



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

**SCAPY, HERRAMIENTA EDA PARA EL ANÁLISIS
SIMBÓLICO DE CIRCUITOS ELÉCTRICOS.**

**TITULACIÓN INTEGRAL INFORME DE RESIDENCIA
PROFESIONAL.**

**PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERA EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES**

PRESENTA:
DIANA LAURA CORONA CERVANTES

NOMBRE DEL ASESOR:
DR. JOSÉ FEDERICO RAMÍREZ CRUZ



APIZACO, TLAX., AGOSTO 2024

Apizaco, Tlaxcala, **27/febrero /2024**

OFICIO No: SC/OPV/00086/24

ASUNTO: Liberación de proyecto para titulación integral

MARTÍN ROJAS RAMÍREZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

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

NOMBRE DEL EGRESADA:	DIANA LAURA CORONA CERVANTES
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	C19370725
NOMBRE DEL PROYECTO:	SCAPY, HERRAMIENTA EDA PARA EL ANÁLISIS SIMBÓLICO DE CIRCUITOS ELÉCTRICOS
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

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

ATENTAMENTE

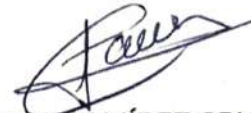
*Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®*



ALAN AUGUSTO GALLEGOS CUELLAR
JEFE DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO



JOSÉ FEDERICO RAMÍREZ CRUZ
ASESOR INTERNO

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

c.c.p. División de Estudios Profesionales
c.c.p. Depto. de Servicios Escolares
c.c.p. Archivo
AAGC/mrol*





Apizaco, Tlax., **21/marzo/2024**
No. OFICIO: DEP-CT/335/2024

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

DIANA LAURA CORONA CERVANTES
EGRESADA DE LA CARRERA DE
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
CON NÚMERO DE CONTROL: C19370725
P R E S E N T E

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** cuyo tema es: **"SCAPY, HERRAMIENTA EDA PARA EL ANÁLISIS SIMBÓLICO DE CIRCUITOS ELÉCTRICOS"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga:

AUTORIZACIÓN DE IMPRESION

Debiendo entregar **3** empastados del mismo, apropiadamente encuadernados, a la brevedad, para presentar su acto de recepción profesional.

Sin otro particular por el momento, le envié un cordial saludo.

ATENTAMENTE

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

MARTÍN ROJAS RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE
ESTUDIOS PROFESIONALES

c.p. archivo



Agradecimientos.

Quiero expresar mi más profundo agradecimiento a todas las personas que han contribuido de manera significativa en la realización de este proyecto.

En primer lugar, agradezco a Dios por su guía y bendiciones en cada etapa de mi vida. Su infinita sabiduría y fortaleza han sido mi fuente de inspiración y me han permitido superar los desafíos que he enfrentado en el camino. Su amor incondicional me ha dado la confianza necesaria para perseverar y alcanzar este logro académico.

Quiero agradecer a mis padres, Eloisa y Asael quienes han sido mi mayor apoyo a lo largo de mi vida y de este arduo proceso. Su amor, dedicación y constante motivación han sido fundamentales en mi formación académica y en el logro de mis metas. Gracias por creer en mí y por brindarme todo el respaldo necesario para alcanzar mis sueños.

A mis hermanos, Eisset y Zabdiel. Su ejemplo de dedicación, esfuerzo y logros ha sido una guía fundamental para mí. Aunque hoy me encuentro agradeciéndoles, no puedo evitar bromear un poco al respecto. Quién iba a pensar que algún día aspiraría a superarlos en algún ámbito, ¡pero aquí estoy, decidida a alcanzar nuevas alturas y seguir sus pasos brillantes! Su apoyo y bromas han sido motivación constante, y estoy agradecida por tenerlos como modelos a seguir.

También quiero agradecer a mi colega y compañero de vida Luis, por su colaboración en este proyecto. Su entusiasmo, conocimientos técnicos y trabajo en equipo han enriquecido esta investigación. Agradezco nuestras conversaciones, debates y la colaboración conjunta que nos ha permitido avanzar hacia la consecución de nuestros objetivos.

Quiero extender mi gratitud al doctor José Federico Ramírez Cruz y al doctor Luis Abraham Sánchez Gaspariano, mi asesores, cuya experiencia, conocimiento y orientación han sido invaluable en el desarrollo de este proyecto. Su apoyo constante, paciencia y compromiso con mi crecimiento académico han sido fundamentales para el éxito. Agradezco su dedicación y por brindarme la oportunidad de aprender de su vasta experiencia. Finalmente, quiero agradecer a todos mis amigos y seres queridos que me han brindado su apoyo incondicional, palabras de aliento y comprensión durante este proceso. Sus palabras de ánimo y gestos de amistad han sido un gran estímulo para mí.

Este logro no habría sido posible sin la colaboración, el apoyo y la contribución de todas estas personas. A cada uno de ustedes, gracias por formar parte de este camino y por su confianza en mí. Este proyecto no solo es el resultado de mi esfuerzo individual, sino también del valioso respaldo de cada una de estas personas.

Diana Laura Corona Cervantes

Índice

Resumen.	1
Abstract.	2
Introducción.	3
1. Generalidades de la empresa.	4
1.1. Información de la empresa.	4
1.2. Misión.	4
1.3. Visión.	4
1.4. Puesto del estudiante.	4
2. Problemas a resolver.	5
3. Objetivos.	5
3.1. Objetivo General.	5
3.2. Objetivos Específicos.	5
4. Justificación.	5
5. Marco Teórico.	7
5.1. Marco Histórico.	7
5.1.1. Análisis simbólico de circuitos.	7
5.1.2. Desarrollo de software con interfaz amigable.	7
5.1.3. Integración de HTML, CSS, JavaScript, Python y Flask.	8
5.2. Marco Referencial.	9
5.3. Marco Conceptual.	10
5.3.1. Análisis Simbólico de Circuitos Eléctricos.	10
5.3.2. Desarrollo de Software con Interfaz Web.	11
6. Desarrollo.	14
6.1. Planificación del proyecto.	14
6.2. Análisis de Requerimientos.	14
6.2.1. Requerimientos Funcionales.	14
6.2.2. Requerimientos de Rendimiento.	14
6.2.3. Requerimientos de Seguridad.	15
6.2.4. Requerimientos Técnicos.	15
6.3. Acondicionamiento y preparación del servidor.	15
6.4. Diseño de la interfaz web.	16
6.4.1. Maquetado y estructura general.	16
6.4.2. Gama de colores.	17
6.4.3. Arquitectura de la plataforma.	17
6.5. Desarrollo de la plataforma web.	18
7. Resultados.	23
8. Conclusiones.	25
8.1. Recomendaciones.	25

8.2. Experiencia Personal y Profesional Adquirida.	25
9. Competencias Desarrolladas.	26
Bibliografía.	27

Índice de figuras

1.	George Boole	7
2.	MATLAB	8
3.	Flask	9
4.	Circuito de un modelo hibrido tipo PI del transistor BJT	10
5.	Cronograma de actividades.	14
6.	Plantilla Base.	16
7.	Paleta Principal.	17
8.	Paleta Secundaria.	17
9.	Página Inicio.	18
10.	Página Inicio Sesión.	19
11.	Página para Registrar Usuarios.	19
12.	Página Inicio, después de iniciar sesión.	20
13.	Página para conocer la Documentación de la Plataforma.	20
14.	Página ¿Quiénes somos?	21
15.	Página de Contacto.	21
16.	Página para Cálculo Simbólico.	22
17.	Página Calculadoras.	22
18.	Circuito Amplificador Integrador.	23
19.	Netlist en un archivo .cir.	23
20.	Solución.	24

1. Resumen.

El presente proyecto se centra en el desarrollo de una plataforma web para el análisis simbólico de circuitos eléctricos, integrando la herramienta SCAPy [10]. La motivación surge de la creciente demanda de soluciones accesibles y colaborativas para estudiantes y profesionales en el campo de la ingeniería eléctrica y electrónica.

La plataforma se ha diseñado con una interfaz de usuario intuitiva, permitiendo cargar netlists de circuitos eléctricos y visualizar resultados de manera clara y comprensible. La integración eficiente de SCAPy en el backend brinda a los usuarios la capacidad de realizar análisis simbólicos avanzados sin la necesidad de software especializado costoso y propietario.

Además, se ha prestado especial atención a la escalabilidad, anticipando un potencial crecimiento en la carga de usuarios y en la complejidad de los circuitos analizados. La documentación clara y accesible se proporciona para orientar a los usuarios en el aprovechamiento óptimo de la plataforma.

Las pruebas de usuario han sido fundamentales en la fase de desarrollo, permitiendo la retroalimentación valiosa que ha contribuido a mejoras continuas en la usabilidad y rendimiento. La colaboración y el intercambio de conocimientos se promueven activamente, fomentando un ambiente propicio para el aprendizaje y la innovación en el análisis simbólico de circuitos eléctricos.

La combinación de tecnologías web modernas, la integración de SCAPy y el enfoque centrado en el usuario hacen de esta plataforma una contribución valiosa al campo, proporcionando una solución accesible, colaborativa y eficiente para el análisis simbólico de circuitos eléctricos.

2. Abstract.

This project focuses on the development of a web platform for symbolic analysis of electrical circuits, integrating SCAPy tool. The motivation arises from the growing demand for accessible and collaborative solutions for students and professionals in the field of electrical and electronic engineering.

The platform has been designed with an intuitive user interface, allowing you to load netlists of electrical circuits and visualize results in a clear and understandable manner. The efficient integration of SCAPy in the backend provides users with the ability to perform advanced symbolic analysis without the need for expensive and proprietary specialized software.

In addition, special attention has been given to scalability, anticipating potential growth in user load and the complexity of analyzed circuits. Clear and accessible documentation is provided to guide users in optimal utilization of the platform.

User testing has been crucial in the development phase, allowing valuable feedback that has contributed to continuous improvements in usability and performance. Collaboration and knowledge exchange are actively encouraged, fostering an environment conducive to learning and innovation in symbolic analysis of electrical circuits.

The combination of modern web technologies, the integration of SCAPy, and a user-centric approach make this platform a valuable contribution to the field, providing an accessible, collaborative, and efficient solution for symbolic analysis of electrical circuits.

3. Introducción.

En el contexto de la ingeniería eléctrica y electrónica, el análisis simbólico de circuitos eléctricos despierta un interés creciente, impulsado por la necesidad de comprender de manera profunda y eficiente el comportamiento de sistemas cada vez más complejos. Este proyecto aborda esta necesidad mediante el desarrollo de una plataforma web innovadora que integra la herramienta SCAPy, proporcionando a estudiantes y profesionales una herramienta accesible y colaborativa para el análisis simbólico de circuitos.

El análisis simbólico, a diferencia de los métodos tradicionales, permite la manipulación algebraica de las ecuaciones que describen un circuito, ofreciendo soluciones exactas en lugar de resultados numéricos aproximados. Este enfoque no solo proporciona una comprensión de los principios teóricos de los circuitos eléctricos, sino que también facilita la optimización y el diseño eficiente de sistemas eléctricos complejos.

La plataforma desarrollada se distingue por su interfaz de usuario intuitiva, que simplifica el proceso de carga de netlists y visualización de resultados. La integración efectiva de SCAPy en el backend brinda a los usuarios la capacidad de realizar análisis simbólicos avanzados sin la necesidad de software costoso y propietario. Este enfoque no solo democratiza el acceso a herramientas especializadas, sino que también fomenta la colaboración y el intercambio de conocimientos en la comunidad técnica.

La escalabilidad y el rendimiento se han considerado cuidadosamente, anticipando la demanda creciente de usuarios y la complejidad de los circuitos a analizar. A través de pruebas exhaustivas, se ha buscado perfeccionar esta plataforma, asegurando una experiencia óptima y una herramienta que evolucione con las necesidades cambiantes de la comunidad técnica.

Este proyecto no solo se posiciona como una contribución técnica al campo del análisis simbólico de circuitos eléctricos, sino que también representa un paso significativo del conocimiento, la colaboración efectiva y la innovación en la ingeniería eléctrica y electrónica. La combinación de tecnologías web modernas, la integración eficaz de SCAPy y un diseño centrado en el usuario converge en una plataforma que impulsa la accesibilidad, la colaboración y la eficiencia en el análisis simbólico de circuitos eléctricos.

4. Generalidades de la empresa.

4.1. Información de la empresa.

Nombre: Benemérita Universidad Autónoma de Puebla.

Dirección: Av. San Claudio esq. 18 Sur edif. FCE1 Col. San Manuel, Ciudad Universitaria, Puebla, Pue. C.P. 72570.

Teléfono: 222 229 55 00 Ext. 7400 y 7402

Sitio Web: <https://www.buap.mx>

4.2. Misión.

La Benemérita Universidad Autónoma de Puebla es una institución pública y autónoma consolidada a nivel nacional, comprometida con la formación integral de profesionales y ciudadanos críticos y reflexivos en los niveles de educación media superior, superior y posgrado, que son capaces de generar, adaptar, recrear, innovar y aplicar conocimientos de calidad y pertinencia social.

La Universidad fomenta la investigación, la creación y la divulgación del conocimiento, promueve la inclusión, la igualdad de oportunidades y la vinculación; coadyuva como comunidad del conocimiento al desarrollo del arte, la cultura, la solución de problemas económicos, ambientales, sociales y políticos de la región y del país, bajo una política de transparencia y rendición de cuentas, principios éticos, desarrollo sustentable, en defensa de los derechos humanos, de tolerancia y honestidad; contribuyendo a la creación de una sociedad proactiva, productiva, justa y segura.

4.3. Visión.

La Benemérita Universidad Autónoma de Puebla es líder en el país y cuenta con posicionamiento internacional. La calidad y pertinencia de su oferta educativa y servicios académicos están sustentadas en una planta académica sólida y reconocida, un modelo educativo pertinente, flexible y enfocado en el estudiante, un desarrollo científico y tecnológico, una amplia influencia en la cultura y las artes, una estructura académica que funciona en redes de cooperación y colaboración nacionales e internacionales, una estructura administrativa y de gestión ágil, funcional, acreditada y flexible que apoya el quehacer académico, bajo una política de transparencia, rendición de cuentas y desarrollo sustentable.

4.4. Puesto del estudiante.

Asistente de Investigación en el programa de posgrado en Ciencias de la Electrónica por la Benemérita Universidad Autónoma de Puebla, colaborando con el Ing. Luis Cortés Ramírez y bajo la supervisión del Dr. Luis Abraham Sánchez Gaspariano para la integración del algoritmo en un sitio web dedicado al análisis simbólico de circuitos eléctricos.

5. Problemas a resolver.

La problemática que motiva este proyecto radica en la falta de accesibilidad en las herramientas existentes para el análisis simbólico de circuitos eléctricos en el ámbito de la ingeniería eléctrica y electrónica. Actualmente, los estudiantes y profesionales enfrentan obstáculos significativos al utilizar software especializado costoso y propietario para llevar a cabo estas tareas.

La carencia de plataformas intuitivas limita el acceso a funciones de análisis simbólico, lo cual dificulta el proceso de aprendizaje y aplicación de conceptos clave en el diseño y evaluación de circuitos eléctricos. Además, la dependencia de software propietario impone barreras económicas, especialmente para estudiantes con recursos limitados.

La falta de soluciones en línea colaborativas también obstaculiza el intercambio de conocimientos y la colaboración en la comunidad académica y profesional. Este proyecto aborda esta problemática al desarrollar una plataforma web en Python [9] que integra SCAPy, ofreciendo una alternativa accesible, intuitiva y colaborativa para el análisis simbólico de circuitos eléctricos. Al superar estas limitaciones, se busca facilitar el aprendizaje, promover la innovación y fomentar la colaboración en el campo de la ingeniería eléctrica y electrónica.

6. Objetivos.

6.1. Objetivo General.

Desarrollar una plataforma web en Python integrando SCAPy, Herramienta EDA para el análisis simbólico de circuitos.

6.2. Objetivos Específicos.

1. Identificar los requerimientos de funcionalidad e interfaz de la plataforma para los usuarios potenciales.
2. Configurar y optimizar el servidor de hosting de manera que cumpla con los requisitos de rendimiento, asegurando una alta disponibilidad y respuesta rápida a las solicitudes de los usuarios para la fase de prueba.
3. Desarrollar un backend robusto en Python que sea capaz de gestionar el flujo de datos y las solicitudes de los usuarios.
4. Diseñar e implementar la interfaz de usuario de la plataforma web para que sea intuitiva y fácil de usar.
5. Integrar la herramienta SCAPy a la plataforma web.
6. Realizar pruebas de usuario y recopilar comentarios para mejorar continuamente la plataforma.

7. Justificación.

La justificación de este proyecto radica en la demanda de herramientas que faciliten el análisis simbólico de circuitos eléctricos, tanto por parte de estudiantes como de profe-

sionales en ingeniería eléctrica y electrónica. La implementación de una plataforma web integrando una herramienta de análisis simbólico permitirá el acceso a esta funcionalidad de manera más accesible y conveniente, eliminando la necesidad de software especializado costoso y propietario. Además, al ser una plataforma en línea, fomentará la colaboración y el intercambio de conocimientos en la comunidad, promoviendo así el aprendizaje y la innovación en este campo.

8. Marco Teórico.

8.1. Marco Histórico.

8.1.1. Análisis simbólico de circuitos.

El cálculo simbólico de circuitos eléctricos tiene una larga historia que se remonta a los albores de la ingeniería eléctrica. Los primeros métodos para analizar circuitos eléctricos se basaban en técnicas manuales que podían ser tediosas y propensas a errores. A finales del siglo XIX, se desarrollaron los primeros métodos de cálculo simbólico para analizar circuitos eléctricos. Estos métodos se basaban en el uso de álgebra simbólica para representar las variables del circuito, como la tensión, la corriente y la resistencia. A la par, George Boole (Figura 1), desarrolló la lógica booleana, un concepto fundamental en el análisis de circuitos lógicos y digitales. Este enfoque simbólico permitió la simplificación y manipulación algebraica de ecuaciones, allanando el camino para una comprensión más profunda de su comportamiento.



Figura 1: George Boole

A principios del siglo XX, se desarrollaron los primeros métodos numéricos para analizar circuitos eléctricos. Estos métodos se basaban en el uso de computadoras para resolver las ecuaciones que describen el comportamiento del circuito. Los métodos numéricos eran más precisos que los métodos simbólicos, pero también eran más costosos y requerían más tiempo de computación.

En la década de 1970, se desarrollaron los primeros programas de computadora para el análisis simbólico de circuitos eléctricos. Estos programas permitieron a los ingenieros analizar circuitos eléctricos de forma rápida y precisa. En la actualidad, existen muchos programas de computadora disponibles para el análisis simbólico de circuitos eléctricos.

8.1.2. Desarrollo de software con interfaz amigable.

Paralelo al avance en el análisis simbólico, el desarrollo de software con interfaz amigable e intuitiva se convirtió en un objetivo clave. El auge de las aplicaciones web ha permitido la creación de interfaces que facilitan la interacción del usuario con herramientas técnicas. Plataformas educativas y de diseño, como MATLAB [5] (Figura 2) y SPICE [8], han

influido en la búsqueda de interfaces intuitivas para facilitar el uso de herramientas de análisis en ingeniería eléctrica.



Figura 2: MATLAB

El desarrollo de plataformas web usando Flask [1] es una tecnología relativamente nueva. Es un microframework web escrito en Python que se utiliza para crear aplicaciones web ligeras y escalables, se lanzó por primera vez en 2010 y ha ganado popularidad en los últimos años.

Flask es una buena opción para el desarrollo de plataformas web porque es fácil de aprender y usar, también es muy flexible y se puede utilizar para crear una amplia gama de aplicaciones web.

Existen muchas plataformas web que se han desarrollado utilizando Flask. Algunas de ellas son:

- Reddit
- Airbnb
- Pinterest
- Netflix

En la actualidad, la combinación de tecnologías web, como HTML [11], CSS [2], y JavaScript [6], junto con frameworks como Flask en Python, ha permitido el desarrollo de aplicaciones web robustas y accesibles. La integración de estas tecnologías proporciona no solo una interfaz gráfica atractiva, sino también una experiencia de usuario amigable que facilita la interacción con el análisis simbólico de circuitos.

8.1.3. Integración de HTML, CSS, JavaScript, Python y Flask.

La adopción de tecnologías web modernas permite la creación de interfaces de usuario intuitivas y dinámicas. HTML (Hypertext Markup Language) proporciona la estructura básica del contenido web, mientras que CSS (Cascading Style Sheets) se encarga de la presentación y el diseño. La incorporación de JavaScript añade interactividad a la interfaz, mejorando la experiencia del usuario con funciones dinámicas.

Python, como lenguaje de programación, se utiliza en el backend del desarrollo web, facilitando la lógica detrás de la interfaz y el análisis simbólico. Flask (Figura 3), un framework ligero de Python, simplifica la creación de aplicaciones web, proporcionando las herramientas necesarias para conectar el frontend con el backend de manera eficiente y escalable.



Figura 3: Flask

8.2. Marco Referencial.

Fontana [4] presenta en su artículo las principales capacidades y posibles aplicaciones de SapWin 4.0, un paquete de software desarrollado por los autores para la simulación de circuitos analógicos lineales. Este programa, una mejora de su versión anterior, ha sido utilizado en diversas escuelas, instituciones, industrias y centros de investigación a lo largo de las últimas dos décadas. SapWin 4.0 se basa en técnicas de análisis simbólico y se enfoca en áreas como análisis, diseño y docencia en Ingeniería Eléctrica. Las sugerencias y correcciones de usuarios de todo el mundo se recopilaron para generar una versión renovada del software, que podría convertirse en una herramienta eficaz en el campo de los circuitos analógicos, superando las limitaciones de los simuladores numéricos clásicos.

Hao Yu [12] describe el desarrollo de una herramienta de simulación de circuitos simbólicos basada en web llamada AICE (Analog IC Explorer), en el contexto del auge de la informática basada en Internet. A diferencia de los simuladores SPICE, las herramientas simbólicas son compactas y pequeñas. La disponibilidad en línea de esta herramienta simbólica ofrece beneficios como multiplataforma, fácil accesibilidad, interacción amigable, documentación sencilla, y apoyo al aprendizaje colaborativo y diseño. La herramienta propuesta incluye un editor de esquemas, generador de funciones simbólicas, sistema de gestión de archivos y administrador de cuentas de usuario. El artículo también presenta las plataformas y herramientas de desarrollo populares, demostrando el servicio en ejecución mediante ejemplos.

Paulú [7] presenta un editor de esquemas gráficos basado en web para el análisis de circuitos eléctricos y electrónicos. Este editor permite el análisis de circuitos lineales y no lineales de tiempo continuo, así como circuitos lineales conmutados periódicamente. En el caso de circuitos lineales, los resultados pueden obtenerse de forma simbólica. Utiliza el software Spice OPUS y Maple con el paquete PraCAN como motor de cálculo. Los circuitos creados en este editor pueden convertirse en bloques funcionales editables mediante código de programación. Además, incluye un modelo de amplificador operacional ideal, lineal y no lineal. El desarrollo del sistema tiene lugar en el Departamento de Teoría de Circuitos, con el propósito de apoyar la educación en ingeniería y la investigación.

Dodović [3] presenta el software SymPyCAP, diseñado para el análisis simbólico de circuitos eléctricos lineales invariables en el tiempo. Desarrollado en Python, utiliza un análisis nodal modificado para resolver ecuaciones en el dominio de la Transformada de Laplace y en el dominio de la Transformada Fasorial. Se ilustra la capacidad de SymPyCAP para resolver circuitos eléctricos mediante ejemplos que incluyen diversos circuitos representativos.

8.3. Marco Conceptual.

8.3.1. Análisis Simbólico de Circuitos Eléctricos.

El análisis simbólico de circuitos eléctricos es un enfoque teórico que implica la manipulación algebraica de símbolos para obtener soluciones simbólicas en lugar de valores numéricos. Se basa en principios fundamentales de la teoría de circuitos y la lógica matemática para representar y entender el comportamiento de los circuitos eléctricos (Figura 4). Este enfoque proporciona ecuaciones simbólicas que describen las relaciones entre voltajes, corrientes y elementos del circuito, permitiendo una comprensión profunda y analítica de su funcionamiento.

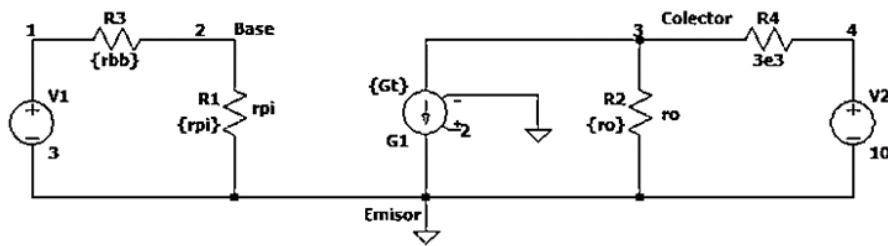


Figura 4: Circuito de un modelo híbrido tipo PI del transistor BJT

Métodos Tradicionales de Análisis de Circuitos.

Antes del análisis simbólico, los ingenieros eléctricos dependían principalmente de métodos analíticos y técnicas algebraicas convencionales para resolver sistemas de ecuaciones derivadas de circuitos eléctricos. Estos métodos, aunque efectivos para circuitos simples, se volvían cada vez más complejos y laboriosos a medida que se enfrentaban a circuitos más complicados, limitando su aplicabilidad y eficiencia.

Álgebra Computacional.

Se apoya en el campo del álgebra computacional, que combina principios matemáticos y técnicas algorítmicas para manipular expresiones matemáticas simbólicas. El uso de herramientas de álgebra computacional, como SCAPy, permite la simplificación y manipulación simbólica de ecuaciones, facilitando la obtención de soluciones más rápidas y precisas.

Ventajas del Análisis Simbólico.

El análisis simbólico de circuitos eléctricos ofrece una serie de ventajas sobre los métodos tradicionales de análisis, como el análisis numérico o el análisis manual. Algunas de las principales ventajas del análisis simbólico son:

1. **Precisión:** Es más preciso que el análisis numérico, ya que no se ve afectado por errores de redondeo. Esto es especialmente importante para el análisis de circuitos con componentes no lineales.
2. **Flexibilidad:** Es más flexible que el análisis numérico, ya que no se limita a un conjunto específico de ecuaciones. Esto permite al ingeniero analizar una amplia gama de circuitos con diferentes topologías y componentes.

-
3. **Eficiencia:** Puede ser más eficiente que el análisis numérico para circuitos con un gran número de componentes. Esto se debe a que el análisis simbólico solo necesita calcular las expresiones simbólicas para las variables del circuito, mientras que el análisis numérico necesita calcular los valores numéricos de las variables para cada punto de operación.
 4. **Comprensión:** Puede proporcionar una mejor comprensión del comportamiento del circuito que el análisis numérico. Esto se debe a que las expresiones simbólicas para las variables del circuito pueden ser interpretadas en términos de los principios físicos que gobiernan el comportamiento del circuito.
 5. **Facilidad de uso:** Es más fácil de usar que el análisis manual, ya que no requiere que el ingeniero escriba las ecuaciones del circuito a mano. Los programas de computadora para el análisis simbólico de circuitos eléctricos pueden generar automáticamente las ecuaciones del circuito a partir de una descripción gráfica del circuito.
 6. **Automatización:** Se puede automatizar fácilmente, lo que lo hace ideal para el análisis de grandes conjuntos de circuitos.
 7. **Reutilización:** Las expresiones simbólicas para las variables del circuito se pueden reutilizar para analizar diferentes configuraciones del circuito.
 8. **Escalabilidad:** El análisis simbólico se puede escalar a circuitos con un gran número de componentes.
 9. **Visibilidad:** Permite visualizar el comportamiento del circuito en términos de variables simbólicas, lo que puede facilitar la comprensión del comportamiento del circuito.
 10. **Comunicación:** Las expresiones simbólicas para las variables del circuito se pueden comunicar fácilmente a otros ingenieros, lo que facilita la colaboración en el diseño y análisis de circuitos.

8.3.2. Desarrollo de Software con Interfaz Web.

El desarrollo de software con interfaz web es un componente esencial de este proyecto. Se utiliza HTML (Hypertext Markup Language) para estructurar el contenido web, CSS (Cascading Style Sheets) para diseñar y dar estilo a la interfaz, y JavaScript para agregar dinamismo e interactividad. Estos elementos se combinan para crear una interfaz de usuario atractiva y amigable.

Evolución de las Tecnologías Web.

La web ha recorrido un largo camino desde sus inicios en los años 90. A lo largo de su evolución, ha experimentado diferentes etapas, cada una con sus propias características y tecnologías distintivas.

Web 1.0 (1991-2004): La era de la información estática.

En esta etapa, la web era principalmente una colección de páginas estáticas con contenido informativo. Los usuarios solo podían leer la información, sin interactuar con ella. Los sitios web eran creados por programadores con conocimientos de HTML y CSS.

Web 2.0 (2005-2010): La era de la participación.

La web 2.0 trajo consigo una mayor participación e interacción de los usuarios. Las redes sociales, blogs, wikis y plataformas de contenido generado por el usuario (UGC) proliferaron. La tecnología Flash permitió animaciones e interactividad en los sitios web.

Web 3.0 (2011-presente): La era semántica.

La web 3.0 se centra en la organización de la información de forma que sea comprensible para las máquinas, permite que los datos sean procesados y analizados por la inteligencia artificial, lo que abre nuevas posibilidades para la búsqueda, la recomendación de contenido y la personalización.

Web 4.0 (2020-presente): La era de la inteligencia artificial y la ubicuidad.

La web 4.0 está en sus primeras etapas de desarrollo, pero se espera que revolucione la forma en que interactuamos con la web. La inteligencia artificial jugará un papel fundamental, permitiendo que los sitios web sean más inteligentes, adaptables y sensibles a las necesidades de los usuarios. La realidad aumentada y la realidad virtual también tendrán un impacto significativo en la experiencia web.

Tecnologías clave que impulsan la evolución

- **HTML y CSS:** Lenguajes de marcado que dan forma y estilo a las páginas web.
- **JavaScript:** Lenguaje de programación que permite la interactividad en los sitios web.
- **AJAX:** Técnica que permite la actualización parcial de las páginas web sin necesidad de recargarlas.
- **Web services:** APIs que permiten la comunicación entre diferentes aplicaciones web.
- **Cloud computing:** Modelo de computación que ofrece recursos informáticos a través de internet.
- **Inteligencia artificial:** Tecnología que permite a las máquinas aprender y tomar decisiones de forma autónoma.

Impacto de la evolución de las tecnologías web:

Acceso a la información: La web ha democratizado el acceso a la información, permitiendo que personas de todo el mundo puedan acceder a una gran cantidad de recursos.

Comunicación: La web ha facilitado la comunicación entre personas, permitiendo que se conecten e interactúen sin importar la distancia.

Comercio electrónico: La web ha revolucionado el comercio, permitiendo que las empresas vendan sus productos y servicios a un público global.

Entretenimiento: La web ofrece una amplia gama de opciones de entretenimiento, como juegos, música y videos.

Educación: La web ha abierto nuevas posibilidades para la educación, permitiendo que las personas aprendan en línea a su propio ritmo.

Frameworks de Desarrollo Web en Python.

Python se ha convertido en uno de los lenguajes de programación más populares para el desarrollo web, gracias a su sintaxis clara y concisa, su amplia comunidad y la gran variedad de frameworks disponibles. Estos agilizan el proceso de desarrollo web al proporcionar una estructura predefinida y componentes reutilizables.

Un framework de desarrollo web es un conjunto de herramientas y bibliotecas que ayudan a los desarrolladores a crear aplicaciones web de manera eficiente. Ofrecen una estructura básica para la organización del código, la gestión de rutas, la interacción con la base de datos y otras tareas comunes.

Los frameworks de Python ofrecen varios beneficios, como:

1. **Aceleran el desarrollo:** Al proporcionar una estructura predefinida, los frameworks permiten a los desarrolladores empezar a construir aplicaciones web rápidamente.
2. **Simplifican el código:** Encapsulan tareas comunes en módulos y bibliotecas, lo que reduce la cantidad de código que se debe escribir.
3. **Promueven la reutilización de código:** Suelen venir con componentes y módulos pre-diseñados que se pueden reutilizar en diferentes proyectos.
4. **Mejoran la seguridad:** Muchos frameworks de Python incluyen características de seguridad integradas, como protección CSRF y validación de entrada.
5. **Facilitan el mantenimiento:** Ayudan a mantener el código organizado y modular, lo que facilita su mantenimiento y actualización.

Existen numerosos frameworks de Python para el desarrollo web, cada uno con sus propias características y ventajas. Los más populares son:

- **Django:** Un framework de pila completa que ofrece una solución completa para el desarrollo web. Es ideal para aplicaciones web complejas y de alto rendimiento.
- **Flask:** Un microframework ligero y flexible que es ideal para aplicaciones web pequeñas y simples.
- **Pyramid:** Un framework flexible y escalable que se puede adaptar a diferentes necesidades.
- **TurboGears:** Un framework rápido y eficiente que se basa en el lenguaje de plantillas Genshi.
- **Web2py:** Un framework de pila completa que incluye una gran cantidad de features predefinidas.

Desarrollo Backend con Python y Flask.

Python, conocido por su legibilidad y versatilidad, se ha convertido en un lenguaje de programación popular para el desarrollo backend de aplicaciones web. Flask, un framework ligero y flexible, ofrece un entorno adecuado para construir aplicaciones web, proporcionando las herramientas necesarias para la creación de rutas, gestión de solicitudes y respuestas, y conexión con bases de datos.

9. Desarrollo.

9.1. Planificación del proyecto.

Para la planificación se realizó el siguiente cronograma (Figura 5), donde se detallan las actividades realizadas para este proyecto.

ACTIVIDAD		AGOSTO					SEPTIEMBRE				OCTUBRE				NOVIEMBRE				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1. PLANIFICACIÓN DEL PROYECTO.	P																		
	R																		
2. ANÁLISIS DE REQUERIMIENTOS.	P																		
	R																		
3. ACONDICIONAMIENTO Y PREPARACIÓN DEL SERVIDOR.	P																		
	R																		
4. DISEÑO DE LA INTERFAZ WEB.	P																		
	R																		
5. DESARROLLO DE LA PLATAFORMA WEB.	P																		
	R																		
6. INTEGRACIÓN DE SCAPy A LA PLATAFORMA WEB.	P																		
	R																		
7. PRUEBAS Y DEPURACIÓN DE LA PLATAFORMA WEB.	P																		
	R																		
8. EVALUACIÓN Y MEJORAS DE LA PLATAFORMA WEB.	P																		
	R																		

Figura 5: Cronograma de actividades.

9.2. Análisis de Requerimientos.

9.2.1. Requerimientos Funcionales.

Interfaz de Usuario Intuitiva: La plataforma deberá contar con una interfaz de usuario intuitiva y fácil de usar, permitiendo a los usuarios cargar netlists, configurar parámetros y visualizar resultados de manera clara.

Funcionalidades de SCAPy Integradas: Integrar de manera efectiva la herramienta SCAPy para realizar el análisis simbólico de circuitos eléctricos a partir de las netlists proporcionadas por los usuarios.

Gestión de Usuarios: Implementar un sistema de gestión de usuarios que permita el registro, autenticación y autorización, diferenciando entre roles de usuario para garantizar la seguridad y el control del acceso.

Documentación Clara y Accesible: Proporcionar documentación clara y accesible que guíe a los usuarios en la carga de netlists, la interpretación de resultados y el aprovechamiento de las funcionalidades de la plataforma.

Pruebas de Usuario: Desarrollar un conjunto integral de pruebas de usuario que evalúen la eficacia, usabilidad y rendimiento de la plataforma, recopilando comentarios para mejoras continuas.

9.2.2. Requerimientos de Rendimiento.

Optimización del Servidor de Hosting: Configurar y optimizar el servidor de hosting para garantizar un rendimiento eficiente, alta disponibilidad y respuestas rápidas a las solicitudes de los usuarios durante la fase de prueba.

Escalabilidad: Diseñar la plataforma de manera que sea escalable para manejar un crecimiento potencial en la carga de usuarios y el volumen de análisis de circuitos.

9.2.3. Requerimientos de Seguridad.

Protección de Datos: Implementar medidas de seguridad para garantizar la protección de los datos de los usuarios, incluyendo encriptación de datos y procedimientos seguros de almacenamiento.

9.2.4. Requerimientos Técnicos.

Tecnologías Web Modernas: Utilizar tecnologías web modernas, incluyendo HTML5, CSS3 y JavaScript, para construir una interfaz atractiva y dinámica.

Desarrollo en Flask y Python: Desarrollar el backend de la plataforma utilizando el framework Flask en Python, aprovechando la versatilidad y eficiencia de estas tecnologías.

Integración de SCAPy: Integrar la herramienta SCAPy de manera efectiva en el backend de la plataforma, permitiendo el análisis simbólico de circuitos eléctricos de forma eficiente.

9.3. Acondicionamiento y preparación del servidor.

Esta etapa comienza con la preparación del entorno de desarrollo, asegurando una configuración eficiente y coherente. Para ello, se aborda la instalación de las herramientas fundamentales: Python, Visual Studio Code y Flask.

La base del desarrollo es la instalación de Python, un lenguaje versátil y ampliamente utilizado. La versión más reciente se descarga desde el sitio oficial de Python (<https://www.python.org/>), y se sigue un proceso guiado para la instalación. Python proporciona las capacidades necesarias para el desarrollo del backend de la plataforma.

Visual Studio Code (VS Code) se selecciona como el entorno de desarrollo integrado (IDE) debido a su versatilidad y soporte extenso para Python. Tras la instalación de VS Code desde su sitio oficial (<https://code.visualstudio.com/>), se integran extensiones específicas para Python, facilitando la escritura de código, la depuración y la gestión de entornos virtuales.

Flask, un framework ligero de Python, se selecciona para el desarrollo del backend de la plataforma web. Dentro del entorno virtual, se inicia un nuevo proyecto Flask mediante la instalación del paquete Flask con el siguiente comando:

```
pip install Flask
```

Listing 1: Comando para instalar Flask

A medida que la plataforma toma forma, se procede al despliegue en un servidor de pruebas. La configuración y optimización del servidor se realizan para garantizar un rendimiento eficiente, alta disponibilidad y respuestas rápidas a las solicitudes de los usuarios.

9.4. Diseño de la interfaz web.

9.4.1. Maquetado y estructura general.

La plantilla base (Figura 6) para esta plataforma permite estandarizar la estructura de contenidos que se mostrarán.

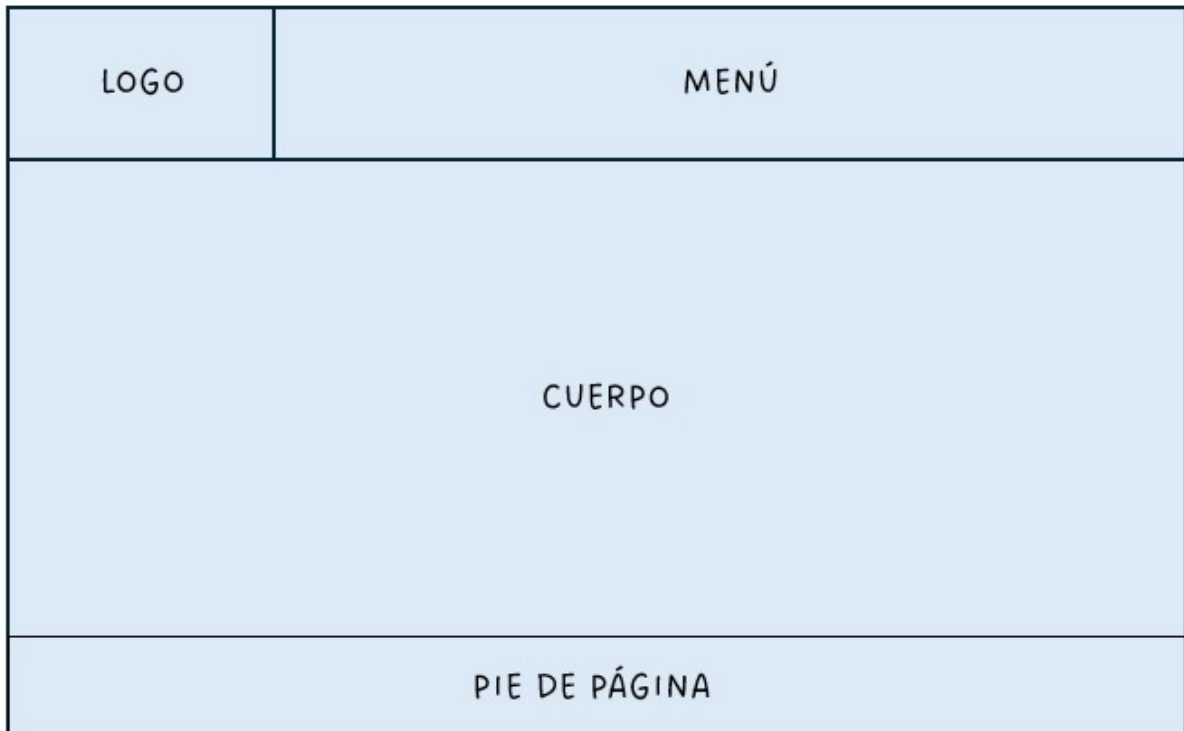


Figura 6: Plantilla Base.

Títulos.

Fuente: arial

Tamaño: 28px

Color: #f9f9f9

Contenido.

Fuente: arial

Tamaño: 16px

Color: #f9f9f9

Contenido.

Fondo: #f9f9f9

Ancho: 100 %

Alto: Auto

9.4.2. Gama de colores.

Se establecen los colores que deben usarse en el contenido, con el fin de homologar y mantener una identidad, por lo que se describe sus valores en hexadecimal.

Paleta principal.



Figura 7: Paleta Principal.

Paleta secundaria.

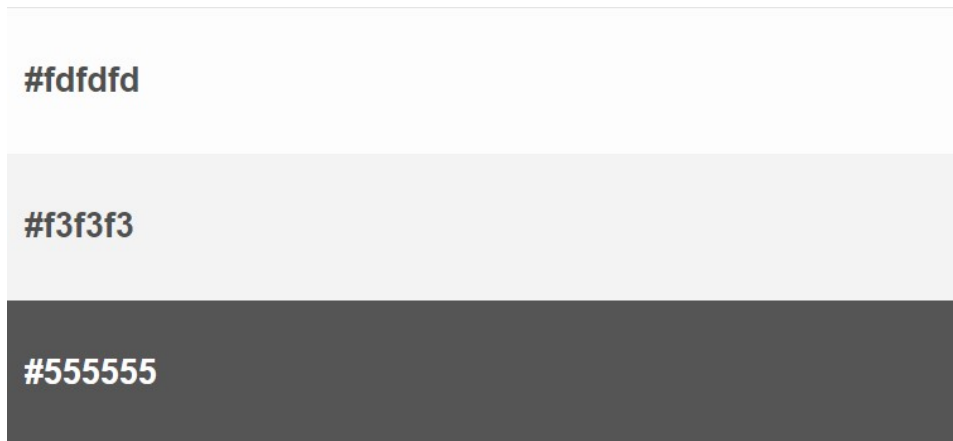


Figura 8: Paleta Secundaria.

9.4.3. Arquitectura de la plataforma.

La arquitectura de la plataforma web se ha diseñado de manera modular y escalable, asegurando un rendimiento eficiente y una fácil incorporación de nuevas funcionalidades. La plataforma sigue un modelo cliente-servidor y utiliza tecnologías modernas para garantizar una experiencia de usuario óptima.

Cliente. La interfaz de usuario se construye utilizando tecnologías web modernas, como HTML, CSS y JavaScript. Se implementa la comunicación asíncrona con el servidor

mediante solicitudes HTTP, lo que permite actualizaciones dinámicas de la interfaz sin necesidad de recargar la página completa.

Servidor. El backend de la plataforma se desarrolla utilizando Flask, un framework ligero de Python. Flask proporciona una estructura flexible y modular para gestionar las solicitudes HTTP, así como para la implementación de rutas, controladores y modelos. Se integra la herramienta SCAPy para realizar el análisis simbólico de los circuitos eléctricos. Las netlists cargadas por los usuarios se procesan para obtener ecuaciones simbólicas y resultados precisos.

Se implementa un sistema de gestión de usuarios para manejar el registro, autenticación y autorización. Esto asegura la seguridad de la plataforma y permite personalizar la experiencia del usuario según los roles asignados.

Bases de Datos. Se utiliza una base de datos para almacenar información crítica, como detalles de usuarios y configuraciones personalizadas. SQLAlchemy, una biblioteca de mapeo objeto-relacional en Python, facilita la interacción con la base de datos de manera eficiente.

Seguridad de Datos. Se implementan medidas de seguridad, como el cifrado de datos y la validación de entrada, para proteger la privacidad de los usuarios y garantizar la integridad de la plataforma.

9.5. Desarrollo de la plataforma web.

A continuación, se muestran las páginas que contiene la plataforma web:



Figura 9: Página Inicio.

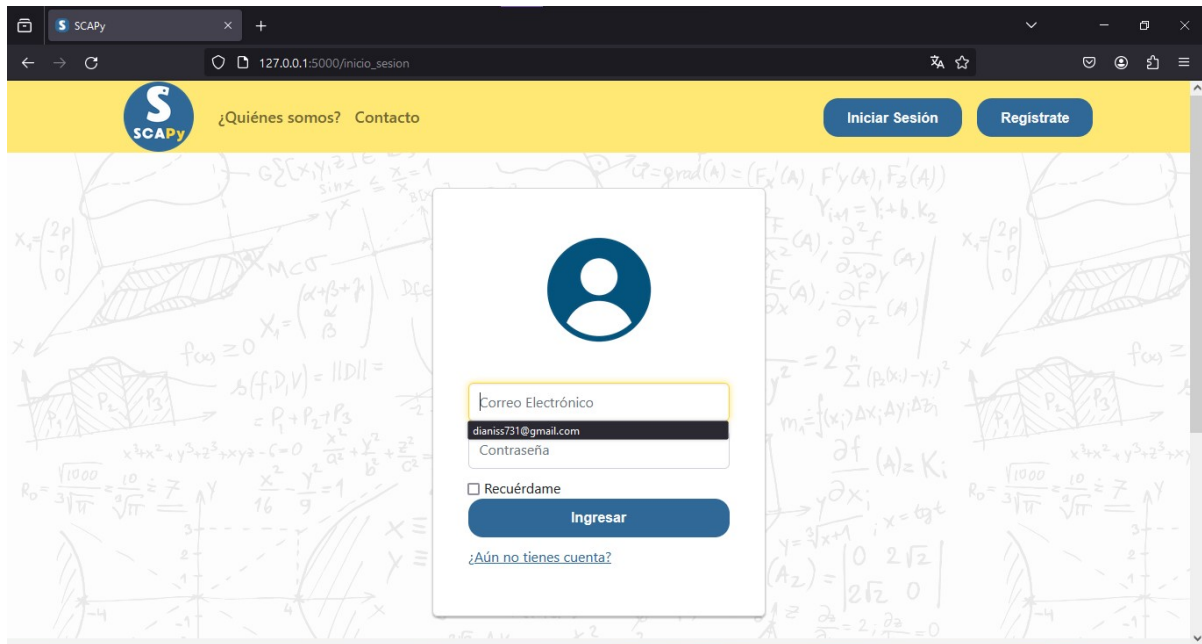


Figura 10: Página Inicio Sesión.

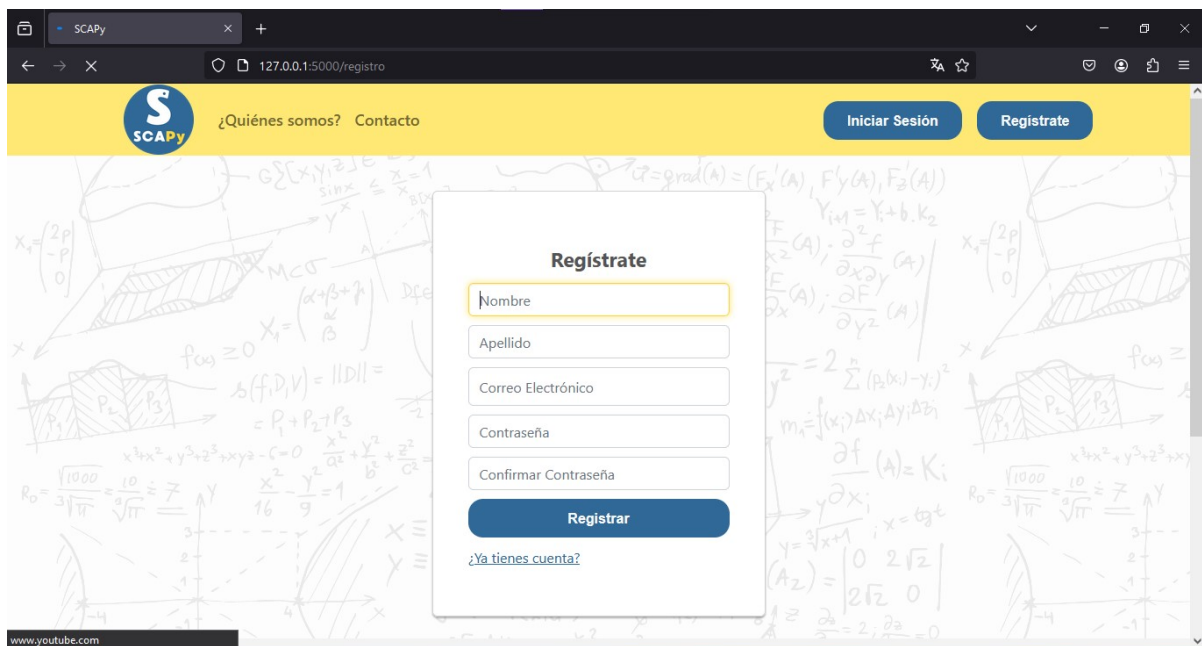


Figura 11: Página para Registrar Usuarios.



Figura 12: Página Inicio, después de iniciar sesión.

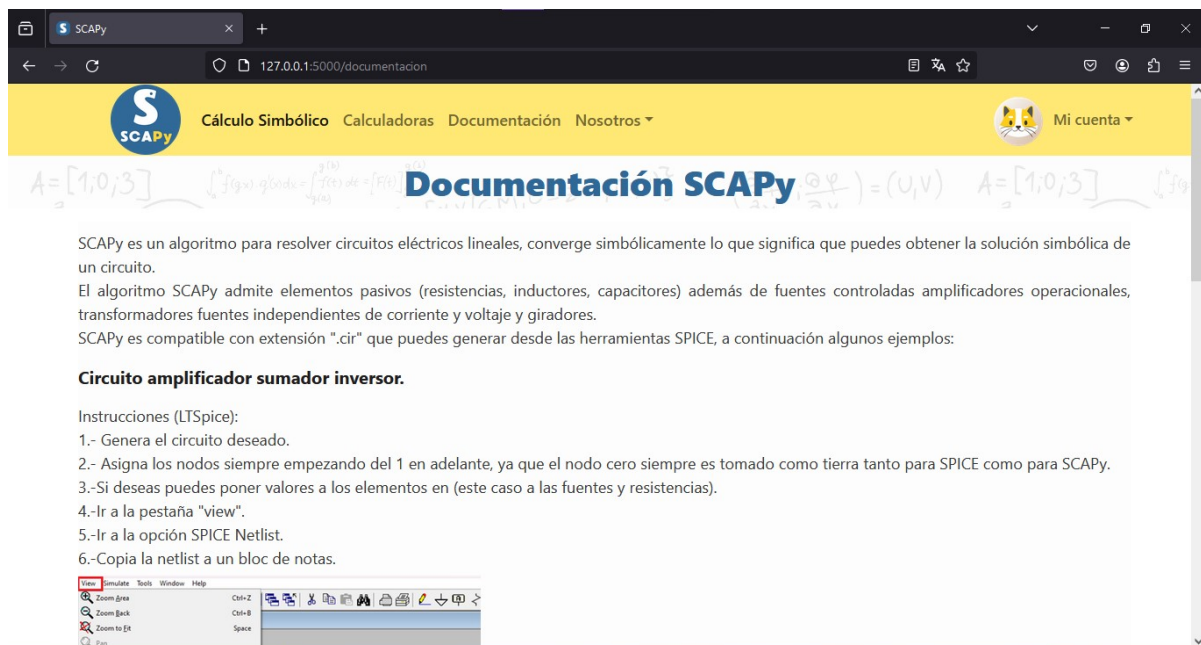


Figura 13: Página para conocer la Documentación de la Plataforma.

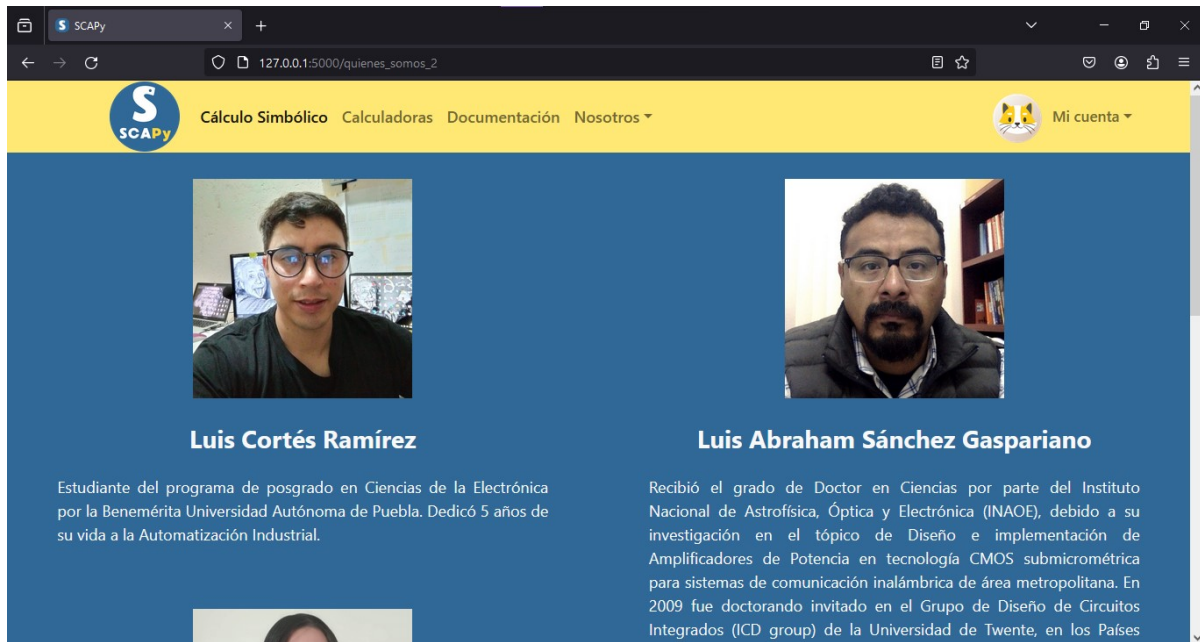


Figura 14: Página ¿Quiénes somos?

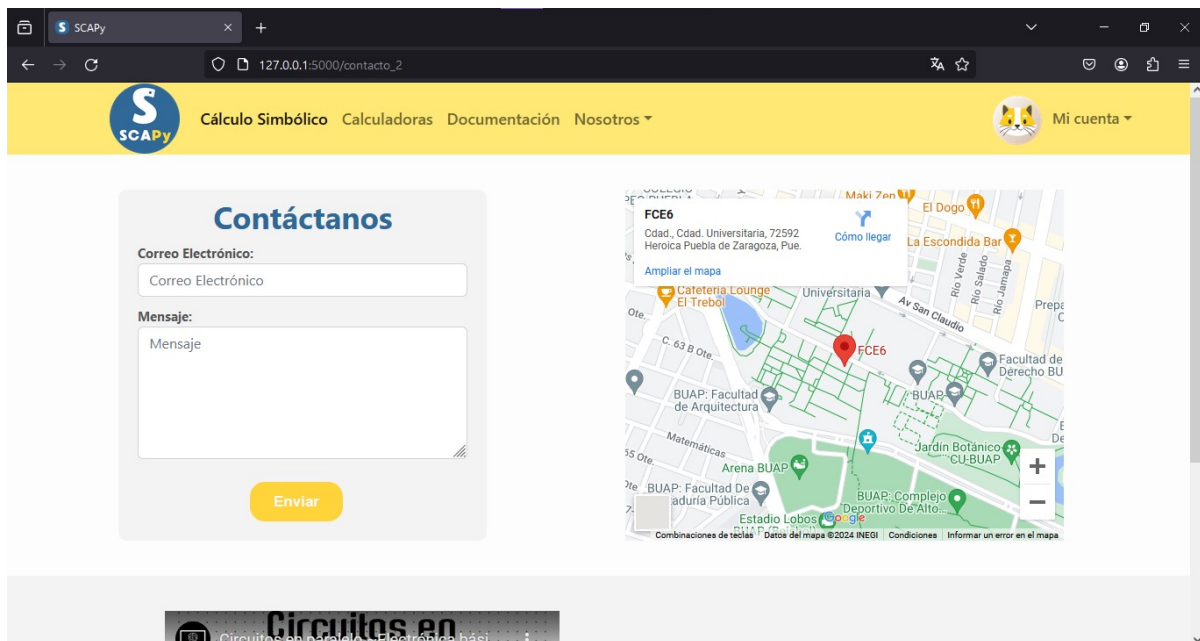


Figura 15: Página de Contacto.



Figura 16: Página para Cálculo Simbólico.



Figura 17: Página Calculadoras.

10. Resultados.

Se adjunta evidencia de los resultados obtenidos al ingresar algunos ejercicios con circuitos de ejemplo.

Se tiene el siguiente circuito (Figura 18) con su respectiva netlist (Figura 20).

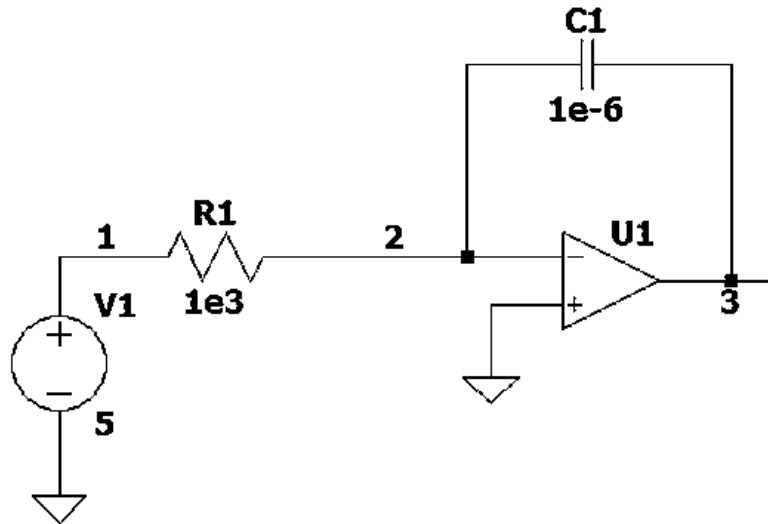


Figura 18: Circuito Amplificador Integrador.

```
O1 2 0 3
R1 1 2 1e3
C1 3 2 1e-6
V1 1 0 5
```

Figura 19: Netlist en un archivo .cir.

Para obtener la solución a este circuito, basta con cargar el archivo .cir de la netlist al sitio web y poder visualizar el resultado.

¡Cálculo Simbólico de Circuitos Eléctricos con SCAPy!

Paso 1:
Carga la netlist de tu circuito en un archivo .cir

Seleccionar archivo Ninguno archivo selec.

Paso 2:
Da click en el botón **Obtener solución** ¡Y visualízala!

Obtener solución

Resultado:
[V1] [0] [-V1/(S*R1*C1)] [-V1/R1] [V1/R1]

Figura 20: Solución.

11. Conclusiones.

En el transcurso de este proyecto, he llegado a varias conclusiones significativas que destacan la relevancia y el impacto potencial de la plataforma desarrollada. La integración de SCAPy en una plataforma web en Python no solo aborda la problemática de accesibilidad en el análisis simbólico de circuitos eléctricos, sino que también ofrece una solución colaborativa y económica para estudiantes y profesionales en ingeniería eléctrica y electrónica.

La plataforma proporciona un acceso más fácil y amigable a las funciones de análisis simbólico, eliminando las barreras económicas y técnicas asociadas con el software propietario. Los resultados obtenidos de las pruebas de usuario respaldan la efectividad de la plataforma, destacando su utilidad y facilidad de uso en un entorno educativo y profesional.

11.1. Recomendaciones.

Considerando el desarrollo de este proyecto, algunas recomendaciones se derivan para mejorar aún más la plataforma:

1. Ampliación de Funcionalidades: Explorar la posibilidad de expandir las funcionalidades de la plataforma para abarcar otros aspectos del análisis de circuitos eléctricos, proporcionando una herramienta más completa y versátil.
2. Optimización Continua: Realizar actualizaciones periódicas para optimizar el rendimiento de la plataforma y abordar posibles mejoras basadas en la retroalimentación continua de los usuarios.
3. Colaboración Institucional: Buscar oportunidades de colaboración con otras instituciones académicas o profesionales para enriquecer la plataforma y promover aún más la colaboración en la comunidad.

11.2. Experiencia Personal y Profesional Adquirida.

Esta experiencia ha sido fundamental para mi desarrollo profesional y académico. La colaboración estrecha con el estudiante de maestría y la aplicación práctica de mis conocimientos en programación y desarrollo web han fortalecido mis habilidades técnicas. La gestión de un proyecto de investigación me ha proporcionado una perspectiva valiosa sobre los desafíos y las oportunidades en el campo de la ingeniería eléctrica.

12. Competencias Desarrolladas.

Durante mi participación en este proyecto de, he tenido la oportunidad de desarrollar diversas competencias que considero fundamentales tanto a nivel técnico como profesional. A continuación, comparto las habilidades que he adquirido a lo largo de esta experiencia:

1. **Desarrollo en Python:** He fortalecido mis habilidades en programación al desarrollar el backend y frontend de la plataforma web utilizando Python.
2. **Integración de Herramientas EDA:** He ganado experiencia significativa al integrar herramientas específicas de Análisis de Diseño Electrónico (EDA), en este caso, SCAPy, en el entorno web del proyecto.
3. **Diseño de Interfaz de Usuario:** Mi participación en la creación y diseño de la interfaz de usuario de la plataforma web me ha permitido desarrollar habilidades en diseño centrado en el usuario.
4. **Configuración y Optimización del Servidor:** He adquirido conocimientos prácticos en la configuración y optimización de servidores de hosting para garantizar el rendimiento y la disponibilidad de la plataforma.
5. **Trabajo Colaborativo:** Colaborar estrechamente con un estudiante de maestría y otros miembros del equipo me ha proporcionado valiosas habilidades en trabajo en equipo y comunicación efectiva.
6. **Documentación Técnica:** Mi contribución a la documentación del proyecto ha mejorado mis habilidades en la redacción de documentos técnicos, explicando de manera clara el funcionamiento de la plataforma.
7. **Pruebas de Usuario:** Participar en la planificación y ejecución de pruebas de usuario ha mejorado mi capacidad para evaluar la usabilidad y recopilar retroalimentación valiosa.
8. **Gestión del Flujo de Datos:** Desarrollar un backend robusto en Python me ha proporcionado habilidades para gestionar eficientemente el flujo de datos dentro de la plataforma.
9. **Resolución de Problemas:** Enfrentar desafíos técnicos y conceptuales me ha permitido desarrollar habilidades sólidas para abordar problemas y encontrar soluciones efectivas.
10. **Comprensión del Análisis Simbólico:** He obtenido un profundo conocimiento en el análisis simbólico de circuitos eléctricos, lo cual considero valioso para futuros proyectos e investigaciones relacionadas.
11. **Colaboración en Proyectos de Investigación:** Participar en un proyecto de investigación académica ha enriquecido mi experiencia en el entorno de la investigación científica y tecnológica.

Estas competencias no solo son relevantes para el éxito de este proyecto en particular, sino que también son transferibles y aplicables en futuros proyectos y en mi desarrollo profesional continuo.

Bibliografía.

- [1] Armin Ronacher. *Flask*. English. 2010. URL: [%7Bhttps://flask.palletsprojects.com/%7D](https://flask.palletsprojects.com/).
- [2] CSS Working Group. *CSS*. 1996. URL: [%7Bhttps://www.w3.org/Style/CSS/%7D](https://www.w3.org/Style/CSS/).
- [3] Matija Dodović et al. “SymPyCAP: Software for symbolic analysis of electric circuit”. En: (2023), págs. 1-5. DOI: [10.1109/IcETRAN59631.2023.10192115](https://doi.org/10.1109/IcETRAN59631.2023.10192115).
- [4] G. Fontana et al. “A new simulation program for analog circuits using symbolic analysis techniques”. En: (2015), págs. 1-4. DOI: [10.1109/SMACD.2015.7301682](https://doi.org/10.1109/SMACD.2015.7301682).
- [5] The MathWorks Inc. *MATLAB version: 9.13.0 (R2022b)*. Natick, Massachusetts, United States, 2022. URL: <https://www.mathworks.com>.
- [6] Netscape Communications. *JavaScript*. 1995.
- [7] Filip Paulů y Jiří Hospodka. “Web-Based Application for Analysis of Electrical Circuits and Systems”. En: (2018), págs. 1-4. DOI: [10.23919/NTSP.2018.8524039](https://doi.org/10.23919/NTSP.2018.8524039).
- [8] Donald Pederson. *SPICE*. Universidad de California, Berkeley, 1973.
- [9] *Python*. Python Software Foundation. URL: <https://www.python.org>.
- [10] Luis Cortés Ramírez. *SCAPy*. 2024.
- [11] World Wide Web Consortium. *HTML*. 1993. URL: [%7Bhttps://www.w3.org/html/%7D](https://www.w3.org/html/).
- [12] Hao Yu y Guoyong Shi. “Developing a web-based symbolic circuit analysis tool for learning and design aid”. En: (2017), págs. 1-4. DOI: [10.1109/SMACD.2017.7981587](https://doi.org/10.1109/SMACD.2017.7981587).