



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA
DIVISIÓN INGENIERÍA ELÉCTRICA

LABORATORIO DE INSTRUMENTACIÓN
ELECTRÓNICA DE SISTEMAS ESPACIALES

SITIO WEB DEL LABORATORIO DE
INSTRUMENTACIÓN ELECTRÓNICA DE
SISTEMAS ESPACIALES



SITIO WEB DEL LABORATORIO DE INSTRUMENTACIÓN ELECTRÓNICA DE SISTEMAS ESPACIALES

LIESE

19 de septiembre de 2025

Índice general

1. Introducción	4
1.1. Presentación del LIESE	4
1.2. Objetivos del sitio web	4
1.3. Alcance del sistema web	4
1.4. Justificación del desarrollo	5
2. Marco Teórico y Conceptual	6
2.1. Django Framework	6
2.2. Tecnologías Frontend	6
2.3. Base de Datos	6
2.4. Servicios Web	7
2.5. Despliegue	7
3. Análisis y Requerimientos	8
3.1. Requerimientos Funcionales	8
3.2. Requerimientos No Funcionales	8
3.3. Casos de Uso	9
4. Diseño y Arquitectura	10
4.1. Arquitectura MVT de Django	10
4.2. Modelos de Datos	11
4.3. Diseño de Base de Datos	12
4.4. Arquitectura de Templates	12
5. Implementación	13
5.1. Estructura del Proyecto Django	13
5.2. Modelos, Vistas y URLs	14
5.3. Templates y Frontend	14
5.4. Sistema de Administración	14
6. Sistema de Verificación y Seguridad	16
6.1. Autenticación de Dos Factores	16
6.2. Generación de Tokens de Verificación	16
6.3. Sistema de Correos HTML	16
7. Interfaz de Usuario y Experiencia	17
7.1. Diseño Responsivo	17
7.2. Páginas Principales	17
7.3. Sistema de Actividades	17
8. Panel de Administración	18
8.1. Django Admin Personalizado	18

8.2. Gestión de Contenido	18
8.3. Workflow de Publicación	18
9. Sistema Operativo Ubuntu Server	19
9.1. Introducción a Ubuntu Server	19
9.2. Configuración básica del servidor	19
9.3. Gestión de red (IP: 192.168.80.6)	19
10. Conexión SSH y Administración Remota	20
10.1. Protocolo SSH	20
10.2. Conexión al servidor	20
10.3. Administración remota	20
11. Servicios de Linux y Systemd	21
11.1. Conceptos de Servicios en Linux	21
11.2. Systemd	21
11.3. Servicio del Sitio Web LIESE	21
12. Despliegue y Configuración	22
12.1. Configuración de Desarrollo	22
12.2. Despliegue con Systemd	22
12.3. Configuración de Producción	22
12.4. Variables de Entorno	22
13. Mejoras y Trabajo Futuro	23
13.1. Seguridad y Certificación	23
13.2. Monetización y Patrocinio	23
13.3. DevOps y Automatización	23
13.4. Internacionalización	23
13.5. Analytics y Monitoreo	23
13.6. Funcionalidades Avanzadas	23
14. Conclusiones	24
14.1. Logros del Proyecto	24
14.2. Lecciones Aprendidas	24
14.3. Impacto y Aplicaciones	24
15. Referencias	25
15.1. Documentación Técnica	25
15.2. Recursos Académicos	26

Capítulo 1

Introducción

1.1. Presentación del LIESE

El Laboratorio de Instrumentación Electrónica de Sistemas Espaciales (LIESE) es una unidad académica adscrita a la División de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México (UNAM). Su misión es impulsar la formación de especialistas y el desarrollo de tecnología en el área espacial, con énfasis en el diseño, construcción e implementación de sistemas electrónicos para aplicaciones satelitales, instrumentación embebida, Internet de las Cosas (IoT) e inteligencia artificial. El LIESE promueve la colaboración interdisciplinaria y la innovación tecnológica, contribuyendo al avance científico y al bienestar social en México.

1.2. Objetivos del sitio web

El sitio web del LIESE tiene como objetivos principales:

- Difundir los proyectos, logros y actividades del laboratorio a la comunidad académica y al público en general.
- Facilitar el reclutamiento de nuevos talentos y la vinculación con otras instituciones.
- Proveer un canal de comunicación actualizado sobre eventos, oportunidades y noticias relevantes.
- Servir como plataforma de gestión y administración de contenido para los miembros del laboratorio.

1.3. Alcance del sistema web

El sistema web desarrollado abarca las siguientes funcionalidades:

- Publicación y gestión de artículos, tesis y noticias.
- Visualización de proyectos y actividades del laboratorio.
- Calendario de eventos académicos y de divulgación.
- Sección de líderes de proyecto y miembros del laboratorio.
- Sistema de oportunidades con autenticación de dos factores (2FA) para solicitudes.
- Panel de administración basado en Django para la gestión de contenido.
- Interfaz responsiva y moderna, accesible desde dispositivos móviles y de escritorio.

1.4. Justificación del desarrollo

El desarrollo de un sitio web institucional es fundamental para fortalecer la presencia digital del LIESE, facilitar la comunicación interna y externa, y promover la transparencia y el acceso a la información. La plataforma permite centralizar la gestión de contenido, automatizar procesos administrativos y ofrecer una experiencia de usuario moderna y segura. Además, la integración de mecanismos de seguridad como la verificación por correo electrónico y la autenticación de dos factores contribuye a la protección de los datos y la confianza de los usuarios.

Capítulo 2

Marco Teórico y Conceptual

2.1. Django Framework

Django es un framework de desarrollo web de alto nivel, escrito en Python, que fomenta el desarrollo rápido y limpio de aplicaciones web. Utiliza el patrón arquitectónico Modelo-Vista-Template (MVT), el cual separa la lógica de negocio, la presentación y el acceso a datos, facilitando la mantenibilidad y escalabilidad del sistema [1]. Django incluye un sistema de administración automática, un ORM (Object-Relational Mapping) para interactuar con bases de datos, y herramientas integradas para la gestión de usuarios, seguridad y envío de correos electrónicos. Entre sus características destacan la protección contra ataques comunes (CSRF, XSS, SQL Injection), el sistema de migraciones para la evolución del esquema de datos y la posibilidad de extender la funcionalidad mediante aplicaciones reutilizables. Es ampliamente utilizado en la industria y en el ámbito académico por su robustez, flexibilidad y comunidad activa.

El patrón MVT de Django se compone de:

- **Modelo:** Define la estructura de los datos y su representación en la base de datos.
- **Vista:** Gestiona la lógica de negocio y responde a las solicitudes del usuario.
- **Template:** Controla la presentación y el renderizado de la información al usuario final.

2.2. Tecnologías Frontend

El frontend del sitio web está construido utilizando HTML5, CSS3 y el framework Bootstrap 5.3.3. HTML5 proporciona la estructura semántica de las páginas, permitiendo una mejor accesibilidad y posicionamiento en buscadores. CSS3 permite el diseño visual, la adaptación responsiva a diferentes dispositivos y la implementación de animaciones y transiciones. Bootstrap es una biblioteca de componentes y utilidades CSS/JS que facilita la creación de interfaces modernas, responsivas y accesibles, acelerando el desarrollo y asegurando la compatibilidad multiplataforma [2]. Se emplean también animaciones CSS y JavaScript para mejorar la experiencia de usuario, así como la integración de iconos y recursos multimedia.

2.3. Base de Datos

Durante el desarrollo, el sistema utiliza SQLite como base de datos por su simplicidad y portabilidad. SQLite es una base de datos relacional embebida, ideal para entornos de desarrollo y pruebas. Para entornos de producción, se recomienda el uso de PostgreSQL, un sistema de gestión de bases de datos relacional robusto, escalable y con soporte avanzado para transacciones, concurrencia y extensiones [3]. Django abstrae el acceso a la base de datos mediante su ORM, permitiendo definir modelos de datos en Python y realizar migraciones automáticas. Esto facilita la evolución del esquema de datos y la portabilidad entre diferentes motores de

base de datos. El uso de migraciones garantiza la integridad y consistencia de los datos a lo largo del ciclo de vida del proyecto.

2.4. Servicios Web

El sitio web implementa servicios de correo electrónico para la verificación de usuarios y la autenticación de dos factores (2FA) en el sistema de oportunidades. Utiliza el protocolo SMTP para el envío de correos, configurado en el archivo de settings de Django [5]. Además, el sistema aprovecha las capacidades de seguridad integradas de Django, como la protección contra CSRF, la gestión de sesiones y la validación de formularios. El uso de formularios seguros y la validación del lado del servidor son fundamentales para prevenir ataques y garantizar la integridad de la información.

2.5. Despliegue

El despliegue del sistema se realiza sobre servidores Ubuntu, un sistema operativo de código abierto ampliamente utilizado en entornos de servidores por su estabilidad, seguridad y soporte a largo plazo (LTS) [4]. El servicio se gestiona mediante Systemd, utilizando el comando `runserver` de Django para exponer la aplicación en la red local, de acuerdo con el archivo de unidad real utilizado en el proyecto. El servicio se ejecuta bajo un usuario específico, define el directorio de trabajo y reinicia automáticamente en caso de fallo. Los archivos estáticos y media se sirven mediante la configuración de Django. Aunque en entornos de producción se recomienda el uso de Gunicorn y Nginx para mayor robustez y rendimiento, en este despliegue se utiliza el servidor de desarrollo de Django para simplificar la administración. El proceso de despliegue incluye la migración de la base de datos y la recolección de archivos estáticos, siguiendo buenas prácticas de administración de servicios en Linux.

Capítulo 3

Análisis y Requerimientos

3.1. Requerimientos Funcionales

Los requerimientos funcionales describen las capacidades y servicios que el sistema debe proporcionar a los usuarios finales. Para el sitio web del Laboratorio de Instrumentación Electrónica de Sistemas Espaciales (LIESE), se identifican los siguientes:

- Registro y gestión de miembros del laboratorio, incluyendo líderes de proyecto y administradores.
- Visualización de información sobre proyectos, artículos, eventos y noticias.
- Solicitud de oportunidades académicas (investigación, tesis, maestría, servicio social) mediante formularios web y verificación de correo electrónico.
- Publicación y administración de artículos, eventos y noticias por parte de usuarios autorizados.
- Sistema de autenticación y autorización para acceso a panel de administración y funcionalidades restringidas.
- Envío de correos electrónicos automáticos para verificación y notificaciones relevantes.
- Gestión de archivos multimedia (imágenes, documentos) asociados a miembros, proyectos y artículos.
- Interfaz de usuario responsiva y accesible para diferentes dispositivos.

3.2. Requerimientos No Funcionales

Los requerimientos no funcionales establecen criterios de calidad, restricciones y condiciones bajo las cuales el sistema debe operar:

- Seguridad: Protección contra ataques comunes (CSRF, XSS, SQL Injection) y gestión segura de sesiones y contraseñas [1].
- Rendimiento: Respuesta eficiente a las solicitudes de los usuarios y manejo adecuado de archivos estáticos y multimedia.
- Escalabilidad: Capacidad de migrar de SQLite a PostgreSQL para soportar mayor volumen de datos y usuarios [3].
- Mantenibilidad: Código estructurado siguiendo el patrón MVT de Django y uso de migraciones para la evolución del esquema de datos.
- Disponibilidad: Despliegue en servidores Ubuntu con servicios gestionados por Systemd y balanceo de carga mediante Nginx [4, 6].

- **Accesibilidad:** Cumplimiento de estándares web y diseño responsivo con Bootstrap [2].
- **Usabilidad:** Interfaz intuitiva y documentación clara para usuarios y administradores.

3.3. Casos de Uso

Los casos de uso describen las interacciones principales entre los usuarios y el sistema. A continuación se presentan los casos de uso más relevantes:

- **Registro de oportunidad académica:** Un visitante completa el formulario de solicitud, recibe un correo de verificación y, tras confirmar su correo, su solicitud es registrada y notificada a los administradores.
- **Gestión de miembros:** Un administrador agrega, edita o elimina miembros y líderes de proyecto desde el panel de administración.
- **Publicación de artículos y eventos:** Usuarios autorizados crean y editan artículos, eventos y noticias, incluyendo la carga de imágenes y documentos.
- **Visualización de información:** Cualquier usuario puede consultar información pública sobre proyectos, artículos, eventos y miembros del laboratorio.
- **Autenticación y acceso:** Los administradores y editores acceden a funcionalidades restringidas mediante autenticación segura.

Capítulo 4

Diseño y Arquitectura

4.1. Arquitectura MVT de Django

El sistema está construido siguiendo el patrón Modelo-Vista-Template (MVT) de Django, que separa la lógica de negocio, la presentación y el acceso a datos [1]. Esta arquitectura facilita la mantenibilidad, escalabilidad y reutilización de componentes. El modelo representa la estructura y reglas de los datos, la vista gestiona la lógica de negocio y las respuestas a las solicitudes del usuario, y el template define la presentación visual de la información.

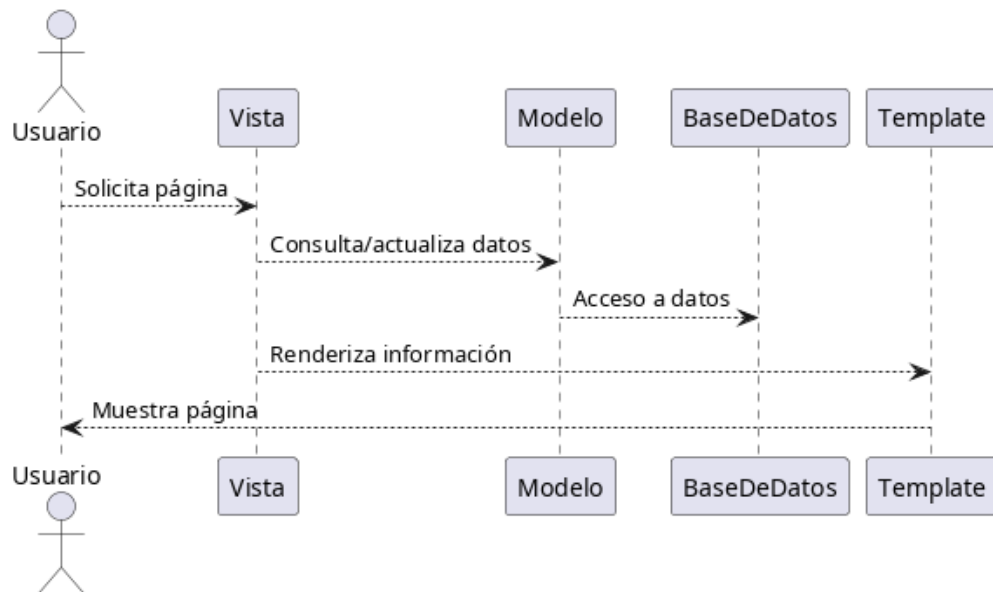


Figura 4.1: Diagrama UML de la arquitectura MVT de Django para el sitio LIESE.

4.2. Modelos de Datos

El modelo de datos del sitio LIESE incluye entidades como Miembro, Proyecto, Artículo, Evento, Noticia y Solicitud de Oportunidad. Cada entidad se implementa como una clase en Django, con atributos que representan los campos de la base de datos y relaciones entre modelos (por ejemplo, un artículo tiene un autor que es un miembro, y un proyecto puede tener varios miembros asociados). El uso del ORM de Django permite definir, consultar y migrar los modelos de manera eficiente y segura.

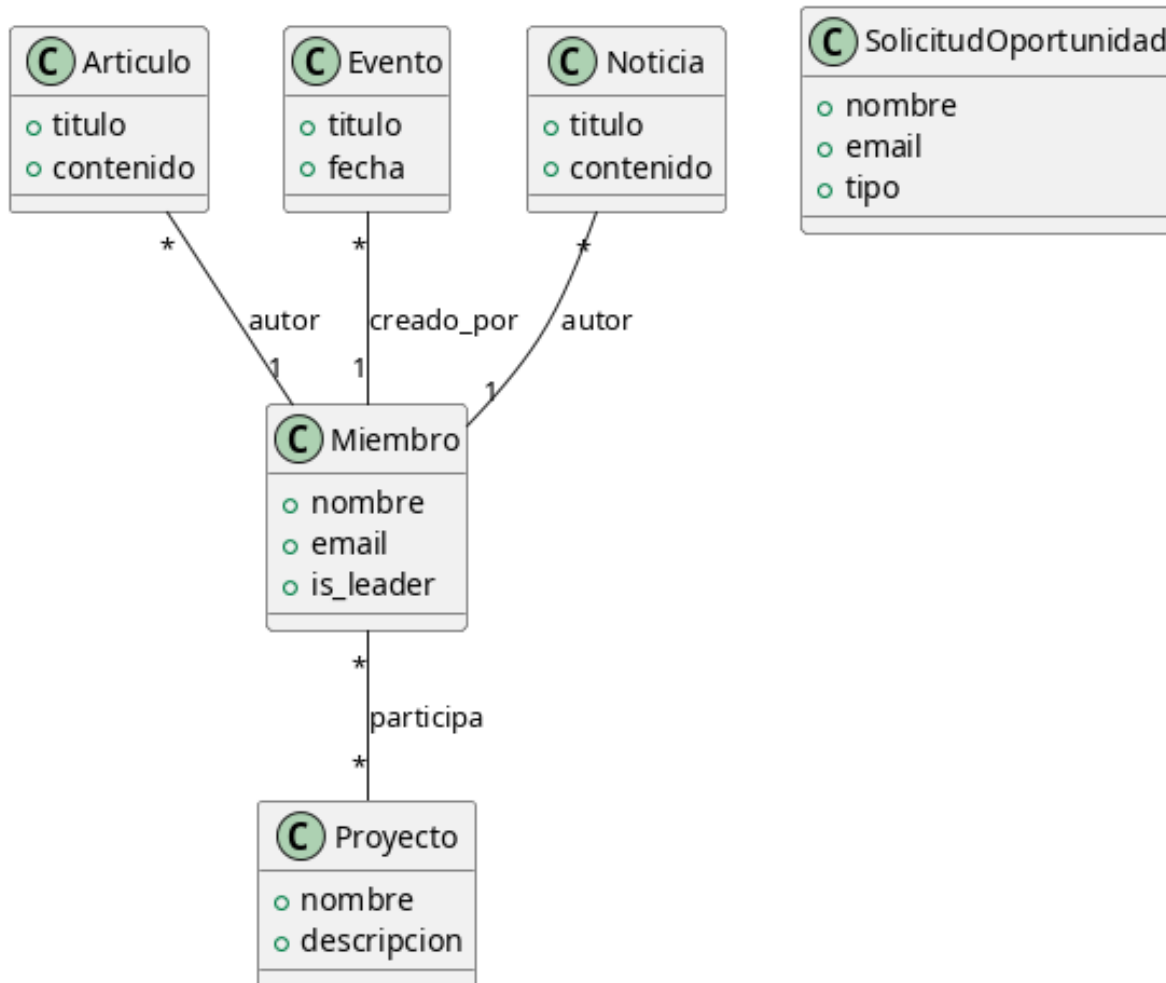


Figura 4.2: Diagrama UML de clases de los modelos principales del sistema LIESE.

4.3. Diseño de Base de Datos

La base de datos se diseña para soportar la gestión de información académica y administrativa del laboratorio. Se emplea SQLite en desarrollo y PostgreSQL en producción, aprovechando la portabilidad y robustez de ambos motores [3]. Las migraciones de Django permiten evolucionar el esquema de datos sin pérdida de información. El diseño contempla claves foráneas para mantener la integridad referencial y relaciones muchos a muchos (por ejemplo, miembros y proyectos).

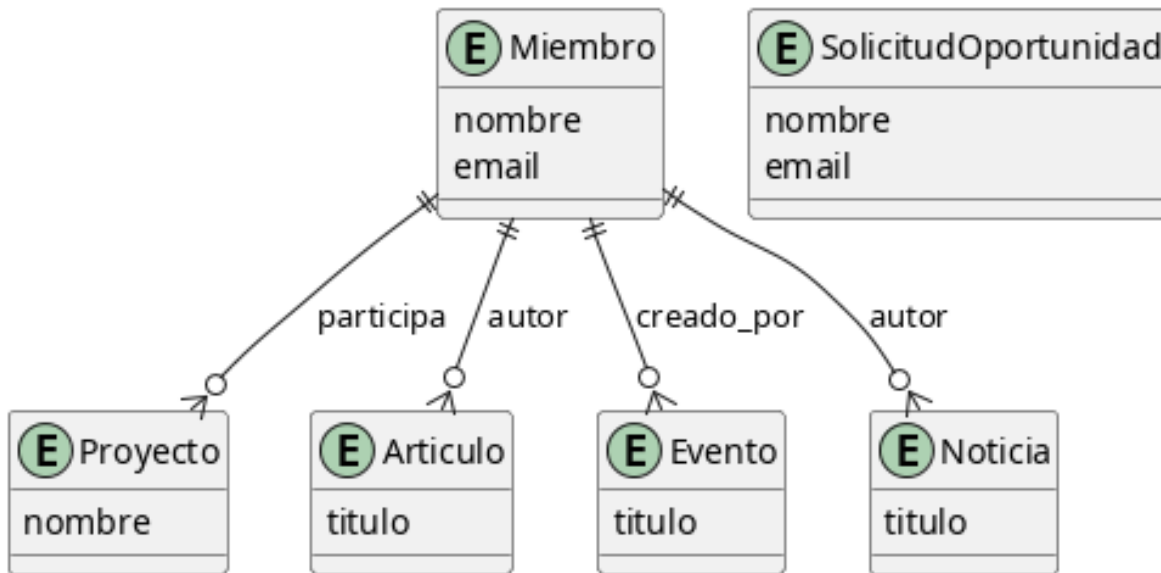


Figura 4.3: Diagrama entidad-relación (ER) simplificado de la base de datos.

4.4. Arquitectura de Templates

La presentación del sitio se implementa mediante el sistema de templates de Django, que permite separar la lógica de presentación del código Python. Se utilizan plantillas HTML con etiquetas y filtros de Django para renderizar información dinámica, y se heredan estructuras comunes como la barra de navegación y el pie de página. El uso de Bootstrap facilita el diseño responsivo y accesible [2].

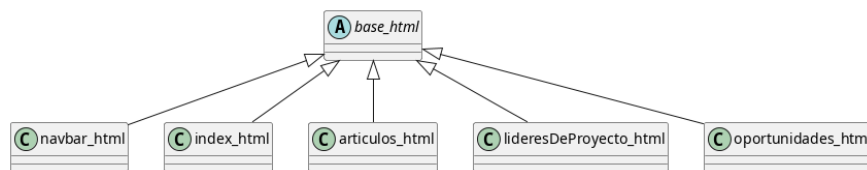


Figura 4.4: Diagrama de herencia de templates en el sistema LIESE.

Capítulo 5

Implementación

5.1. Estructura del Proyecto Django

La estructura del proyecto sigue la convención estándar de Django, separando la configuración, la aplicación principal y los recursos estáticos y de plantillas. A continuación se muestra un diagrama UML de la organización de carpetas y archivos principales:

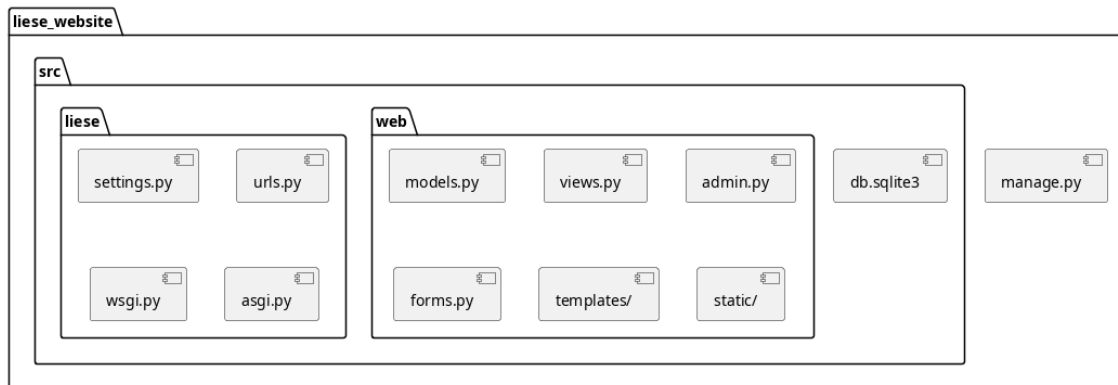


Figura 5.1: Estructura de carpetas y archivos principales del proyecto Django LIESE.

5.2. Modelos, Vistas y URLs

El flujo de interacción entre usuario, vistas y modelos se ilustra en el siguiente diagrama UML de secuencia, usando como ejemplo el proceso de solicitud de oportunidad académica:

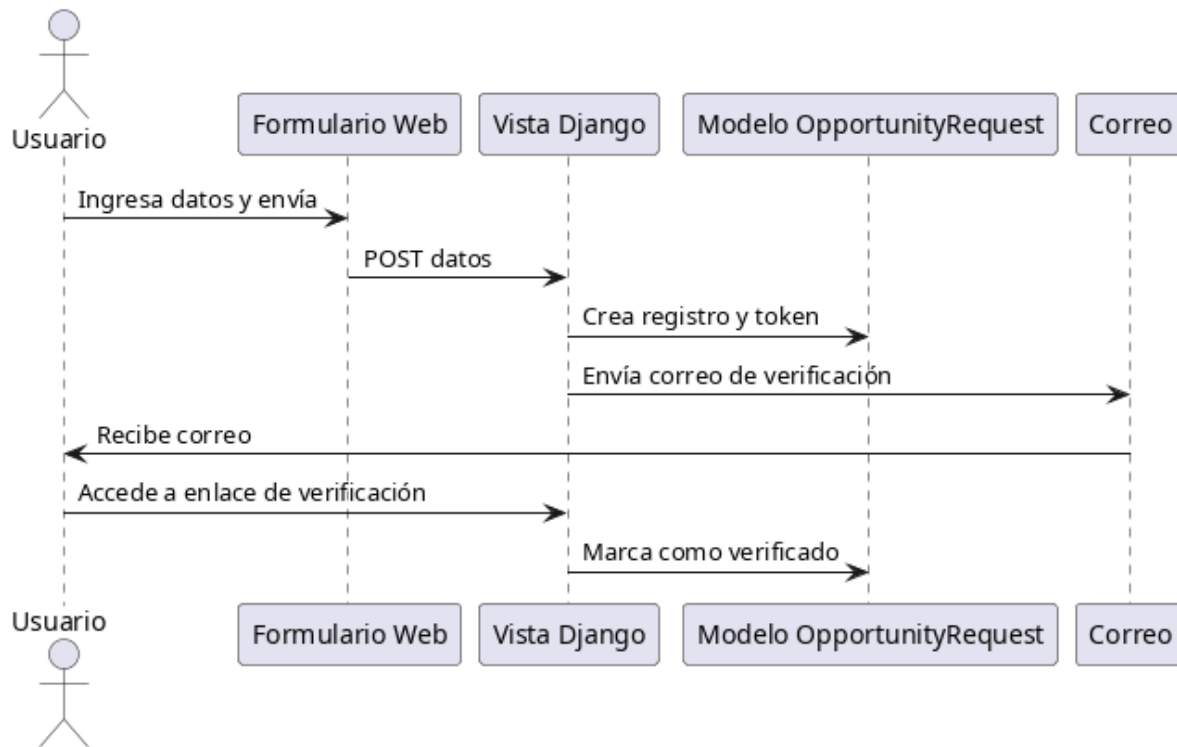


Figura 5.2: Diagrama de secuencia UML para el flujo de solicitud de oportunidad.

5.3. Templates y Frontend

El frontend utiliza el sistema de templates de Django y Bootstrap para la presentación. La herencia de plantillas y la organización de los archivos HTML ya se ilustró en el capítulo de arquitectura.

5.4. Sistema de Administración

El panel de administración de Django permite gestionar usuarios, miembros, proyectos y contenidos. El siguiente diagrama UML de colaboración muestra la interacción entre el administrador, el panel y los modelos:

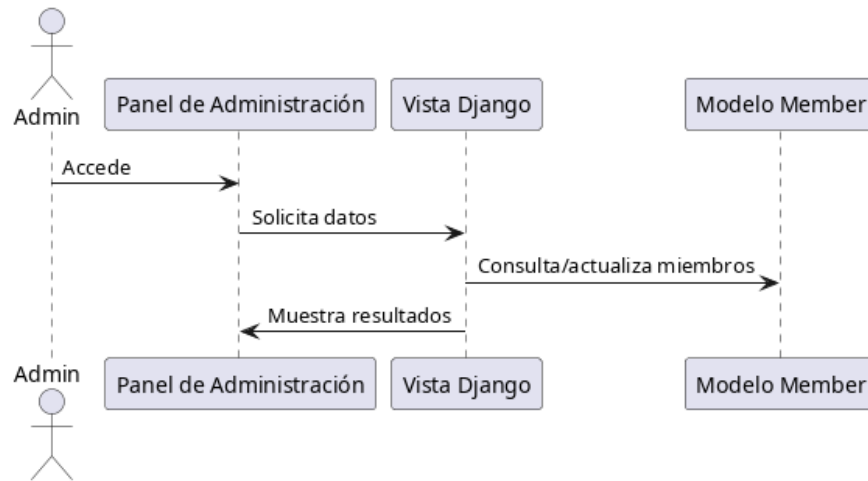


Figura 5.3: Diagrama de colaboración UML para la administración de miembros.

Capítulo 6

Sistema de Verificación y Seguridad

- 6.1. Autenticación de Dos Factores
- 6.2. Generación de Tokens de Verificación
- 6.3. Sistema de Correos HTML

Capítulo 7

Interfaz de Usuario y Experiencia

- 7.1. Diseño Responsivo
- 7.2. Páginas Principales
- 7.3. Sistema de Actividades

Capítulo 8

Panel de Administración

8.1. Django Admin Personalizado

8.2. Gestión de Contenido

8.3. Workflow de Publicación

Capítulo 9

Sistema Operativo Ubuntu Server

9.1. Introducción a Ubuntu Server

9.2. Configuración básica del servidor

9.3. Gestión de red (IP: 192.168.80.6)

Capítulo 10

Conexión SSH y Administración Remota

10.1. Protocolo SSH

10.2. Conexión al servidor

10.3. Administración remota

Capítulo 11

Servicios de Linux y Systemd

11.1. Conceptos de Servicios en Linux

11.2. Systemd

11.3. Servicio del Sitio Web LIESE

Capítulo 12

Despliegue y Configuración

- 12.1. Configuración de Desarrollo
- 12.2. Despliegue con Systemd
- 12.3. Configuración de Producción
- 12.4. Variables de Entorno

Capítulo 13

Mejoras y Trabajo Futuro

- 13.1. Seguridad y Certificación
- 13.2. Monetización y Patrocinio
- 13.3. DevOps y Automatización
- 13.4. Internacionalización
- 13.5. Analytics y Monitoreo
- 13.6. Funcionalidades Avanzadas

Capítulo 14

Conclusiones

14.1. Logros del Proyecto

14.2. Lecciones Aprendidas

14.3. Impacto y Aplicaciones

Capítulo 15

Referencias

15.1. Documentación Técnica

- Django Software Foundation. (2024). *Django documentation*. <https://docs.djangoproject.com/en/5.1/> [1]
- Bootstrap. (2024). *Bootstrap 5.3 Documentation*. <https://getbootstrap.com/docs/5.3/getting-started/introduction/> [2]
- The PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*. <https://www.postgresql.org/docs/> [3]
- Canonical Ltd. (2024). *Ubuntu Documentation*. <https://ubuntu.com/server/docs> [4]
- Django Software Foundation. (2024). *Django Email*. <https://docs.djangoproject.com/en/5.1/topics/email/> [5]
- Nginx, Inc. (2024). *Nginx Documentation*. <https://nginx.org/en/docs/> [6]

Bibliografía

- [1] Django Software Foundation. (2024). Django documentation. <https://docs.djangoproject.com/en/5.1/>
- [2] Bootstrap. (2024). Bootstrap 5.3 Documentation. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- [3] The PostgreSQL Global Development Group. (2024). PostgreSQL Documentation. <https://www.postgresql.org/docs/>
- [4] Canonical Ltd. (2024). Ubuntu Documentation. <https://ubuntu.com/server/docs>
- [5] Django Software Foundation. (2024). Django Email. <https://docs.djangoproject.com/en/5.1/topics/email/>
- [6] Nginx, Inc. (2024). Nginx Documentation. <https://nginx.org/en/docs/>

15.2. Recursos Académicos