PROYECTO FINAL*

,[1] Daniel Estuardo Suy Fuentes, 202004811,[1] and Angela Beatriz Canel Hernández, 201906569[1]

[1] *Facultad de Ingeniería, Escuela de Mecánica Eléctrica, Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.*

La plataforma .Academia de Ingeniería.es un sitio web desarrollado con el objetivo de compartir los conocimientos adquiridos. El sitio cuenta con una página principal con información general, un sistema de registro e inicio de sesión de usuarios (estudiantes) para poder acceder a los cursos, un catálogo de cursos con detalles de cada uno, un proceso de matrícula y simulación de pago, así como funcionalidades administrativas para gestionar cursos y usuarios. Los estudiantes podrán crear una cuenta con sus datos personales, elegir los cursos de su interés del catálogo publicado, realizar una matrícula simulada especificando forma de pago, y luego podrán acceder a los contenidos de los cursos matriculados. Por parte del administrador, se podrán realizar tareas como creación/edición de cursos, gestión de usuarios, generación de reportes de estudiantes, ventas y cursos. También manejo de cuentas bloqueadas por intentos fallidos de inicio de sesión. El sitio utiliza PostgreSQL como gestor de base de datos para manejar los datos de usuario, cursos, matrículas, etc. Y en el código se aplicará encriptación a los datos sensibles como las contraseñas.

I. OBJETIVOS

A. General

- Desarrollar una plataforma web de gestión de cursos utilizando Python con Django, integrada a una base de datos PostgreSQL, que permita administrar usuarios, cursos e inscripciones, así como la generación de reportes.

B. Específicos

- Construir el sistema de gestión de la plataforma de cursos mediante el framework Django de Python, implementando funcionalidades de registro y autenticación de usuarios, creación y asignación de cursos, procesamiento de matrículas e integración con pasarela de pagos.
- Diseñar y crear una base de datos relacional en PostgreSQL para el manejo de los datos del sistema, aplicando operaciones de inserción, lectura, actualización y eliminación de registros desde el código en Python.
- Generar reportes de la información almacenada en la base de datos contenidos datos como usuarios, cursos, matriculados, ventas, entre otros; en formato Excel y PDF utilizando librerías de Python para interactuar con PostgreSQL.

II. GITHUB

`https://github.com/DanielSuy/Proyecto-Final`

*

III. MARCO TEÓRICO

A. Python

Python es un lenguaje de programación de alto nivel que se ha vuelto extremadamente popular en los últimos años debido a su simplicidad, legibilidad y versatilidad. Fue creado por Guido van Rossum y lanzado por primera vez en 1991. Desde entonces, Python ha ganado una gran comunidad de desarrolladores y se ha convertido en uno de los lenguajes más utilizados en una amplia gama de aplicaciones, desde desarrollo web hasta análisis de datos y aprendizaje automático.

Características de Python: Python se destaca por varias características que lo hacen único y atractivo para los desarrolladores:

1. Sintaxis clara y legible: Python utiliza una combinación de espacios en blanco significativos y una estructura de bloques indentada en lugar de llaves y puntos y comas, lo que facilita la lectura y comprensión del código.

2. Multiparadigma: Python admite diferentes estilos de programación, como programación orientada a objetos, programación imperativa y programación funcional. Amplia biblioteca estándar: Python cuenta con una biblioteca estándar extensa y poderosa que incluye módulos para realizar una amplia variedad de tareas, como manipulación de archivos, acceso a bases de datos, procesamiento de texto, generación de gráficos y mucho más. Esto permite a los desarrolladores ahorrar tiempo y esfuerzo al utilizar módulos preexistentes en lugar de reinventar la rueda.

3. Amplia biblioteca estándar: Python cuenta con una biblioteca estándar extensa y poderosa que incluye módulos para realizar una amplia variedad de tareas, como manipulación de archivos, acceso a ba-

ses de datos, procesamiento de texto, generación de gráficos y mucho más.

4. Comunidad activa: Python cuenta con una comunidad de desarrolladores muy activa y comprometida. Esto se refleja en la gran cantidad de bibliotecas y marcos de trabajo de terceros disponibles para Python.

5. Interfaz con otros lenguajes: Python se puede utilizar en conjunto con otros lenguajes de programación. Por ejemplo, Python se puede utilizar para escribir scripts de automatización que interactúen con código escrito en C++ o Java.

Aplicaciones de Python Python se utiliza en una amplia gama de aplicaciones en diversos campos. Algunas de las áreas en las que Python ha demostrado ser especialmente útil incluyen:

1. Desarrollo web: Python es ampliamente utilizado en el desarrollo web, gracias a marcos de trabajo populares como Django y Flask. Estos marcos simplifican la creación de aplicaciones web robustas y escalables.

2. Análisis de datos: Python ha ganado popularidad en el campo del análisis de datos debido a su facilidad de uso y a las bibliotecas como NumPy, Pandas y Matplotlib. Estas herramientas permiten a los científicos de datos manipular y analizar grandes conjuntos de datos.

3. Aprendizaje automático: Bibliotecas como TensorFlow, Keras y Scikit-learn brindan a los desarrolladores las herramientas necesarias para construir y entrenar modelos de aprendizaje automático de forma rápida y sencilla.

4. Automatización de tareas: Python es ideal para automatizar tareas repetitivas y tediosas. Su sintaxis clara y legible, combinada con bibliotecas como Selenium y Beautiful Soup, permite crear scripts que realicen tareas como la extracción de datos de sitios web o la manipulación de archivos en lote.

Figura 1: Python

B. Django

Django es un framework de desarrollo web de alto nivel y de código abierto que utiliza el lenguaje de programación Python. Proporciona a los desarrolladores una estructura y un conjunto de herramientas para crear aplicaciones web de manera eficiente y robusta.

Características de Django:

1. Arquitectura MVC: Django sigue el patrón de diseño Modelo-Vista-Controlador (MVC). Esto significa que separa la lógica de negocio (modelo), la presentación (vista) y la interacción con el usuario (controlador), lo que facilita la organización y el mantenimiento del código.

2. ORM (Mapeo objeto-relacional): Django utiliza un ORM para interactuar con la base de datos. Esto permite a los desarrolladores trabajar con objetos en lugar de consultas SQL directas, lo que facilita la manipulación y el acceso a los datos.

3. Sistema de enrutamiento: Django tiene un sistema de enrutamiento incorporado que mapea las URL a las vistas correspondientes. Esto permite una gestión eficiente de las rutas y la navegación dentro de la aplicación web.

4. Plantillas: Django proporciona un lenguaje de plantillas que permite a los desarrolladores generar contenido dinámico de manera sencilla. Las plantillas de Django separan la lógica de presentación del código de la aplicación, facilitando la creación de interfaces de usuario atractivas.

5. Seguridad: Django tiene características de seguridad incorporadas, como la protección contra ataques de inyección SQL, ataques de cross-site scripting (XSS) y ataques de falsificación de solicitudes entre sitios (CSRF). Esto ayuda a garantizar que las aplicaciones desarrolladas con Django sean seguras.

Figura 2: Logo Django

C. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto y altamente escalable.

Es conocido por su robustez, confiabilidad y capacidad de manejar grandes volúmenes de datos. Se basa en el modelo de bases de datos relacionales. Fue desarrollado en la Universidad de California, Berkeley en la década de 1980 y se ha convertido en una de las bases de datos más populares y utilizadas en la actualidad.

Características y ventajas de PostgreSQL: Soporte para múltiples plataformas, capacidad para manejar grandes volúmenes de datos, compatibilidad con estándares SQL, extensibilidad y escalabilidad. Además, cuenta con una amplia gama de herramientas y complementos que facilitan su administración y desarrollo.

Arquitectura de PostgreSQL: PostgreSQL sigue una arquitectura cliente-servidor, donde el servidor de PostgreSQL es responsable de administrar y almacenar los datos, mientras que los clientes se conectan al servidor para acceder y manipular los datos. La arquitectura de PostgreSQL se basa en un proceso principal (postmaster) y varios procesos secundarios (backends) que manejan las solicitudes de los clientes.

Funcionalidades avanzadas de PostgreSQL: PostgreSQL ofrece una variedad de funcionalidades avanzadas, como soporte para transacciones ACID, integridad referencial, vistas materializadas, replicación, particionamiento de tablas, indexación avanzada y funciones almacenadas. Estas funcionalidades hacen de PostgreSQL una opción potente y flexible para aplicaciones empresariales y de alto rendimiento.

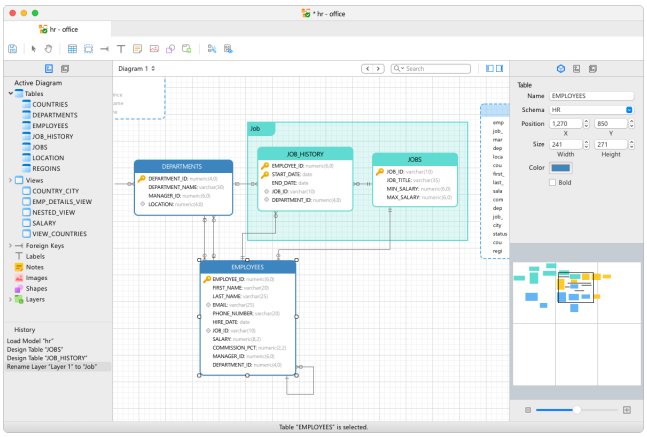


Figura 3: PostgreSQL

IV. MARCO PRÁCTICO

Configuración del entorno de desarrollo

1. Se instaló Python 3.8.2 y el framework Django 5.0 en el equipo local para construir el sistema.

2. Se instaló y configuró PostgreSQL 12 como sistema gestor de base de datos.

Creación del proyecto Django

1. Se utilizó el comando "django-admin"para crear un nuevo proyecto "ProyectoFinal".

2. Se definieron las apps principales del sistema: çore", üsuarios", "pagos", çatálogo".

3. Se configuró la base de datos default en settings para conectar con PostgreSQL.

```
DATABASES = {
    'default': {
        "ENGINE": "django.db.backends.postgresql",
        "NAME": "proyecto",
        "USER": "postgres",
        "PASSWORD": "523811",
        "HOST": "localhost",
        "PORT": "5432",
    }
}
```

Figura 4: Conexión de Django con PostgreSQL

Desarrollo del modelo de datos

1. Se diagramó un modelo entidad-relación de acuerdo a los requerimientos.

2. Se crearon los modelos de Usuario, Curso, Matricula, Factura en las apps correspondientes.

3. Se establecieron las relaciones 1-N entre los modelos.

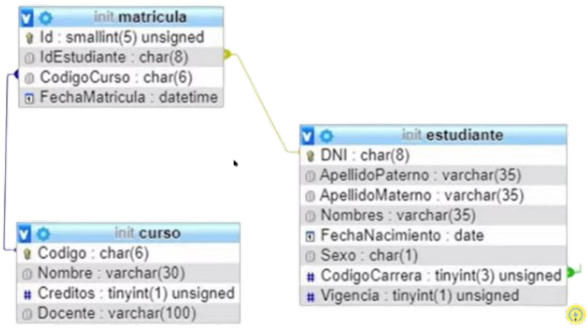


Figura 5: Diagrama entidad-relación

Interfaz de administrador

1. Se habilitó el sitio de administración automatico de Django.

2. Se registraron los modelos creados para interactuar por defecto.

3. Permite gestionar usuarios, cursos, matrículas inicialmente.

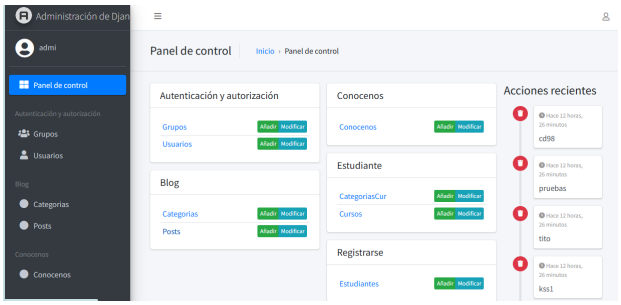
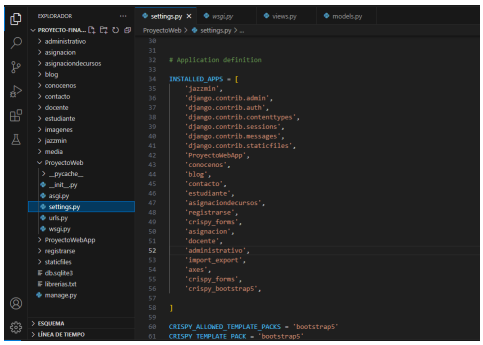


Figura 6: Panel de administración

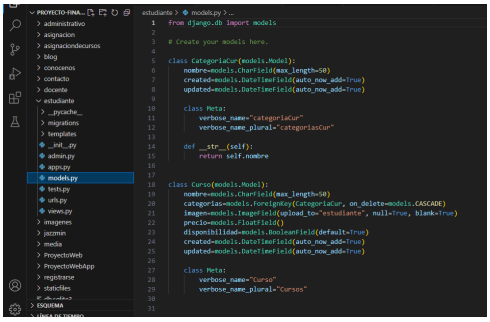


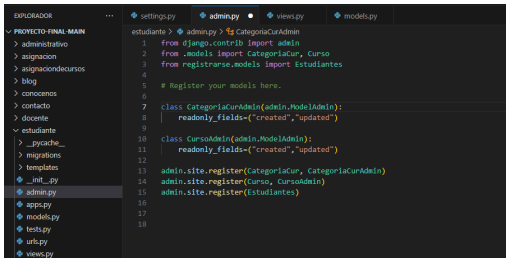
Funcionalidades principales

1. Se construyeron plantillas base con HTML/CSS/JS

2. Se crearon vistas para inicio, registro, login, catálogo de cursos.

3. Se implementaron lógicas de negocio en los views y forms.





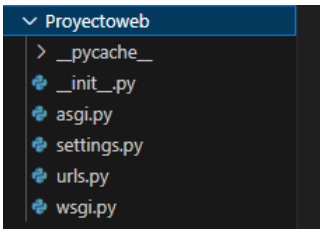
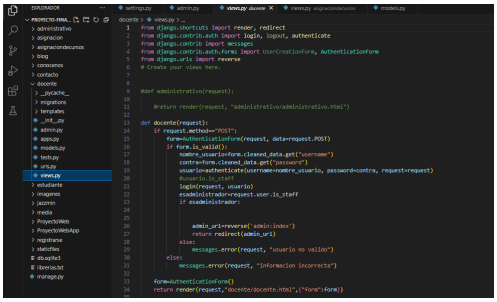
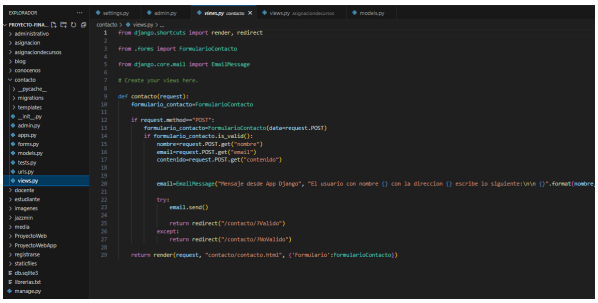
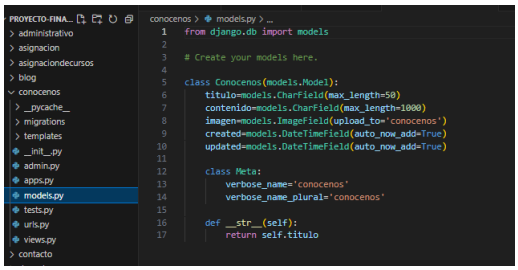


Figura 7: Vistas creadas

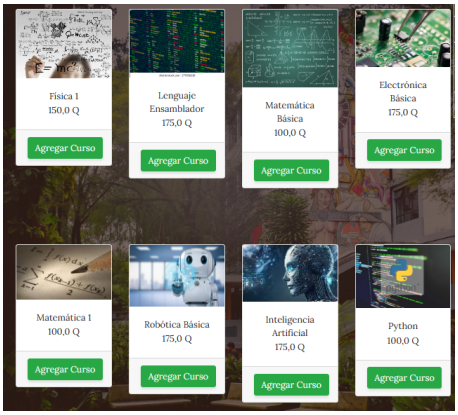


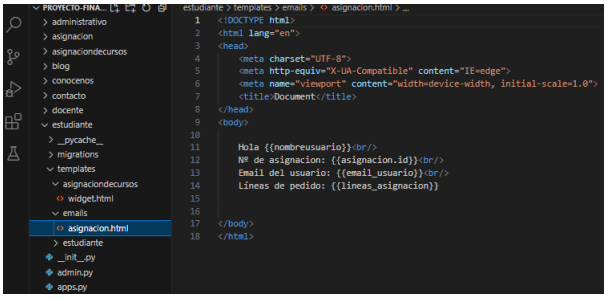
V. CÓDIGOS



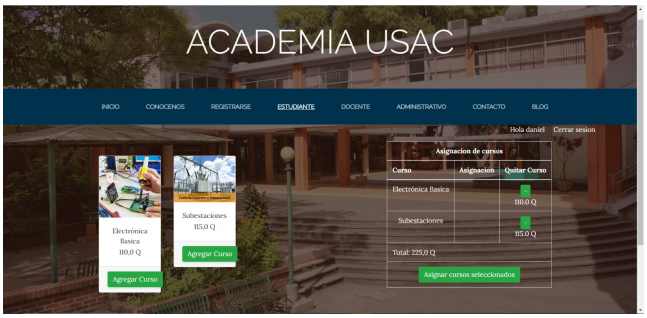


B. Cátalogo de Cursos





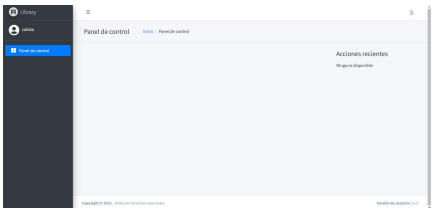
C. Asignación de cursos



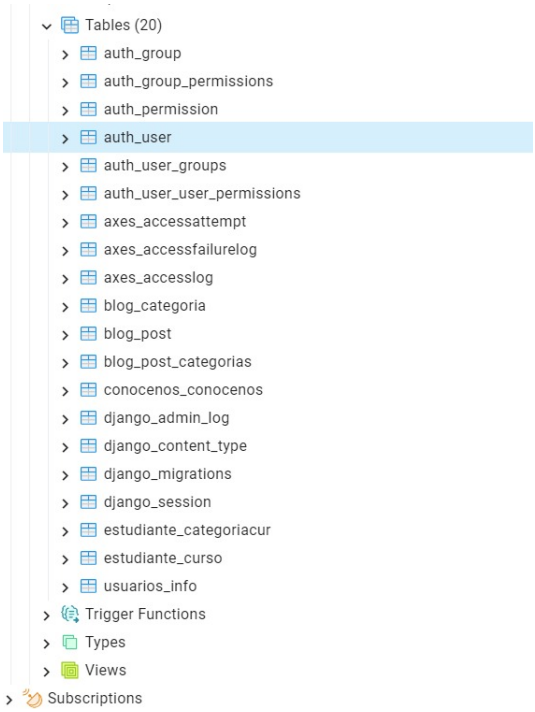
VI. RESULTADOS

A. Pagina Principal

D. Panel Profesor



E. Tablas Migradas en PostgreSQL



VII. CONCLUSIONES

1. Python es un lenguaje de programación versátil y poderoso. Su amplia biblioteca estándar, comunidad activa y capacidad para integrarse con otros lenguajes hacen de Python una opción atractiva para una amplia gama de aplicaciones, desde el desarrollo web hasta el análisis de datos y el aprendizaje automático.

2. El framework Django brinda una estructura sólida y un conjunto de herramientas que facilitan y aceleran el desarrollo de aplicaciones web complejas. Su patrón MVC, ORM, plantillas y énfasis en la seguridad permiten crear sitios web robustos y escalables de manera rápida y organizada.

3. PostgreSQL se consolida como una base de datos relacional madura y confiable, con funcionalidades avanzadas orientadas a aplicaciones empresariales y de alto rendimiento. Su arquitectura cliente-servidor, soporte multiplataforma y capacidad de gestionar grandes volúmenes de datos la convierten en una opción ideal para ser integrada en proyectos web con Python y Django.

4. La combinación de Python, Django y PostgreSQL forma una plataforma de desarrollo web muy sólida y flexible. Python aporta legibilidad y potencia, Django provee la estructura y herramientas para aplicaciones web, y PostgreSQL entrega un almacenamiento de datos escalable y con funciones avanzadas. Juntos empoderan el desarrollo ágil de soluciones web seguras y de alto desempeño.

5. El proyecto desarrollado demuestra la viabilidad de construir un sistema de gestión de cursos en línea aprovechando las capacidades de Python, Django y PostgreSQL; cumpliendo con funcionalidades de administración de contenido, usuarios, reportes y pagos de forma satisfactoria.

---

[1] Lutz, M. (2013). *Learning Python: Powerful Object-Oriented Programming.* O'Reilly Media.

[2] Beazley, D. M. (2019). *Python Essential Reference.* Addison-Wesley Professional.

[3] McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython.* O'Reilly Media.

[4] Holovaty, A., & Kaplan-Moss, J. (2009). *The Definitive Guide to Django: Web Development Done Right.* Apress.

[5] Mele, A. (2018). *Django 2 by Example: Build powerful and reliable Python web applications from scratch.* Packt Publishing.

[6] Gorman, S., & Volkov, I. (2019). *Practical PostgreSQL.* Apress.