

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

Componente de actividades – Fase I

Autores

Dana Paola Alegría Durango (dalegriadurango22@correo.unicordoba.edu.co)

María Inés Bedoya Ortega (mbedoyaortega16@correo.unicordoba.edu.co)

Johan Mercado Fernández (jmercadofernandez82@correo.unicordoba.edu.co)

Andreina Ojeda Márquez (aojedamarquez27@correo.unicordoba.edu.co)

María José Padilla Urueta (mpadillaurueta68@correo.unicordoba.edu.co)

Tutor

Alexander Toscano Ricardo (atoscano@correo.unicordoba.edu.co)



Breve reseña

El componente de gestión de actividades de la plataforma educativa facilita la gestión de actividades interactivas como evaluaciones, cuestionarios, tareas y talleres. Permite a los docentes organizar y gestionar el proceso de enseñanza de manera eficiente, alineando las actividades con los contenidos curriculares disponibles. Ofrece herramientas para gestionar tareas, hacer seguimiento y generar informes sobre el progreso de las actividades. Se integra con xAPI, un estándar de la industria para el intercambio de datos de experiencias de aprendizaje, permitiendo registrar y rastrear las interacciones de los estudiantes en diferentes plataformas y sistemas educativos. Esta solución optimiza la gestión de actividades, promueve la flexibilidad y personalización del aprendizaje, y proporciona valiosa información para la evaluación formativa y la toma de decisiones pedagógicas informadas, beneficiando a docentes y estudiantes en el proceso de enseñanza y aprendizaje.

| | |
|---|-----------|
| ETAPA 1 DISEÑO DE LA APLICACIÓN Y ANÁLISIS DE REQUISITOS | 5 |
| INTRODUCCIÓN | 5 |
| PROPÓSITO DEL DOCUMENTO | 5 |
| ALCANCE DEL PROYECTO | 6 |
| DEFINICIONES Y ACRÓNIMOS | 6 |
| DESCRIPCIÓN GENERAL | 6 |
| OBJETIVOS DEL SISTEMA | 7 |
| FUNCIONALIDAD GENERAL | 8 |
| USUARIOS DEL SISTEMA | 8 |
| RESTRICCIONES | 8 |
| REQUISITOS FUNCIONALES | 8 |
| CASOS DE USO | 8 |
| DESCRIPCIÓN DETALLADA DE CADA CASO DE USO | 8 |
| DIAGRAMAS DE FLUJO DE CASOS DE USO | 8 |
| PRIORIDAD DE REQUISITOS | 8 |
| REQUISITOS NO FUNCIONALES | 8 |
| REQUISITOS DE DESEMPEÑO | 8 |
| REQUISITOS DE SEGURIDAD | 9 |
| REQUISITOS DE USABILIDAD | 9 |
| REQUISITOS DE ESCALABILIDAD | 9 |
| MODELADO E/R | 9 |
| DIAGRAMA DE ENTIDAD-RELACIÓN | 9 |
| DESCRIPCIÓN DE ENTIDADES Y RELACIONES | 9 |
| REGLAS DE INTEGRIDAD | 9 |
| ANEXOS (SI ES NECESARIO) | 9 |
| DIAGRAMAS ADICIONALES | 9 |
| REFERENCIAS | 9 |
| ETAPA 2: PERSISTENCIA DE DATOS CON BACKEND | 10 |
| INTRODUCCIÓN | 10 |
| PROPÓSITO DE LA ETAPA | 10 |
| ALCANCE DE LA ETAPA | 10 |

| | |
|--|-----------|
| DEFINICIONES Y ACRÓNIMOS | 10 |
| DISEÑO DE LA ARQUITECTURA DE BACKEND | 10 |
| DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA | 10 |
| COMPONENTES DEL BACKEND | 10 |
| DIAGRAMAS DE ARQUITECTURA | 10 |
| ELECCIÓN DE LA BASE DE DATOS | 10 |
| EVALUACIÓN DE OPCIONES (SQL o NoSQL) | 10 |
| JUSTIFICACIÓN DE LA ELECCIÓN | 10 |
| DISEÑO DE ESQUEMA DE BASE DE DATOS | 11 |
| IMPLEMENTACIÓN DEL BACKEND | 11 |
| ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN | 11 |
| CREACIÓN DE LA LÓGICA DE NEGOCIO | 11 |
| DESARROLLO DE ENDPOINTS Y APIs | 11 |
| AUTENTICACIÓN Y AUTORIZACIÓN | 11 |
| CONEXIÓN A LA BASE DE DATOS | 11 |
| CONFIGURACIÓN DE LA CONEXIÓN | 11 |
| DESARROLLO DE OPERACIONES CRUD | 11 |
| MANEJO DE TRANSACCIONES | 11 |
| PRUEBAS DEL BACKEND | 11 |
| DISEÑO DE CASOS DE PRUEBA | 11 |
| EJECUCIÓN DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN | 12 |
| MANEJO DE ERRORES Y EXCEPCIONES | 12 |
| ETAPA 3: CONSUMO DE DATOS Y DESARROLLO FRONTEND | 13 |
| INTRODUCCIÓN | 13 |
| PROPÓSITO DE LA ETAPA | 13 |
| ALCANCE DE LA ETAPA | 13 |
| DEFINICIONES Y ACRÓNIMOS | 13 |
| CREACIÓN DE LA INTERFAZ DE USUARIO (UI) | 13 |
| DISEÑO DE LA INTERFAZ DE USUARIO (UI) CON HTML Y CSS | 13 |
| CONSIDERACIONES DE USABILIDAD | 13 |
| MAQUETACIÓN RESPONSIVA | 13 |

| | |
|---|-----------|
| PROGRAMACIÓN FRONTEND CON JAVASCRIPT (JS) | 13 |
| DESARROLLO DE LA LÓGICA DEL FRONTEND | 13 |
| MANEJO DE EVENTOS Y COMPORTAMIENTOS DINÁMICOS | 13 |
| USO DE BIBLIOTECAS Y FRAMEWORKS (SI APLICABLE) | 14 |
| CONSUMO DE DATOS DESDE EL BACKEND | 14 |
| CONFIGURACIÓN DE CONEXIONES AL BACKEND | 14 |
| OBTENCIÓN Y PRESENTACIÓN DE DATOS | 14 |
| ACTUALIZACIÓN EN TIEMPO REAL (SI APLICABLE) | 14 |
| INTERACCIÓN USUARIO-INTERFAZ | 14 |
| MANEJO DE FORMULARIOS Y VALIDACIÓN DE DATOS | 14 |
| IMPLEMENTACIÓN DE FUNCIONALIDADES INTERACTIVAS | 14 |
| MEJORAS EN LA EXPERIENCIA DEL USUARIO | 14 |
| PRUEBAS Y DEPURACIÓN DEL FRONTEND | 14 |
| DISEÑO DE CASOS DE PRUEBA DE FRONTEND | 14 |
| PRUEBAS DE USABILIDAD | 14 |
| DEPURACIÓN DE ERRORES Y OPTIMIZACIÓN DEL CÓDIGO | 15 |
| IMPLEMENTACIÓN DE LA LÓGICA DE NEGOCIO EN EL FRONTEND | 15 |
| MIGRACIÓN DE LA LÓGICA DE NEGOCIO DESDE EL BACKEND (SI NECESARIO) | 15 |
| VALIDACIÓN DE DATOS Y REGLAS DE NEGOCIO EN EL FRONTEND | 15 |
| INTEGRACIÓN CON EL BACKEND | 15 |
| VERIFICACIÓN DE LA COMUNICACIÓN EFECTIVA CON EL BACKEND | 15 |
| PRUEBAS DE INTEGRACIÓN FRONTEND-BACKEND | 15 |

Etapa 1 Diseño de la Aplicación y Análisis de Requisitos

Introducción

En el constante avance de la tecnología educativa, la creación de herramientas innovadoras para la gestión de actividades se convierte en una prioridad para mejorar la experiencia de aprendizaje. En este contexto, surge la propuesta de desarrollar un Gestor de Actividades, empleando tecnologías de vanguardia como H5P y xAPI. Este proyecto representa un hito significativo en la evolución de la enseñanza digital, al integrar dos pilares fundamentales: la versatilidad de H5P, que permite la creación de actividades interactivas y multimedia de manera ágil y accesible; y la potencia de xAPI, un estándar de seguimiento de experiencia de aprendizaje que proporciona un profundo conocimiento sobre el progreso y rendimiento del estudiante. Este documento tiene como objetivo presentar la concepción y desarrollo de este Gestor de Actividades, destacando su capacidad para revolucionar la forma en que se diseñan, gestionan y evalúan las actividades educativas en entornos digitales. Al ofrecer una solución integral basada en tecnologías punteras, se aspira a potenciar la interactividad, personalización y seguimiento del aprendizaje, brindando a educadores y estudiantes una herramienta poderosa para alcanzar sus objetivos educativos con mayor eficiencia y efectividad.

Propósito del Documento

El punto de este componente es crear un espacio específico dentro de la plataforma educativa para la creación, gestión y asignación de diversas actividades, como evaluaciones, quiz, tareas y talleres. Esto permitirá a los profesores organizar y gestionar de forma eficiente el proceso de aprendizaje, alineando las actividades con los contenidos disponibles en la plataforma.

- 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 y 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 y programación y sistemas, para generar productos 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 continúa 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

El proyecto busca desarrollar un sistema integral para la creación, organización y evaluación de actividades en una plataforma educativa, se usarán herramientas como H5P y xAPI. Abarca la creación de diversos tipos de actividades, su planificación y organización en el calendario académico, la asignación a estudiantes o grupos, el seguimiento del progreso y la entrega de trabajos, la comunicación entre profesores y estudiantes, y la generación de informes sobre el rendimiento y el logro de objetivos.

- Cargar actividad H5P
- Editar actividad H5p
- Descargar actividad H5P
- Buscar 1 actividad H5P
- Buscar actividades H5P, todas las existentes
- Buscar actividades H5P por usuario
- Buscar por tema actividad H5P
- Listar actividad H5P
- Listar actividades H5P
- Visualizar actividad H5P

Funcionalidades xAPI

- Registro de creación de actividad

con H5P.

- Registro de actualización de actividades de H5P (CRUD)
- Registro del progreso de usuarios por actividad. (“en progreso” y “finalizado”)
- Registro de interacción con la actividad.

Funcionalidades Futuras:

- Registro de actividades descargadas por usuarios.
- Visualización del estado de finalización de las actividades.
- Formatos y configuraciones variadas.
- Funcionalidades de xAPI

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).

OVA: Objetos Virtuales de Aprendizaje (Virtual Learning Objects)

H5P: Paquete HTML5, creación de contenido educativo.

xApi: interfaz de programación de aplicaciones.

DEFINICIONES

- **xAPI:** xAPI es una especificación de eLearning que permite recopilar datos sobre la amplia gama de experiencias que una persona tiene dentro de las actividades de capacitación en línea y fuera de línea. El uso de xAPI de un formato compartido tanto para la recepción como para el envío de datos hace que xAPI sea una herramienta ideal para compartir el aprendizaje entre múltiples sistemas. Esta API captura datos en un formato consistente sobre las actividades de una persona o grupo de muchas tecnologías. Sistemas muy diferentes pueden comunicarse de forma segura capturando y compartiendo este flujo de actividades utilizando el vocabulario simple de xAPI. (Viñuela, n.d.)
- **H5P:** H5P es una abreviatura para paquete HTML5. Les permite a los educadores crear contenido, como por ejemplo presentaciones, exámenes y videos interactivos. El contenido H5P puede ser creado en el Banco de contenido, o en el sitio h5p.com o con la app LUMI para escritorio, y añadido a su curso como una Actividad H5P o incrustado dentro de cualquier otra actividad o recurso. H5P se puede usar en una amplia variedad de contextos educativos, desde la educación infantil hasta la formación empresarial.
- **Actividad:** Una actividad es un conjunto de acciones o tareas que se realizan con un propósito específico. Puede ser una acción física o mental, y puede ser individual o colectiva. Las actividades se pueden clasificar en diferentes categorías según diferentes criterios, como su objetivo, su duración o su nivel de complejidad. Una **actividad educativa** es cualquier tarea o experiencia diseñada para facilitar el aprendizaje de los estudiantes. Estas actividades pueden ser formales o informales, y pueden realizarse dentro o fuera del aula. En este caso, hablamos de actividades en H5P.
- **Actividad H5P:** En el contexto de H5P, una actividad es un tipo de contenido interactivo que se puede crear utilizando la herramienta de autoría H5P. Las actividades de H5P están diseñadas para ser atractivas y efectivas para estudiantes de todas las edades y se pueden usar en una variedad de entornos educativos.
- **Nombre:** Un identificador único para la actividad, generalmente un título breve y descriptivo.
- **Archivo:** El archivo digital que contiene el contenido y el código de la actividad. Este archivo suele estar en formato ".h5p".
- **Descripción:** Una breve descripción general de la actividad, que describe su propósito, objetivos de aprendizaje y público objetivo.
- **Enunciado:** La pregunta o instrucción principal que guía la interacción del alumno con la actividad. Debe ser claro, conciso y relevante para los objetivos de aprendizaje.

Descripción General

Objetivos del Sistema

El objetivo general de este componente, es la creación y organización de actividades para una plataforma educativa, optimizando la gestión del proceso de aprendizaje por parte de los profesores. Esto se logrará a través de la creación, organización y evaluación eficiente de actividades personalizadas y alineadas con los objetivos de aprendizaje y los contenidos disponibles en la plataforma. El módulo busca mejorar la eficiencia y productividad de los profesores, aumentar la flexibilidad del aprendizaje, mejorar la evaluación, promover la participación de los estudiantes y optimizar su aprendizaje.

| Funcionalidad | Administrador | Docente investigador | Docente invitado | Alumno | Invitado |
|--|---------------|----------------------|------------------|--------|----------|
| Cargar actividad | | ✓ | ✓ | | |
| Editar actividad | | ✓ | ✓ | | |
| Descargar actividad | | ✓ | ✓ | ✓ | ✓ |
| Buscar actividad (es) | | ✓ | ✓ | ✓ | ✓ |
| Buscar por tema | | ✓ | ✓ | ✓ | ✓ |
| Listar actividad | | ✓ | ✓ | ✓ | ✓ |
| Visualizar actividad | | ✓ | ✓ | ✓ | ✓ |
| Historial de creación de actividad | ✓ | ✓ | ✓ | | |
| Registro de actualización de actividades | ✓ | ✓ | ✓ | | |
| Registro de progreso de usuarios por actividad | ✓ | ✓ | ✓ | | |
| Registro de interacción por actividad | ✓ | ✓ | ✓ | | |

Conceptos de las entidades

Funcionalidad General

Usuarios del Sistema

Restricciones

Requisitos Funcionales

Casos de Uso

Descripción detallada de cada caso de uso

Diagramas de Flujo de Casos de Uso

Diagramas de Secuencia

Prioridad de Requisitos

Requisitos No Funcionales

Requisitos de Desempeño

Requisitos de Seguridad

Requisitos de Usabilidad

Requisitos de Escalabilidad

Modelado E/R

Caracterización de los datos

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