia:2	
Esfuerzo:4	
	Flujo: SF, DCOGRAI, TVTEIDF, AFTD, RRCEFS, AD, MR.
SF: Selecciona una fase.	
DCOGRAI: Da clic a la opción “Generar retroalimentación automática por IA”	
TVTEIDF: Toma los valores de todos los estudiantes que ingresaron datos en la fase.	
AFTD: Aplica fai tuning a los datos	
RRCEFS: Retorna los resultados por cada uno de los estudiantes por la fase selecciona	
AD: Acepta los datos	
MR: Muestra la retroalimentación.	

Caso No. 6 Generar retroalimentación automática por IA

ID:	CU-6	
Nombre	Generar retroalimentación automática por IA.	
Actores	Docente Administrador	
Objetivo	Este caso debe generar sugerencias de mejoras al contenido escrito por el usuario mediante inteligencia artificial.	
Urgencia	2	
Esfuerzo	4	
Pre-condiciones	El contenido debe haber sido registrado por el usuario.	
Flujo Normal	Docente Administrador	Sistema
	Selecciona una fase.	
	Da clic a la opción “Generar retroalimentación automática por IA”	
		Toma los valores de todos los estudiantes que ingresaron datos en la fase
		Aplica fai tuning a los datos

		Retorna los resultados por cada uno de los estudiantes por la fase selecciona
	Acepta los datos	
		muestra la retroalimentación.
Flujo alternativo 1		la IA no responde
		Notifica al usuario que intente más tarde.

Diagramas de Flujo de Casos de Uso 7.

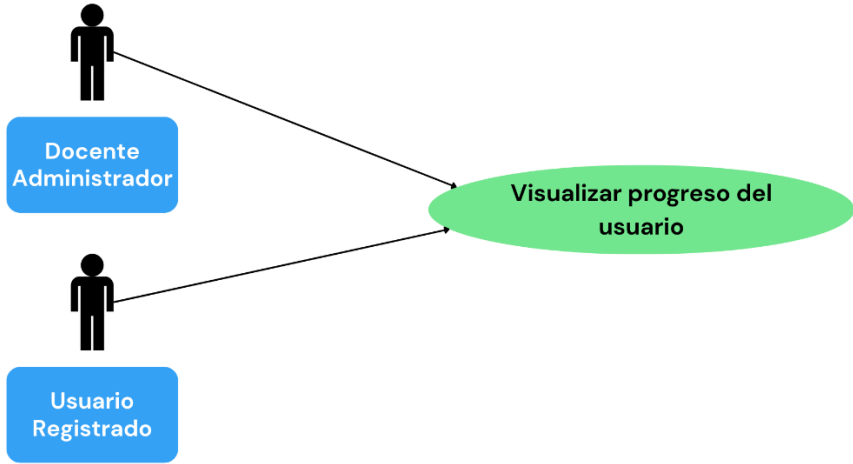
<pre> graph LR U[Usuario Registrado] --> CU((Guardar avances del usuario)) </pre>		7. Guardar avances del usuario
Urgencia:4		
Esfuerzo:2		
		Flujo: CA, DE, GAC, MMGC.
CA: completa una actividad.		
DE: detecta el evento.		
GAC: Guarda automáticamente los cambios.		
MMGC: Muestra mensaje “Guardado Correctamente”		

Caso No. 7 Guardar avances del usuario

ID:	CU-7	
Nombre	Guardar avances del usuario	
Actores	Usuario Registrado	
Objetivo	Este caso debe permitir que cada fase se almacena en el servidor por seguridad.	
Urgencia	4	
Esfuerzo	2	
Pre-condiciones	Haber realizado actividades y formularios	
Flujo Normal	Usuario Registrado	Sistema
	completa una actividad.	
		detecta el evento.
		Guarda automáticamente los cambios.

		Muestra mensaje “Guardado Correctamente”
Flujo alternativo 1		no responde
		Guarda temporalmente en un Sistema de almacenamiento local.

Diagramas de Flujo de Casos de Uso 8.

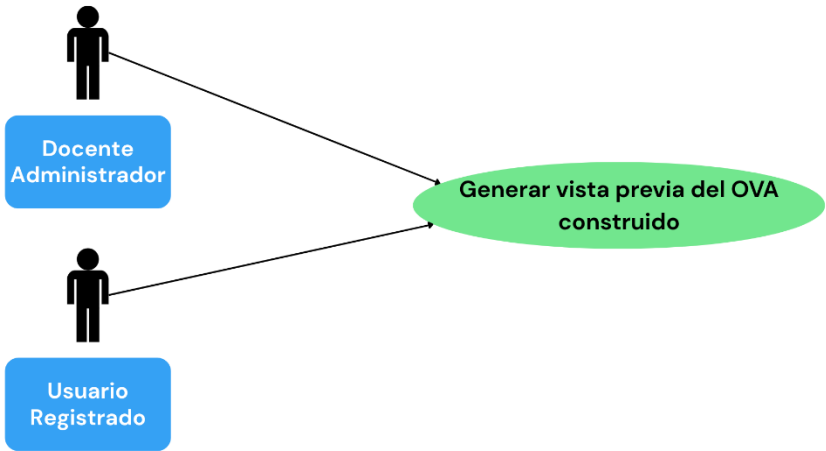
		8. Visualizar progreso del usuario
Urgencia:2		
Esfuerzo:1		
		Flujo: ODP, CDP, RPGA.
ODP: Obtener datos del progreso		
CDP: consulta los datos del progreso.		
RPGA: Retorna porcentaje y gráficos de avance.		

Caso No. 8 Visualizar progreso del usuario

ID:	CU-8		
Nombre	Visualizar progreso del usuario		
Actores	Usuario Registrado, Docente Administrador		
Objetivo	Este caso debe mostrar al usuario su progreso en los módulo ADDIE.		
Urgencia	2		
Esfuerzo	1		
Pre-condiciones	haber registros de avances guardados.		
Flujo Normal	Usuario Registrado	Docente Administrador	Sistema
	Obtener datos del progreso		
			consulta los datos del progreso.
			Retorna porcentaje y gráficos de avance.

Flujo alternativo 1		no hay avances registrados
		muestra “0% completado”.

Diagramas de Flujo de Casos de Uso 9.

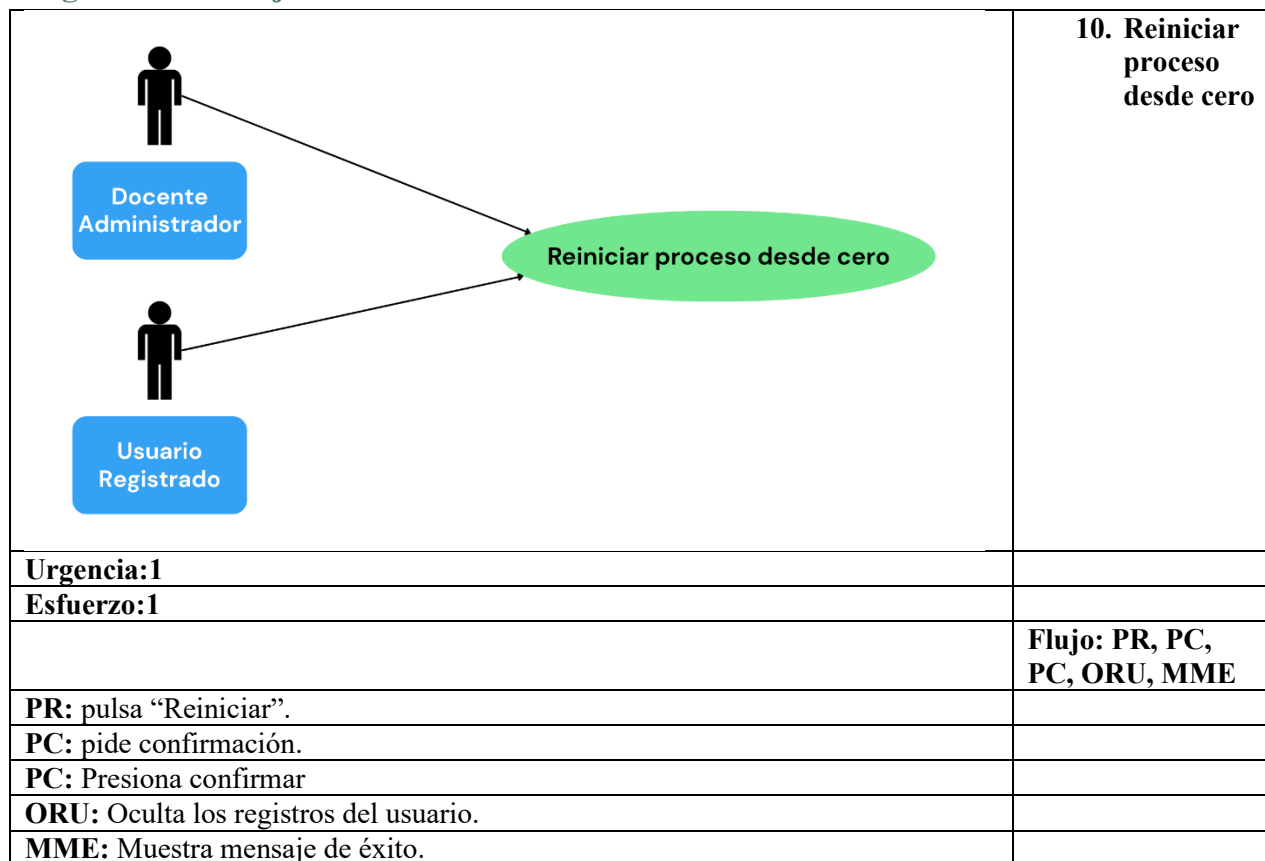
		9. Generar vista previa del OVA construido
Urgencia:1		
Esfuerzo:2		
		Flujo: PVP, RC, RI, RLI, II.
PVP: pulsa “Vista previa”.		
RC: Recolecta la informacion		
RI: Retorna a la interfaz		
RLI: Renderiza la interfaz		
II: Interactua con la Interfaz		

Caso No. 9 Generar vista previa del OVA construido

ID:	CU-9		
Nombre	Generar vista previa del OVA construido		
Actores	Usuario Registrado, Docente Administrador		
Objetivo	Este caso debe permitir ver cómo quedaría el OVA final basado en sus respuestas.		
Urgencia	1		
Esfuerzo	2		
Pre-condiciones	haber completado al menos una fase del modelo ADDIE.		
Flujo Normal	Usuario Registrado	Docente Administrador	Sistema
	pulsa “Vista previa”.		
			Recolecta la informacion.
			Retorna a la interfaz
			Renderiza la interfaz
	Interactua con la Interfaz		

Flujo alternativo 1	pulsa “Vista previa”.	
		Recolecta la informacion
		faltan datos
		notifica al usuario.

Diagramas de Flujo de Casos de Uso 10.



Caso No. 10 Reiniciar proceso desde cero

ID:	CU-18		
Nombre	Reiniciar proceso desde cero		
Actores	Usuario Registrado, Docente Administrador		
Objetivo	Este caso debe permitir eliminar todos los avances del usuario.		
Urgencia	1		
Esfuerzo	1		
Pre-condiciones	Haber completado todos los módulos		
Flujo Normal	Usuario Registrado	Docente Administrador	Sistema
	pulsa “Reiniciar”.		
			Pide confirmar.
	Presiona confirmar		
			Oculta los registros del usuario.

		Muestra mensaje de éxito.
Flo alternativo 1	pulsa “Reiniciar”.	
		Pide confirmar.
	Presiona cancelar	
		Muestra mensaje de Cancelación.

Prioridad de Requerimientos.

		Urgencia				
Impacto		1-Baja	2-Menor	3-Moderada	4-Alta	5-Obligatoria
	5-Muy alto	5	10	15	20	25
						CU-1
	4-Alto	4	8	12	16	20
			CU-6			CU-2
	3-Medio	3	6	9	12	15
				CU-4	CU-3	
	2-Bajo	2	4	6	8	10
		CU-9		CU-5	CU-7	
	1-Muy bajo	1	2	3	4	5
		CU-10	CU-8			

4.Requisitos No Funcionales.

Requisitos de Desempeño

La plataforma debe garantizar una carga rápida de cada módulo del modelo ADDIE, incluso con múltiples usuarios conectados simultáneamente.

Las operaciones de guardado automático deben ejecutarse en menos de 2 segundos después de detectar cambios en los formularios.

El sistema debe poder atender al menos 100 usuarios simultáneos sin afectar la estabilidad del servidor.

Requisitos de Seguridad

Todos los datos personales y contenidos generados por los usuarios deben ser almacenados de forma segura, utilizando cifrado en tránsito (HTTPS).

El acceso a los proyectos de OVA estará restringido mediante autenticación con correo y contraseña.

Solo el usuario creador o un docente administrador podrá modificar los aportes registrados.

Se deben implementar controles para evitar el registro de usuarios con correos duplicados o no válidos.

Requisitos de Usabilidad

La interfaz debe ser intuitiva, limpia y segmentada por fases para facilitar el flujo de trabajo del usuario.

Se debe incluir ayuda contextual o mensajes orientadores en cada sección para acompañar el diseño del OVA.

El diseño de la plataforma debe seguir principios de diseño centrado en el usuario, promoviendo una experiencia clara y sin sobrecarga cognitiva.

Requisitos de Escalabilidad

El sistema debe estar preparado para ser ampliado con nuevas funcionalidades, sin comprometer la estabilidad de la plataforma actual.

La arquitectura debe permitir la incorporación de inteligencia artificial, carga de recursos multimedia y exportación SCORM sin rediseñar la base del sistema.

El backend debe estar diseñado para escalar horizontalmente en caso de aumento de demanda de usuarios concurrentes.

Requisitos de Disponibilidad

La plataforma debe estar disponible el 95% del tiempo en condiciones normales de operación.

Se deben implementar mecanismos de respaldo automático para asegurar la recuperación ante pérdida de datos.

Compatibilidad Multiplataforma

El sistema debe ser accesible desde navegadores modernos (Chrome, Firefox, Edge, Safari).

La interfaz debe ser completamente responsiva, adaptándose a dispositivos móviles y tabletas.

Accesibilidad

El sistema debe garantizar accesibilidad básica, permitiendo navegación con teclado y compatibilidad con lectores de pantalla.

En futuras versiones, se incluirán opciones de alto contraste, cambio de tamaño de fuente y otros ajustes para personas con discapacidad visual o cognitiva.

Modelado E/R

El modelo Entidad–Relación (E/R) describe cómo los distintos actores y componentes del sistema interactúan entre sí. En nuestro caso, modelamos el proceso de creación de Objetos Virtuales de Aprendizaje (OVA) bajo el modelo ADDIE, en el cual los estudiantes construyen los contenidos y los docentes los validan y exportan en formato SCORM.

Se incluyeron entidades principales como Usuario, OVA, Fases ADDIE, Formulario–Pregunta–Respuesta, Recurso, Evaluación Docente, Avance Usuario, y las entidades propias del estándar SCORM.

El diseño busca garantizar trazabilidad pedagógica (cada fase se guarda de forma explícita), flexibilidad tecnológica (soporte de recursos multimedia) y compatibilidad con plataformas educativas (exportación a SCORM).

Enunciado

Entrevistador: ¿Qué desea que hagan los diferentes tipos de usuarios?

Cliente: Los invitados solo podrán ver información general. Los alumnos podrán crear sus OVAs siguiendo el modelo ADDIE, llenar formularios, subir recursos y visualizar su avance. Los docentes revisarán, evaluarán y aprobarán los OVAs, y son los únicos que podrán exportarlos a formato SCORM.

Entrevistador: ¿Cómo deben estructurarse los OVAs?

Cliente: Cada OVA debe estar compuesto por las 5 fases del modelo ADDIE: Análisis, Diseño, Desarrollo, Implementación y Evaluación. En cada fase, los alumnos completan formularios y suben recursos.

Entrevistador: ¿Cómo será la visualización del OVA?

Cliente: Habrá un visor interno que mostrará el OVA fase por fase. El alumno podrá ver su avance porcentual y el docente podrá revisarlo.

Entrevistador: ¿Qué pasa con los recursos y el formato SCORM?

Cliente: Los recursos subidos se integrarán dentro de cada fase. Al exportar, la plataforma generará un paquete SCORM con un manifest, organizaciones, SCOs y metadatos, de manera que pueda ser usado en plataformas como Moodle.

Entrevistador: ¿Qué restricciones deben cumplirse?

Cliente: Un OVA debe tener sus 5 fases obligatoriamente. No se puede exportar a SCORM si el OVA no está aprobado por un docente ni si está incompleto.

Restricciones del sistema

Usuarios

- Cada usuario debe tener un correo único.
- Solo los docentes pueden aprobar OVAs y exportar a SCORM.
- Los usuarios invitados no pueden crear OVAs ni subir recursos.

OVAs

- Todo OVA debe tener las 5 fases (Análisis, Diseño, Desarrollo, Implementación, Evaluación).
- Un OVA no puede publicarse sin completar todas las fases.
- Un OVA aprobado no puede modificarse (solo duplicarse o reiniciarse).

Fases

- Cada fase pertenece a un único OVA.
- El progreso de una fase no puede superar el 100%.
- No se puede eliminar una fase individual: solo el OVA completo.

Formularios y preguntas

- Los formularios son creados por la plataforma, no por los alumnos.
- Todas las preguntas obligatorias deben responderse antes de avanzar de fase.

Respuestas

- Solo el alumno creador del OVA puede responder sus formularios.
- Una respuesta no puede editarse después de ser evaluada por el docente.

Recursos

- Solo se permiten tipos de archivo autorizados (pdf, jpg, mp4).
- Cada recurso debe estar vinculado a un OVA y una fase.
- Los recursos no pueden exceder el tamaño máximo definido.

Evaluación

- Una evaluación debe estar asociada a un OVA existente.
- Solo los docentes pueden aprobar o rechazar un OVA.
- Un OVA debe tener estado aprobado antes de exportarse a SCORM.

SCORM

- Todo paquete SCORM debe contener un Manifest.

- Todo Manifest debe tener al menos una Organization.
- Cada Organization debe incluir al menos un SCO.
- Cada SCO debe estar asociado a un Resource.

Entidades - Atributos

Usuario

- idUsuario: USR001
- nombre: "Juan Pérez"
- email: "juanperez@mail.com"
- contraseña: "hash_123"
- rol: "alumno" (invitado | alumno | docente | admin)
- fechaRegistro: 2025-01-01
- estado: "activo"

OVA

- idOVA: OVA001
- titulo: "Introducción a la Programación"
- descripcion: "Curso básico basado en ADDIE"
- estado: "en_progreso" (en_progreso | completado | publicado)
- fechaCreacion: 2025-01-15
- idUsuarioCreador: USR001

FaseAnálisis

- idAnálisis: FA001
- idOVA: OVA001
- problema: "Falta de recursos digitales"
- objetivos: "Mejorar el acceso a material de apoyo"
- contexto: "Estudiantes de secundaria en zona rural"

FaseDiseno

- idDiseno: FD001
- idOVA: OVA001
- estrategias: "Uso de videos y actividades prácticas"
- objetivosEspecificos: "Aprender conceptos básicos de programación"
- estructura: "Secciones temáticas semanales"

FaseDesarrollo

- idDesarrollo: FDEV001
- idOVA: OVA001
- recursosCreados: "Video explicativo y PDF guía"
- actividades: "Ejercicios en línea"

FaseImplementacion

- idImplementacion: FIM001
- idOVA: OVA001
- planAplicacion: "Aplicar en Moodle con 20 estudiantes"
- soporte: "Guía de uso para docentes"

FaseEvaluacion

- idEvaluacionFase: FEVAL001
- idOVA: OVA001
- instrumentos: "Encuesta de satisfacción y prueba corta"
- resultadosEsperados: "80% de estudiantes comprendan los temas"

Formulario

- idFormulario: FRM001
- fase: "Análisis"
- idFase: FA001

- titulo: "Formulario de análisis inicial"
- estado: "activo"

Pregunta

- idPregunta: P001
- idFormulario: FRM001
- tipo: "texto"
- enunciado: "¿Cuál es el problema que busca resolver?"
- obligatoria: true

Respuesta

- idRespuesta: R001
- idPregunta: P001
- idUsuario: USR001
- valorTexto: "Falta de material digital en matemáticas"
- fecha: 2025-01-20

Recurso

- idRecurso: A001
- idOVA: OVA001
- tipo: "video"
- nombreRecurso: "explicacion.mp4"
- url: "/files/explicacion.mp4"
- tamaño: "25MB"
- formato: "mp4"

EvaluacionDocente

- idEvaluacion: EVAL001
- idOVA: OVA001

- idUsuario: USR001
- idDocente: DOC001
- estado: "aprobado" (pendiente | aprobado | reprobado)
- observaciones: "Bien planteado, mejorar ejemplos"
- fecha: 2025-01-22

AvanceUsuario

- idAvance: AV001
- idOVA: OVA001
- idUsuario: USR001
- fase: "Análisis"
- porcentaje: 20
- fechaActualizacion: 2025-01-22

SCORMPackage

- idPackage: PKG001
- idOVA: OVA001
- version: "2004"
- estado: "listo"
- urlZIP: "/exports/ova001_scorm.zip"

Manifest

- idManifest: M001
- idPackage: PKG001
- identifier: "manifest001"
- version: "2004"

Organization

- idOrganization: ORG001

- idManifest: M001
- title: "Introducción a la Programación"
- items: ["SCO001", "SCO002", "SCO003"]

SCO

- idSCO: SCO001
- idOrganization: ORG001
- title: "Fase de Análisis"
- launchUrl: "analisis.html"
- masteryScore: 80

Resource

- idResource: RES001
- idManifest: M001
- identifier: "res_fase1"
- href: "analisis.html"
- files: ["analisis.html", "style.css", "video.mp4"]

Metadata

- idMetadata: MD001
- ownerType: "OVA"
- ownerId: OVA001
- lomJSON: {título: "Introducción a la Programación", idioma: "es", descripcion: "OVA completo"}

Diagrama de Entidad-Relación

El diagrama E/R refleja las entidades, atributos y relaciones del sistema. Cada OVA se compone de cinco fases obligatorias (Análisis, Diseño, Desarrollo, Implementación y Evaluación), y cada fase contiene formularios, recursos y respuestas de estudiantes.

Los docentes validan los OVAs mediante evaluaciones, y los OVAs aprobados pueden exportarse en paquetes SCORM.

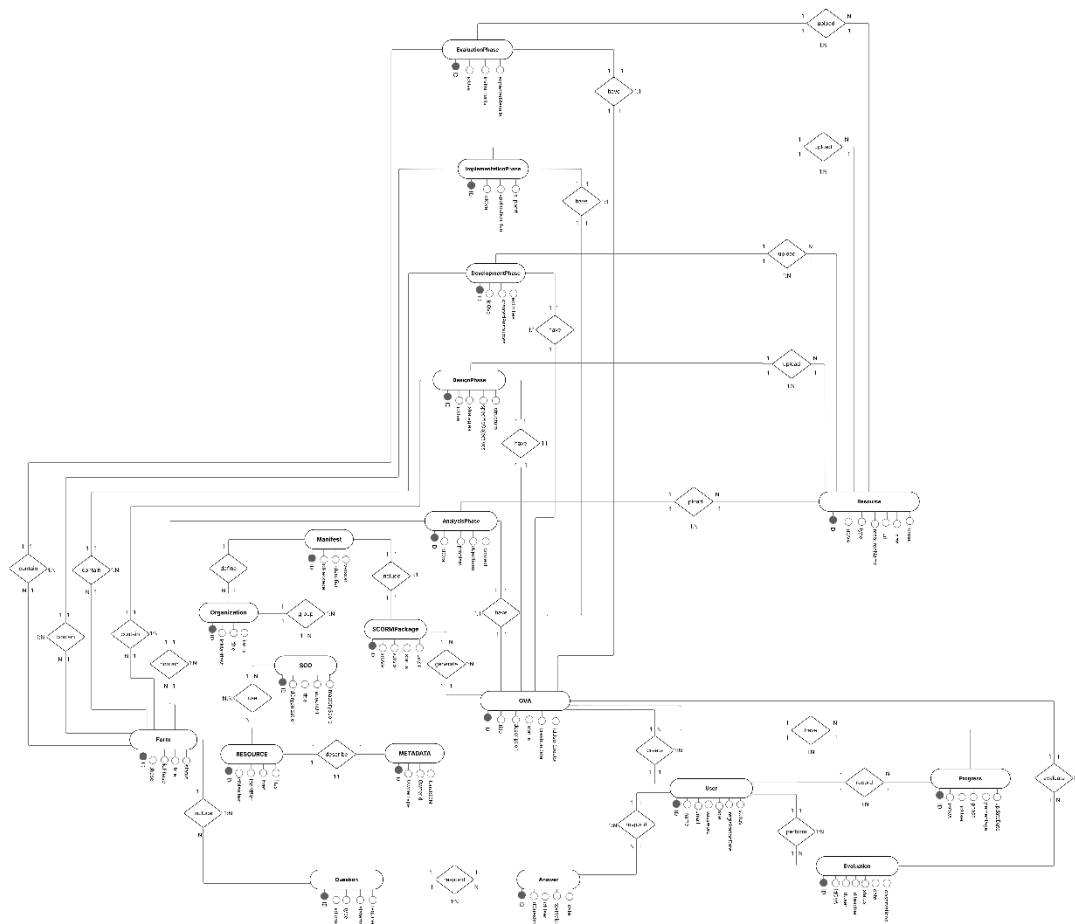


Diagrama Relacional

Script de modelo relacional

Descripción de Entidades y Relaciones

Usuario

Representa a las personas que usan la plataforma. Cada usuario tiene credenciales (email y contraseña), un rol (invitado, alumno, docente) y estado. El atributo email es único. Los usuarios pueden crear OVAs (alumnos) o evaluar y exportar OVAs (docentes).

OVA

Objeto Virtual de Aprendizaje creado por un usuario. Contiene información general (título, descripción, estado) y referencia al creador. Un OVA es la entidad raíz que agrupa las 5 fases ADDIE, sus recursos, formularios, evaluaciones y paquetes SCORM asociados.

FaseAnálisis, FaseDiseno, FaseDesarrollo, FaseImplementacion, FaseEvaluacion

Fase representa la existencia de una etapa del modelo ADDIE dentro de un OVA. tipo indica cuál es (análisis, diseño, desarrollo, implementación, evaluación). Cada fase tiene una tabla de detalle con atributos específicos propios de esa etapa (por ejemplo problema en Análisis, estrategias en Diseño). El uso de tablas de detalle permite modelar atributos exclusivos de cada fase sin inflar una sola tabla.

Formulario

Formulario asociado a una Fase. Contiene preguntas que el alumno responde. Los formularios son creados por la plataforma para guiar al alumno.

Pregunta

Elemento de Formulario que puede ser de distintos tipos (texto, selección, múltiple). Guarda el enunciado, opciones (si aplica) y si es obligatoria.

Respuesta

Respuesta de un Usuario a una Pregunta. Puede contener texto, selección o referencia a un Recurso (archivo subido) si la respuesta incluye adjunto.

Recurso (Archivo)

Material multimedia (PDF, imagen, video, audio) subido por el usuario y asociado explícitamente a una Fase. Sirve tanto para visualización en el visor interno como para empaquetado SCORM.

EvaluacionDocente

Registro de la evaluación que realiza un docente sobre un OVA (o sus fases). Contiene estado (aprobado/reprobado), observaciones y fecha. Solo docentes pueden realizar estas evaluaciones.

AvanceUsuario

Registra el progreso de un usuario en una Fase de un OVA (porcentaje y fecha). Permite calcular el avance global del OVA.

SCORMPackage, Manifest, Organization, SCO, Resource (SCORM), Metadata

Entidades relacionadas con la exportación SCORM. SCORMPackage es el ZIP final. Manifest contiene la estructura, Organization el índice/orden, SCO las unidades rastreables, Resource el descriptor de recursos dentro del manifest y Metadata la información LOM.

Usuario crea OVA (1:N) — un usuario puede crear múltiples OVAs.

OVA tiene Fase (1:1 por tipo, en total 5 fases por OVA) — cada OVA debe contener sus 5 fases.

Fase contiene Formulario (1:N) — una fase puede tener varios formularios.

Formulario incluye Pregunta (1:N) — formulario → preguntas.

Pregunta recibir Respuesta (1:N) — preguntas pueden recibir respuestas de varios usuarios.

Usuario responder Respuesta (1:N) — un usuario puede generar muchas respuestas.

Fase tiene Recurso (1:N) — recurso está ligado a la fase concreta.

OVA recibir EvaluacionDocente (1:N) — un OVA puede tener varias evaluaciones en distintos momentos.

Usuario (docente) realizar EvaluacionDocente (1:N) — docente realiza evaluaciones.

OVA tiene AvanceUsuario (1:N) y Usuario registrar AvanceUsuario (1:N) — avance registrado por usuario y fase.

OVA generar SCORMPackage (1:N) — OVA exportado produce paquetes SCORM.

SCORMPackage incluir Manifest (1:1).

Manifest definir Organization (1:N).

Organization agrupar SCO (1:N).

SCO usar Resource (SCORM) (N:M).

OVA/SCO/Resource tener Metadata (1:1).

Reglas de Integridad Referencial

Colecciones (NoSLQ)

Anexos

Diagramas Adicionales

Referencias

Etapla 2: Persistencia de Datos con Backend

Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

Diseño de la Arquitectura de Backend

Descripción de la Arquitectura Propuesta

Componentes del Backend

Diagramas de Arquitectura

Elección de la Base de Datos

Evaluación de Opciones (SQL o NoSQL)

Justificación de la Elección

Diseño de Esquema de Base de Datos

Implementación del Backend

Elección del Lenguaje de Programación

Creación de la Lógica de Negocio

Desarrollo de Endpoints y APIs

Autenticación y Autorización

Conexión a la Base de Datos

Configuración de la Conexión

Desarrollo de Operaciones CRUD

Manejo de Transacciones

Pruebas del Backend

Diseño de Casos de Prueba

Ejecución de Pruebas Unitarias y de Integración

Manejo de Errores y Excepciones

Etapla 3: Consumo de Datos y Desarrollo Frontend

Introducción

Propósito de la Etapa

Alcance de la Etapa

Definiciones y Acrónimos

Creación de la Interfaz de Usuario (UI)

Diseño de la Interfaz de Usuario (UI) con HTML y CSS

Consideraciones de Usabilidad

Maquetación Responsiva

Programación Frontend con JavaScript (JS)

Desarrollo de la Lógica del Frontend

Manejo de Eventos y Comportamientos Dinámicos

Uso de Bibliotecas y Frameworks (si aplicable)

Consumo de Datos desde el Backend

Configuración de Conexiones al Backend

Obtención y Presentación de Datos

Actualización en Tiempo Real (si aplicable)

Interacción Usuario-Interfaz

Manejo de Formularios y Validación de Datos

Implementación de Funcionalidades Interactivas

Mejoras en la Experiencia del Usuario

Pruebas y Depuración del Frontend

Diseño de Casos de Prueba de Frontend

Pruebas de Usabilidad

Depuración de Errores y Optimización del Código

Implementación de la Lógica de Negocio en el Frontend

Migración de la Lógica de Negocio desde el Backend (si necesario)

Validación de Datos y Reglas de Negocio en el Frontend

Integración con el Backend

Verificación de la Comunicación Efectiva con el Backend

Pruebas de Integración Frontend-Backend