

Informe de arquitectura.

Análisis de contexto y requisitos.

Equipo 4

VICTOR MANUEL RANGEL MEJIA

ZAIRA CITLALI GARCÍA SILVA

MARIANA HERNANDEZ DIMAS

a. Comprensión del entorno empresarial.

Crear exámenes para profesores de matemáticas es altamente engorroso y agotador, ya que implica escribir fórmulas a mano o introducirlas meticulosamente en un ordenador, haciendo el proceso innecesariamente laborioso e improductivo.

MindFlex IT observó una oportunidad en este ámbito, que era disminuir la duración, la complejidad y el esfuerzo que los profesores deben soportar al elaborar evaluaciones matemáticas para los alumnos.

Si ponemos este concepto en práctica, usamos un software propio de reconocimiento para ecuaciones matemáticas y guardaremos sus formas L^AT_EX en una base de datos no relacional convertida a JSON, permitiéndonos crear exámenes a partir de esas muestras matemáticas almacenadas allí.

Nuestro objetivo incluye extraer ecuaciones de imágenes o PDFs, modificarlas de antemano, y también después de enviarlas al repositorio.

b. Identificación de las necesidades tecnológicas.

Nuestro sistema requiere la combinación de varias herramientas avanzadas para cumplir con los requisitos especificados.

Creación autónoma de evaluaciones en formato PDF, permitiendo a los educadores determinar los temas, el nivel de dificultad y la cantidad de preguntas.

Las tecnologías OCR para descifrar expresiones numéricas en una instantánea o PDF, obteniendo la representación L^AT_EX, canalizando a un almacenamiento inmediato y rápido dentro de una base de datos NoSQL resultan valiosas.

Texto: Gestionar a los usuarios y sus tareas dando ciertos privilegios a los líderes y educadores.

Crear un registro para llevar un seguimiento de los exámenes anteriores y asegurar la fácil disponibilidad de materiales de revisión.

Necesitamos un sistema que utilice OCR, que pueda manejar grandes cantidades de datos y que cree rápidamente PDFs para cada informe. Sigue siendo crucial establecer un portal intuitivo, con una navegación sencilla, que permita a los usuarios manipular el sistema con una formación mínima.

c. Análisis de Sistemas Legados

El desarrollo de la aplicación de escritorio inicialmente se concibió como una app de web que usará un modelo GPT-2 (Generative Pre-trained Transformer) , cual tenía un buen enfoque pero realmente no era nada realista ya que en producción se tenían un gran costo computacional , deficiencias en la extracción de fórmulas matemáticas en imágenes que contenían más de una fórmula matemática.

Por ello se cambió al enfoque de una aplicación de escritorio cual puedes hacer captura de pantalla a la sección del pdf que contiene las fórmulas matemáticas necesarias para el profesor.

El cambio fue muy radical ya que pasamos de un GPT a un transformador de visión (ViT) cual es un transformador de red neuronal que se utiliza específicamente para tareas de procesamiento visual, como el reconocimiento de imágenes, conjunto a una ResNet backbone para el math de las fórmulas y su Látex, la ResNet son redes troncales aquellas que proporcionan un enlace principal de comunicación entre redes más pequeñas o entre redes de diferentes tipos, finalmente con un Transformer cual es el aprendizaje de las neuronas de la ResNet.

d. Evaluación de Infraestructura Actual

Caso de prueba	Descripción	Precondiciones	Criterios de aceptación	Resultado
CP-1	Login con usuario registrado	<ul style="list-style-type: none">• Usuario registrado	El usuario accede a la plataforma con su correo y contraseña de forma exitosa	✓
CP-2	Login con Google	<ul style="list-style-type: none">• Tener correo de Google dado de alta en la plataforma	El usuario accede a la plataforma con su correo de forma exitosa	✓
CP-3	Registro de un usuario	<ul style="list-style-type: none">• El usuario no tiene que estar registrado• El servidor debe ser funcional	El usuario se registra de forma exitosa	✓

CP-4	Cruds de usuarios	<ul style="list-style-type: none"> • Iniciar sesión con el usuario correspondiente 	Se llevan a cabo los cambios de manera correcta	✓
CP-5	Creación de exámenes	<ul style="list-style-type: none"> • Iniciar sesión con el usuario correspondiente 	El profesor puede visualizar el PDF generado	✓
CP-6	Escaneo de imágenes	<ul style="list-style-type: none"> • Iniciar sesión con el usuario correspondiente 	El usuario puede visualizar las fórmulas escaneadas	✓
CP-7	Edición de fórmulas matemáticas	<ul style="list-style-type: none"> • Iniciar sesión con el usuario correspondiente • Tener fórmula que editar 	El usuario puede visualizar los cambios de forma exitosa	✓
CP-8	Historial de Exámenes	<ul style="list-style-type: none"> • Iniciar sesión con el usuario correspondiente 	El usuario puede consultar los exámenes de forma exitosa	✓
CP-9	Visualización de calificaciones	<ul style="list-style-type: none"> • Iniciar sesión con el usuario correspondiente • Seleccionar el alumno correspondiente 	El profesor puede visualizar de forma correcta las calificaciones del alumno	✓
CP-10	Descarga de Exámenes	<ul style="list-style-type: none"> • Iniciar sesión con el usuario correspondiente • Seleccionar que si lo desea guardar 	El profesor puede consultar el PDF descargado	✓
CP-11	Guardado de Exámenes	<ul style="list-style-type: none"> • Iniciar sesión con el usuario correspondiente • Seleccionar que si lo desea guardar 	El profesor puede consultar en historial de exámenes el examen correspondiente	✓

1.Diseño de Arquitectura de Software.

La arquitectura está estructurada en distintos módulos lo que garantiza una división de funciones para una ejecución eficiente y un mantenimiento flexible.

1.1 Módulo de Cliente.

Usuarios principales.

- Administradores: Responsables de gestionar usuarios, asignar roles y monitorear resultados.
- Profesores: acceda a herramientas específicas para generar y gestionar exámenes.

Interfaces de usuario.

- Usuario Administrador y Usuario Tutor: Estos sistemas lo hacen más fácil e individual para cada tipo de usuario
- Interfaces para Usuario Administrador y Tutor: Estos sistemas simplifican el proceso y lo adaptan a tipos de usuarios individuales.

Interacción con el Backend.

- Los usuarios se autentican a través de OAuth, lo que garantiza un acceso seguro.
- Interacción perfecta con las capas de generación de exámenes y escáner de preguntas.

1.2 Módulo de Generación de Exámenes.

Funciones principales.

- EL profesor pueda elegir ciertos temas, cuente cuántas preguntas y elija qué tan difíciles deben ser los temas.
- Generación de exámenes en formato PDF.
- Administre archivos PDF (creados por objetos) en Firebase para mantener registros de exámenes.

Flujo de trabajo.

- 1.Configuración inicial por parte del profesor (temas, preguntas).
- 2.Generación del examen y conversión a los formatos seleccionados.
- 3.Guardado para referencia posterior.

1.3 Módulo de escáner de preguntas.

Procesos Automatizados.

- Escaneo de preguntas físicas para convertirlas en contenido digital.
- Procesador de fórmulas basado en predicción y traducción a LaTeX.
- Convertimos las preguntas escaneadas a un formato legible y luego las guardamos en un almacenamiento de datos.

2. Propuesta de Soluciones de Software

2.1 Generación de Exámenes Personalizados

- Configuración basada en temas, dificultad, y cantidad de preguntas.

2.2 Procesamiento de Fórmulas Escaneadas

- Traducción automática de fórmulas complejas a LaTeX, facilitando su edición y reutilización.
- Exportación directa de preguntas escaneadas a formatos compatibles para su integración en el sistema.

2.3 Exportación Flexible

- Los datos generados pueden ser exportados a:
 - Excel: Ideal para análisis estadísticos y reportes personalizados.
 - HTML: Para la publicación y visualización en plataformas web.
 - LaTeX: Específicamente útil para la creación de documentos científicos y educativos.

3. Integración y Migración de Datos

Evolución del proyecto y cambios implementados.

Inicialmente planteado.

El proyecto se planeó originalmente como una aplicación web que utilizaba un modelo GPT-2 (Transformador generativo pre entrenado).

- Esta estrategia mostró excelentes habilidades en la extracción de solo una fórmula matemática en la imagen, pero planteó desafíos de producción considerables.
- Altos costos computacionales.

Cambio realizado.

- Migrado a una aplicación de escritorio con funcionalidades específicas para profesores.
- Le permite extraer una sección del PDF que contiene las fórmulas matemáticas necesarias.
- Se cambió el uso del GPT y ahora se utiliza un transformador de visión (ViT) para procesos de tareas visuales, incluida la clasificación de imágenes.
- Utiliza una arquitectura combinada con:
 - ResNet ↔ Red neuronal convolucional (CNN) como red fundamental para procesar fórmulas matemáticas y producir su representación LaTeX.
 - El modelo Transformer se centra en el aprendizaje y la mejora continua en función de las cualidades que aprende.

Arquitectura de software.

El sistema adoptó inicialmente una arquitectura monolítica en la que todos los componentes están integrados.

El diseño ahora utiliza una construcción monolítica con unidades de microservicio (en cortos casos), que incorpora distintos módulos para el manejo de las imágenes, datos y las bases de datos, lo que facilita la evolución y mejora la productividad operativa.

4. Interoperabilidad y Estándares Abiertos

4.1 Uso de Formatos Estándar

- Excel: Un formato ampliamente utilizado para la gestión y análisis de datos.
- HTML: Garantiza la compatibilidad con navegadores web y sistemas de gestión de contenido.
- LaTeX: Asegura que los documentos generados cumplen con estándares de alta calidad para la publicación académica.

4.2 Autenticación y Seguridad

- OAuth: Utilizado para la autenticación segura de usuarios.
- La compatibilidad con estándares abiertos asegura que el sistema puede integrarse con otras plataformas.

5. Escalabilidad y Flexibilidad del Diseño

Al utilizar Firebase y MongoDB nos permite manejar grandes cantidades de datos de una forma más eficiente pero sin afectar el rendimiento. Al no depender una base de datos relacional nos permite expandir de mejor forma TestFlow

Los módulos que tenemos actualmente, los hemos diseñado para que funcionen de una manera independiente y que podamos expandirlos.

Esto con el fin de implementar nuevos módulos sin afectar el rendimiento de los ya creados.

3. Infraestructura de Hardware y Tecnologías

a. Evaluación de Opciones de Servidores

Se tomó en cuenta Render para el backend y Cloudflare para el frontend en el despliegue del sistema, principalmente debido al conocimiento previo que se tenía sobre ambos por parte del equipo al igual de su integración con GitHub, lo que nos facilitó al momento del despliegue, además de usarlos por su escalabilidad y sus costos, se implementó con nuestro propio dominio. No se consideraron otras opciones de servidores, ya que estos servidores cumplían con lo necesario para nuestro sistema.

b. Adquisición de Tecnologías Adicionales

Decidimos usar Firebase Storage para guardar de manera segura los exámenes que generaban los profesores y así implementar su historial. Esto al principio no estaba contemplado, pero al buscar soluciones para el almacenamiento Firebase Storage resultó ser una opción eficiente para almacenar los archivos de una forma segura y accesible en la nube.

Diagrama de arquitectura:

https://drive.google.com/file/d/1Riyv_PuxBOt6lh0NdEVV6a2tAvi5AE8Q/view?usp=drive_link

,