



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco

**TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO**

**INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES**

**DESARROLLO DE BACKEND PARA UN SISTEMA INTEGRAL DE GESTIÓN DE
LA CAPACITACIÓN DOCENTE EN EL TECNOLÓGICO CAMPUS APIZACO**

PRESENTA:

MARIA FERNANDA GARCIA FLORES

ASESOR INTERNO:

Dr. JUAN RAMOS RAMOS

ASESOR EXTERNO:

LIC. MIQUELINA SÁNCHEZ PULIDO

JEFA DEL DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

19 DE FEBRERO, 2025



Tzompantepec, Tlaxcala, **05/marzo /2025**

OFICIO No: SC/OPV/0065/25

ASUNTO: Liberación de proyecto
para titulación integral

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

NOMBRE DEL EGRESADO:	GARCÍA FLORES MARÍA FERNANDA
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	19371000
NOMBRE DEL PROYECTO:	Desarrollo de Backend para un sistema integral de gestión de la capacitación docente en el TecNM Campus Apizaco
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

ATENTAMENTE

*Excelencia en Educación Tecnológica
Pensar para Servir, Servir para Triunfar*

MIQUELINA SÁNCHEZ PULIDO
JEFA DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

c.c.p. División de Estudios Profesionales
c.c.p. Depto. de Servicios Escolares
c.c.p. Archivo
MSP/mral*



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DEPARTAMENTO DE
SISTEMAS Y COMPUTACIÓN

JUAN RAMOS RAMOS
ASESOR INTERNO



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241)
4172010 Ext. 144, e-mail: sistemas@apizaco.tecnm.mx |
apizaco.tecnm.mx





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Tzompantepec, Tlaxcala, **26/marzo/2025**

MARIA FERNANDA GARCIA FLORES.
EGRESADA DE LA CARRERA DE INGENIERÍA
EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES.
CON NÚMERO DE CONTROL 19371000.
PRESENTE.

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **"DESARROLLO DE BACKEND PARA UN SISTEMA INTEGRAL DE GESTIÓN DE LA CAPACITACIÓN DOCENTE EN EL TECN CAMPUS APIZACO"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **2 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®



ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

ccp. Archivo
ijss/aehs -



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010
Ext. 304, e-mail: dep_titulacion@apizaco.tecnm.mx | apizaco.tecnm.mx



AGRADECIMIENTOS

Quiero expresar mis más sinceros agradecimientos a las personas que han sido clave y quienes han acompañado este proceso con su apoyo, comprensión y perseverancia.

En primer lugar, quiero expresar mis más profundos agradecimientos a mis padres. A ellos, por ser mi fuente constante de inspiración y fortaleza. Gracias por enseñarme el valor del esfuerzo, la disciplina y la perseverancia. Han estado a mi lado en todo momento, ofreciendo su apoyo emocional y económico, sin esperar nada a cambio permitiéndome llegar hasta aquí. Este logro es el reflejo de su sacrificio y compromiso, me siento inmensamente agradecida por todo lo que me han dado. No tengo palabras suficientes para expresar lo que significan para mí, pero este proyecto es en parte para ustedes.

A mi hermana, a quien también le debo un agradecimiento especial. Gracias por ser mi compañera en esta aventura, por brindarme el apoyo en los momentos difíciles y por siempre estar a mi lado. Agradezco profundamente sus palabras y su capacidad para escucharme, su apoyo ha sido esencial para mantener mi mente clara y motivada.

A mi asesor de proyecto “Juan Ramos Ramos”, quiero darle las gracias de manera especial por su dedicación, paciencia y conocimiento. Gracias por guiarme con su experiencia, por brindarme su tiempo y por ayudarme a superar los desafíos que se presentaron a lo largo de este proyecto. Su visión crítica y constructiva fue fundamental para mejorar mi trabajo, y sin su orientación. Aprecio profundamente su compromiso y profesionalismo, así como su disposición para ayudarme en cada detalle por más pequeño que fuera.

A todas estas personas que han estado conmigo en este camino, mi más sincero agradecimiento. Este proyecto no solo es un reflejo de mi trabajo, sino también del esfuerzo y la dedicación de quienes han creído en mí y me han apoyado.

RESUMEN

Este proyecto tiene como objetivo el desarrollo de un sistema para la capacitación docente en el TecNM Campus Apizaco, proporcionando una plataforma eficiente y estructurada para la gestión de cursos dirigidos al personal académico. El enfoque principal del desarrollo se centra en el backend, utilizando tecnologías modernas y herramientas que optimizan el almacenamiento, procesamiento y presentación de la información.

Para el desarrollo del backend, se utilizó el framework Django, escrito en Python, que permite una gestión eficiente de bases de datos, autenticación de usuarios y generación de reportes. Adicionalmente, se integraron tecnologías complementarias como JavaScript, que facilita la interacción con el sistema desde el lado del cliente.

En cuanto al almacenamiento de datos, se empleó SQLite, una base de datos ligera y eficiente, junto con el ORM de Django para manejar las consultas y transacciones de forma estructurada. Para la gestión y administración de la base de datos, se utilizó DBeaver, una herramienta que facilita la visualización y manipulación de los datos almacenados.

El control de versiones se llevó a cabo mediante Git, lo que permitió un desarrollo colaborativo, asegurando la trazabilidad de los cambios y la integración continua del código.

Además, se incorporaron diversas librerías y paquetes de Python para mejorar la funcionalidad del sistema:

- pip pillow: utilizada para el procesamiento y manipulación de imágenes dentro del sistema.
- pip reportlab: empleada para la generación de documentos en PDF, permitiendo la creación de reportes automatizados sobre el desempeño y progreso de los docentes en los cursos de capacitación.

- requirements.txt: archivo donde se registraron todas las dependencias del proyecto para facilitar su instalación y despliegue en otros entornos.

El sistema está diseñado para facilitar la administración y seguimiento de los cursos de capacitación docente, ofreciendo funcionalidades como:

- Registro y autenticación de usuarios: los docentes pueden registrarse en la plataforma y acceder con credenciales únicas.
- Gestión de cursos: permite la creación, edición y eliminación de cursos de capacitación, con detalles como temario, duración y disponibilidad.
- Seguimiento del progreso: los docentes pueden visualizar su avance en los cursos.
- Generación de reportes en PDF: utilizando la librería ReportLab, el sistema genera informes detallados sobre el rendimiento de los participantes en los cursos.

Este sistema representa una herramienta clave para la mejora del desarrollo profesional del cuerpo docente en el TecNM Campus Apizaco, asegurando una gestión eficiente de los procesos de capacitación. Gracias a su arquitectura basada en Django y el uso de tecnologías modernas, se garantiza una plataforma estable, escalable y de fácil mantenimiento.

ÍNDICE GENERAL

AGRADECIMIENTOS	
RESUMEN	
I. GENERALIDADES DEL PROYECTO	1
Introducción	1
Descripción de la empresa.....	2
Misión	2
Visión.....	2
Área de trabajo	3
Planteamiento del problema	3
Objetivos.....	4
Objetivo general.....	4
Objetivos específicos.....	4
Justificación	5
II. MARCO TEÓRICO	5
Marco teórico del proyecto.....	5
¿Qué es Backend?	6
Teams (Comunicación de trabajo)	10
Herramientas de productividad y análisis	10
Draw.io	10
DrawSQL	12
Herramientas de Desarrollo	13
Framework-Django	13
Lenguajes de programación.....	17

Python	17
Base de Datos y ORM	18
¿Qué es una base de datos?	18
ORM	21
SQLite.....	21
Dbeaver	23
Tecnología de control de versiones	24
Git	25
Librerías	27
Requirements.txt.....	27
Pip pillow	28
Pip reportlab	28
III. DESARROLLO DEL PROYECTO	29
Proceso de desarrollo	29
Desarrollo del software (CAPACITA-TEC).....	30
1. Análisis de información	30
2. Creación del entorno (Framework – Django)	32
3. Repositorio de GitHub.....	37
4. Configuración de las aplicaciones del proyecto	39
5. Instalación de librerías	41
IV.RESULTADOS	43
1. Análisis de requerimientos	43
2. Definición de la estructura de modelos y base de datos	45
3. Gestión de vistas y rutas (URLs)	50
4. Personalización del panel de administración (admin)	52

5.	Registro e inicio de sesión	53
6.	Roles de usuario	55
7.	Proceso de autenticación en vistas.....	57
8.	Permisos de vistas para cada usuario	57
9.	Documentos en PDF.....	60
IV.	CONCLUSIONES.....	63
VI.	COMPETENCIAS DESARROLLADAS Y/O APLICADAS.....	64
VII.	FUENTES DE INFORMACIÓN.....	65
VIII.	ANEXOS.....	67

ÍNDICE ILUSTRATIVO

Ilustración 1 Logo oficial del Tecnológico “Campus Apizaco”	2
Ilustración 2 Grafico de que es backend	7
Ilustración 3 Logo "Teams"	10
Ilustración 4 Logo Draw.io	10
Ilustración 5 Logo DrawSQL	12
Ilustración 6 Logo Django.....	13
Ilustración 7 Logo Python.....	17
Ilustración 8 Grafico de Base de Datos	19
Ilustración 9 Logo SQLite	21
Ilustración 10 Logo DBeaver	23
Ilustración 11 Tecnología de control de versiones	24
Ilustración 12 Logo Git	25
Ilustración 13 Análisis de Base de Datos CAPACITA-TEC	32
Ilustración 14 Creación del entorno virtual	33
Ilustración 15 Creación de entorno virtual en carpeta de proyecto	33
Ilustración 16 Activación de entorno virtual	33
Ilustración 17 Instalación de Django.....	33
Ilustración 18 Implementación de instalación Django	34
Ilustración 19 Creación de proyecto en carpeta de trabajo	34
Ilustración 20 Ejecución del servidor de desarrollo	34
Ilustración 21 Acceso al navegador Django	35
Ilustración 22 Panel del sitio de administrador Django (Inicio de sesión).....	36
Ilustración 23 Panel del sitio de administrador Django (Logueo aceptado).....	36
Ilustración 24 Generación del archivo	37
Ilustración 25 Creación de archivo. gitignore	38
Ilustración 26 Panel de repositorio Git.....	39
Ilustración 27 Configuración de las aplicaciones.....	39
Ilustración 28 Registro de aplicación en Settings.py	40
Ilustración 29 Ejecución de migraciones	41

Ilustración 30 Instalación de librería ReportLab	42
Ilustración 31 Instalación de librería Pillow.....	42
Ilustración 32 Demostración exitosa de la descarga de librerías	42
Ilustración 33 Fuente Monserrat descargada y alojada.....	43
Ilustración 34 Diagrama de procedimiento capacitación docente	44
Ilustración 35 Diagrama de casillas CAPACITA-TEC	45
Ilustración 36 Estructura final	46
Ilustración 37 Creación del modelo instructor -app catálogos-.....	47
Ilustración 38 Registro de los modelos en admin.py	47
Ilustración 39 Panel de administración de Django	49
Ilustración 40 Diagrama final (Entidad-Relación)	50
Ilustración 41 views.py de app plan de capacitación.....	51
Ilustración 42 urls.py de app plan capacitación	51
Ilustración 43 Estructura de la carpeta principal y de la app	52
Ilustración 44 Organización del urls.py principal del proyecto	52
Ilustración 45 Configuración de AbstractUser	53
Ilustración 46 Panel del admin	53
Ilustración 47 Diagrama de flujo “Registro de cuenta nueva”	54
Ilustración 48 Diagrama de flujo del proceso de autenticación (Login)	55
Ilustración 49 Asignación de roles utilizados en “Login y Registro de cuenta” - models.py, app Usuarios-.....	56
Ilustración 50 Registro en el admin.py -App Usuarios-.....	56
Ilustración 51 Roles de usuario asignados en el panel de administración	57
Ilustración 52 Asignación de @login_required	57
Ilustración 53 Asignación de permisos de cada rol -app.py de cada app-	58
Ilustración 54 Filtros de acciones que puede realizar el usuario -app.py de la app eventos-	59
Ilustración 55 Visualización del panel de administración los permisos de cada rol.....	59
Ilustración 56 Asignación de @permission_required	60
Ilustración 57 Asignación de librerías de PDF.....	61
Ilustración 58 views.py para generación del PDF	62

Ilustración 59 Ficha técnica.....	67
Ilustración 60 Oficio de comisión	68
Ilustración 61 Constancia de acreditación del curso	69
Ilustración 62 Pase de lista	70
Ilustración 63 Curriculum vitae del Instructor	71
Ilustración 64 Encuesta de opinión	72
Ilustración 65 Documento “Programa de institución.....	73

I. GENERALIDADES DEL PROYECTO

Introducción

El Tecnológico Nacional de México (TECNM) Campus Apizaco, como parte de su compromiso con la mejora continua de la calidad educativa, ha identificado la necesidad de contar con un sistema integral que facilite la gestión y organización de los procesos relacionados con la capacitación docente. Estos procesos, que abarcan desde la programación de cursos hasta el seguimiento y evaluación de los docentes, deben ser manejados de manera eficiente para garantizar su efectividad y alineación con las necesidades académicas y pedagógicas de la institución.

La importancia de contar con una plataforma centralizada para la gestión de la capacitación docente radica en la optimización de los recursos, la mejora en la administración del tiempo y, sobre todo, en la creación de un ambiente que favorezca el aprendizaje y el desarrollo profesional continuo de los docentes. Con ello, se busca no solo la actualización constante de los educadores en cuanto a los avances pedagógicos, tecnológicos y científicos, sino también la alineación de la capacitación con los objetivos institucionales del TECNM Campus Apizaco.

Este proyecto tiene como objetivo el desarrollo del backend de un sistema integral de gestión de la capacitación docente que permita a los administradores gestionar los cursos, programas de formación, y mantener un registro detallado de la participación y los avances de los docentes. A través de este sistema, los encargados de la gestión educativa podrán crear, modificar y consultar información relevante sobre las actividades de capacitación, mientras que los docentes podrán acceder a los contenidos de los cursos, registrarse, realizar seguimientos y evaluar su desempeño.

El desarrollo de este backend se enfocará en proporcionar una solución robusta, segura y eficiente, capaz de integrar distintos módulos relacionados con la gestión de la capacitación. A través del uso de tecnologías modernas de desarrollo web y programación, se buscará optimizar el acceso y procesamiento de datos, garantizando

que el sistema sea fácil de usar, escalable y compatible con las necesidades futuras del TECNM Campus Apizaco.

Descripción de la empresa

Tecnológico Nacional de México “Campus Apizaco”.



Ilustración 1 Logo oficial del Tecnológico “Campus Apizaco”

Dirección: Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec. Municipio de Tzompantepec, Tlaxcala, México. C.P. 90491.

Misión

Formar profesionistas que sean líderes visionarios, emprendedores, con capacidad de generar soluciones disruptivas, innovando en la ciencia y la tecnología, comprometidos con el desarrollo económico, responsabilidad social teniendo una visión sostenible y sustentable.

Visión

Ser una institución de educación superior tecnológica de vanguardia, formadora de ciudadanos del mundo, con conocimientos intercultural, respetando la diversidad social.

Valores

- Historias: Se demuestra día a día a través de la transparencia con la ejecución de las actividades laborales cotidianas.

- Responsabilidad: Es la actitud que caracteriza al personal en el cumplimiento de su tarea con eficacia y eficiencia en beneficio de la comunidad.
- Lealtad: Es la actitud que caracteriza al personal del Instituto Tecnológico de Apizaco, aplica sus conocimientos y habilidades en el logro de las metas institucionales, sin olvidar en ningún momento el lema:

“Pensar para servir, servir para triunfar ®”

Área de trabajo

Departamento de Sistemas y Computación

Planteamiento del problema

El proceso de gestión de la capacitación docente en el Tecnológico Nacional de México (TECNM) Campus Apizaco enfrenta varios desafíos que impactan tanto en la eficiencia administrativa como en la experiencia de los docentes. Uno de los principales problemas es la falta de centralización en la gestión de la capacitación, ya que actualmente se utilizan métodos manuales o sistemas dispersos que dificultan la organización y el acceso rápido a la información relacionada con los cursos y programas de formación. Esta dispersión no solo genera pérdida de tiempo, sino también una mayor probabilidad de errores en la planificación y ejecución de las actividades de capacitación.

A esto se suma la inaccesibilidad de la información para los docentes. En muchos casos, los profesores tienen dificultades para conocer en tiempo real qué programas de formación están disponibles, cuándo se imparten, cuáles son los requisitos y materiales necesarios, entre otros detalles. Esta falta de acceso oportuno a la información puede resultar en una baja participación en las actividades formativas, así como en una experiencia educativa para los docentes, quienes, al no contar con una plataforma organizada, se ven limitados en su capacidad de aprovechamiento de las oportunidades de capacitación.

Otro reto importante es la dificultad para el seguimiento y la evaluación del desempeño docente. Sin una plataforma centralizada que registre y analice de forma continua la

participación de los docentes en los cursos, se dificulta el monitoreo de su progreso y la medición de la efectividad de los programas de capacitación. Este vacío en el seguimiento impide a los responsables de la gestión educativa tomar decisiones informadas sobre qué áreas necesitan refuerzo, y cómo mejorar la calidad de los programas de formación.

Objetivos

Objetivo general

Desarrollar un sistema integral de gestión del backend para la capacitación docente en el TECNM Campus Apizaco, que permita centralizar, automatizar y optimizar los procesos relacionados con la planificación, seguimiento y evaluación de los programas de formación docente, mejorando la eficiencia administrativa, la accesibilidad de la información y la calidad de la formación recibida por los docentes.

Objetivos específicos

- Centralizar la gestión de la capacitación docente: Crear una plataforma que permita almacenar y gestionar toda la información relacionada con los programas de capacitación, facilitando el acceso a los administradores y docentes, y mejorando la organización de los procesos.
- Automatizar los procesos administrativos: Desarrollar funcionalidades que automaticen la inscripción de docentes a los cursos, la asignación de recursos (aulas, materiales, equipos), la emisión de constancias y la generación de reportes de participación, reduciendo errores y mejorando la eficiencia operativa.
- Facilitar el seguimiento y evaluación del desempeño docente: Implementar herramientas que permitan monitorear el progreso de los docentes durante su capacitación, generando reportes.

- Optimizar la asignación de recursos: Crear módulos que permitan gestionar de manera eficiente los recursos necesarios para los cursos de capacitación, asegurando su disponibilidad y una distribución adecuada según la demanda.

Justificación

La gestión de la capacitación docente en el TECNM Campus Apizaco enfrenta varios retos que afectan la eficiencia administrativa y la calidad educativa. La falta de un sistema centralizado y automatizado para gestionar los programas de formación, así como la ausencia de un seguimiento adecuado del desempeño docente, genera ineficiencias, sobrecarga de trabajo y dificultades para la toma de decisiones informadas. Además, la dispersión de la información y la falta de seguridad en el manejo de los datos personales de los docentes agravan estos problemas.

La implementación de un sistema integral de gestión de la capacitación docente solucionará estos problemas al centralizar la información, automatizar los procesos administrativos y mejorar la accesibilidad de los docentes a los programas de formación. Esto permitirá una mayor eficiencia operativa, un seguimiento adecuado del desempeño docente, y la creación de programas formativos más alineados con las necesidades reales. Además, garantizará la protección de la información sensible, mejorando la transparencia y la confianza de los docentes en el sistema. En definitiva, este proyecto contribuirá a optimizar los recursos, mejorar la calidad de la capacitación y fortalecer la toma de decisiones estratégicas, beneficiando tanto a los docentes como a la institución en su conjunto.

II. MARCO TEÓRICO

Marco teórico del proyecto

El marco teórico de este proyecto se centra en la importancia de la gestión eficiente de la capacitación docente y cómo la implementación de un sistema integral puede mejorar los procesos relacionados. La capacitación continua de los docentes es fundamental

para mantener la calidad educativa, ya que influye directamente en su desempeño y en la formación de los estudiantes. Un sistema de gestión bien diseñado permite organizar y coordinar los programas de formación, facilitando la planificación, seguimiento y evaluación de los cursos, lo que contribuye a mejorar la eficiencia administrativa y optimizar los recursos.

La automatización de los procesos administrativos juega un papel crucial en este contexto, ya que permite agilizar tareas repetitivas como la inscripción de docentes, la asignación de recursos y la generación de reportes. Esto no solo reduce errores humanos, sino que también libera tiempo y recursos para que los administradores se concentren en tareas estratégicas. Además, los sistemas de gestión de bases de datos permiten centralizar toda la información relacionada con la capacitación docente, lo que facilita su acceso y análisis, contribuyendo a una toma de decisiones más informada y precisa.

La seguridad en el manejo de los datos es otro aspecto fundamental en el diseño de un sistema de gestión, ya que se deben proteger los datos personales y académicos de los docentes. Es esencial implementar medidas de seguridad como el cifrado de información y el control de acceso, cumpliendo con las normativas legales sobre protección de datos. Esto garantiza la privacidad y la integridad de la información, lo que a su vez aumenta la confianza de los usuarios en el sistema.

En conjunto, la centralización de la información, la automatización de procesos, la seguridad en el manejo de datos y el análisis basado en datos son elementos clave para desarrollar un sistema de gestión de la capacitación docente que optimice los recursos, mejore la eficiencia y fortalezca la calidad educativa.

¿Qué es Backend?

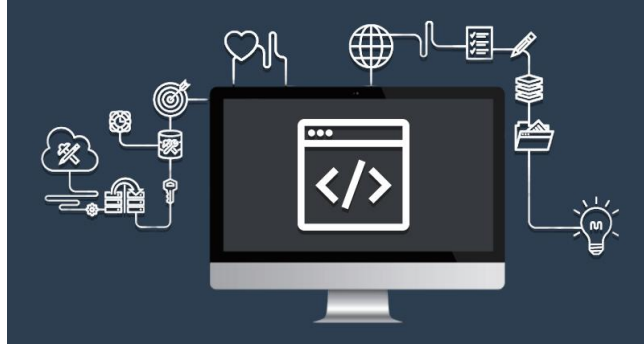


Ilustración 2 Grafico de que es backend

*Jesús. (2023, 16 enero). Backend: ¿Qué es? ¿Cómo se aprende? Tutoriales Dongee.
<https://www.dongee.com/tutoriales/backend-que-es-como-se-aprende/>*

Backend, conocido como “el lado del servidor”, se refiere a la parte de una aplicación web o sitio web que no es visible para el usuario final, pero que es esencial para su funcionamiento.

En él se configuran todos los aspectos lógicos de una página web o aplicación; abarca la lógica, el almacenamiento de datos y las funciones de seguridad necesarias para que una aplicación funcione correctamente y sea fiable, de tal forma que todas las acciones solicitadas en la página web sean ejecutadas de manera correcta.

En pocas palabras, el backend es la programación de todo lo que el usuario final no ve.

¿Pero cuáles son las principales responsabilidades del backend?

- **Gestión de datos:** El backend se encarga de la interacción con la base de datos, lo que implica el almacenamiento, recuperación y manipulación de datos. Esto puede incluir la creación, lectura, actualización y eliminación de registros en la base.
- **Lógica de negocio:** Contiene la lógica y las reglas que definen cómo funciona la aplicación. Esto incluye procesos como la autenticación y autorización de usuarios, la gestión de pedidos en una tienda en línea, entre otros.
- **Seguridad:** El backend es responsable de implementar medidas de seguridad para proteger los datos y a la aplicación contra amenazas y ataques cibernéticos. Esto implica la autenticación de usuarios, la autorización de acceso y la encriptación de datos.

- Integraciones: Muchas aplicaciones web necesitan interactuar con otros sistemas y servicios, como APIs de terceros, servicios de pago, servicios de correo electrónico, redes sociales, entre otros, y el backend facilita estas integraciones.
- Rendimiento y escalabilidad: Implica la gestión eficiente de los recursos del servidor y la optimización del código. El backend se encarga de optimizar el rendimiento de la aplicación y permitir que esta se pueda escalar para manejar un mayor número de usuarios y cargas de trabajo.
- Mantenimiento: En el backend se realizan las actualizaciones y correcciones de errores de la aplicación. Los desarrolladores pueden mantener y mejorar ciertos aspectos de la aplicación sin afectar la experiencia del usuario final.
- Administración y monitoreo: Proporciona herramientas para administrar y supervisar el funcionamiento de la aplicación, lo que permite a los desarrolladores realizar diagnósticos y tomar medidas correctivas cuando sea necesario.

El backend es una parte esencial de cualquier aplicación web, ya que maneja todas las operaciones detrás de escena que permiten que la aplicación funcione de manera efectiva y segura. Y trabaja en conjunto con el frontend, que es la parte de la aplicación que interactúa directamente con el usuario, para brindar una experiencia completa y funcional.

Componentes del backend

El backend de una aplicación web o de software incluye todos los componentes que no son visibles para el usuario, estos trabajan juntos para proporcionar funcionalidad, gestionar datos y procesar solicitudes del usuario. Entre los componentes principales del backend se encuentran:

- Servidores: corresponde a una máquina física o virtual que recopila, aloja, procesa y gestiona los recursos necesarios para ejecutar una aplicación web. Los servidores al recibir las peticiones de los usuarios ejecutan la lógica necesaria y devuelven las respuestas a través de un protocolo de comunicación, normalmente

HTTP. Los desarrolladores web pueden programar diferentes servidores para ejecutar específicas acciones de código según las entradas recibidas.

- **Lógica de aplicación:** es la secuencia de operaciones que los programadores codifican en el backend para terminar las tareas. Por esta razón los sitios web contienen funciones y algoritmos, permitiendo que se produzca información y acciones a partir del comportamiento de los usuarios dentro de una plataforma. Los especialistas en backend crean una lógica que se ejecuta casi exclusivamente en servidores, interpretando datos y produciendo salidas.
- **Frameworks:** son las guías que se usan al estructurar el código, lógica y otros aspectos relacionados con la arquitectura web, son plantillas para el backend, que utilizan los programadores para que la escritura y edición del código de su servidor se les faciliten. Algunos marcos incluyen bibliotecas de datos y herramientas que brindan acceso a segmentos funcionales del código.
- **Bases de datos:** contienen la información a la que acceden los servidores y así completar funciones directas del sitio web, además disponen de opciones para clasificar la información a la que acceden los usuarios. Por lo general estos datos son números enteros, caracteres y matrices.
- **APIs:** permiten que los programas de software establezcan comunicación con otros servidores y bases de datos con el objetivo de intercambiar información. Su función es simplificar la presentación de toda la información almacenada en múltiples bases de datos en un solo lugar.

Estos componentes trabajan juntos para crear la infraestructura y la funcionalidad necesarias en el backend de una aplicación web o de software. La elección y configuración de estos componentes dependen de las necesidades específicas de la aplicación y su arquitectura.

Tener un backend en una aplicación o sistema web proporciona una base sólida para la gestión de datos, la seguridad, la escalabilidad y la eficiencia. Además de que permite una separación clara entre la lógica de negocio y la interfaz de usuario, lo que facilita la administración y el desarrollo a largo plazo de la aplicación. [1]

Teams (Comunicación de trabajo)



Ilustración 3 Logo "Teams"

(s. f.). ¿Qué es y cómo funciona Microsoft Teams? | NFON. <https://blog.nfon.com/es/que-es-y-como-funciona-microsoft-teams/>

Microsoft Teams es una plataforma que se usó para la implementación de la metodología Scrum, ya que facilita la comunicación, colaboración y organización del equipo. Entre sus beneficios destacan la comunicación efectiva mediante canales dedicados y reuniones de Daily Scrum, la gestión de tareas con herramientas como Planner y Trello, y la colaboración en documentos mediante SharePoint y OneDrive. [2]

Herramientas de productividad y análisis

Draw.io



Ilustración 4 Logo Draw.io

SaaS Rank. (s.f). Draw.io - SAAS rank. <https://saasrank.es/producto/draw-io/>

Draw.io es una herramienta de diagramación, de diagrama de flujo, de proceso, entre otras muchas funciones. Es una herramienta gratuita con la que se puede dibujar cualquier tipo de mapas mentales, mapas conceptuales, esquemas o diferentes representaciones gráficas, como diagrama de jerarquía o conjuntos.

Con Draw.io nuestros alumnos tienen una herramienta gratuita en la que pueden realizar los esquemas de sus unidades, enriquecerlos con imágenes, iconos y enlaces. Pueden colaborar y cooperar gracias a la sencillez para compartirlos.

Es una herramienta que en principio puede estar pensada para uso profesional, pero realmente se puede usar si se tiene un nivel de destrezas mínimas. Simple, sencilla e intuitiva. Sus principales ventajas son:

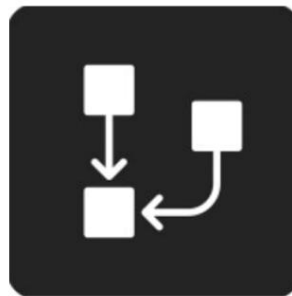
Integración: se puede utilizar en versión online o de escritorio. En versiones online se puede vincular a diferentes cuentas y guardar los trabajos en sistemas de almacenamiento en la nube (Google Drive, OneDrive, Dropbox) o descargarlos. Además, permite importar diagramas de otras aplicaciones.

Plantillas y librerías: tanto para profes como para alumnos los inicios a veces son duros, por lo que la herramienta dispone de una serie de plantillas organizadas por tipos que hacen el trabajo más sencillo y más rápido. Además, consta de una serie de librerías con una gran cantidad de formas, líneas, dibujos, iconos, que se pueden ir ampliando. Y si estas opciones no son suficientes, se pueden importar o crear las que se necesiten.

Sencillez de uso: la interfaz de Draw.io es sencilla, cuenta con las herramientas básicas para poder desarrollar la experiencia, lo que facilita la utilización por parte de los alumnos, ya que se enfrentan a instrumentos conocidos. Utiliza el sistema drag and drop (arrastrar y soltar), con el cual rápidamente se pueden llevar a la zona de trabajo formas, flechas, imágenes, etc. Otra de las características reseñables es el área de trabajo, que se va extendiendo a medida que el diagrama, representación o dibujo crece, sin necesidad de incluir páginas, ya que lo hace la aplicación. Una gran ventaja es el guardado automático, así nunca se pierde el trabajo ante cierres inesperados. Y, por último, algo que le resulta muy atractivo a nuestros alumnos, es la facilidad para exportar su trabajo, se puede guardar fácilmente a cualquier herramienta de edición de Google o Microsoft y se puede transformar en formato imagen, PDF u otros.

Draw.io es una aplicación que se utiliza para completar otras aplicaciones que no tengan zona para dibujar diagramas. Lo interesante es trabajar dos objetivos al mismo tiempo, por un lado, abordar la competencia digital de nuestros alumnos y, por otro, organizar y reforzar las ideas que están tratando en un determinado tema. Se debe pensar en que la herramienta no se utiliza para una asignatura en concreto, sino que podemos abordar el temario de cualquier asignatura. [3]

DrawSQL



DrawSQL



Ilustración 5 Logo DrawSQL

DrawSQL. (s.f.). DrawSQL logo <https://www.bestaffiliateprograms.io/affiliate-programs/drawsql>

DrawSQL es una herramienta colaborativa para planificar y gestionar visualmente esquemas de bases de datos. Con una interfaz intuitiva basada en JointJS, permite a los equipos crear diagramas ER, colaborar en tiempo real e integrar diagramas en flujos de trabajo de desarrollo, lo que resulta ideal para equipos de software y arquitectos de bases de datos.

Entre sus características más destacadas se encuentran:

- Generación automática de migraciones: DrawSQL facilita la creación de migraciones para Laravel a partir de los diagramas generados, permitiendo construir el esquema de la base de datos con mayor facilidad. También es compatible con otros frameworks.

- Colaboración en equipo: Los usuarios pueden trabajar en conjunto en el mismo diagrama de base de datos, mejorando la eficiencia en proyectos compartidos.
- Opciones de privacidad: Es posible crear diagramas públicos, accesibles mediante un enlace compartible, o diagramas privados que solo pueden ver los miembros del equipo.
- Modo de presentación: DrawSQL ofrece una funcionalidad de presentación para mostrar diagramas en reuniones, facilitando la comprensión de personas no técnicas.
- Compatibilidad con múltiples DBMS: Soporta sistemas de gestión de bases de datos como MySQL, PostgreSQL y Microsoft SQL Server.
- Uso de plantillas: Los usuarios pueden clonar y reutilizar diagramas existentes como plantillas en distintos proyectos.
- Visualización en redes sociales: Los diagramas públicos pueden mostrarse con atractivas tarjetas de vista previa en plataformas sociales.
- DrawSQL es una herramienta versátil que mejora la visualización y gestión de esquemas de bases de datos, facilitando tanto la colaboración como la documentación de proyectos. [4]

Herramientas de Desarrollo

Framework-Django



Ilustración 6 Logo Django

Django es un software que puede utilizar para desarrollar aplicaciones web de forma rápida y eficiente. La mayoría de las aplicaciones web tienen varias funciones comunes, como la autenticación, la recuperación de información de una base de datos y la administración de cookies. Los desarrolladores tienen que codificar una funcionalidad similar en cada aplicación web que escriban. Django facilita su trabajo al agrupar las diferentes funciones en una gran colección de módulos reutilizables, llamada marco de aplicación web. Los desarrolladores utilizan el marco web de Django para organizar y escribir su código de manera más eficiente y reducir significativamente el tiempo de desarrollo web.

- ¿Por qué los desarrolladores web eligen Django?

Hay varios marcos web en el mercado. Django se escribió en el lenguaje Python y es uno de los muchos marcos web de Python. Sin embargo, los desarrolladores a menudo prefieren el marco web Django a otros por las siguientes razones:

- Velocidad de desarrollo

El marco Django está bien organizado y es fácil de instalar y aprender, por lo que puede comenzar en cuestión de horas. Los diseñadores de Django crearon el marco para implementar rápidamente cualquier arquitectura web en el código. Permite el desarrollo rápido y el diseño limpio y pragmático. Puede escribir código en solo unas pocas líneas porque Django proporciona una estructura lista para usar para varias tareas comunes de desarrollo web, como:

1. Autenticación de usuarios
2. Administración de contenido
3. Mapas del sitio
4. Fuentes RSS
5. Rentabilidad

Django es un proyecto de Python gratuito y de código abierto con una comunidad activa que revisa y mantiene el software. Una organización sin fines de lucro llamada Django Software Foundation promueve y respalda el uso y el mantenimiento de Django.

Organiza encuentros, reuniones y eventos comunitarios regulares que animan a otros desarrolladores a revisar y contribuir al proyecto Django. El resultado es un marco web de alta calidad y rico en funciones que es de uso gratuito.

- ¿Cómo funciona Django?

Cualquier aplicación web consta de dos partes: código de servidor y código de cliente. El cliente o visitante del sitio web tiene un navegador. Cuando escriben una URL en su navegador, envían una solicitud a la máquina del servidor web en la que se ejecuta la aplicación web. El servidor procesa la solicitud mediante una base de datos y envía información al cliente como respuesta. El código de cliente muestra la información al visitante como una página web.

Django administra el código para este sistema de solicitud y respuesta mediante el uso de una arquitectura Modelo-Vista-Template (MVT).

- Administración del sitio

El sitio de administración de Django facilita la provisión de una página de administración para su sitio. Los administradores del sitio pueden usar la página para crear, editar o ver los modelos de datos del sitio.

- Model

Los modelos de Django actúan como interfaz entre la base de datos y el código del servidor. Son la única fuente definitiva de información sobre sus datos. Estos modelos de datos contienen los campos y las operaciones esenciales que necesita para interactuar con la base de datos. Los modelos de Django convierten así las tablas de la base de datos en clases u objetos en el código Python. Esto se denomina “mapeo objeto-relacional”.

En general, cada modelo se asigna a una sola tabla de base de datos y tiene atributos que representan los campos de la base de datos. Si, por ejemplo, su sitio web contiene detalles de empleados, podría representarse como:

Una tabla de empleados con campos de nombres y direcciones de cada uno de ellos.

Un modelo de empleado denominado “empleado de clase” con dos atributos o campos de modelo, denominados “nombre” y “dirección”.

- Ver

Las vistas de Django procesan la solicitud utilizando los modelos. Puede escribir una función de vista para cada tipo de solicitud que los visitantes de su sitio web pueden hacer a este. Una función de vista puede tomar la solicitud como entrada y devolver una respuesta, que puede ser un código de error, una imagen, un archivo o cualquier tipo de dato.

- Plantilla

Las plantillas de Django administran la presentación de la página web en el navegador. Dado que la mayoría de las páginas web están en el lenguaje de marcado de hipertexto (HTML), puede escribir el código de la plantilla de Django en un estilo similar a HTML. Un archivo de plantilla contiene ciertos componentes:

Las partes estáticas de la salida HTML final, como imágenes, botones y encabezados. Sintaxis especial que describe cómo insertar contenido o datos dinámicos, que cambia con cada solicitud.

Los siguientes componentes conforman el sistema de plantillas de Django.

El lenguaje de plantillas

El lenguaje de plantillas es el lenguaje de programación que se utiliza para escribir el código de la plantilla HTML. Django admite su propio lenguaje de plantillas Django y una alternativa popular llamada Jinja2.

- El motor de plantillas

El motor de plantillas procesa el archivo de plantilla y crea la salida HTML final. Incluye los datos de la respuesta en esta salida.

Por ejemplo, cuando el visitante de su sitio web solicita información de empleados, la plantilla de Django rellenará la página web que ve con el encabezado de su sitio web, una tabla que contiene los nombres y las direcciones de todos los empleados y un botón que dice Siguiente.

- ¿Qué otros módulos pueden usar en Django?

Aunque la arquitectura Modelo-Vista-Template (MVT) define la estructura básica de cualquier aplicación, Django tiene varios otros módulos para mejorar su sitio web. A continuación, presentamos algunos ejemplos.

- Formularios

La mayoría de los sitios web requieren formularios para tareas como el registro y el pago, o para recopilar información del visitante del sitio web. Django proporciona muchas herramientas y bibliotecas que puede usar para administrar los formularios de su sitio web. Puede simplificar y automatizar el procesamiento de formularios y hacerlo de forma más segura que escribiendo el código por su cuenta.

Django gestiona el procesamiento de formularios de tres maneras:

1. Creación de formularios mediante la preparación y reestructuración de los datos para su visualización
2. Validación de formularios mediante la comprobación de formularios HTML en el lado del cliente
3. Procesamiento de formularios mediante la recepción de los datos enviados. [5]

Lenguajes de programación

Python



Ilustración 7 Logo Python

Python: ¡Hola, mundo! [blog.vermiip.es. https://blog.vermiip.es/primeros-pasos-con-python-hola-mundo/](https://blog.vermiip.es/primeros-pasos-con-python-hola-mundo/)

Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas. Es muy atractivo en el campo del Desarrollo Rápido de Aplicaciones (RAD) porque ofrece tipificación dinámica y opciones de encuadernación dinámicas.

Python es relativamente simple, por lo que es fácil de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes.

Por tanto, esto reduce el costo de mantenimiento y de desarrollo del programa porque permite que los equipos trabajen en colaboración sin barreras significativas de lenguaje y experimentación.

Además, soporta el uso de módulos y paquetes, lo que significa que los programas pueden ser diseñados en un estilo modular y el código puede ser reutilizado en varios proyectos. Una vez se ha desarrollado un módulo o paquete, se puede escalar para su uso en otros proyectos, y es fácil de importar o exportar.

Por otro lado, uno de los beneficios más importantes de Python es que tanto la librería estándar como el intérprete están disponibles gratuitamente, tanto en forma binaria como en forma de fuente.

Tampoco hay exclusividad, ya que Python y todas las herramientas necesarias están disponibles en todas las plataformas principales. Por lo tanto, es una opción multiplataforma, bastante tentadora para los desarrolladores que no quieren preocuparse por pagar altos costos de desarrollo.

En definitiva, es un lenguaje de programación relativamente fácil de aprender, y las herramientas necesarias están disponibles para todos de forma gratuita. Esto hace que sea accesible para casi todo el mundo. Si dispones de tiempo para aprender, conseguirás crear esos proyectos que tienes en mente. [6]

Base de Datos y ORM

¿Qué es una base de datos?



Ilustración 8 Grafico de Base de Datos

*Jesús. (2023, 16 enero). Backend: ¿Qué es? ¿Cómo se aprende? Tutoriales Dongee.
<https://www.dongee.com/tutoriales/backend-que-es-como-se-aprende/>*

Una base de datos o banco de datos es una herramienta que almacena información perteneciente a un mismo contexto, organizada y sistematizada lógicamente para su posterior recuperación, análisis o transmisión. Una biblioteca, un listado completo de clientes o el historial de las páginas visitadas en un navegador web son algunos ejemplos de bases de datos.

Las bases de datos responden al cometido de almacenar y organizar la información para poder acudir a ella posteriormente, lo que ha sido una necesidad de la humanidad desde tiempos remotos. Sin embargo, solo tras el surgimiento de la electrónica y la computación, se ha podido satisfacer a muy gran escala, almacenando enormes cantidades de datos en espacios físicos limitados, a través de su conversión en señales eléctricas o magnéticas.

Las bases de datos se pueden organizar de acuerdo a diferentes modelos y paradigmas, cada uno dotado de características, ventajas y desventajas, según su estructura, su jerarquía, su capacidad de transmisión o de interrelación, entre otros criterios posibles. Estas diferencias constituyen el campo de los modelos de base de datos y permiten el diseño y la implementación de algoritmos y otros mecanismos lógicos de gestión para administrar los datos guardado. Existen diferentes formas de clasificar las bases de datos, atendiendo a características puntuales:

Según su variabilidad

Dependiendo de cómo sean los procesos de acceso y modificación de los datos, es decir, su variabilidad, se puede distinguir entre:

- Bases de datos estáticas. Típicas de la inteligencia empresarial y otras áreas de análisis histórico, son aquellas de las cuales se puede extraer información, pero no modificar la ya existente, o sea que permiten únicamente la lectura de sus datos, no su reescritura.
- Bases de datos dinámicas. Son aquellas que además de las operaciones básicas de consulta (lectura), permiten manejar procesos de actualización, reorganización, añadidura y borrado de información.

Según su estructura

Dependiendo de cómo se encuentra organizada la información en una base de datos, se puede distinguir entre:

- Bases de datos jerárquicas. Son aquellas cuya información se dispone en estructura de árbol invertido, de manera tal que de cada nodo de información se desprenden otros nodos subsiguientes.
- Bases de datos de red. Son aquellas cuyos registros no están conectados entre sí de manera jerárquica, sino a través de enlaces en una red de información.
- Bases de datos relacionales. Son aquellas cuya información se dispone en tablas conocidas como “relaciones”, compuestas por un conjunto de campos y registros, y que se comunican entre sí mediante llaves primarias y foráneas.
- Bases de datos multidimensionales. Son aquellas cuyos datos se ordenan en una sola tabla, añadiendo un campo por cada métrica o hecho registrado.
- Bases de datos orientadas a objetos. Son aquellas que representan la información mediante objetos generados a través de un lenguaje de programación.

Big data

Se suele hablar de big data (“grandes datos”, en inglés) para referirse a las bases de datos muy voluminosas que normalmente provienen del registro de las actividades de los usuarios de la tecnología digital (datos de navegación, de ubicaciones, de compras, entre otros). Estas enormes cantidades de datos se manejan a través de sistemas

hiperespecializados de gestión de bases de datos, como SQL, Oracle o Power BI. La necesidad de manejar inmensas cantidades de datos inaugura el campo del saber de la data analytics. [7]

ORM

Sus siglas significan Object Relational Mapping (mapeo relacional de objetos) y su función es abstraer la base de datos, de modo que tú como programador puedas hacer consultas sin conocer SQL, y en su lugar, seguir usando el lenguaje de programación que ya conoces.

Lo que hace un ORM es mapear las bases de datos a objetos (por eso su nombre) y estos objetos tendrán métodos para interactuar con ellos y hacer el CRUD sin comunicarte directamente a la base de datos. [8]

SQLite



Ilustración 9 Logo SQLite

NuGET Gallery | SQLite. (s. f.). <https://www.nuget.org/profiles/SQLite>

SQLite es el sistema de bases de datos más extendido y utilizado del mundo. El software, que fue diseñado originalmente por D. Richard Hipp en 2000 para el Ejército de los Estados Unidos, es ahora de dominio público y, por tanto, está libre de derechos de autor. El nombre “SQLite” se compone de “SQL” (abreviatura de Structured Query Language) y “lite”, término coloquial para una versión más ligera y menos intensiva en memoria.

- ¿Cómo funciona SQLite?

SQLite no requiere una larga fase de aprendizaje, sino que puede integrarse directamente en una aplicación sin necesidad de utilizar software de servidor adicional. De este modo, la aplicación recibe automáticamente amplias funciones de base de datos sin necesidad de herramientas adicionales. Todas las tablas, referencias, enlaces y similares se alojan en un único archivo que ahorra espacio, por lo que también es posible almacenar opcionalmente este archivo en la memoria principal.

Por ello, el uso de un archivo en diferentes sistemas es muy simple. También porque cualquier dato puede almacenarse dentro de una tabla. A continuación, solo se convierten si es necesario, de modo que incluso el cambio entre sistemas con diferente orden de bytes es posible.

Como sistema de gestión de bases de datos relacionales, SQLite utiliza un principio de dos claves para identificar las entradas de una tabla y vincular diferentes tablas entre sí. Para ello se utilizan una clave primaria y una clave externa.

- Clave primaria: es un valor único que puede asignarse a una fila concreta de la tabla.
- Clave externa: se utiliza para vincular varias tablas entre sí.

- Versatilidad

SQLite no solo es compatible con la mayoría de los comandos del lenguaje SQL, sino que su compatibilidad con la mayoría de los almacenes de datos habituales es también una gran ventaja. No en vano, la biblioteca se utiliza en numerosas aplicaciones como Facebook o WhatsApp, así como en todos los sistemas operativos comunes para teléfonos de sobremesa y móviles, navegadores y muchos programas comerciales. Dado que el intercambio entre dos sistemas diferentes también es posible sin mayores problemas, numerosos desarrolladores confían en la solución como base de su software.

[9]

Dbeaver



Ilustración 10 Logo DBeaver

Platzi. (s. f.). Incrementá tu productividad con base de datos con DBeaver. <https://platzi.com/tutoriales/1566-fundamentos-db/14346-incrementa-tu-productividad-con-base-de-datos-con-dbeaver/>

DBeaver es un potente software para la gestión de bases de datos, libre y de código abierto para múltiples sistemas operativos. Aunque DBeaver nació en 2010, fruto de su popularidad en la comunidad de código abierto, ha experimentado una rápida expansión de sus características iniciales, incorporando las principales bases de datos SQL y NoSQL.

- ¿Qué funcionalidades permite el software de DBeaver?

DBeaver permite la mayoría de las funcionalidades básicas de cualquier gestor de bases de datos y mucho más. A través de su amigable interfaz podemos:

1. Crear todos los componentes de una base de datos: esquemas, tablas, disparadores, funciones, usuarios, roles, etc.
2. Realizar consultas SQL y NoSQL
3. Crear/Modificar/Eliminar registros
4. Exportar y migrar datos
5. Generar backups
6. Generar datos simulados para realizar pruebas
7. Crear diagramas del modelo entidad-relación
8. Visualización de información especial

- ¿Qué bases de datos incorpora?

Esta herramienta permite la conexión con la mayoría de bases de datos SQL de alto nivel (PostgreSQL, MySQL, , MariaDB, Oracle, SQLite) y también para bases de datos NoSQL (MongoDB, Cassandra) . Además, nos incorpora acceso a otros tipos de información como archivos CSV, DBF, JSON o XML. De esta forma, DBeaver nos permite acceder a todas nuestras bases de datos sin necesidad de cambiar continuamente de una plataforma a otra y aceptando los distintos lenguajes específicos para cada una de ellas. [10]

Tecnología de control de versiones



Ilustración 11 Tecnología de control de versiones

A. Ayerdi, DocuWare <https://start.docuware.com/es/blog/control-versiones#:~:text=Es%20una%20herramienta%20que%20organiza,historial%20de%20revis>

El control de versiones rastrea los cambios realizados en cada versión de un documento. Es una herramienta que organiza, etiqueta y conserva cada borrador hasta que se completa una versión final. Con el control de versiones, las modificaciones, quién las hizo y cuándo se produjeron se almacenan en un registro de auditoría que proporciona un historial de revisiones y actualizaciones. Es especialmente importante para los documentos que requieren una cantidad significativa de revisiones, necesitan una revisión anual o están sujetos a normativas de cumplimiento.

- ¿Por qué es tan importante el control de versiones?

En el entorno empresarial actual, es esencial poder identificar qué ha cambiado entre las versiones de un documento y asegurarse de que los usuarios siempre distribuyen o comienzan un nuevo ciclo de revisión con la más actual. El control de documentos

garantiza que sólo haya una versión maestra de cada documento y que exista un historial completo de versiones del mismo.

Implantar un sistema de control de versiones elimina el tiempo dedicado a abrir manualmente los documentos para intentar determinar cuál es el más actual. También evitarás ofrecer o recibir incidentalmente comentarios sobre el documento equivocado.
[11]

Git



Ilustración 12 Logo Git

*Drupal, V. E. S. Y. (s. f.). Trabajando con Git: fichero en el histórico de cambios. Vabadus.
<https://vabadus.es/blog/otros/trabajando-con-git-fichero-en-el-historico-de-cambios>*

Hoy en día, Git es, con diferencia, el sistema de control de versiones moderno más utilizado del mundo. Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente Linus Torvalds, el famoso creador del kernel del sistema operativo Linux, en 2005. Un asombroso número de proyectos de software dependen de Git para el control de versiones, incluidos proyectos comerciales y de código abierto. Los desarrolladores que han trabajado con Git cuentan con una buena representación en la base de talentos disponibles para el desarrollo de software, y este sistema funciona a la perfección en una amplia variedad de sistemas operativos e IDE (entornos de desarrollo integrados).

Git, que presenta una arquitectura distribuida, es un ejemplo de DVCS (sistema de control de versiones distribuido, por sus siglas en inglés). En lugar de tener un único espacio para todo el historial de versiones del software, como sucede de manera habitual en los sistemas de control de versiones antaño populares, como CVS o Subversion (también conocido como SVN), en Git, la copia de trabajo del código de cada

desarrollador es también un repositorio que puede albergar el historial completo de todos los cambios.

Además de contar con una arquitectura distribuida, Git se ha diseñado teniendo en cuenta el rendimiento, la seguridad y la flexibilidad.

- Rendimiento

Las características básicas de rendimiento de Git son muy sólidas en comparación con muchas otras alternativas. La confirmación de nuevos cambios, la ramificación, la fusión y la comparación de versiones anteriores se han optimizado en favor del rendimiento. Los algoritmos implementados en Git aprovechan el profundo conocimiento sobre los atributos comunes de los auténticos árboles de archivos de código fuente, cómo suelen modificarse con el paso del tiempo y cuáles son los patrones de acceso.

- Seguridad

Git se ha diseñado con la principal prioridad de conservar la integridad del código fuente gestionado. El contenido de los archivos y las verdaderas relaciones entre estos y los directorios, las versiones, las etiquetas y las confirmaciones, todos ellos objetos del repositorio de Git, están protegidos con un algoritmo de hash criptográficamente seguro llamado "SHA1". De este modo, se salvaguarda el código y el historial de cambios frente a las modificaciones accidentales y maliciosas, y se garantiza que el historial sea totalmente trazable.

- Flexibilidad

Uno de los objetivos clave de Git en cuanto al diseño es la flexibilidad. Git es flexible en varios aspectos: en la capacidad para varios tipos de flujos de trabajo de desarrollo no lineal, en su eficiencia en proyectos tanto grandes como pequeños y en su compatibilidad con numerosos sistemas y protocolos.

Git se ha ideado para posibilitar la ramificación y el etiquetado como procesos de primera importancia (a diferencia de SVN) y las operaciones que afectan a las ramas y las etiquetas (como la fusión o la reversión) también se almacenan en el historial de cambios. No todos los sistemas de control de versiones ofrecen este nivel de seguimiento. [12]

Librerías

Requirements.txt

Un archivo requirements.txt es un documento utilizado en proyectos de Python para especificar las librerías y sus versiones necesarias para que el programa funcione correctamente. Este archivo es esencial para la instalación y gestión de dependencias, asegurando que todos los desarrolladores y entornos de despliegue tengan las mismas versiones de las librerías.

Elementos de un archivo requirements son dependencias que aparecerán, en un principio, en el archivo requirements.txt, pues son todas las librerías, con una versión concreta, para el desarrollo del programa.

- charset-normalizer: esta es una librería que se utiliza para codificar un texto y te permite leer un texto de mejor forma.
- idna: esta es un protocolo de Python que se encarga de controlar los dominios internacionalizados.
- requests: esta es la librería requests del lenguaje de programación Python. Se utiliza para trabajar con requests y responses.
- urllib: la librería urllib forma parte de las dependencias de Python que se utilizan para manejar los enlaces o URL que aparecen en un programa de código. Esta librería tiene 4 módulos y el primero de ellos, urllib.requests, es el que emplearás para intervenir los módulos de la librería requests. Este módulo se utiliza para abrir URL o enlaces, como el método urllib.requests.open.

Ya que conoces los elementos de un archivo requirements, puedes crearlos y empezar a utilizarlos en el programa de código que estás reproduciendo. Para crear ambos ficheros, solo tienes que dirigirte a la herramienta de crear un nuevo folder de tu IDE dentro de un programa de código y nombrar el fichero.

- `pip list`: es un comando que se utiliza para conocer todas las dependencias que existen en los programas y, por lo tanto, en el fichero de requerimientos del mismo.
- `pip install -r`: es un comando que se ingresa en la consola y se utiliza para instalar el archivo `requirements.txt` desde `pip`.

Para ejecutar ambos ficheros, deberás agregar el fichero `txt` en el fichero `-dev.txt`. Esto lo haces utilizando el comando anterior de la siguiente manera: `-r.requirements.txt`.

Las dependencias de cada uno de los ficheros siempre deben estar organizadas por orden alfabético. [13]

Pip pillow

Pillow es una biblioteca de procesamiento de imágenes en Python que proporciona una amplia gama de funcionalidades para abrir, manipular y guardar imágenes en varios formatos. Es una herramienta poderosa y versátil que te permite realizar tareas como:

- Cargar imágenes desde archivos.
- Modificar el tamaño y la resolución de una imagen.
- Aplicar filtros y efectos a las imágenes.
- Realizar operaciones de recorte y rotación.
- Guardar imágenes en diferentes formatos, como JPEG, PNG, BMP, etc.

Pillow (PIL) es una biblioteca esencial para cualquier proyecto que involucre procesamiento de imágenes en Python. Con su amplia gama de funcionalidades y su facilidad de uso, puedes realizar tareas de manipulación de imágenes de manera eficiente y efectiva. [14]

Pip reportlab

ReportLab es un toolkit de código abierto para crear documentos PDF desde Python. Se trata de una librería muy extensa y con muchas funcionalidades, desde pequeños textos y figuras geométricas a grandes gráficos e ilustraciones, todo ello puede ser incluido dentro de un PDF. En este artículo estaremos sopesando sus características generales y sus principales funciones para crear este tipo de documentos.

ReportLab incluye una API de bajo nivel para generar documentos PDF directamente desde Python, y un lenguaje de plantillas de más alto nivel —similar a HTML y a los sistemas de plantilla que se emplean en el desarrollo web— llamado RML. Generalmente la segunda opción suele ser más conveniente para aquellos que deban hacer un uso exhaustivo de las capacidades de la librería al momento de generar documentos. ReportLab permite dibujar líneas, rectángulos, círculos y otras figuras de una forma sencilla. Por ejemplo, para dibujar una línea invocamos el método `line()` indicando la posición de los dos puntos del segmento: `x1, y1, x2, y2`. [15]

III. DESARROLLO DEL PROYECTO

Proceso de desarrollo

El ciclo de vida del desarrollo de software se describe por varias tareas necesarias para recrear una aplicación de software. El proceso de desarrollo pasa por varias etapas a medida que los desarrolladores agregan nuevas características y corrigen errores del software.

- Planificación

La fase de la planificación incluye normalmente tareas como análisis de procesos, estimación de tiempos. Se recopilan requisitos de varias partes interesadas, como clientes, expertos internos y externos, así como directivos para crear un documento de especificaciones con los requisitos del software. [16]

- Diseño

El diseño del software implica crear un anteproyecto para que lo sigan los desarrolladores. Esta fase suele incluir la selección de los lenguajes de programación y los marcos de trabajo que se utilizarán. [17]

- Implementación

Se codifica el producto. Se analizan los requisitos para identificar tareas de codificación más pequeñas que puedan hacerse diariamente para conseguir el resultado final.

- Pruebas

Se combinan las pruebas automáticas y manuales para comprobar si el software tiene errores. Incluye probar el software para detectar errores y comprobar si cumple con los requisitos.

- Mantenimiento

Se resuelven problemas y administra los cambios hechos en el software. Además, se supervisa el rendimiento general del sistema, experiencia de usuario para identificar nuevas mejoras en el software existente. [18]

Desarrollo del software (CAPACITA-TEC)

1. Análisis de información

Se identificaron aspectos clave a considerar, desde la elección del framework hasta las librerías necesarias para cumplir con los requerimientos del proyecto. Este análisis incluyó la evaluación de herramientas como Django, ReportLab y Pillow, asegurando que fueran adecuadas para la arquitectura y funcionalidad del sistema.

Además, durante esta etapa, se definieron los cinco roles principales de usuarios del software, estableciendo claramente las funciones específicas que cada uno desempeñaría:

- a) **Jefe de Capacitación:** Responsable consultar los diferentes catálogos de información. Supervisa los registros de los cursos entrantes y validación de estos, gestiona y mantiene actualizado los registros de los eventos a impartir y evalúa el

progreso y desempeño de los docentes, así como el de los instructores, asegurando el cumplimiento de los objetivos establecidos.

- b) **Jefe Académico:** Encargado de supervisar los catálogos de información, así como de estructurar, planificar y coordinar los cursos a registrar, garantizando la calidad del contenido impartido. Además, tiene la responsabilidad de emitir y entregar los oficios correspondientes a cada docente, asegurando que se cumplan con los requisitos académicos del docente.
- c) **Subdirección Académica:** Desempeña un rol estratégico enfocado en la toma de decisiones clave para la mejora de los cursos impartidos. Este rol incluye la validación de la información de los cursos registrados, así como la capacidad de consultar y gestionar la información contenida en los catálogos existentes.
- d) **Instructor:** Responsable de impartir los cursos asignados, con acceso a la información de los docentes inscritos en cada curso, asegurando así un seguimiento efectivo del aprendizaje de los participantes. Además, el instructor tiene la tarea de evaluar a los participantes para garantizar que adquieran las habilidades necesarias. Este rol también incluye la generación de una ficha técnica donde puede registrar previamente los detalles y objetivos del curso que impartirá. Asimismo, el instructor podrá consultar y descargar las constancias generadas por los cursos que ha impartido exitosamente.
- e) **Docente:** Participa en el rol de estudiante, con la capacidad de inscribirse en los cursos de su interés. El docente puede realizar una encuesta de satisfacción para evaluar diversos aspectos, como la calidad del instructor, los temas tratados y la dinámica general del curso. Además, tiene acceso a sus constancias emitidas por los cursos que haya finalizado con una calificación aprobatoria, lo que le permite llevar un registro de su desarrollo profesional.

Como parte de este proceso, también se definieron las vistas que manejaría cada rol dentro del sistema, diseñando interfaces específicas que permitieran a cada usuario interactuar fácilmente con las funciones correspondientes a su rol. Además, se llevó a cabo la definición detallada de los atributos que iba a contener cada tabla de la base de datos, asegurando que la estructura estuviera alineada con los requerimientos funcionales y permitiera un manejo eficiente de la información.

Estas definiciones fueron esenciales para estructurar el software de manera que atendiera las necesidades específicas de cada usuario, optimizando tanto la funcionalidad como la experiencia personalizada de uso.

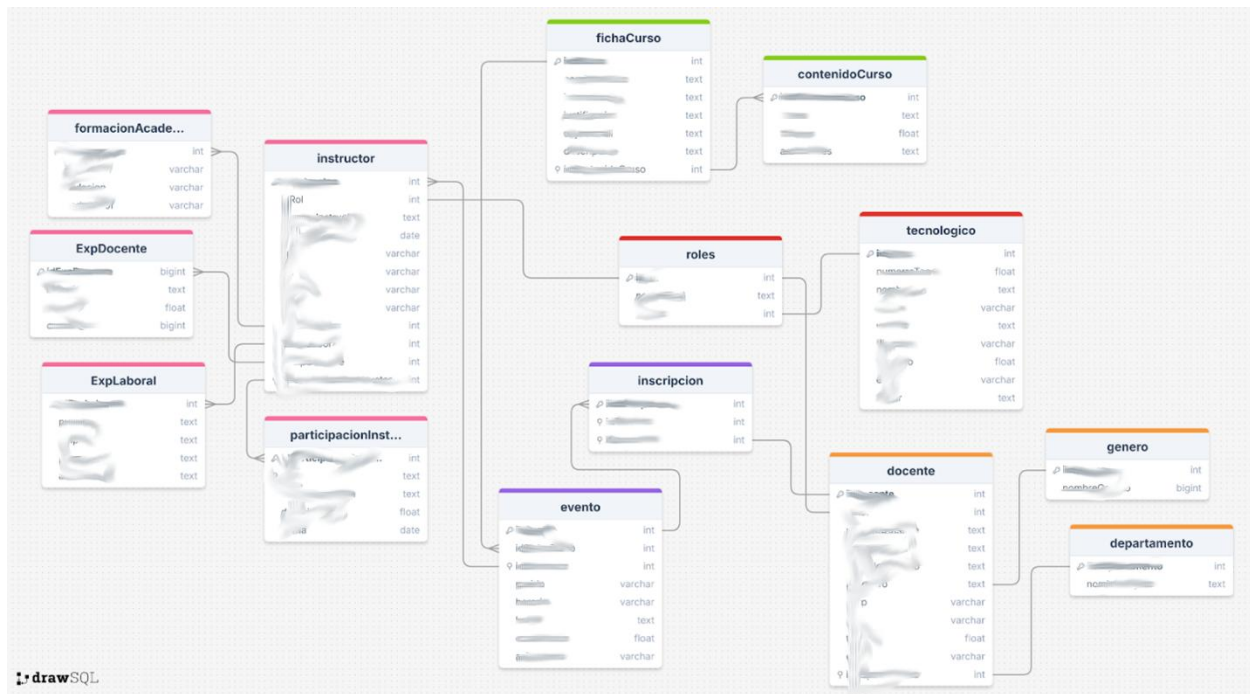


Ilustración 13 Análisis de Base de Datos CAPACITA-TEC

Fuente: Creación propia

2. Creación del entorno (Framework – Django)

Al crear un proyecto de software es fundamental la ejecución de un entorno virtual, ya que garantizará la correcta ejecución de las actividades vinculadas al desarrollo y pruebas del backend, permitiendo el funcionamiento del sistema.

Requisito fundamental: Tener instalado Python

1. Creación del entorno virtual

- Crear una carpeta en donde ira el proyecto

```
C:\>mkdir CapacitacionDoc
```

Ilustración 14 Creación del entorno virtual

Fuente: Creación propia

- Crear un entorno virtual en la carpeta del proyecto

```
C:\>cd CapacitacionDoc  
C:\CapacitacionDoc>python -m venv enCap
```

Ilustración 15 Creación de entorno virtual en carpeta de proyecto

Fuente: Creación propia

- Activar el entorno virtual

```
C:\CapacitacionDoc>cd enCap  
C:\CapacitacionDoc\enCap>cd Scripts  
C:\CapacitacionDoc\enCap\Scripts>activate  
(enCap) C:\CapacitacionDoc\enCap\Scripts>
```

Ilustración 16 Activación de entorno virtual

Fuente: Creación propia

- Instala Django

```
(enCap) C:\CapacitacionDoc>pip install django
```

Ilustración 17 Instalación de Django

Fuente: Creación propia

```

Collecting django
  Downloading Django-5.0.3-py3-none-any.whl (8.2 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.2/8.2 MB 262.2 kB/s eta 0:00:00
Collecting asgiref<4,>=3.7.0
  Downloading asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.2/41.2 kB 396.7 kB/s eta 0:00:00
Collecting tzdata
  Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 345.4/345.4 kB 238.2 kB/s eta 0:00:00
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-5.0.3 sqlparse-0.4.4 tzdata-2024.1

[notice] A new release of pip available: 22.3.1 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip

```

Ilustración 18 Implementación de instalación Django

Fuente: Creación propia

2. Crear el proyecto de Django

- Crear el proyecto en la carpeta de trabajo

```
django-admin startproject capacitaciondoc
```

Ilustración 19 Creación de proyecto en carpeta de trabajo

Fuente: Creación propia

- Ejecutar el servidor de desarrollo

```

(enCap) C:\CapacitacionDoc>cd capacitaciondoc

(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 17, 2025 - 00:37:42
Django version 5.1.4, using settings 'capacitaciondoc.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

Ilustración 20 Ejecución del servidor de desarrollo

Fuente: Creación propia

- Acceder a la aplicación en el navegador

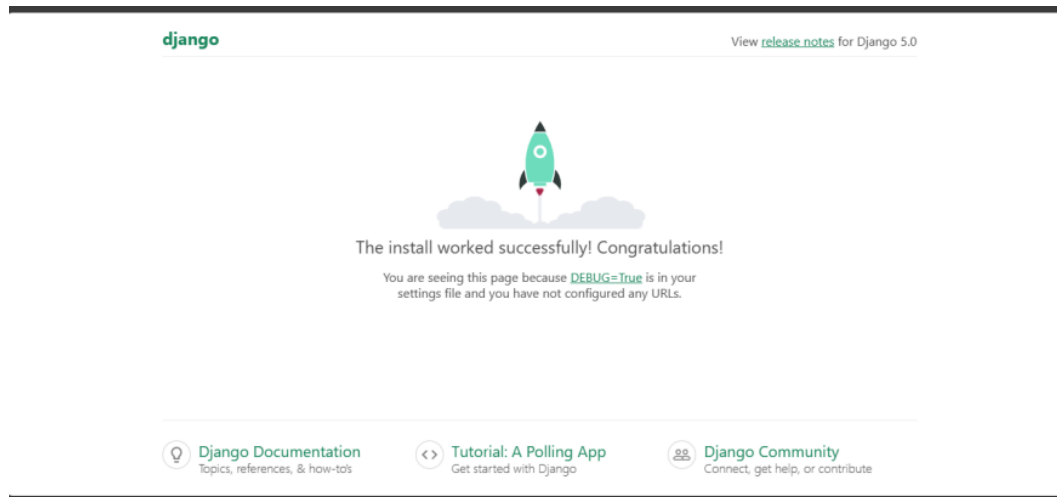


Ilustración 21 Acceso al navegador Django

Fuente: Creación propia

3. Crear el super usuario

- Dirigirse a la carpeta del proyecto
- Ejecutar el comando para crear el super usuario

Comando: “python manage.py createsuperuser”

- Realizar el llenado de datos que pide
- Iniciar el servidor y acceder al panel con la url/admin

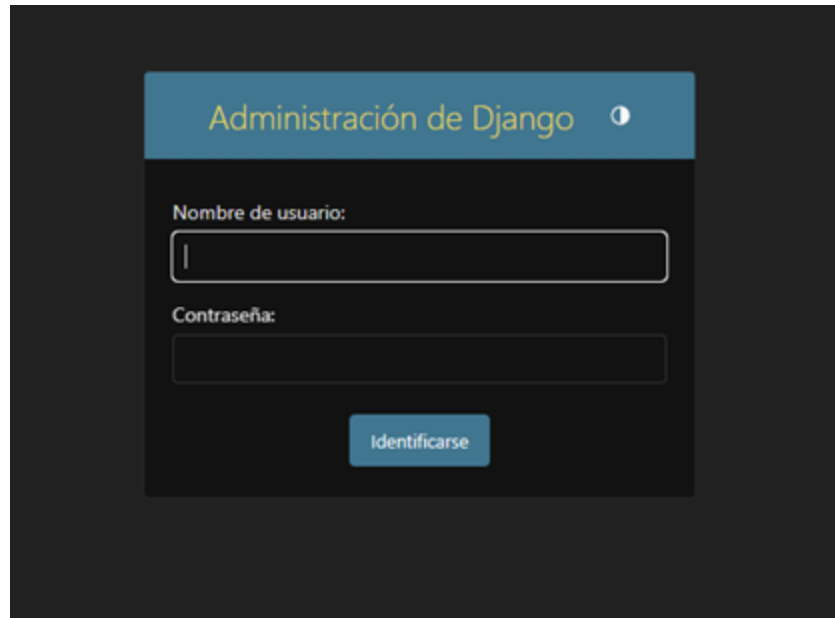


Ilustración 22 Panel del sitio de administrador Django (Inicio de sesión)

Fuente: Creación propia



Ilustración 23 Panel del sitio de administrador Django (Logueo aceptado)

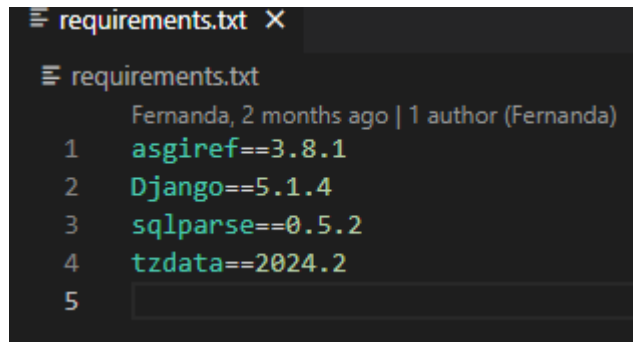
Fuente: Creación propia

Al implementar un proyecto en Django se recomienda generar el archivo “requirements.txt”, ya que este contendrá una lista de todas las dependencias del proyecto de Python, es decir, las librerías y las versiones necesarias para ejecutar el proyecto correctamente. Pasos para seguir:

1. Entorno virtual
- Activar el entorno virtual
2. Generar el archivo
- Comando a utilizar

“pip freeze > requirements.txt”

- Salida del archivo

A screenshot of a code editor window titled 'requirements.txt'. The editor shows the following content: 'Fernanda, 2 months ago | 1 author (Fernanda)' followed by a list of dependencies: 'asgiref==3.8.1', 'Django==5.1.4', 'sqlparse==0.5.2', and 'tzdata==2024.2'. The lines are numbered 1 through 5 on the left margin.

```
requirements.txt X
requirements.txt
Fernanda, 2 months ago | 1 author (Fernanda)
1 asgiref==3.8.1
2 Django==5.1.4
3 sqlparse==0.5.2
4 tzdata==2024.2
5
```

Ilustración 24 Generación del archivo

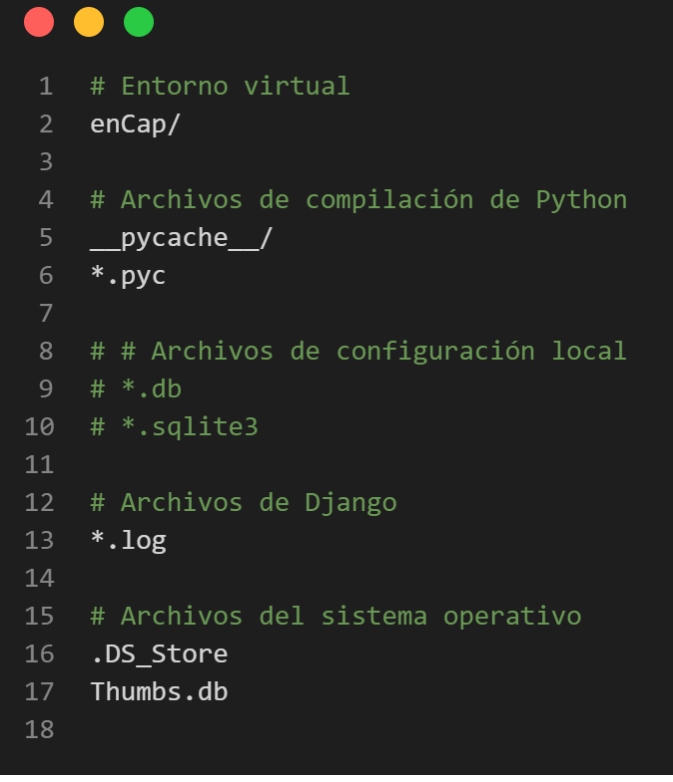
Fuente: Creación propia

3. Repositorio de GitHub

Al crear un repositorio en GitHub y subir el proyecto en el cual se estará trabajando lo ideal es crear un archivo “. gitignore”, lo cual le indica a Git que archivos subir o no, asegurando de no subir archivos innecesarios como configuraciones locales o entornos virtuales

1. Crear el archivo. gitignore

Ir a la raíz del proyecto y agregar los archivos que no serán rastreados (Base de datos, entorno virtual, archivos de Django y archivos del sistema operativo)



```
1  # Entorno virtual
2  enCap/
3
4  # Archivos de compilación de Python
5  __pycache__/
6  *.pyc
7
8  # # Archivos de configuración local
9  # *.db
10 # *.sqlite3
11
12 # Archivos de Django
13 *.log
14
15 # Archivos del sistema operativo
16 .DS_Store
17 Thumbs.db
18
```

Ilustración 25 Creación de archivo. gitignore

Fuente: Creación propia

2. Inicio del repositorio Git

- Abrir la terminal dentro de la carpeta del proyecto
- Ejecutar el comando de inicialización

“git init”

- Agregar los archivos al repositorio Git

“git add”

- Realización del primer commit

“git commit -m “Mi primer commit”

- Copia del URL del repositorio creado

“git remote add origin [url del repositorio]

- Subir el proyecto al repositorio

“git push -u origin main”

main 4 Branches 0 Tags			Go to file	Add file	Code
FerTecn Se eliminan identificadores de archivos y se cargan modelos de Oficio...			697e076 · 4 days ago	233 Commits	
	capacitaciondoc	Se eliminan identificadores de archivos y se cargan modelos ...	4 days ago		
	.gitignore	Se agregan permisos sobre docente	last month		
	README.md	Initial commit	2 months ago		
	requirements.txt	Primer commit	2 months ago		

Ilustración 26 Panel de repositorio Git

Fuente: Creación propia

4. Configuración de las aplicaciones del proyecto

Al trabajar con Django, una de las primeras tareas es crear aplicaciones dentro del proyecto, estas se conocen como módulos independientes que contiene la funcionalidad del sistema. Ya que al crearlas se integran al panel de administración de Django para la gestión de los datos de manera visual a través de una interfaz web.

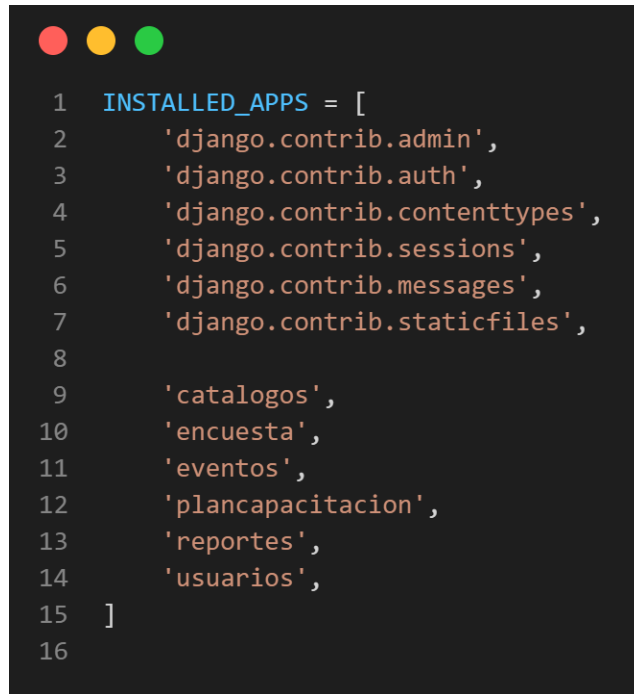
1. Crear la aplicación correspondiente

```
(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py startapp catalogos
(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py startapp encuesta
(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py startapp eventos
(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py startapp plancapacitacion
(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py startapp reportes
(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py startapp usuarios
```

Ilustración 27 Configuración de las aplicaciones

Fuente: Creación propia

2. Registrar la aplicación en Settings.py



```
1  INSTALLED_APPS = [  
2      'django.contrib.admin',  
3      'django.contrib.auth',  
4      'django.contrib.contenttypes',  
5      'django.contrib.sessions',  
6      'django.contrib.messages',  
7      'django.contrib.staticfiles',  
8  
9      'catalogos',  
10     'encuesta',  
11     'eventos',  
12     'plancapacitacion',  
13     'reportes',  
14     'usuarios',  
15 ]  
16
```

Ilustración 28 Registro de aplicación en Settings.py

Fuente: Creación propia

3. Ejecutar las migraciones correspondientes

```
(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py makemigration
No changes detected

(enCap) C:\CapacitacionDoc\capacitaciondoc>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

Ilustración 29 Ejecución de migraciones

Fuente: Creación propia

5. Instalación de librerías

En este proyecto es esencial la implementación de PDF e imágenes, con base a ellos se necesitan ocupar las librerías como ReportLab y Pillow. Además de que en la personalización de la apariencia de los documentos se necesita utilizar una fuente específica como Monserrat permitiéndole dar un toque más estético

- Instalar ReportLab

```
(enCap) C:\CapacitacionDoc\capacitaciondoc>pip install reportlab
Collecting reportlab
  Using cached reportlab-4.3.1-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: pillow>=9.0.0 in c:\capacitaciondoc\enCap\lib\site-packages (from reportlab) (11.1.0)
Collecting chardet (from reportlab)
  Using cached chardet-5.2.0-py3-none-any.whl.metadata (3.4 kB)
Using cached reportlab-4.3.1-py3-none-any.whl (1.9 MB)
Using cached chardet-5.2.0-py3-none-any.whl (199 kB)
Installing collected packages: chardet, reportlab
Successfully installed chardet-5.2.0 reportlab-4.3.1
```

Ilustración 30 Instalación de librería ReportLab

Fuente: Creación propia

- Instalar Pillow

```
(enCap) C:\CapacitacionDoc\capacitaciondoc>pip install pillow
Collecting pillow
  Using cached pillow-11.1.0-cp312-cp312-win_amd64.whl.metadata (9.3 kB)
Using cached pillow-11.1.0-cp312-cp312-win_amd64.whl (2.6 MB)
Installing collected packages: pillow
Successfully installed pillow-11.1.0
```

Ilustración 31 Instalación de librería Pillow

Fuente: Creación propia

```
(enCap) C:\CapacitacionDoc>pip show pillow reportlab
Name: pillow
Version: 11.1.0
Summary: Python Imaging Library (Fork)
Home-page: https://python-pillow.github.io
Author:
Author-email: "Jeffrey A. Clark" <aclark@aclark.net>
License: MIT-CMU
Location: C:\CapacitacionDoc\enCap\Lib\site-packages
Requires:
Required-by: reportlab
---
Name: reportlab
Version: 4.3.1
Summary: The Reportlab Toolkit
Home-page: https://www.reportlab.com/
Author: Andy Robinson, Robin Becker, the ReportLab team and the community
Author-email: reportlab-users@lists2.reportlab.com
License: BSD license (see license.txt for details), Copyright (c) 2000-2025, ReportLab Inc.
Location: C:\CapacitacionDoc\enCap\Lib\site-packages
Requires: chardet, pillow
Required-by:
```

Ilustración 32 Demostración exitosa de la descarga de librerías

Fuente: Creación propia

- Descargar la fuente Monserrat

Pasos:

- a) Descargar la fuente en Google Fonts
- b) Extraer los archivos
- c) Guardas el archivo en static/fonts/ dentro del proyecto de django

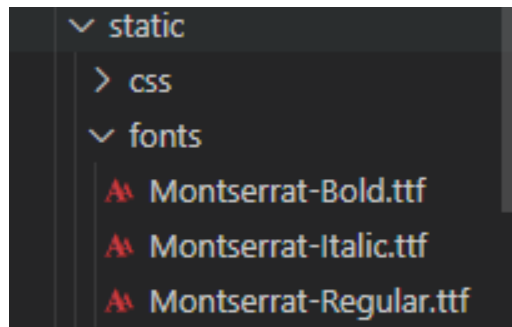


Ilustración 33 Fuente Monserrat descargada y alojada

Fuente: Creación propia

IV.RESULTADOS

1. Análisis de requerimientos

En esta fase, se identificaron las necesidades del sistema, el flujo de datos y las interacciones entre los diferentes componentes a ocupar para garantizar un sistema robusto y escalable en base a los objetivos del proyecto.

El análisis de requerimientos permite definir aspectos clave como la arquitectura y los diferentes modelos de datos como la lógica y la integración de estos mismos, además se establece consideraciones de seguridad, rendimiento y escalabilidad para asegurar un backend optimizado y confiable.

Para proporcionar una visión mas estructurada se elaboraron diagramas que ayudaron a mostrar la secuencia que se iba a tener y las diferentes interacciones entre los distintos

módulos, así mismo establecer las dependencias entre las funciones del sistema con una planificación adecuada, de manera que estos diagramas fueron fundamentales para la representación gráfica y herramienta de referencia durante los requerimientos del proyecto para un mejor ciclo de trabajo.

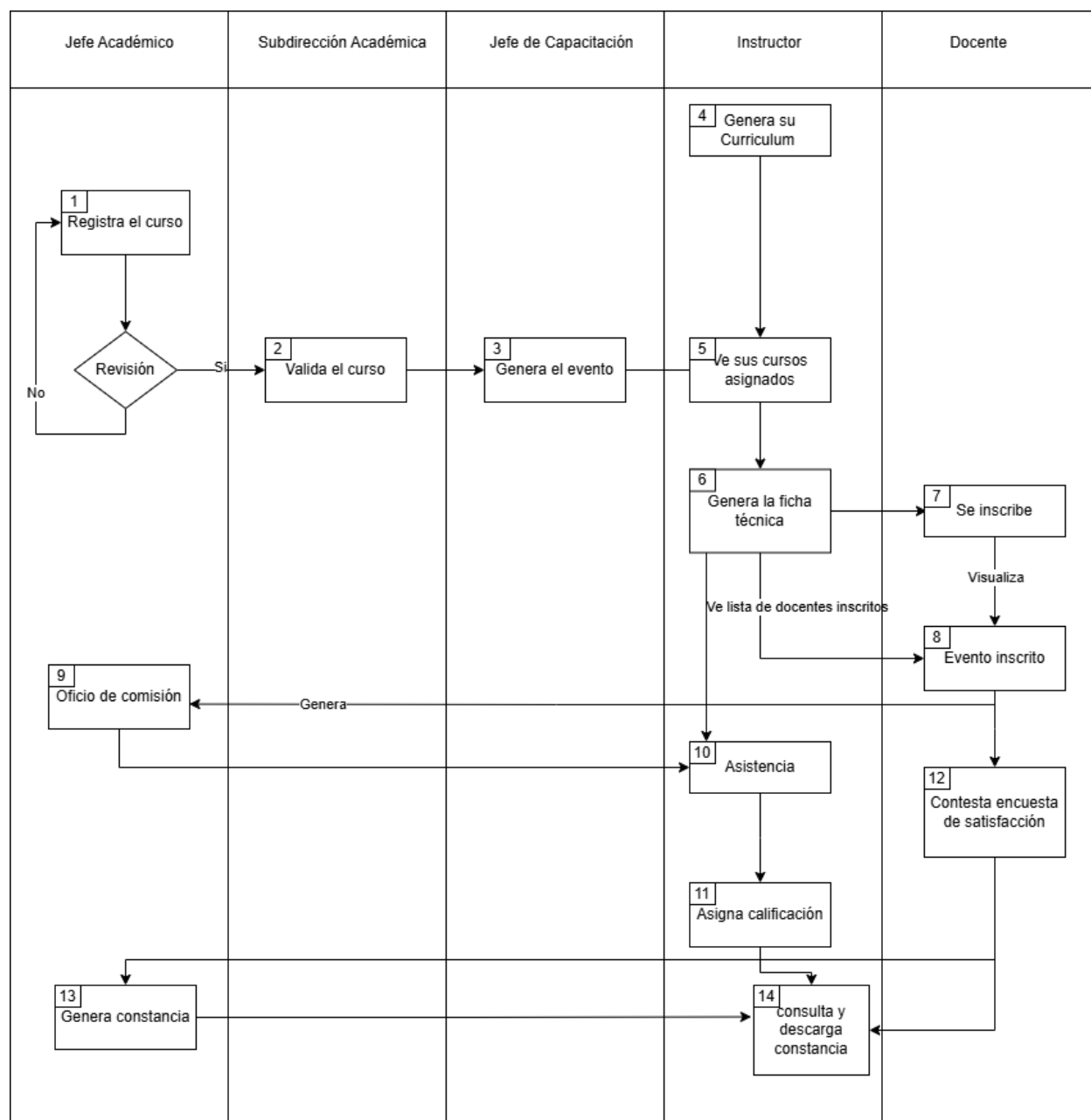


Ilustración 34 Diagrama de procedimiento capacitación docente

Fuente: Creación propia

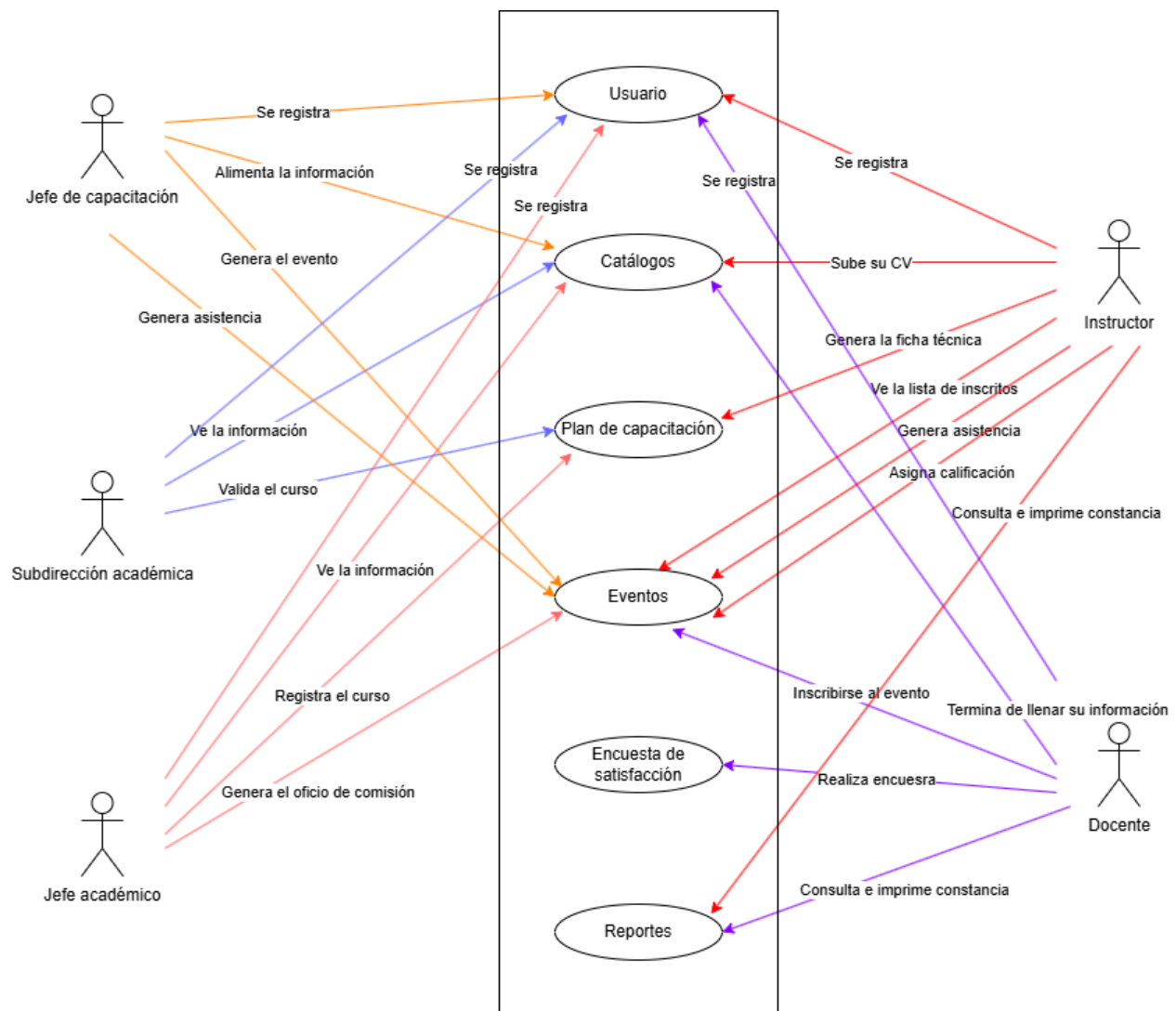


Ilustración 35 Diagrama de casillas CAPACITA-TEC

Fuente: Creación propia

2. Definición de la estructura de modelos y base de datos

Se definió la estructura del proyecto las funcionalidades en las diferentes aplicaciones (apps). Cada aplicación fue diseñada para cubrir una funcionalidad específica. Se implementaron los modelos necesarios para representar las entidades principales del sistema, definidos en los archivos `models.py` de cada aplicación. Las relaciones entre las

entidades se realizaron ocupando las claves primarias y foráneas, asegurando los datos y permitiendo que las tablas se relacionaran correctamente en la base de datos.

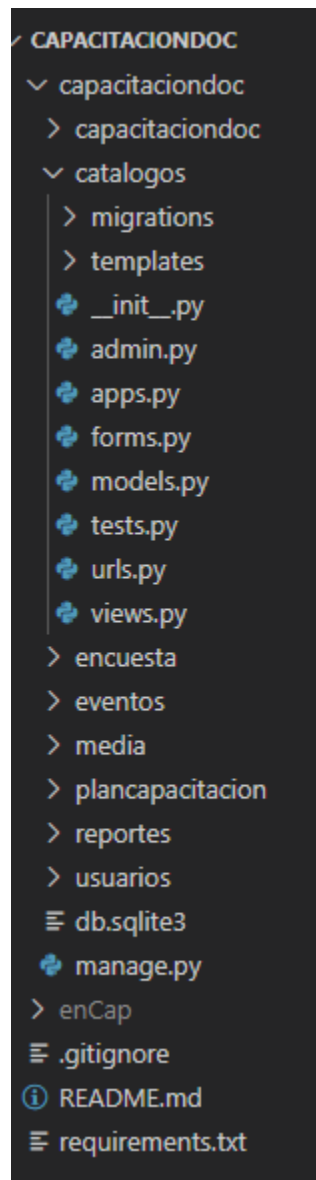


Ilustración 36 Estructura final

Fuente: Creación propia



```

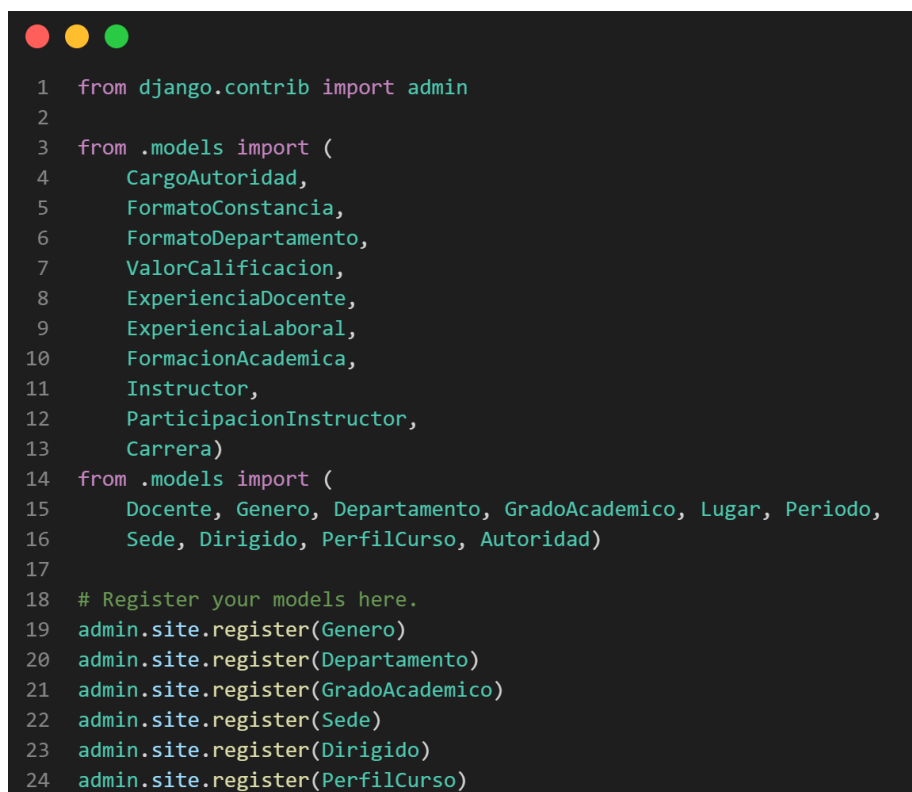
1 class Instructor(models.Model):
2     # ...
3     # ...
4     # ...
5     # ...
6     # ...
7
8     def __str__(self):
9         return f"{self.user.last_name}"
10
11 class Meta:
12     verbose_name_plural = 'Instructores'

```

Ilustración 37 Creación del modelo instructor -app catálogos-

Fuente: Creación propia

Posteriormente en el archivo admin.py de cada aplicación se registran los modelos con las adecuadas migraciones “**python manage.py makemigrations**” y “**python manage.py migrate**” de esta manera se mostrarán en el panel de administración de Django de manera que permite obtener los datos a través de una interfaz gráfica haciendo la búsqueda de datos más fácil.



```

1 from django.contrib import admin
2
3 from .models import (
4     CargoAutoridad,
5     FormatoConstancia,
6     FormatoDepartamento,
7     ValorCalificacion,
8     ExperienciaDocente,
9     ExperienciaLaboral,
10    FormacionAcademica,
11    Instructor,
12    ParticipacionInstructor,
13    Carrera)
14
15 from .models import (
16    Docente, Genero, Departamento, GradoAcademico, Lugar, Periodo,
17    Sede, Dirigido, PerfilCurso, Autoridad)
18
19 # Register your models here.
20 admin.site.register(Genero)
21 admin.site.register(Departamento)
22 admin.site.register(GradoAcademico)
23 admin.site.register(Sede)
24 admin.site.register(Dirigido)
25 admin.site.register(PerfilCurso)

```

Ilustración 38 Registro de los modelos en admin.py

Fuente: Creación propia

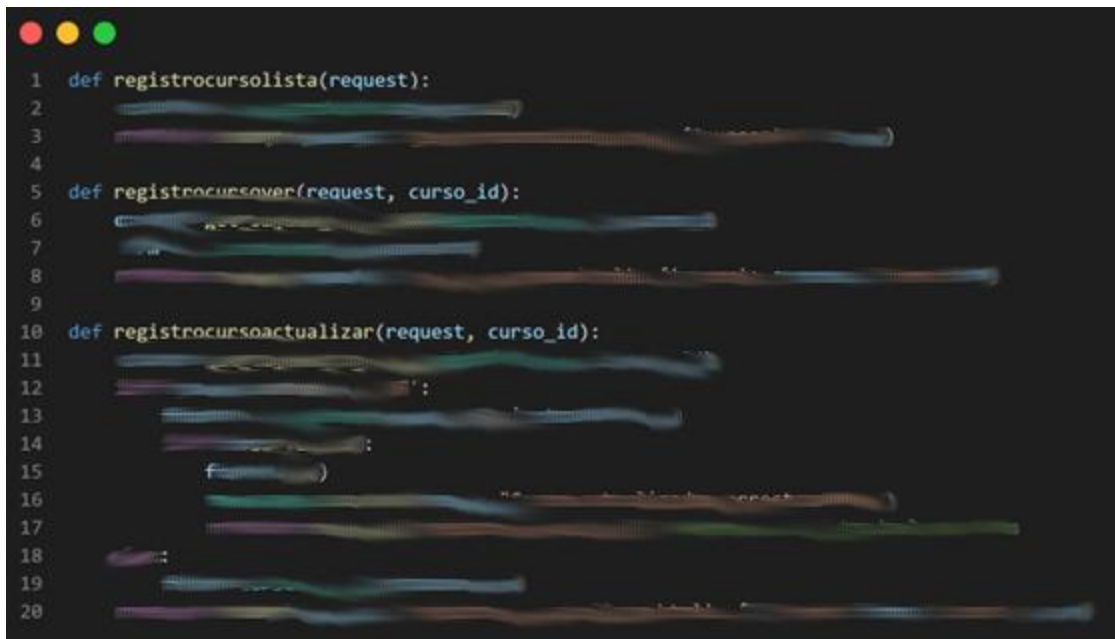
AUTENTICACIÓN Y AUTORIZACIÓN		
Grupos	+ Agregar	✎ Modificar
CATALOGOS		
Autoridades	+ Agregar	✎ Modificar
Cargos de Autoridad	+ Agregar	✎ Modificar
Carreras	+ Agregar	✎ Modificar
Departamentos	+ Agregar	✎ Modificar
Dirigido a	+ Agregar	✎ Modificar
Docentes	+ Agregar	✎ Modificar
Experiencia docentes	+ Agregar	✎ Modificar
Experiencia laborales	+ Agregar	✎ Modificar
Formacion academicas	+ Agregar	✎ Modificar
Formato constancias	+ Agregar	✎ Modificar
Formato departamentos	+ Agregar	✎ Modificar
Genero	+ Agregar	✎ Modificar
Grado académico	+ Agregar	✎ Modificar
Instructores	+ Agregar	✎ Modificar
Lugares	+ Agregar	✎ Modificar
Participacion instructors	+ Agregar	✎ Modificar
Perfiles de Curso	+ Agregar	✎ Modificar
Periodos	+ Agregar	✎ Modificar
Sedes	+ Agregar	✎ Modificar
Valores de Calificación	+ Agregar	✎ Modificar

ENCUESTA		
Encuestas	+ Agregar	✎ Modificar
EVENTOS		
Asistencias	+ Agregar	✎ Modificar
Calificaciones	+ Agregar	✎ Modificar
Eventos	+ Agregar	✎ Modificar
Evidencias	+ Agregar	✎ Modificar
Inscripcion	+ Agregar	✎ Modificar
Oficios de comisión	+ Agregar	✎ Modificar
PLANCAPACITACION		
Contenidos Temáticos	+ Agregar	✎ Modificar
Criterios de Evaluación	+ Agregar	✎ Modificar
Fichas técnicas	+ Agregar	✎ Modificar
Registro de cursos	+ Agregar	✎ Modificar
Validación de cursos	+ Agregar	✎ Modificar
REPORTES		
Constancias de docentes	+ Agregar	✎ Modificar
Constancias de instructores	+ Agregar	✎ Modificar
USUARIOS		
Usuarios	+ Agregar	✎ Modificar

Ilustración 39 Panel de administración de Django

Fuente: Creación propia

Se utilizó un ORM (Object-Relational Mapping) para la gestión de las operaciones con la base de datos simplificando la creación y consulta de datos. Ocupando DBeaver para la administración visual de la base de datos, lo que permitió inspeccionar las tablas, realizar consultas SQL y verificar la integración de los datos. Cada modelo fue migrado a la base de datos, asegurando la estructura de los **models.py** de esta manera asegura una sincronización adecuada entre el diseño del modelo como la estructura de las tablas del sistema, con esta estructura se logró una integración funcional, manteniendo un diseño limpio permitiendo un desarrollo ágil.



```

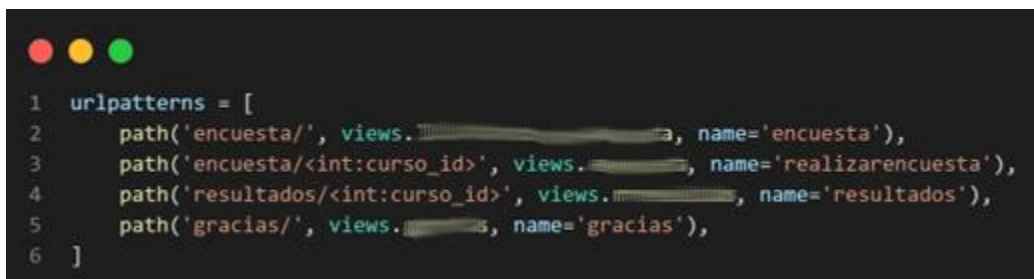
1 def registrocurso(request):
2
3
4
5 def registrocursover(request, curso_id):
6
7
8
9
10 def registrocursoactualizar(request, curso_id):
11
12
13
14
15
16
17
18
19
20

```

Ilustración 41 views.py de app plan de capacitación

Fuente: Creación propia

Las rutas fueron definidas en los archivos **urls.py** de cada aplicación así mismo desde el archivo principal de **urls.py**, esto permitió una separación de la lógica entre las diferentes áreas del sistema y facilitó la gestión de las vistas a lo largo del desarrollo. Además, se diseñaron vistas para funciones específicas como: creación y gestión de usuarios, visualización, validación de cursos y asignación de eventos



```

1 urlpatterns = [
2     path('encuesta/', views. , name='encuesta'),
3     path('encuesta/<int:curso_id>', views. , name='realizarencuesta'),
4     path('resultados/<int:curso_id>', views. , name='resultados'),
5     path('gracias/', views. , name='gracias'),
6 ]

```

Ilustración 42 urls.py de app plan capacitación

Fuente: Creación propia

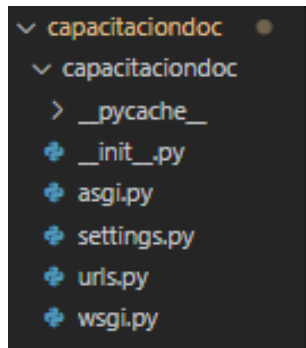


Ilustración 43 Estructura de la carpeta principal y de la app

Fuente: Creación propia

```

1  from django.contrib import admin
2  from django.urls import path, include
3  from django.conf import settings
4  from usuarios.views import custom_403_view, custom_404_view, home
5
6  urlpatterns = [
7      path('admin/', admin.site.urls),
8      path('', home, name="home"),
9      path('usuarios/', include('usuarios.urls')),
10     path('catalogos/', include('catalogos.urls')),
11     path('plancapacitacion/', include('plancapacitacion.urls')),
12     path('eventos/', include('eventos.urls')),
13     path('encuesta/', include('encuesta.urls')),
14     path('reportes/', include('reportes.urls')),
15 ]

```

Ilustración 44 Organización del urls.py principal del proyecto

Fuente: Creación propia

4. Personalización del panel de administración (admin)

En el panel de administración de Django fue personalizado para ajustarse a las necesidades del sistema, se configuro el modelo de usuario para que utilizara la **CURP** como identificador principal sustituyendo el “**username**”.

De manera que para lograr esta modificación se implementó un modelo de usuario personalizado con la importación de clase como “**AbstractUser**” configurando los campos necesarios para el control de autenticación y permisos.

Posteriormente se complementa la configuración de settings.py para el modelo de usuario personalizado especificando que se usara de manera predeterminada ocupando el modelo user en la app de usuarios

```
AUTH_USER_MODEL = 'usuarios.CustomUser'
```

Ilustración 45 Configuración de AbstractUser

Fuente: Creación propia



Ilustración 46 Panel del admin

Fuente: Creación propia

5. Registro e inicio de sesión

Durante el proceso de registro se valida que la CURP ingresada cumpla con lo requerido. Para asegurar que cada usuario tenga una identificación única y valido el sistema implementa verificaciones durante el registro, antes de registrar un nuevo usuario el sistema consulta en la base de datos asegurando de que la CURP no haya sido registrada anteriormente asegurando las credenciales de los usuarios contra accesos no autorizados.

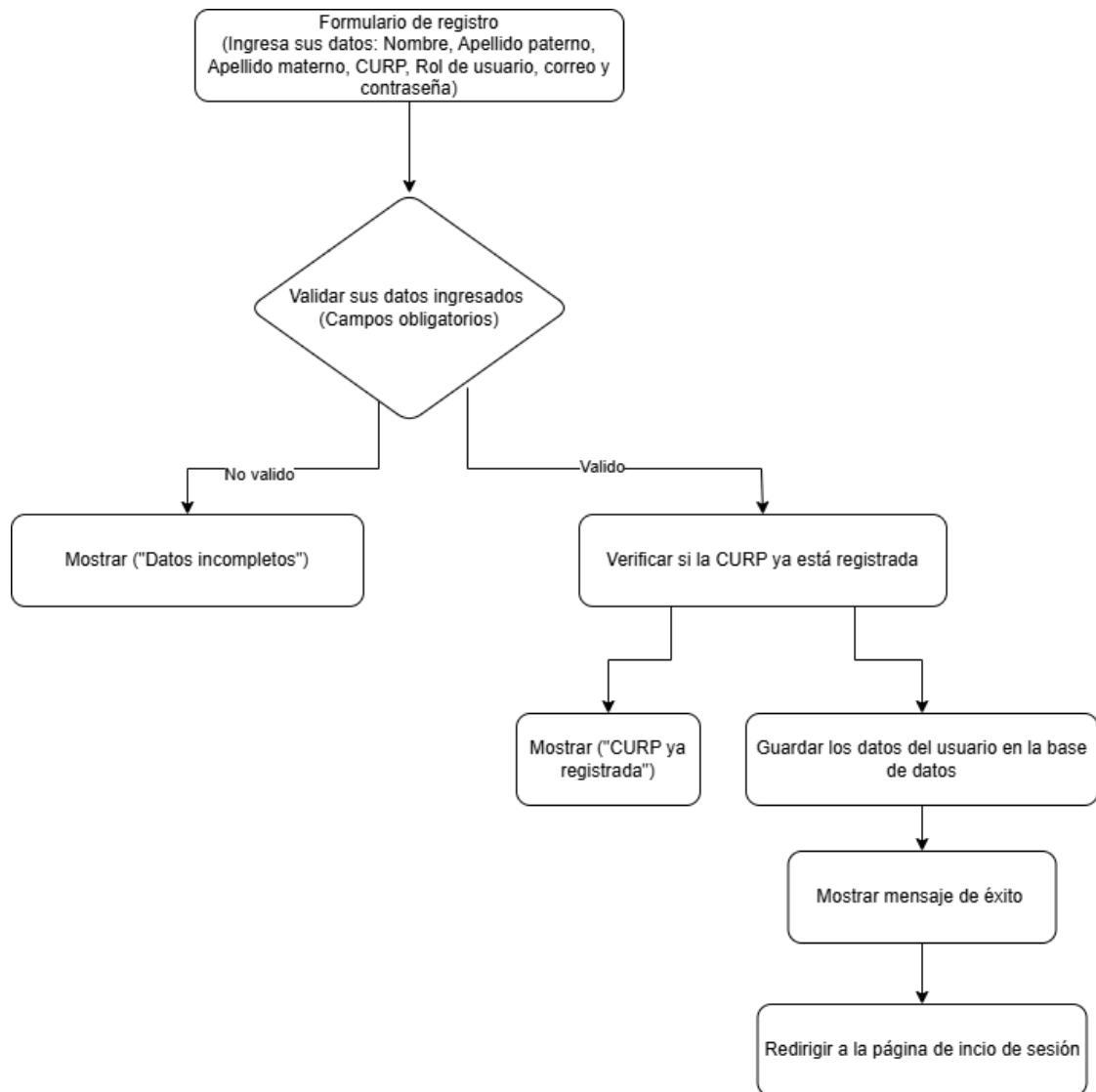


Ilustración 47 Diagrama de flujo "Registro de cuenta nueva"

Fuente: Creación propia

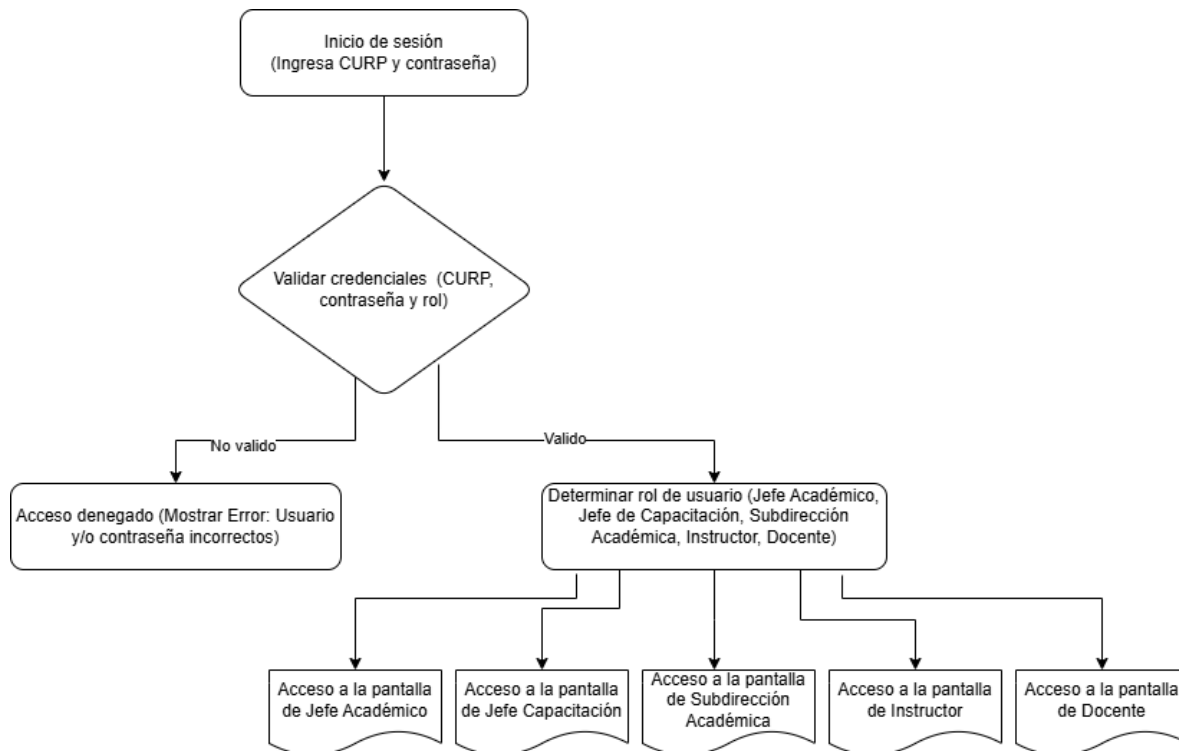


Ilustración 48 Diagrama de flujo del proceso de autenticación (Login)

Fuente: Creación propia

6. Roles de usuario

Para tener los diferentes tipos de usuarios dentro de un mismo sistema se ocuparon los roles en los grupos (Group) que proporciona Django permitiendo agrupar a los usuarios bajo un mismo perfil. Los roles establecidos son: Instructor, Jefe académico, Jefe de Capacitación, Docente y Subdirección Académica, cada uno de estos roles se registraron en los grupos facilitando la asignación y administración de permisos.

```

1 class CustomUser(AbstractUser):
2     """
3     """
4     """
5     """
6     max_length=50,
7     choices=[
8         ('Docente', 'Docente'),
9         ('Instructor', 'Instructor'),
10        ('Jefe de Capacitación', 'Jefe de Capacitación'),
11        ('Jefe Académico', 'Jefe Académico'),
12        ('Subdirección', 'Subdirección')
13    ],
14    verbose_name="Rol"
15 )
16 """
17 """
18 USERNAME_FIELD = 'curp' #Agregado: Define al CURP como el identificador inicial para permitir el login con CURP
19 objects = CustomUserManager() # Asigna el nuevo UserManager
20
21 def save(self, *args, **kwargs):
22     # Asignar el valor de CURP al campo username
23     self.username = self.curp
24     super().save(*args, **kwargs) # llamar al método save del modelo base
25
26 def __str__(self):
27     return f" {self.curp} "
28
29 def get_user_full_name(self):
30     return f" {self.curp} "

```

Ilustración 49 Asignación de roles utilizados en "Login y Registro de cuenta" -models.py, app Usuarios-

Fuente: Creación propia

```

1 from django.contrib import admin
2 from django.contrib.auth.admin import UserAdmin
3 from .models import CustomUser
4
5
6 @admin.register(CustomUser)
7 class CustomUserAdmin(UserAdmin):
8     model = CustomUser
9     fieldsets = UserAdmin.fieldsets + (
10         (None, {'fields': ('curp', 'password', 'username', 'email', 'first_name', 'last_name', 'is_active', 'is_staff', 'is_superuser', 'date_joined', 'last_login')})),
11     )
12     add_fieldsets = UserAdmin.add_fieldsets + (
13         (None, {'fields': ('curp', 'password', 'username', 'email', 'first_name', 'last_name', 'is_active', 'is_staff', 'is_superuser', 'date_joined', 'last_login')})),
14     )
15     list_display = ('curp', 'password', 'username', 'email', 'first_name', 'last_name', 'is_active', 'is_staff', 'is_superuser', 'date_joined', 'last_login')
16     search_fields = ('curp', 'password', 'username', 'email', 'first_name', 'last_name', 'is_active', 'is_staff', 'is_superuser', 'date_joined', 'last_login')
17     list_filter = ('curp',)
18

```

Ilustración 50 Registro en el admin.py -App Usuarios-

Fuente: Creación propia

☐	GRUPO
☐	Docente
☐	Instructor
☐	Jefe Académico
☐	Jefe de Capacitación
☐	Subdirección

5 grupos

Ilustración 51 Roles de usuario asignados en el panel de administración

Fuente: Creación propia

7. Proceso de autenticación en vistas

Django proporciona un sistema de autenticación robusto, este incluye el decorador “**@login_required**” el cual restringe el acceso a ciertas vistas con el fin de que solo el usuario identificado pueda entrar al sistema.

Se define la url específica para el inicio de sesión en el archivo urls.py posteriormente en cada vista de cada app se importa el decorador, de manera que protege el sistema de usuarios no registrados.

```

1  @login_required(login_url='signin')
2  ...
3  def eventolista(request):
4      ...
5      return render(request, ...)
6
7  @login_required(login_url='signin')
8  @...
9  def eventover(request, evento_id):
10     ...
11     ...

```

Ilustración 52 Asignación de @login_required

Fuente: Creación propia

8. Permisos de vistas para cada usuario

El sistema fue diseñado con un control estricto de accesos para proteger la información y garantizar la privacidad. En cada vista se utilizó los permisos adicionales como “@permission_required” configurados en el modelo de usuario (Explicado en el paso 6), de esta manera permite realizar acciones personalizadas.

El decorador se usa para asegurarse de que un usuario tenga ciertos permisos de acceder a una vista o a una acción, si el usuario no tiene el permiso requerido se mostrara un error 403 hace énfasis de “**Permiso denegado**” -configurado en la urls.py de la carpeta principal del proyecto-.

```
1 # Permisos
2 def configurar_grupos_y_permisos(sender, **kwargs):
3     # Crea o obtiene los grupos en la BD
4     docente_group, created = Group.objects.get_or_create(name='Docente')
5     instructor_group, created = Group.objects.get_or_create(name='Instructor')
6     academico_group, created = Group.objects.get_or_create(name='Jefe Académico')
7     capacitacion_group, created = Group.objects.get_or_create(name='Jefe de Capacitación')
8     subdireccion_group, created = Group.objects.get_or_create(name='Subdirección')
9
10    try:
11        # Obtener permisos definidos en los modelos
12        permisos_eventos = Permission.objects.filter(content_type__app_label='eventos', codename__icontains='evento')
13        permisos_inscripciones = Permission.objects.filter(content_type__app_label='eventos', codename__icontains='inscripcion')
14        permisos_asistencias = Permission.objects.filter(content_type__app_label='eventos', codename__icontains='asistencia')
15        permisos_calificaciones = Permission.objects.filter(content_type__app_label='eventos', codename__icontains='calificacion')
16        permisos_evidencias = Permission.objects.filter(content_type__app_label='eventos', codename__icontains='evidencia')
17        permisos_oficios_comision = Permission.objects.filter(content_type__app_label='eventos', codename__icontains='oficiocomision')
18
19        def limpiar_permisos(grupo):
20            permisos_actuales = grupo.permissions.filter(content_type__app_label='eventos')
21            grupo.permissions.remove(*permisos_actuales)
22
23        limpiar_permisos(docente_group)
24        limpiar_permisos(instructor_group)
25        limpiar_permisos(academico_group)
26        limpiar_permisos(capacitacion_group)
27        limpiar_permisos(subdireccion_group)
```

Ilustración 53 Asignación de permisos de cada rol -app.py de cada app-

Fuente: Creación propia

```

1 # Asignar permisos a los grupos
2 #
3 instructor_group.permissions.add(
4     [
5         'crear_evento',
6     ]
7 )
8 #
9 docente_group.permissions.add(
10     [
11         'crear_evento',
12     ]
13 )
14 #
15 academico_group.permissions.add(
16     [
17         'crear_evento',
18     ]
19 )
20 #
21 capacitacion_group.permissions.add(
22     [
23         'crear_evento',
24     ]
25 )
26 #
27 subdireccion_group.permissions.add(
28     [
29         'crear_evento',
30     ]
31 )

```

Ilustración 54 Filtros de acciones que puede realizar el usuario -app.py de la app eventos-

Fuente: Creación propia

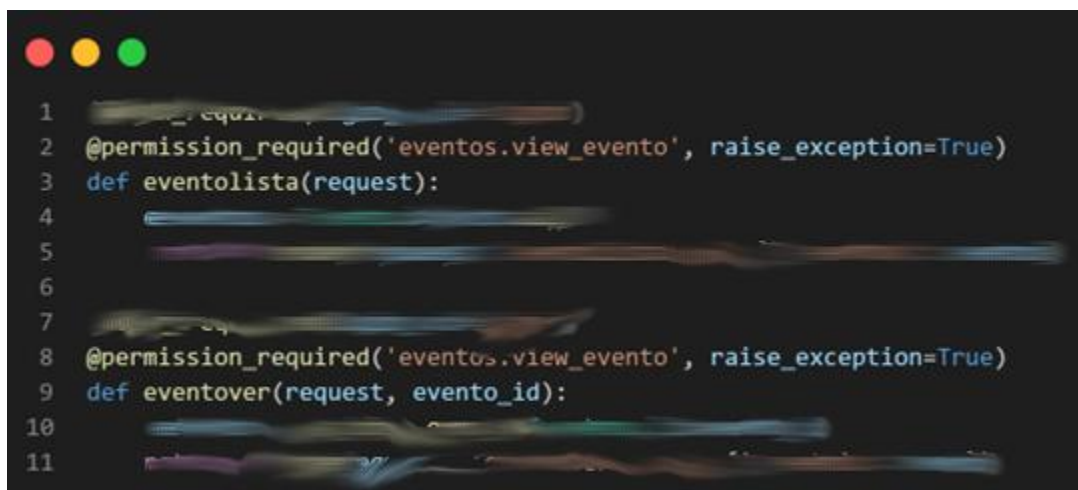


Ilustración 55 Visualización del panel de administración los permisos de cada rol

Fuente: Creación propia

Posteriormente estos permisos se agregan al views.py de cada app con el decorador llevando una estructura donde:

@permission_required (app.permiso_codename, parámetro)



```
1 from django.shortcuts import render
2 @permission_required('eventos.view_evento', raise_exception=True)
3 def eventolista(request):
4     return render(request, 'eventos/evento_list.html', {'eventos': eventos})
5     return render(request, 'eventos/evento_list.html', {'eventos': eventos})
6
7 from django.shortcuts import render
8 @permission_required('eventos.view_evento', raise_exception=True)
9 def eventover(request, evento_id):
10     return render(request, 'eventos/evento_detail.html', {'evento': evento})
11     return render(request, 'eventos/evento_detail.html', {'evento': evento})
```

Ilustración 56 Asignación de @permission_required

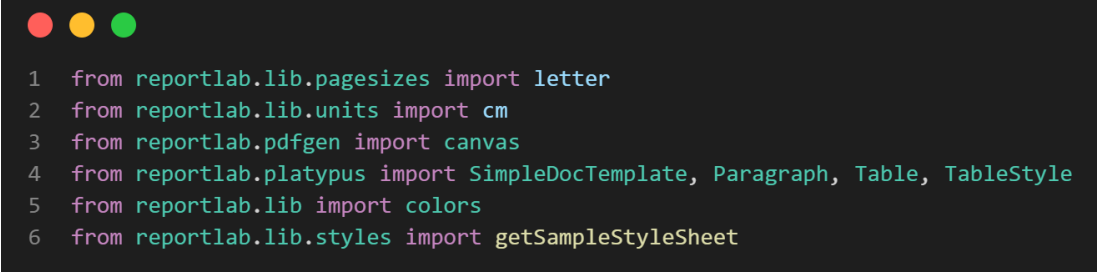
Fuente: Creación propia

9. Documentos en PDF

Una funcionalidad destacada del sistema es la capacidad de generar documentos en formato PDF a partir de los datos almacenados. Incluye reportes como el curriculum vitae del instructor, las constancias generadas, oficio de comisión y ficha técnica. Los documentos fueron diseñados para tener un formato profesional facilitando las tareas y distribución de reportes entre los usuarios del sistema.

Para generar el PDF de cada vista:

- Extrae la librería instalada de reportlab.
- Se crea la vista que genera el PDF usando canvas.Canvas() de Reportlab.
- Personaliza el contenido del PDF de manera que se puede poner el estilo de texto, gráficos, color, tablas, entre otros.
- Configura la respuesta HTTP para que el navegador lo detecte como archivo PDF.



```
1 from reportlab.lib.pagesizes import letter
2 from reportlab.lib.units import cm
3 from reportlab.pdfgen import canvas
4 from reportlab.platypus import SimpleDocTemplate, Paragraph, Table, TableStyle
5 from reportlab.lib import colors
6 from reportlab.lib.styles import getSampleStyleSheet
```

Ilustración 57 Asignación de librerías de PDF

Fuente: Creación propia

```

1 def fichatextoapi(request, curso_id=None):
2     curso = get_object_or_404(Curso, id=curso_id)
3     ficha = get_object_or_404(Ficha, curso=curso)
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

Ilustración 58 views.py para generación del PDF

Fuente: Creación propia

IV. CONCLUSIONES

El desarrollo de este proyecto de software fue una experiencia enriquecedora y desafiante tanto a nivel técnico como personal ya que, durante este proceso, logre aplicar y profundizar más mis conocimientos en el desarrollo de software con soluciones eficientes y significativas, utilizando herramientas que me permitieron construir un sistemas eficiente y adaptable para las necesidades reales de los usuarios.

Este proyecto me dio la oportunidad de enfrentar desafíos que fortalecieron mis habilidades en base de datos, análisis de información y codificación en la parte del backend, así como la identificación y resolución de problemas que se enfrentaron en el proyecto.

Además, considero que el sistema desarrollado será de gran utilidad ya facilitará la gestión de datos permitiendo una búsqueda más ágil y precisa así mismo minimizará errores humanos mejorando la interacción que tendrá cada rol con el sistema de manera eficiente y dinámica.

Optimizara tareas y contribuirá a la mejora continua proporcionando un mejor seguimiento y planificación, lo cual, representara un gran avance importante en la estructura y desarrollo de la capacitación docentes.

Así mismo esta experiencia no solo me permitió concluir con mis residencias profesionales, sino que también me dio la oportunidad de familiarizarme con herramientas clave que sin duda serán de gran ayuda al aplicarlas en el área laboral. a través de este proceso, adquirí una visión más clara del ciclo de vida del desarrollo de software, desde el análisis, planificación de la estructura de código, pruebas y la entrega del sistema ayudándome a mis conocimientos y habilidades para enfrentar futuros proyectos. Para concluir este proyecto ha sido un paso fundamental en mi formación académica y profesional permitiendo crecer tanto a nivel técnico como personal preparándome mejor y teniendo una visión mas clara de lo que es el mundo laboral.

VI. COMPETENCIAS DESARROLLADAS Y/O APLICADAS.

- Manejo de base de datos y ORM

Se desarrollo la capacidad de diseñar y gestionar base de datos mediante el ORM de Django, lo que permitió estructurar la información de manera eficiente. A través de la creación y aplicación de migraciones, se logró actualizar la estructura de la base de datos sin comprometer la integridad de los datos existentes.

- Implementación de seguridad y control de acceso

Se adquirieron habilidades en la implementación de sistemas de autoidentificación personalizadas y para garantizar la seguridad de datos se aplicaron técnicas de cifrado de contraseñas, así mismo se configuraron permisos y roles de usuario dentro del sistema asegurando un control de acceso adecuado en vistas y en el panel de administración permitiendo que cada usuario solo tuviera acceso a las funcionalidades correspondientes de cada rol.

- Desarrollo de funcionalidades avanzadas

Se creo la generación de documentos PDF mediante librerías especializadas. Además, la creación de vistas protegidas permitiendo una mejor comunicación del sistema y seguridad en él.

- Trabajo en equipo y colaboración

Coordinación con otros integrantes del equipo para la asignación de tareas y responsabilidades y el uso de metodologías para la organización del flujo de trabajo y dar seguimiento a las actividades. Comunicación efectiva para la integración de los diferentes módulos del sistema y el uso de comentarios para mejorar la legibilidad de la estructura y código.

- Comunicación con los interesados del proyecto

Capacidad de comunicarme de manera clara y eficaz para entender sus necesidades del proyecto, por lo que trabaje conjunto con los clientes, permitiéndome tener una mejor visión de la comunicación efectiva entre cliente y desarrollador teniendo un enfoque más claro.

- Resolución de problemas

Se desarrollo la identificación y solución de errores durante la implementación del sistema para la detección de fallos en la lógica de la vistas y modelos como las pruebas de funcionalidad para verificar la correcta ejecución del sistema.

A lo largo del proyecto, se adquirió competencias clave en el desarrollo del backend, base de datos, seguridad, generación de documentos y trabajo en equipo. Estas habilidades fortalecieron la capacidad de análisis, resolución de problemas y colaboración, asegurando que el sistema desarrollado cumpliera con los requisitos funcionales y de seguridad establecidos.

VII. FUENTES DE INFORMACIÓN

- [1] A. Ken, «gluo,» 03 Octubre 2023. [En línea]. Available: <https://www.gluo.mx/blog/backend-que-es-y-para-que-sirve>. [Último acceso: 28 Enero 2025].
- [2] NFON, «NFON,» 12 Julio 2022. [En línea]. Available: <https://blog.nfon.com/es/que-es-y-como-funciona-microsoft-teams/>. [Último acceso: 28 Enero 2025].
- [3] J. V. Fernando, «Observatorio de tecnología educativa,» [En línea]. Available: https://intef.es/observatorio_tecno/draw-io-mucho-mas-que-mapas-mentales/. [Último acceso: 28 Enero 2025].
- [4] JointJS, «DrawSQL,» [En línea]. Available: <https://www.jointjs.com/showcase/drawsql>. [Último acceso: 30 Enero 2025].
- [5] aws, «Amazon,» [En línea]. Available: <https://aws.amazon.com/es/what-is/django/>. [Último acceso: 31 Enero 2025].
- [6] aula21, «Centro de formación técnica para la industria,» 2024. [En línea]. Available: <https://www.cursosaula21.com/que-es-python/>. [Último acceso: 31 Enero 2025].
- [7] Concepto, «Base de datos,» 24 Diciembre 2024. [En línea]. Available: <https://concepto.de/base-de-datos/>. [Último acceso: 31 Enero 2025].
- [8] D. web, «EDteam,» 2022. [En línea]. Available: <https://ed.team/blog/que-es-un-orm>. [Último acceso: 02 Febrero 2025].
- [9] E. E. IONOS, «IONOS,» 27 Junio 2023. [En línea]. Available: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/sqlite/>. [Último acceso: 03 Febrero 2025].
- [10] M.Febrer, «Geo innova,» 20 Octubre 2022. [En línea]. Available: <https://geoinnova.org/blog-territorio/dbeaver-gestion-de-bases-de-datos/>. [Último acceso: 03 Febrero 2025].
- [11] A. Ayerdi, «DocuWare,» 06 Mayo 2024. [En línea]. Available: <https://start.docuware.com/es/blog/control-versiones#:~:text=Es%20una%20herramienta%20que%20organiza,historial%20de%20revis>. [Último acceso: 03 Febrero 2025].

- [12] Atlassian, «Atlassian,» 2024. [En línea]. Available:
] <https://www.atlassian.com/es/git/tutorials/what-is-git>. [Último acceso: 04 Febrero 2025].
- [13] X. Mallón, «Keep coding,» 14 Noviembre 2024. [En línea]. Available:
] <https://keepcoding.io/blog/elementos-de-un-archivo-requirements/>. [Último acceso: 04 Febrero 2025].
- [14] J. Web, «Codigos Python,» 17 Septiembre 2023. [En línea]. Available:
] <https://codigospython.com/procesamiento-de-imagenes-con-pillow-pil-en-python/>. [Último acceso: 05 Febrero 2025].
- [15] R. Python, «Recursos Python,» 03 Julio 2018. [En línea]. Available:
] <https://recursospython.com/guias-y-manuales/crear-documentos-pdf-en-python-con-reportlab/>. [Último acceso: 05 Febrero 2025].
- [16] AWS, «Amazon,» [En línea]. [Último acceso: 05 Febrero 2025].
]
- [17] «S.Cloud,» 18 Diciembre 2024. [En línea]. Available: <https://starkcloud.com/starkcloud-blog/que-es-el-desarrollo-de-software/>. [Último acceso: 05 Febrero 2025].
- [18] S. Piccinno, «Right People Group,» [En línea]. Available:
] <https://rightpeoplegroup.com/es/blog/que-es-el-desarrollo-de-software-guia-para-gestores-de-proyectos-informaticos>. [Último acceso: 05 Febrero 2025].
- [19] «Desarrollo de software,» 2024. [En línea]. Available: ¿Qué es el SDLC? - Explicación del ciclo de vida del desarrollo de software - AWS. (s. f.). Amazon Web Services, Inc.
] <https://aws.amazon.com/es/what-is/sdlc/>.

VIII. ANEXOS

FICHA TÉCNICA DEL SERVICIO DE ACTUALIZACION PROFESIONAL Y FORMACIÓN DOCENTE M00-SC-029-A02

Instituto Tecnológico de Apizaco

Nombre del Servicio:

Instructor(a):

- 1) Introducción:
- 2) Justificación:
- 3) Objetivo General:
- 4) Descripción del Servicio:

- a. Especificar tipo de servicio: Curso – Taller
- b. Duración en horas del curso:
- c. Contenido temático del curso

Temas / Subtemas (9)	Tiempo Programado (Hrs) (10)	Actividades de aprendizaje (11)

d. Elementos didácticos para el desarrollo del curso:

e. Criterio de evaluación:

No.	Criterio (13)	Valor (14)	Instrumento de evaluación (15)

5) Competencias a desarrollar:

6) Fuentes de Información:

Nombre y Firma del Facilitador(a)

Sello (20)

Nombre y Firma del jefe(a) de
Desarrollo Académico.

Ilustración 59 Ficha técnica

(s.f). Formato adquirido por la institución

Apizaco, Tlaxcala, **17/Junio/2024**
 Oficio No. MM-489/2024

Asunto: COMISIÓN.

VICENTE FLORES LARA
DOCENTE DEL DEPTO. DE METAL MECÁNICA
PRESENTE

Con base en su desarrollo profesional en el desempeño de sus funciones y aunado a su alto sentido de responsabilidad, se le **COMISIONA** a participar en los siguientes **cursos y/o actividades Intersemestrales**, durante el periodo comprendido del **10 de junio al 16 de agosto 2024**:

FECHA	CURSO/ACTIVIDAD	LUGAR	HORARIO
10-14 Jun/2024	Semana 1: Gestión del Curso.	Depto. MM.	09:00 a 17:00 hrs.
17-21 Jun/2024	Semana 2: Arte de servir en la Educación.	SALA MAGNA	09:00 a 14:00 hrs.
24-28 Jun/2024	Semana 3: Entrevista a Estudiantes de nuevo ingreso a la MIM.	DEP	09:00 a 17:00 hrs.
29 Jun-02 Ago/2024	Semana 4: Estrategias para el desarrollo sostenible.	SALA MAGNA	09:00 a 15:00 hrs.
05-09 Ago/2024	Semana 5: Uso de Software para el Diseño de equipos térmicos.	R-2	09:00 a 15:00 hrs.
12-16 Ago/2024	Semana 6: Instrumentación Didáctica.	Depto. MM.	09:00 a 15:00 hrs.

Sin otro particular por el momento, y deseándole el mayor de los éxitos, me despido de usted.

ATENTAMENTE
 Excelencia en Educación Tecnológica.
 Pensar para Servir. Servir para Triunfar.

JORGE AGUILAR VÁZQUEZ
JEFE DEL DEPTO. DE METAL MECÁNICA

ccp. Archivo

JAV*petry.

 SECRETARÍA DE EDUCACIÓN PÚBLICA
 INSTITUTO TECNOLÓGICO NACIONAL
 DE MÉXICO
 INSTITUTO TECNOLÓGICO DE APOZAC
 DEPARTAMENTO DE
 METAL MECÁNICA

 Apizaco, Tlaxcala, 17 de junio de 2024.
 Atte.
 JAV*petry.


Ilustración 60 Oficio de comisión

(s.f). Formato adquirido por la institución



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

**EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL INSTITUTO TECNOLÓGICO DE APIZACO**

OTORGA EL PRESENTE

RECONOCIMIENTO

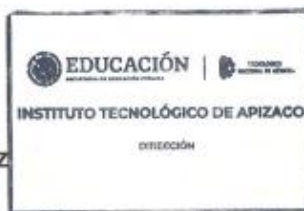
A

**FRANCISCO JAVIER
SÁNCHEZ PÉREZ**

POR IMPARTIR EL CURSO
ELABORACIÓN DE MATERIAL DIDÁCTICO DE CÁLCULO DIFERENCIAL
REALIZADO DEL 12 AL 16 DE JUNIO DE 2023
CON DURACIÓN DE 30 HORAS

AV. INSTITUTO TECNOLÓGICO NO. 418, AHUASHUATEPEC, TZOMPANTEPEC, TLAX., JUNIO 16 DE 2023.

**YESICA IMELDA SAAVEDRA BENÍTEZ
DIRECTORA**



TNM_209_034_2023/001



Ilustración 61 Constancia de acreditación del curso

(s.f). Formato adquirido por la institución

	Nombre del documento: Lista de Asistencia de Formación y Actualización Docente y Profesional		Código: ITAPI-AD-PO-	
	Referencia a la Norma ISO 9001:2015		Revisión: 1	
	7.2, 8.1, 8.2, 8.5		Página 1 de 1	

Folio:

CURSO

Nombre del curso: _____

Nombre del(a) instructor(a): _____

Periodo: _____ Duración: _____ Horario: _____

Sede: INSTITUTO TECNOLÓGICO DE APIZACO Plantel: INSTITUTO TECNOLÓGICO DE APIZACO

No.	Nombre del participante	Puesto y área de adscripción	M	H	Asistencia				
					L	M	M	J	V
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
14									
16									
15									

LIC. MARÍA DEL CARMEN NOLASCO VÁZQUEZ

Nombre y firma del(a) instructor(a) _____ Nombre y firma del(a) coordinador(a) _____

Ilustración 62 Pase de lista

(s.f). Formato adquirido por la institución

FORMATO CURRICULUM VITAE DEL INSTRUCTOR
M00-SC-029-A01

1. DATOS PERSONALES.

Nombre: (1)
Fecha de Nacimiento: (2)
CURP: (3)
RFC: (4)
Teléfono de contacto: (5)
Correo electrónico: (6)

2. FORMACIÓN ACADÉMICA (7)

Formación Académica	Institución	Titulación	Cedula Prof.
Licenciatura			
Maestría			
Doctorado			
Especialidad			
Otros estudios			

3. EXPERIENCIA LABORAL (9)

No.	Puesto	Empresa	Permanencia	Actividades
1				
2				
3				

4. EXPERIENCIA DOCENTE

No.	Matéria	Periodo
1		
2		
3		

5. PRODUCTOS EN ACADÉMICOS

No.	Actividad/Producto	Descripción	Fecha
1			
2			
3			

6. PARTICIPACIÓN COMO INSTRUCTOR

No.	TITULO	Institución o Empresa u Organización	Duración (horas)	Fecha
1				
2				
3				

Ilustración 63 Curriculum vitae del Instructor

(s.f). Formato adquirido por la institución

	Nombre del documento: Encuesta de Opinión de Formación y Actualización Docente y Profesional	Código: ITAPI-AD-PO-003-04
	Referencia a la Norma ISO 9001:2015 7.2,8.1,8.2,8.5	Revisión 1
	Página: 1 de 1	
Clave de identificación		

ENCUESTA DE OPINIÓN			
Nombre del Curso:		Fecha de realización:	
Nombre Institución:			
Nombre del Facilitador(a):			
La presente encuesta tiene como finalidad conocer su opinión sobre el curso de capacitación en el que participó, las respuestas nos servirán para mejorarlo.			
INSTRUCCIÓN: Solicitamos exprese su opinión sobre los siguientes aspectos escribiendo el número correspondiente en el recuadro de la derecha según la siguiente escala:			
5	Totalmente de acuerdo	4	Parcialmente de acuerdo
3	Indiferente	2	Parcialmente en desacuerdo
1	En desacuerdo		

INSTRUCTOR		CURSO	
1	Expuso el objetivo y temario del curso.	11	La distribución del tiempo fue adecuada para cubrir el contenido.
2	Mostró dominio del contenido abordado.	12	Los temas fueron suficientes para alcanzar el objetivo del curso.
3	Fomentó la participación del grupo.	13	El curso comprendió ejercicios de práctica relacionados con el contenido.
4	Aclaró las dudas que se presentaron.	14	El curso cubrió sus expectativas.
5	Dio retroalimentación a los ejercicios		
6	Aplicó una evaluación final relacionada con los contenidos del curso.		
7	Inició y concluyó puntualmente las sesiones.		

MATERIAL DIDÁCTICO		INFRAESTRUCTURA	
8	El material didáctico fue útil a lo largo del curso.	15	La iluminación del aula fue adecuada.
9	La impresión del material didáctico fue legible.	16	La ventilación del aula fue adecuada.
#	La variedad del material didáctico fue suficiente para apoyar su aprendizaje.	17	El aseo del aula fue adecuado.
		18	El servicio de los sanitarios fue adecuado (limpieza, abasto de papel, toallas, jabón, etc.).
		19	Recibió apoyo del personal que coordinó el curso.

COMENTARIOS O SUGERENCIAS	

¡Gracias!

(s.f). Formato adquirido por la institución

	Nombre del documento: Formato para Programa Institucional de Formación y Actualización Docente y Profesional		Código: ITAPI-AD-PO-003-06
			Revisión: 2
	Referencia a la Norma ISO 9001:2015 7.2,8.1,8.2,8.5		Página 1 de 9

INSTITUTO TECNOLÓGICO DE APIZACO
PROGRAMA INSTITUCIONAL DE FORMACIÓN Y ACTUALIZACIÓN DOCENTE Y PROFESIONAL
PERIODO: JUNIO/AGOSTO 2023

No.	Nombre del Servicio	Objetivo	Período de Realización	Lugar O Sede	No. De Horas X Curso	Instructor	Dirigido A	Perfil del Curso Formación Docente/Actualización Profesional

Elaboró Jefa de Desarrollo Académico	Aprobó Subdirección Académica
Cecilia Yarely Arreguin Lara	Luz Fabiola Sánchez Parra
Fecha: Junio/ 2024	

ITAPI-AD-PO-003-06

Rev.0

Ilustración 65 Documento “Programa de institución

(s.f). Formato adquirido por la institución