

Codex

DOCUMENTO DE PROPUESTA DE DISEÑO DE
SOFTWARE EDUCATIVO I

CREACIÓN DE OBJETO VIRTUAL DE APRENDIZAJE – OVA PARA LA CONCEPTUALIZACIÓN DEL USO DE GIT Y GITHUB.

INTEGRANTES:

Mauro Andrés Monterroza Sevilla
Alexander Domínguez Niño
Maria Claudia Oquendo Méndez
Isacar Torreglosa Díaz
German David Rivera Rosario



TUTOR:

Alexander Toscano Ricardo

✉@kikret

🐙@atoscano

REPOSITORIO:

https://github.com/area-de-informatica/ds1_pa_codex.git

Codex

Creación de objeto virtual de aprendizaje-OVA para la conceptualización del uso de Git y Github

Autores

Mauro Andrés Monterroza Sevilla

mmonterrozasevilla@correo.unicordoba.edu.co

Alexander Domínguez Niño

adomingueznino@correo.unicordoba.edu.co

Maria Claudia Oquendo Méndez

moquendomendez@correo.unicordoba.edu.co

Isacar Torreglosa Díaz

itorreglosadiaz@correo.unicordoba.edu.co

German David Rivera Rosario

Griverarosario73@correo.unicordoba.edu.co

Tutor

Alexander Toscano Ricardo

atoscano@correo.unicordoba.edu.co

Repositorio

https://github.com/area-de-informatica/ds1_pa_codex.git

Descripción del Software

Se propone el desarrollo de un Objeto Virtual de Aprendizaje (OVA) orientado a la enseñanza de Git y GitHub, brindando a los usuarios una experiencia educativa estructurada y dinámica. Este software educativo contará con objetivos claros, contenido didáctico, actividades prácticas y evaluaciones que permitirán reforzar el aprendizaje de conceptos fundamentales como control de versiones, gestión de repositorios y colaboración en proyectos.

El OVA estará diseñado para ser sostenible, escalable y reutilizable en el tiempo, facilitando su adaptación a diferentes contextos educativos. Su estructura modular permitirá la incorporación de nuevos contenidos o actualizaciones sin afectar su funcionamiento general. Además, se priorizará una interfaz intuitiva y accesible, garantizando una experiencia de aprendizaje eficiente e interactiva.

<i>Etapas 1: Diseño de la Aplicación y Análisis de Requisitos</i>	6
Introducción	6
Propósito del Documento.....	6
Alcance del Proyecto	7
Definiciones y Acrónimos	9
Descripción General	9
Objetivos del Sistema	9
Funcionalidad General.....	9
Usuarios del Sistema.....	11
Restricciones	11
Requisitos Funcionales	11
Mockup de la Interfaz de Usuario (UI)	11
Casos de Uso.....	12
Descripción detallada de cada caso de uso	13
Diagramas de Flujo de Casos de Uso.....	13
Prioridad de Requisitos	13
Requisitos no Funcionales	13
Requisitos de Desempeño	13
Requisitos de Seguridad.....	13
Requisitos de Usabilidad.....	13
Requisitos de Escalabilidad	13
Modelado E/R	13
Diagrama de Entidad-Relación	13
Diagrama relacional.....	13
Descripción de Entidades y Relaciones	13
Reglas de Integridad	13
Anexos (si es necesario)	13
Diagramas Adicionales	13
Referencias.....	13
<i>Etapas 2: Persistencia de Datos con Backend</i>	14
Introducción	14
Propósito de la Etapa	14
Alcance de la Etapa.....	14
Definiciones y Acrónimos	14
Diseño de la Arquitectura de Backend	14
Descripción de la Arquitectura Propuesta.....	14
Componentes del Backend.....	14
Diagramas de Arquitectura	14
Elección de la Base de Datos	14
Evaluación de Opciones (SQL o NoSQL)	14
Justificación de la Elección.....	14
Diseño de Esquema de Base de Datos	14
Implementación del Backend	14
Elección del Lenguaje de Programación.....	14
Creación de la Lógica de Negocio	14
Desarrollo de Endpoints y APIs.....	14
Autenticación y Autorización	14
Conexión a la Base de Datos	14
Configuración de la Conexión	15
Desarrollo de Operaciones CRUD	15
Manejo de Transacciones.....	15
Pruebas del Backend	15

Diseño de Casos de Prueba	15
Ejecución de Pruebas Unitarias y de Integración	15
Manejo de Errores y Excepciones	15
<i>Etapas 3: Consumo de Datos y Desarrollo Frontend</i>	<i>16</i>
Introducción	16
Propósito de la Etapa	16
Alcance de la Etapa	16
Definiciones y Acrónimos	16
Creación de la Interfaz de Usuario (UI)	16
Diseño de la Interfaz de Usuario (UI) con HTML y CSS	16
Consideraciones de Usabilidad	16
Maquetación Responsiva	16
Programación Frontend con JavaScript (JS)	16
Desarrollo de la Lógica del Frontend	16
Manejo de Eventos y Comportamientos Dinámicos	16
Uso de Bibliotecas y Frameworks (si aplicable)	16
Consumo de Datos desde el Backend	16
Configuración de Conexiones al Backend	16
Obtención y Presentación de Datos	16
Actualización en Tiempo Real (si aplicable)	16
Interacción Usuario-Interfaz	16
Manejo de Formularios y Validación de Datos	16
Implementación de Funcionalidades Interactivas	17
Mejoras en la Experiencia del Usuario	17
Pruebas y Depuración del Frontend	17
Diseño de Casos de Prueba de Frontend	17
Pruebas de Usabilidad	17
Depuración de Errores y Optimización del Código	17
Implementación de la Lógica de Negocio en el Frontend	17
Migración de la Lógica de Negocio desde el Backend (si necesario)	17
Validación de Datos y Reglas de Negocio en el Frontend	17
Integración con el Backend	17
Verificación de la Comunicación Efectiva con el Backend	17
Pruebas de Integración Frontend-Backend	17

Etapas 1: Diseño de la Aplicación y Análisis de Requisitos

Introducción

Propósito del Documento

El presente documento tiene como finalidad documentar el proceso de diseño, análisis e implementación de software de tipo educativo, comercial, OVA, componente o módulo de aplicaciones. Se divide en tres etapas para facilitar el entendimiento y aplicación a gran escala en la asignatura de diseño de software.

- Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Esta etapa cumple la tarea de recoger todas las competencias desarrolladas en todas las áreas de formación del currículo de la Licenciatura en Informática con Énfasis en Medios Audiovisuales y ponerlas a prueba en el diseño y análisis de un producto educativo que se base en las teorías de aprendizaje estudiadas, articule las estrategias de enseñanza con uso de TIC y genere innovaciones en educación con productos interactivos que revelen una verdadera naturaleza educativa. Estos productos deben aprovechar las fortalezas adquiridas en las áreas de tecnología e informática, técnicas y herramientas, medios audiovisuales, programación y sistemas, para generar productos de software interactivos que permitan a los usuarios disfrutar de lo que aprenden, a su propio ritmo. Todo esto en el marco de un proceso metodológico (metodologías de desarrollo de software como MODESEC, SEMLI, etc.) que aproveche lo aprendido en la línea de gestión y lo enriquezca con elementos de la Ingeniería de Software.

- Etapa 2: Persistencia de Datos con Backend – Servidor

En la etapa 2 se continua con los lineamientos de la etapa 1, para seguir adicionando elementos de diseño e implementación de software, enfocados en el desarrollo de APIs, servidores o microservicios que permitan soportar aplicaciones cliente del software educativo; en este sentido, el curso presenta los conceptos de los sistemas de bases de datos, su diseño lógico, la organización de los sistemas manejadores de bases de datos, los lenguaje de definición de datos y el lenguaje de manipulación de datos SQL y NoSQL; de tal manera que los estudiantes adquieran las competencias para analizar, diseñar y desarrollar aplicaciones para gestionar y almacenar grandes cantidades de datos, mediante el uso de técnicas adecuadas como el diseño y modelo lógico y físico de base datos, manejo de los sistemas de gestión de bases de datos, algebra relacional, dominio del lenguaje SQL como herramienta de consulta, tecnología cliente / servidor; igualmente, se definirán los elementos necesarios

para el acceso a dichas bases de datos, como la creación del servidor API, utilizando tecnologías de vanguardia como node.js, express, Nest.js, Spring entre otros; para, finalmente converger en el despliegue de la API utilizando servicios de hospedaje en la nube, preferiblemente gratuitos. También podrá implementar servidores o API's con inteligencia artificial o en su defecto crear una nueva capa que consuma y transforme los datos obtenidos de la IA.

El desarrollo del curso se trabajará por proyectos de trabajo colaborativo que serán evaluados de múltiples maneras, teniendo en cuenta más el proceso que el resultado.

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

La etapa 3 el estudiante está en capacidad de establecer la mejor elección de herramientas de consumo de datos y técnicas en aras de lograr el mejor producto a nivel de software o hardware acorde a los requerimientos funcionales y no funcionales del problema a solucionar. En este punto el estudiante puede consumir los datos a través de un cliente que puede ser una aplicación de celular, una aplicación de escritorio, una página web, IoT (internet de las cosas) o incluso, artefactos tecnológicos.

El diseño gráfico es de los requisitos esenciales en la capa de presentación, por lo tanto, se requieren los cursos de diseño gráfico vistos previamente. Los elementos anteriores nos permiten elegir el paradigma y tecnología para desarrollar nuestras aplicaciones, teniendo en cuenta que podríamos desarrollar aplicaciones de tipo cliente.

Alcance del Proyecto

Alcance del Proyecto OVA sobre la conceptualización del uso de GIT y GITHUB

El OVA tiene como objetivo desarrollar un entorno educativo digital interactivo y estructurado que permita a los usuarios adquirir competencias teóricas y prácticas en el uso de Git como sistema de control de versiones distribuido y GitHub como plataforma de hospedaje y colaboración de proyectos. Desde una perspectiva técnica, el OVA estará construido bajo una estructura modular y flexible que permitirá su crecimiento y actualización sin afectar su funcionamiento principal, garantizando que pueda adaptarse fácilmente a nuevos contenidos y mejoras futuras.

El OVA presentará los conceptos de manera progresiva y amigable, utilizando recursos como videos breves, textos explicativos claros, infografías didácticas y simulaciones básicas de comandos. Se diseñarán actividades interactivas guiadas paso a paso, con retroalimentación inmediata directamente en el navegador, permitiendo a los estudiantes aprender de forma práctica y segura, sin requerir instalaciones adicionales. También se garantizará que el OVA sea accesible desde computadores, tabletas y teléfonos móviles, empleando un diseño responsivo que se adapte a diferentes tamaños de pantalla. Para fomentar el aprendizaje autónomo, el OVA incluirá herramientas de autoevaluación, visualización del progreso, y la posibilidad de repetir actividades las veces que se necesite. A nivel técnico, también se dejarán sentadas las bases para que, en el futuro, se puedan integrar funciones como foros de discusión y modos de estudio offline mediante tecnologías como aplicaciones web progresivas.

Finalmente, se cuidará especialmente que la plataforma cumpla principios de accesibilidad web, usando interfaces intuitivas, colores de alto contraste, y glosarios emergentes que expliquen términos técnicos, facilitando el uso del OVA incluso para estudiantes sin experiencia previa en control de versiones ni en herramientas de desarrollo.

Funcionalidades

- Ejecutar comandos básicos.
- Generar actividades.
- Generar reporte de actividades completadas.
- Mostrar progreso.
- Desplegar explicaciones breves.
- Mostrar sección de preguntas frecuentes (FAQ).
- Generar PDF.
- Generar ayudas contextuales.
- Conectar con API.
- Integrar con CMS.
- Integrar con chatbot.

Activar las opciones de accesibilidad:

- Activar modo nocturno.
- Activar Subtítulos.
- Mostrar Controles de Audio.
- Seleccionar Idioma.
- Activar lector de pantalla.

Definiciones y Acrónimos

Descripción General

Objetivos del Sistema

Brindar una experiencia educativa interactiva y progresiva que facilite la comprensión y aplicación de conceptos fundamentales relacionados con Git y GitHub, mediante contenidos multimedia, ejercicios prácticos, recursos de apoyo y herramientas de accesibilidad, promoviendo el aprendizaje autónomo y significativo en diferentes dispositivos y contextos.

Funcionalidad General

Ejecutar comandos básicos

Simula la ejecución de comandos de Git (como git init, git add, git commit, etc.) en un entorno controlado para facilitar el aprendizaje práctico.

Generar actividades

Crea ejercicios interactivos y personalizados para que el usuario aplique los conceptos vistos, incluyendo actividades de selección, emparejamiento, ordenamiento y simulaciones.

Generar reporte de actividades completadas

Produce informes automáticos sobre el progreso del usuario, mostrando qué actividades ha completado, sus resultados y tiempos de respuesta.

Mostrar progreso

Visualiza en tiempo real el avance del estudiante a través de barras de progreso o indicadores de módulos completados.

Desplegar explicaciones breves

Muestra conceptos clave en formato de textos cortos, ventanas emergentes o infografías para reforzar el aprendizaje sin sobrecargar al usuario.

Mostrar sección de preguntas frecuentes (FAQ)

Presenta una lista de dudas comunes relacionadas con Git y GitHub, junto con respuestas claras para aclarar inquietudes rápidamente.

Generar PDF

Permite exportar contenidos clave, reportes o resúmenes de aprendizaje en formato PDF, útil como material de estudio o evidencia.

Generar ayudas contextuales

Ofrece sugerencias y explicaciones adicionales dentro de las actividades, adaptadas al momento de uso o al error cometido.

Conectar con API

Facilita la comunicación con servicios externos para ampliar funcionalidades o recopilar datos, como autenticación, almacenamiento o reportes.

Integrar con CMS

Permite incrustar el OVA dentro de un Sistema de Gestión de Contenidos (como Moodle o WordPress), para su uso en entornos institucionales.

Integrar con chatbot

Incluye un asistente conversacional que responde preguntas o guía al usuario durante la navegación o resolución de actividades.

Activar modo nocturno

Cambia la interfaz a una paleta de colores oscuros para reducir la fatiga visual, especialmente en condiciones de poca luz.

Activar subtítulos

Muestra subtítulos en los contenidos audiovisuales, favoreciendo la accesibilidad para usuarios con dificultades auditivas.

Mostrar controles de audio

Proporciona controles para pausar, adelantar, retroceder o ajustar el volumen del contenido sonoro incluido.

Seleccionar idioma

Permite cambiar el idioma de la interfaz y los contenidos del OVA para adaptarse a diferentes públicos.

Activar lector de pantalla

Ofrece compatibilidad con lectores de pantalla, facilitando la navegación y comprensión del contenido a personas con discapacidad visual.

Usuarios del Sistema

Funcionalidad	Docente	Estudiante	Invitado
Ejecutar comandos básicos.	✓	✓	
Generar actividades.	✓		
Generar reporte de actividades completadas.	✓	✓	✓
Mostrar progreso.	✓	✓	✓
Desplegar explicaciones breves.		✓	✓
Mostrar sección de preguntas frecuentes (FAQ).		✓	✓
Generar PDF.		✓	✓
Generar ayudas contextuales.		✓	✓
Conectar con API.	✓		
Integrar con CMS.	✓		
Integrar con chatbot.	✓		
Activar modo nocturno.	✓	✓	✓
Activar Subtítulos.	✓	✓	✓
Mostrar Controles de Audio.	✓	✓	✓
Seleccionar Idioma.	✓	✓	✓
Activar lector de pantalla.	✓	✓	✓

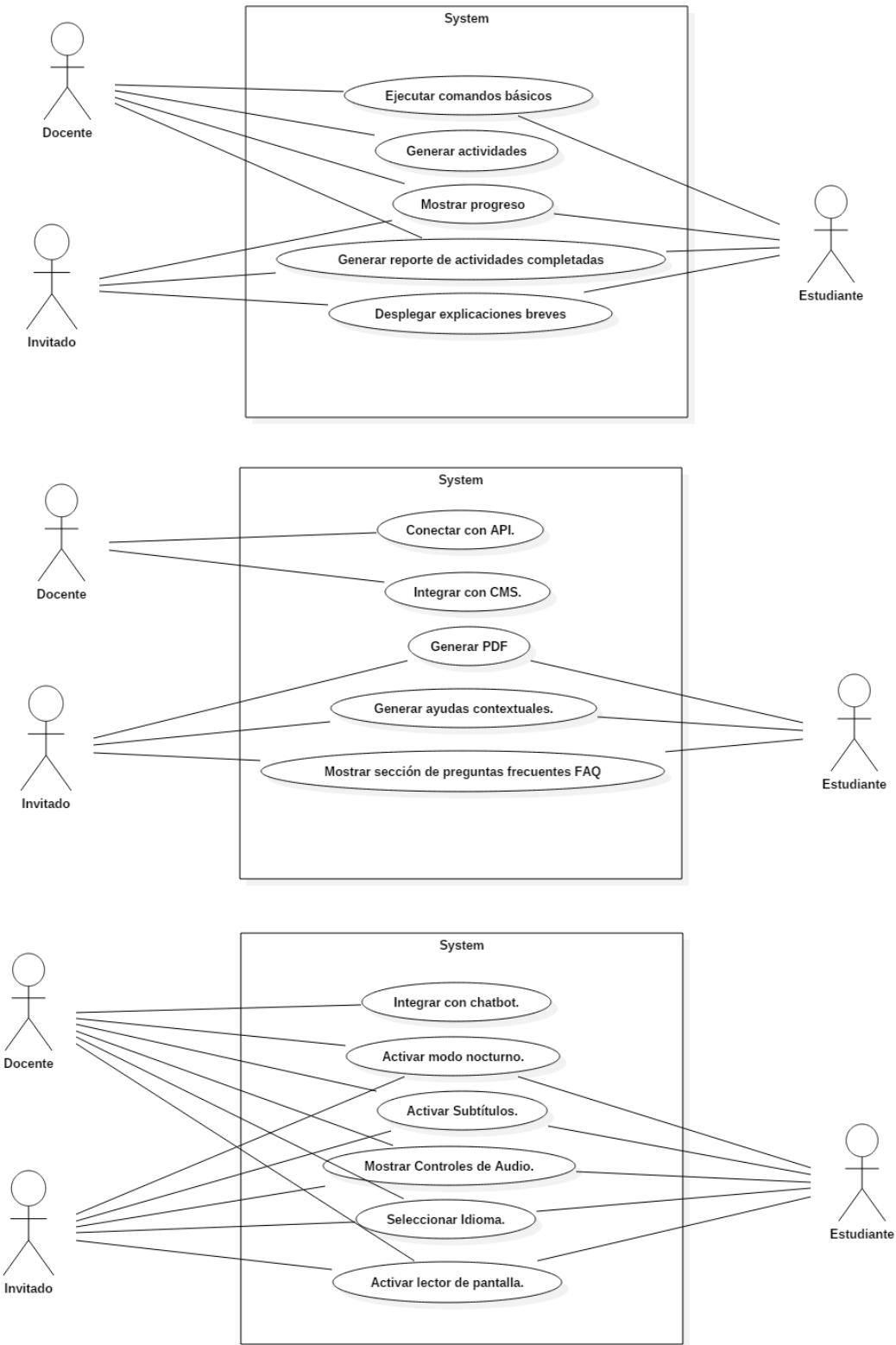
Restricciones

Requisitos Funcionales

Mockup 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

Prioridad de Requisitos

Requisitos no Funcionales

Requisitos de Desempeño

Requisitos de Seguridad

Requisitos de Usabilidad

Requisitos de Escalabilidad

Modelado E/R

Diagrama de Entidad-Relación

Diagrama relacional

Descripción de Entidades y Relaciones

Reglas de Integridad

Anexos (si es necesario)

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

Etapas 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