

VSystem

Yider Daniel Melo Infante

ID 814580

MGs. Eddy Javier Pérez Ómbita

Universidad Cooperativa de Colombia

Sistemas distribuidos

Villavicencio- Meta

2024

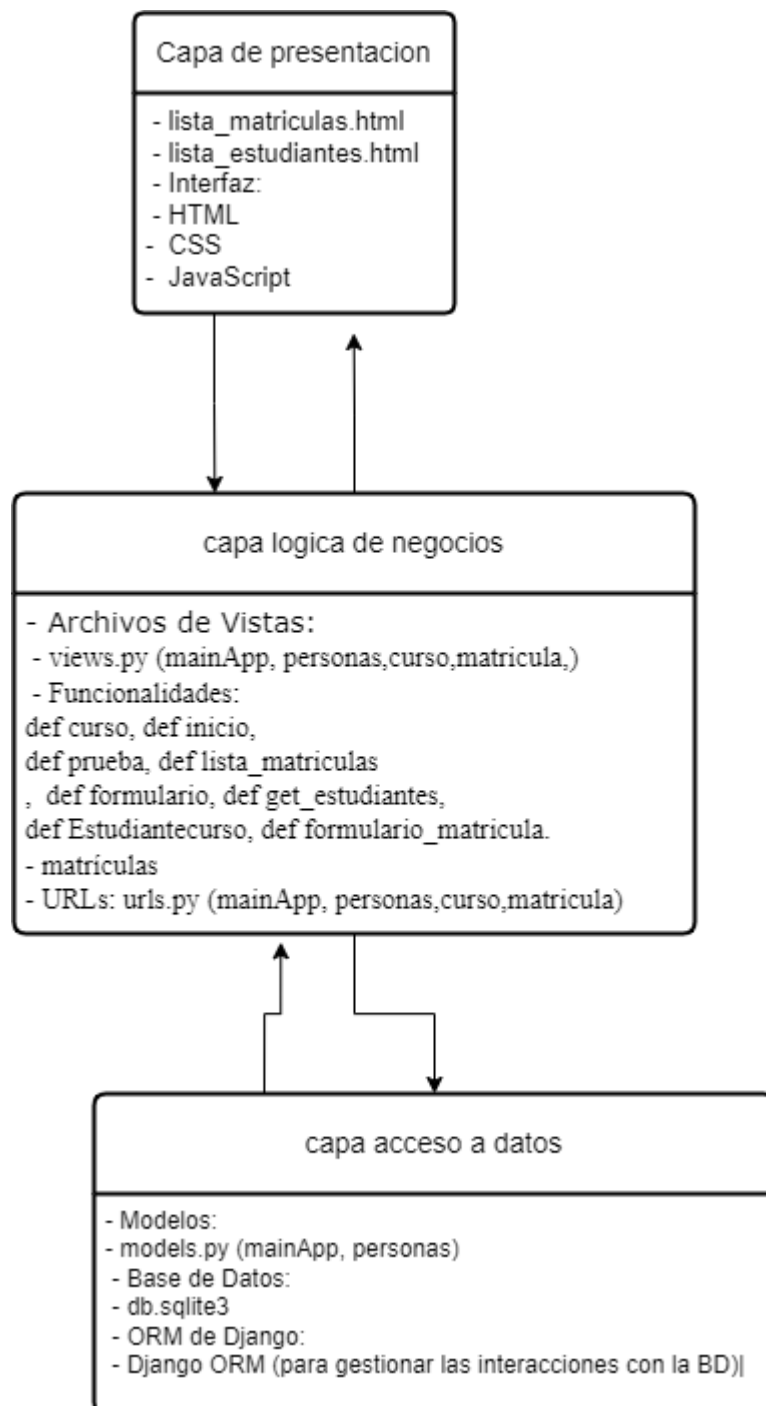
Contenido

Descripción General.....	9
Diagrama de arquitectura	9
Descripción de las Capas	10
Capa de Presentación.....	10
Capa de Lógica de Negocios.....	10
Capa de Acceso a Datos.....	10
Tecnologías y Herramientas	10
Patrones y Principios.....	11
Ejemplos de Flujo de Trabajo	11
Link repositorio en GitHub	12

Descripción General

El proyecto VSystem es una aplicación de gestión académica que busca facilitar la administración de cursos, estudiantes, estudiantes-cursos y su matrícula. Su objetivo es facilitar la interacción entre los usuarios (administradores, profesores y estudiantes) y proporcionar un sistema que permita el seguimiento del progreso académico.

Diagrama de arquitectura



Descripción de las Capas

Capa de Presentación

En esta capa tenemos los templates en donde se encuentra las plantillas html como lo son index.html, prueba.html, lista-estudiantes.html, curso.html , formulario.html, estudiante_curso.html y lista-matriculas.html

Capa de Lógica de Negocios

En la capa de lógica de negocios tenemos el funcionamiento de cada uno de los métodos. También se crean las clases para cada modelo y las funciones , entre esas tenemos: class Curso, def curso, def inicio, def prueba, class Matricula, def lista_matriculas, class Persona, class EstudianteCurso, def formulario, def get_estudiantes, def Estudiantecurso, def formulario_matricula. Estas son los modelos y funciones con los que trabaja el proyecto VSystem.

Capa de Acceso a Datos

Se encarga de interactuar con la base de datos mediante Django ORM en cada uno de los modelos de VSystem se creo una base de datos para que manejaran documentos de manera más limpia y sencilla, las bases que se tienen son:

1. def Meta(self):

db_table = 'curso'

2. class Meta:

db_table = 'Matricula'

3. class Meta:

db_table = 'persona'

4. class Meta:

db_table = 'Estudiante_Curso'

Tecnologías y Herramientas

Capa de Presentación: HTML, CSS, JavaScript, Django

Capa de Lógica de Negocios: Django, Python

Capa de Acceso a Datos: Django ORM , db.sqlite3

Patrones y Principios

MVC (Modelo-Vista-Controlador): Se utiliza para separar la lógica de negocio, la presentación y el manejo de datos.

Ejemplos de Flujo de Trabajo

1. Registro de un nuevo estudiante

Capa de Presentación:

El usuario accede a la vista del formulario de registro de estudiantes en la página web. Este formulario está implementado en un archivo HTML (ej. formulario_estudiante.html), donde el usuario ingresa los datos del estudiante (nombre, apellidos, DNI, etc.).

Capa de Lógica de Negocios:

Al enviar el formulario, los datos pasan a la vista correspondiente en el archivo views.py. Allí, se ejecuta la lógica de validación de los datos (asegurarse de que todos los campos requeridos están completos, el formato es correcto, etc.). Si los datos son válidos, se crea una nueva instancia del modelo Persona.

Capa de Acceso a Datos:

La nueva instancia del modelo Persona es guardada en la base de datos utilizando el ORM de Django. En este caso, los datos del estudiante se almacenan en las tablas asociadas a Persona.

2. Listar todos los cursos

Capa de Presentación:

El usuario visita la página donde se listan todos los cursos disponibles. La plantilla lista_cursos.html se carga y muestra una tabla con los cursos.

Capa de Lógica de Negocios:

La vista en views.py consulta el modelo Curso para obtener todos los cursos disponibles en la base de datos. Esta consulta es filtrada para solo mostrar los cursos activos.

Capa de Acceso a Datos:

El ORM de Django genera una consulta SQL que recupera los datos de los cursos desde la tabla correspondiente en la base de datos

3. Asignar un estudiante a un curso (Matrícula)

Capa de Presentación:

Un usuario con permisos administrativos accede al formulario de asignación de estudiantes a un curso. El formulario en `formulario_matricula.html` permite seleccionar un estudiante y un curso desde listas desplegables.

Capa de Lógica de Negocios:

Una vez enviado el formulario, la vista en `views.py` procesa la asignación. El código toma el estudiante seleccionado y el curso elegido, valida que ambos existan y luego crea una nueva entrada en el modelo `EstudianteCurso`, asociando al estudiante con el curso.

Capa de Acceso a Datos:

El ORM de Django guarda esta relación en la base de datos, probablemente en una tabla intermedia que mantiene la relación entre los estudiantes y los cursos.

Link repositorio en GitHub

<https://github.com/Yider18/VSystem.git>