

REPÚBLICA DEL ECUADOR



**INSTITUTO SUPERIOR
TECNOLÓGICO TENA**
Tecnología, Innovación y Desarrollo

**CARRERA DE TECNOLOGÍA SUPERIOR EN
DESARROLLO DE SOFTWARE**

**DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE
PROCESOS EN LA UNIDAD DE INTEGRACIÓN CURRICULAR DEL
INSTITUTO SUPERIOR TECNOLÓGICO TENA**

Trabajo de Integración Curricular, presentado como requisito parcial para optar por el título de Tecnólogo Superior en Desarrollo de Software.

AUTOR: Caicedo Rivadeneyra Anderson Guillermo

TUTOR: Ing. Quilumba Shiguango Salomón Isaac

Tena - Ecuador

2024-IIS

APROBACIÓN DEL TUTOR

Ing. Salomón Quilumba

PROFESOR DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA.

CERTIFICA:

En calidad de Tutor del Proyecto Integrador denominado: **Desarrollo de un Sistema Web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena**, de autoría del Sr. **Caicedo Rivadeneyra Anderson Guillermo**, con CC. **1501196222** estudiante de la Carrera de **Tecnología Superior en Desarrollo de Software del Instituto Superior Tecnológico Tena**, **CERTIFICO** que se ha realizado la revisión prolija del Trabajo antes citado, cumple con los requisitos de fondo y de forma que exigen los respectivos reglamentos e instituciones.

Tena, 10 de febrero de 2025.



Ing. Salomón Quilumba

TUTOR DEL TIC

CERTIFICACIÓN DEL TRIBUNAL CALIFICADOR

Tena, 12 de marzo de 2025

Los Miembros del Tribunal de Grado abajo firmantes, certificamos que el Trabajo de Titulación denominado: **Desarrollo de un Sistema Web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena**, presentado por el Sr. **Caicedo Rivadeneyra Anderson Guillermo**, con CC: **1501196222**, estudiante de la Carrera de Tecnología Superior en Desarrollo de Software del Instituto Superior Tecnológico Tena, ha sido corregida y revisada; por lo que autorizamos su presentación.

Atentamente;



Firmado electrónicamente por:
**FAUSTO PANTALEON
CLAUDIO ESPIN**

Ing. Fausto Claudio

PRESIDENTE DEL TRIBUNAL



Firmado electrónicamente por:
**ÍTALO MARCELO LARA
PÍLLCO**

Ing. Ítalo Lara

MIEMBRO DEL TRIBUNAL



Firmado electrónicamente por:
**MARCO PATRICIO
GUANIPATIN RAMIREZ**

Ing. Marco Guanipatin

MIEMBRO DEL TRIBUNAL AUTORÍA

DECLARACIÓN DE AUTORÍA

Yo, **Anderson Guillermo Caicedo Rivadeneyra**, con CC: **1501196222**, declaro ser autor del presente Trabajo de Titulación denominado: **Desarrollo de un Sistema Web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena** y absuelvo expresamente al Instituto Superior Tecnológico Tena, y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo al Instituto Superior Tecnológico Tena, la publicación de mi trabajo de Titulación en el repositorio institucional- biblioteca Virtual.

AUTOR:



CAICEDO RIVADENEYRA ANDERSON GUILLERMO

CÉDULA: 150119622-2

FECHA: Tena, 10 de febrero de 2025.

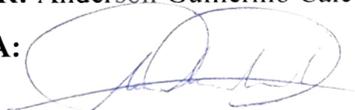
CARTA DE AUTORIZACIÓN POR PARTE DEL AUTOR

Yo, **Caicedo Rivadeneyra Anderson Guillermo, C.C.: 1501196222**, declaro ser autor del Trabajo de Titulación titulado: Desarrollo de un Sistema Web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena, como requisito para la obtención del Título de: TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE: autorizo al Sistema Bibliotecario del Instituto Superior Tecnológico Tena, para que con fines académicos, muestre al mundo la producción intelectual del Instituto, a través de la visualización de su contenido que consta en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio el Instituto. El Instituto Superior Tecnológico Tena, no se responsabiliza por el plagio o copia del presente trabajo que realice un tercero. Para constancia de esta autorización, en la ciudad de Tena, 02 de abril de 2020, firma el autor.

AUTOR: Anderson Guillermo Caicedo Rivadeneyra.

FIRMA:



CÉDULA: 150119622-2

DIRECCIÓN: Km 1 ½ vía (Tena - Archidona)

CORREO ELECTRÓNICO: anderson.caicedo@est.itstena.edu.ec

CELULAR: 0969473847

DATOS COMPLEMENTARIOS

TUTOR: Ing. Quilumba Shiguango Salomón Isaac

TRIBUNAL DEL GRADO:

Ing. Fausto Claudio. (Presidente).

Ing. Ítalo Lara. (Miembro).

Ing. Marco Guanipatin. (Miembro).

DEDICATORIA

Dedico este proyecto con todo mi amor y gratitud a las siguientes personas:

A mis padres, por ser mi mayor ejemplo de perseverancia, esfuerzo y amor incondicional. Sus sacrificios y enseñanzas me han permitido superar cada desafío que he enfrentado en este camino.

A mis hermanos, por su constante apoyo, palabras de aliento y por ser una fuente de inspiración para mí. Gracias por estar siempre presentes en los momentos más importantes.

A mis profesores y mentores, especialmente al Ing. Salomón Quilumba, cuya orientación y sabiduría han sido esenciales en el desarrollo de este proyecto. Su compromiso y paciencia me han guiado en cada etapa de mi formación profesional.

A mis amigos, por su compañía, su apoyo desinteresado y por ser parte fundamental de esta travesía. Su confianza en mí ha sido un motor para seguir adelante.

A todos aquellos que, de alguna forma, contribuyeron con su tiempo, conocimientos y palabras de ánimo a que este trabajo sea una realidad.

Finalmente, dedico este esfuerzo a Dios, quien me ha dado la fortaleza, la claridad y las oportunidades necesarias para alcanzar este objetivo.

Autor: Anderson Caicedo.

AGRADECIMIENTO

En primer lugar, agradezco a Dios por darme la fuerza y la sabiduría para completar este proyecto, y por permitirme superar cada reto con determinación y optimismo.

A mi familia, en especial a mis padres, por su apoyo incondicional, su paciencia y sus palabras de ánimo. Su amor y confianza han sido mi mayor motivación para concluir este trabajo.

A mi tutor, el Ing. Salomón Quilumba, por su valiosa orientación, sus aportes técnicos y su compromiso. Su experiencia fue clave para el desarrollo de este proyecto de sistema web.

A mis profesores del Instituto Superior Tecnológico Tena, por compartir sus conocimientos y por sembrar en mí la pasión por la tecnología y el desarrollo de soluciones innovadoras.

A la Unidad de TIC del Instituto Superior Tecnológico Tena, por facilitarme recursos y espacios que fueron esenciales para el análisis y diseño del sistema.

A mis compañeros y amigos, quienes, con su apoyo moral, ideas y sugerencias enriquecieron este trabajo. Su disposición y amistad han sido invaluable durante todo este proceso.

Finalmente, agradezco a todas las personas que, de una u otra manera, contribuyeron al logro de este proyecto. Su apoyo y colaboración serán siempre recordados con aprecio y gratitud.

Autor: Anderson Caicedo.

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	II
CERTIFICACIÓN DEL TRIBUNAL CALIFICADOR.....	III
DECLARACIÓN DE AUTORÍA.....	IV
CARTA DE AUTORIZACIÓN POR PARTE DEL AUTOR.....	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
ÍNDICE GENERAL	VIII
ÍNDICE DE FIGURAS.....	XI
A. TEMA.....	1
B. RESUMEN	2
C. FUNDAMENTACIÓN DEL TEMA.....	4
3.1. NECESIDAD.....	4
1.1. ACTUALIDAD	4
3.1. IMPORTANCIA	4
3.2. PRESENTACIÓN DEL PROBLEMA PROFESIONAL A RESPONDER	4
3.5. DELIMITACIÓN	5
3.5.1. Delimitación Espacial.....	5
3.5.2. Delimitación Temporal.....	5
3.5.3. Delimitación Técnica.....	5
3.5.4. Unidades de Observación	5
3.6. BENEFICIARIOS	6
3.6.1. Directos.....	6
3.6.2. Indirectos	6
D. OBJETIVOS.....	7
4.1. OBJETIVO GENERAL	7
4.2. OBJETIVOS ESPECÍFICOS.....	7
E. ASIGNATURAS INTEGRADORAS.....	8
F. FUNDAMENTACIÓN TEÓRICA	9
6.1. DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN DE PROCESOS EN LA UNIDAD DE INTEGRACIÓN CURRICULAR DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA	9
6.2. MARCO LEGAL.....	12
6.3. MARCO CONCEPTUAL	14
G. METODOLOGÍA	17
7.1. MATERIALES	17
7.2. UBICACIÓN DEL ÁREA DE ESTUDIO.....	18
7.3. TIPO DE DESARROLLO	19
7.4. METODOLOGÍA PARA CADA OBJETIVO.	19
H. RESULTADOS.....	25

8.1.	RESULTADOS DE LOS REQUERIMIENTOS PARA EL DESARROLLO DEL SOFTWARE	25
8.2.	RESULTADOS DE LA APLICACIÓN DE LA INTERFAZ DEL SOFTWARE	26
8.4.	RESULTADOS DE LA CODIFICACIÓN EN EL LENGUAJE DE PROGRAMACIÓN PYTHON Y DJANGO	27
8.5.	ANÁLISIS DE LAS TECNOLOGÍAS SELECCIONADAS	28
8.6.	CAPTURAS DE PANTALLA DEL SISTEMA	31
I.	CONCLUSIONES	35
J.	RECOMENDACIONES	36
K.	BIBLIOGRAFÍA	37
11.1.	REFERENCIAS	37
L.	ANEXOS	38
	MANUAL TÉCNICO DE KATINATEC	39
	<i>INTRODUCCIÓN</i>	40
	<i>ESTRUCTURA DEL PROYECTO</i>	42
	<i>ESTRUCTURA DEL CÓDIGO</i>	45
	<i>HTML RELACIONADO CON SCHEDULE Y SCHEDULEDETAIL</i>	48
	MANUAL DE USUARIO DE KATINATEC	58
	<i>INTRODUCCIÓN</i>	59
	<i>PÁGINA DE INICIO DE KATINATEC (PÁGINA PRINCIPAL)</i>	62
	<i>PÁGINA DE REGISTRO DE USUARIO:</i>	64
	<i>PÁGINA DE INICIO DE SESIÓN:</i>	66
	<i>PÁGINA DE RECUPERACIÓN DE CONTRASEÑA:</i>	67
	<i>PÁGINA DE INICIO (HOME):</i>	69
	<i>PÁGINA DE INTENTO DE ENCUESTA</i>	71
	<i>PÁGINA DE PERFIL DEL USUARIO (USUARIO)</i>	73
	<i>PÁGINA DE ACTUALIZACIÓN DE PERFILES (USUARIO)</i>	75
	<i>PREGUNTAS FRECUENTES (FAQ)</i>	77
	<i>SOPORTE TÉCNICO</i>	79
	<i>CONCLUSIÓN</i>	80
	IMÁGENES DE ACTIVIDADES REALIZADAS	81

ÍNDICE DE TABLAS

<i>Tabla 1: Asignaturas integradoras.....</i>	8
<i>Tabla 2: Tabla Comparativa de Ventajas y Desventajas de las Tecnologías Utilizadas.</i>	28

ÍNDICE DE FIGURAS

<i>Figura 1: Ubicación del Instituto Superior Tecnológico Tena.</i>	<i>19</i>
<i>Figura 2: Diagrama Entidad-Relación del sistema KatinaTec.</i>	<i>21</i>
<i>Figura 3: Modelo de datos implementado en Django (Parte 1).</i>	<i>21</i>
<i>Figura 4: Modelo de datos implementado en Django (Parte 2).</i>	<i>22</i>
<i>Figura 5: Modelo de datos implementado en Django (Parte 3).</i>	<i>22</i>
<i>Figura 6: Modelo de datos implementado en Django (Parte 4).</i>	<i>23</i>
<i>Figura 7: Diagrama de la arquitectura del sistema MVC.....</i>	<i>23</i>
<i>Figura 8: Comparación de Tecnologías Web para el Desarrollo de Sistemas KatinaTec.</i>	<i>25</i>
<i>Figura 9: Diagrama Entidad-Relación del Sistema KatinaTec.....</i>	<i>26</i>
<i>Figura 10: Arquitectura del Sistema basada en el patrón MVC.</i>	<i>27</i>
<i>Figura 11: Gráfico de Tiempos de Respuesta y Rendimiento del Sistema.</i>	<i>27</i>
<i>Figura 12: Captura de Pantalla de la Interfaz Administrativa de Django.</i>	<i>28</i>
<i>Figura 13: Interfaz de Registro de Usuarios.</i>	<i>31</i>
<i>Figura 14: Interfaz de Inicio de Sesión del Sistema KatinaTec.</i>	<i>32</i>
<i>Figura 15: Panel del Usuario Estándar.</i>	<i>32</i>
<i>Figura 16: Panel del Administrador de Carrera.</i>	<i>33</i>
<i>Figura 17: Panel del Administrador Principal.....</i>	<i>33</i>
<i>Figura 18: Generación y Consulta de Reportes en el Sistema.</i>	<i>34</i>
<i>Figura 19: Gestión de Procesos Curriculares en KatinaTec.</i>	<i>34</i>

A. TEMA

Desarrollo de un sistema Web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena

B. RESUMEN

El presente trabajo aborda el desarrollo de KatinaTec, un sistema web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena. El problema identificado radica en la falta de herramientas tecnológicas que permitan optimizar la organización, seguimiento y análisis de los procesos curriculares. Este proyecto se desarrolló bajo un enfoque de diseño, analizando tecnologías de código abierto y definiendo los requerimientos funcionales y no funcionales del sistema. Se diseñó el modelado de base de datos y la arquitectura utilizando el framework Django, cumpliendo con estándares de software libre. Antes de implementar el proyecto, se llevaron a cabo pruebas funcionales preliminares con datos simulados, generando reportes dinámicos y demostrando la viabilidad del sistema para su futura implementación. Las conclusiones destacan que KatinaTec contribuya significativamente a la mejora en la gestión de la Unidad de Integración Curricular mediante la automatización de procesos, la generación de reportes en tiempo real y la promoción del uso de herramientas tecnológicas accesibles y sostenibles.

Palabras clave: gestión de procesos, sistema web, Django, software libre, integración curricular, automatización, tecnología educativa.

ABSTRACT

This project presents the development of KatinaTec, a web-based system designed to improve process management in the Curricular Integration Unit at the Instituto Superior Tecnológico Tena. The main issue identified is the lack of technological tools to streamline the organization, tracking, and analysis of curricular processes. To address this, the system was designed using a structured approach, analyzing open-source technologies and defining both functional and non-functional requirements. The database model and architecture were built using the Django framework, adhering to open-source software standards. Before full implementation, preliminary functional tests were carried out with simulated data, generating dynamic reports and demonstrating the system's feasibility for future deployment. The findings highlight that KatinaTec will significantly enhance institutional management by automating processes, providing real-time reporting, and promoting the use of accessible and sustainable technological tools.

Keywords: process management, web system, Django, open-source software, curricular integration, automation, educational technology.

Reviewed by:



**LANGUAGES
CENTER
ISPTENA
TENA - NAPO - ECUADOR**

LANGUAJE CENTER COORDINATOR

C. FUNDAMENTACIÓN DEL TEMA

3.1. Necesidad

El Instituto Superior Tecnológico Tena enfrenta desafíos en la gestión de los procesos académicos y administrativos de la Unidad de Integración Curricular. Actualmente, los métodos manuales utilizados para gestionar la información son ineficientes, propensos a errores y carecen de una solución integral para el seguimiento adecuado de las actividades curriculares. La falta de automatización de estos procesos genera demoras, errores y dificultades para la toma de decisiones. La necesidad de un sistema web que centralice y facilite la gestión de esta información es fundamental para mejorar la eficiencia y la calidad de los procesos académicos y administrativos en la institución.

1.1. Actualidad

En la actualidad, el Instituto Superior Tecnológico Tena no contaba con un sistema automatizado para la gestión de los procesos de la Unidad de Integración Curricular, lo que afectaba el desarrollo de las actividades administrativas. La implementación de un sistema web como KatinaTec permitió agilizar la gestión de la información, mejorar el seguimiento de los procesos y facilitar la generación de reportes, brindando a los usuarios una interfaz intuitiva para acceder y actualizar datos clave. Esto no solo optimizó la gestión interna, sino que también contribuyó al fortalecimiento de la toma de decisiones.

3.1. Importancia

La implementación de un sistema web para la gestión de procesos en la Unidad de Integración Curricular fue crucial para mejorar la eficiencia de las actividades administrativas en el Instituto Superior Tecnológico Tena. Este sistema permitió gestionar de manera centralizada la información de los procesos, lo que redujo el riesgo de errores y facilitó el acceso a datos actualizados y relevantes. Además, la capacidad de generar reportes detallados ayudó en la toma de decisiones informadas. La automatización de estos procesos fortaleció la organización interna de la institución y contribuyó a un entorno educativo más eficiente.

3.2. Presentación del problema profesional a responder

La Unidad de Integración Curricular del Instituto Superior Tecnológico Tena carece de una herramienta tecnológica adecuada para gestionar los procesos

administrativos de manera eficiente. La falta de un sistema automatizado incrementa la probabilidad de errores en la administración de la información y dificulta el seguimiento de los procesos. Este problema afecta la toma de decisiones y la eficacia operativa, lo que motiva la necesidad de desarrollar un sistema web que optimice la gestión de estos procesos.

Campo: Tecnologías de la Información y Comunicación

Área: Unidad de Integración Curricular

Aspecto: Gestión de procesos administrativos

Sector: Instituto Superior Tecnológico Tena

Línea de investigación: Tecnologías de la Información y Comunicación

3.5. Delimitación

3.5.1. Delimitación Espacial

El proyecto se desarrolló en el Instituto Superior Tecnológico Tena, ubicado en la vía Tena-Archidona, km 1.5.

3.5.2. Delimitación Temporal

El trabajo se llevó a cabo durante el período académico 2024-IIS.

3.5.3. Delimitación Técnica

El sistema web propuesto fue desarrollado utilizando tecnologías de código abierto y software libre, lo que permitió su implementación de manera accesible y sostenible. La arquitectura del sistema se basó en el marco de trabajo Django, y se implementó una base de datos estructurada para gestionar la información de los procesos administrativos. Además, se realizaron pruebas de funcionalidad para asegurar la correcta operación del sistema.

3.5.4. Unidades de Observación

Las unidades de observación estuvieron enfocadas en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena, especialmente en la gestión de sus procesos académicos y administrativos.

3.6. Beneficiarios

3.6.1. Directos

Los principales beneficiarios directos son la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena y el Instituto Superior Tecnológico Tena. La implementación del sistema web mejoró la gestión interna de la unidad y optimizó los procesos académicos y administrativos.

3.6.2. Indirectos

Profesores

Estudiantes

D. OBJETIVOS

4.1. Objetivo General

Desarrollar un sistema web para la gestión de procesos de la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena.

4.2. Objetivos Específicos

- Analizar sobre las tecnologías web adecuadas para el desarrollo del sistema y definición de los requerimientos funcionales y no funcionales del sistema.
- Diseñar el modelado de base de datos y arquitectura del sistema utilizando tecnologías de código abierto y software libre.
- Realizar pruebas de funcionalidad del sistema mediante ingreso de datos y generación de reportes.

E. ASIGNATURAS INTEGRADORAS

Tabla 1: Asignaturas integradoras.

ASIGNATURAS INTEGRADORAS	
Asignaturas	Resultados de Aprendizaje
Base de Datos Avanzada	Elabora el modelamiento, ilustración y evaluación del proceso de base de datos.
Programación de Aplicaciones Web	Realiza el desarrollo de sus aplicaciones usando plataformas actuales de programación.
Tendencias Actuales de Programación	Realiza el desarrollo de sus aplicaciones usando plataformas actuales de programación. Tiene cultura de actualización permanentemente.

F. FUNDAMENTACIÓN TEÓRICA

6.1. Desarrollo de un sistema Web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena

La evolución de la tecnología ha transformado significativamente la manera en que las instituciones educativas administran y optimizan sus procesos académicos y administrativos. En este contexto, el desarrollo de un sistema web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena no solo aborda la necesidad de mejorar la eficiencia operativa, sino también de proporcionar un acceso más ágil y seguro a la información, facilitando la toma de decisiones basada en datos confiables.

- **Importancia de la gestión de procesos en la educación superior**

La gestión de procesos en la educación superior es fundamental para garantizar que las instituciones funcionen de manera eficiente, transparente y orientada a resultados. Este enfoque permite optimizar los recursos, minimizar los errores humanos y mejorar la experiencia tanto de estudiantes como de docentes. Además, asegura que los procesos académicos cumplan con los estándares de calidad y requisitos legales aplicables.

- **Automatización de procesos académicos y administrativos**

La automatización de procesos ayuda a reducir la carga administrativa manual, eliminando tareas repetitivas y propensas a errores. Esto incluye actividades como el registro de estudiantes, la validación de requisitos académicos, la programación de horarios, la generación de reportes y la gestión de evaluaciones. La automatización también promueve la consistencia y trazabilidad en la gestión de información.

- **Tecnologías web en la educación**

Las tecnologías web son herramientas clave para modernizar y mejorar la interacción entre los usuarios y los servicios institucionales. Un sistema basado en la web permite el acceso remoto, la colaboración en tiempo real y la integración de

módulos funcionales. En el contexto educativo, estas tecnologías pueden facilitar el seguimiento de los avances de los estudiantes, la gestión de tareas administrativas y la generación de reportes consolidados.

- **Diseño y desarrollo de sistemas web con Django**

Django, un framework de desarrollo web de código abierto, es una herramienta ideal para construir sistemas robustos y escalables. Su arquitectura de desarrollo rápido, junto con características como la seguridad integrada, la modularidad y la facilidad de uso, permite crear aplicaciones que cumplan con los requerimientos específicos de instituciones educativas. Además, Django promueve el cumplimiento de estándares de desarrollo y asegura la interoperabilidad con otras plataformas tecnológicas.

- **Base de datos para la gestión de información académica**

Un sistema web eficiente requiere una base de datos bien diseñada que permita almacenar y gestionar grandes volúmenes de información académica y administrativa. PostgreSQL, por ejemplo, es una excelente opción de código abierto que ofrece características avanzadas de rendimiento y seguridad. Una base de datos bien estructurada asegura la integridad de los datos, facilita la búsqueda rápida y soporta la generación de reportes dinámicos.

- **Interoperabilidad entre módulos y funcionalidades**

La interoperabilidad es crucial para garantizar que los diferentes módulos del sistema puedan comunicarse y compartir información de manera efectiva. Esto incluye la integración de módulos como el de registro de usuarios, la gestión de procesos académicos, la generación de reportes y el control de acceso. Al implementar estándares abiertos, se asegura que el sistema pueda adaptarse y crecer según las necesidades futuras de la institución.

- **Generación de reportes automatizados**

La generación de reportes automatizados permite a los administradores y usuarios obtener información relevante de manera rápida y precisa. Estos reportes pueden incluir estadísticas de procesos académicos, rendimiento de los estudiantes, análisis de asignaturas y métricas de eficiencia operativa. La automatización elimina la necesidad de consolidar manualmente la información, reduciendo los errores y ahorrando tiempo.

- **Impacto de los sistemas web en la eficiencia operativa**

La implementación de sistemas web mejora la eficiencia operativa al reducir los tiempos de procesamiento, optimizar los recursos y proporcionar acceso a la información en tiempo real. Esto permite a las instituciones educativas tomar decisiones más informadas y responder con mayor agilidad a las necesidades de estudiantes, docentes y personal administrativo.

- **Buenas prácticas en el desarrollo de sistemas web**

Para garantizar el éxito del sistema, es esencial seguir buenas prácticas en su diseño y desarrollo. Esto incluye:

Análisis de requerimientos detallado: comprender las necesidades específicas de los usuarios finales.

Diseño centrado en el usuario (UI/UX): asegurar que la plataforma sea intuitiva y accesible.

Pruebas rigurosas: validar la funcionalidad, seguridad y rendimiento del sistema.

Documentación completa: facilitar el mantenimiento y futuras actualizaciones del sistema.

Cumplimiento de estándares: adherirse a normativas legales y estándares de desarrollo internacionales.

6.2. Marco Legal

El marco legal que sustenta el desarrollo del sistema web para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena se basa en una variedad de normativas nacionales e internacionales que aseguran la calidad, seguridad y confidencialidad de los datos e información gestionados. A continuación, se presentan las principales leyes y normativas relevantes:

- **Constitución de la República del Ecuador**

La Constitución del Ecuador establece principios fundamentales que respaldan la educación y el acceso a tecnologías:

Derecho a la educación (Art. 26): Garantiza el acceso a una educación de calidad como un derecho humano fundamental.

Derecho al acceso universal a las tecnologías de la información y comunicación (Art. 35): Promueve la inclusión tecnológica como un derecho que contribuye al desarrollo y la igualdad de oportunidades.

Ley Orgánica de Educación Superior (LOES)

La LOES regula la gestión educativa y fomenta el uso de tecnologías en las instituciones de educación superior:

Promoción del uso de TIC: Estimula la modernización de los procesos educativos y administrativos mediante el uso de tecnologías.

- ✓ **Artículo 35:** Asigna recursos para la investigación, ciencia, tecnología e innovación, promoviendo la incorporación de herramientas tecnológicas en la educación superior (Ley Orgánica de Educación Superior [LOES], 2018, art. 35).
- ✓ **Artículo 36:** Dispone que universidades y escuelas politécnicas deben destinar presupuesto para infraestructura tecnológica y desarrollo de proyectos de investigación (LOES, 2018, art. 36).
- ✓ **Artículo 143:** Promueve la interconexión de bibliotecas y el acceso a bibliotecas digitales y archivos en línea, fomentando el uso de TIC para la educación y la investigación (LOES, 2018, art. 143).

- ✓ **Artículo 144:** Obliga a la digitalización de los trabajos de titulación, impulsando el uso de plataformas electrónicas para la gestión del conocimiento (LOES, 2018, art. 144).

Aseguramiento de la calidad: Garantiza la eficiencia y transparencia en los procesos académicos y administrativos, alineándose con estándares nacionales.

- ✓ **Artículo 94:** Define el Sistema Interinstitucional de Aseguramiento de la Calidad, que supervisa y evalúa el cumplimiento de estándares de calidad en la educación superior a través del Consejo de Educación Superior, el Consejo de Aseguramiento de la Calidad de la Educación Superior y las Instituciones de Educación Superior (LOES, 2018, art. 94).
- ✓ **Artículo 96:** Establece que las instituciones de educación superior deben implementar políticas efectivas de evaluación y mejora continua para garantizar la calidad de sus carreras y programas académicos (LOES, 2018, art. 96).
- **Reglamento General de Protección de Datos Personales**

Esta normativa establece medidas para proteger la privacidad y confidencialidad de los datos personales:

Protección de datos: Define principios de licitud, integridad y confidencialidad en el tratamiento de datos personales.

Obligaciones de las instituciones: Exige medidas técnicas y organizativas para garantizar la seguridad de los datos.

Sanciones por incumplimiento: Detalla penalidades por el manejo inadecuado de información.

- **Normativas Internacionales relacionadas con la gestión de procesos**

El proyecto puede alinearse con estándares internacionales reconocidos:

ISO 9001: Asegura la calidad en la gestión de procesos.

ISO/IEC 27001: Proporciona directrices para gestionar la seguridad de la información y minimizar riesgos.

- **Código de Ética del Instituto Superior Tecnológico Tena**

El código de ética de la institución establece principios sobre el uso responsable y ético de los sistemas de información, promoviendo una gestión transparente y confiable.

- ✓ **Artículo 101:** El Reglamento y Código de Ética establecieron que el Consejo de Aseguramiento de la Calidad de la Educación Superior aprobaría el reglamento que regula las actividades de los evaluadores y otros funcionarios. En este reglamento, el Código de Ética incluiría requisitos, incompatibilidades, prohibiciones y normas de selección.

- **Conclusiones del Marco Legal**

El marco legal proporciona una base sólida para el diseño e implementación del sistema web KatinaTec, asegurando que:

- ✓ Se garantice la protección de la información sensible de los estudiantes y del personal administrativo.
- ✓ Se cumplan las normativas nacionales e internacionales para promover la calidad y seguridad en la gestión de procesos.
- ✓ Se fomente un ambiente tecnológico seguro que contribuya al fortalecimiento académico y administrativo del Instituto Superior Tecnológico Tena.

6.3. Marco Conceptual

- **Automatización de Procesos**

La automatización de procesos se refiere a la utilización de tecnologías y sistemas informáticos para ejecutar tareas de manera eficiente, reduciendo la intervención humana. Según (Davenport & Short, 1990), esto permite optimizar recursos,

mejorar la precisión y minimizar tiempos de ejecución. Los sistemas automatizados son diseñados para reemplazar procesos manuales repetitivos, facilitando una mayor agilidad en la gestión.

En el contexto de la Unidad de Integración Curricular, la automatización busca gestionar eficientemente los procesos administrativos y académicos, garantizando que las tareas se ejecuten de manera estandarizada y con menor probabilidad de error humano.

- ✓ **BPM (Business Process Management):** Esta metodología se enfoca en analizar, modelar y automatizar procesos organizacionales. Permite identificar ineficiencias y diseñar flujos de trabajo más efectivos. Según (Weske, 2012), el BPM combina herramientas de software con buenas prácticas para lograr una gestión efectiva de los procesos.
- ✓ **Flujos de Trabajo:** Los flujos de trabajo son representaciones gráficas de los procesos que indican cómo se desarrollan las tareas, los responsables y los tiempos asignados. Según (Van Der Aalst et al., 2016), los flujos de trabajo son esenciales para garantizar la trazabilidad y consistencia en los procesos automatizados.
- **Optimización de Procesos**

La optimización de procesos implica identificar áreas de mejora y aplicar técnicas que permitan maximizar la eficiencia, eficacia y calidad de los resultados. Según (Hammer, 1990), la optimización es fundamental para mantener la competitividad en un entorno dinámico.

En el caso del sistema web, la optimización se logra mediante:

- ✓ **Eliminación de redundancias:** Identificar tareas duplicadas o innecesarias y eliminarlas.
- ✓ **Integración de tecnologías:** Uso de herramientas como Django y bases de datos relacionales para unificar y centralizar la información.
- ✓ **Minimización de errores:** Automatizando tareas críticas que requieren alta precisión, como generación de reportes y verificación de datos.
- **Sistemas de Gestión de Procesos**

Los sistemas de gestión de procesos (SGP) permiten automatizar, controlar y monitorizar el flujo de tareas dentro de una organización. Según (Smith & Fingar, 2003), estos sistemas proporcionan herramientas para modelar procesos, ejecutar tareas y medir el desempeño.

Componentes clave de los SGP:

1. **Motor de Procesos:** Es el núcleo del sistema que ejecuta las tareas automatizadas según los flujos de trabajo definidos.
 2. **Panel de Control:** Permite a los administradores supervisar el estado de los procesos en tiempo real y tomar decisiones informadas.
 3. **Base de Datos Centralizada:** Asegura la integridad y consistencia de la información utilizada en los procesos.
 4. **Herramientas de Análisis:** Facilitan la identificación de cuellos de botella y el monitoreo de indicadores clave de desempeño (KPI).
- **Django como Herramienta para la Automatización**

Django es un framework web de código abierto que facilita el desarrollo de aplicaciones escalables y seguras. Su arquitectura basada en el modelo-vista-controlador (MVC) permite gestionar procesos complejos de manera eficiente. Según (Holovaty & Kaplan-Moss, 2005), Django es ideal para la creación de sistemas personalizados, debido a su robustez y flexibilidad.

Ventajas de Django en la automatización de procesos:

1. **Rapidez de Desarrollo:** Ofrece componentes predefinidos que aceleran la implementación de funcionalidades clave.
 2. **Seguridad Incorporada:** Incluye medidas para proteger contra ataques comunes como inyecciones SQL y cross-site scripting (XSS).
 3. **Escalabilidad:** Soporta el manejo de grandes volúmenes de datos y usuarios concurrentes.
 4. **Facilidad de Integración:** Permite la conexión con herramientas externas y APIs para ampliar la funcionalidad del sistema.
- **Gestión de Información en Procesos**

La gestión de información es fundamental en la optimización de procesos, ya que permite almacenar, acceder y analizar datos de manera eficiente. Según (Turban et

al., 2011), los sistemas de gestión de información contribuyen a la toma de decisiones basadas en datos confiables.

En la Unidad de Integración Curricular, los datos manejados incluyen información académica, administrativa y reportes generados a partir de las actividades curriculares. La implementación de un sistema centralizado garantiza:

1. **Acceso Controlado:** Restringir el acceso a los datos según roles y permisos definidos.
2. **Integridad de Datos:** Asegurar que la información almacenada sea precisa y consistente.
3. **Generación de Reportes:** Automatizar la creación de reportes para agilizar la toma de decisiones.
- **Estrategias de Implementación para la Unidad de Integración Curricular**

Para optimizar los procesos mediante la automatización, se propone lo siguiente:

1. **Modelado de Procesos:** Utilizar herramientas como diagramas de flujo para mapear las actividades actuales y diseñar el sistema.
2. **Desarrollo Iterativo:** Implementar el sistema en fases, priorizando funcionalidades críticas como la gestión de registros y generación de reportes.
3. **Capacitación:** Formar al personal en el uso del sistema para garantizar su adopción exitosa.
4. **Pruebas y Ajustes:** Realizar pruebas funcionales y de usabilidad para identificar áreas de mejora antes del despliegue final.

G. METODOLOGÍA

7.1. Materiales

Para llevar a cabo este estudio, se utilizaron los siguientes materiales:

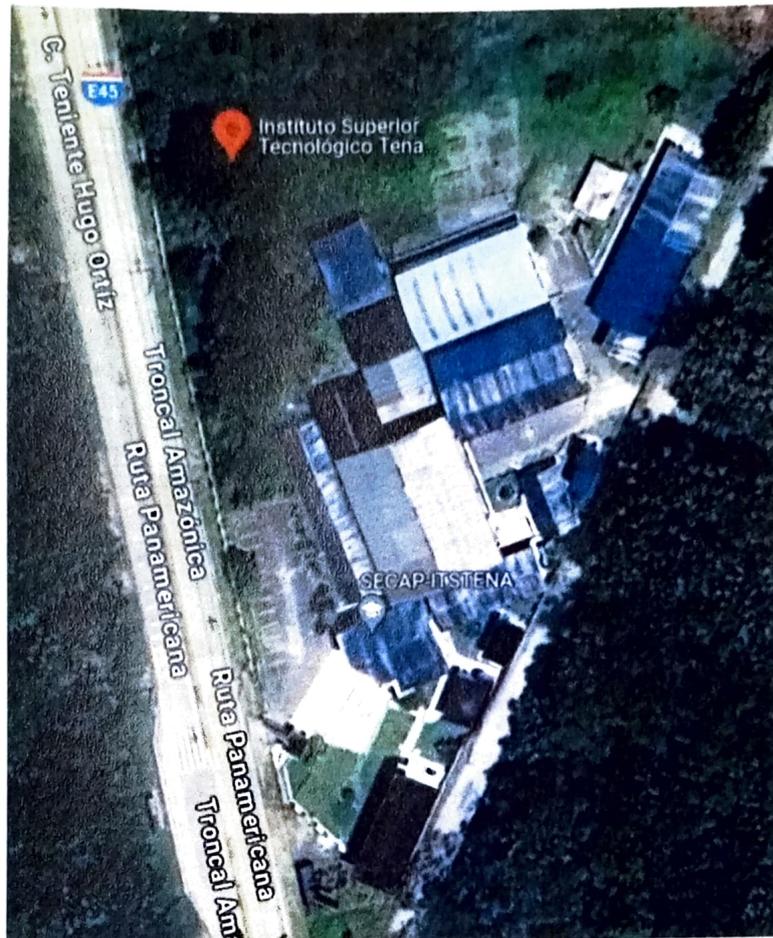
- ✓ **Computadoras y dispositivos móviles:** Para el desarrollo y las pruebas del sistema web, garantizando que sea accesible desde diferentes dispositivos.

- ✓ **Software de desarrollo:** Microsoft Visual Studio Code como entorno de desarrollo para la creación del sistema, con soporte para Python 3.13.2, HTML, Bootstrap 5.3.3, JavaScript y CSS.
- ✓ **Servidor en PythonAnywhere:** Para realizar las pruebas de funcionalidad del sistema, simulando el comportamiento en un entorno de producción sin desplegarlo completamente.
- ✓ **Base de datos:** Herramientas de base de datos para almacenar y procesar los datos de los procesos de la Unidad de Integración Curricular, utilizando SQLite (versión incluida en Python 3.13.2) para pruebas locales y PostgreSQL 17.4 como sistema de gestión de bases de datos principal.
- ✓ **Herramientas de análisis de datos:** Para procesar los datos generados durante las pruebas del sistema y realizar los reportes de funcionalidad, empleando bibliotecas en Python como Pandas y Matplotlib.

7.2. Ubicación del Área de estudio.

El proyecto se desarrolló en el Instituto Superior Tecnológico Tena, ubicado en Km 1 ½ vía (Tena - Archidona). Este lugar fue relevante, ya que aquí se implementaron y validaron los procesos automatizados mediante el sistema web.

Figura 1: Ubicación del Instituto Superior Tecnológico Tena.



Nota: Fuente Google Maps.

7.3. Tipo de desarrollo

El trabajo realizado fue un desarrollo de tipo experimental, en el que se diseñó, implementó y probó el sistema web en un entorno controlado, pero sin su despliegue final. El objetivo fue validar la funcionalidad del sistema mediante pruebas con datos reales.

7.4. Metodología para cada objetivo.

Objetivo Específico 1

Analizar sobre las tecnologías web adecuadas para el desarrollo del sistema y definición de los requerimientos funcionales y no funcionales del sistema.

Metodología:

1. Investigación de tecnologías:

- Se realizó un análisis comparativo de herramientas y frameworks disponibles en el mercado que se ajustaran a los requerimientos del sistema.
- Se priorizaron tecnologías de código abierto como Python, PostgreSQL, HTML, CSS, Bootstrap y JavaScript.

2. Recopilación de requerimientos:

- A través de reuniones con los responsables de la Unidad de Integración Curricular, se identificaron los requerimientos funcionales y no funcionales del sistema.
- Los requerimientos fueron documentados y validados con los usuarios finales.

3. Definición de estándares:

- Se establecieron lineamientos técnicos y de diseño para garantizaron un sistema robusto y escalable.

Objetivo Específico 2

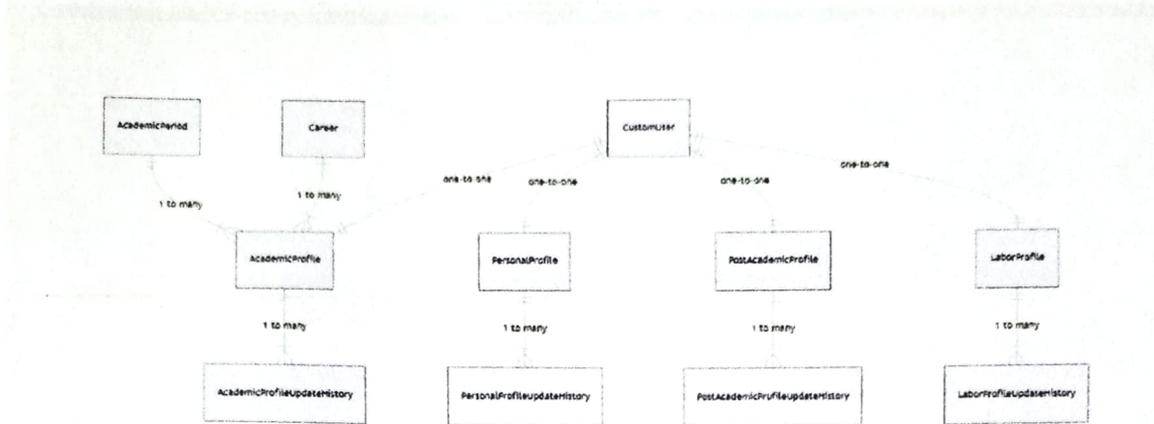
Diseñar el modelado de base de datos y arquitectura del sistema utilizando tecnologías de código abierto y software libre.

Metodología:

1. Diseño del modelo de datos:

- Se identificaron las entidades principales y sus relaciones mediante diagramas ER (Entidad-Relación).
- Se utilizó PostgreSQL como base de datos, garantizando su integración con Django para una administración eficiente.

Figura 2: Diagrama Entidad-Relación del sistema KatinaTec.

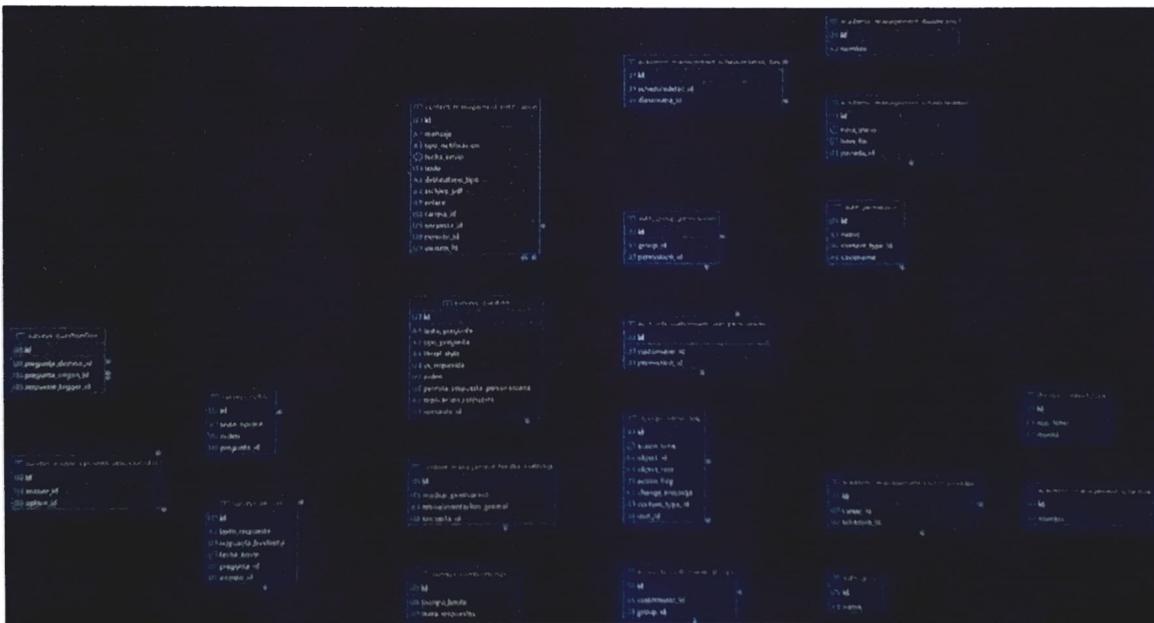


Nota: Diseñado en HTML con el uso de CSS y Bootstrap.

2. Modelado en Django:

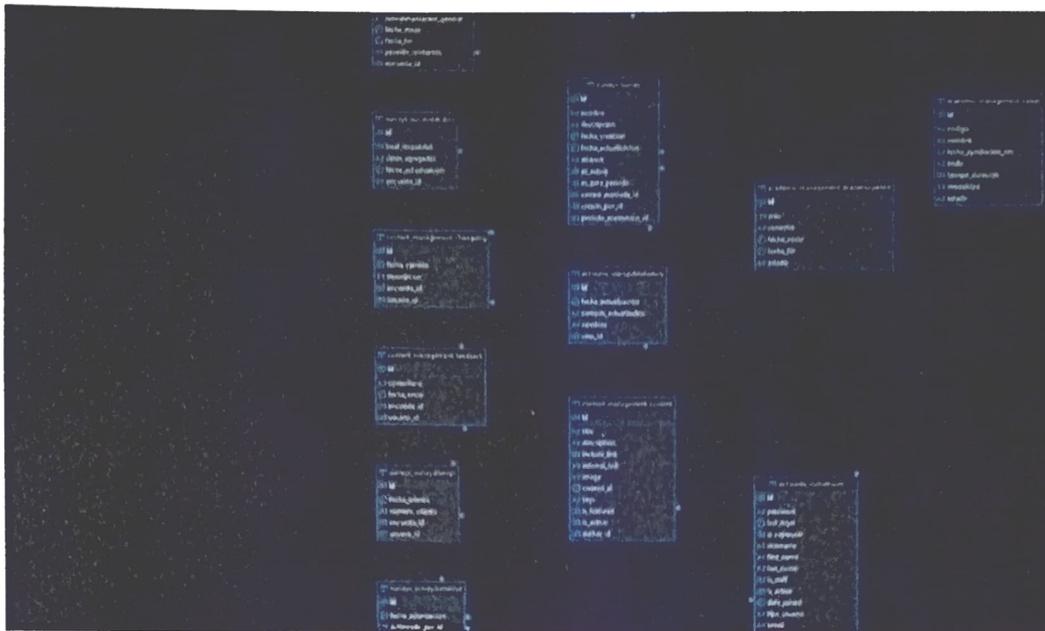
- Se tradujeron las estructuras del modelo ER en modelos ORM de Django.

Figura 3: Modelo de datos implementado en Django (Parte 1).



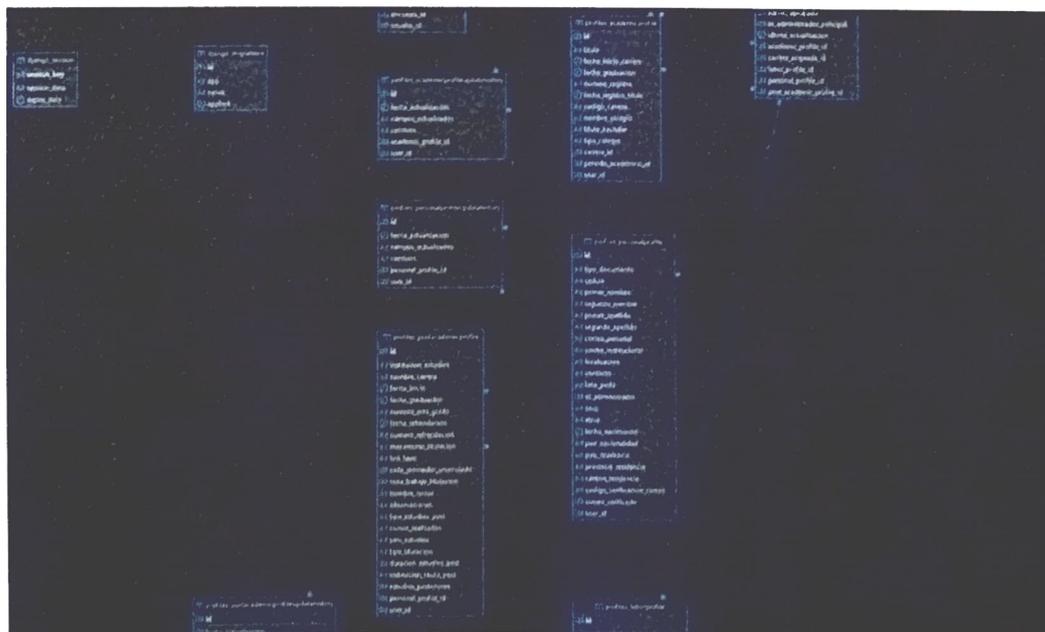
Nota: Captura en la Visual Studio Code con la extensión SQLite3 Editor.

Figura 4: Modelo de datos implementado en Django (Parte 2).



Nota: Captura del gestor de base de datos DBBeaver.

Figura 5: Modelo de datos implementado en Django (Parte 3).



Nota: Captura del gestor de base de datos DBBeaver.

1. Preparación del entorno de pruebas:

- Se utilizó un servidor en PythonAnywhere para alojar la aplicación y realizar pruebas en un entorno controlado.

2. Definición de casos de prueba:

- Se diseñaron casos de prueba que cubran los principales flujos del sistema, incluyendo:
 - Registro de procesos.
 - Generación de reportes.
 - Consulta y modificación de datos.

3. Ejecución de pruebas:

- Se probó el sistema en diferentes navegadores y dispositivos para evaluar la experiencia del usuario y garantizar la compatibilidad.
- Se utilizaron datos de prueba para validar la funcionalidad de los módulos principales.

4. Corrección de errores:

- Los errores encontrados durante las pruebas se documentaron y corrigieron antes de una nueva iteración de pruebas.

5. Evaluación final:

- Se generaron reportes detallados sobre el rendimiento y la estabilidad del sistema.

H. RESULTADOS

8.1. Resultados de los Requerimientos para el Desarrollo del Software

Descripción del Resultado:

Se realizó un análisis exhaustivo de las tecnologías web más adecuadas para el desarrollo del sistema, priorizando tecnologías de código abierto y software libre.

Las tecnologías seleccionadas fueron:

- ✓ Python 3.13.2 con Django 5.1.3 para el backend.
- ✓ Bootstrap 5.3.3 para el frontend.
- ✓ PostgreSQL 17.4 como sistema de gestión de bases de datos.

Requerimientos funcionales:

- ✓ Registro y autenticación de usuarios.
- ✓ Gestión de procesos curriculares.
- ✓ Generación de reportes automatizados con datos procesados del sistema.

Requerimientos no funcionales:

- ✓ Eficiencia.
- ✓ Seguridad.
- ✓ Usabilidad.

Figura 8: Comparación de Tecnologías Web para el Desarrollo de Sistemas KatinaTec.

Comparación de Tecnologías

Tecnología	Rendimiento	Escalabilidad	Comunidad de Soporte
Django	Alto	Alta	Grande y activa
Flask	Moderado	Media	Grande y activa
Node.js	Alto	Muy Alta	Muy grande
Spring Boot	Muy Alto	Muy Alta	Amplia y estable
Ruby on Rails	Moderado	Media	Comunidad fiel

Nota. Diseñado en HTML con el uso de CSS y Bootstrap.

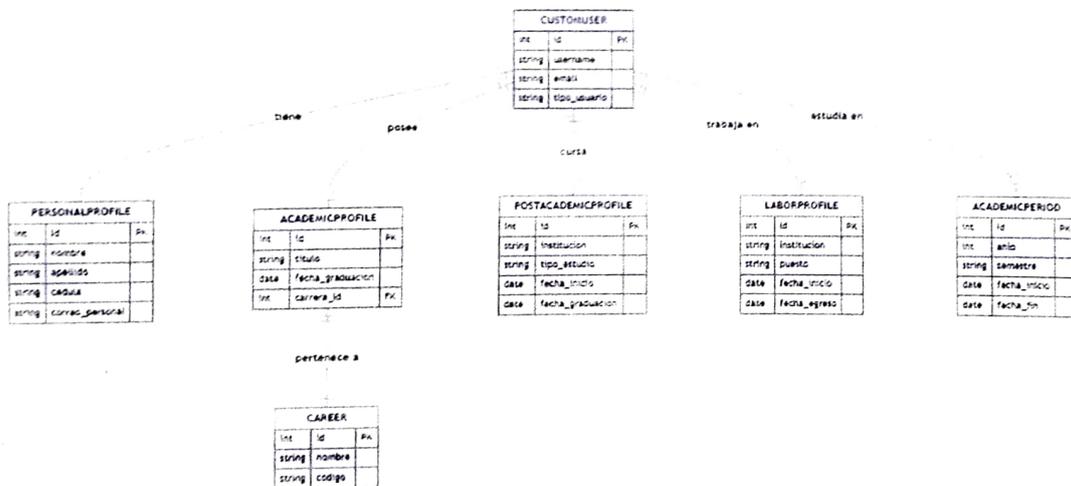
8.2. Resultados de la Aplicación de la Interfaz del Software

Descripción del Resultado:

El diseño del modelo de base de datos se realizó utilizando PostgreSQL. La estructura asegura la integridad referencial entre estudiantes, procesos de integración curricular y gestión de graduados, permitiendo la generación de reportes, la recopilación de información y la administración eficiente de los datos académicos y profesionales.

Diagrama Entidad-Relación (ER): Se generó un diagrama que refleja las relaciones entre las principales entidades del sistema. Las entidades principales incluyen CUSTOMUSER, que representa a los usuarios, y sus relaciones con PERSONALPROFILE, ACADEMICPROFILE, POSTACADEMICPROFILE, LABORPROFILE y ACADEMICPERIOD, permitiendo organizar datos personales, educativos y profesionales de manera estructurada.

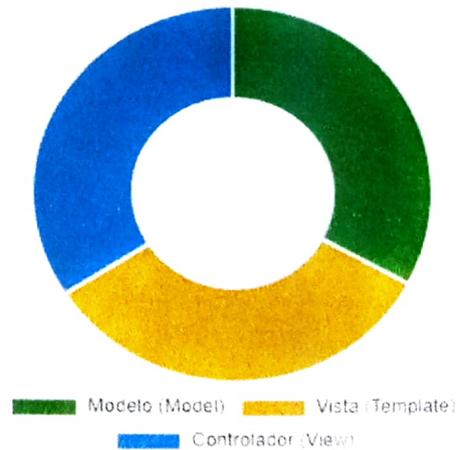
Figura 9: Diagrama Entidad-Relación del Sistema KatinaTec.



Nota: Diseñado en HTML con el uso de CSS y Bootstrap.

8.3. El sistema fue diseñado con el patrón Modelo-Vista-Controlador (MVC).

Figura 10: Arquitectura del Sistema basada en el patrón MVC.



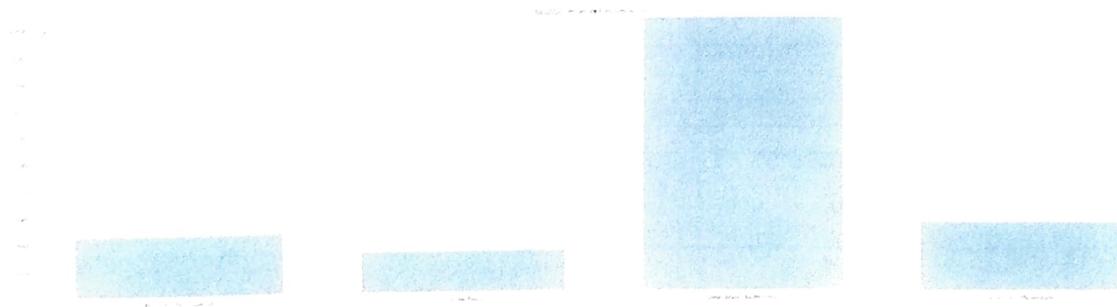
Nota: Diseñado en HTML con el uso de CSS y Bootstrap.

8.4. Resultados de la Codificación en el Lenguaje de Programación Python y Django

Descripción del Resultado:

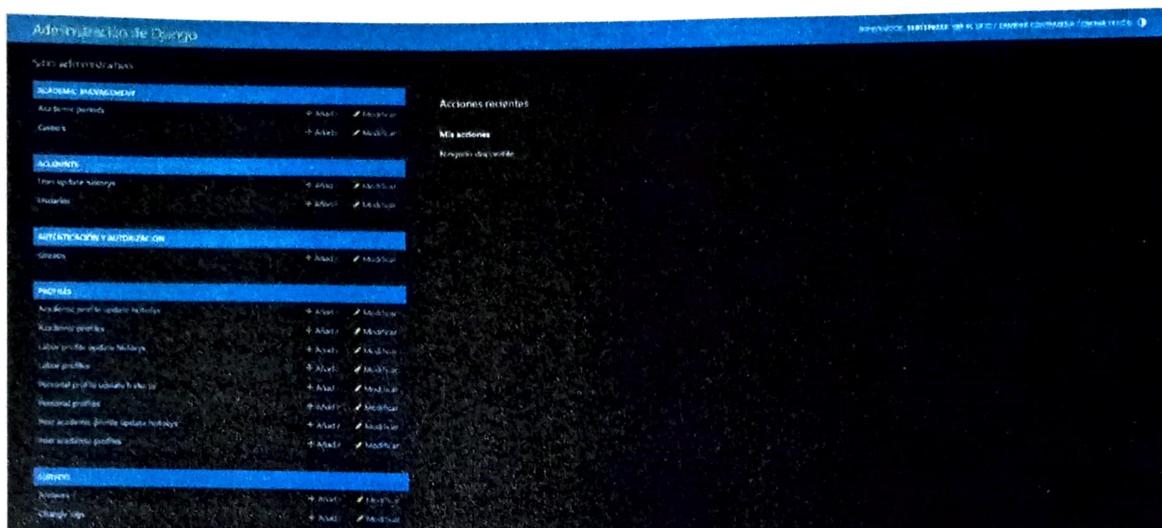
- ✓ Se desarrollaron las siguientes funcionalidades con Django:
- ✓ Autenticación y roles.
- ✓ Recopilación de información.
- ✓ Generación de reportes automatizados.
- ✓ Se realizaron pruebas de funcionalidad que validaron el ingreso de datos y la generación de reportes.

Figura 11: Gráfico de Tiempos de Respuesta y Rendimiento del Sistema.



Nota: Diseñado en HTML con el uso de CSS y Bootstrap.

Figura 12: Captura de Pantalla de la Interfaz Administrativa de Django.



Nota: Diseñado en html, con el uso de CSS y Bootstrap.

8.5. Análisis de las Tecnologías Seleccionadas

Tecnologías Evaluadas:

- ✓ Python 3.13.2/Django 5.1.3.
- ✓ PostgreSQL 17.4.
- ✓ Bootstrap 5.3.3.

Tabla 2: Tabla Comparativa de Ventajas y Desventajas de las Tecnologías Utilizadas.

Tecnología	Ventajas	Desventajas
Python/Django	Framework robusto con una arquitectura basada en MVC.	Curva de aprendizaje moderada para principiantes.
	Ofrece herramientas integradas como autenticación, administración y ORM (mapeo objeto-relacional).	Puede ser más lento que frameworks más livianos para aplicaciones extremadamente simples.
	Seguridad avanzada integrada, como protección	Las migraciones de bases de datos pueden ser

	contra CSRF, SQL Injection y XSS.	complicadas si no se manejan correctamente.
	Gran comunidad y documentación detallada.	Puede ser excesivo para proyectos pequeños debido a la cantidad de configuraciones requeridas.
	Compatible con un ecosistema rico en bibliotecas (NumPy, Pandas, etc.), lo que facilita la integración de funcionalidades avanzadas como inteligencia artificial.	
	Escalable y adaptable para proyectos de pequeño a gran tamaño.	
PostgreSQL	Base de datos relacional potente, ideal para proyectos con estructuras de datos complejas como encuestas.	Configuración inicial más compleja en comparación con otras bases de datos como SQLite o MySQL.
	Soporta tipos de datos avanzados, como JSON y arrays, que son útiles para manejar respuestas dinámicas.	Puede consumir más recursos que alternativas más simples, lo que no es ideal para proyectos pequeños o pruebas rápidas.
	Transacciones robustas y mecanismos de integridad referencial que aseguran la consistencia de los datos.	El uso avanzado requiere conocimientos más profundos de SQL, lo que podría aumentar el tiempo de desarrollo inicial.

	Ampliamente utilizado en producción, con soporte para consultas complejas y paralelización.	
	Gran soporte para integraciones con Django a través del ORM, lo que simplifica el desarrollo.	
Bootstrap	Framework de CSS ampliamente utilizado, facilita la creación de interfaces responsivas rápidamente.	Dependencia de estilos predefinidos, lo que puede limitar la personalización si no se ajustan los componentes.
	Documentación clara y extensa, con una comunidad activa que ofrece ejemplos y soporte.	Puede agregar peso adicional al proyecto si se incluyen componentes no utilizados, afectando el rendimiento.
	Compatible con navegadores modernos y con soporte para temas oscuros y claros.	Los proyectos pueden terminar viéndose "genéricos" si no se personalizan los estilos predeterminados.
	Ofrece una amplia variedad de componentes reutilizables (botones, formularios, tablas, etc.) que aceleran el desarrollo.	Requiere conocimientos básicos de CSS/HTML para aprovechar completamente sus capacidades de personalización.
	Fácil de integrar con Django y sistemas de plantillas, lo que	

	facilita el desarrollo rápido de la interfaz de usuario.	
--	--	--

8.6. Capturas de Pantalla del Sistema

A continuación, se presentan capturas de pantalla del sistema KatinaTec, mostrando diferentes interfaces y funcionalidades desarrolladas.

Ubicación: Relacionada con la funcionalidad de creación de cuentas, Sección 8.2.

Descripción: Captura de la pantalla donde los usuarios pueden registrarse en el sistema, ingresando sus datos personales y credenciales de acceso.

Figura 13: Interfaz de Registro de Usuarios.



Nota: Captura del sistema en navegador web.

Ubicación: Relacionada con la autenticación de usuarios, Sección 8.2.

Descripción: Muestra la pantalla de acceso al sistema donde los usuarios pueden iniciar sesión ingresando sus credenciales.

Figura 14: Interfaz de Inicio de Sesión del Sistema KatinaTec.

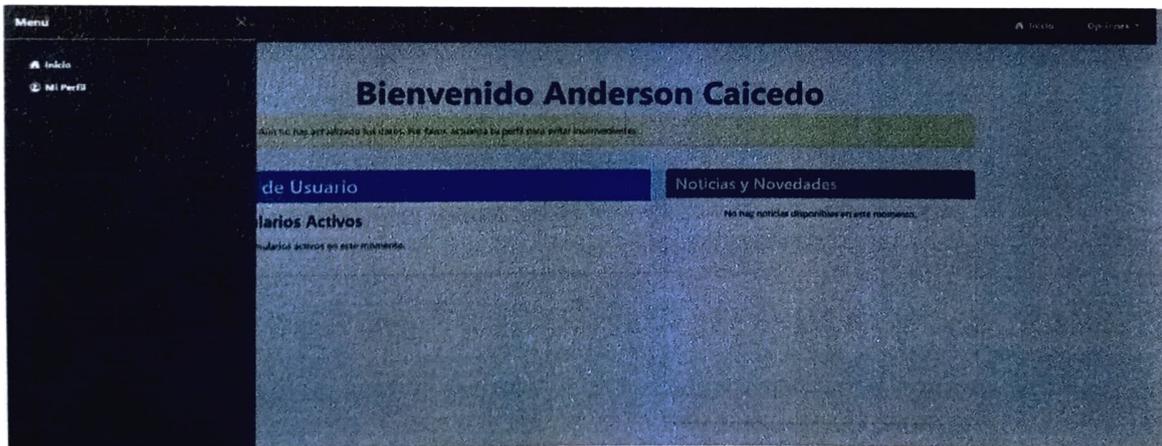


Nota: Captura del sistema en navegador web.

Ubicación: Relacionada con la navegación en el sistema, Sección 8.4.

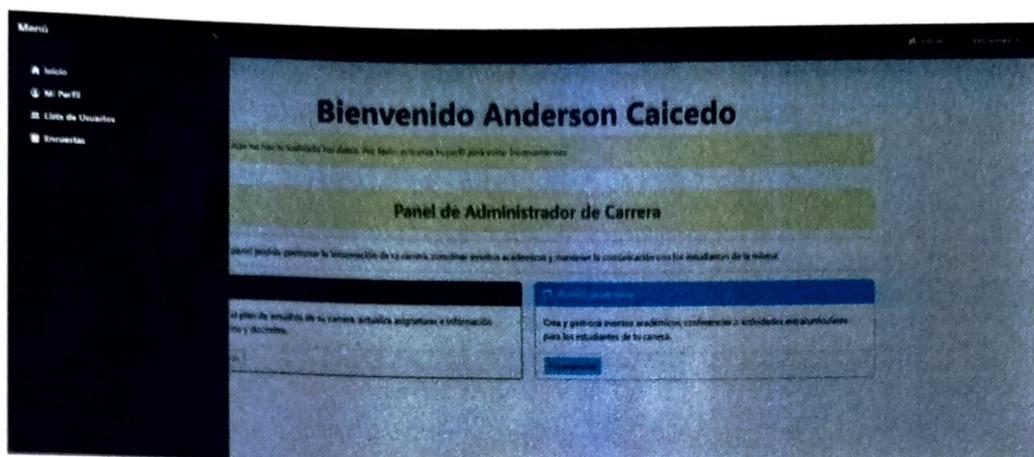
Descripción: Captura de la interfaz principal después de iniciar sesión, donde los usuarios pueden acceder a las diferentes funcionalidades del sistema, como consulta de reportes y gestión de procesos curriculares.

Figura 15: Panel del Usuario Estándar.



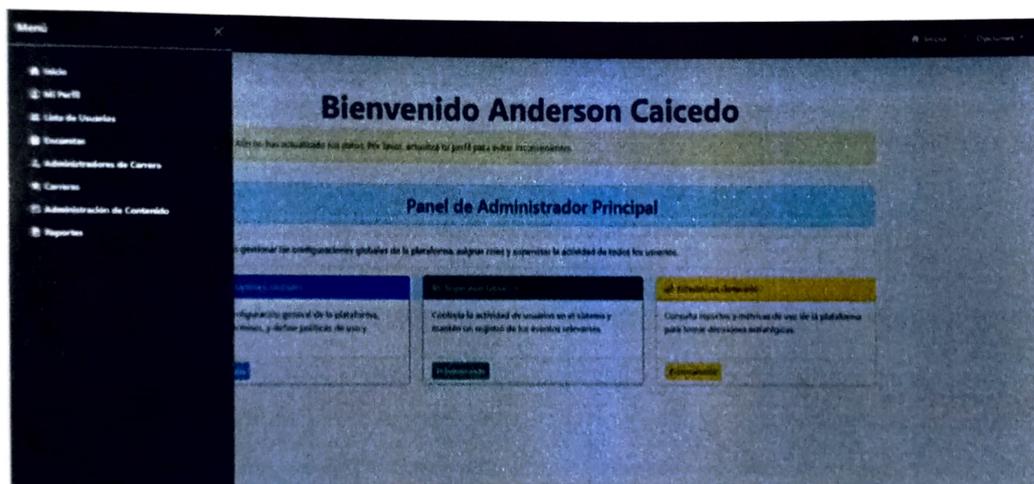
Nota: Captura del sistema en navegador web.

Figura 16: Panel del Administrador de Carrera.



Nota: Captura del sistema en navegador web.

Figura 17: Panel del Administrador Principal.

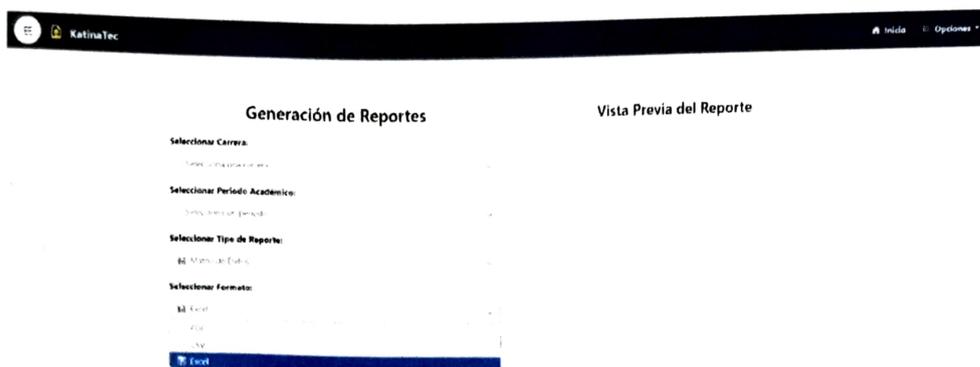


Nota: Captura del sistema en navegador web.

Ubicación: Relacionada con la funcionalidad de generación de reportes, Sección 8.4.

Descripción: Muestra la interfaz donde los usuarios pueden aplicar filtros para consultar reportes, exportarlos en diferentes formatos y visualizar los resultados de los datos procesados.

Figura 18: Generación y Consulta de Reportes en el Sistema.

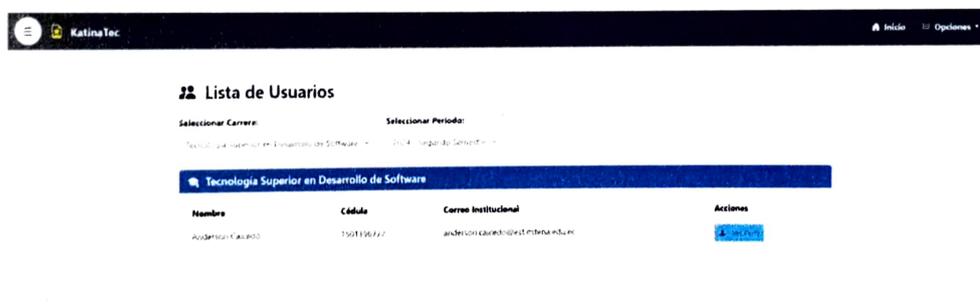


Nota: Captura del sistema en navegador web.

Ubicación: Relacionada con la funcionalidad de gestión de procesos académicos, Sección 8.4.

Descripción: Captura de la pantalla donde los usuarios pueden visualizar, filtrar y administrar los procesos curriculares dentro del sistema.

Figura 19: Gestión de Procesos Curriculares en KatinaTec.



Nota: Captura del sistema en navegador web.

I. CONCLUSIONES

- La adopción de Django, PostgreSQL y Bootstrap en el desarrollo de KatinaTec demostró la idoneidad de estas tecnologías para la gestión de procesos en la Unidad de Integración Curricular del Instituto Superior Tecnológico Tena. La elección de herramientas de código abierto y software libre aseguró escalabilidad, seguridad y cumplimiento con estándares tecnológicos modernos.
- La automatización y centralización de los procesos académicos y administrativos mediante KatinaTec optimizó la eficiencia operativa de la institución. La implementación de un modelo de base de datos bien estructurado permitió un acceso más ágil y seguro a la información, reduciendo errores humanos y mejorando la administración académica.
- Las pruebas realizadas, incluyendo ingreso de datos y generación de reportes, confirmaron que KatinaTec cumple con los requerimientos funcionales y no funcionales definidos. La capacidad del sistema para procesar información en tiempo real refuerza su utilidad para la toma de decisiones y su viabilidad para una implementación completa en el entorno académico.

J. RECOMENDACIONES

- Basado en los resultados obtenidos, se recomienda continuar con la expansión de KatinaTec, incorporando nuevas funcionalidades que mejoren la gestión académica y administrativa. Se debe considerar la integración con otras plataformas educativas y herramientas digitales para ampliar su alcance y optimizar la automatización de procesos.
- Para garantizar el máximo aprovechamiento del sistema, es fundamental implementar un plan de capacitación continua para el personal administrativo y los usuarios del sistema. Esto permitió minimizar errores, reducir la resistencia al cambio y mejorar la eficiencia operativa dentro de la institución.
- Asegurar la protección de los datos es una prioridad. Se recomienda fortalecer las medidas de seguridad del sistema mediante autenticación multifactor, cifrado de datos avanzado y auditorías periódicas. Además, se debe establecer un protocolo de evaluación periódica del sistema que incluya análisis de rendimiento y recopilación de feedback de los usuarios para ajustes y mejoras continuas.

K. BIBLIOGRAFÍA

11.1. Referencias

- Guevara Albán, C. S., Romero Fernández, A., & Cañizares Galarza, F. P. (2020). Sistema de información para el seguimiento a graduados en instituciones de educación superior de Ecuador. *Dilemas Contemporáneos: Educación, Política y Valores*.
<https://dilemascontemporaneoseducacionpoliticayvalores.com/index.php/dilemas/article/view/1632/1303>
- Hernández, L. Á., Acosta Miranda, M. L., Zavaleta Olea, E. R., & Domínguez Estudillo, D. (2022). Sistema de seguimiento de egresados como herramienta del desarrollo institucional. *Programación matemática y software*, 14(2), 19-28.
https://www.researchgate.net/publication/369431582_Sistema_de_Seguimiento_de_Egresados_como_Herramienta_del Desarrallo_Institucional
- Campaña Álvarez, M. A. (2015). Diseño de un sistema de gestión de seguimiento de inserción laboral a los profesionales graduados en la Universidad Técnica de Ambato y su incidencia en los indicadores de calidad (Tesis de maestría). Universidad Politécnica Salesiana, Guayaquil, Ecuador.
<https://dspace.ups.edu.ec/bitstream/123456789/9189/1/UPS-GT000874.pdf>
- Django Software Foundation. (2024). Documentación de Django (versión 5.1). Django Project.
<https://docs.djangoproject.com/es/5.1/>
- Python Software Foundation. (2024). Aprende Python. Python.org.
<https://es.python.org/aprende-python/>
- Bootstrap. (2025). Getting started (v5.3).
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

L. ANEXOS

REPÚBLICA DEL ECUADOR



INSTITUTO SUPERIOR
TECNOLÓGICO TENA
Tecnología, Innovación y Desarrollo

CARRERA:

TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE

PROYECTO:



MANUAL TÉCNICO DE KINATEC

AUTOR: Caicedo Rivadeneyra Anderson Guillermo

INTRODUCCIÓN

El presente manual técnico tiene como objetivo explicar cómo funciona parte del código del sistema KatinaTec, desarrollado para el Instituto Superior Tecnológico Tena. Este sistema está diseñado para gestionar la información académica de manera eficiente, incluyendo carreras, períodos académicos, jornadas y otros aspectos relacionados con los graduados del instituto.

Este manual no busca proporcionar una visión detallada de todo el proyecto, sino enfocarse en partes específicas del código para facilitar su comprensión. Está dirigido a desarrolladores, administradores del sistema y cualquier persona interesada en entender cómo se implementan ciertas funcionalidades en el proyecto.

Propósito del Proyecto

El propósito de este proyecto es desarrollar un Sistema Web que permita recopilar, almacenar, gestionar, manejar y generar reportes con la información sobre los graduados del Instituto Superior Tecnológico Tena. Esto mejorará la eficiencia en la administración de datos, reducirá errores y facilitará el seguimiento de los graduados.

Tecnologías Utilizadas

- ✓ Django: Versión 5.1.3. de Django utilizada.
- ✓ Python: Versión 3.13.2. de Python
- ✓ Base de Datos: Versión 17.4. de PostgreSQL
- ✓ Frontend: HTML, CSS, JavaScript.
- ✓ Bootstrap 5.3.3.

Alcance

El sistema permitirá:

- **Recopilación de datos:** Captura de información sobre los graduados.
- **Almacenamiento seguro:** Almacenamiento estructurado y seguro de los datos.
- **Gestión y manejo:** Actualización y consulta eficiente de la información.
- **Generación de reportes:** Creación de informes personalizados sobre los graduados.

Archivos Importantes

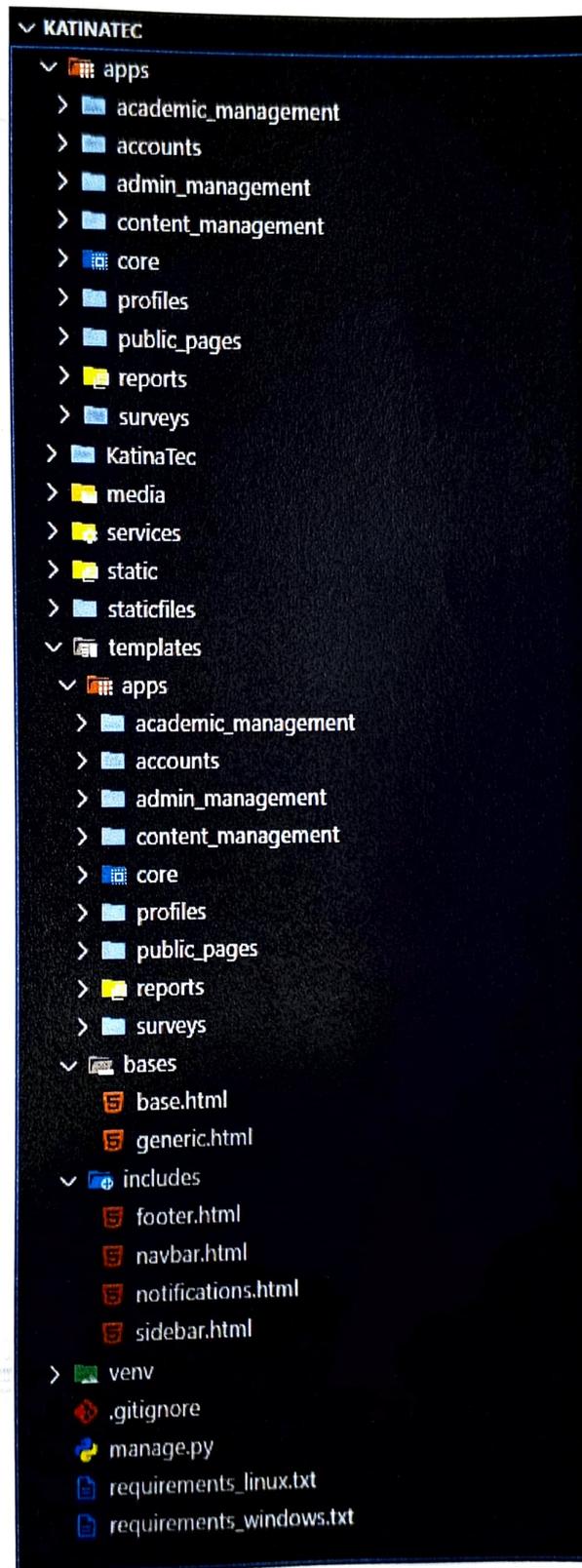
- **settings.py:** Configuración del proyecto (base de datos, middleware, aplicaciones instaladas, etc.).
- **urls.py:** Rutas principales del proyecto.
- **models.py:** Definición de los modelos de la base de datos (especialmente el modelo de graduados).
- **views.py:** Lógica de las vistas (gestión de graduados y generación de reportes).
- **templates/:** Plantillas HTML para la interfaz de usuario.
- **static/:** Archivos estáticos (CSS, JS, imágenes).

Requisitos de Hardware

- Procesador: Intel Core i5 o equivalente.
- Memoria RAM: 8 GB o superior.
- Almacenamiento: Al menos 10 GB de espacio libre.
- Conexión a Internet: Para la instalación de dependencias y actualizaciones.

ESTRUCTURA DEL PROYECTO

Descripción de la Estructura de Carpetas



Estructura de la Aplicación `academic_management/`

Este documento describe la organización de archivos y carpetas de la aplicación `academic_management/`, que forma parte del proyecto Django y se encarga de la gestión de la información académica, incluyendo carreras, períodos académicos y jornadas.

1. Ubicación dentro del Proyecto

La aplicación `academic_management/` se encuentra dentro de la carpeta `apps/`, que contiene todas las aplicaciones del proyecto. Cada aplicación es un módulo independiente con una funcionalidad específica.

2. Contenido de `academic_management/`

Dentro de la carpeta `academic_management/` encontramos los siguientes archivos y subcarpetas:

- **migrations/**: Contiene las migraciones de la base de datos, que son archivos generados por Django para aplicar cambios en los modelos a la base de datos.
- **__init__.py**: Indica que la carpeta es un paquete de Python.
- **admin.py**: Configuración del panel de administración de Django, donde se registran los modelos para su gestión desde la interfaz administrativa.
- **apps.py**: Configuración específica de la aplicación, incluyendo su nombre y método de carga.
- **forms.py**: Define los formularios utilizados en la aplicación para la entrada y validación de datos.
- **models.py**: Define los modelos de la base de datos, que representan las tablas y sus relaciones.
- **tests.py**: Contiene pruebas unitarias para la aplicación.
- **urls.py**: Define las rutas (URLs) de la aplicación y las asocia con sus vistas correspondientes.
- **views.py**: Contiene la lógica de las vistas, manejando solicitudes HTTP y generando respuestas, como renderizar plantillas o devolver datos en formato JSON.

3. Carpeta templates/ dentro de academic_management/

La aplicación cuenta con una carpeta templates/ que contiene todas las plantillas HTML necesarias para su funcionamiento. Estas plantillas están organizadas en subcarpetas para mantener una estructura clara y ordenada.

Ubicadas en templates/apps/academic_management/, estas plantillas incluyen:

- **manage_academic.html:** Utilizada para la gestión de información académica, como carreras, períodos académicos y jornadas. Proporciona una interfaz para agregar, editar y eliminar estos elementos.
- **users_by_career.html:** Muestra una lista de usuarios asociados a una carrera específica. Es útil para administradores que necesitan gestionar estudiantes o graduados por carrera.

4. Archivos adicionales en otras aplicaciones

Algunas aplicaciones dentro del proyecto pueden contener archivos adicionales según sus necesidades específicas. Por ejemplo:

- **accounts/:** Puede incluir archivos como:
 - **utils.py:** Funciones auxiliares y herramientas reutilizables.
 - **validators.py:** Validaciones personalizadas para formularios o modelos.
- **core/:** Puede incluir archivos como:
 - **middleware.py:** Middleware personalizado para la gestión de solicitudes y respuestas en el proyecto.

ESTRUCTURA DEL CÓDIGO

Modelos

Los modelos en Django representan la estructura de la base de datos. A continuación, se describe el modelo `ScheduleDetail`:

```
class ScheduleDetail(models.Model):
    jornada = models.ForeignKey(Schedule, on_delete=models.CASCADE, related_name='detalles')
    dias = models.ManyToManyField(DiaSemana, related_name='detalles_jornada')
    hora_inicio = models.TimeField()
    hora_fin = models.TimeField()

    class Meta:
        constraints = [
            models.UniqueConstraint(fields=['jornada', 'hora_inicio', 'hora_fin'], name='unique_schedule_time')
        ]
        verbose_name = "Detalle de Jornada"
        verbose_name_plural = "Detalles de Jornadas"

    def __str__(self):
        return f"{self.jornada.nombre} - {self.hora_inicio.strftime('%H:%M')} a {self.hora_fin.strftime('%H:%M')}"

    def clean(self):
        if self.hora_inicio >= self.hora_fin:
            raise ValidationError("La hora de inicio debe ser anterior a la hora de fin.")
        if not self.dias.exists():
            raise ValidationError("Debe seleccionar al menos un día de la semana.")
        if self.jornada and self.hora_inicio and self.hora_fin:
            if ScheduleDetail.objects.filter(
                jornada=self.jornada,
                hora_inicio=self.hora_inicio,
                hora_fin=self.hora_fin
            ).exclude(pk=self.pk).exists():
                raise ValidationError("Ya existe un horario con este rango de horas en la misma jornada.")
```

Explicación:

- **Campos:**
 - **jornada:** Relación con el modelo `Schedule` que representa la jornada académica.
 - **dias:** Relación muchos a muchos con el modelo `DiaSemana` para los días de la semana.
 - **hora_inicio y hora_fin:** Campos de tiempo para las horas de inicio y fin de la jornada.
- **Meta:**
 - **constraints:** Restricción de unicidad para evitar duplicados en el rango de horas dentro de la misma jornada.
 - **verbose_name:** Nombres legibles para la interfaz de administración.
- **Métodos:**
 - **__str__:** Representación legible del objeto.
 - **clean:** Validaciones personalizadas para asegurar la integridad de los datos.

Formularios

Los formularios en Django se utilizan para la entrada y validación de datos. A continuación, se describe el formulario `ScheduleDetailForm`:

```
class ScheduleDetailForm(forms.ModelForm):
    dias = forms.ModelMultipleChoiceField(
        queryset=DiaSemana.objects.none(),
        widget=forms.CheckboxSelectMultiple,
        label="Días de la Semana",
        required=True
    )

    class Meta:
        model = ScheduleDetail
        fields = ('jornada', 'dias', 'hora_inicio', 'hora_fin')
        labels = {
            'jornada': 'Jornada',
            'dias': 'Días de la Semana',
            'hora_inicio': 'Hora de Inicio',
            'hora_fin': 'Hora de Fin',
        }
        widgets = {
            'jornada': forms.Select(attrs={'class': 'form-control'}),
            'hora_inicio': forms.TimeInput(attrs={'type': 'time', 'class': 'form-control', 'placeholder': 'HH:MM'}),
            'hora_fin': forms.TimeInput(attrs={'type': 'time', 'class': 'form-control', 'placeholder': 'HH:MM'}),
        }

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['dias'].queryset = DiaSemana.objects.all()
```

Explicación:

- **Campos:**
 - dias: Campo de selección múltiple para los días de la semana, utilizando un widget de casillas de verificación.
- **Meta:**
 - model: Especifica el modelo asociado al formulario.
 - fields: Campos del modelo que se incluyen en el formulario.
 - labels: Etiquetas personalizadas para los campos.
 - widgets: Widgets personalizados para mejorar la interfaz de usuario.
- **Método `__init__`:**
 - Inicializa el queryset para el campo dias con todos los días de la semana disponibles.

Formularios

Los formularios en Django se utilizan para la entrada y validación de datos. A continuación, se describe el formulario **ScheduleDetailForm**:

```
class ScheduleDetailForm(forms.ModelForm):
    dias = forms.ModelMultipleChoiceField(
        queryset=DiaSemana.objects.all(),
        widget=forms.CheckboxSelectMultiple,
        label="Días de la Semana",
        required=True
    )

    class Meta:
        model = ScheduleDetail
        fields = ['jornada', 'dias', 'hora_inicio', 'hora_fin']
        labels = {
            'jornada': 'Jornada',
            'dias': 'Días de la Semana',
            'hora_inicio': 'Hora de Inicio',
            'hora_fin': 'Hora de Fin',
        }
        widgets = {
            'jornada': forms.Select(attrs={'class': 'form-control'}),
            'hora_inicio': forms.TimeInput(attrs={'type': 'time', 'class': 'form-control', 'placeholder': 'HH:MM'}),
            'hora_fin': forms.TimeInput(attrs={'type': 'time', 'class': 'form-control', 'placeholder': 'HH:MM'}),
        }

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['dias'].queryset = DiaSemana.objects.all()
```

Explicación:

- **Campos:**
 - dias: Campo de selección múltiple para los días de la semana, utilizando un widget de casillas de verificación.
- **Meta:**
 - model: Especifica el modelo asociado al formulario.
 - fields: Campos del modelo que se incluyen en el formulario.
 - labels: Etiquetas personalizadas para los campos.
 - widgets: Widgets personalizados para mejorar la interfaz de usuario.
- **Método `__init__`:**
 - Inicializa el queryset para el campo dias con todos los días de la semana disponibles.

Vistas

Las vistas en Django manejan la lógica de la aplicación. A continuación, se describe la vista `edit_schedule_details`:

```
@login_required
def edit_schedule_details(request, pk):
    try:
        schedule = get_object_or_404(Schedule, pk=pk)
        if request.method == 'POST':
            detail_ids = request.POST.getlist("detail_id[]")
            hora_inicio_list = request.POST.getlist("hora_inicio[]")
            hora_fin_list = request.POST.getlist("hora_fin[]")
            if len(hora_inicio_list) != len(hora_fin_list) or len(hora_inicio_list) != len(detail_ids):
                return JsonResponse({'success': False, 'message': 'Los datos enviados son inconsistentes.'},
                    status=400)
            for i, detail_id in enumerate(detail_ids):
                hora_inicio = hora_inicio_list[i]
                hora_fin = hora_fin_list[i]
                if not hora_inicio or not hora_fin:
                    return JsonResponse({'success': False, 'message': 'Las horas de inicio y fin son obligatorias.'},
                        status=400)
                if detail_id:
                    try:
                        detail = ScheduleDetail.objects.get(id=detail_id)
                    except ScheduleDetail.DoesNotExist:
                        return JsonResponse({'success': False, 'message': f'El detalle con ID {detail_id} no
                            existe.'}, status=404)
                    if ScheduleDetail.objects.filter(
                        jornada=schedule,
                        hora_inicio=hora_inicio,
                        hora_fin=hora_fin
                    ).exclude(pk=detail_id).exists():
                        return JsonResponse({'success': False, 'message': f'Ya existe un horario con esas horas para
                            la jornada {schedule.nombre}'}, status=400)
                    detail.hora_inicio = hora_inicio
                    detail.hora_fin = hora_fin
                    detail.save()
                    dias_seleccionados = request.POST.getlist(f'dias_{detail_id}[]')
                    detail.dias.set(dias_seleccionados)
                else:
                    if ScheduleDetail.objects.filter(
                        jornada=schedule,
                        hora_inicio=hora_inicio,
                        hora_fin=hora_fin
                    ).exists():
                        return JsonResponse({'success': False, 'message': f'Ya existe un horario con esas horas para
                            la jornada {schedule.nombre}'}, status=400)
                    new_detail = ScheduleDetail.objects.create(
                        jornada=schedule,
                        hora_inicio=hora_inicio,
                        hora_fin=hora_fin
                    )
                    dias_seleccionados = request.POST.getlist('dias_new[]')
                    new_detail.dias.set(dias_seleccionados)
            return JsonResponse({'success': True, 'message': 'Jornada y horarios actualizados correctamente.'})
    except Exception as e:
        print("Error al procesar la solicitud:", str(e))
        return JsonResponse({'success': False, 'message': str(e)}, status=500)
```

Explicación:

- **Método `edit_schedule_details`:**
 - Obtiene el objeto `Schedule` por su ID.
 - Procesa los datos enviados desde el formulario.
 - Valida y actualiza los detalles existentes o crea nuevos detalles.
 - Maneja errores y devuelve respuestas JSON apropiadas.

Modal para Editar Detalles de la Jornada

```
{% for schedule in schedules %}
<div class="modal fade" id="editScheduleDetailModal-{{ schedule.id }}" tabindex="-1" aria-hidden="true">
  <div class="modal-dialog modal-lg">
    <form id="editScheduleDetailForm-{{ schedule.id }}" method="POST" action="{% url
'academic_management:edit_schedule_details' schedule.id %}">
      {% csrf_token %}
      <div class="modal-content">
        <div class="modal-header bg-dark text-white">
          <h4 class="mb-0">✎ Editar Jornada: {{ schedule.nombre }}</h4>
          <button type="button" class="btn-close btn-close-white" data-bs-dismiss="modal"></button>
        </div>
        <div class="modal-body">
          <div class="form-group mb-4">
            <label class="form-label fw-bold">✎ Nombre de la Jornada</label>
            <input type="text" name="nombre_jornada" class="form-control" value="{{ schedule.nombre }}"
required>
          </div>
          <div id="horarios-container-{{ schedule.id }}">
            {% for detail in schedule.detalles.all %}
              <div class="horario-group mb-4 p-3 border rounded bg-light">
                <input type="hidden" name="detail_id[]" value="{{ detail.id }}">
                <div class="row g-3">
                  <div class="col-md-6">
                    <label>🕒 Hora de Inicio</label>
                    <input type="time" name="hora_inicio[]" class="form-control" value="{{
detail.hora_inicio }}" required>
                  </div>
                  <div class="col-md-6">
                    <label>🕒 Hora de Fin</label>
                    <input type="time" name="hora_fin[]" class="form-control" value="{{
detail.hora_fin }}" required>
                  </div>
                  <div class="col-md-12">
                    <label>📅 Dias de la Semana</label>
                    <div class="d-flex flex-wrap gap-3">
                      {% for checkbox in dias_disponibles %}
                        <div class="form-check">
                          <input type="checkbox" class="form-check-input" name="dias_{{
detail.id }}" value="{{ checkbox.id }}" {% if checkbox in detail.dias.all %}checked{% endif %}>
                          <label>{{ checkbox.nombre }}</label>
                        </div>
                      {% endfor %}
                    </div>
                  </div>
                </div>
                <div class="text-end mt-3">
                  <button type="button" class="btn btn-outline-danger btn-sm remove-horario">
                    <i class="bi bi-trash"></i> Eliminar Horario
                  </button>
                </div>
              </div>
            {% endfor %}
          </div>
          <div class="text-end mt-3">
            <button type="button" class="btn btn-outline-success add-horario" data-schedule-id="{{ schedule.id
}}">
              <i class="bi bi-plus lg me-2"></i> Agregar Otro Horario
            </button>
          </div>
          <div class="modal-footer">
            <button type="submit" class="btn btn-success">
              <i class="bi bi-save me-2"></i> Guardar Cambios
            </button>
          </div>
        </div>
      </div>
    </form>
  </div>
</div>
{% endfor %}
```

Explicación sobre las plantillas expuestas de ejemplo:

El sistema permite gestionar las jornadas académicas y sus detalles, como los horarios y los días de la semana asociados, a través de una serie de plantillas dinámicas e interactivas. Estas plantillas están diseñadas para mejorar la experiencia del usuario al integrar modales y tablas para facilitar la interacción.

1. Tabla de Jornadas (Schedule)

Función: Muestra una lista de todas las jornadas académicas registradas en el sistema.

Estructura:

- **Cabecera:** Muestra el nombre de la jornada y las acciones disponibles (editar, ver detalles y eliminar).
- **Cuerpo:** Itera sobre la lista de jornadas (schedules) y muestra cada una en una fila de la tabla.

Acciones:

- **Editar:** Abre un modal para editar los detalles de la jornada.
- **Ver detalles:** Abre un modal para mostrar los detalles completos de la jornada, incluyendo los horarios y días asociados.
- **Eliminar:** Permite eliminar una jornada específica, siempre y cuando no tenga detalles asociados.

Uso: Este template se utiliza en la vista principal de gestión académica (`manage_academic.html`) para listar y gestionar las jornadas.

2. Modal para Ver Detalles de la Jornada (viewScheduleDetailModal)

Función: Muestra los detalles completos de una jornada específica, incluyendo los horarios y los días de la semana asociados.

Estructura:

- **Cabecera:** Muestra el nombre de la jornada.
- **Cuerpo:**
 - Itera sobre los detalles de la jornada (`schedule.detalles.all`).
 - Muestra los días de la semana asociados (`detail.dias.all`).
 - Muestra el horario de inicio y fin (`detail.hora_inicio` y `detail.hora_fin`).
 - Si no hay detalles registrados, muestra un mensaje indicando que no hay información disponible.

Uso: Este modal se abre al hacer clic en el botón "Ver Detalles" en la tabla de jornadas.

3. Modal para Agregar Jornada (addScheduleModal)

Función: Permite agregar una nueva jornada académica.

Estructura:

- **Formulario:** Contiene un campo para ingresar el nombre de la jornada.
- **Botón de Guardar:** Envía el formulario para guardar la nueva jornada en la base de datos.

Uso: Este modal se abre al hacer clic en el botón "Agregar Jornada" en la sección de gestión académica.

4. Modal para Editar Detalles de la Jornada (editScheduleDetailModal)

Función: Permite editar los detalles de una jornada existente, incluyendo los horarios y los días de la semana asociados.

Estructura:

- **Cabecera:** Muestra el nombre de la jornada que se está editando.
- **Cuerpo:**
 - Nombre de la Jornada: Campo para editar el nombre de la jornada.
 - **Horarios:** Itera sobre los detalles de la jornada (schedule.detalles.all) y muestra los horarios y días asociados.
 - **Días de la Semana:** Permite seleccionar los días de la semana asociados a cada horario.
 - **Botón para Agregar Horario:** Permite agregar un nuevo horario a la jornada.
 - **Botón de Guardar Cambios:** Envía el formulario para actualizar los detalles de la jornada.

Uso: Este modal se abre al hacer clic en el botón "Editar Detalles" en la tabla de jornadas.

5. Template para Nuevo Horario (horario-template)

Función: Define la estructura de un nuevo horario que puede ser agregado dinámicamente en el modal de edición de detalles de la jornada.

Estructura:

- **Campos:**
 - Hora de inicio (hora_inicio).
 - Hora de fin (hora_fin).
 - Días de la semana (dias_new[]): Permite seleccionar los días de la semana asociados al horario.
 - Botón de **Eliminar**: Permite eliminar el horario si no es necesario.

Uso: Este template se utiliza como base para agregar nuevos horarios en el modal de edición de detalles de la jornada.

Resumen de la Interacción entre los Templates

- **Tabla de Jornadas:** Muestra la lista de jornadas y permite realizar acciones como editar, ver detalles y eliminar.
- **Modales:**
 - **Ver Detalles:** Muestra información detallada de una jornada.
 - **Agregar Jornada:** Permite crear una nueva jornada.
 - **Editar Detalles:** Permite modificar los horarios y días asociados a una jornada.
- **Template de Horario:** Define la estructura de un nuevo horario que puede ser agregado dinámicamente.

Importancia de los Templates en la Gestión de Schedule y ScheduleDetail

- **Interfaz Amigable:** Los templates proporcionan una interfaz intuitiva para que los usuarios puedan gestionar las jornadas y sus detalles sin complicaciones.
- **Dinamismo:** El uso de modales y templates dinámicos permite agregar, editar y eliminar información sin recargar la página, mejorando la experiencia del usuario.
- **Reutilización:** Los templates están diseñados para ser reutilizables, lo que facilita la mantenibilidad del código y la consistencia en la interfaz.

Plantilla base (base.html) e inclusiones (Includes)

1. base.html (Plantilla Base)

¿Qué hace?

- **Define la estructura común:** Proporciona el esqueleto básico de todas las páginas del sitio.
- **Carga recursos estáticos:** Incluye hojas de estilo (CSS), scripts (JavaScript) y el favicon.
- **Incluye fragmentos reutilizables:** Usa `{% include %}` para agregar componentes como la barra de navegación, el pie de página y la barra lateral.
- **Permite personalización:** Define bloques (`{% block %}`) que las plantillas hijas pueden sobrescribir para agregar contenido específico.

Estructura clave:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>{% block title %}KatinaTec - Instituto Superior Tecnológico Tena{% endblock %}</title>

  <!-- Favicon -->
  {% load static %}
  <link rel="icon" type="image/x-icon" href="{% static 'icons/favicon.ico' %}">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
  <!-- Bootstrap Icons -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">
  <!-- Estilos personalizados -->
  <link rel="stylesheet" href="{% static 'css/bases/styles.css' %}">
</head>
<body class="d-flex flex-column min-vh-100 bg-light">

  <!-- Navbar -->
  {% include 'includes/navbar.html' %}

  <!-- Contenedor principal -->
  <div class="wrapper d-flex">
    {% if user.is_authenticated %}
      {% include 'includes/notifications.html' %}
      {% include 'includes/sidebar.html' %}
    {% endif %}

    <!-- Contenido principal -->
    <main class="flex-grow 1 p 3">
      {% block content %}
        <!-- Contenido de cada página -->
        {% endblock %}
      </main>
    </div>

  <!-- Footer -->
  {% include 'includes/footer.html' %}

  <!-- Scripts -->
  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" async></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" defer></script>
</body>
</html>
```

2. navbar.html (Barra de Navegación)

¿Qué hace?

- Proporciona navegación: Contiene enlaces a las principales secciones del sitio, como "Inicio", "Mi Perfil", "Cerrar sesión", etc.
- Adapta su contenido: Muestra opciones diferentes dependiendo de si el usuario está autenticado o no.
- Es responsive: Incluye un botón para colapsar el menú en dispositivos móviles.

Estructura clave:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark shadow-lg">
  <div class="container-fluid d-flex align-items-center">
    <!-- Logo y enlace de inicio -->
    <a class="navbar-brand mx-auto fw-bold text-light" href="{% url 'core:home' %}">
      <i class="bi bi-file-earmark-person fs-4 me-2 text-warning"></i> KattinaTec
    </a>

    <!-- Menú desplegable para móviles -->
    <button class="navbar-toggler border-0 ms-auto" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav">
      <i class="bi bi-three-dots fs-3 text-white"></i>
    </button>

    <!-- Enlaces de navegación -->
    <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
      <ul class="navbar-nav align-items-center">
        <li class="nav-item">
          <a class="nav-link text-light fw-bold px-3" href="{% url 'core:home' %}">
            <i class="bi bi-house-door-fill me-1"></i> Inicio
          </a>
        </li>
        <!-- Hac enlaces según el estado de autenticación -->
      </ul>
    </div>
  </div>
</nav>
```

3. footer.html (Pie de Página)

¿Qué hace?

- **Muestra información de contacto:** Incluye la dirección, teléfono y correos electrónicos del instituto.
- **Proporciona enlaces a redes sociales:** Permite a los usuarios acceder a las redes sociales del instituto.
- **Muestra el año de copyright:** Indica el período de vigencia del sitio.

Estructura clave:

```
<footer class="bg-dark text-light shadow-lg py-4">
  <div class="container text-center">
    <p class="fw-bold mb-2">© 2018 - 2025 Instituto Superior Tecnológico Tena</p>
    <p class="text-secondary mb-2">Dirección: Km 1 ½ Vía (Tena - Archidona)</p>
    <p class="text-secondary mb-2">
      <i class="bi bi-telephone me-1"></i> Contacto: 062311709
    </p>
    <p>Enlaces a redes sociales</p>
    <div class="d-flex justify-content-center gap-3 mt-3">
      <a href="https://www.facebook.com/isttenaoficial" target="_blank" class="text-light fs-4">
        <i class="bi bi-facebook"></i>
      </a>
      <a href="https://twitter.com/isttenaoficial" target="_blank" class="text-light fs-4">
        <i class="bi bi-twitter-x"></i>
      </a>
    </div>
  </div>
</footer>
```

4. sidebar.html (Barra Lateral)

¿Qué hace?

- Proporciona un menú lateral: Muestra opciones de navegación adicionales para usuarios autenticados.
- Organiza enlaces por categorías: Agrupa enlaces como "Administración", "Gestión Administrativa", "Reportes", etc.
- Es colapsable: Permite a los usuarios expandir o contraer secciones del menú.

Estructura clave:

```
<div class="offcanvas offcanvas-start bg-dark text-white shadow-lg" tabindex="-1" id="sidebar">
  <div class="offcanvas-header border-bottom">
    <h5 class="offcanvas-title fw-bold text-uppercase" id="sidebarLabel">
      <i class="bi bi-list me-2"></i>Menú Principal
    </h5>
    <button type="button" class="btn-close btn-close-white" data-bs-dismiss="offcanvas"></button>
  </div>
  <div class="offcanvas-body p-4">
    <ul class="nav flex-column">
      <li class="nav-item">
        <a class="nav-link text-light fw-bold" href="{% url 'core:home' %}">
          <i class="bi bi-house-door-fill me-2 text-warning"></i>Inicio
        </a>
      </li>
      <li>
        <!-- Mas enlaces según el tipo de usuario -->
      </li>
    </ul>
  </div>
</div>
```

Resumen de Cómo Funciona Todo Junto

1. `base.html` es el esqueleto base que define la estructura común y carga los recursos necesarios.
2. `navbar.html`, `footer.html`, `sidebar.html` y `notifications.html` son fragmentos reutilizables que se incluyen en `base.html`.
3. Las plantillas hijas (como `manage_academic.html`) extienden `base.html` y sobrescriben los bloques `title` y `content` para agregar contenido específico.
4. Cuando un usuario visita una página, Django combina `base.html` con la plantilla hija y los fragmentos incluidos para generar el HTML final, que se muestra en el navegador.

Beneficios de Esta Estructura

1. Reutilización de código: Evita duplicar código en cada página.
2. Consistencia: Todas las páginas mantienen una apariencia uniforme.
3. Facilidad de mantenimiento: Los cambios en la estructura común solo se hacen en `base.html` o los fragmentos incluidos.
4. Modularidad: Cada componente (`navbar`, `footer`, etc.) se gestiona de forma independiente.

REPÚBLICA DEL ECUADOR



INSTITUTO SUPERIOR
TECNOLÓGICO TENA
Tecnología, Innovación y Desarrollo

CARRERA:

TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE

PROYECTO:



MANUAL DE USUARIO DE KATINATEC

(Exclusivo para Egresados)

AUTOR: Caicedo Rivadeneyra Anderson Guillermo

INTRODUCCIÓN

Este manual está diseñado para guiar a los usuarios en el uso eficiente y correcto del sistema **Katinatec**, desarrollado en el marco de la carrera de **Tecnología Superior en Desarrollo de Software**. El propósito de este manual es proporcionar una explicación de todas las funcionalidades del sistema, para que los usuarios puedan aprovechar al máximo sus herramientas y características.

A lo largo de este manual, se incluirán instrucciones paso a paso, capturas de pantalla y ejemplos prácticos que permitirán a los usuarios familiarizarse rápidamente con el sistema. Además, se describirán las diferentes secciones y características principales, asegurando que los usuarios puedan navegar y realizar tareas de manera eficiente.

Este documento está dirigido exclusivamente a los usuarios del sistema. Se asume que el usuario tiene un nivel básico de conocimientos en el uso de tecnologías de software, por lo que se evitará la jerga técnica innecesaria para facilitar la comprensión.

Acciones del Usuario (Egresados) en el Sistema KatinaTec

1. Acceso y Registro:

- **Iniciar sesión** con sus credenciales (correo electrónico y contraseña).
- **Registrarse** como nuevo usuario si es un egresado que aún no tiene cuenta en el sistema.

2. Gestión de Perfil:

- **Actualizar y completar** su información personal, académica y laboral.
- **Ver y modificar** sus estudios previos (colegio o bachillerato) y estudios posteriores.

3. Participación en Encuestas:

- **Responder encuestas activas** con diferentes tipos de preguntas (respuesta corta, párrafo, opción única, opción múltiple, etc.).
- **Ver las encuestas disponibles**, las fechas de vigencia y el tiempo restante para completarlas.
- **Guardar su progreso y enviar las encuestas completadas.**

4. Información y Noticias:

- **Leer noticias y novedades** destacadas del instituto para mantenerse informado.
- **Explorar los beneficios** de la plataforma y estar al tanto de eventos y actualizaciones importantes.

5. Conexión con el Instituto:

- **Seguir en redes sociales** (Facebook, Twitter, YouTube) para estar siempre actualizado con las publicaciones del instituto.
- **Contribuir al desarrollo** del instituto mediante su participación activa en la plataforma.

6. Notificaciones al Correo:

- **Recibir notificaciones** directamente en su correo electrónico para mantenerse informado sobre eventos importantes, actualizaciones o recordatorios enviados por el administrador.

7. Navegación y Soporte:

- **Navegar** entre las diferentes secciones del sistema (perfil, encuestas, noticias, etc.).
- **Cerrar sesión** de manera segura cuando termine su actividad en el sistema.
- **Solicitar ayuda o soporte** en caso de tener dudas o problemas técnicos relacionados con el sistema.

PÁGINA DE INICIO DE KATINATEC (PÁGINA PRINCIPAL)



La página de inicio de **KatinaTec** es la primera pantalla que los usuarios ven al acceder a la plataforma. Está diseñada para ofrecer una visión general del sistema y sus funcionalidades, además de guiar a los usuarios hacia el registro o inicio de sesión. También incluye enlaces a las redes sociales del **Instituto Superior Tecnológico Tena**, para que los usuarios puedan mantenerse conectados y actualizados.

Contenido de la Página de Inicio

1. Bienvenida y Descripción General

- **Título:** "Bienvenido a KatinaTec".
- **Descripción Breve:** KatinaTec es una herramienta diseñada para facilitar el seguimiento de los egresados del **Instituto Superior Tecnológico Tena**. Permite a los egresados gestionar su información académica y laboral, mientras que el instituto puede hacer un seguimiento detallado de su progreso.

2. Acceso al Sistema

- **Iniciar Sesión:** Los usuarios que ya tienen una cuenta pueden acceder a sus perfiles y utilizar todas las funcionalidades del sistema.
- **Registrarse:** Los nuevos usuarios (egresados) pueden registrarse para crear una cuenta y comenzar a utilizar la plataforma.

3. Funcionalidades Principales

- **Gestión de Egresados:** Los egresados pueden actualizar su perfil, información sobre estudios y experiencia laboral. Además, podrán mantenerse conectados con el instituto y con otros egresados.

- **Generación de Reportes:** Los usuarios pueden crear informes detallados sobre los egresados, sus empleos actuales y sus estudios posteriores.
 - **Acceso Seguro:** El sistema garantiza la seguridad de los datos personales mediante un sistema de autenticación y el uso de correos electrónicos verificados.
4. **¿Qué es KatinaTec?**
- KatinaTec es una plataforma creada para dar seguimiento a los egresados del Instituto Superior Tecnológico Tena. Su objetivo es facilitar la gestión de información académica y laboral tanto para los egresados como para el personal administrativo del instituto.
5. **Beneficios de Usar KatinaTec**
- **Mantenerse en contacto:** Los egresados pueden conectarse con otros egresados y con el instituto, lo que facilita la creación de una red de apoyo y colaboración.
 - **Mejorar oportunidades laborales:** La plataforma ofrece a los egresados la posibilidad de acceder y valorar oportunidades laborales que se ajusten a su perfil.
 - **Contribuir al desarrollo:** Los egresados pueden ayudar al instituto participando activamente en la plataforma, contribuyendo con su experiencia y conocimientos.
 - **Generar reportes:** La creación de informes es sencilla, permitiendo realizar un seguimiento eficiente de los egresados.
6. **Síguenos en Redes Sociales**
- **Facebook:** Accede a las últimas publicaciones de la página oficial del Instituto Superior Tecnológico Tena.
 - **Twitter:** Consulta los últimos tweets del instituto.
 - **YouTube:** Visita el canal oficial de YouTube del instituto para ver contenido actualizado y eventos grabados.

PÁGINA DE REGISTRO DE USUARIO:

Registro de Usuario

Tipo de Documento:

Número de Documento:

Correo Institucional:

Primer Nombre:

Segundo Nombre:

Primer Apellido:

Segundo Apellido:

Contraseña:

Confirmar Contraseña:

Crear Cuenta

[¿Ya tienes una cuenta? Inicia sesión](#)

Campos de Registro:

1. Escudo Espiritual:

El usuario debe ingresar su escudo espiritual, que es un identificador personal o simbólico (si aplica a tu sistema). Este campo es opcional o requerido, según se configure.

2. Tipo de Documento:

El usuario debe seleccionar el tipo de documento que utilizará para registrarse. Puede ser cédula de identidad, pasaporte, etc. El usuario seleccionará la opción que corresponda.

3. Cédula:

Este campo es obligatorio. El usuario debe ingresar su número de cédula, que servirá como su nombre de usuario único para acceder al sistema. Es importante ingresar el número correctamente.

4. Correo Institucional:

El usuario debe ingresar su correo electrónico institucional. Este correo será utilizado para notificaciones importantes y recuperación de contraseña, entre otros.

5. Primer Nombre:

El usuario debe ingresar su primer nombre.

6. Segundo Nombre:

Si aplica, el usuario puede ingresar su segundo nombre. Este campo es opcional.

7. Primer Apellido:

El usuario debe ingresar su primer apellido.

8. **Segundo Apellido:**

El usuario puede ingresar su segundo apellido si lo tiene. Este campo es opcional.

9. **Contraseña:**

El usuario debe crear una contraseña segura para acceder a su cuenta en el sistema. Esta contraseña debe contener al menos 8 caracteres e incluir una combinación de letras, números y caracteres especiales.

10. **Repetir Contraseña:**

El usuario debe ingresar nuevamente la misma contraseña para confirmarla. Esta medida es para evitar errores tipográficos y asegurar que ambas contraseñas coincidan.

Botón de Registro:

- Una vez que todos los campos sean completados correctamente, el usuario puede hacer clic en el **botón de registro** para enviar los datos. Si todo está correcto, el sistema creará la cuenta y el usuario podrá acceder a la plataforma.

Mensajes de Error:

- Si algún campo obligatorio no es completado, el sistema mostrará un mensaje de error, por ejemplo:
"El campo 'Cédula' es obligatorio."
- Si las contraseñas no coinciden, se mostrará un mensaje como:
"Las contraseñas no coinciden. Intente nuevamente."
- Si el correo electrónico ya está registrado, aparecerá el mensaje:
"Este correo electrónico ya está en uso. Por favor, ingrese uno diferente."

Redirección al Inicio de Sesión:

- Una vez registrado, el usuario será redirigido a la página de inicio de sesión, donde podrá ingresar sus datos (cédula y contraseña) para acceder al sistema.

PÁGINA DE INICIO DE SESIÓN:

KatinaTec

Inicio | Iniciar sesión | Registrarse

Iniciar Sesión

Cédula

Contraseña

Mantener sesión iniciada

Iniciar Sesión

[¿No tienes una cuenta? Regístrate.](#)

[¿Olvidaste tu contraseña?](#)

Elementos de la Página:

1. Campo de Nombre de Usuario (Cédula):

El usuario debe ingresar su cédula, que funciona como nombre de usuario único.

2. Campo de Contraseña:

El usuario debe ingresar su contraseña asociada a su cuenta para confirmar su identidad.

3. Botón de Inicio de Sesión:

Después de ingresar los datos, el usuario hace clic en este botón para proceder con la autenticación.

4. Mensajes de Error:

- **Error de autenticación:** Si el nombre de usuario (cédula) o la contraseña son incorrectos, el sistema muestra un mensaje de error como: *"Cédula o contraseña incorrectos. Por favor, intente nuevamente."*

5. Opción de "Mantener sesión iniciada" (opcional):

Si el usuario marca esta opción, su sesión se mantendrá activa durante un periodo determinado, incluso si cierra el navegador. Esto evita tener que iniciar sesión cada vez que acceda al sitio.

6. Redirección tras inicio exitoso:

Si los datos ingresados son correctos, el usuario será redirigido automáticamente a la página principal del sistema, donde podrá acceder a las funcionalidades disponibles.

PÁGINA DE RECUPERACIÓN DE CONTRASEÑA:



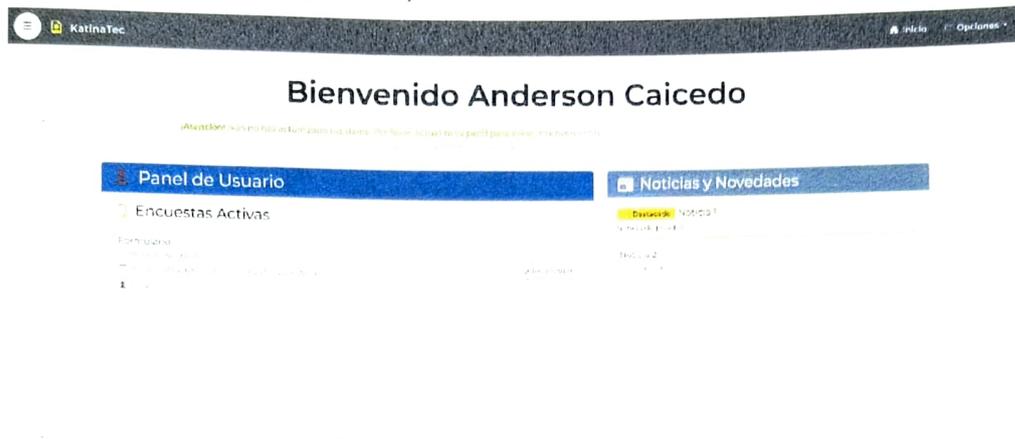
Propósito:

La página de recuperación de contraseña permite a los usuarios restablecer su contraseña si la han olvidado. El sistema solicita el correo electrónico registrado para enviar instrucciones detalladas sobre cómo crear una nueva contraseña y acceder nuevamente a su cuenta.

Proceso de Recuperación de Contraseña:

1. **Campo de Correo Electrónico:** El usuario debe ingresar su **correo electrónico institucional** en el campo proporcionado. Este correo debe ser el que el usuario registró previamente en el sistema. El sistema enviará un enlace a este correo para iniciar el proceso de recuperación de contraseña.
2. **Botón de Enviar Enlace de Recuperación:** Después de ingresar el correo electrónico, el usuario debe hacer clic en el **botón de "Enviar enlace de recuperación"**. Al hacer esto, el sistema verificará si el correo electrónico ingresado está registrado en la base de datos.
3. **Mensaje de Confirmación:**
 - o **Si el correo es válido:** Si el correo electrónico está registrado, el sistema enviará un mensaje al usuario confirmando que el enlace de recuperación ha sido enviado a su correo. El mensaje puede ser algo como:
"Si el correo electrónico ingresado es correcto, recibirás un enlace para restablecer tu contraseña en unos minutos."

PÁGINA DE INICIO (HOME):



La página de inicio es el primer punto de contacto del usuario con el sistema después de iniciar sesión. En esta sección, el usuario encontrará información relevante y accesos rápidos a las principales funcionalidades del sistema.

Elementos de la Página:

1. Mensaje de Bienvenida

Ubicado en la parte superior de la página, este mensaje está personalizado con el nombre del usuario. Su propósito es confirmar que ha iniciado sesión correctamente y ofrecer una experiencia más amigable.

2. Recordatorio de Actualización de Datos

Si el usuario no ha actualizado su perfil recientemente, aparecerá un mensaje de atención recordándole la importancia de completar o actualizar su información personal. Mantener los datos actualizados es fundamental para garantizar un correcto funcionamiento del sistema y evitar inconvenientes.

3. Panel de Usuario

Esta sección muestra información clave sobre las encuestas activas y las noticias destacadas.

• Encuestas Activas:

- Nombre del Formulario: Indica el título de la encuesta.
- Fechas de Disponibilidad: Especifica el periodo durante el cual la encuesta estará activa.
- Creador de la Encuesta: Muestra el nombre de la persona o entidad que la creó.

- El usuario puede hacer clic en la encuesta para acceder a ella y completarla.
- **Noticias y Novedades:**
 - Noticias Destacadas: Contiene información relevante publicada por el administrador del sistema.
 - Los anuncios pueden incluir detalles sobre nuevas funcionalidades, mantenimientos programados o avisos importantes.
 - Se presentan en formato de lista, con un título breve y una descripción.

PÁGINA DE INTENTO DE ENCUESTA

Formulario

Avance

Tiempo restante: 28.04

Pregunta 1

Pregunta 2

Pregunta 3

Pregunta 4

Pregunta 5

Esta página permite al usuario responder a una encuesta activa. Muestra las preguntas que deben ser contestadas y proporciona espacios para que el usuario ingrese sus respuestas, según el tipo de pregunta. A continuación, se describen los elementos principales y los tipos de preguntas disponibles:

Información de la Encuesta:

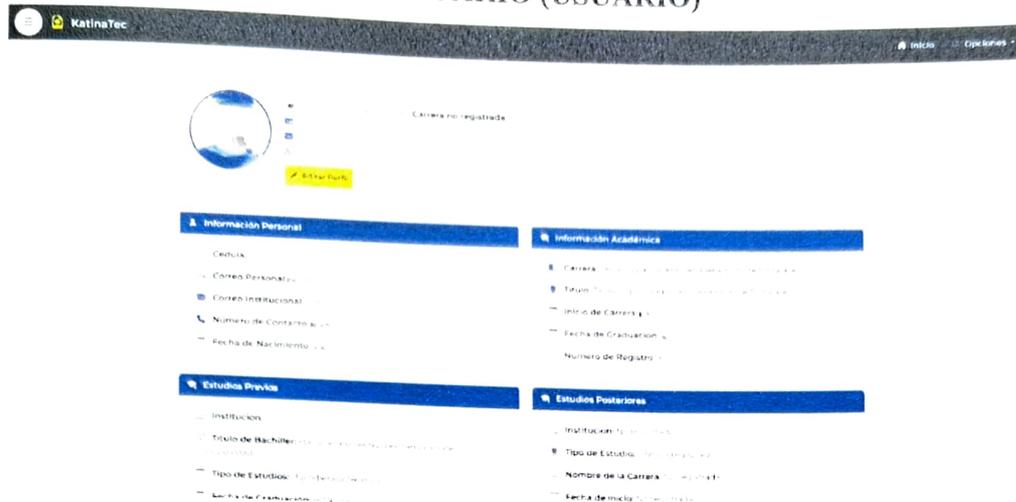
- **Nombre del Formulario:** Muestra el nombre de la encuesta (por ejemplo, "Formulario de prueba").
- **Creado por:** Indica quién creó la encuesta (por ejemplo, "tic.tsd (Administrador principal)").
- **Ámbito:** Especifica a quién está dirigida la encuesta (por ejemplo, "Todas las carreras").
- **Tiempo Restante:** Muestra el tiempo restante para completar la encuesta (por ejemplo, "28.04").

Tipos de Preguntas:

- **Respuesta Corta:** El usuario debe escribir una respuesta breve en un campo de texto.
- **Párrafo:** El usuario debe escribir una respuesta más larga en un campo de texto más grande.
- **Opción Única:** El usuario debe seleccionar solo una opción de una lista proporcionada (por ejemplo, botones de opción).
- **Opción Múltiple:** El usuario puede seleccionar varias opciones de una lista (por ejemplo, casillas de verificación).

- **Opción Múltiple con Opción Personalizada:** El usuario puede seleccionar varias opciones de una lista y también tiene la opción de agregar una respuesta personalizada.
- **Casilla de Verificación:** El usuario debe marcar una o más casillas para indicar su respuesta.

PÁGINA DE PERFIL DEL USUARIO (USUARIO)



La página de perfil del usuario de tipo U está organizada en tarjetas que presentan la información más relevante del usuario. A continuación, se describe el contenido de cada una:

1. Información Personal

Muestra los datos básicos del usuario, incluyendo:

- Cédula
- Correos electrónicos (personal e institucional)
- Número de contacto
- Fecha de nacimiento

2. Estudios Previos

Esta tarjeta contiene detalles sobre los estudios realizados a nivel de colegio o bachillerato:

- Institución educativa
- Título obtenido
- Tipo de estudios
- Fecha de graduación

3. Información Académica

Presenta información sobre la carrera que el usuario está cursando o ha cursado:

- Nombre de la carrera
- Título obtenido
- Fechas de inicio y graduación
- Número de registro

4. Información Laboral

Incluye detalles sobre la experiencia laboral del usuario:

- Institución
- Ubicación
- Correo laboral
- Fechas de inicio y egreso
- Descripción de la experiencia laboral
- Si no hay información registrada, se indica como "**No registrada**".

5. Estudios Posteriores

Contiene información sobre estudios adicionales realizados por el usuario:

- Institución educativa
- Tipo de estudio
- Nombre de la carrera
- Fechas de inicio y graduación
- Si no hay información registrada, se indica como "**No registrada**".

PÁGINA DE ACTUALIZACIÓN DE PERFILES (USUARIO)

1. Formulario de información personal de los graduados

The screenshot shows the 'Actualización de Perfiles' page with the 'Información Personal' tab selected. The form includes the following fields:

- Número de cédula (no editable)
- ¿Cuál es su número de contacto?
- ¿Cuál es tu correo personal?
- Correo institucional (no editable)
- País de Nacionalidad
- País de Residencia
- ¿En qué provincia reside?
- ¿Cuál es su fecha de nacimiento?

Este formulario permite a los graduados ingresar y actualizar su información personal, incluyendo su número de cédula, correos electrónicos, nombre, apellido, contacto, foto de perfil, ubicación, género, fecha de nacimiento, nacionalidad y residencia. Algunos campos, como la cédula y el correo institucional, son de solo lectura. Además, se validan los formatos de imagen y correos electrónicos permitidos.

2. Formulario de estudios previos (Bachillerato)

The screenshot shows the 'Actualización de Perfiles' page with the 'Información de Estudios Previos' tab selected. The form includes the following fields:

- Nombre del Colegio
- Título de Bachiller
- Fecha de Graduación
- Tipo de Colegio
- Tipo de Estudios

Este formulario recopila información sobre la educación secundaria del usuario, como el nombre del colegio, el título de bachiller, la fecha de graduación, el tipo de estudios y el tipo de colegio. Incluye validaciones para evitar fechas de graduación posteriores a la fecha actual.

3. Formulario para el perfil académico del instituto

The screenshot shows the 'Actualización de Perfiles' (Profile Update) interface. The main heading is 'Actualización de Perfiles' with a subtitle 'Permite actualizar los datos de tu perfil académico.' and a 'Volver al Perfil' button. A sidebar on the left contains navigation options: Perfil Personal, Estudios, Perfil Académico (selected), Estudios Posteriores, and Perfil Laboral. The main content area is titled 'Información Académica' and contains the following fields:

- Carrera Universitaria:** A dropdown menu with 'Ingeniería en Sistemas de Información' selected.
- Código de Carrera:** A text input field with '00000000000000000000'.
- Título Académico:** A dropdown menu with 'Ingeniería en Sistemas de Información' selected.
- Período Académico:** A dropdown menu with '2019-2020' selected.
- Fecha de Inicio:** A date picker with '2019-09-01' selected.
- Fecha de Graduación:** A date picker with '2020-06-30' selected.
- Número de Registro:** A text input field with '00000000000000000000'.
- Fecha de Registro de Estudio:** A date picker with '2019-09-01' selected.
- Asignaturas Repetidas:** A text input field with '0'.
- (Pérdida Graduada?):** A dropdown menu with 'No' selected.

A 'Guardar Cambios' button is located at the bottom right of the form.

Permite registrar y gestionar información sobre la trayectoria académica del usuario en el instituto, incluyendo el título obtenido, código de carrera, período académico, fechas de inicio y graduación, número de registro profesional y otros datos relevantes. Se validan las fechas para asegurar coherencia en la información ingresada.

4. Formulario para el perfil de estudios posteriores

The screenshot shows the 'Estudios Posteriores' (Posterior Studies) form. The sidebar on the left contains navigation options: Perfil Personal, Estudios (selected), Perfil Académico, Estudios Posteriores, and Perfil Laboral. The main content area is titled 'Estudios Posteriores' and contains the following fields:

- Institución:** A text input field with 'Universidad de los Andes'.
- Nombre de la Carrera:** A text input field with 'Ingeniería en Sistemas de Información'.
- Tipo de Estudio:** A dropdown menu with 'Ingeniería' selected.
- País de Estudios Posteriores:** A dropdown menu with 'Colombia' selected.
- Fecha de Inicio:** A date picker with '2019-09-01' selected.
- Fecha de Graduación:** A date picker with '2020-06-30' selected.
- Cantidad de Duración:** A text input field with '0'.
- Tipo de Duración:** A dropdown menu with 'Años' selected.
- Cursos Realizados:** A large text area for listing completed courses.

Este formulario permite registrar estudios adicionales realizados por el usuario después de su graduación en el instituto. Incluye información sobre la institución, la carrera, fechas de inicio y graduación, cursos realizados y duración de los estudios. Se realizan validaciones para asegurar que las fechas sean coherentes.

5. Formulario para el perfil laboral

The screenshot shows a web form titled 'Perfil Laboral'. On the left, there is a navigation menu with options like 'Inicio', 'Inicio de Sesión', 'Perfil Laboral', 'Perfil Académico', 'Perfil Personal', and 'Perfil Profesional'. The main form area has a header 'Perfil Laboral' and a sub-header 'Datos Laborales'. The form is organized into two columns. The left column has fields for 'Institución Laboral', 'Datos Laborales' (a large text area), 'Correo Laboral', and 'Fecha de Inicio'. The right column has fields for 'Ubicación Laboral', 'Correo de la Institución', 'Tipo de Institución', and 'Fecha de Egreso'. At the bottom right, there is a green button labeled 'Guardar Cambios'.

Este formulario captura información sobre la experiencia laboral del usuario, incluyendo el nombre de la empresa, cargo ocupado, tiempo de trabajo, tipo de contrato y ubicación. También permite registrar si el usuario ha trabajado en el área de su formación académica. Se incluyen validaciones para fechas y otros datos clave.

PREGUNTAS FRECUENTES (FAQ)

1. ¿Cómo me registro en KatinaTec?

- Si eres un egresado del Instituto Superior Tecnológico Tena, puedes registrarte haciendo clic en "Registrarse" en la página de inicio. Solo necesitarás tu correo electrónico y algunos datos personales. Después de completar el formulario, recibirás un correo de confirmación para activar tu cuenta.

2. ¿Cómo accedo a mi perfil?

- Para acceder a tu perfil, simplemente haz clic en "Iniciar sesión" en la página principal y utiliza tus credenciales (correo electrónico y contraseña). Una vez dentro, podrás ver y actualizar tu perfil desde el panel principal.

3. ¿Cómo actualizo mi información personal?

- Para actualizar tu información personal, ve a la sección "Gestión de Perfil". Aquí podrás modificar tus datos personales, académicos y laborales. Asegúrate de guardar los cambios al finalizar.

4. ¿Cómo participo en encuestas?

- Las encuestas activas estarán disponibles en tu panel de usuario. Solo tienes que seleccionarlás y responder las preguntas correspondientes. Recuerda guardar tu progreso y enviarla antes de la fecha de vencimiento.

5. ¿Qué debo hacer si olvido mi contraseña?

- Si olvidas tu contraseña, haz clic en "Olvidé mi contraseña" en la página de inicio de sesión. Luego, sigue las instrucciones que recibirás por correo electrónico para restablecerla.

6. ¿Puedo recibir notificaciones por correo electrónico?

- Sí, el sistema envía notificaciones importantes a tu correo, como recordatorios de encuestas, actualizaciones o noticias relevantes. Asegúrate de tener tu dirección de correo electrónico correctamente registrada en tu perfil.

7. ¿Cómo puedo seguir al Instituto en redes sociales?

- Puedes seguir al Instituto en sus redes sociales (Facebook, Twitter, YouTube) haciendo clic en los enlaces disponibles en la página de inicio del sistema.

8. ¿Cómo cierro sesión del sistema de manera segura?

- Para cerrar sesión de forma segura, haz clic en el botón de "Cerrar sesión" que se encuentra en la parte superior de la página. Esto garantizará que tu cuenta quede protegida.

SOPORTE TÉCNICO

Información de contacto para soporte técnico:

- Si experimentas problemas técnicos o tienes alguna duda sobre el uso del sistema, puedes contactar al equipo de soporte a través de los siguientes canales:
 - **Correo electrónico:** garow.soporte@gmail.com
 - **Teléfono:** +593 969 473 847
 - **Formulario de contacto:** En el sitio web de KatinaTec, puedes acceder a un formulario de contacto en el pie de la página en la sección "*Accede a nuestra página de soporte y ayuda*" para enviar tus dudas o problemas directamente al equipo de soporte.

Procedimiento para obtener ayuda o resolver problemas:

1. **Identifica el problema:** Asegúrate de que el problema sea técnico (como no poder acceder a tu cuenta o un error al cargar la página) o relacionado con el uso del sistema.
2. **Consulta las preguntas frecuentes (FAQ):** En caso de problemas comunes, revisa las preguntas frecuentes en el manual para ver si la solución ya está disponible.
3. **Contacta al soporte técnico:** Si el problema persiste o necesitas asistencia personalizada, contacta al soporte técnico utilizando los canales mencionados. Proporciona la mayor cantidad de información posible, como el error exacto que estás viendo, el navegador que estás utilizando y cualquier otra información relevante.
4. **Espera respuesta:** El equipo de soporte responderá a tu solicitud dentro de las 24 horas hábiles para ayudarte a resolver el problema.

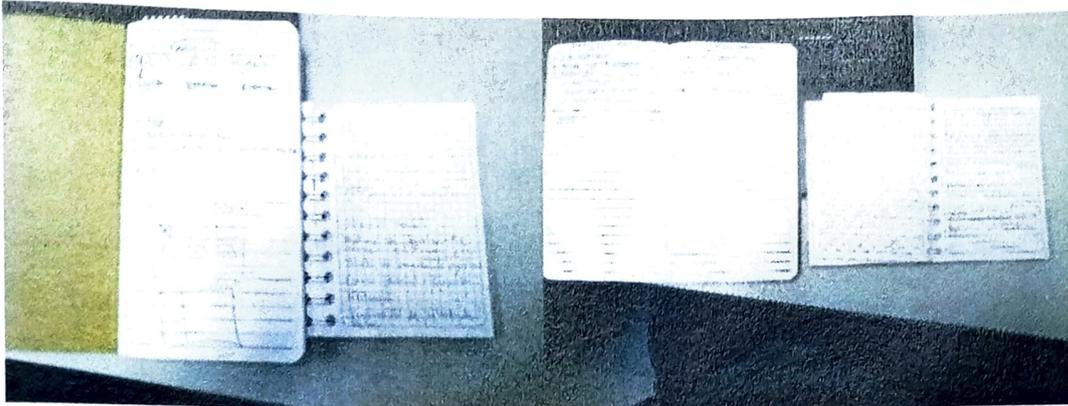
CONCLUSIÓN

Resumen de lo que el usuario ha aprendido:

- En este manual, has aprendido cómo acceder al sistema **KatinaTec**, registrar tu cuenta como egresado, actualizar tu perfil con tus datos personales y académicos, participar en encuestas activas, recibir notificaciones importantes al correo electrónico y mantenerte informado sobre las últimas noticias del Instituto Superior Tecnológico Tena.
- También has aprendido a navegar por el sistema de manera eficiente, gestionar tus encuestas y mantener tu cuenta segura con el cierre de sesión adecuado.

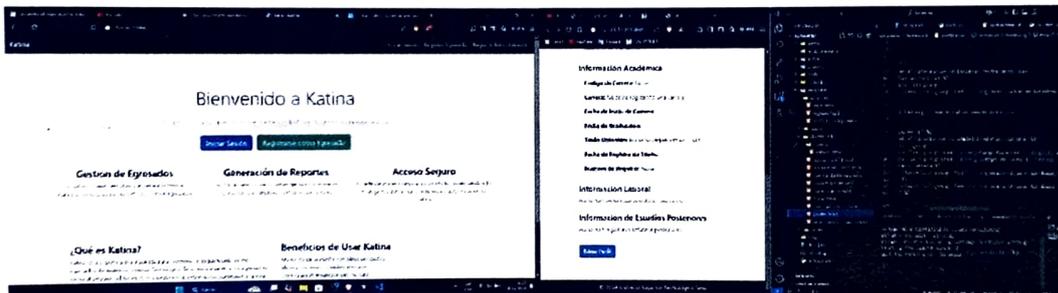
IMÁGENES DE ACTIVIDADES REALIZADAS

Anexo 1: Anotaciones sobre los requerimientos.



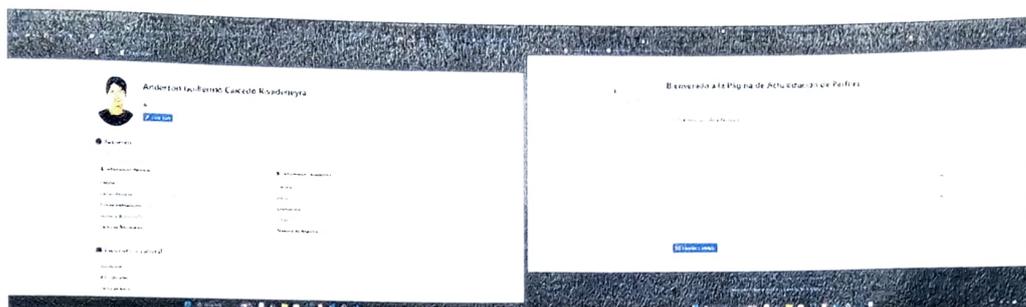
Nota: Se llevó a cabo una recopilación de información para definir la estructura y funcionalidades del sistema web en Django. Este proceso permitió organizar ideas, visualizar la distribución de módulos y establecer las principales acciones que debe realizar la plataforma.

Anexo 2: Capturas de las páginas principales.



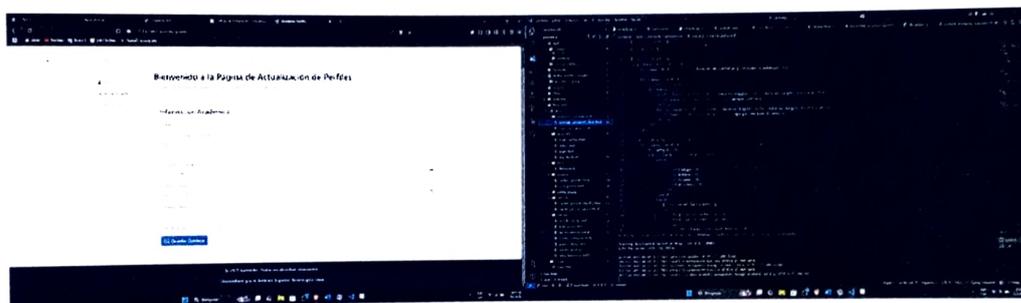
Nota: En esta fase, se realizó la verificación del diseño y la información presentada en la página de inicio del sistema web. Se revisaron los elementos visuales, la distribución del contenido y la claridad de los mensajes mostrados a los usuarios.

Anexo 3: Captura de los diseños de los perfiles.



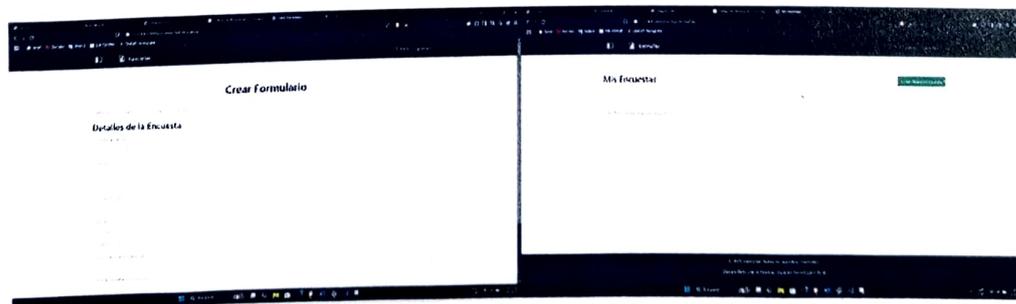
Nota: La primera imagen muestra la vista del perfil de usuario, donde se pueden visualizar los datos personales, académicos y laborales. La segunda imagen presenta la interfaz de actualización de información académica, donde el usuario puede modificar sus datos y guardar los cambios.

Anexo 4: Formulario de actualización de información.



Nota: En esta imagen se muestra la interfaz de gestión académica, donde los administradores pueden visualizar y administrar la lista de carreras y períodos académicos registrados en la plataforma. Se presentan opciones para agregar nuevas carreras y gestionar períodos académicos.

Anexo 5: Formularios de creación de encuestas.



Nota: Esta imagen muestra la interfaz principal de gestión de encuestas, donde los usuarios pueden buscar encuestas por nombre o descripción, filtrar por categorías y acceder a opciones de administración. Si no hay encuestas creadas, se muestra un mensaje indicándolo.

Anexo 6: Pagina de generación de reportes.



Nota: Esta imagen muestra el código fuente en Django Template correspondiente a la página de generación de reportes. Aquí se incluyen los selectores para elegir la carrera, período académico, tipo de reporte y formato del archivo. También se observa un error en la consola de depuración, indicando que la vista `generate_report` no ha sido encontrada, lo que sugiere que la URL o la función aún no están definidas correctamente en `urls.py` o `views.py`.