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

Componente Gestión de Ova – Fase I

Keimer Enrique Muñoz Mora

Gloria Elena Cordero Almario

Luis Carlos Suárez Bravo

Tutor: Alexander Toscano

Breve reseña

En muchos entornos educativos, los recursos educativos están dispersos en diferentes plataformas y formatos, lo que dificulta su acceso y gestión. El componente propuesto es un sistema integral diseñado para la recopilación, gestión y almacenamiento de Objetos Virtuales de Aprendizaje (OVA), abarcando una amplia variedad de formatos de contenidos. Ofrecerá una interfaz de usuario intuitiva y accesible desde múltiples dispositivos, junto con un sistema de búsqueda avanzada para facilitar la localización de recursos relevantes. Además, permitirá la personalización según las necesidades de los usuarios y organizaciones educativas, integrándose de manera fluida con sistemas de gestión del aprendizaje existentes y priorizando la seguridad de los datos mediante medidas como la encriptación y el control de acceso. El componente propuesto proporcionará una solución integral y escalable para la recopilación y gestión de OVA, ofreciendo una experiencia de aprendizaje enriquecida y adaptada a las necesidades individuales y organizativas.

Etapa 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 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 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 trabajara 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 componente propuesto es un sistema integral diseñado específicamente para la gestión eficiente de Objetos Virtuales de Aprendizaje (OVA). Este sistema permite administrar los OVA teniendo en cuenta sus características principales, como la página de inicio, los contenidos y la evaluación. Además, incluye una interfaz intuitiva con búsqueda avanzada, segura y se integra fácilmente con los sistemas de gestión del aprendizaje existente. En el siguiente apartado, se detallarán las características de la presente versión y se sugerirán algunas para futuras actualizaciones.

- Subir un OVA.
- Actualizar un OVA.
- Eliminar un OVA.
- Buscar un OVA.
- Exportar OVA.
- Valorar OVA.

Funcionalidades Futuras:

- Interoperabilidad con Herramientas de Autoría
- Importar OVA externo.
- Recomendación Personalizada de Contenidos.
- Recolección de datos XAPI.
- Análisis de datos de Uso con XAPI.

Definiciones y Acrónimos

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

Es un conjunto de reglas y protocolos, permite que diferentes aplicaciones se comuniquen entre sí y compartan datos o funcionalidades.

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

Es un software que facilita la creación, manipulación y administración de bases de datos, permitiendo almacenar, organizar y recuperar información de manera eficiente.

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

Es un lenguaje estándar utilizado para interactuar con bases de datos relacionales. Permite realizar consultas, inserciones, actualizaciones y eliminaciones de datos, así como la creación y modificación de esquemas de base de datos.

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

Es un protocolo de comunicación utilizado para la transferencia de información en la World Wide Web. Define cómo se transmiten los mensajes y cómo navegadores y servidores web interactúan para solicitar y enviar recursos, páginas web, imágenes y otros archivos.

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

Es un estilo arquitectónico para el diseño de sistemas de software distribuido, basado en la representación de recursos a través de URLs y la manipulación de dichos recursos utilizando operaciones estándar de HTTP, como GET, POST, PUT y DELETE.

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

Es un formato de intercambio de datos ligero y legible por humanos, basado en la sintaxis de los objetos de JavaScript. Se utiliza comúnmente para transmitir datos estructurados entre un servidor y un cliente en aplicaciones web y es independiente del lenguaje de programación.

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

Es un estándar abierto (RFC 7519) que define un formato compacto y seguro para la transferencia de información entre dos partes, generalmente utilizado en autenticación y autorización en aplicaciones web. Los JWT están representados como cadenas JSON y son firmados digitalmente para verificar su autenticidad y asegurar la integridad de los datos.

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

Es un conjunto de operaciones básicas utilizadas en sistemas de gestión de bases de datos y aplicaciones web para manipular datos. Estas operaciones son fundamentales para el mantenimiento de la información: crear nuevos registros (Create), leer datos existentes (Read), actualizar registros existentes (Update) y eliminar registros (Delete).

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

Es una técnica de programación que permite convertir datos entre el modelo de objetos utilizado en un lenguaje de programación orientado a objetos y el modelo relacional utilizado en una base de datos relacional.

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

Es un patrón de diseño de software que divide una aplicación en tres componentes principales: el Modelo, que representa los datos y la lógica de la aplicación; la Vista, que se encarga de mostrar la interfaz de usuario y recibir las entradas del usuario; y el Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando las interacciones del usuario y actualizando el Modelo en consecuencia.

API RESTful: API que sigue los principios de REST.

API que sigue los principios de REST, utilizando URLs para identificar recursos y métodos HTTP estándar para manipularlos. Es flexible y fácilmente entendida por los desarrolladores.

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

Es un enfoque de desarrollo que automatiza los procesos de construcción, pruebas y despliegue de código para entregar cambios de manera rápida y segura. La Integración Continua se concentra en fusionar y probar código frecuentemente, mientras que la Entrega Continua automatiza el despliegue regular y confiable del código en entornos de producción.

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

Es un modelo de distribución de software donde las aplicaciones son alojadas por un proveedor de servicios y accesibles a través de internet, generalmente mediante suscripciones.

SSL/TLS: Capa de sockets seguros/Seguridad de la Capa de Transporte (Secure Sockets

Layer/Transport Layer Security).

Son protocolos de seguridad que cifran las comunicaciones en internet para proteger la privacidad y la integridad de los datos transmitidos entre clientes y servidores.

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

Es el lenguaje estándar utilizado para crear páginas web, permitiendo la estructuración y presentación de contenido en línea.

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

Es un lenguaje utilizado para definir el estilo y la presentación de documentos HTML, permitiendo controlar el diseño, la tipografía, los colores y otros aspectos visuales de una página web.

JS: JavaScript.

Es un lenguaje de programación usado para agregar interactividad y funcionalidad dinámica a páginas web.

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

Es una interfaz de programación que representa la estructura de un documento HTML como un árbol de objetos, permitiendo a los programas acceder y manipular el contenido, la estructura y el estilo de una página web de manera dinámic

UI: Interfaz de Usuario (User Interface).

Es el medio por el que los usuarios interactúan con un dispositivo, aplicación o sistema informático, permitiéndoles actuar y recibir retroalimentación de forma intuitiva y eficiente.

UX: Experiencia del Usuario (User Experience).

Es la impresión general que tiene un usuario al interactuar con un producto o servicio, incluyendo aspectos como la usabilidad, la accesibilidad y la satisfacción.

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

Es una aplicación web que carga una sola página HTML y actualiza el contenido de manera dinámica utilizando JavaScript, proporcionando una experiencia de usuario fluida y rápida.

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

Es una técnica de desarrollo web que permite realizar solicitudes al servidor de forma asíncrona, sin necesidad de recargar la página completa, lo que mejora la velocidad y la interactividad de las aplicaciones web.

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

Es una plataforma que facilita la creación, edición, gestión y publicación de contenido digital, como páginas web, blogs o tiendas en línea, sin necesidad de conocimientos técnicos avanzados.

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

Es una red de servidores distribuidos geográficamente que almacenan copias de contenido web estático, como imágenes, archivos de estilo y scripts, para entregarlo a los usuarios de manera más rápida y eficiente.

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

Es el proceso de mejorar la visibilidad y la clasificación de un sitio web en los resultados de búsqueda orgánica, mediante técnicas como la optimización del contenido, la estructura del sitio y la construcción de enlaces.

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

Es un software que proporciona herramientas integradas para escribir, depurar y compilar código de programación en un solo lugar, facilitando el desarrollo de software.

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

Es una interfaz de usuario que permite interactuar con un sistema informático mediante comandos de texto, en lugar de utilizar una interfaz gráfica.

PWA: Aplicación Web Progresiva (Progressive Web App).

Es una aplicación web que utiliza tecnologías modernas para proporcionar una experiencia similar a la de una aplicación nativa en dispositivos móviles, incluyendo funciones como el acceso offline, las notificaciones push y la instalación en la pantalla de inicio.

OVA: Objetos Virtuales de Aprendizaje (Virtual Learning Objects)

Un OVA (Objeto Virtual de Aprendizaje), es un recurso educativo interactivo en línea, que facilita el aprendizaje. Está compuesto por elementos como texto, imágenes y videos. Puede distribuirse con documentos HTML para fácil acceso en navegadores web estándar.

XAPI: Experiencia de Aprendizaje Electrónico (Experience API)

Es un estándar de tecnología de aprendizaje electrónico que permite el seguimiento y la recopilación de datos sobre las actividades en línea.

GESTIÓN DEL OVA: (OVA Management)

La gestión del OVA se refiere al conjunto de procesos y estrategias utilizados para administrar y Evaluación organizar los recursos educativos digitales conocidos como Objetos Virtuales de Aprendizaje. Esto incluye la creación, distribución, actualización, y seguimiento de los OVAs para garantizar su efectividad en el proceso de enseñanza y aprendizaje.

VALORACIÓN DE OVA: (OVA Assessment)

La valoración de un OVA consiste en asignar una puntuación entre 1 y 5 para evaluar su calidad, efectividad y relevancia en el contexto educativo. Cuanto mayor sea la valoración, mayor será la probabilidad de que el OVA sea recomendado en un entorno de sugerencias.

Descripción General

Objetivos del Sistema

El sistema tiene como objetivos principales facilitar la recopilación, gestión y almacenamiento de Objetos Virtuales de Aprendizaje (OVA) desde diversas fuentes y formatos, así como proporcionar una interfaz de usuario intuitiva y accesible para una fácil navegación y recuperación de recursos educativos. Además, busca permitir la personalización y adaptabilidad de los OVA según las necesidades específicas de los usuarios y las organizaciones educativas, integrarse fluidamente con sistemas de gestión del aprendizaje existentes y priorizar la seguridad y privacidad de los datos del usuario y los contenidos educativos almacenados en el sistema.

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

Etapa 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

Etapa 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