



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN**

**APLICACIÓN WEB PARA EL MODELO DE GESTIÓN TURÍSTICA
SOSTENIBLE COMUNITARIA DE CHIMBORAZO UTILIZANDO
DJANGO**

**Trabajo de Titulación para optar al título de Ingeniero en Tecnologías de
la Información**

Autores:

Baltazar Yucailla Cristofer Vinicio

Lucero Peña Luis Fernando

Tutor:

Ing. Pamela Alexandra Buñay Guisñan, MsC.

Riobamba, Ecuador. 2024

DECLARATORIA DE AUTORÍA

Nosotros, Cristofer Vinicio Baltazar Yucailla y Luis Fernando Lucero Peña, con cédula de ciudadanía 1851000131 y 2350213548, autores del trabajo de investigación titulado: Aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo utilizando Django, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 08 de febrero de 2024.



Cristofer Vinicio Baltazar Yucailla

C.I: 1851000131



Luis Fernando Lucero Peña

C.I: 2350213548

DICTAMEN FAVORABLE DEL PROFESOR TUTOR



Dirección
Académica
VICERRECTORADO ACADÉMICO

en movimiento



UNACH-RGF-01-04-08.11
VERSIÓN 01: 06-09-2021

ACTA FAVORABLE - INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN

En la Ciudad de Riobamba, a los 17 días del mes de abril de 2024, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por los estudiantes **CRISTOFER VINICIO BALTAZAR YUCAILLA** con CC: **1851000131** y **LUIS FERNANDO LUCERO PEÑA** con CC: **2350213548**, de la carrera de **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado **"APLICACIÓN WEB PARA EL MODELO DE GESTIÓN TURÍSTICA SOSTENIBLE COMUNITARIA DE CHIMBORAZO UTILIZANDO DJANGO"**, por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



Emite e interfirmación por:
PAMELA ALEXANDRA
BUÑAY GUIÑAN

MsC. Pamela Alexandra Buñay Guisñan
TUTORA

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo utilizando Django**, presentado por **Cristofer Vinicio Baltazar Yucailla**, con cédula de identidad número **1851000131** y **Luis Fernando Lucero Peña**, con cédula de identidad número **2350213548**, bajo la tutoría de **Msc. Pamela Alexandra Buñay Guisñan**; certificamos que recomendamos la **APROBACIÓN** de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba el 23 de mayo de 2024.

Presidente del Tribunal de Grado
PhD. Fernando Molina



Firma

Miembro del Tribunal de Grado
Mgs. Milton López



Firma

Miembro del Tribunal de Grado
Mgs. Lady Espinoza



Firma

CERTIFICADO ANTIPLAGIO

Original



Dirección
Académica
VICERRECTORADO ACADÉMICO

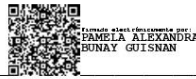


UNACH-RGF-01-04-08.15
VERSIÓN 01: 06-09-2021

CERTIFICACIÓN

Que, **BALTAZAR YUCAILLA CRISTOFER VINICIO** con CC: **1851000131** y **LUCERO PEÑA LUIS FERNANDO** con CC: **2350213548**, estudiantes de la Carrera **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, Facultad de **INGENIERÍA**; han trabajado bajo mi tutoría el trabajo de investigación titulado "**APLICACIÓN WEB PARA EL MODELO DE GESTIÓN TURÍSTICA SOSTENIBLE COMUNITARIA DE CHIMBORAZO UTILIZANDO DJANGO**", cumple con el 6%, de acuerdo al reporte del sistema Anti plagio **TURNITIN**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 14 de mayo de 2024



MsC. Pamela Alexandra Buñay Guisñan
TUTORA

DEDICATORIA

Cristofer Baltazar. -

Dedico este trabajo de investigación a mi familia, quienes han sido el pilar de mi fortaleza y la luz guía en mi camino académico. Con el corazón lleno de gratitud, esta tesis es un símbolo de mi profundo agradecimiento hacia ustedes.

A mi madre María Transito Yucailla, por su apoyo incansable, su amor incondicional y su sabiduría que ha sido mi faro en los momentos más difíciles. Su fe inquebrantable en mí me ha inspirado a perseguir mis sueños con determinación y pasión.

A mis amigos por su amistad genuina, con quienes compartimos muchos momentos los cuales han inspirado a resolver desafíos a lo largo de este trayecto.

Con amor y gratitud.

Luis Lucero. -

Dedico este trabajo de investigación a mis padres y hermano, pilares de mi vida y mi mayor fuente de inspiración. Esta tesis es el reflejo de mi gratitud hacia ustedes. Cada página, cada investigación, cada descubrimiento, ha estado impregnado de su amor y apoyo incondicional, elementos cruciales en mi camino académico.

Han sido la luz en mis momentos de duda. Su sabiduría y paciencia han sido la brújula que me ha orientado hacia el éxito. Sin su presencia constante y su inquebrantable fe en mis capacidades, este logro no habría sido posible.

Extiendo esta dedicatoria mis amigos, cuya amistad ha sido un pilar invaluable en este viaje. Su ánimo, comprensión y los momentos compartidos han aportado luz y alegría en los días más desafiantes.

Con profundo amor, gratitud y aprecio.

AGRADECIMIENTO

Expresamos nuestra profunda gratitud hacia Dios por brindarnos la perseverancia necesaria para superar los obstáculos que encontramos en nuestro camino. Un agradecimiento especial a nuestros padres, pilares fundamentales de nuestro viaje, cuyo amor incondicional, apoyo y sacrificios han sido esenciales para alcanzar nuestras metas. Este éxito es tan suyo como nuestro.

Extendemos nuestro reconocimiento a la Universidad Nacional de Chimborazo por acogernos y brindarnos la oportunidad de crecer profesionalmente, y a nuestros profesores por su entrega y paciencia en compartir su sabiduría, jugando un rol crucial en nuestro desarrollo académico.

Un agradecimiento especial para nuestra tutora, MsC. Pamela Buñay, cuya asesoría y soporte constante han sido fundamentales en la elaboración de nuestro proyecto de investigación. También, valoramos enormemente la confianza puesta en nosotros por parte del MsC. Daniel Guerrero.

Finalmente, a nuestros amigos y compañeros universitarios, les agradecemos por su amistad y solidaridad. Sus alentadoras palabras y apoyo constante han sido un motor que nos impulsó a dar lo mejor de nosotros y completar este significativo episodio de nuestra vida académica

ÍNDICE GENERAL

DECLARATORIA DE AUTORÍA

DICTAMEN FAVORABLE DEL PROFESOR TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

CERTIFICADO ANTIPLAGIO

DEDICATORIA

AGRADECIMIENTO

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

ABSTRACT

CAPÍTULO I. INTRODUCCION.....	15
1.1 Planteamiento y justificación de la investigación.....	16
1.2 Formulación del problema.....	16
1.3 Objetivos.....	16
1.3.1 General.....	16
1.3.2 Específicos.....	16
CAPÍTULO II. MARCO TEÓRICO.....	17
2.1 Django	17
2.1.1 Características de Django	17
2.1.2 ORM de Django	18
2.1.3 Mapeo de modelos.....	19
2.1.4 Migraciones en Django.....	19
2.1.5 Administración, seguridad y autenticación	19
2.1.6 Django vs Flask	20
2.2 Estructura de un proyecto Django	21
2.2.1 Arquitectura Modelo-Vista-Plantilla (MVT)	22
2.2.2 Sistema de plantillas	23
2.3 Django REST Framework	24
2.4 Metodología de desarrollo de software	24
2.4.1 Metodologías tradicionales.....	25

2.4.2	Metodologías ágiles	25
2.4.3	Kanban.....	25
2.5	Modelo de calidad	25
2.5.1	FURPS.....	26
2.6	JMeter.....	26
CAPÍTULO III. METODOLOGIA		27
3.1	Tipo de investigación	27
3.1.1	Según la fuente de la investigación	27
3.1.2	Según el objeto de estudio.....	27
3.2	Técnicas de recolección de datos.....	27
3.3	Población de estudio y tamaño de la muestra.....	27
3.4	Métodos de análisis, y procesamiento de datos.....	27
3.5	Identificación de variables.....	27
3.5.1	Variable independiente	27
3.5.2	Variable dependiente	28
3.6	Operacionalización de variables.....	28
3.7	Desarrollo	29
3.7.1	Análisis de requerimientos	29
3.7.2	Diagramas de casos de uso	31
3.7.3	Diseño de arquitectura	32
3.7.4	Modelado.....	35
3.7.5	Desarrollo de la aplicación web aplicando Kanban.....	39
3.7.6	Planificación de pruebas.....	45
3.7.7	Ejecución de las pruebas.....	45
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN		47
4.1	Resultados.....	47
4.2	Valoración de indicadores	49
4.2.1	Tiempo de desarrollo	49
4.2.2	Número de líneas de código	49
4.2.3	Numero de funcionalidades	49
4.2.4	Eficacia	49
4.2.5	Tiempo de respuesta	50
4.2.6	Utilización de recursos	51

4.2.7	Valores obtenidos del estudio en base al modelo FURPS.....	51
4.3	Discusión	51
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES		53
5.1	Conclusiones.....	53
5.2	Recomendaciones	53
BIBLIOGRAFÍA		55
ANEXOS		58

ÍNDICE DE TABLAS

Tabla 1: Django vs Flask.....	20
Tabla 2: Parámetros y valores del modelo FURPS	26
Tabla 3: Herramienta para la recolección de datos	27
Tabla 4: Operacionalización de variables	28
Tabla 5: Requisitos Funcionales.....	29
Tabla 6: Requisitos no Funcionales.....	30
Tabla 7: Diccionario de datos.....	36
Tabla 8: Tabla de medición de JMeter	45
Tabla 9: Cantidad de procedimientos realizados.....	48
Tabla 10: Requerimientos solicitados a la aplicación web.....	50
Tabla 11: Utilización de recursos	51
Tabla 12: Valores obtenidos del estudio en base al modelo FURPS	51

ÍNDICE DE FIGURAS

Figura 1: ORM de Django.....	18
Figura 2: Django estructura.....	22
Figura 3: Modelo-Vista-Plantilla	23
Figura 4: Django REST Framework	24
Figura 5: Flujo de la aplicación.....	29
Figura 6: Caso de uso de aplicación web	31
Figura 7: Caso de uso administrador.....	31
Figura 8: Caso de uso operario.....	32
Figura 9: Caso de uso invitado/turista.....	32
Figura 10: Vista conceptual.....	32
Figura 11: Vista lógica	33
Figura 12: Vista física	34
Figura 13: Diagrama de clase.....	35
Figura 14: Diagrama de base de datos	36
Figura 15: Flujo de trabajo	39
Figura 16: Etapa 1 – tablero Kanban.....	40
Figura 17: Etapa 2 – tablero Kanban.....	40
Figura 18: Etapa 3 – tablero Kanban.....	41
Figura 19: Etapa 4 – tablero Kanban.....	41
Figura 20: Pantalla de inicio de sesión.....	41
Figura 21: Pantalla de ingreso de datos exógenos	42
Figura 22: Pantalla de ingreso de datos endógenos	42
Figura 23: Código de reportes exógenos.....	43
Figura 24: Código de reportes endógenos.....	43
Figura 25: Código QR.....	44
Figura 26: Visualización de reportes.....	44
Figura 27: Configuración JMeter	46
Figura 28: HTTP Request	46
Figura 29: Resumen de resultados - JMeter	47
Figura 30: Vista de árbol de resultados - JMeter	48
Figura 31: Porcentajes de resultados de evaluación inicial.....	49
Figura 32: Eficacia de la aplicación web	50
Figura 33: Tiempo de respuesta de la aplicación web	51

RESUMEN

La investigación previa sobre el framework Django representó un componente esencial en el proyecto. Proporcionó la base tecnológica necesaria para el desarrollo de la aplicación web orientada a la gestión turística sostenible comunitaria de Chimborazo. La elección de Django como marco de desarrollo se fundamentó en su reputación por su robustez, eficacia, así como su capacidad para facilitar el desarrollo rápido y seguro de aplicaciones web complejas. Además, la investigación implicó un análisis de las características y ventajas de Django que posteriormente sirvió para el desarrollo de la aplicación web. Se empleó la metodología de desarrollo Kanban para organizar y gestionar el proceso de desarrollo. La aplicación web desarrollada se destacó como una solución integral gracias al uso del framework Django, además de ser escalable permitiendo una fácil adaptación a las necesidades cambiantes de la comunidad y del sector turístico. La misma que fue sometida a una rigurosa evaluación de rendimiento utilizando la herramienta JMeter. Se analizaron métricas como eficacia, tiempo de respuesta y uso de recursos del sistema. Los resultados revelaron una eficacia del 100%, cumpliendo con los estándares del modelo de calidad FURPS. El tiempo de respuesta promedio fue de 4600 ms, por debajo del límite sugerido por el modelo FURPS. Además, el uso de recursos del sistema fue del 25%, en consonancia con las expectativas del modelo de calidad. Esto confirmó la eficacia y el desempeño óptimo del sistema.

Palabras claves: Aplicación Web, Desarrollo de Software, Django, Gestión Turística, Metodología Kanban, Modelo FURPS, Rendimiento, Sostenibilidad Comunitaria.

ABSTRACT

Previous research on the Django framework represented an essential component of the project. It provided the necessary technological foundation for the development of the web application aimed at sustainable community tourism management in Chimborazo. The choice of Django as the development framework was based on its reputation for robustness and efficiency and its ability to facilitate the rapid and secure development of complex web applications. Additionally, the research involved an analysis of Django's features and advantages, which subsequently served the development of the web application. The Kanban development methodology was employed to organize and manage the development process. The developed web application stood out as a comprehensive solution thanks to the use of the Django framework, in addition to being scalable, allowing for easy adaptation to the changing needs of the community and the tourism sector. It was subjected to a rigorous performance evaluation using the JMeter tool. Metrics such as efficiency, response time, and system resource usage were analysed. The results revealed 100% efficiency, meeting the standards of the FURPS quality model. The average response time was 4600 ms, below the limit suggested by the FURPS model. Additionally, system resource usage was 25%, in line with the quality model's expectations. This confirmed the system's efficacy and optimal performance.

Keywords: Web Application, Software Development, Django, Tourism Management, Kanban Methodology, FURPS Model, Performance, Community Sustainability.



Revised by
Mario N. Salazar

CAPÍTULO I. INTRODUCCION

En la era actual, las aplicaciones web han ganado terreno en el mercado de software por las ventajas que presentan y la creciente necesidad de internet en muchas operaciones, relegando a un segundo plano a las tradicionales aplicaciones de escritorio [1].

Este avance también está relacionado con la introducción de nuevos lenguajes de programación, herramientas y enfoques en la creación de aplicaciones web, aunque esto plantea inquietudes adicionales que los desarrolladores de software deben enfrentar. La gestión de la complejidad del software y la selección adecuada de las herramientas de desarrollo se vuelven cruciales para el proceso de adaptación requerido en el sistema [2].

Con el fin de optimizar los resultados es importante automatizar las operaciones y por ello la utilización de frameworks de software, permite reducir el código redundante, repetitivo y no estandarizado.

Django se conoce como un marco "baterías incluidas", que afirma tener la mayoría de las extensiones y bibliotecas necesarias para poner en marcha una aplicación genérica, permite al desarrollador dedicar más tiempo a la implementación de la lógica empresarial [3].

Esta investigación surgió de un proyecto de investigación denominada "Gestión Turística Sostenible en las Organizaciones Comunitarias de la provincia de Chimborazo". Se creó un aplicativo para el modelo de gestión turística para las comunitarias de Chimborazo, haciendo uso del Framework Django. Además, se implementó la metodología ágil de desarrollo Kanban y se aplicó el modelo de calidad FURPS, se evaluó el rendimiento del sistema.

La estructuración del trabajo de investigación es la siguiente: El Capítulo I describe el planteamiento del problema. El Capítulo II, permite conocer la base conceptual relacionada con la temática abordada. El Capítulo III explica la metodología utilizada y las herramientas. El Capítulo IV muestra los resultados obtenidos al evaluar el rendimiento del aplicativo mediante el modelo de calidad FURPS. Finalmente, dentro del Capítulo V se exponen las conclusiones que se obtuvo de esta investigación y las recomendaciones.

1.1 Planteamiento y justificación de la investigación

La investigación se centró en abordar los desafíos organizativos y operativos en el turismo de las organizaciones comunitarias en Chimborazo. La ausencia de un sistema automatizado para la gestión turística sostenible generó la necesidad de implementar un aplicativo web.

Para alcanzar los objetivos de la investigación se implementó el aplicativo web para el modelo de gestión turística sostenible comunitaria utilizando Django, es un Framework que permite usar Python con sus diferentes funciones e incluso integrar una interfaz. También se empleó la metodología de desarrollo Kanban, finalmente se evaluó el rendimiento de la aplicación utilizando el modelo de calidad FURPS.

1.2 Formulación del problema

¿Cómo el uso del Framework Django incidirá en el rendimiento de la aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo?

1.3 Objetivos

1.3.1 General

Implementar una aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo utilizando Django.

1.3.2 Específicos

- Investigar el Framework Django para el desarrollo de software.
- Desarrollar la aplicación web para el modelo de gestión turística con el Framework Django.
- Evaluar el rendimiento de la aplicación web utilizando el modelo de calidad FURPS

CAPÍTULO II. MARCO TEÓRICO

2.1 Django

Django, un robusto marco web creado en el lenguaje de programación Python, destaca por su enfoque integral conocido como "baterías incluidas". Este enfoque se traduce en la provisión de una extensa gama de herramientas y funciones listas para su implementación inmediata, lo que simplifica significativamente el proceso de desarrollo web. Entre las características clave se encuentran el soporte integrado para tareas comunes como controles de accesos para los usuarios, pruebas, modelos de bases de datos, formularios, gestión en las rutas de URL, plantillas y una interfaz de administración.

Adicionalmente, Django promueve una estructura de proyecto coherente que facilita la adhesión a las mejores prácticas de desarrollo, fomentando la creación de aplicaciones web de manera eficiente y organizada. La amplia adopción de Django y la existencia de una comunidad activa compuesta por millones de desarrolladores respaldan su eficacia y confiabilidad en la creación eficiente de aplicaciones web sólidas. Este marco no solo simplifica el desarrollo, sino que también se posiciona como una herramienta fundamental para aquellos que buscan una solución completa y poderosa en el mundo del desarrollo web [4].

2.1.1 Características de Django

Django es conocido como un framework que está diseñado para crear aplicativos webs con el lenguaje de Python, ofrece características predefinidas que simplifican significativamente el proceso de creación. Al integrar funcionalidades listas para usar, Django elimina la necesidad de escribir código desde cero y reduciendo el tiempo de búsquedas con posibles errores en el framework. El uso de Django agiliza el desarrollo de sistemas de información web al dar una solución rápida, segura, escalable y de fácil mantenimiento [5].

Este marco se destaca por su estructura de diseño MVT (Modelo, Vista, Plantilla), que promueve un desarrollo de aplicaciones ágil y eficiente. En Django, el modelo actúa como la base de datos, la vista controla la funcionalidad y la plantilla se encarga de la interacción del usuario, proporcionando la interfaz de comunicación. Es notable por su capacidad para manejar cambios en los registros con comandos como 'python manage.py make migrations'. La eficiencia y rapidez en el desarrollo hacen de Django una opción muy popular en el mercado actual para la creación rápida de diversos tipos de aplicaciones web [6].

De acuerdo con Espinosa [7], el Framework Django se destaca como la opción óptima para el desarrollo de aplicaciones web, considerando criterios específicos como Rendimiento, Usabilidad, Portabilidad y Seguridad.

2.1.2 ORM de Django

En la Figura 1, se representa el funcionamiento del ORM para la manipulación de los registros en el framework Django.

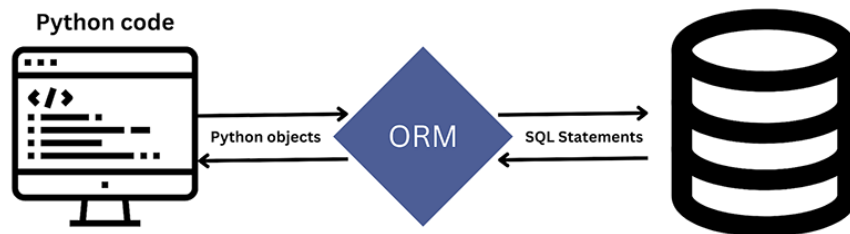


Figura 1: ORM de Django

Fuente:[8]

El componente de Mapeo Objeto-Relacional (ORM) en Django es un elemento indispensable que facilita la creación de aplicativos modernos. El ORM facilita a los desarrolladores interactuar con los registros mediante un enfoque orientado a objetos, en lugar de escribir consultas SQL directamente. Esto simplifica significativamente el proceso de desarrollo, ya que los desarrolladores pueden trabajar con modelos de datos en Python, lo que mejora la eficiencia y la legibilidad del código. El ORM de Django se destaca por su capacidad para abstraer las complejidades de las relaciones con los registros, permitiendo a los desarrolladores centrarse más en la lógica de la aplicación y menos en los detalles específicos del almacenamiento de los datos [9].

El ORM de Django se basa en el principio "Don't Repeat Yourself". Este método enfoca la reutilización tanto del código y reduce la redundancia, teniendo como resultado un desarrollo mucho más eficiente y mejor mantenible. Además, el ORM de Django es altamente flexible, permitiendo a los desarrolladores adaptarlo a diversas necesidades y tipos de proyectos. Su integración con el Django REST Framework amplía aún más sus capacidades, facilitando la creación de aplicaciones web modernas y escalables que pueden manejar eficientemente tanto las operaciones de backend como las necesidades de las interfaces de usuario [10].

Una de las fortalezas clave del ORM de Django es su enfoque en la seguridad. Proporciona protección contra vulnerabilidades comunes de bases de datos, como las inyecciones SQL, mediante el uso de consultas parametrizadas y la validación automática de datos. Esto asegura que el código generado por el ORM sea seguro por defecto, reduciendo el riesgo de errores de seguridad comunes. Además, el ORM es altamente personalizable y extensible, lo que permite a los desarrolladores adaptarlo a las necesidades específicas de su aplicación, lo que lo hace una herramienta mucho más versátil y poderosa para cualquier desarrollador de Django [11].

2.1.3 Mapeo de modelos

El mapeo de modelos en Django es un componente fundamental del framework que facilita la relación entre los registros y la lógica del aplicativo. Django utiliza un patrón Modelo Vista Controlador que divide claramente el Modelo en el cual están los datos, la interfaz de usuario donde se encuentran las vistas y el controlador donde se encuentra la lógica. En este contexto, los modelos en Django representan la estructura de datos y definen cómo se almacenan y validan los registros. Utilizando el ORM (Object-Relational Mapping) de Django, los modelos permiten a los desarrolladores trabajar con objetos Python en lugar de escribir consultas SQL directamente, simplifica significativamente el desarrollo y mantenimiento de aplicaciones web [12].

Los modelos de Django no solo encapsulan datos y validaciones, sino que también ofrecen una variedad de funcionalidades para el manejo de registros. Esto incluye consultas en los modelos, tanto en administradores que permiten realizar acciones en los registros de forma eficiente y también segura. Los modelos en Django son altamente personalizables y pueden ajustarse a diversas necesidades de desarrollo, en aplicaciones simples hasta sistemas complejos. Además, el mapeo de modelos en Django garantiza la integridad de los datos y facilita la implementación de prácticas de desarrollo como las migraciones de base de datos, que son esenciales para el mantenimiento y la evolución de las aplicaciones [13].

2.1.4 Migraciones en Django

Es un aspecto crucial del desarrollo web que facilita la gestión de cambios en los modelos de base de datos y su correspondiente esquema. Las migraciones en Django permiten a los desarrolladores rastrear y aplicar cambios en los registros de forma controlada y eficiente. Este proceso automatizado ayuda a la consistencia de la información y asegura que las alteraciones en los modelos de Django se reflejen adecuadamente en la base de datos subyacente. Las migraciones son fundamentales para el desarrollo ágil y sostenible de aplicaciones, proporcionando una forma robusta y segura de manejar la evolución de la estructura de la información en todo su ciclo de vida del proyecto [14].

2.1.5 Administración, seguridad y autenticación

Django como un framework web seguro, implementa diversas estrategias para resguardar la información de los usuarios y de las aplicaciones web. Entre estas estrategias, se encuentran el uso eficaz del ORM de Django para prevenir ataques de inyección SQL, la implementación de tokens CSRF para proteger contra la falsificación de solicitudes entre sitios y la protección XSS para evitar ataques de scripting entre sitios. Estas herramientas y técnicas demuestran el compromiso de Django con la seguridad robusta y la protección de datos, haciendo de Django una opción confiable para el desarrollo de aplicativos que sean seguros [15].

Dentro de la administración la interfaz de Django es una herramienta crucial para la gestión de contenido en sitios web. Esta interfaz, diseñada para ser utilizada por administradores de confianza, facilita la adición, edición y eliminación de contenido del sitio de manera eficiente y segura. La interfaz de administración de Django se destaca por su facilidad de uso y su capacidad para ser personalizada según las necesidades específicas del proyecto, esto la transforma en una herramienta efectiva para administrar sitios web. Este enfoque en la administración eficiente y segura es un aspecto clave que contribuye a la popularidad y eficacia de Django como framework de desarrollo web [16].

2.1.6 Django vs Flask

Django sobresale a diferencia de Flask, siendo este más adecuado para proyectos más pequeños y sencillos debido a su naturaleza minimalista y modular, Django ofrece un conjunto de características más completo y un enfoque "baterías incluidas". Esto significa que Django viene con una amplia gama de funcionalidades integradas, como un ORM (Object-Relational Mapping) robusto, un sistema de administración listo para usar y una estructura de proyecto bien definida. Estas características hacen de Django una opción preferida para proyectos a gran escala que requieren una arquitectura más sólida y una gestión de datos más compleja, proporcionando una base más sólida y menos dependencia de extensiones de terceros en comparación con Flask [17].

La Tabla 1 presenta una comparación de los frameworks del lenguaje Python más relevantes:

Tabla 1: Django vs Flask

Características	Django	Flask
Tipo de Marco	Marco de aplicación web completo y robusto.	Microframework orientado a resultados rápidos y flexibilidad.
Patrón de Diseño	Sigue el principio MVC (Modelo-Vista-Controlador), con una estructura y reglas definidas.	No impone un patrón de diseño específico, proporcionando flexibilidad en la estructura de la aplicación.
Manejo de Solicitudes y Enrutamiento	Gestiona las solicitudes de manera explícita, requiriendo que el objeto de solicitud sea proporcionado a la función o clase de vista.	Maneja el contexto de solicitud automáticamente, sin necesidad de pasar explícitamente el objeto de solicitud.
Base de Datos y Migraciones	Soporta varios sistemas de registros y ofrece un ORM maduro con migraciones fáciles.	Ofrece libertad en selección de la base de datos, pero no proporciona soporte integrado para migraciones.

Seguridad	Incluye protección integrada contra problemas comunes de seguridad web como CSRF, XSS, inyección SQL, etc.	Requiere que los desarrolladores implementen o utilicen extensiones para manejar aspectos de seguridad como CSRF.
Configuraciones	Posee configuraciones extensivas y separadas para diferentes entornos (desarrollo, prueba, producción).	Permite configuraciones flexibles, generalmente gestionadas a través de variables de entorno.
Despliegue	Proporciona un marco definido para el despliegue, incluyendo la configuración de Heroku y otros servicios en la nube.	Facilita el despliegue con Heroku y otros servicios en la nube, pero requiere configuración adicional.
Manejo de Errores	Usa las excepciones de Python para el manejo de errores, con clases de excepción específicas de Django.	Utiliza técnicas de manejo de errores de Python, con posibilidad de ligar excepciones a manejadores de errores personalizados.
Caché	Soporta varios sistemas de caché como Memcached, caché de base de datos y caché de sistema de archivos.	Admite varios sistemas de caché, pero no los utiliza por defecto en el proyecto del estudio.
Flexibilidad y Extensibilidad	Ofrece muchas características integradas, pero puede ser menos flexible para cambios significativos o aplicaciones pequeñas.	Alta flexibilidad para elegir bibliotecas y extensiones según sea necesario, ideal para desarrollo rápido y aplicaciones pequeñas.
Soporte al Desarrollador y Documentación	Documentación extensa y detallada con una gran comunidad de soporte.	Documentación extensa con ejemplos y detalles de implementación, soporte de una amplia comunidad.

Fuente: [18]

2.2 Estructura de un proyecto Django

La estructura del framework Django se compone de varios elementos esenciales que colaboran para asegurar su funcionalidad y eficiencia. Inicialmente, el cliente, comúnmente un navegador, solicita información al servidor mediante el protocolo HTTP. Esta petición

viaja a través de Internet por una red TCP/IP hasta alcanzar el HttpServer, responsable de atender y gestionar estas solicitudes. En este marco, se utilizan servidores web como Nginx, Apache e IIS. Un aspecto vital de la estructura de Django es el uso del WSGI (Interfaz de Pasarela de Servidor Web), que facilita la comunicación entre los servidores web y las aplicaciones. Además, Django emplea Middleware, un componente de software en Python configurado en los ajustes, que se tiene que ajustar a las solicitudes y respuestas durante la interacción entre un cliente y un servidor [19].

La Figura 2, visualiza la estructura del proyecto del framework Django

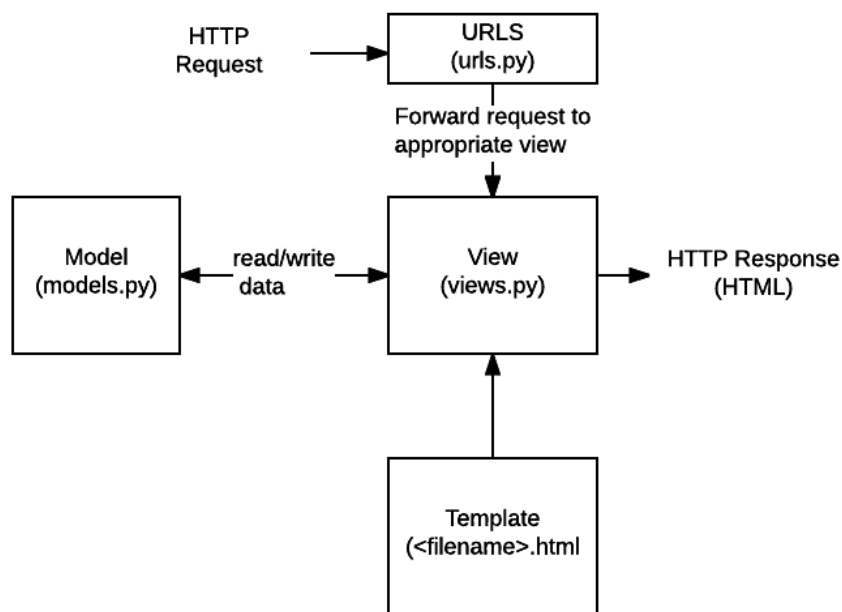


Figura 2: Django estructura

Fuente: [20]

2.2.1 Arquitectura Modelo-Vista-Plantilla (MVT)

La Figura 3, modela el diseño de Modelo Vista Plantilla

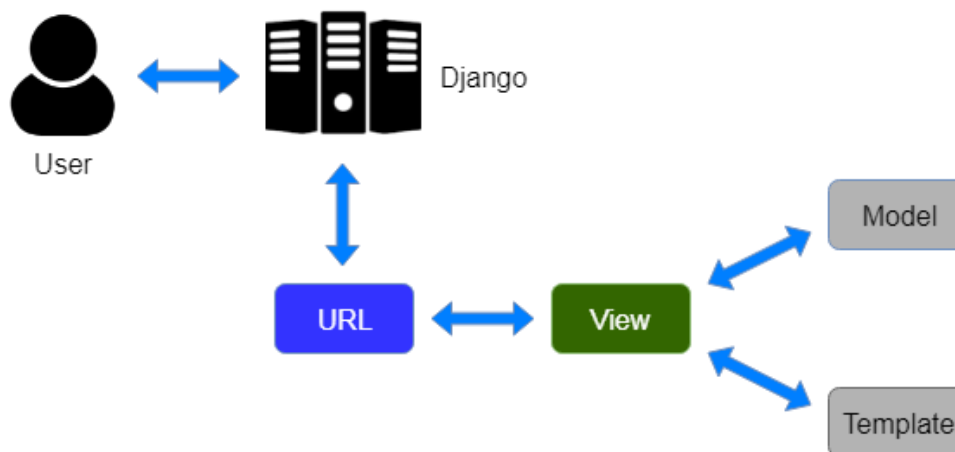


Figura 3: Modelo-Vista-Plantilla

Fuente: [21]

En el marco de trabajo Django, se adopta el diseño de modelo vista plantilla, el cual es una adaptación del ya conocido patrón modelo vista controlador. Esta arquitectura es crucial en la estructuración y organización de proyectos dentro de Django. En el MVT, el 'Modelo' es quien realiza la gestión de los registros, lo que implica la manipulación y el acceso a los datos. La 'Vista' actúa como intermediario, procesando los datos obtenidos del Modelo y transfiriéndolos a las Plantillas. Finalmente, la 'Plantilla' se encarga de presentar estos datos en un formato accesible y comprensible para el usuario final. Esta separación de responsabilidades es fundamental debido a que facilita al gestionar los proyectos complejos al desacoplar los componentes de procesamiento de datos y presentación, mejorando así la mantenibilidad y escalabilidad de las aplicaciones [22].

2.2.2 Sistema de plantillas

La importancia de separar la presentación del código HTML de la lógica de programación en Python es una práctica esencial en el desarrollo web. El sistema de plantillas de Django es destacado por su capacidad para resolver este desafío, permitiendo a los desarrolladores escribir código HTML de manera independiente y más organizada. Este enfoque no solo mejora la legibilidad y el mantenimiento del código, sino que también facilita la colaboración entre diseñadores y desarrolladores. El sistema de plantillas de Django se presenta como una solución eficaz que evita la mezcla de código HTML con Python, resulta en un desarrollo web más estructurado y eficiente [23].

Al proporcionar una plantilla estándar, los desarrolladores pueden enfocarse en aspectos más complejos y específicos de sus proyectos, reduciendo el tiempo dedicado a configuraciones generales. Este enfoque demuestra cómo el sistema de plantillas de Django puede ser optimizado para mejorar la eficiencia en las etapas tempranas del desarrollo de aplicaciones web [24].

2.3 Django REST Framework

Es un conjunto de instrumentos que facilita la generación de API web y se integra perfectamente con Django, facilitando la implementación de las funcionalidades esenciales en un api rest. En sus diversos elementos se incluyen las vistas en base a clases, al igual que sus conjuntos de vistas y también los serializadores [25].

En la Figura 4, se representa cómo es el funcionamiento de Django REST Framework

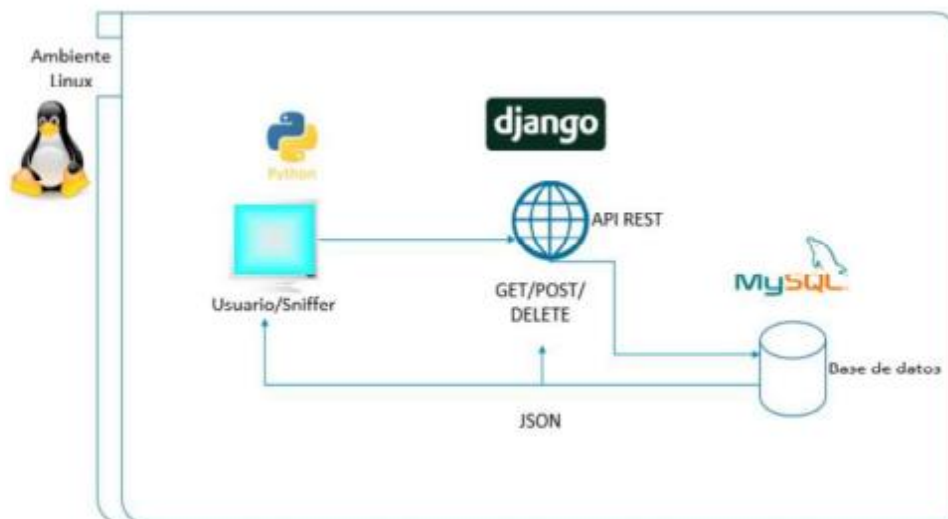


Figura 4: Django REST Framework

Fuente: [26]

Django REST Framework se destaca por su facilidad de uso, su capacidad para manejar la serialización de datos y su soporte para autenticación y permisos. Este enfoque facilita la creación de servicios web que son tanto escalables como seguros, haciendo de Django REST Framework una herramienta esencial para los desarrolladores que buscan implementar funcionalidades REST en sus aplicaciones Django [27].

El framework proporciona un conjunto robusto de herramientas para construir APIs web, destacando por su capacidad para simplificar y acelerar el desarrollo de APIs complejas y de alto rendimiento. Django REST Framework facilita la integración de datos entre el backend y el frontend, y su arquitectura modular permite a los desarrolladores personalizar y extender la funcionalidad según sea necesario, lo que lo hace una elección perfecta para proyectos que necesitan una comunicación sofisticada entre el usuario y el servidor [28].

2.4 Metodología de desarrollo de software

La metodología abarca una serie de procedimientos lógicos destinados a alcanzar un objetivo específico que requiere habilidades y conocimientos especializados. Es una fase particular dentro de un trabajo o proyecto, que parte de un fundamento teórico y desemboca en la

selección de técnicas concretas o métodos para dirigir el desarrollo hacia los objetivos establecidos. Consiste en la agrupación de procedimientos aplicados en una actividad específica con el propósito de afirmar y mejorar su eficiencia. La metodología establece los pasos a seguir y la manera de ejecutarlos para llevar a cabo exitosamente una tarea [29].

2.4.1 Metodologías tradicionales

A este tipo de enfoques también se les denomina modelos de proceso prescriptivo; fueron diseñados para establecer una estructura y un orden concretos durante el desarrollo de software. A lo largo del tiempo, se ha demostrado que estas metodologías han sido efectivas al proporcionar una distribución útil de tareas y un mapa eficiente. Debido a su naturaleza estructurada, estas metodologías no permiten retroceder a la fase anterior y carecen de adaptabilidad al cambio, es decir, son poco flexibles. En consecuencia, las actividades del flujo de proceso durante el desarrollo del proyecto quedan predefinidas [30].

2.4.2 Metodologías ágiles

Durante una reunión llevada a cabo en Utah, Estados Unidos, se introdujo el término "ágil" en el ámbito del desarrollo de software. Esta reunión contó con la participación de 17 expertos, incluyendo pioneros e impulsores de las metodologías de software. Su objetivo era establecer los valores y también los principios que permitieran a los equipos el desarrollo de software de manera rápida y adaptable a los cambios, proporcionando así alternativas las cuales se caracterizan por su rigidez y enfoque en la documentación. Este encuentro fue el catalizador para la creación de The Agile Alliance, una organización sin ánimo de lucro que se dedica a incentivar los fundamentos del desarrollo en las metodologías ágiles. Todo comenzó con la redacción del Manifiesto Ágil, que destaca la importancia de valorar a las personas y sus interacciones, así como la priorización del software funcional sobre la documentación exhaustiva[31].

2.4.3 Kanban

Taiichi Onho, de Toyota, empleó el término japonés Kanban para describir un sistema visual aplicado en las etapas de producción. Este sistema coordina la entrega de cada módulo en una serie de montaje según la necesidad, con el propósito de prevenir la excesiva producción y el almacenaje no requerido de productos. Dentro del marco de gestión ágil de proyectos, Kanban se refiere a métodos visuales destinados a aumentar la eficiencia en la realización de las actividades de un proyecto. Las reglas fundamentales abarcan la representación visual del flujo de trabajo, la fijación de un límite para el trabajo en progreso y la medición del tiempo requerido para finalizar una tarea [32].

2.5 Modelo de calidad

Los modelos de calidad en específico los modelos de calidad de software son marcos que proporcionan directrices, guías y buenas prácticas para evaluar y mejorar tanto el producto

como los procesos de desarrollo de cada etapa del software. Estos modelos son esenciales para poder garantizar que el software cumpla con los estándares y las expectativas del cliente, destacando la complacencia del cliente como indicador clave de calidad [33].

Un modelo de calidad de software abarca una variedad de atributos y elementos de calidad, cada uno de ellos acompañado por criterios específicos que definen su nivel de excelencia. Estos aspectos proporcionan la base para evaluar y Gestionar exhaustivamente la calidad de un producto de software. Un ejemplo de modelo de calidad de software es el FURPS se rige como una herramienta valiosa y completa para garantizar que los productos de software no solo cumplan con las expectativas y normativas, sino que también superen los estándares establecidos.

2.5.1 FURPS

El modelo de calidad FURPS, desarrollado por Hewlett-Packard en 1987, aborda diversos aspectos para evaluar la calidad del software, incluyendo el rendimiento. Este modelo destaca métricas esenciales, como el tiempo en el que tarda en dar una respuesta, el consumo que se tiene de los recursos y también de la eficacia, para determinar qué tecnología ofrece características superiores. La medición del rendimiento web se enfoca en el intervalo temporal desde el acceso a un sitio o aplicación hasta su presentación completa, y durante este proceso mediante el Protocolo de Transferencia de Hipertexto, los recursos del servidor que pueden verse afectados [34].

El modelo de calidad FURPS se detalla en la Tabla 2, a continuación:

Tabla 2: Parámetros y valores del modelo FURPS

Parámetros	Modelo FURPS
Eficacia	95%
Tiempo de respuesta	5s
Uso de recursos	25%

Fuente: [35]

2.6 JMeter

Apache JMeter, desarrollado completamente en Java, es un marco de código abierto dedicado a realizar pruebas de carga de rendimiento en aplicaciones. Destacado como uno de los softwares más populares en este ámbito, se distingue por ofrecer una documentación exhaustiva. Su flexibilidad es ampliamente reconocida, ya que cuenta con una amplia gama de módulos y configuraciones personalizables creadas por la comunidad de usuarios. Al igual que Robot Framework Swarm, cuenta con una interfaz gráfica y puede ser ejecutado mediante línea de comandos. Además, su versatilidad se extiende a la capacidad de realizar pruebas de carga en una variedad de protocolos y entornos, lo que lo convierte en una herramienta integral para la evaluación del rendimiento de aplicaciones web [36].

CAPÍTULO III. METODOLOGIA

El proyecto de investigación se enfocó en realizar un análisis cuantitativo, con el propósito de evaluar el rendimiento utilizando el modelo de calidad de FURPS.

3.1 Tipo de investigación

3.1.1 Según la fuente de la investigación

- Investigación documental: Se realizó mediante el análisis de una variedad extensa de documentos, tales como libros, artículos y ensayos, con el fin de examinar el uso de Django en el desarrollo web, siendo fundamental como base para el proyecto.

3.1.2 Según el objeto de estudio

- Investigación aplicada: Esta investigación se caracteriza por su enfoque aplicado, al centrarse en el aplicativo para el modelo de gestión turística sostenible comunitaria de Chimborazo, empleando Django. Logró integrar la teoría con la práctica para una implementación efectiva.
- Investigación descriptiva: Esta investigación adoptó una naturaleza descriptiva, con el propósito de analizar y exponer los resultados derivados de la evaluación del rendimiento de la aplicación web utilizando el modelo de calidad FURPS.

3.2 Técnicas de recolección de datos

La herramienta empleada para recopilar datos en este estudio se describe en la Tabla 3, a continuación:

Tabla 3: Herramienta para la recolección de datos

Herramienta
JMeter

3.3 Población de estudio y tamaño de la muestra

La evaluación del rendimiento de la aplicación web se realizó utilizando la herramienta JMeter, por lo tanto, la población de análisis es considerada como infinita.

3.4 Métodos de análisis, y procesamiento de datos.

Las mediciones de rendimiento se llevaron a cabo mediante la aplicación JMeter, que generó resultados numéricos relacionados con la eficacia, tiempos de respuesta y uso de recursos.

3.5 Identificación de variables

3.5.1 Variable independiente

Framework Django

3.5.2 Variable dependiente

Rendimiento de la aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo.

3.6 Operacionalización de variables

La operacionalización de variables de esta investigación se detalla en la Tabla 4, a continuación:

Tabla 4: Operacionalización de variables

PROBLEMA	TEMA	OBJETIVOS	VARIABLES	CONCEPTUALIZACION	DIMENSION	INDICADORES
¿Cómo el uso del framework Django incidirá en el rendimiento de la aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo?	Aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo utilizando Django	General	Independiente	Conjunto de herramientas y librerías de código abierto en Python diseñado para simplificar y acelerar el desarrollo de aplicaciones web	Framework	- Tiempo de Desarrollo - Número de Líneas de Código - Número de funcionalidades
		Específicos	Dependiente	Se determina al conjunto de aspectos que deben ser considerados para el desarrollo del modelo de gestión turística sostenible comunitaria de Chimborazo dentro de la Aplicación Web		Rendimiento
		<ul style="list-style-type: none"> Investigar el framework django para el desarrollo de software. Desarrollar la aplicación web para el modelo de gestión turística con el framework django. Evaluar el rendimiento de la aplicación web utilizando 	Rendimiento de la aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo			

el modelo de calidad FURPS.

3.7 Desarrollo

3.7.1 Análisis de requerimientos

En la Figura 5, se aprecia la representación de forma comprensible el flujo del sistema.

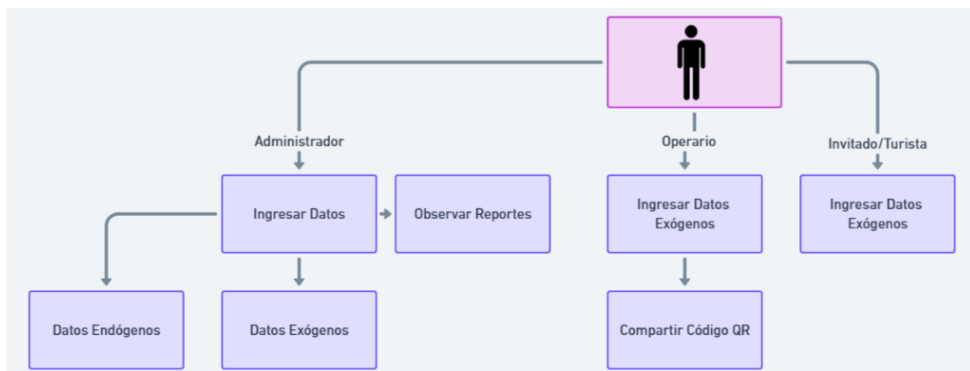


Figura 5: Flujo de la aplicación

Requisitos funcionales

Se detallan en la Tabla 5, a continuación:

Tabla 5: Requisitos funcionales

Requisito	Descripción
Interfaz Intuitiva	Crear una interfaz de usuario que sea fácil de entender y utilizar para cualquier tipo de usuario.
Ingreso de Usuarios	Permitir que los usuarios se identifiquen como Administrador, Operario o Invitado/Turista.

Ingreso de Datos Endógenos (Administrador)	Permitir al Administrador ingresar datos endógenos en la aplicación.
Ingreso de Datos Exógenos (Administrador)	Permitir al Administrador ingresar datos exógenos en la aplicación.
Visualización de Reportes (Administrador)	Permitir al Administrador observar y analizar reportes generados por la aplicación, utilizando datos históricos y visualizando gráficas.
Ingreso de Datos Exógenos (Operario)	Permitir al Operario ingresar datos exógenos en la aplicación.
Compartir Código QR (Operario)	Permitir al Operario compartir un código QR con Invitados/Turistas.
Ingreso de Datos Exógenos (Invitado/Turista)	Permitir al Invitado/Turista ingresar datos exógenos en la aplicación mediante el código QR.

Requisitos no funcionales

Los requisitos no funcionales se detallan en la Tabla 6:

Tabla 6: Requisitos no funcionales

Requisito	Descripción
Seguridad	Establecer procedimientos de seguridad sólidos para salvaguardar tanto los datos endógenos y exógenos contra accesos no autorizados y brechas de seguridad.
Rendimiento	Garantizar tiempos de respuesta rápidos y un procesamiento eficiente de los datos, incluso durante picos de uso.
Disponibilidad	Mantener la aplicación accesible para los usuarios, reduciendo al mínimo las interrupciones del servicio.
Usabilidad	Elaborar una interfaz de usuario que resulte instintiva y sencilla de explorar para usuarios de todo tipo, con un diseño que sea atractivo y uniforme.
Funcionalidad	Asegurar que todas las funcionalidades requeridas en la aplicación operen de manera coherente.

3.7.2 Diagramas de casos de uso

Actores

En la Figura 6, se representa la diagramación, teniendo como actor principal al usuario y se definió tres acciones a las que el sistema debe dar respuesta.

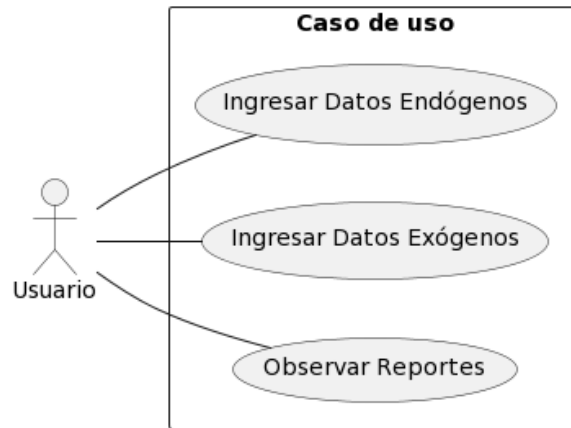


Figura 6: Caso de uso de aplicación web

Usuario - Administrador

En la Figura 7, se visualiza la diagramación, teniendo como actor principal al administrador y se definió 2 acciones a las que el sistema debe dar respuesta.

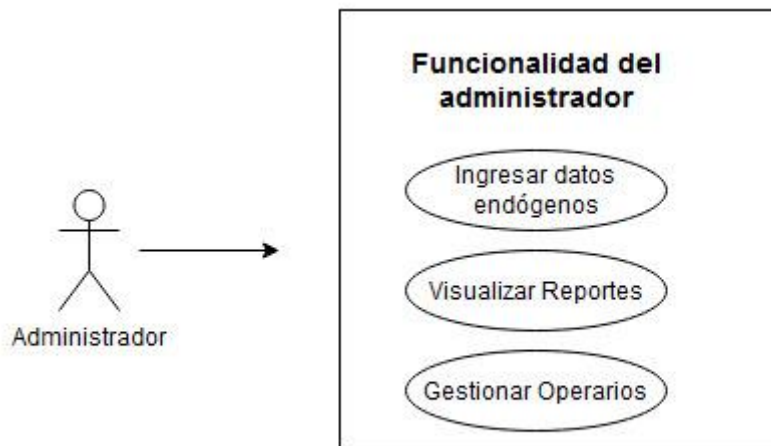


Figura 7: Caso de uso administrador

Usuario - Operario

En la Figura 8, se exhibe la diagramación, teniendo como actor principal al operario y se definió 2 acciones a las que el sistema debe dar respuesta.

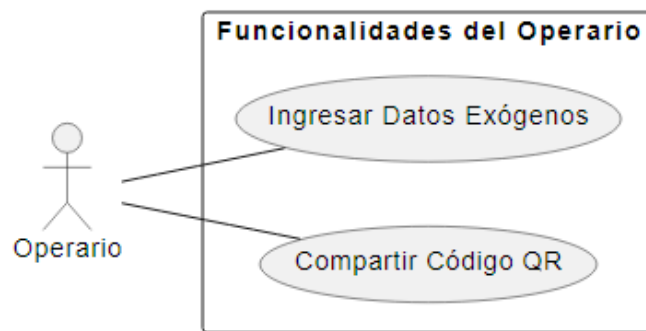


Figura 8: Caso de uso operario

Usuario – Invitado/Turista

En la Figura 9, se aprecia la diagramación, teniendo como actor principal al Invitado/Turista y se definió 1 acción a la que el sistema debe dar respuesta.

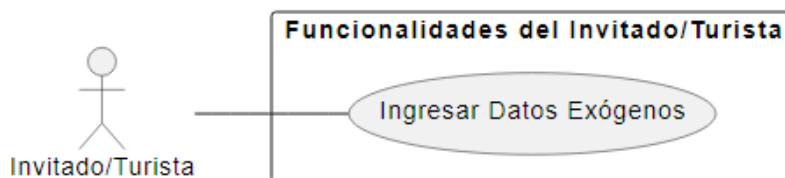


Figura 9: Caso de uso invitado/turista

3.7.3 Diseño de arquitectura

Vista conceptual

La aplicación web se estructura mediante los siguientes componentes: módulo para introducir datos externos, módulo para ingresar los datos internos, módulo para visualizar informes externos, módulo para visualizar informes internos y el módulo de código QR. En la Figura 10, se muestra una representación conceptual.

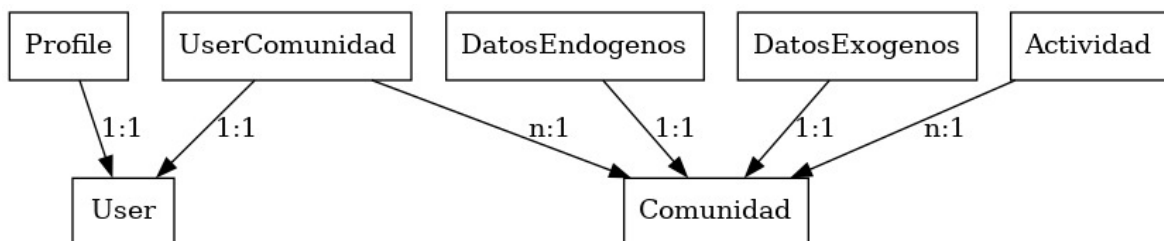


Figura 10: Vista conceptual

En este diagrama, la aplicación se compone en módulos de la siguiente manera:

- **Módulo de ingreso de datos exógenos:** referente a todos los datos exógenos que requiere el modelo.
- **Módulo de ingreso de datos endógenos:** referente a todos los datos endógenos que requiere el modelo.
- **Módulo de visualización de reportes exógenos:** referente a la generación de los reportes de todos los datos exógenos del modelo.
- **Módulo de Visualización de reportes endógenos:** referente a la generación de los reportes de todos los datos endógenos del modelo.
- **Módulo de código QR:** referente a la generación del código qr para el invitado o turista.

Vista lógica

La Figura 11, representa la vista lógica con el objetivo de abordar y representar los aspectos funcionales accesibles para los usuarios.

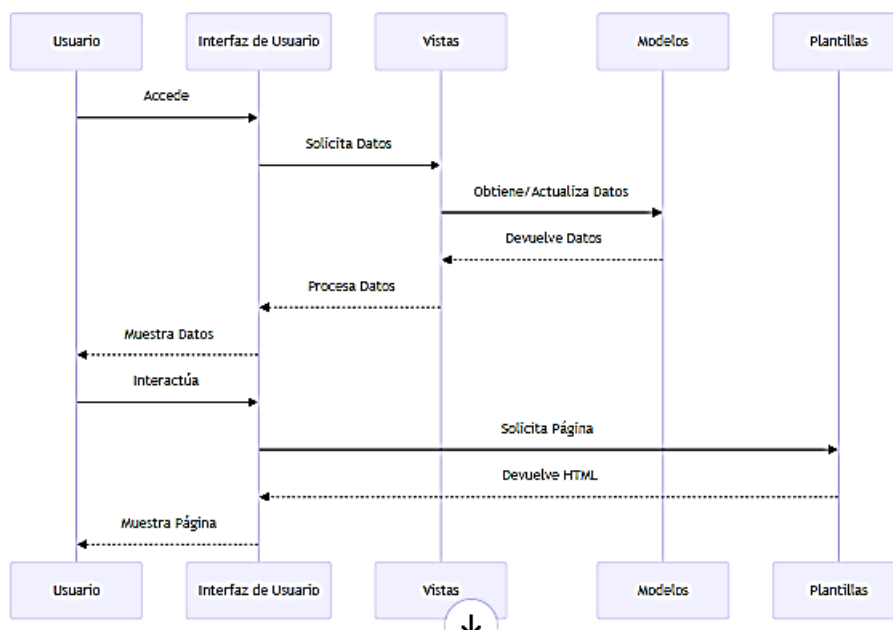


Figura 11: Vista lógica

Vista física

La Figura 12, visualiza la vista física con el objetivo de representar la gestión de recursos en el servidor.

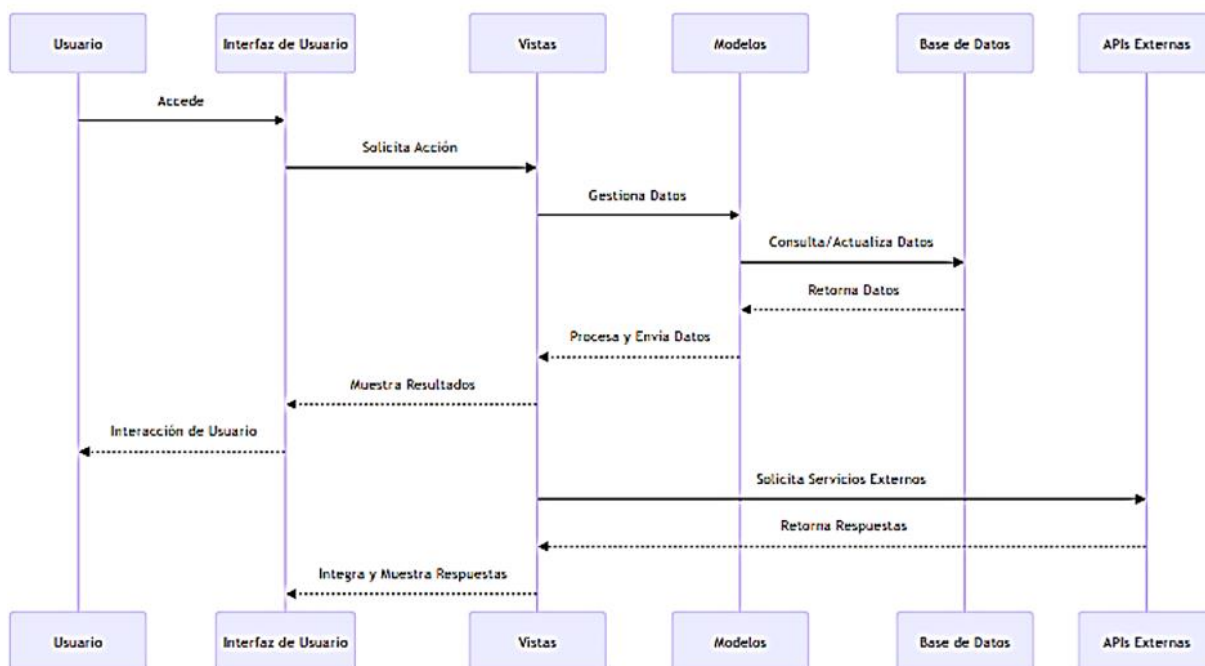


Figura 12: Vista física

3.7.4 Modelado

La Figura 13, muestra el diagrama de clase con el objetivo de representar las entidades y sus relaciones.

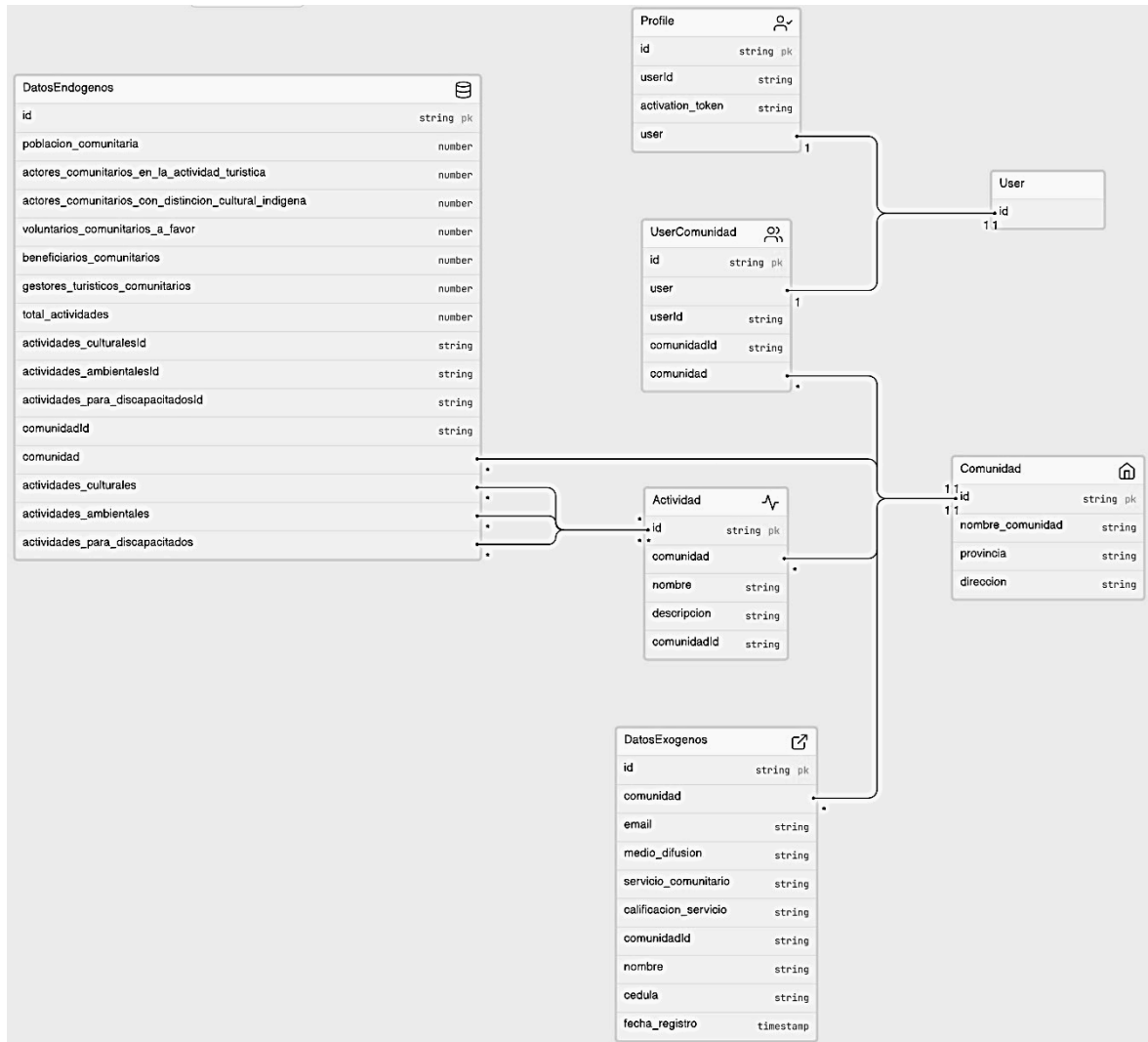


Figura 13: Diagrama de clase

Diagrama de Base de Datos

La Figura 14, muestra la diagramación de los registros con el objetivo de representar la estructura de registros obtenida de SQL Server.

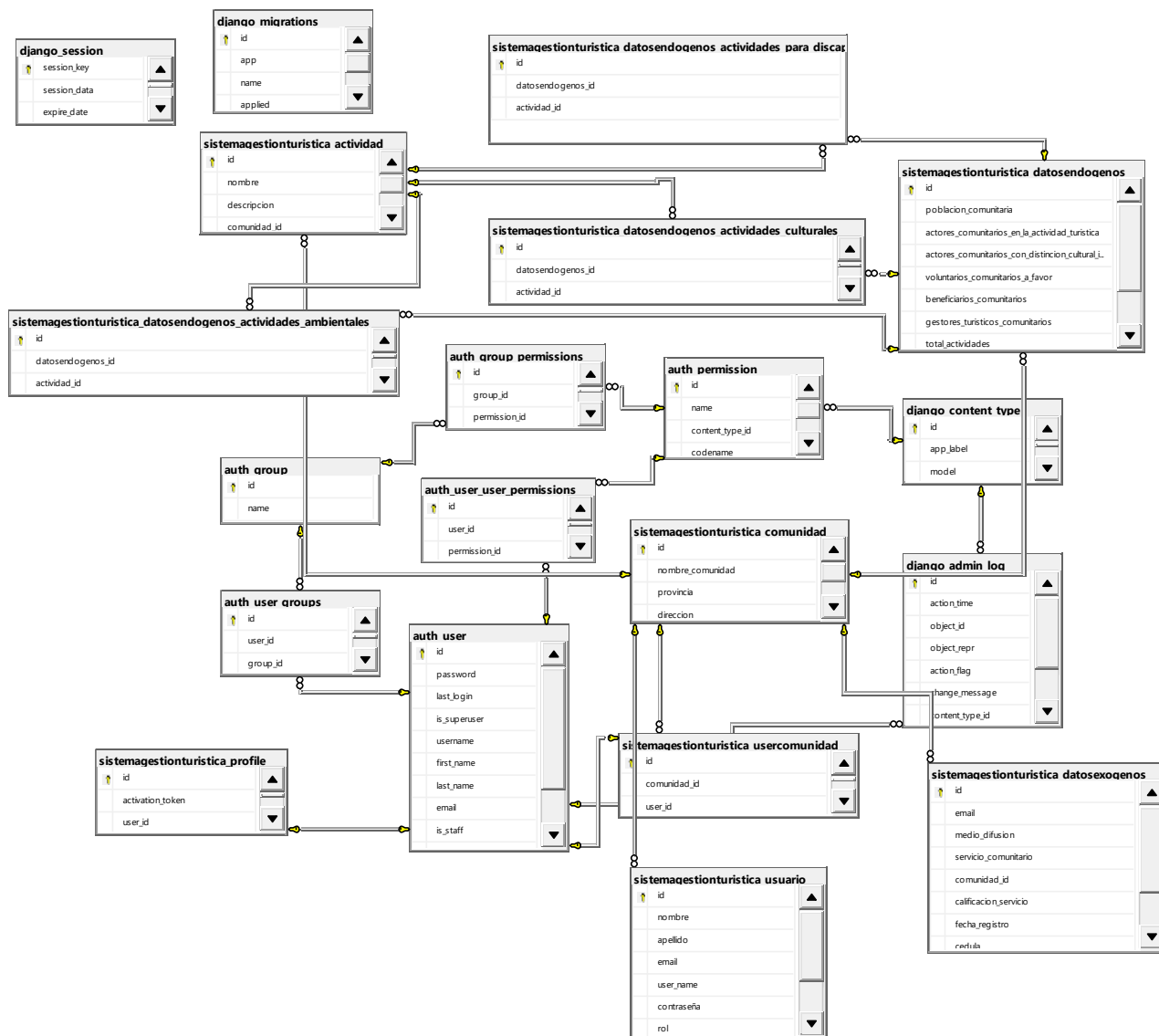


Figura 14: Diagrama

Diccionario de datos

El diccionario de datos se detalla en la Tabla 7:

Tabla 7: Diccionario de datos

Tabla	Campo	Tipo de Dato	Longitud Máxima	Descripción
Comunidad	nombre_comunidad	CharField	100	Nombre de la comunidad
	provincia	CharField	100	Provincia donde se encuentra la comunidad

	direccion	CharField	100	Dirección de la comunidad
UserComunidad	Usuario	OneToOneField	100	ID usuario
	Comunidad	ForeignKey		ID comunidad
Usuario	nombre	CharField	100	Nombre del usuario
	apellido	CharField	100	Apellido del usuario
	email	EmailField		Dirección de correo electrónico del usuario
	user_name	CharField	100	Nombre de usuario
	contraseña	CharField	100	Contraseña del usuario (considerar almacenamiento más seguro)
Actividad	Nombre	CharField	255	Nombre de la actividad
	Descripción	TextField	255	Descripción de la actividad
	Comunidad	ForeignKey		ID comunidad
DatosEndogenos	poblacion_comunitaria	IntegerField	100	Población de la comunidad
	actores_comunitarios_en_la_actividad_turistica	IntegerField	100	Número de actores comunitarios en turismo
	actores_comunitarios_con_distincion_cultural_indigena	IntegerField	100	Número de actores comunitarios con distinción cultural indígena
	voluntarios_comunitarios_a_favor	IntegerField	100	Número de voluntarios comunitarios a favor
	beneficiarios_comunitarios	IntegerField	100	Número de beneficiarios comunitarios

	gestores_turisticos_comunitarios	IntegerField	100	Número de gestores turísticos comunitarios
	total_actividades	IntegerField	100	Total de actividades en la comunidad
	actividades_culturales	IntegerField	100	Número de actividades culturales
	actividades_ambientales	IntegerField	100	Número de actividades ambientales
	actividades_para_discapacitados	IntegerField	100	Número de actividades para personas con discapacidad
	Comunidad	ForeignKey		ID comunidad
DatosExogenos	email	EmailField	100	Dirección de correo electrónico para contacto externo
	medio_difusion	CharField	100	Medio de difusión utilizado
	infraestructura	CharField	100	Infraestructura disponible
	servicio_comunitario	CharField	100	Tipo de servicio comunitario ofrecido
	Calificación servicio	CharField	100	Calificación de servicio
	Comunidad	ForeignKey		ID comunidad
	Nombre	CharField	100	Nombre del turista
	Cédula	CharField	20	Cédula del turista
	Fecha de registro	DateTimeField		Fecha de registro del turista
Perfil	Usuario	OneToOneField		ID usuario

	activation_token	CharField	64	Token de activación de la cuenta
--	------------------	-----------	----	----------------------------------

3.7.5 Desarrollo de la aplicación web aplicando Kanban

La aplicación web en el modelo de gestión turística sostenible comunitaria de Chimborazo se desarrolló utilizando la metodología de desarrollo ágil Kanban, se definió las actividades a realizar en la fase desarrollo de software, para la representación de las actividades mediante tarjetas se utilizó Microsoft Planner.

Actividades del tablero Kanban

Las cuatro actividades definidas para el tablero de Kanban fueron:

- **Pendiente:** En esta columna del tablero se ubica la etapa de desarrollo. La importancia de cada tarea es determinada por su posición, las tareas situadas en la parte superior son las de mayor prioridad.
- **En progreso:** En esta columna del tablero se encuentran las tareas que se encuentran en desarrollo.
- **En revisión:** En esta columna se encuentran las tareas que han completado la fase de progreso y están listas para ser evaluadas antes de considerarse terminadas
- **Terminado:** En esta columna se encuentran todas las tareas que han sido completadas en las fases anteriores.

Flujo de trabajo

Comienza con la identificación de las tareas pendientes, que se trasladan a en progreso durante su desarrollo. Luego pasan a: en revisión para su evaluación, y finalmente ha terminado una vez completadas. En la Figura 15, se representa el flujo de trabajo.

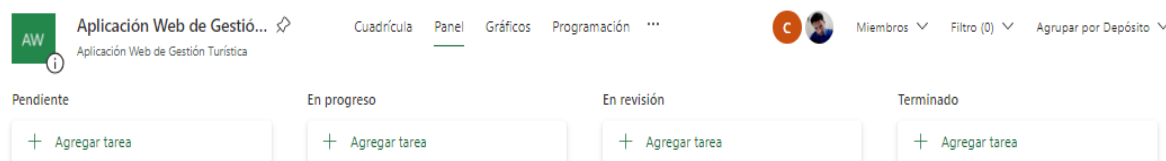


Figura 15: Flujo de trabajo

Pendiente: Colocan las tareas las cuales se organizan según su importancia y urgencia, empezando por aquellas de mayor necesidad. En la Figura 16, se visualiza las actividades pendientes.

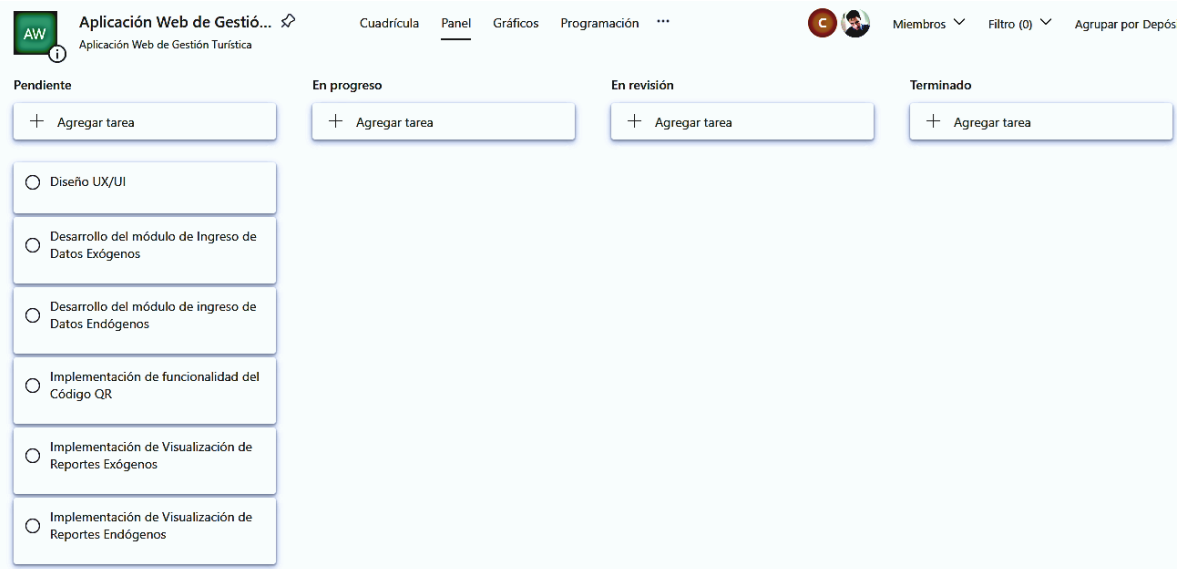


Figura 16: Etapa 1 – tablero Kanban

En progreso: Se coloca las tareas en desarrollo. La Figura 17, muestra el progreso del desarrollo.

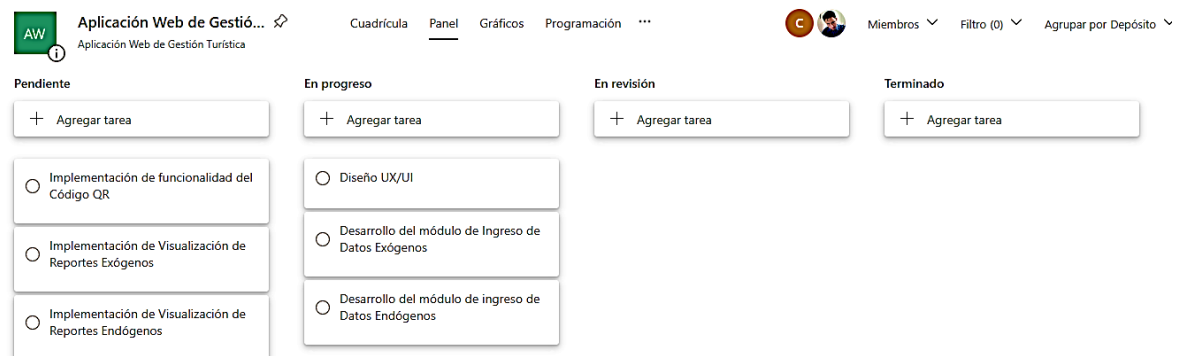


Figura 17: Etapa 2 – tablero Kanban

En Revisión: En esta columna se encuentran las tareas listas para su respectiva revisión. En la Figura 18, se detallan las actividades en revisión.

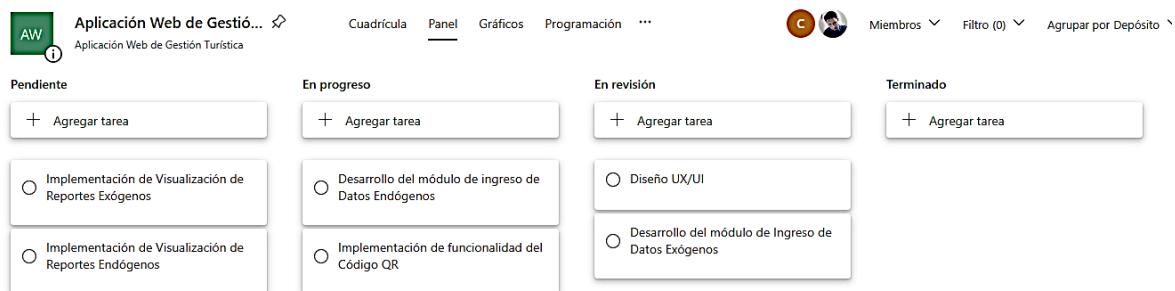


Figura 18: Etapa 3 – tablero Kanban

Terminado: Se encuentran las tareas que han pasado satisfactoriamente por las etapas previas y han cumplido con todos los requisitos necesarios. En la Figura 19, se contempla las actividades terminadas.

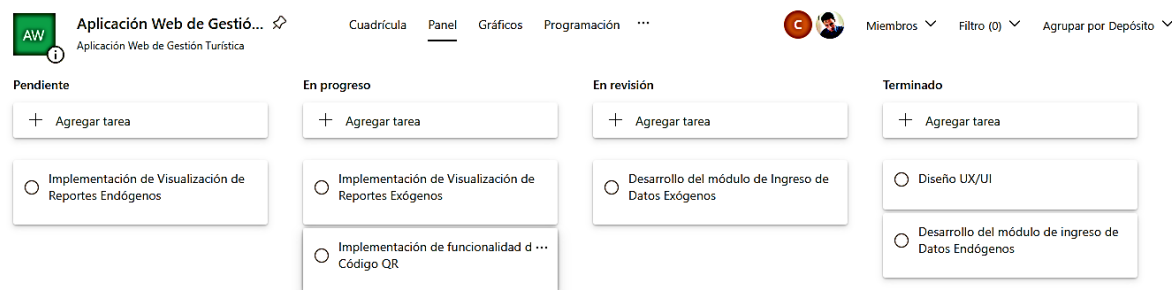


Figura 19: Etapa 4 – tablero Kanban

Interfaz del sistema de Autenticación basado en roles

La interfaz de autenticación permite el ingreso de los diferentes usuarios dependiendo de sus roles por ejemplo: Administrador, operario e invitado. La Figura 20, visualiza el inicio de sesión se basó en tres roles principales: Administrador, operario e invitado.

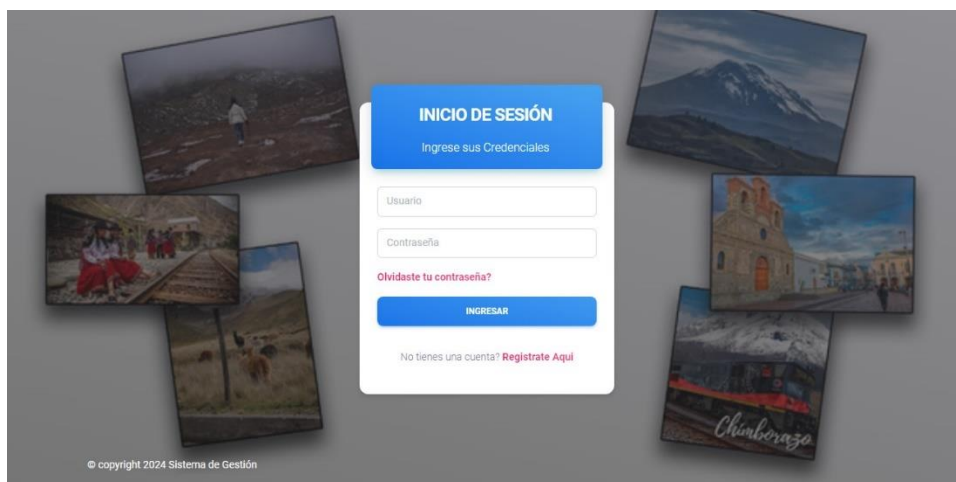


Figura 20: Pantalla de inicio de sesión

Ingreso de datos exógenos

La Figura 21, destaca la plantilla de Django para una integración directa con el servidor, lo que facilitó la inserción y actualización de datos exógenos de manera eficaz mediante el ORM de Django.



Figura 21: Pantalla de ingreso de datos exógenos

Ingreso de datos endógenos

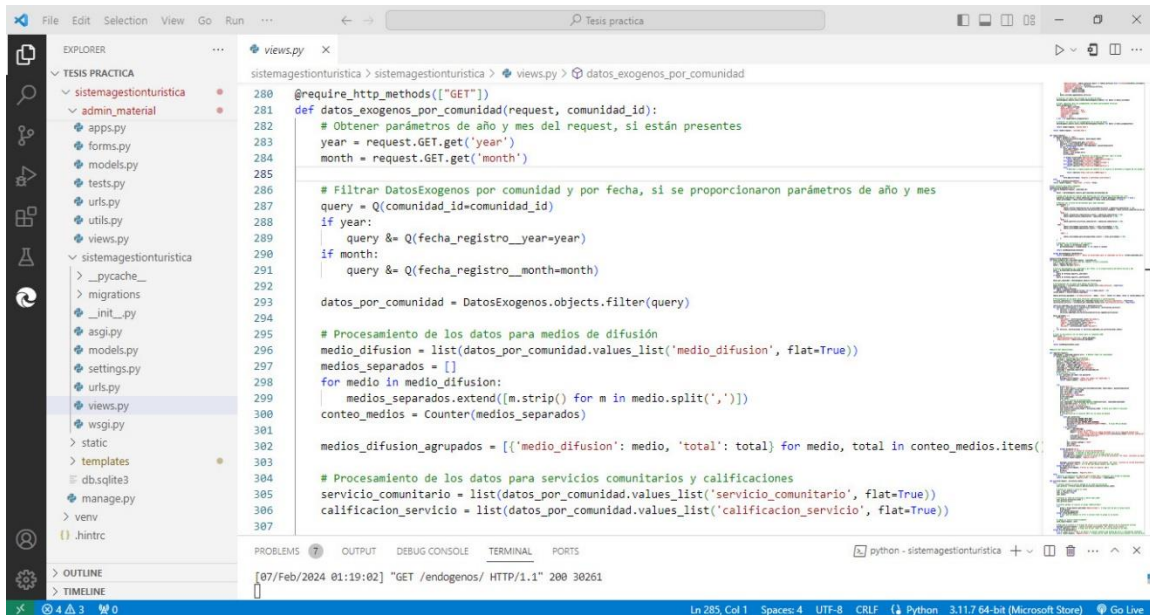
La Figura 22, destaca la plantilla de Django para integrarla directamente con el servidor, lo que permitió realizar la inserción y actualización de datos endógenos de forma eficiente a través del ORM de Django.



Figura 22: Pantalla de ingreso de datos endógenos

Implementación del reporte de datos exógenos

La Figura 23, destaca la implementación de reportes de datos exógenos que posibilitó el acceso a los datos almacenados mediante el ORM de Django.

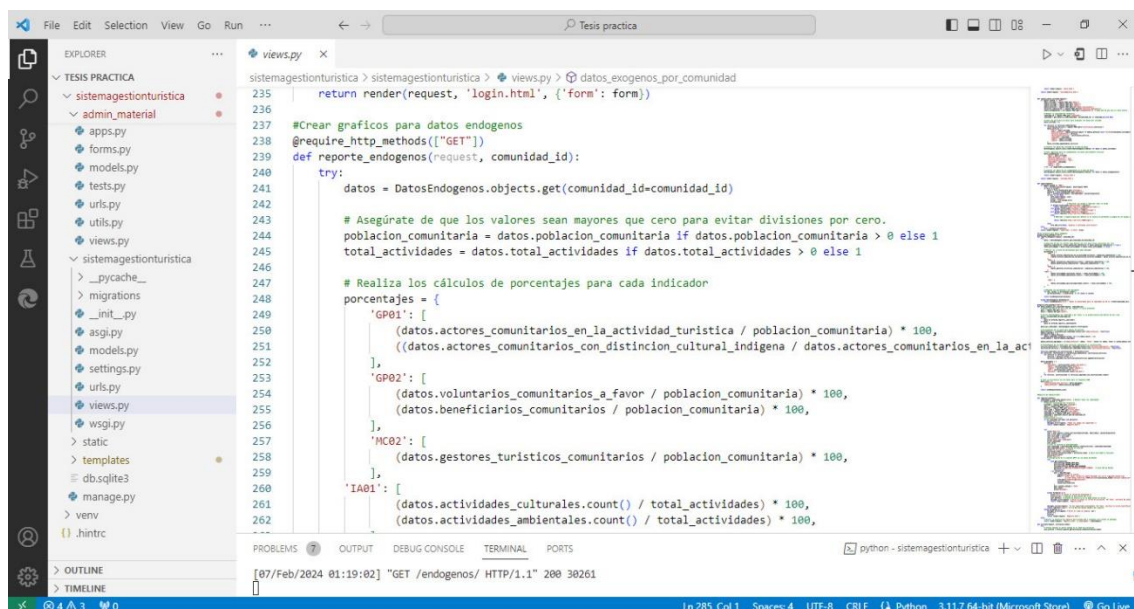


```
280 @require_http_methods(["GET"])
281 def datos_exogenos_por_comunidad(request, comunidad_id):
282     # Obtener parámetros de año y mes del request, si están presentes
283     year = request.GET.get('year')
284     month = request.GET.get('month')
285
286     # Filtrar DatosExogenos por comunidad y por fecha, si se proporcionaron parámetros de año y mes
287     query = Q(comunidad_id=comunidad_id)
288     if year:
289         query &= Q(fecha_registro__year=year)
290     if month:
291         query &= Q(fecha_registro__month=month)
292
293     datos_por_comunidad = DatosExogenos.objects.filter(query)
294
295     # Procesamiento de los datos para medios de difusión
296     medio_difusion = list(datos_por_comunidad.values_list('medio_difusion', flat=True))
297     medios_separados = []
298     for medio in medio_difusion:
299         medios_separados.extend([m.strip() for m in medio.split(',')])
300     conteo_medios = Counter(medios_separados)
301
302     medios_difusion_agrupados = [{'medio_difusion': medio, 'total': total} for medio, total in conteo_medios.items()]
303
304     # Procesamiento de los datos para servicios comunitarios y calificaciones
305     servicio_comunitario = list(datos_por_comunidad.values_list('servicio_comunitario', flat=True))
306     calificacion_servicio = list(datos_por_comunidad.values_list('calificacion_servicio', flat=True))
307
```

Figura 23: Código de reportes exógenos

Implementación del reporte de datos endógenos

La Figura 24, visualiza la implementación de los reportes de los datos endógenos que facilitó el acceso a los datos almacenados mediante el ORM de Django.

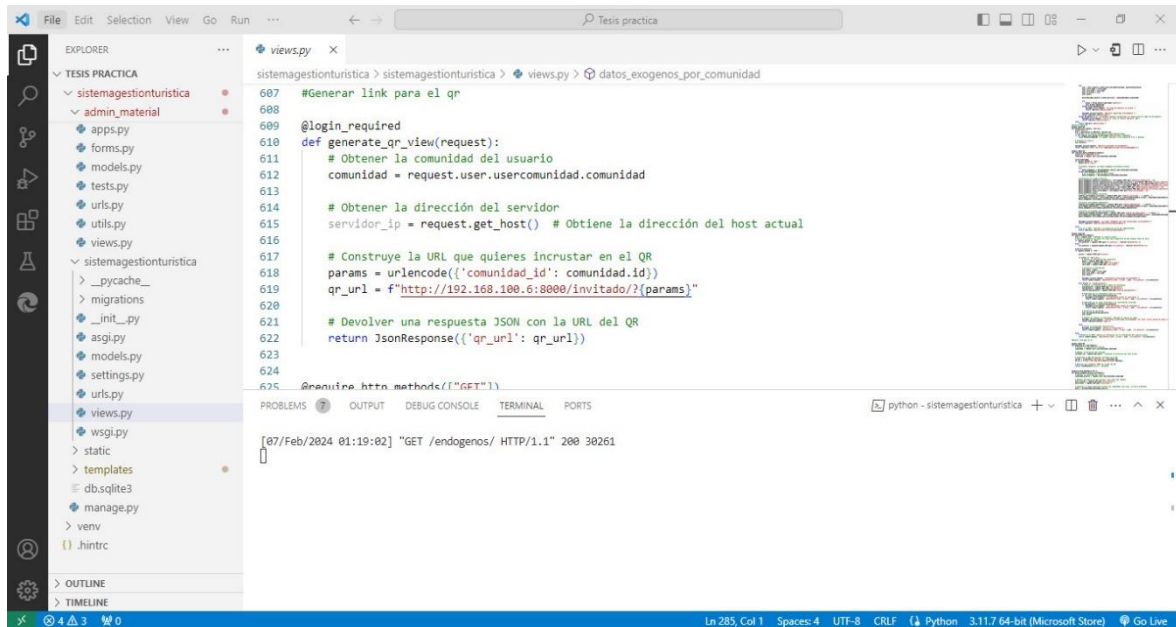


```
235 return render(request, 'login.html', {'form': form})
236
237 #Crear graficos para datos endogenos
238 @require_http_methods(["GET"])
239 def reporte_endogenos(request, comunidad_id):
240     try:
241         datos = DatosEndogenos.objects.get(comunidad_id=comunidad_id)
242
243         # Asegúrate de que los valores sean mayores que cero para evitar divisiones por cero.
244         poblacion_comunitaria = datos.poblacion_comunitaria if datos.poblacion_comunitaria > 0 else 1
245         total_actividades = datos.total_actividades if datos.total_actividades > 0 else 1
246
247         # Realiza los cálculos de porcentajes para cada indicador
248         porcentajes = {
249             'GP01': [
250                 (datos.actores_comunitarios_en_la_actividad_turistica / poblacion_comunitaria) * 100,
251                 ((datos.actores_comunitarios_con_distincion_cultural_indigena / datos.actores_comunitarios_en_la_actividad_turistica) * 100,
252             ],
253             'GP02': [
254                 (datos.voluntarios_comunitarios_a_favor / poblacion_comunitaria) * 100,
255                 (datos.beneficiarios_comunitarios / poblacion_comunitaria) * 100,
256             ],
257             'MC02': [
258                 (datos.gestores_turisticos_comunitarios / poblacion_comunitaria) * 100,
259             ],
260             'IAB01': [
261                 (datos.actividades_culturales.count() / total_actividades) * 100,
262                 (datos.actividades_ambientales.count() / total_actividades) * 100,
263             ]
264         }
265     except:
266         pass
267
```

Figura 24: Código de reportes endógenos

Implementación de código QR

En la Figura 25, se encuentra la implementación del código QR que facilitó a los turistas completar la encuesta de datos exógenos utilizando un dispositivo móvil con escáner QR.



```
607 #Generar link para el qr
608
609 @login_required
610 def generate_qr_view(request):
611     # Obtener la comunidad del usuario
612     comunidad = request.user.usercomunidad.comunidad
613
614     # Obtener la dirección del servidor
615     servidor_ip = request.get_host() # Obtiene la dirección del host actual
616
617     # Construye la URL que quieres incrustar en el QR
618     params = urlencode({'comunidad_id': comunidad.id})
619     qr_url = f"http://192.168.100.6:8000/invitado/{params}"
620
621     # Devolver una respuesta JSON con la URL del QR
622     return JsonResponse({"qr_url": qr_url})
623
624
625 @csrf_exempt
626 @authentication_classes((None,))
627 def generate_qr_view(request):
628     # Obtener la comunidad del usuario
629     comunidad = request.user.usercomunidad.comunidad
630
631     # Obtener la dirección del servidor
632     servidor_ip = request.get_host() # Obtiene la dirección del host actual
633
634     # Construye la URL que quieres incrustar en el QR
635     params = urlencode({'comunidad_id': comunidad.id})
636     qr_url = f"http://192.168.100.6:8000/invitado/{params}"
637
638     # Devolver una respuesta JSON con la URL del QR
639     return JsonResponse({"qr_url": qr_url})
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[07/Feb/2024 01:19:02] "GET /endogenos/ HTTP/1.1" 200 30261
```

Figura 25: Código QR

Visualización de Reportes

En la Figura 26, se representan los reportes generados por parte de los datos exógenos y de los datos endógenos, las cuales se representan en barras y pastel.

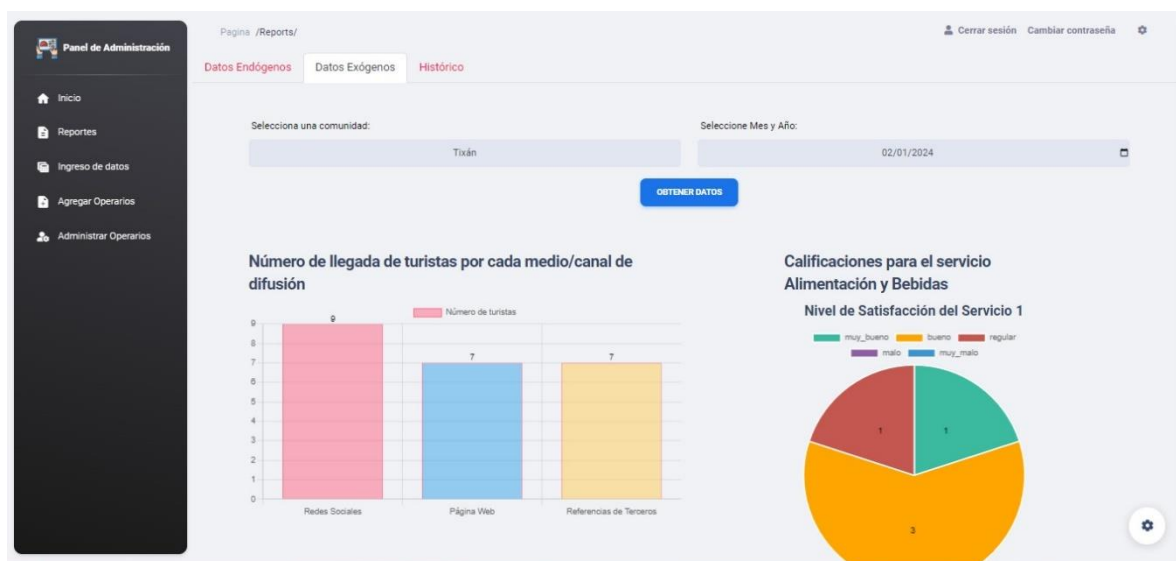


Figura 26: Visualización de reportes

3.7.6 Planificación de pruebas

Para evaluar la aplicación web del modelo de gestión turística sostenible comunitaria en Chimborazo utilizando Django, se ha seleccionado Apache JMeter como la herramienta principal para llevar a cabo las pruebas de rendimiento. JMeter es una herramienta reconocida y empleada para llevar a cabo pruebas de carga, estrés y rendimiento en aplicaciones web. La elección de JMeter se fundamenta en su capacidad para simular condiciones de uso real mediante la generación de múltiples solicitudes concurrentes, permitiendo evaluar la respuesta del sistema bajo diferentes niveles de carga. La Tabla 8, muestra los indicadores de las medidas que se realizó con la herramienta JMeter.

Tabla 8: Tabla de medición de JMeter

Indicador	Descripción
% Eficacia	Mide la eficacia general del sistema en cumplir con los objetivos establecidos para la gestión turística sostenible.
% Tiempo de respuesta	Evalúa el tiempo que el sistema necesita para responder a una solicitud, analizando la velocidad y eficacia operativa.
% Utilización de Recursos	Supervisa la eficacia en la utilización de los recursos del sistema, contemplando la CPU, la memoria y el ancho de banda.

3.7.7 Ejecución de las pruebas

Se diseñaron pruebas para cada módulo de la aplicación web, replicando condiciones de uso real mediante la simulación de la actividad de usuarios concurrentes. Para cada método disponible, GET y POST, se configuraron parámetros relevantes para garantizar la representación precisa de los escenarios de uso. Este enfoque permitió evaluar la capacidad del sistema para manejar cargas significativas y detectar áreas donde podría haber limitaciones o puntos de mejora en el rendimiento. En la Figura 27, se exhibe la configuración de JMeter que se realizó para cada módulo.

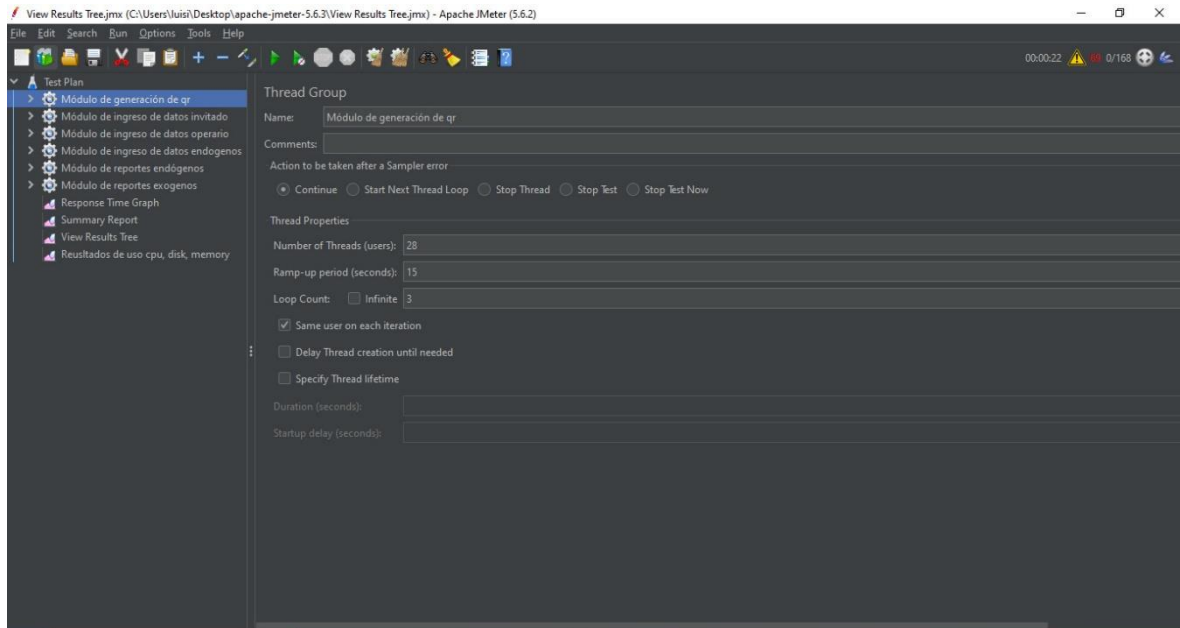


Figura 27: Configuración JMeter

En la Figura 28, se visualiza la configuración del HTTP Request para los métodos GET y POST para el aplicativo.

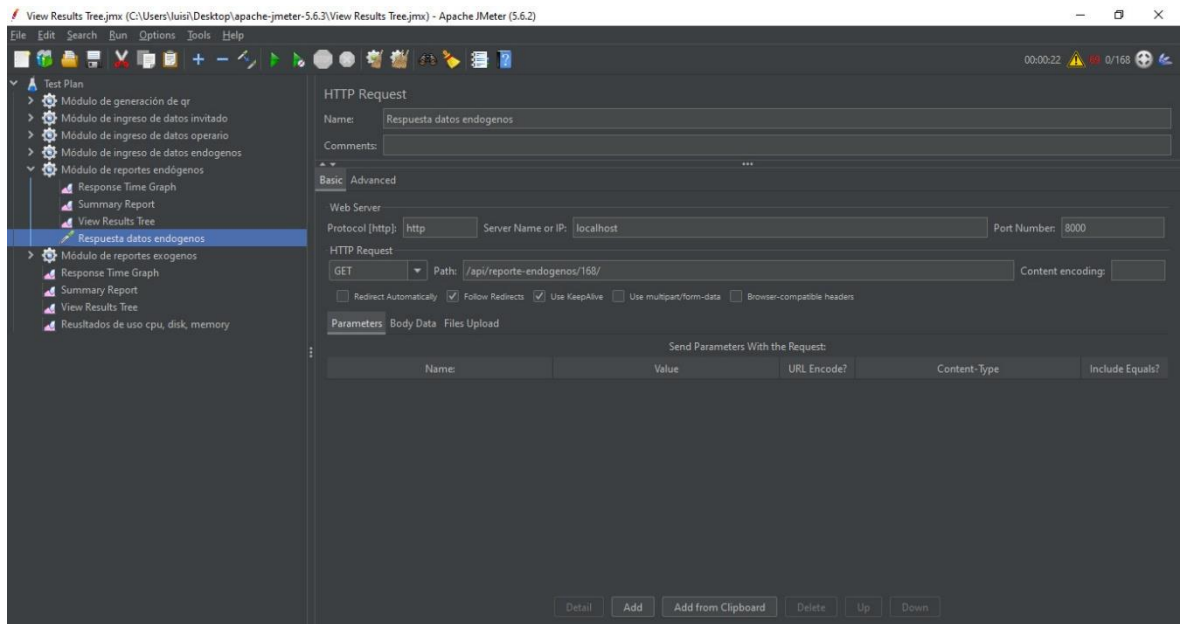
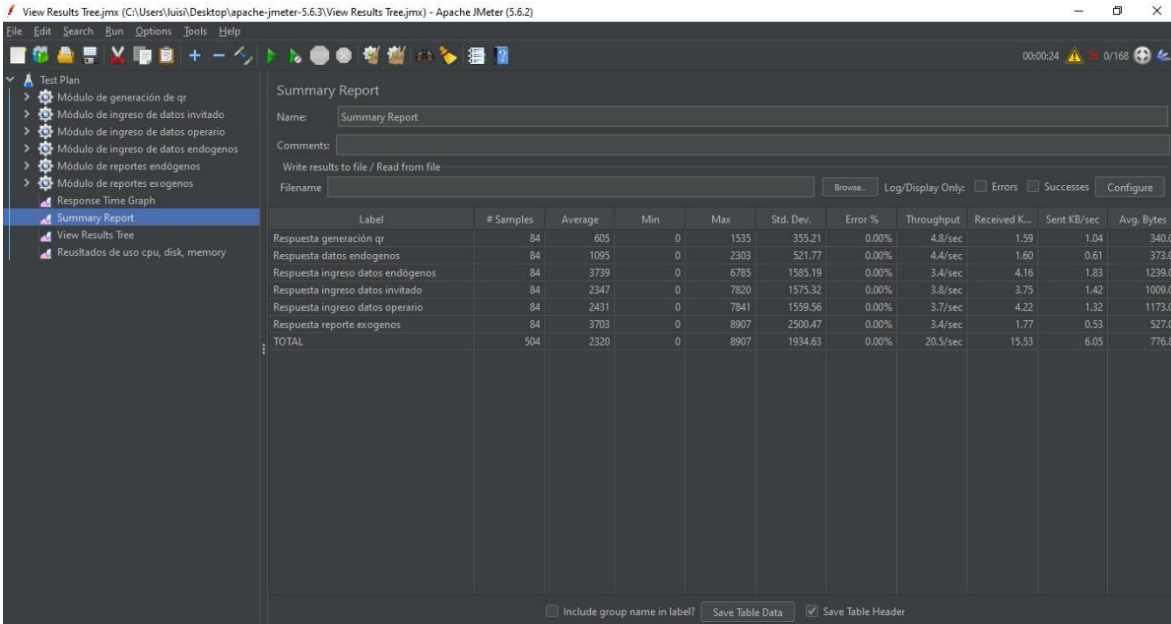


Figura 28: HTTP Request

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1 Resultados

Al término del desarrollo del aplicativo web destinada al modelo de gestión turística, se llevó a cabo pruebas de rendimiento mediante la herramienta JMeter. Los resultados obtenidos se vinculan con el modelo de calidad FURPS. La figura 29, presenta un resumen por la herramienta JMeter donde se observa cada módulo con su respectivo porcentaje de error y total de solicitudes realizadas.



The screenshot shows the Apache JMeter 5.6.2 Summary Report window. The window title is "View Results Tree.jmx (C:\Users\luisa\Desktop\apache-jmeter-5.6.3\View Results Tree.jmx) - Apache JMeter (5.6.2)". The interface includes a menu bar (File, Edit, Search, Run, Options, Tools, Help), a toolbar, and a sidebar with a "Test Plan" tree. The main area displays a "Summary Report" for a test named "Summary Report". Below the report title, there are fields for "Name", "Comments", and "Write results to file / Read from file". A table of results is displayed with the following columns: Label, # Samples, Average, Min, Max, Std. Dev., Error %, Throughput, Received K..., Sent KB/sec, and Avg. Bytes. The table contains data for several test modules and a total row. At the bottom of the window, there are checkboxes for "Include group name in label?", "Save Table Data", and "Save Table Header".

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
Respuesta generación qr	84	605	0	1535	355.21	0.00%	4.8/sec	1.59	1.04	340.0
Respuesta datos endogenos	84	1095	0	2303	521.77	0.00%	4.4/sec	1.60	0.61	373.0
Respuesta ingreso datos endógenos	84	3739	0	6785	1585.19	0.00%	3.4/sec	4.16	1.83	1239.0
Respuesta ingreso datos invitado	84	2347	0	7820	1575.32	0.00%	3.8/sec	3.75	1.42	1009.0
Respuesta ingreso datos operario	84	2431	0	7841	1559.56	0.00%	3.7/sec	4.22	1.32	1173.0
Respuesta reporte exogenos	84	3703	0	8907	2500.47	0.00%	3.4/sec	1.77	0.53	527.0
TOTAL	504	2320	0	8907	1934.63	0.00%	20.5/sec	15.53	6.05	776.8

Figura 29: Resumen de resultados - JMeter

En la figura 30, se evidencia el árbol de resultados entregados por la herramienta JMeter, las cuales nos arroja que todas las solicitudes obtuvieron una respuesta por parte del sistema.

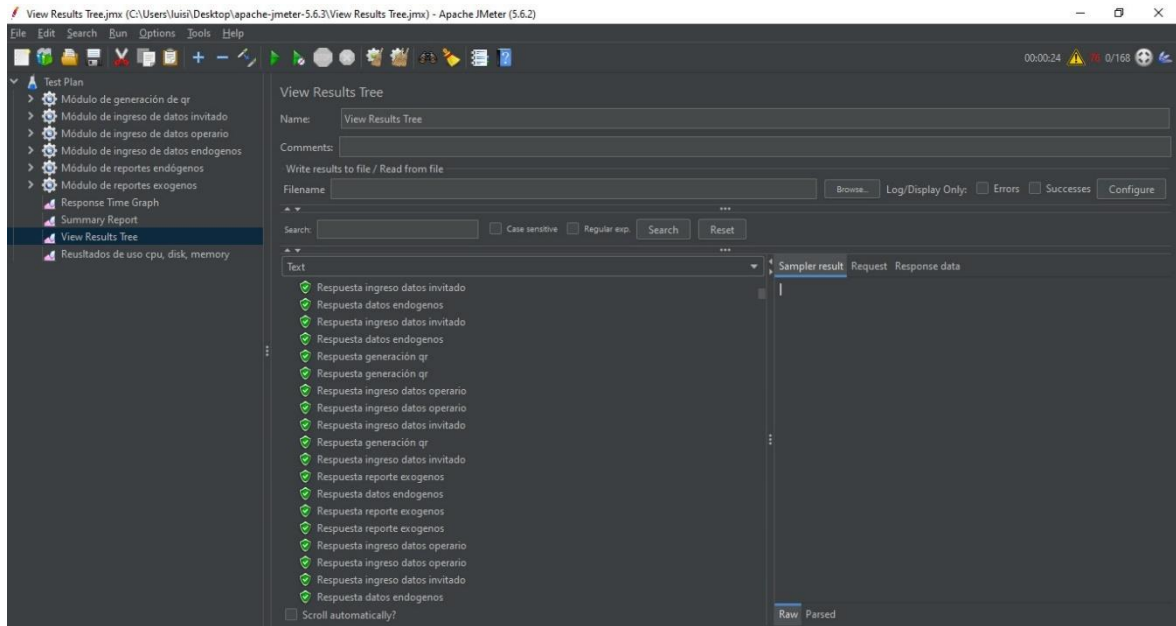


Figura 30: Vista de árbol de resultados - JMeter

En la Tabla 9, se exhibe la cantidad de solicitudes durante la ejecución de pruebas, que para en módulo se realizaron 3 bucles de pruebas de carga con 28 solicitudes a la aplicación web en un periodo de tiempo de 15 segundos, en total se registraron 504.

Tabla 9: Cantidad de procedimientos realizados

Módulo	Cantidad de Pruebas
Reportes Endógenos	84
Reportes Exógenos	84
Ingreso de datos invitado	84
Ingreso de datos operario	84
Ingreso de datos endógenos	84
Generación de QR	84

La figura 31, detalla la evaluación inicial realizada a la aplicación web que nos revela el porcentaje de éxito de las pruebas realizadas.



Figura 31: Porcentajes de resultados de evaluación inicial

4.2 Valoración de indicadores

4.2.1 Tiempo de desarrollo

Abarcó un lapso de cuatro meses donde se desarrollaron las diferentes funcionalidades.

4.2.2 Número de líneas de código

Después de cuatro meses de desarrollo, la aplicación web para el modelo de gestión turística sostenible comunitaria de Chimborazo, fue finalizada, abarcando un total de 18,500 líneas de código.

4.2.3 Numero de funcionalidades

Tras un período de desarrollo, la aplicación web incorporó un conjunto de 6 funcionalidades principales que son:

- Reportes Endógenos
- Reportes Exógenos
- Ingreso de datos invitado
- Ingreso de datos operario
- Ingreso de datos endógenos
- Generación de QR

4.2.4 Eficacia

En la tabla 10, se evidencian los resultados de la eficacia de las solicitudes realizadas a la aplicación web, que en este sentido fue de 100% del indicador eficacia.

Tabla 10: Requerimientos solicitados a la aplicación web

Parámetros	Indicadores	Requerimientos
Eficacia	Requerimientos exitosos del aplicativo	100%
	Requerimientos fallidos del aplicativo	0%

En la figura 32, se representa de manera gráfica el porcentaje de eficacia que obtuvo la aplicación web tras la realización de las pruebas, que en este sentido fue de 100% del indicador eficacia.

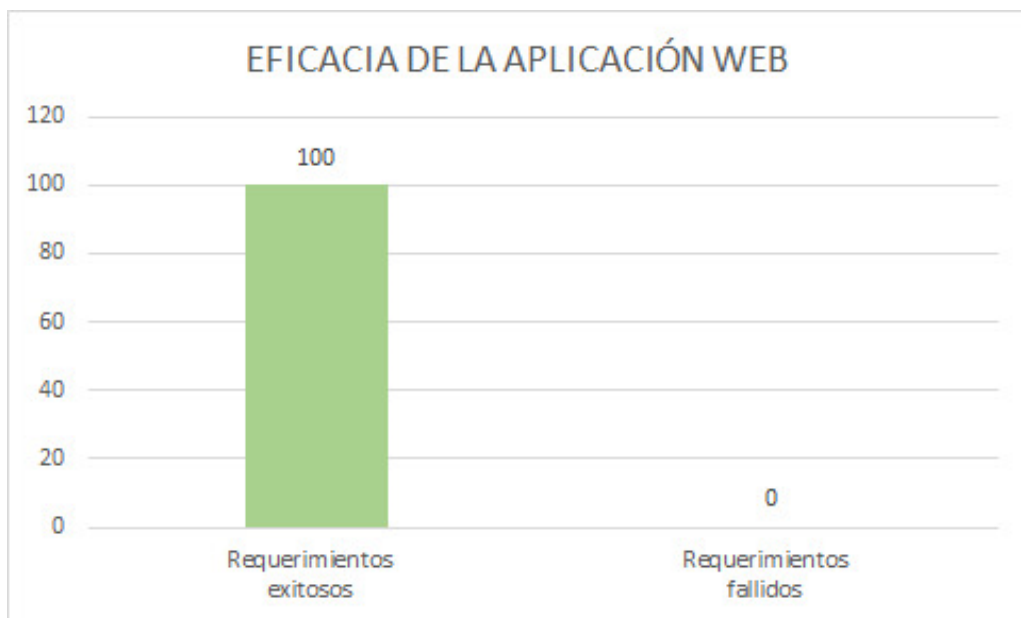


Figura 32: Eficacia de la aplicación web

4.2.5 Tiempo de respuesta

La figura 33, registra el tiempo de las respuestas que obtuvieron al realizarse varias solicitudes a cada uno de los módulos, el tiempo de respuesta tuvo un rango de 480 a 4600 milisegundos.

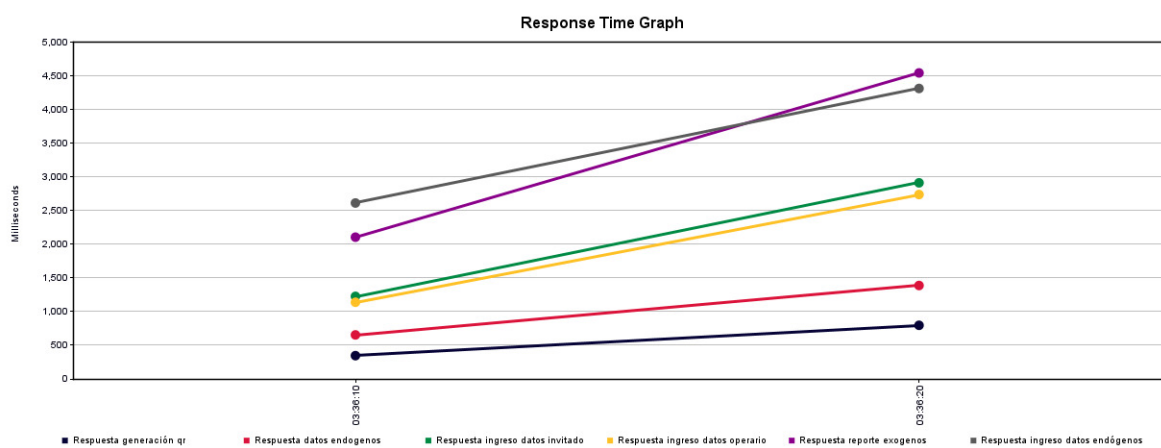


Figura 33: Tiempo de respuesta de la aplicación web

4.2.6 Utilización de recursos

En la tabla 11, se percibe el porcentaje de utilización de recursos que requirió la aplicación web al momento de realizarse las pruebas.

Tabla 11: Utilización de recursos

Parámetro	Indicador	% Consumo
Uso de recursos	Uso de CPU	25%
	Uso de RAM	37%
	Uso de disco duro	10%

4.2.7 Valores obtenidos del estudio en base al modelo FURPS

En la tabla 12, exhiben los valores que se obtuvieron del estudio en base al modelo FURPS realizado a la aplicación web.

Tabla 12: Valores obtenidos del estudio en base al modelo FURPS

Parámetros	Valoraciones del estudio
Eficacia	100%
Tiempo de respuesta	4600ms
Uso de recursos	25%

4.3 Discusión

El impacto significativo de la elección del framework Django en el desarrollo fue determinante. Se seleccionó Django por su reconocida robustez y capacidad para la creación de aplicaciones web de alta calidad. Para la evaluación de los indicadores de rendimiento de la aplicación web, se empleó la herramienta JMeter. Esta decisión permitió un análisis detallado y preciso del comportamiento del sistema bajo diversas condiciones de uso, garantizando que la aplicación desarrollada en Django cumpliera con los requerimientos del modelo de gestión turística sostenible.

Al comparar los resultados obtenidos con los estándares establecidos por el modelo de calidad FURPS, se constató que la eficacia del aplicativo alcanzó el 100%, superando el umbral del 95% establecido por el modelo. En términos de tiempo de respuesta, el aplicativo demostró un rendimiento superior, con un tiempo promedio de respuesta de 4600 ms, notablemente inferior al máximo de 5 segundos sugerido por FURPS. Además, el uso promedio de recursos del sistema fue del 25%, alineándose con las expectativas previstas por el modelo de calidad. Estos resultados evidencian la alta calidad de rendimiento del aplicativo, validando así la eficacia de las tecnologías y metodologías implementadas.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El Framework Django demuestra su eficacia y versatilidad como herramienta. Su estructura, fundamentada en el lenguaje Python, facilita la creación de aplicaciones web robustas y escalables. Django provee una serie de características como la seguridad integrada, un ORM (Object-Relational Mapping) para manejar bases de datos y un sistema de plantillas flexible, lo convierte en una solución ideal para desarrollar la aplicación web destinada a la gestión turística sostenible comunitaria en Chimborazo. La adaptabilidad de Django para cumplir con las necesidades específicas del proyecto y su capacidad para soportar el desarrollo rápido y eficiente han sido factores clave en su elección como la plataforma de desarrollo.
- Para elaborar la aplicación web del modelo de gestión turística sostenible comunitaria de Chimborazo, se optó por SQL Server para la administración de la base de datos. En el frontend, se implementaron plantillas y formularios de Django, junto con lenguajes como JavaScript, HTML y CSS, para gestionar las solicitudes a la API. Respecto al backend, se empleó Python 3.11 en el entorno de desarrollo de Visual Studio Code. Es relevante mencionar el uso del Framework Django, el cual agilizó el proceso de creación de la aplicación web y facilitó la organización del proyecto gracias a su arquitectura MVT y su compatibilidad con metodologías ágiles como Kanban.
- Las pruebas de rendimiento del aplicativo web que se realizó con la herramienta JMeter, que se basó en el modelo de calidad FURPS, se llevó a cabo con 84 iteraciones distribuidas en 6 módulos, totalizando 504 iteraciones. Los resultados muestran una eficacia del 100%, un tiempo de respuesta de 4600 ms y un uso de recursos del 25%. Estos resultados indican que la aplicación cumple con los criterios establecidos por el modelo de calidad FURPS.

5.2 Recomendaciones

- Antes de implementar Django, es esencial realizar una investigación exhaustiva sobre sus herramientas de seguridad, como CSRF (Cross-Site Request Forgery) y el manejo de sesiones, para garantizar la integridad y seguridad de las aplicaciones web. Se insta a implementar medidas de protección contra CSRF y configurar el manejo de sesiones para establecer una comunicación encriptada.
- En el desarrollo de las aplicaciones web, se sugiere adoptar la metodología Kanban como su metodología de desarrollo, que facilita la planificación flexible y dinámica. Utilizando tableros Kanban, es posible gestionar visualmente las tareas, asignar prioridades y monitorear el progreso.
- Para llevar a cabo pruebas exhaustivas de rendimiento en aplicaciones web, se sugiere utilizar JMeter. Esta herramienta ofrece una interfaz gráfica intuitiva para configurar y ejecutar pruebas de carga, rendimiento y estrés, permitiendo la

personalización según las necesidades del proyecto. Al combinar JMeter con el modelo de calidad FURPS, se establece un marco completo de evaluación que abarca todos los aspectos críticos del rendimiento de la aplicación web.

BIBLIOGRAFÍA

- [1] J. R. M. Ríos, M. P. Z. Ordóñez, M. J. C. Segarra, y F. G. G. Zerda, «COMPARACIÓN DE METODOLOGÍAS EN APLICACIONES WEB», n.º 25, 2018.
- [2] M. R. Valarezo Pardo, J. A. Honores Tapia, A. S. Gómez Moreno, y L. F. Vines Sánchez, «Comparación de tendencias tecnológicas en aplicaciones web», *3c Technol. Glosas Innov. Apl. Pyme*, vol. 7, n.º 3, pp. 28-49, 2018.
- [3] D. Ghimire, «Comparative study on Python web frameworks: Flask and Django», 2020.
- [4] W. S. Vincent, *Django for Beginners: Build websites with Python and Django*. WelcomeToCode, 2022.
- [5] C. L. Vidal-Silva, A. Sánchez-Ortiz, y J. Serrano, «Experiencia académica en desarrollo rápido de sistemas de información web con Python y Django», vol. 14, 2021.
- [6] H. Gore *et al.*, «Django: Web Development Simple & Fast», *Ann. Romanian Soc. Cell Biol.*, vol. 25, n.º 6, Art. n.º 6, may 2021.
- [7] R. Espinosa-Hurtado, «Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web», *CEDAMAZ*, vol. 11, n.º 2, Art. n.º 2, dic. 2021, doi: 10.54753/cedamaz.v11i2.1182.
- [8] A. Hakim «Exploring Django ORM». Accedido: 24 de enero de 2024. [En línea]. Disponible en: <https://www.linkedin.com/pulse/exploring-django-orm-anas-al-hakim>
- [9] Martin Paul Eve, «Using Django ORM, Click, and Rich to create useful command-line python apps», ene. 2022, doi: <https://doi.org/10.59348/pvsyt-hbv27>.
- [10] Valentino Gagliardi, «Decoupled Django with the Django REST Framework», *Apress EBooks*, pp. 63-89, ene. 2021, doi: https://doi.org/10.1007/978-1-4842-7144-5_6.
- [11] Valentino Gagliardi, «Modern Django and the Django REST Framework», *Apress EBooks*, pp. 31-40, ene. 2021, doi: https://doi.org/10.1007/978-1-4842-7144-5_3.
- [12] Songtao Chen, Shahed Ahmmed, Kanhaiya Lal, y Chunhua Deming, «Django Web Development Framework: Powering the Modern Web», *Am. J. Trade Policy*, vol. 7, n.º 3, pp. 99-106, dic. 2020, doi: <https://doi.org/10.18034/ajtp.v7i3.675>.
- [13] Daniel Rubio, «Django Model Queries and Managers», *Apress EBooks*, pp. 341-401, ene. 2017, doi: https://doi.org/10.1007/978-1-4842-2787-9_8.
- [14] David Ashley, «Django», *Apress EBooks*, pp. 183-199, ene. 2020, doi: https://doi.org/10.1007/978-1-4842-6339-6_6.
- [15] Kulanda Duisebekova, Roman Khabirov, y Azamat Zholzhan, «DJANGO AS SECURE WEB-FRAMEWORK IN PRACTICE», *M Tynyşpaev Atyndağ Qazaq Kôlik Žãne Kommun. Akad. Habarşysy*, vol. 116, n.º 1, pp. 275-281, mar. 2021, doi: <https://doi.org/10.52167/1609-1817-2021-116-1-275-281>.
- [16] Adrian Holovaty y Jacob Kaplan-Moss, «The Django Administration Site», *Apress EBooks*, pp. 83-94, ene. 2008, doi: https://doi.org/10.1007/978-1-4302-0331-5_6.
- [17] Anuj Kumar Sewani, «Why we use Django rather than Flask in Asset Management System», *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, n.º VI, pp. 4053-4056, jun. 2021, doi: <https://doi.org/10.22214/ijraset.2021.35756>.

- [18] Kinsta, «Flask vs Django: Elijamos Tu Próximo Framework Python», Kinsta®. Accedido: 24 de enero de 2024. [En línea]. Disponible en: <https://kinsta.com/es/blog/flask-vs-django/>
- [19] V. Pomata, «Arquitectura de desarrollo web con Django y apps con Flutter», 2020, [En línea]. Disponible en: <https://openaccess.uoc.edu/bitstream/10609/106467/7/dviejoTFG0120memoria.pdf>
- [20] Dev. mozilla, «Introducción a Django - Aprende desarrollo web | MDN». Accedido: 25 de enero de 2024. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>
- [21] JavaTpoint «Django MVT - javatpoint», www.javatpoint.com. Accedido: 8 de febrero de 2024. [En línea]. Disponible en: <https://www.javatpoint.com/django-mvt>
- [22] K. M. Vamsi, P. Lokesh, K. N. Reddy, y P. Swetha, «Visualization of Real World Enterprise Data using Python Django Framework», *Mater. Sci. Eng.*, 2020.
- [23] Adrian Holovaty y Jacob Kaplan-Moss, «The Django Template System», *Apress EBooks*, pp. 31-58, ene. 2008, doi: https://doi.org/10.1007/978-1-4302-0331-5_4.
- [24] David Martínez, D. González, Juan M. Sánchez, y Renato M. Toasa, «Library in django framework to standardize early-stage web application development», jun. 2023, doi: <https://doi.org/10.23919/cisti58278.2023.10211921>.
- [25] B. Barrera y J. Carlos, «Desarrollo de una aplicación híbrida inteligente y un modelo de análisis de datos en fases para resumir y representar los comentarios de clientes sobre productos y servicios de bares y restaurantes mediante Django, IONIC, Angular, Deep Learning y redes sociales», 2022, [En línea]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/23420/1/UPS-CT010079.pdf>
- [26] C. J. A. Aguilar, «DESARROLLO DE UN PROTOTIPO DE MÓDULO DE ANÁLISIS DE TRÁFICO BASADO EN WIRESHARK PARA DETECCIÓN DE ATAQUES DE DENEGACIÓN USANDO INSPECCIÓN DE TRAMAS LORAWAN QUE PROVEA UNA CAPA DE INTEGRACIÓN API REST.», 2022, [En línea]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/23416/1/CD%2012845.pdf>
- [27] Daniel Rubio, «REST Services with Django», *Apress EBooks*, pp. 549-566, ene. 2017, doi: https://doi.org/10.1007/978-1-4842-2787-9_12.
- [28] Valentino Gagliardi, «Modern Django and the Django REST Framework», *Apress EBooks*, pp. 31-40, ene. 2021, doi: https://doi.org/10.1007/978-1-4842-7144-5_3.
- [29] E. Gabriel, «Metodologías de desarrollo de software», 2015, [En línea]. Disponible en: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- [30] P. P. A. Nicole, «PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR FACULTAD DE INGENIERÍA INGENIERÍA EN SISTEMAS DE INFORMACIÓN», [En línea]. Disponible en: https://repositorio.puce.edu.ec/bitstream/handle/22000/21056/Tesis_Automatizaci%C3%B3nDeProcesos_Angie%20Pilco.pdf?sequence=1&isAllowed=y
- [31] «actas.pdf». Accedido: 18 de diciembre de 2023. [En línea]. Disponible en: <https://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>

- [32] J. Gaete, R. Villarroel, I. Figueroa, H. Cornide-Reyes, y R. Muñoz, «Enfoque de aplicación ágil con Scrum, Lean y Kanban», *Ingeniare Rev. Chil. Ing.*, vol. 29, 2021.
- [33] «DMCS_U3_Contenido.pdf». Accedido: 21 de diciembre de 2023. [En línea]. Disponible en: https://dmd.unadmexico.mx/contenidos/DCEIT/BLOQUE2/DS/06/DMCS/U3/descargables/DMCS_U3_Contenido.pdf
- [34] A. F. Yungan Gualli, C. H. Morales Alarcón, J. E. Delgado Altamirano, y L. M. Espinoza Tinoco, «Modelo FURPS para el análisis del rendimiento de frameworks JSF», *3C TIC Cuad. Desarro. Apl. Las TIC*, vol. 8, n.º 4, pp. 65-83, dic. 2019, doi: 10.17993/3ctic.2019.84.65-83.
- [35] A. Zambrano, E. Alexis, y P. A. B. Guisñan, «APLICACIÓN WEB PARA EL SERVICIO DE TRÁMITES ACADÉMICOS DE LA UNACH USANDO UNA ARQUITECTURA BASADA EN MICROSERVICIOS», 2022, [En línea]. Disponible en: <http://dspace.unach.edu.ec/bitstream/51000/10247/1/Erick%20Alexis%20Alvarado%20Zambrano.%20Aplicaci%C3%B3n%20web%20para%20el%20servicio%20de%20tr%C3%A1mites%20acad%C3%A9micos%20de%20la%20unach%20usando%20una%20arquitectura%20basada%20en%20microservicios.pdf>
- [36] M. Czuper, «Applying automated performance testing with Apache JMeter», 2022, [En línea]. Disponible en: <https://aaltodoc.aalto.fi/server/api/core/bitstreams/77126686-ec2b-4f0a-93f7-5d07984b7852/content>

ANEXOS

Anexo 1: Acta entrega del sistema



CERTIFICACIÓN

A petición de las partes interesadas: yo Msc. Daniel Marcelo Guerrero Vaca. Docente de la Facultad de Ciencias Políticas y Administrativas de la Universidad Nacional de Chimborazo y Director del proyecto de investigación **GESTIÓN TURÍSTICA SOSTENIBLE EN LAS ORGANIZACIONES COMUNITARIAS DE LA PROVINCIA DE CHIMBORAZO.**

Certifico:

Que, los señores **LUIS FERNANDO LUCERO PEÑA**, portador del número de cédula de ciudadanía No. 2350213548 y **CRISTOFER VINICIO BALTAZAR YUCAILLA**, portador del número de cédula de ciudadanía No. 1851000131, estudiantes de la carrera de Ingeniería en Tecnologías de la Información de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo, entregaron el Sistema para el Modelo de Gestión Turística Sostenible Comunitaria de Chimborazo, como parte de su trabajo de Titulación denominado: **"APLICACIÓN WEB PARA EL MODELO DE GESTIÓN TURÍSTICA SOSTENIBLE COMUNITARIA DE CHIMBORAZO UTILIZANDO DJANGO"**

Es todo cuanto puedo certificar en honor a la verdad. Facultando a los interesados hacer uso del presente, para los fines que crea conveniente.

Riobamba. 14 de febrero de 2024



Msc. Daniel Marcelo Guerrero Vaca
Director del proyecto de investigación

Anexo 2: Manual de usuario



SISTEMA DE GESTIÓN TURÍSTICA COMUNITARIA

Manual de Usuario

VERSION

1.0.0

FECHA DE CREACION

10/04/2024

REALIZADO POR

Cristofer Baltazar y Luis Lucero



Índice

1. Introducción	3
2. Visión general del sistema	3
3. Administrador	4
Pantalla principal de administrador.....	4
Login Administrador	4
Registro administrador	5
Activación de cuenta Administrador.....	5
Dashborad Administrador.....	6
Ingreso de datos endógenos	7
Reporte de datos endógenos	8
Reporte de datos exógenos.....	9
Reporte histórico.....	11
Agregar operario.....	12
Administración de operario.....	14
Cambio de contraseña	15
Recuperación de contraseña	17
Configuración de aspecto del sistema	20
4. Operario	22
Login operario	22
Ingreso de datos exógenos.....	23
Generador de QR.....	25
Escaneo de código QR mediante teléfono	25
Editar perfil operario.....	27



1. Introducción

Este documento está destinado para el usuario final del sistema, conteniendo así la descripción de cada módulo que servirá como guía. Ofreciendo instrucciones detalladas sobre todas las funciones y características del sistema.

2. Visión general del sistema

La visión general del sistema se concentra en la creación de una aplicación web que aprovecha el poderoso framework Django en Python, conocido por su enfoque "baterías incluidas". Este proyecto nace de una investigación sobre la Gestión Turística Sostenible en las Organizaciones Comunitarias de la provincia de Chimborazo, donde se ha desarrollado una aplicación web específicamente diseñada para respaldar y optimizar este modelo de gestión para las comunidades



3. Administrador

Pantalla principal de administrador

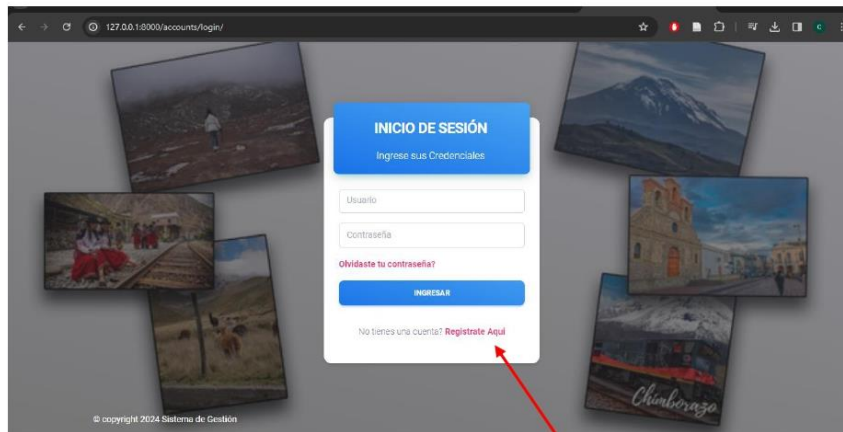
La pantalla principal será la siguiente, en la cual se mostrará el nombre del sistema y un botón el cual permite ingresar al sistema.



Dar clic en el botón "INGRESAR" para poder acceder al sistema.

Login Administrador

En la siguiente pantalla se mostrará el inicio de sesión (Login) en el cual colocaremos nuestras credenciales para ingresar al sistema. En caso de no contar con credenciales todavía, se puede dar clic en "Regístrate Aquí" para crear un usuario.

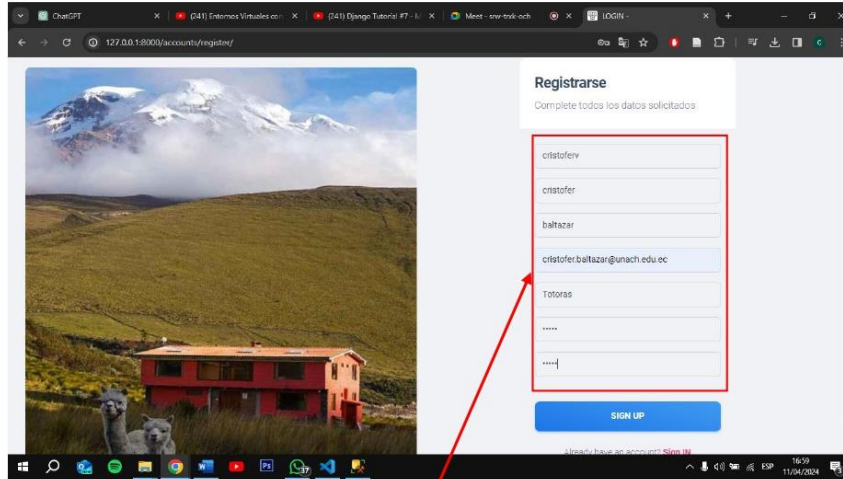


Dar clic en el botón "Regístrate Aquí"



Registro administrador

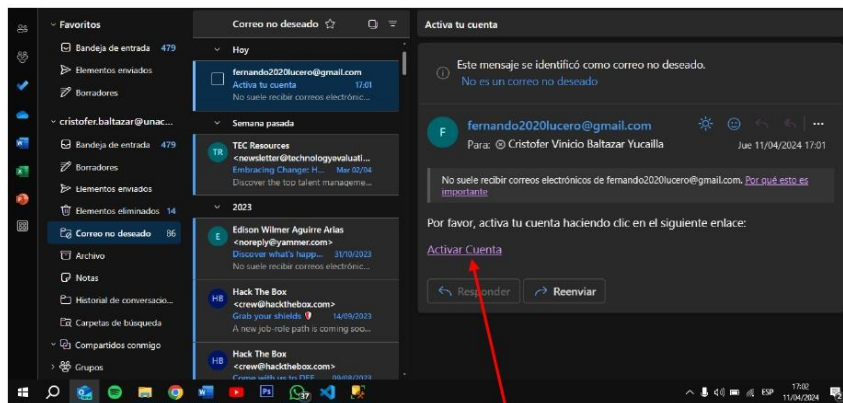
En la pantalla de registro de administrador se deberá llenar todos los datos solicitantes para poder acceder al sistema a través de un usuario.



Ingresar todos los datos en los campos correspondientes

Activación de cuenta Administrador

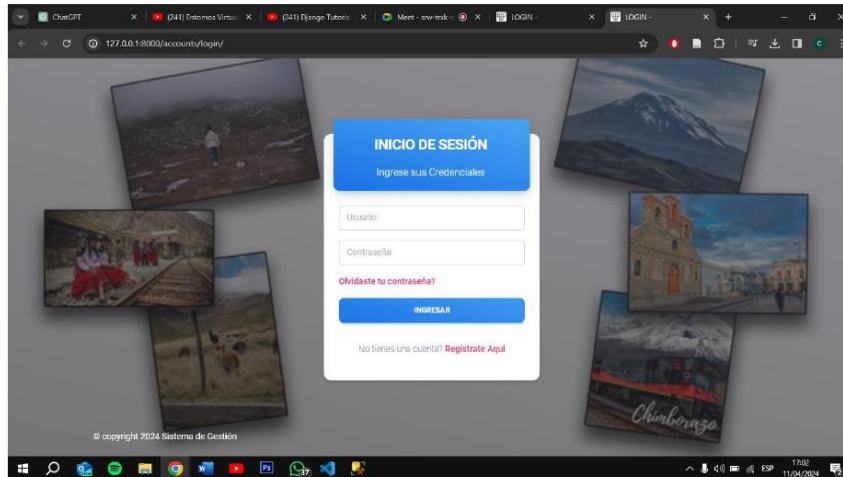
Para la activación de la cuenta deberá ingresar a su correo con el cual se registró y le llegará un enlace el cual deberá dar clic para poder activar la cuenta. Esta es una medida de seguridad para verificar al usuario.



Enlace de verificación enviada al correo

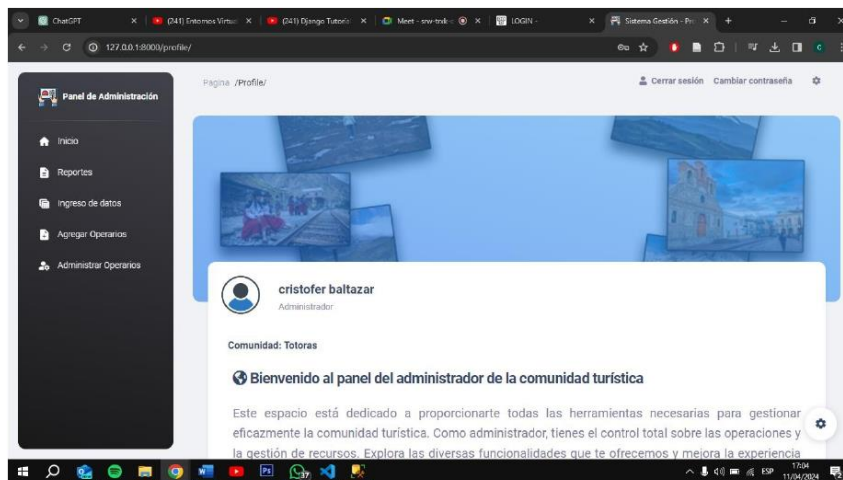


Al abrir el link nos muestra la siguiente pantalla la cual es el inicio de sesión. Deberá ingresar las credenciales creadas anteriormente.



Dashborad Administrador

Para el administrador se le mostrará el siguiente dashboard donde podrá gestionar todo sobre su comunidad.





Ingreso de datos endógenos

Se deberá ingresar al módulo de "Ingreso de datos" el cual nos permite ingresar los datos de la comunidad los cuales son considerados como datos endógenos.

Panel de Administración

- Inicio
- Reportes
- Ingreso de datos
- Agregar Operarios
- Administrar Operarios

Ingreso de datos endógenos de Totoras

Población comunitaria: 200

• Actores comunitarios en la actividad turística: 40

• Actores comunitarios con distinción cultural indígena: 120

• Voluntarios comunitarios a favor: 10

• Beneficiarios comunitarios: 20

Se llenarán los datos como se muestra a continuación, para posteriores reportes que se requieran.

Panel de Administración

- Inicio
- Reportes
- Ingreso de datos
- Agregar Operarios
- Administrar Operarios

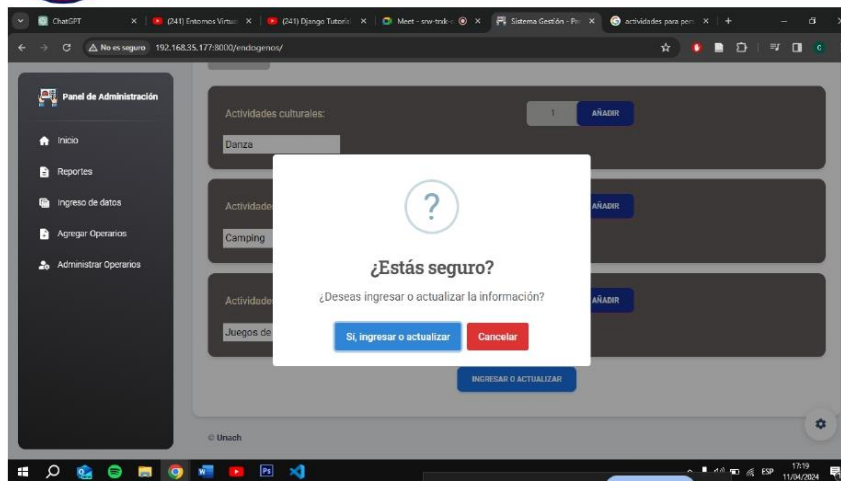
Actividades culturales: 1 **AÑADIR**
Danza

Actividades ambientales: 1 **AÑADIR**
Camping

Actividades para discapacitados: 2 **AÑADIR**
Juegos de mesa Pintura

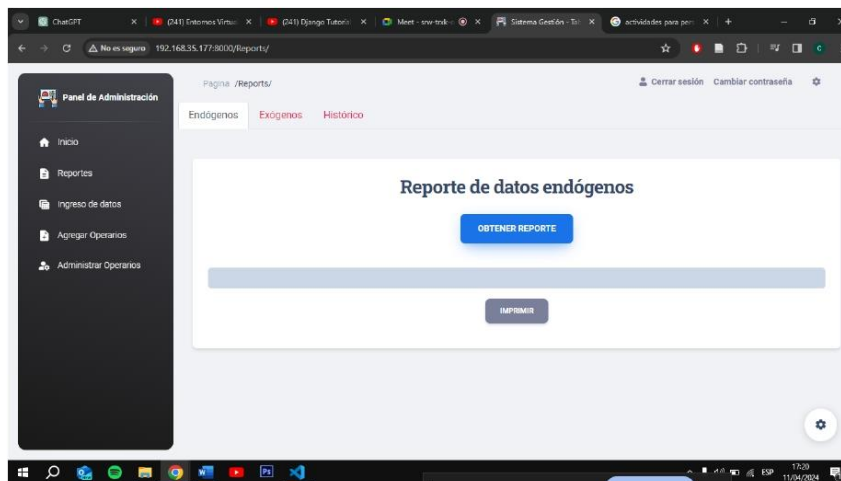
INGRESAR O ACTUALIZAR

Una vez llenado los datos, se deberá guardarlos en el cual se mostrará un mensaje el cual nos pedirá confirmación para guardar los datos.



Reporte de datos endógenos

En el módulo de reportes podremos escoger los reportes que se quieren, en este caso ingresaremos a los reportes Endógenos y daremos clic en obtener reporte.



Al dar clic se mostrará lo siguiente el cual es el reporte de datos endógenos que se podrá imprimir dando clic en los respectivos botones.



Código-Indicador	Valor %	Interpretación
CP01	20%	Grado porcentual de miembros de la comunidad relacionados directa e indirectamente con la actividad turística en el territorio comunitario.
	300%	Grado porcentual de miembros de la comunidad que comparten las costumbres, tradiciones, sabores y vestimenta identitaria cultural.
SP02	5%	Grado porcentual de miembros de la comunidad a favor del desarrollo del emprendimiento turístico comunitario.
	10%	Grado porcentual de miembros de la comunidad que son beneficiados directamente de la comercialización de servicios y actividades turísticas.
MD02	5%	Grado porcentual de miembros de la comunidad que son parte de la organización de gestión comunitaria OGC.
IA01	25%	Grado porcentual de actividades con enfoque y uso de patrimonios culturales del territorio comunitario.
	25%	Grado porcentual de actividades con enfoque y uso de patrimonios ambientales del territorio comunitario.
IA02	50%	Grado porcentual de actividades con inclusión de personas con discapacidades físicas y/o sensoriales en el territorio comunitario.

Reporte de datos exógenos

En reportes al escoger la opción de exógenos nos mostrará la siguiente pantalla donde podremos escoger las fechas de reportes.

Página: /reports/

Endógenos Exógenos Histórico

Selección una comunidad: Totoras

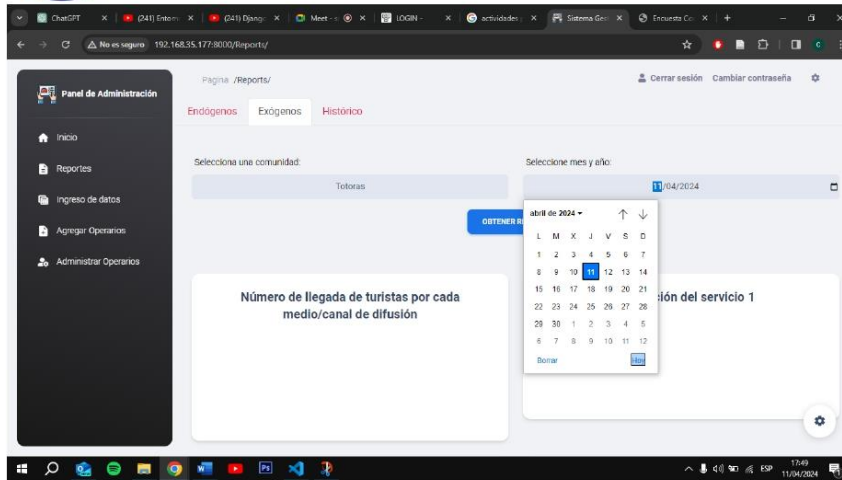
Selección mes y año: 11/04/2024

OBTENER REPORTE

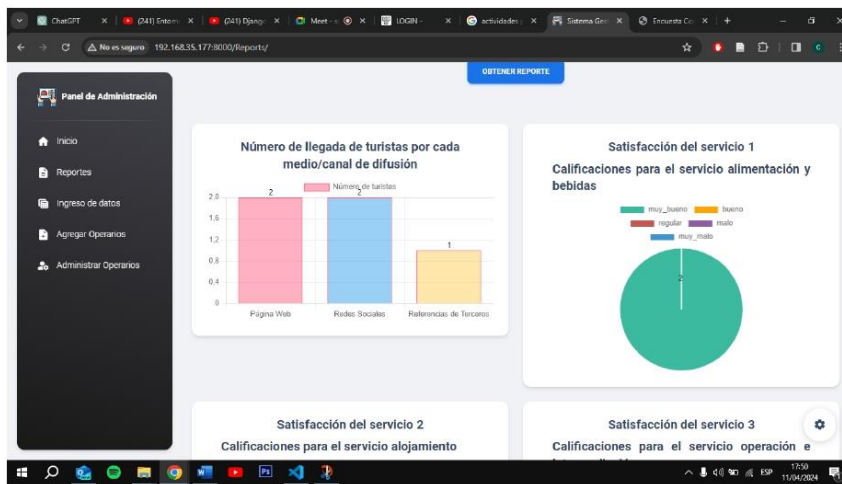
Número de llegada de turistas por cada medio/canal de difusión

Satisfacción del servicio 1

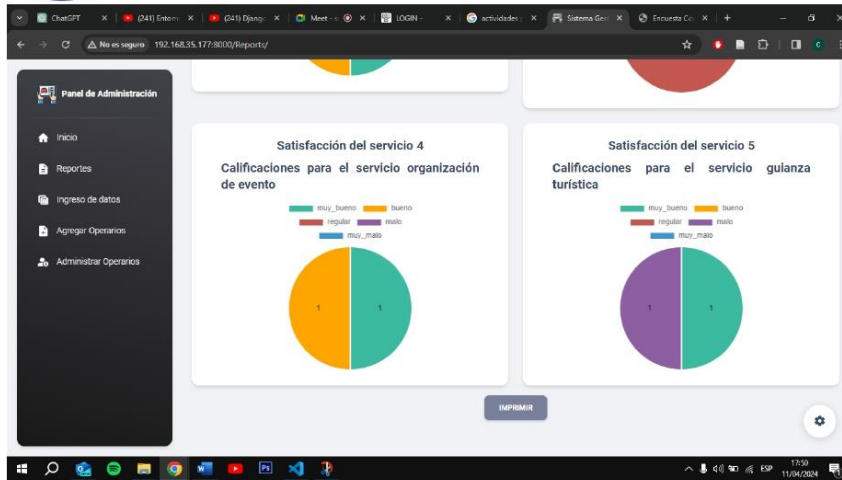
Se deberá de escoger la comunidad, el día, mes y año el cual servirá para el reporte que se desea obtener



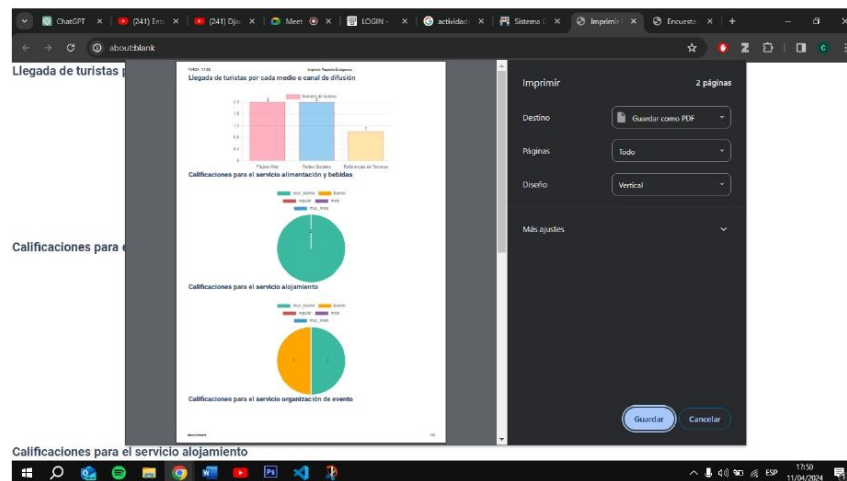
Una vez hecho el paso anterior se deberá dar clic en obtener reporte el cual generará las siguientes gráficas.



Al final se tendrá el botón que nos permitirá imprimir

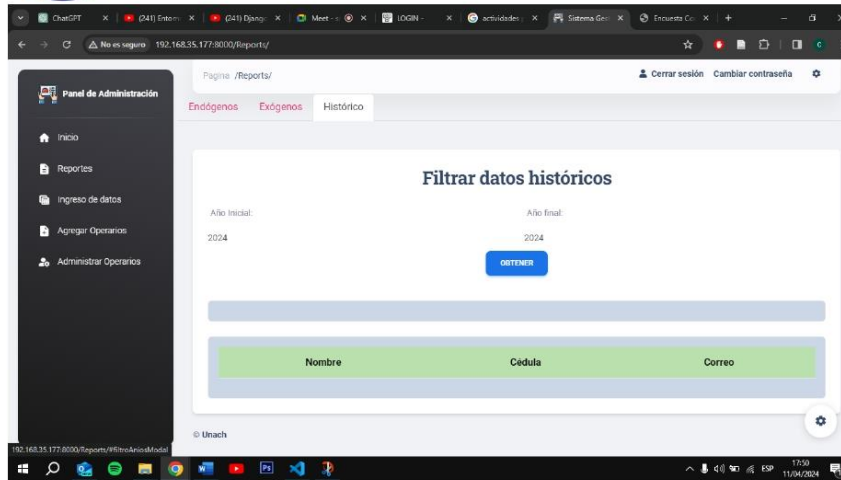


Al querer imprimir se mostrará lo siguiente, y se dará clic en imprimir el reporte

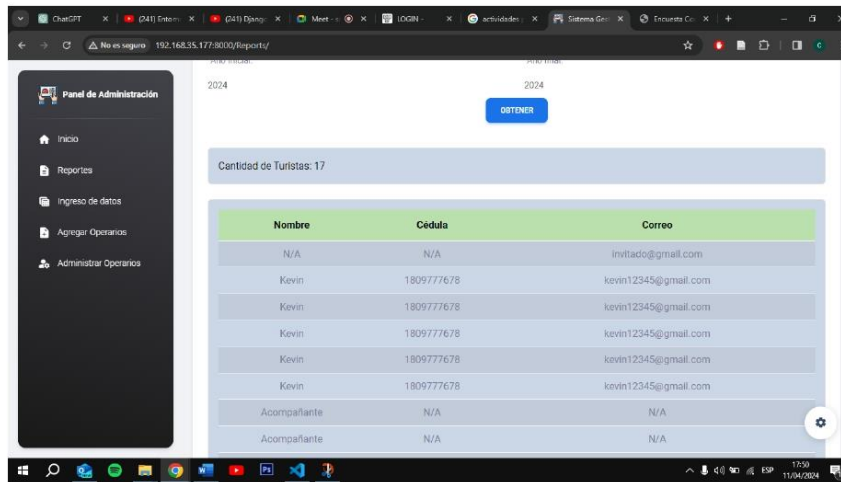


Reporte histórico

Se podrá escoger la opción de filtrar datos históricos, el cual nos provee de los datos que se han venido recolectando para el modelo en la aplicación web.

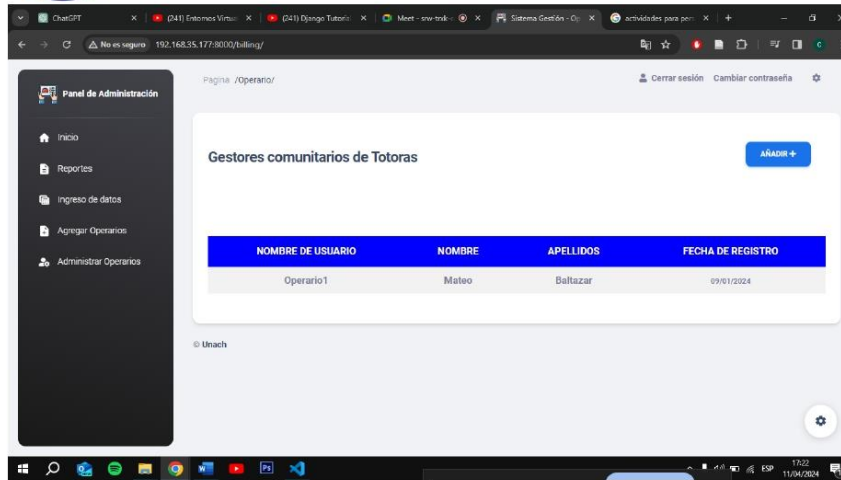


Al filtrar y dar clic en obtener se nos mostrará el histórico de los turistas que se hayan registrado en el sistema.

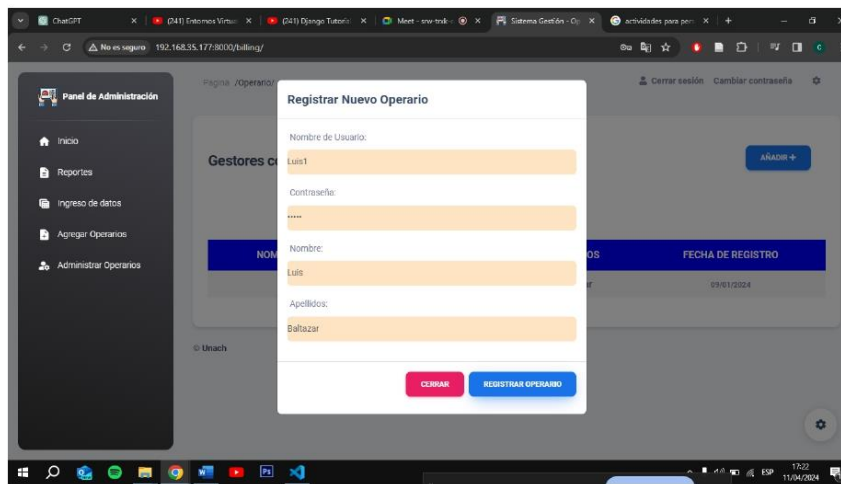


Agregar operario

En el modulo agregar operario se podrá crear tantos operarios que requiera la comunidad. Se deberá dar clic en Anadir y se llenará los datos que pide.



Los datos a agregar para la creación del operario de la comunidad son los siguientes, Nombre de usuario, contraseña, nombre y apellido.



Después de agregar se podrá visualizar todos los operarios creados para dicha comunidad. Se mostrará en forma de tabla los operarios.



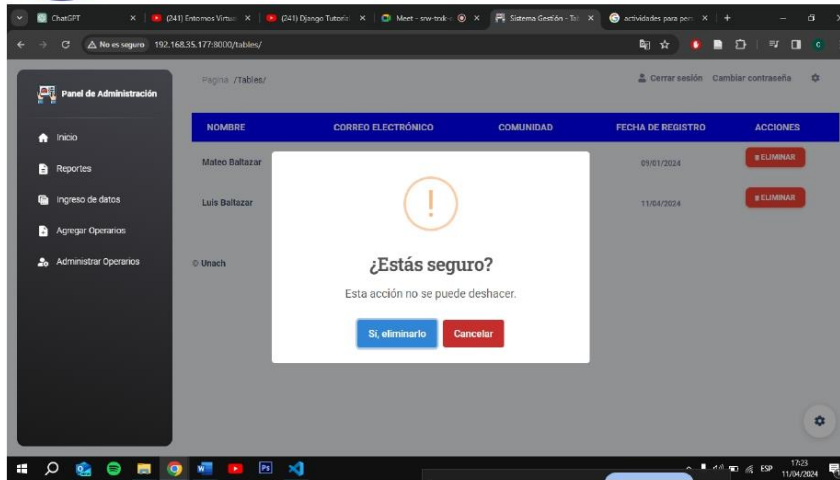
NOMBRE DE USUARIO	NOMBRE	APELLIDOS	FECHA DE REGISTRO
Operario1	Mateo	Baltazar	09/01/2024
Luis1	Luis	Baltazar	11/04/2024

Administración de operario

En el módulo de administración de operarios podremos gestionar a los operarios de la comunidad, se podrá visualizar el nombre, el correo electrónico, comunidad y fecha de registro. Las acciones que se pueden realizar son eliminar a dicho operario de la comunidad.

NOMBRE	CORREO ELECTRÓNICO	COMUNIDAD	FECHA DE REGISTRO	ACCIONES
Mateo Baltazar	Vacio	Totoras	09/01/2024	ELIMINAR
Luis Baltazar	Vacio	Totoras	11/04/2024	ELIMINAR

Al querer eliminar dicho operario se abrirá una ventana emergente donde se deberá confirmar la eliminación del operario de dicha comunidad.

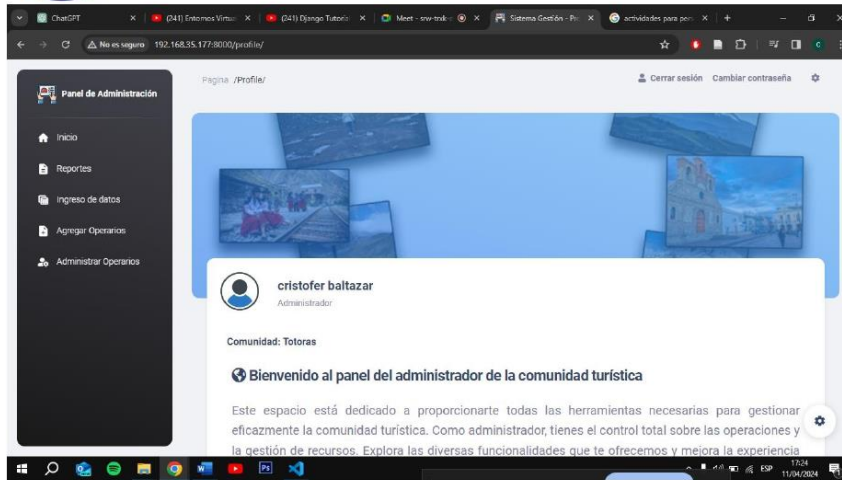


Al confirmar la eliminación del operario se borrará de la lista de los operarios que tiene la comunidad.

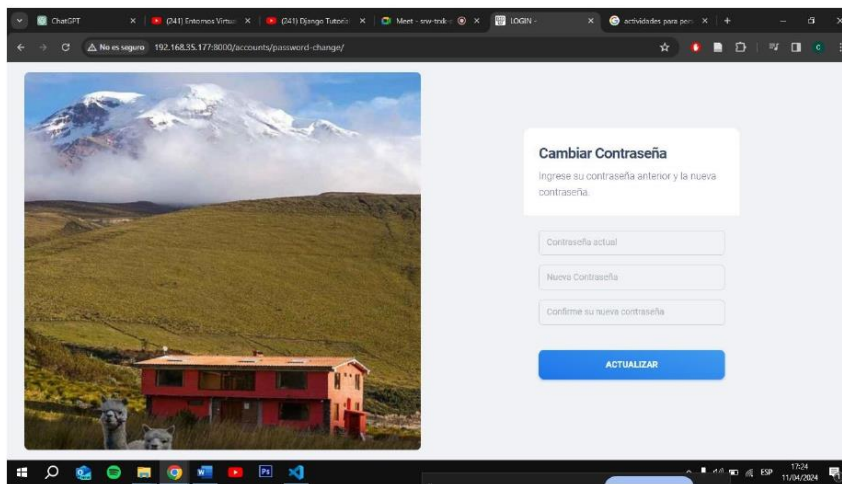


Cambio de contraseña

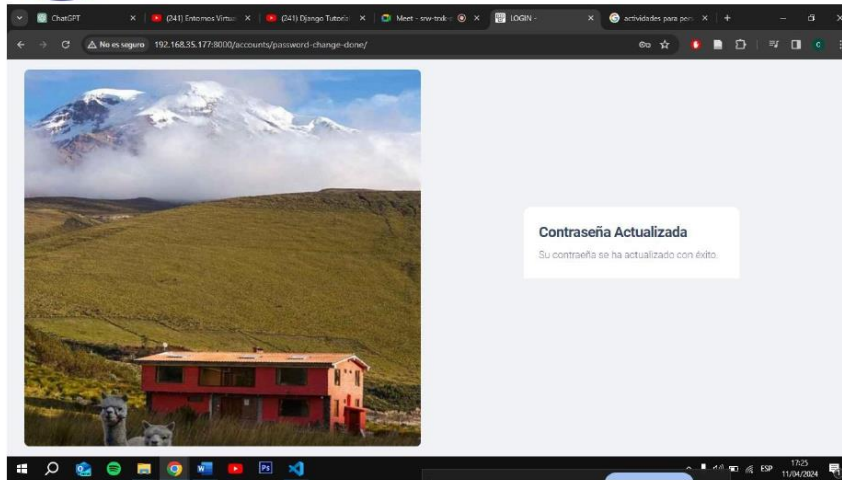
Para el cambio de contraseña se deberá escoger en la parte superior la opción de cambio de contraseña.



El cual nos redirige a la pantalla de cambio de contraseña, para cambiarlo deberemos ingresar la contraseña actual y la nueva contraseña.

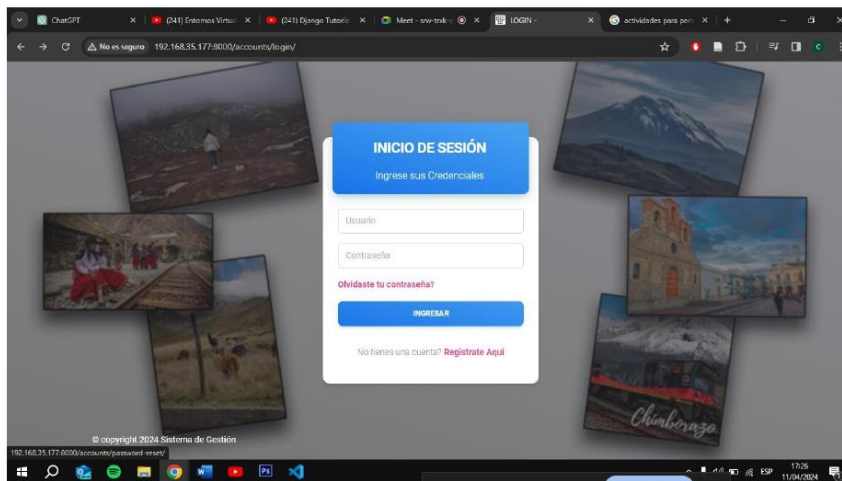


Una vez actualizado se mostrará la venta de contraseña actualizada.

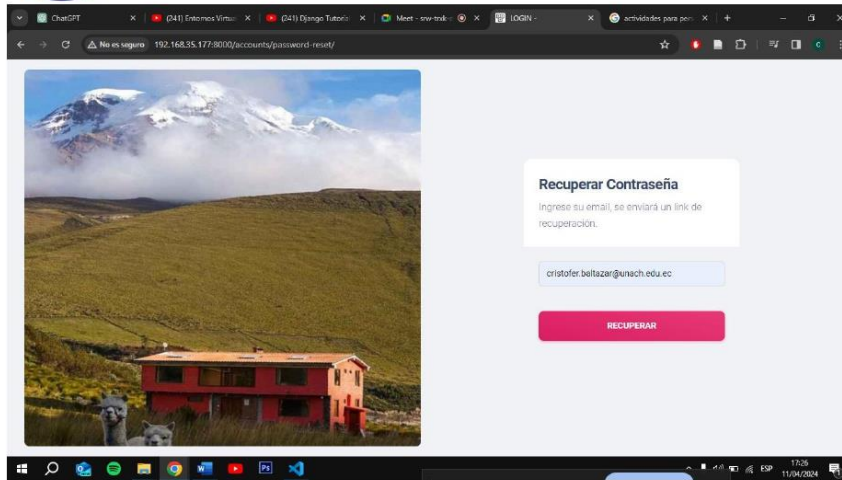


Recuperación de contraseña

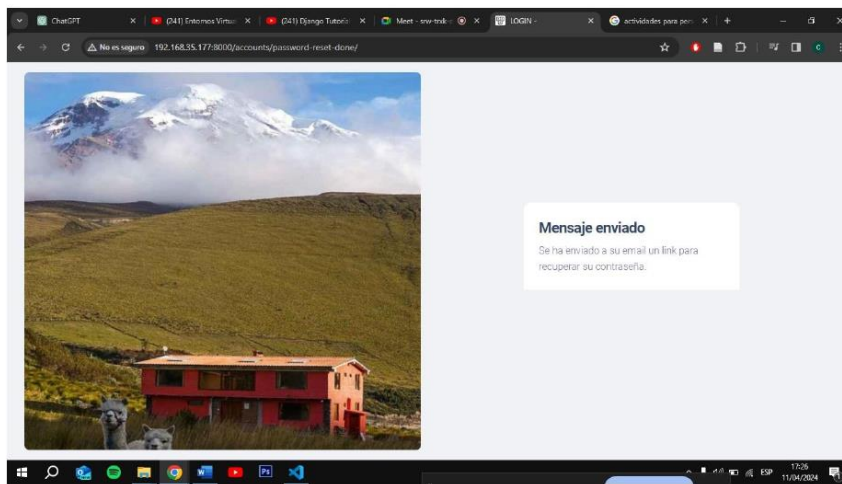
Para recuperar la contraseña se deberá dar clic en olvídate tu contraseña.



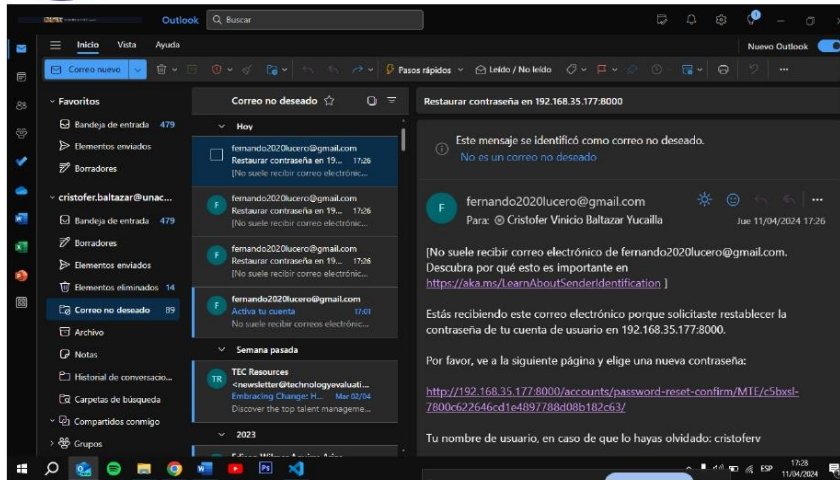
Para recuperar la contraseña se deberá colocar el correo con el cual fue registrado el usuario al cual llegar el link de recuperación de clave.



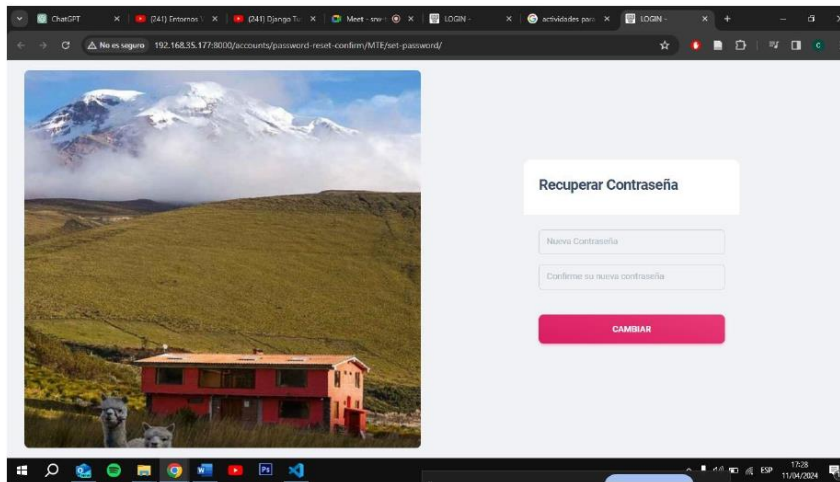
Al dar clic en recuperar se enviara el link de recuperación y mostrara el siguiente mensaje “Mensaje enviado”



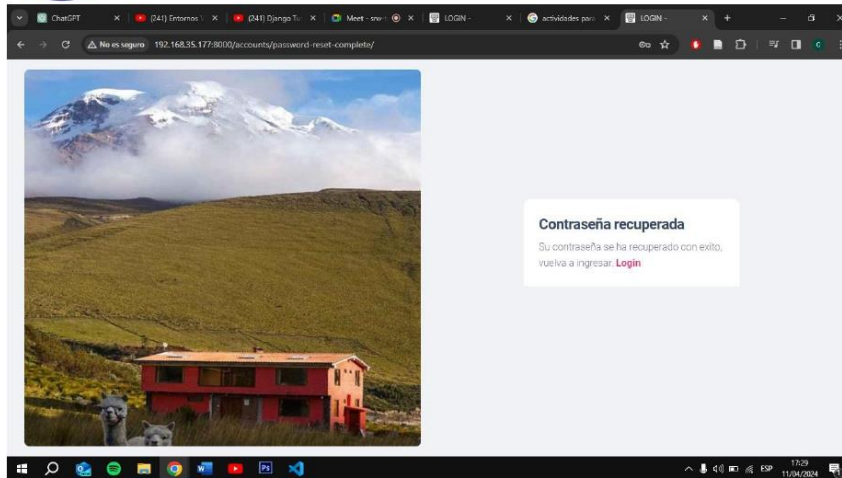
Al correo con el cual fue registrado el usuario, llegará el link de recuperación de contraseña



Al abrir el link se mostrará lo siguiente el cual nos permite recuperar la contraseña, debemos colocar la nueva contraseña.

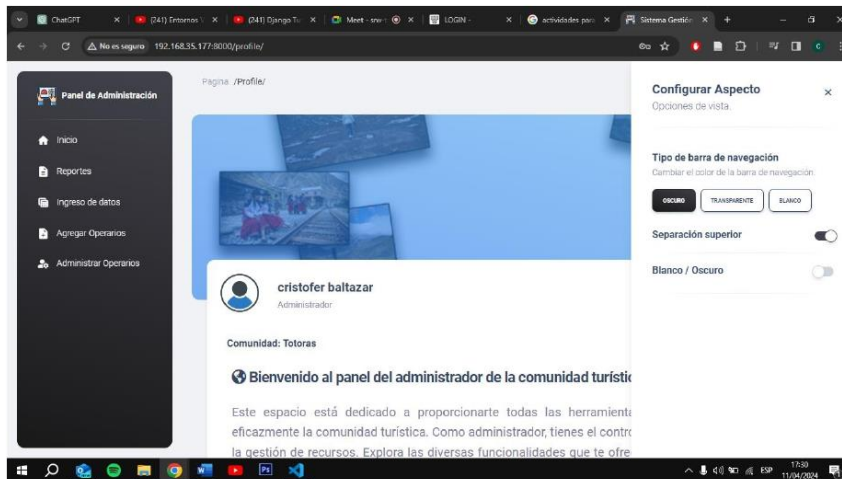


Al enviar la contraseña nueva nos mostrará contraseña recuperada y podremos ingresar al sistema.

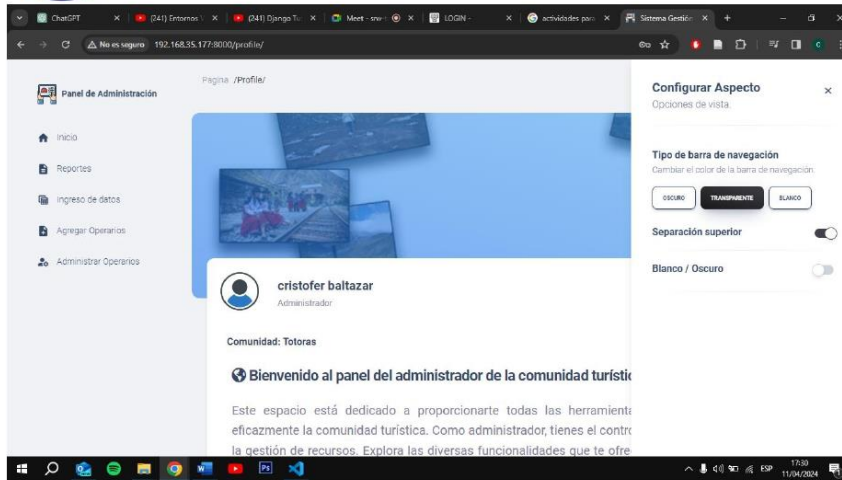


Configuración de aspecto del sistema

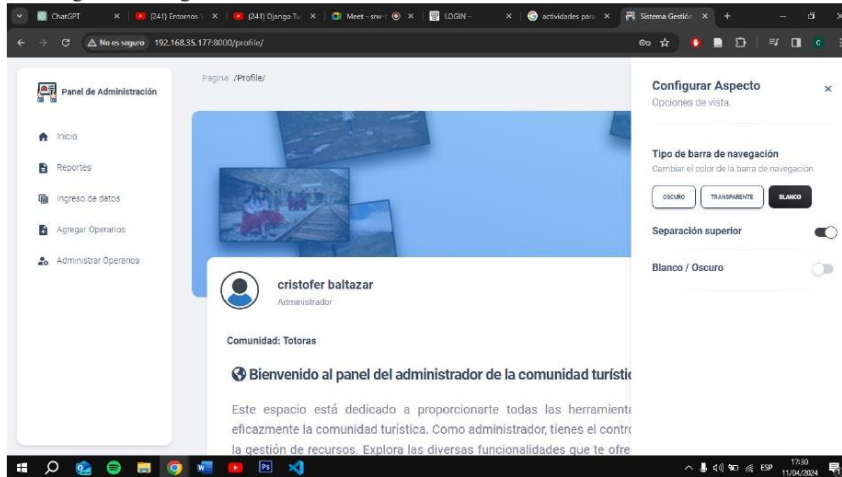
Se podrá personalizar el aspecto del sistema el cual nos muestra todo el tipo de personalización.



Modo transparente, este modo hace transparente a la barra de navegación que se encuentra a la izquierda.

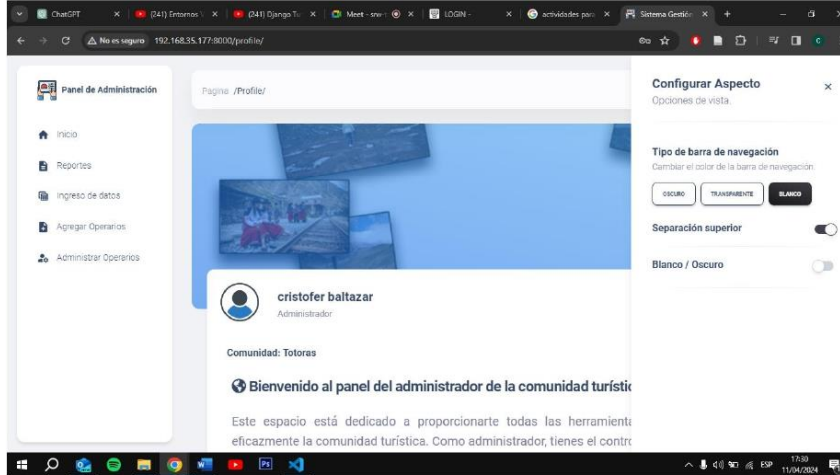


Modo blanco, este modo coloca a la barra de navegación en color blanco, como se muestra en la siguiente imagen.

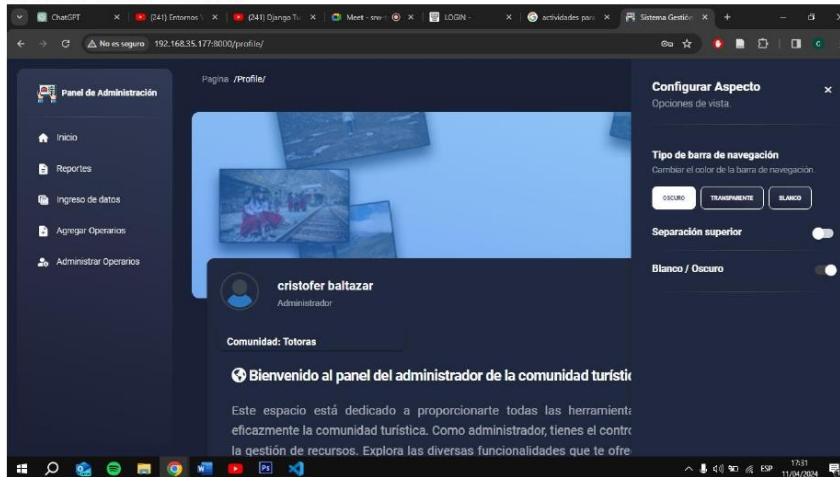




Opción de separación superior, esta opción realiza una separación en la parte superior del sistema.



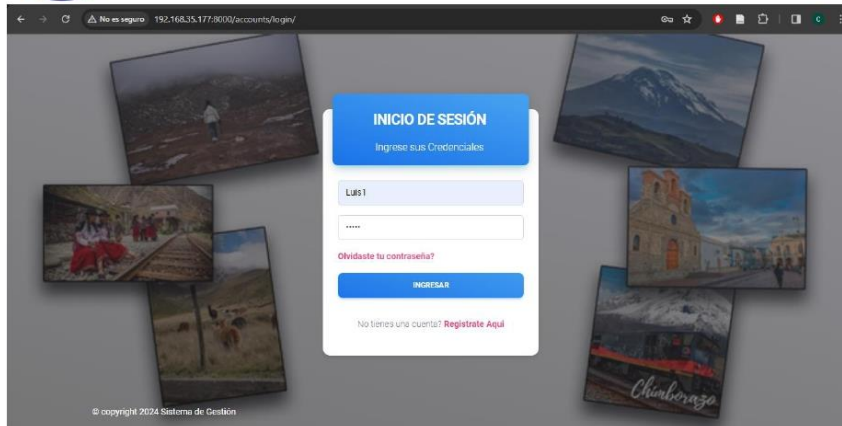
Modo blanco/oscuro, este modo cambia totalmente de aspecto a todo el sistema, poniéndolo en modo oscuro o blanco.



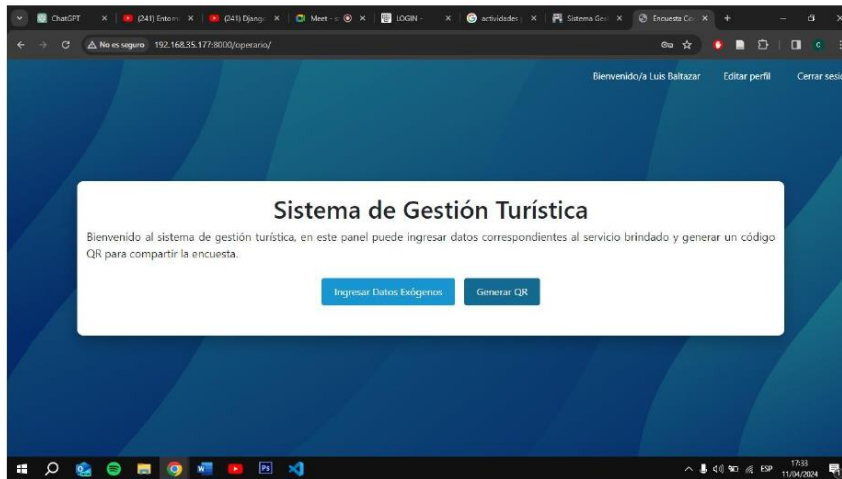
4. Operario

Login operario

Par ingresar al sistema el operario deberá ser creado por el administrador de la comunidad.



Al ingresar al sistema, se mostrará lo siguiente al operario, tendrá 2 opciones ingresar datos exógenos y generar QR.



Ingreso de datos exógenos

Al dar clic en la opción de Ingresar datos exógenos, nos mostrará lo siguiente en el cual se deberá llenar los datos que nos pide el modelo.



Bienvenido/a Luis Baltazar Editar perfil Cerrar sesión

[Ingresar Datos Exógenos](#) [Generar QR](#)

Formulario de datos exógenos de Totoras

¿Cuál es su nombre?

Escriba su número de cédula

Escriba su correo electrónico

Número de acompañantes

¿Por qué medio o canal de difusión conoció este lugar?

Se deberá llenar todo lo que pide el modelo el cual son también calificaciones que se deben llenar.

Bienvenido/a Luis Baltazar Editar perfil Cerrar sesión

¿Que servicios utilizó?

(A&B) Alimentación y Bebidas
Califique el servicio de alimentación y bebidas
 Muy Bueno Bueno Regular Malo Muy Malo

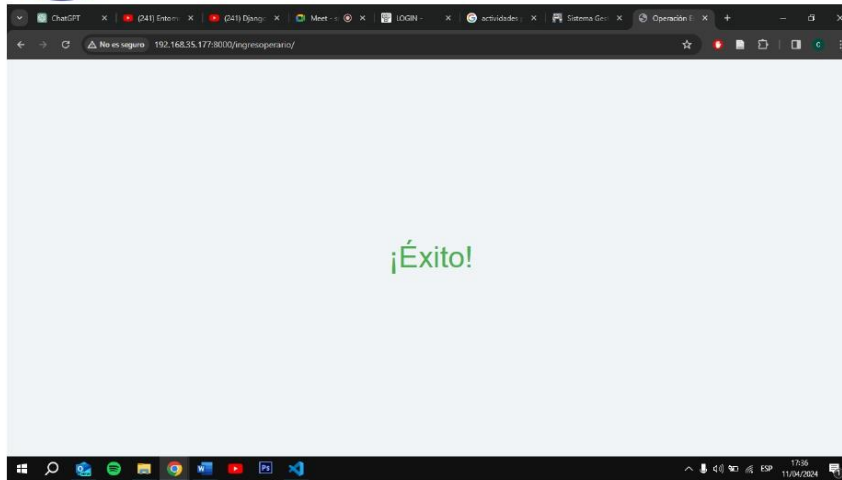
(A1) Alojamiento
Califique el servicio de alojamiento
 Muy Bueno Bueno Regular Malo Muy Malo

(OP) Operación e Intermediación
Califique el servicio de operación e intermediación
 Muy Bueno Bueno Regular Malo Muy Malo

(OE) Organización de Eventos
Califique el servicio de organización de eventos
 Muy Bueno Bueno Regular Malo Muy Malo

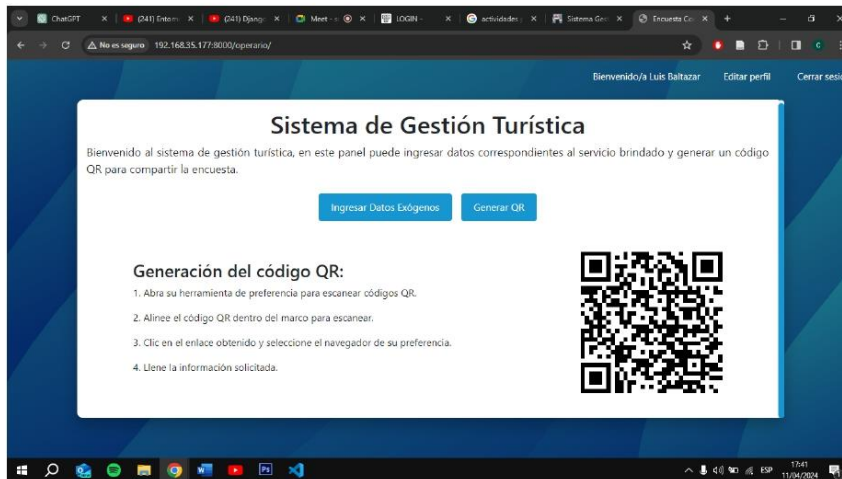
(GT) Guianza Turística
Califique el servicio de guianza turística

Una vez enviado los datos, al operario se le mostrará la siguiente pantalla de éxito.



Generador de QR

Al dar clic en Generar QR se generará un código de QR que se podrá compartir con los turistas el cual podrán ingresar sus propios datos.



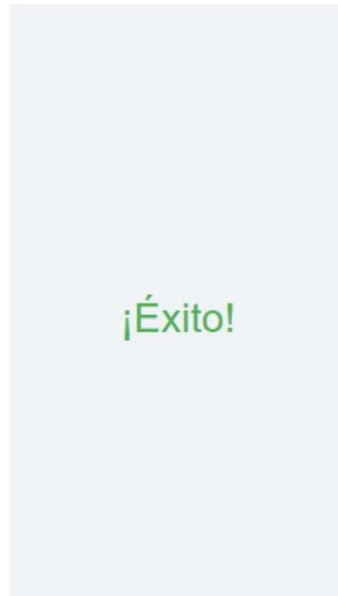
Escaneo de código QR mediante teléfono

Se muestra un ejemplo que al escanear en el móvil se muestra lo siguiente.



Bienvenid@ Turista	Bienvenid@ Turista
Encuesta de datos Exógenos Comunidad:Totoras	Encuesta de datos Exógenos Comunidad:Totoras
¿Cuál es su nombre? <input type="text"/>	¿Cuál es su nombre? <input type="text" value="Fernando"/>
Escriba su número de cedula <input type="text"/>	Escriba su número de cedula <input type="text" value="1867890987"/>
Escriba su correo electrónico <input type="text"/>	Escriba su correo electrónico <input type="text" value="fernando2020lucero@gmail.com"/>
Número de acompañantes <input type="text"/>	Número de acompañantes <input type="text" value="4"/>
¿Por qué medio o canal de difusión conoció este lugar? <input type="checkbox"/> Página web <input type="checkbox"/> Redes Sociales <input type="checkbox"/> Referencias de terceros	¿Por qué medio o canal de difusión conoció este lugar? <input checked="" type="checkbox"/> Página web <input checked="" type="checkbox"/> Redes Sociales <input checked="" type="checkbox"/> Referencias de terceros
¿Qué servicios utilizó? <input type="checkbox"/> (A&B) Alimentación y Bebidas <input type="checkbox"/> (AL) Alojamiento <input type="checkbox"/> (OP) Operación e Intermediación <input type="checkbox"/> (OE) Organización de Eventos <input type="checkbox"/> (GT) Guianza Turística	¿Qué servicios utilizó? <input checked="" type="checkbox"/> (A&B) Alimentación y Bebidas <input checked="" type="checkbox"/> (AL) Alojamiento
<input type="button" value="Enviar"/>	<input type="radio"/> Muy Bueno <input type="radio"/> Bueno <input type="radio"/> Regular <input type="radio"/> Malo <input type="radio"/> Muy Malo <input checked="" type="checkbox"/> (AL) Alojamiento

Al completar la encuesta, se nos mostrará la pantalla de éxito al enviarlo.



Editar perfil operario

El operario podrá editar su información personal y podrá cambiar su contraseña como se muestra en la imagen.

