**Introducción:**

La Plataforma de Ejercicios de Álgebra es un sistema diseñado para facilitar el aprendizaje y la práctica de conceptos y habilidades algebraicas. Esta plataforma ofrece a profesores y alumnos una herramienta interactiva y colaborativa que les permite crear, asignar, completar y evaluar ejercicios, tareas y exámenes relacionados con el álgebra.

Nuestro objetivo principal es proporcionar a los profesores una plataforma intuitiva y eficiente para crear y administrar recursos educativos de álgebra, y a los alumnos, un entorno virtual donde puedan practicar y desarrollar sus habilidades matemáticas.

Características clave:

Creación de ejercicios: Los profesores tienen la capacidad de crear una amplia variedad de ejercicios de álgebra, personalizando el enunciado, la solución y otros parámetros relevantes, como el nivel de dificultad y la categoría temática.

Asignación de tareas: Los profesores pueden crear tareas personalizadas seleccionando una serie de ejercicios y asignándolos a los alumnos. Las tareas pueden tener fechas de vencimiento y pueden ser programadas de acuerdo con el plan de estudios.

Realización de tareas y exámenes: Los alumnos pueden acceder a las tareas asignadas y realizar los ejercicios correspondientes. También pueden realizar exámenes programados por los profesores.

Evaluación automática: El sistema ofrece una función de evaluación automática de las respuestas de los alumnos, lo que permite una retroalimentación inmediata sobre la precisión de sus soluciones.

Seguimiento del progreso: Tanto los profesores como los alumnos tienen acceso a estadísticas y seguimiento del progreso, lo que les permite evaluar el rendimiento individual y colectivo en el dominio del álgebra.

Esperamos que esta Plataforma de Ejercicios de Álgebra ayude a los profesores a optimizar su tiempo de planificación y evaluación, y a los alumnos a fortalecer sus habilidades matemáticas a través de la práctica activa y el feedback inmediato. Nuestro objetivo es fomentar el aprendizaje colaborativo y el compromiso activo con el álgebra, brindando una experiencia educativa enriquecedora y efectiva.

**Desarrollo:**

1. *Requisitos del sistema:*

La Plataforma de Ejercicios de Álgebra requiere un entorno adecuado para su correcto funcionamiento. A continuación, se detallan los requisitos mínimos del sistema:

* Entorno de desarrollo: Python 3.11.2 y el framework Django en su versión 4.2.1
* Sistema operativo: La plataforma es compatible con los principales sistemas operativos, incluyendo Windows, macOS y Linux.
* Base de datos: La plataforma es compatible con bases de datos relacionales, como PostgreSQL, MySQL o SQLite. Se requiere que el sistema tenga una instancia de la base de datos configurada y accesible.
* Servidor web: Para desplegar la plataforma en un entorno de producción, se recomienda utilizar un servidor web compatible con Django, como Apache o Nginx.
* Dependencias y bibliotecas: La plataforma hace uso de diferentes bibliotecas y dependencias de terceros como jQuery, y otras bibliotecas específicas de álgebra. Se proporcionará un archivo de requisitos detallando todas las dependencias necesarias.
* Navegadores compatibles: La plataforma se ha diseñado para ser compatible con los navegadores web modernos, como Google Chrome, Mozilla Firefox y Microsoft Edge.

Es importante tener en cuenta que estos son los requisitos mínimos recomendados, y pueden variar según las necesidades específicas del proyecto. Se recomienda contar con un entorno de desarrollo que cumpla con estos requisitos para garantizar un funcionamiento adecuado de la plataforma.

1. *Instalación y configuración:*

Para instalar y configurar la Plataforma de Ejercicios de Álgebra en tu entorno local o en un servidor, sigue los siguientes pasos:

1. Requisitos previos: Asegúrate de tener instalado Python 3.x en tu sistema. Puedes descargarlo desde el sitio web oficial de Python (https://www.python.org) y seguir las instrucciones de instalación adecuadas para tu sistema operativo.
2. Clonar el repositorio: Clona el repositorio de la plataforma desde el repositorio Git utilizando el comando git clone [URL\_DEL\_REPOSITORIO]. Si prefieres, puedes descargar el código fuente en formato ZIP desde el repositorio y descomprimirlo en una ubicación deseada en tu sistema.
3. Crear un entorno virtual (opcional pero recomendado): En el directorio del proyecto, crea un entorno virtual utilizando la herramienta venv de Python.

Ejecuta el comando **python3 -m venv myenv** para crear un nuevo entorno virtual llamado "myenv". Luego, activa el entorno virtual ejecutando source myenv/bin/activate en sistemas Unix/Linux o myenv\Scripts\activate en Windows.

1. Instalar las dependencias: Utiliza el gestor de paquetes **pip** para instalar las dependencias necesarias.

Ejecuta el comando **pip install -r requirements.txt** en la raíz del proyecto para instalar todas las bibliotecas y dependencias requeridas.

1. Configuración de la base de datos: Configura la conexión a la base de datos modificando el archivo *settings.py*. Asegúrate de proporcionar la información correcta sobre el motor de base de datos, la dirección, el puerto, el nombre de la base de datos y las credenciales de acceso.
2. Realizar migraciones: Ejecuta el comando **python manage.py migrate** para aplicar las migraciones y crear las tablas necesarias en la base de datos.
3. Crear un superusuario (opcional): Si deseas acceder al panel de administración de Django, puedes crear un superusuario utilizando el comando python manage.py createsuperuser. Sigue las instrucciones para establecer un nombre de usuario y una contraseña para el superusuario.
4. Ejecutar el servidor: Finalmente, ejecuta el servidor de desarrollo de Django con el comando **python manage.py runserver**. Esto iniciará el servidor en http://localhost:8000 (o en otro puerto si está ocupado). Accede a esta dirección en tu navegador para ver la plataforma en funcionamiento.

Hay que tener en cuenta que estos pasos son una guía general para la instalación y configuración del proyecto. Pueden variar dependiendo de tu configuración específica o de los requisitos adicionales del proyecto.

1. *Arquitectura y estructura del proyecto:*

La Plataforma de Ejercicios de Álgebra sigue una arquitectura basada en el framework Django, que utiliza el patrón de diseño Modelo-Vista-Controlador (MVC) para organizar y gestionar los componentes del proyecto. A continuación, se describen los principales elementos de la arquitectura y la estructura del proyecto:

* Modelos de datos:
* La plataforma de ejercicios de álgebra utiliza una variedad de modelos de datos para representar las entidades principales y establecer relaciones entre ellas, por ejemplo, se establecen modelos para los dos tipos de usuarios que operan en la plataforma, como pueden ser profesor y alumno. Dependiendo de los requisitos específicos del proyecto, es posible que se agreguen otros modelos o se realicen modificaciones en los existentes.
* Vistas:
* Plantillas:
  + Crea plantillas HTML para cada una de las vistas mencionadas anteriormente, utilizando el motor de plantillas de Django para generar contenido dinámico.
  + Diseña una interfaz de usuario atractiva y fácil de usar que permita a los usuarios interactuar con la plataforma de manera intuitiva.
* Controladores:
  + Define controladores (vistas basadas en clases o funciones) que manejen la lógica de negocio de tu plataforma, como la autenticación de usuarios, el procesamiento de formularios, el almacenamiento y recuperación de datos en la base de datos, etc.
* Integración con Django Admin:
  + La implementación de Django nos deja provechar las características de administración para gestionar los modelos de datos de forma fácil y conveniente. Django Admin te permitirá administrar usuarios, ejercicios, tareas, exámenes, etc., directamente desde el panel de administración.

1. *Dependencias y bibliotecas utilizadas:*

La plataforma de ejercicios de álgebra se ha desarrollado utilizando varias dependencias y bibliotecas de terceros para mejorar la funcionalidad y la experiencia de usuario. A continuación, se enumeran las principales bibliotecas utilizadas junto con sus versiones correspondientes:

Django (versión 3.11.2): Un framework de desarrollo web de alto nivel que proporciona un entorno robusto para construir aplicaciones web rápidas y seguras.

Bootstrap (versión 4.6): Un framework de diseño y desarrollo web que proporciona componentes y estilos predefinidos para crear interfaces modernas y responsivas.

jQuery (versión 3.7.0): Una biblioteca de JavaScript rápida, pequeña y rica en funciones que simplifica la manipulación del DOM y la gestión de eventos en las páginas web.

MathJax (versión 3.0): Una biblioteca de JavaScript que permite mostrar fórmulas matemáticas en la plataforma utilizando la notación de TeX o MathML.

SQLite (versión 3): Un sistema de gestión de bases de datos relacional de código abierto que se utiliza como backend para almacenar los datos de la plataforma.

1. *Configuración de la base de datos:*

La plataforma de ejercicios de álgebra utiliza el sistema de gestión de bases de datos SQLite para almacenar y gestionar los datos de forma local.

Django utiliza una base de datos SQLite por defecto para entornos de desarrollo. No es necesario realizar ninguna instalación adicional, ya que SQLite viene incluido en la biblioteca estándar de Python.

En el archivo de configuración del proyecto Django, (*settings.py*) dentro de la variable DATABASES, se establecen los siguientes valores:

Texto

Descripción generada automáticamente

1. *Autenticación y autorización:*

La plataforma de ejercicios de álgebra cuenta con un sistema de autenticación y autorización para gestionar el acceso de los usuarios y asignar roles específicos. A continuación, se describen las principales características de este sistema:

Registro de usuarios: Los usuarios pueden registrarse en la plataforma proporcionando su nombre de usuario, dirección de correo electrónico y contraseña. Se utiliza un formulario de registro que valida los campos ingresados y crea una cuenta de usuario.

Inicio de sesión: Los usuarios registrados pueden iniciar sesión en la plataforma utilizando su nombre de usuario y contraseña. Se verifica la autenticidad de las credenciales y se genera una sesión de usuario válida.

Roles de usuario: Se definen dos roles de usuario en la plataforma: "profesor" y "alumno". El rol de "profesor" permite acceder a funciones adicionales, como la creación de ejercicios, la asignación de tareas y la evaluación de respuestas de los alumnos.

Gestión de permisos: Se implementa un sistema de gestión de permisos para restringir el acceso a determinadas funciones y vistas según el rol del usuario. Por ejemplo, solo los profesores tienen permisos para crear ejercicios, mientras que los alumnos solo pueden realizar tareas y exámenes.

Seguridad de contraseñas: Las contraseñas de los usuarios se almacenan de forma segura utilizando técnicas de hash y almacenamiento seguro. Esto garantiza que las contraseñas no sean accesibles incluso si la base de datos se ve comprometida.

El sistema de autenticación y autorización se implementa utilizando las funcionalidades proporcionadas por Django. Se utilizan las vistas y formularios predeterminados de Django, así como el sistema de modelos de usuario para almacenar y gestionar la información de los usuarios.

1. *Modelos de datos:*

Texto

Descripción generada automáticamente

* + Usuario:

Define los campos comunes para profesores y alumnos, como nombre de usuario, contraseña, dirección de correo electrónico, matricula, etc.

* + Profesor:

Extiende el modelo de Usuario e incluye campos adicionales específicos de los profesores, como nombre completo, materias impartidas, etc.

* + Alumno:

Extiende el modelo de Usuario e incluye campos adicionales específicos de los alumnos, como matricula.

Texto

Descripción generada automáticamente

* Tarea: Representa una tarea asignada por un profesor a un grupo específico de estudiantes. Contiene información sobre la fecha de inicio, la fecha de entrega y los ejercicios incluidos en la tarea.

Texto

Descripción generada automáticamente

* Ejercicio: Almacena los detalles de un ejercicio de álgebra, como el enunciado, las instrucciones y las soluciones.

Texto

Descripción generada automáticamente

* + Examen: Representa un examen programado y contiene una lista de ejercicios relacionados.

1. *Vistas y controladores:*

La plataforma de ejercicios de álgebra utiliza una combinación de vistas y controladores para gestionar las interacciones entre los usuarios y el sistema. A continuación, se describen las principales vistas y controladores utilizados en el proyecto:

**Vistas basadas en funciones** (Function-based views): Se utilizan vistas basadas en funciones para manejar las acciones más simples y directas, como la renderización de páginas estáticas, el procesamiento de formularios simples y la autenticación de usuarios. Estas vistas están definidas como funciones de Python y se enlazan a URLs específicas utilizando el enrutador de URL de Django.

**Controladores** (Controllers): Los controladores son responsables de recibir las solicitudes de los usuarios, interactuar con los modelos de datos correspondientes y enviar la respuesta adecuada. En Django, los controladores se implementan mediante las vistas y los controladores de Django, que permiten la gestión de las acciones del usuario, la obtención y modificación de los datos y la renderización de las plantillas HTML correspondientes.

El enrutamiento de las URL hacia las vistas y controladores se realiza mediante el archivo de configuración de URL (*urls.py*) del proyecto. Las URL se mapean a las vistas y controladores correspondientes, lo que permite una navegación fluida a través de las diferentes páginas y funcionalidades de la plataforma.

Además, se utilizan decoradores y clases de autenticación proporcionados por Django para gestionar la autenticación y autorización de los usuarios en las vistas y controladores. Esto garantiza que solo los usuarios autenticados y con los permisos adecuados puedan acceder a determinadas funcionalidades y recursos.

La combinación de vistas y controladores en la plataforma de ejercicios de álgebra permite gestionar de manera eficiente las interacciones entre los usuarios y el sistema, proporcionando una experiencia fluida y segura.

* + Registro de usuarios:

Vistas para que los profesores y alumnos puedan registrarse en la plataforma.

Cuando el usuario sea estudiante deberá marcar la casilla de *Soy alumno.*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

En las vistas basadas en funciones, el controlador es una función de Python que recibe la solicitud HTTP como argumento y realiza las operaciones necesarias para generar la respuesta. Por ejemplo, la función de vista puede acceder a los modelos de datos, procesar formularios, realizar consultas a la base de datos y renderizar una plantilla HTML para mostrar la respuesta al usuario.

* + Inicio de sesión:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Vistas para que los usuarios puedan iniciar sesión en la plataforma o ingresar como invitado (limitando su interacción a resolución de ejercicios de práctica y lectura del temario de la materia).

* + Perfil de usuario:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Vistas para que los usuarios puedan ver y editar su perfil. El cambio de la matricula, al ser información delicada que puede influir de manera negativa en la logistica del registro de los alumnos, se limita a sólo poder ser modificada por el administrador

* + Creación de ejercicios:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Vistas para que los profesores puedan crear nuevos ejercicios de álgebra.

* + Creación de tareas:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Vistas para que los profesores puedan crear tareas y asignar ejercicios a los alumnos.

* + Entrega de tareas:

Interfaz de usuario gráfica, Aplicación, PowerPoint

Descripción generada automáticamente

Vistas para que los alumnos puedan ver las tareas asignadas y enviar sus respuestas.

* Realización de exámenes:

Vistas para que los alumnos puedan realizar exámenes y enviar sus respuestas.

1. *Plantillas y diseño de interfaz*

La plataforma de ejercicios de álgebra utiliza plantillas HTML para generar las páginas web y define un diseño de interfaz atractivo y fácil de usar. A continuación, se describen los principales aspectos relacionados con las plantillas y el diseño de interfaz:

**Plantillas base** (Base templates): Se utilizan plantillas base para definir la estructura y los elementos comunes de todas las páginas de la plataforma. Estas plantillas incluyen el encabezado, el pie de página, la barra de navegación y otros componentes que se repiten en todas las páginas. El uso de plantillas base permite mantener una apariencia y una funcionalidad coherentes en toda la plataforma.

**Plantillas específicas** (Specific templates): Se crean plantillas específicas para cada página o funcionalidad de la plataforma. Estas plantillas heredan de la plantilla base y definen los elementos y el contenido específico de cada página. Por ejemplo, puede haber plantillas para la página de inicio, la creación de ejercicios, la realización de tareas y la visualización de resultados. Se utilizan etiquetas y filtros de Django en las plantillas para acceder a los datos dinámicos y mostrarlos correctamente.

**Estilos CSS** (CSS Styles): Se aplican estilos CSS a las plantillas para mejorar la apariencia visual de la plataforma. Se utiliza un archivo de estilo CSS principal que define las reglas y los estilos generales, como colores, fuentes, márgenes y espaciado. Además, se pueden utilizar hojas de estilo CSS adicionales para personalizar el diseño de páginas específicas o componentes particulares.

**Interactividad y experiencia del usuario** (User Interaction): Se implementan elementos interactivos en las plantillas para mejorar la experiencia del usuario. Esto puede incluir formularios interactivos, botones de acción, barras de progreso y mensajes de confirmación. Se utilizan scripts de JavaScript o frameworks de JavaScript como jQuery para agregar funcionalidad interactiva a las páginas y permitir una interacción suave y fluida con la plataforma.

**Diseño responsivo** (Responsive Design): Se adopta un diseño responsivo para que la plataforma se adapte y se vea bien en diferentes dispositivos y tamaños de pantalla. Se utilizan técnicas como el diseño fluido, los media queries y las rejillas CSS para garantizar que el contenido se ajuste y se muestre correctamente en dispositivos móviles, tabletas y ordenadores de escritorio.

El diseño de interfaz de la plataforma se basa en principios de usabilidad, accesibilidad y diseño visual atractivo. Se tiene en cuenta la legibilidad del texto, la organización lógica de los elementos, el contraste de colores y la facilidad de navegación para proporcionar a los usuarios una experiencia agradable y efectiva al utilizar la plataforma.

1. *Funcionalidades principales:*
2. *Pruebas y depuración:*
3. *Despliegue y puesta en producción:*
4. *Guía de usuario:*
5. *Mantenimiento y mejoras futuras:*