



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO DISEÑO E IMPLEMENTACIÓN DE UN SERVIDOR QUE PERMITA ESTUDIAR LA EXPERIENCIA DE USUARIO EN APLICACIONES Y WEBS

Autor: Miguel Sánchez-Beato Gil

Director: Francisco Martín Martínez

Madrid

Junio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

DISEÑO E IMPLEMENTACIÓN DE UN SERVIDOR

QUE PERMITA ESTUDIAR LA EXPERIENCIA DE

USUARIO EN APLICACIONES Y WEBS

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Miguel Sánchez-Beato Gil

Fecha: 08/ 06/ 2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Francisco Martín Martínez

Fecha: 08/ 06/ 2025



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO DISEÑO E IMPLEMENTACIÓN DE UN SERVIDOR QUE PERMITA ESTUDIAR LA EXPERIENCIA DE USUARIO EN APLICACIONES Y WEBS

Autor: Miguel Sánchez-Beato Gil

Director: Francisco Martín Martínez

Madrid

Agradecimientos

Me gustaría agradecer en primer lugar a mi tutor Francisco Martín por darme la oportunidad de llevar a cabo este proyecto, estando siempre disponible para prestarme la ayuda que he necesitado.

También quiero agradecer a mi madre, Esmeralda, por estar ahí para mí en cualquier momento y por saber siempre qué decir, incluso cuando ni siquiera yo sabía qué necesitaba, y a mi padre, Javier, por enseñarme todo lo que sabe y ofrecerme su apoyo desde el primer momento, ya que, sin ellos, nada de esto habría sido posible.

Por último, agradecer también a mi mejor amigo, Álvaro González, por aguantarme durante estos años, sobre todo cuando ni yo me aguantaba a mí mismo, y por hacer soportables todas las tardes de estudio, que han sido muchas.

DISEÑO E IMPLEMENTACIÓN DE UN SERVIDOR QUE PERMITA ESTUDIAR LA EXPERIENCIA DE USUARIO EN APLICACIONES Y WEBS

Autor: Sánchez-Beato Gil, Miguel.

Director: Martín Martínez, Francisco.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Este proyecto aborda la analítica web enfocada en la experiencia de usuario, desarrollando una plataforma propia y gratuita para monitorizar la interacción en sitios web. Frente a herramientas comerciales que imponen limitaciones, se propone una solución transparente, ligera y modular que facilita el análisis de datos respetando la privacidad.

Palabras clave: analítica web, experiencia de usuario, API REST.

1. Introducción

El presente proyecto se enmarca en el ámbito de la analítica web con foco en la experiencia de usuario (UX). Su propósito es el diseño e implementación de una plataforma propia y gratuita que permita estudiar cómo interactúan los usuarios con una página o aplicación web. Frente a herramientas como Google Analytics o Matomo, que imponen restricciones de pago o privacidad, este sistema busca ofrecer una alternativa ligera, escalable y fácilmente integrable, priorizando la transparencia de datos y la simplicidad de uso.

2. Definición del proyecto

La herramienta desarrollada se basa en una arquitectura modular de tres capas: frontend, backend y base de datos. La Figura 1 muestra un esquema general del sistema y su funcionamiento.

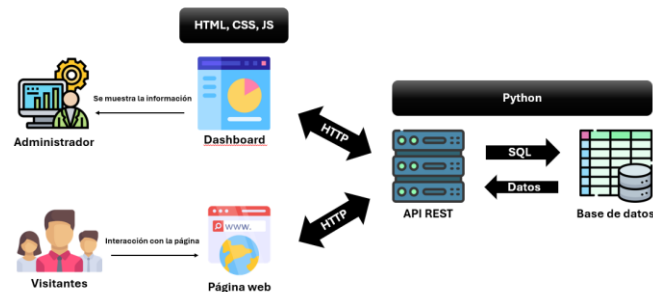


Figura 1 Esquema de la herramienta

En el sistema intervienen dos tipos de usuarios: los visitantes de la página web y el administrador que consulta las estadísticas. Los visitantes interactúan con la web, generando eventos (clics, cambios de página, formularios) que son enviados mediante peticiones HTTP a una API REST desarrollada en Python utilizando Flask. Esta API se encarga de almacenar los datos en una base de datos relacional, gestionada mediante SQLAlchemy.

El administrador accede a un panel implementado con HTML, CSS y JavaScript, que consulta la API REST para recuperar y visualizar los datos almacenados. Esta separación entre el cliente (frontend), el servidor (API) y la base de datos facilita la escalabilidad, el mantenimiento y la futura ampliación del sistema.

El diseño garantiza la independencia entre componentes, permitiendo por ejemplo cambiar la base de datos o actualizar el panel sin afectar al resto de la arquitectura.

3. Descripción del modelo/sistema/herramienta

La base de datos, diseñada con un modelo relacional e implementada en SQLAlchemy, almacena información sobre sesiones, interacciones, formularios, cambios de página y eventos, asegurando integridad y escalabilidad. Se ilustra en la Figura 2:

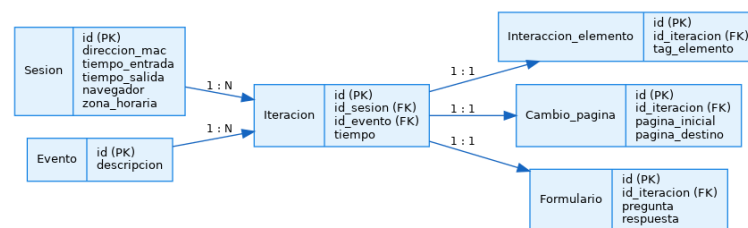


Figura 2 Diagrama de la BBDD

La API REST, desarrollada en Python con Flask, actúa como intermediaria entre la base de datos y la interfaz web, gestionando la recogida de eventos, cálculo de métricas y entrega de datos en formato JSON. La API REST, desarrollada en Python con Flask, actúa como intermediaria entre la base de datos y la interfaz web, gestionando la recogida de eventos, cálculo de métricas y entrega de datos en formato JSON. Los tipos de rutas se muestran en la Figura 3:



Figura 3 Tipos de rutas de API REST

La interfaz web, creada con HTML, CSS y JavaScript, presenta los datos analíticos en paneles claros y accesibles para el administrador, con filtros para analizar sesiones individuales, facilitando la interpretación incluso para usuarios no técnicos. Los paneles disponibles se representan en la Figura 4:

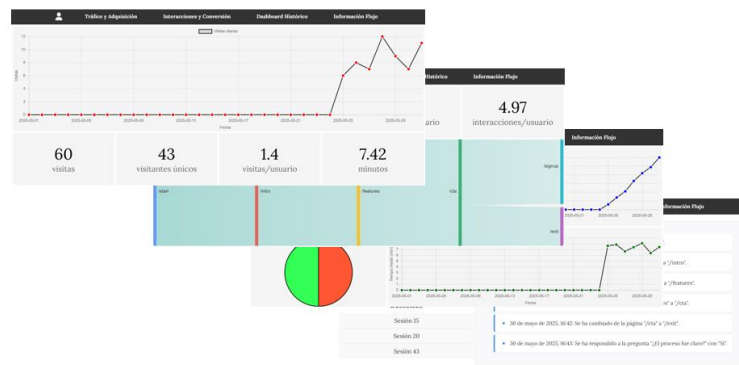


Figura 4 Paneles web

4. Resultados

Para demostrar la utilidad de la herramienta, se ha realizado una simulación, insertando datos artificiales de una página de captación.

Posteriormente, se analizan los múltiples paneles disponibles con información relevante sobre la página web, incluyendo métricas, distintos gráficos e información de cada visitante de forma individual.

5. Conclusiones

La solución presentada destaca por su accesibilidad, su modelo de código abierto y su independencia de terceros. Permite a pequeños negocios, desarrolladores o asociaciones obtener datos críticos sin necesidad de pagar suscripciones ni comprometer la privacidad de sus usuarios. Su diseño modular garantiza la escalabilidad y futuras mejoras, como autenticación, internacionalización o anonimización de datos.

En conclusión, este proyecto proporciona una alternativa real y funcional a las soluciones actuales de analítica web, facilitando el acceso al análisis de experiencia de usuario y fomentando el control local de los datos.

DESIGN AND IMPLEMENTATION OF A SERVER TO STUDY USER EXPERIENCE IN APPLICATIONS AND WEBSITES

Author: Sánchez-Beato Gil, Miguel.

Supervisor: Martín Martínez, Francisco.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This project focuses on web analytics with an emphasis on user experience, developing a proprietary and free platform to monitor interaction on websites. In contrast to commercial tools that impose limitations, it proposes a transparent, lightweight, and modular solution that facilitates data analysis while respecting privacy.

Keywords: web analytics, user experience, REST API.

1. Introduction

This project falls within the field of web analytics with a focus on user experience (UX). Its goal is to design and implement a proprietary and free platform that allows the study of how users interact with a web page or application. Unlike tools such as Google Analytics or Matomo, which impose payment or privacy restrictions, this system aims to provide a lightweight, scalable, and easily integrable alternative, prioritizing data transparency and simplicity of use.

2. Project definition

The developed tool is based on a modular three-layer architecture: frontend, backend, and database. Figure 1 presents a general diagram of the system and how it works.

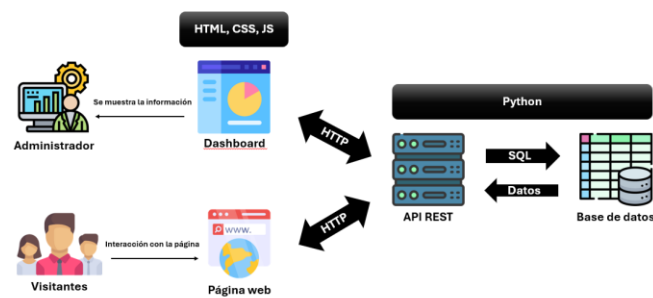


Figure 1 Tool overview

The system involves two types of users: website visitors and the administrator who views the statistics. Visitors interact with the website, generating events (clicks, page changes, forms) that are sent via HTTP requests to a REST API developed in Python using Flask.

This API is responsible for storing the data in a relational database managed with SQLAlchemy.

The administrator accesses a dashboard implemented with HTML, CSS, and JavaScript, which queries the REST API to retrieve and display the stored data. This separation between the client (frontend), the server (API), and the database facilitates scalability, maintenance, and future system expansion.

The design ensures independence between components, allowing, for example, the database to be changed or the dashboard to be updated without affecting the rest of the architecture.

3. System description

The database, designed with a relational model and implemented using SQLAlchemy, stores information about sessions, interactions, forms, page changes, and events, ensuring integrity and scalability. This is illustrated in Figure 2:

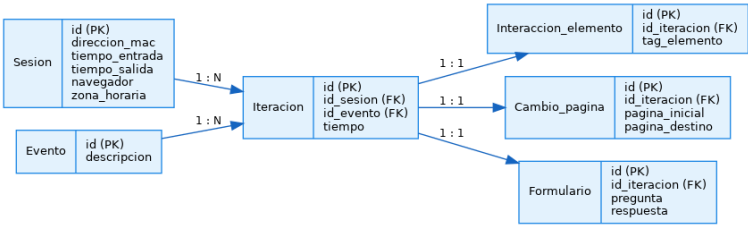


Figure 2 Database scheme

The REST API, developed in Python with Flask, acts as an intermediary between the database and the web interface, managing event collection, metric calculation, and data delivery in JSON format. The types of routes are shown in Figure 3:

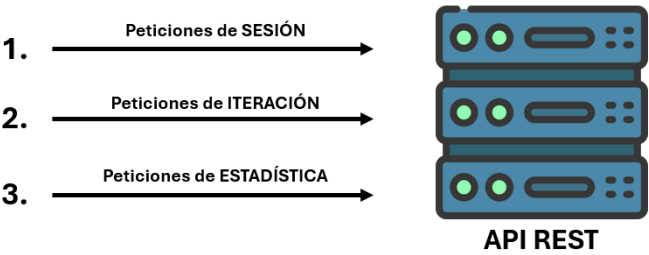


Figure 3 Types of endpoints

The web interface, created with HTML, CSS, and JavaScript, presents the analytical data in clear and accessible panels for the administrator, with filters to analyze individual sessions,

making interpretation easy even for non-technical users. The available panels are shown in Figure 4:

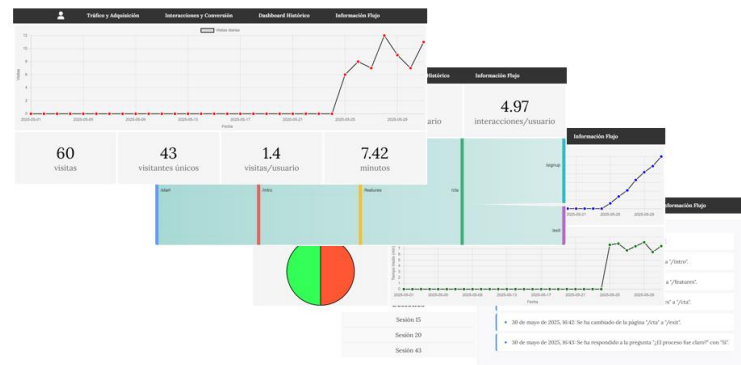


Figure 4 Dashboard panels

4. Results

To demonstrate the usefulness of the tool, a simulation was conducted by inserting artificial data from a landing page. Subsequently, the multiple available panels with relevant information about the website are analyzed, including metrics, various charts, and detailed information about each individual visitor.

5. Conclusions

The presented solution stands out for its accessibility, open-source model, and independence from third parties. It allows small businesses, developers, or organizations to obtain critical data without the need to pay subscriptions or compromise their users' privacy. Its modular design ensures scalability and future enhancements, such as authentication, internationalization, or data anonymization.

In conclusion, this project provides a real and functional alternative to current web analytics solutions, facilitating access to user experience analysis and promoting local data control.

Índice de la memoria

Capítulo 1. Introducción	7
1.0 Motivación del proyecto.....	10
Capítulo 2. Descripción de las Tecnologías.....	13
2.0 Protocolo HTTP	14
2.1 API REST	16
2.2 Bases de datos relacionales	16
2.3 HTML, CSS y JavaScript [9]	18
2.4 React [10]	18
Capítulo 3. Estado de la Cuestión	21
3.0 Google Analytics	21
3.1 Matomo Analytics	23
3.2 Plausible	24
3.3 Comparativa	26
Capítulo 4. Definición del Trabajo	29
4.0 Justificación.....	29
4.0.1 Limitaciones económicas de las herramientas actuales.....	29
4.0.2 Valor diferencial de la herramienta desarrollada	30
4.0.3 Por qué invertir o apostar por esta solución.....	30
4.1 Objetivos	31
4.2 Metodología.....	32
4.3 Planificación y Estimación Económica	33
Capítulo 5. Sistema/Modelo Desarrollado.....	35
5.0 Diseño de base de datos.....	35
5.0.1 Tabla de Sesiones	36
5.0.2 Tabla de iteraciones	38
5.0.3 Tabla de interacciones con elementos.....	40
5.0.4 Tabla de cambios de página.....	41
5.0.5 Tabla de formularios	43
5.0.6 Tabla de eventos	44

5.1	Servidor y comunicaciones	45
5.1.1	Peticiones de sesión.....	45
5.1.2	Peticiones de iteración	47
5.1.3	Peticiones de estadística.....	49
5.2	Web de visualización.....	52
5.2.1	Filtro de usuarios	53
5.2.2	Panel de tráfico y adquisición	55
5.2.3	Panel de interacciones y conversión	56
5.2.4	Panel histórico.....	58
5.2.5	Panel de información de flujo	60
5.3	Caso de uso.....	61
Capítulo 6. Análisis de Resultados.....		63
6.0	Panel de tráfico y adquisición	65
6.1	Panel de interacciones y conversión.....	66
6.2	Panel histórico	67
6.3	Panel de información de flujo.....	69
6.4	Diagnóstico final	70
Capítulo 7. Conclusiones y Trabajos Futuros.....		73
7.0	Grado de cumplimiento de los objetivos	73
7.1	Próximos pasos.....	74
Capítulo 8. Bibliografía.....		77
ANEXO I - ALINEACIÓN CON LOS ODS.....		81
8.0	ODS 8 - Trabajo decente y crecimiento económico [21]	81
8.1	ODS 10 - Reducción de las desigualdades [22]	81

Índice de figuras

Figura 1 Esquema de la herramienta	5
Figura 2 Diagrama de la BBDD	6
Figura 3 Tipos de rutas de API REST	7
Figura 4 Paneles web.....	7
Figura 5 Evolución de búsquedas de 'user experience' [2]	9
Figura 6 Panel antiguo de Google Analytics [3]	10
Figura 7 Diagrama de metodología	13
Figura 8 Protocolo HTTP [5].....	14
Figura 9 Logo de Google Analytics [11].....	21
Figura 10 Panel de Google Analytics [12]	22
Figura 11 Logo de Matomo Analytics [14]	23
Figura 12 Panel de Matomo Analytics [15]	24
Figura 13 Logo de Plausible Analytics [16].....	25
Figura 14 Panel de Plausible Analytics [17]	25
Figura 15 Cronograma de tareas.....	33
Figura 16 Proceso seguido.....	34
Figura 17 Diagrama de la base de datos	36
Figura 18 Página inicial.....	53
Figura 19 Botón de filtrado de usuario	54
Figura 20 Desplegable de usuarios	55
Figura 21 Panel de tráfico y adquisición	55
Figura 22 Gráfico de visitas ampliado.....	56
Figura 23 Panel de interacciones y conversión	57
Figura 24 Porcentaje de usuarios que van de /cta a /exit.....	58
Figura 25 Panel histórico.....	59
Figura 26 Porcentaje de usuarios que usan Google Chrome	59

Figura 27 Vista inicial del panel de información de flujo	60
Figura 28 Panel de información.....	61
Figura 29 Ejemplo de flujo de usuario	62
Figura 30 Esquema de simulación.....	64
Figura 31 Panel de tráfico y adquisición en la implementación.....	65
Figura 32 Panel de interacciones y conversión en la implementación	66
Figura 33 Panel histórico en la implementación	68
Figura 34 Ejemplo de flujo de usuario 1	69
Figura 35 Ejemplo de flujo de usuario 2	70

Índice de tablas

Tabla 1 Tipos de respuestas HTTP	15
Tabla 2 Comparativa de herramientas	27
Tabla 3 Campos de tabla de sesión.....	37
Tabla 4 Campos de la tabla de iteraciones	39
Tabla 5 Campos de la tabla de interacción con elemento.....	40
Tabla 6 Campos de la tabla de cambio de página.....	42
Tabla 7 Campos de la tabla de formularios	43
Tabla 8 Asignaciones de eventos.....	44
Tabla 9 Peticiones de sesión.....	46
Tabla 10 Peticiones de iteración.....	47
Tabla 11 Peticiones de evento	49
Tabla 12 Peticiones de medidas.....	50
Tabla 13 Peticiones de gráficos	51
Tabla 14 Áreas de mejora.....	71
Tabla 15 Grado de cumplimiento de objetivos.....	73

Capítulo 1. INTRODUCCIÓN

A lo largo de la historia reciente, la forma de comunicarse para intercambiar conocimiento o dar servicios ha cambiado drásticamente, debido principalmente a la llegada de Internet. Previamente, el consumo de información se realizaba de forma física o personal, con el llamado *boca a boca*. Posteriormente, se encontró una forma de ofrecer información de forma más eficiente, a través de los periódicos y derivados, que fueron el método predominante hasta la década de 1990, con la invención de la *World Wide Web*, que surgió una nueva manera masiva de llegar a más personas, las páginas web, y que no requería de estar en un lugar específico para acceder a dicha información.

A partir de ese momento, la comunicación sufrió un cambio drástico, pero ésta aún tenía una desventaja crucial: no había manera de obtener *feedback*, ya que los sitios web de entonces eran completamente estáticos, es decir, el usuario simplemente accedía a la información, sin ningún elemento interactivo en ellas y no se analizaba que parte de la información mostrada le era o no de interés. Debido a esto, y junto a la evolución de Internet, las páginas web migraron hacia un estado más dinámico, con muchos más elementos y enlaces hacia otras páginas con el fin de retener al usuario para venderle servicios o que consuma publicidad. Este progreso habilitó una forma de obtener *feedback*, a través de una nueva técnica: la analítica web.

La analítica web es una disciplina que pretende estudiar la experiencia de los usuarios en páginas web, por medio de la medición, procesamiento, análisis y reporte de información relacionada con el sitio web, y extraída directamente de éste. Actualmente, se ha convertido en el estándar, ya que proporciona una ventaja competitiva a la hora de detectar debilidades y corregirlas.

Algunas de las ventajas [1] que la analítica web aporta son:

- Mejora de la experiencia de usuario: permite identificar áreas problemáticas y centrarse en ellas.
- Optimización de la tasa de conversión, que es un indicador esencial en cualquier negocio online, y representa el número de usuarios que se convierten en clientes.
- Apoyo en la toma de decisiones estratégicas.
- Aumento en la eficacia de las campañas de marketing, gracias al análisis de los canales de tráfico de usuarios.
- Permite analizar y entrenar algoritmos de IA para ver elementos son de interés para el usuario y dar un servicio más personalizable (sistemas de recomendación como en plataformas de videos online).

En la Figura 5 se muestra la evolución de la búsqueda de *user experience* en Google a lo largo de los últimos años. Se aprecia una clara tendencia ascendente, que refleja la importancia de la analítica en cualquier espacio donde se permita facilitar el proceso que deben seguir los usuarios. La adopción de un marco donde el usuario está en el centro viene influenciada por el marketing y estrategia empresarial. Representa un cambio de una visión tradicional centrada en los bienes, donde los productos tangibles son el foco principal, a una visión centrada en los servicios que ve el servicio como la base fundamental del intercambio. Esta perspectiva postula que la unidad principal de intercambio no son los bienes, sino la prestación de servicios al usuario.

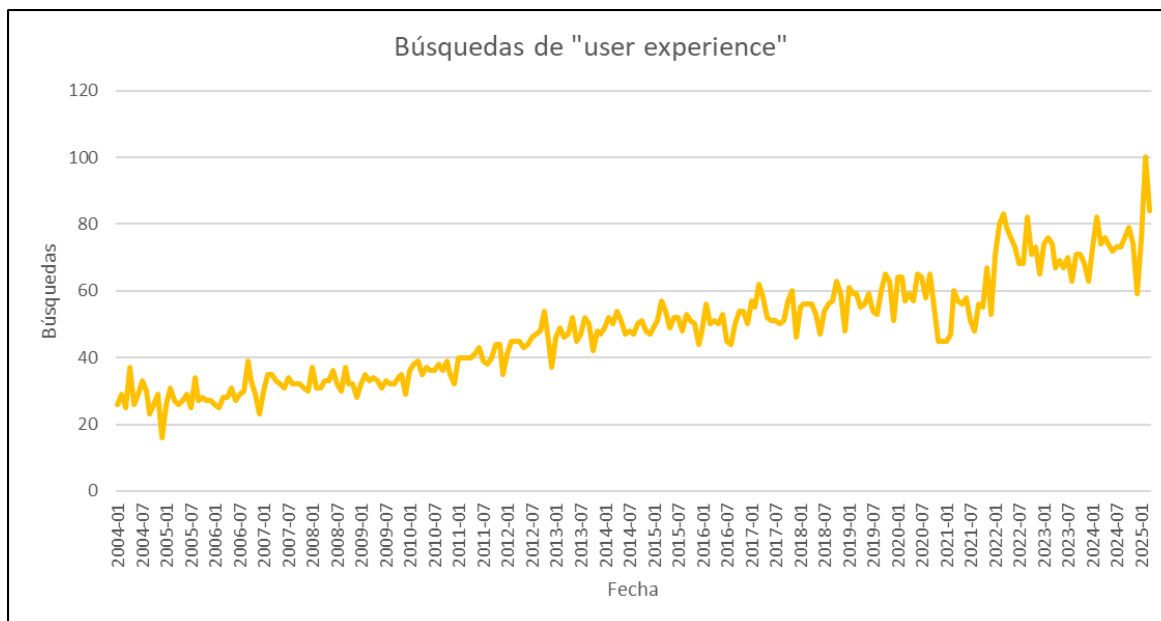


Figura 5 Evolución de búsquedas de 'user experience' [2]

La analítica web se puede realizar a través de herramientas como Google Analytics, Matomo o Plausible. Estas herramientas permiten monitorizar el tráfico de usuarios en una página web, y tienen las siguientes características:

- **Recogida automática de datos:** capta en tiempo real y continuamente el flujo de usuarios de la página.
- **Medición y seguimiento del tráfico:** calcula métricas importantes que permiten cuantificar el paso de los visitantes.
- **Análisis del comportamiento del usuario:** examina el comportamiento de cada usuario individualmente.
- **Configuración y seguimiento de objetivos:** permite establecer eventos clave y seguirlos de cerca.
- **Segmentación de audiencias:** agrupa a los usuarios según diversos criterios para entender mejor algunos perfiles.
- **Visualización y generación de informes:** proporciona paneles e informes personalizables.

1.0 MOTIVACIÓN DEL PROYECTO

Como se ha explicado anteriormente, a lo largo de los años la analítica web se ha convertido en una herramienta fundamental para comprender el flujo de usuarios en entornos online. Actualmente, existen infinidad de aplicaciones que aspiran a resolver este problema, como pueden ser Google Analytics, Matomo o Plausible, que son las plataformas de analítica web por excelencia y que serán descritas más adelante.

En el contexto digital actual, cada vez son más las pequeñas empresas, proyectos personales o iniciativas sociales que deciden crear su propia página web para ofrecer servicios, compartir contenido o simplemente tener presencia en línea. Sin embargo, muchas de estas iniciativas carecen de perfiles técnicos entre sus responsables, lo que dificulta considerablemente el acceso a herramientas de análisis complejas o con interfaces no intuitivas, como la que se muestra en la Figura 6:

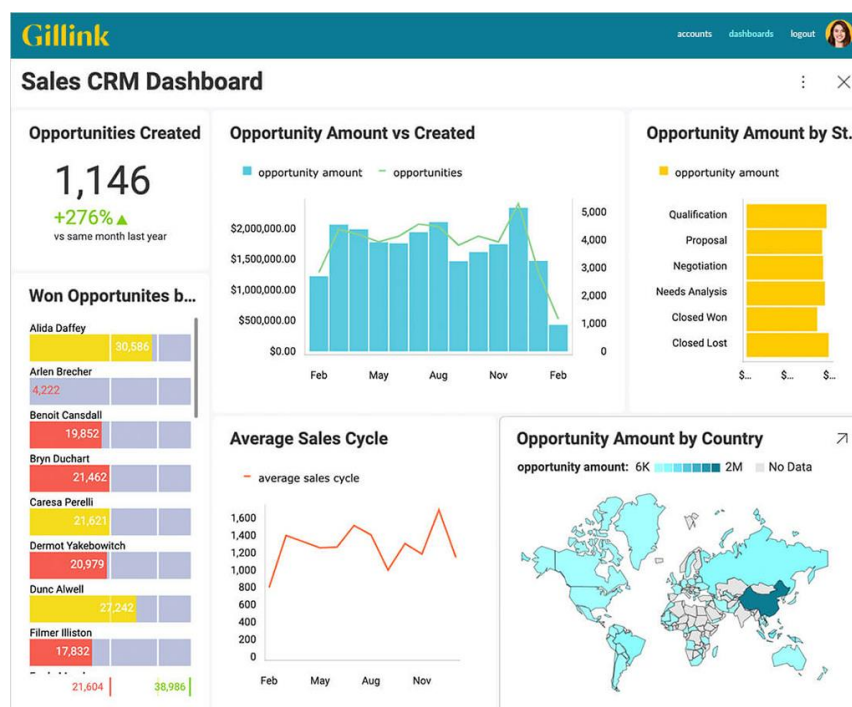


Figura 6 Panel antiguo de Google Analytics [3]

En muchos casos, las empresas que pretenden optimizar sus webs son pequeñas o medianas, y han subcontratado a otra compañía para diseñar la página web. Esto significa que no hay un informático que pueda interpretar los complejos panels que devuelven Google Analytics, Matomo o Plausible. Lo que podría ser una ventaja, el alto nivel de análisis, se convierte así en un inconveniente para las personas poco preparadas.

La necesidad de normalizar el acceso a la información sobre la experiencia de usuario es evidente. No solo se trata de entender qué secciones reciben más tráfico, sino también de tener visibilidad sobre cómo los usuarios navegan, qué les resulta confuso y en qué momento abandonan la página. Estos datos, bien interpretados, permiten tomar decisiones informadas que mejoren tanto la accesibilidad como la experiencia general del usuario, lo que repercute directamente en la calidad de los servicios digitales ofrecidos.

Por este motivo, este proyecto tiene como objetivo ofrecer una alternativa que disponga de una interfaz accesible y fácil de utilizar, y que reúna las características más apreciadas por los usuarios. Pese a su leve complejidad, esta plataforma pretende ofrecer una visión general del comportamiento de la página web, manteniendo la sencillez para facilitar su uso por cualquier tipo de usuario.

En resumen, la motivación social del proyecto parte de una realidad: muchas personas y colectivos quieren mejorar su presencia digital, pero carecen de las herramientas adecuadas. Esta solución pretende reducir esa brecha técnica, facilitando el análisis web a cualquier persona con una mínima formación digital.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

El proyecto utiliza un gran número de tecnologías distintas: lenguajes de programación, protocolos y arquitecturas distintos. En este capítulo se van a detallar todas ellas.

A continuación, en la Figura 7, se muestra un esquema de funcionamiento de la aplicación, con las tecnologías empleadas en cada elemento:

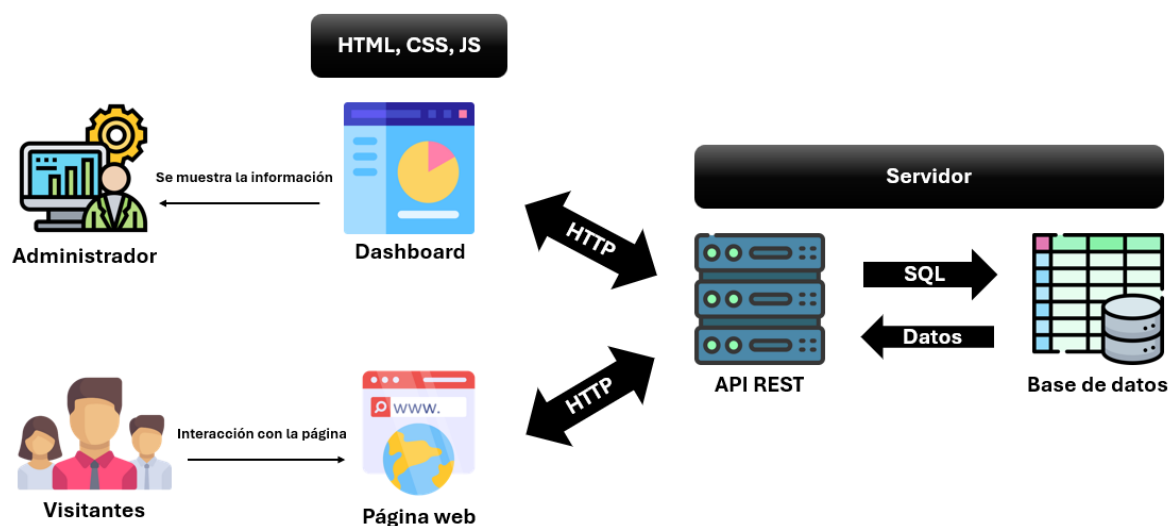


Figura 7 Diagrama de metodología [4], [5], [6], [7]

Se puede observar que, los visitantes hacen uso de la página web, la cual se comunica con la API REST mediante el protocolo HTTP, realizando peticiones que provocan el almacenamiento de información en la base de datos. Esto sucede periódicamente, y cuando el administrador de la página web accede al panel, éste se comunica también con la API REST para extraer los datos necesarios de la base de datos. Idealmente, el acceso para obtener acceso a la información debería tener una clave de acceso para poder limitarlo a administradores.

En este capítulo, se explican brevemente las tecnologías utilizadas.

2.0 PROTOCOLO HTTP

El protocolo HTTP [8] (Hypertext Transference Protocol) es un protocolo de capa de aplicación que permite transferir información entre dispositivos conectados mediante la red. Típicamente, uno de los extremos de la comunicación es el cliente, que realiza peticiones al otro extremo, que se define como el servidor.

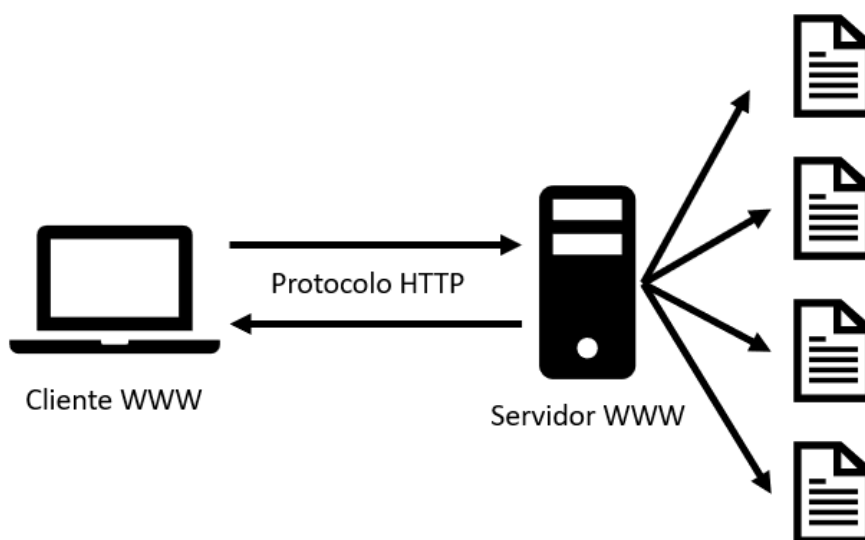


Figura 8 Protocolo HTTP [9]

En la Figura 8 se representa la comunicación entre cliente y servidor mediante el protocolo HTTP.

Una solicitud HTTP es el método que tiene el cliente de pedir información al servidor, escribir información en el servidor o borrar información ya existente. Una solicitud tiene 3 partes principales:

- Método: indica la acción que se quiere realizar. Hay múltiples opciones, pero las principales son:
 - GET: se espera recibir información del servidor.
 - POST: se espera escribir información en el servidor.
 - PUT: se espera modificar información ya existente en el servidor.

- DELETE: se espera eliminar información del servidor.
- Encabezados: almacenan información en el formato de clave-valor. Contienen información básica acerca de la solicitud.
- Cuerpo: contiene la información que se envía al servidor. Dependiendo del método, se utilizará el cuerpo de diferente forma.

Las respuestas HTTP se envían del servidor al cliente, después de haber recibido una solicitud. La parte más importante de las respuestas es el código de estado, que indica la resolución de la solicitud, y viene acompañado de un breve texto explicativo dirigido al usuario. Los tipos de códigos se muestran en la siguiente tabla (Tabla 1):

Tabla 1 Tipos de respuestas HTTP

<i>Identificador numérico</i>	<i>Tipo de código</i>
1xx	Códigos informativos
2xx	Códigos de éxito
3xx	Códigos de redirección
4xx	Códigos de error de cliente
5xx	Códigos de error del servidor

2.1 API REST

Una API [10] (Interfaz de Programación de Aplicaciones) comprende las aplicaciones y protocolos que permiten integrar los sistemas de software de las aplicaciones, es decir, permite interactuar con un sistema operativo para recopilar información o ejecutar una función.

Como se ha explicado, una API es una arquitectura general, que tiene múltiples implementaciones; una de ellas es REST (Transferencia de Estado Representacional), en la que las solicitudes contienen una representación del estado del recurso solicitado en formato JSON, HTML, XML, Python, PHP o texto. De todos ellos, el formato más común es el JSON (JavaScript Object Notation), ya que es interpretable por las personas, lo que facilita la accesibilidad.

Las características que una API REST debe tener son:

- Una arquitectura cliente-servidor, utilizando el protocolo HTTP.
- Independencia de solicitudes, sin almacenamiento del estado de la solicitud, tanto en cliente como en servidor.
- Optimización de la comunicación cliente-servidor mediante datos en caché.
- Estandarización en la transferencia de elementos.
- Un sistema jerárquico que clasifica para cada cliente los diferentes tipos de servidores.

2.2 BASES DE DATOS RELACIONALES

Las bases de datos relacionales [11] son estructuras de almacenamiento de información en las que los datos se representan en tablas con unas reglas específicas. Cada fila de las tablas se denomina "registro", y tiene un identificador único dentro de la tabla. Cada columna es un atributo que define una propiedad específica acerca de cada registro.

Para garantizar la integridad de los datos, se pueden aplicar restricciones sobre los datos introducidos en la base de datos, por ejemplo:

- Restricción de unicidad para cierto atributo, resultando que, dentro de una misma tabla, no pueda haber más de un registro con el mismo valor de ese atributo. Esta restricción se presupone para los identificadores.
- Restricción de nulidad. Normalmente, se permiten valores nulos en algunas columnas, ya que no siempre se tiene toda la información dentro de un registro, pero en ciertas ocasiones, cuando se trabaja con herramientas que no interpretan correctamente datos nulos, conviene restringirlos.

Estas bases de datos siguen el modelo relacional, el cual define un método de creación de tablas, que permitan posteriormente relacionarlas entre sí. Esto se realiza mediante dos tipos de claves:

- Clave primaria: esta clave suele ser el identificador numérico, y debe ser única para cada registro. Identifica inequívocamente cada registro.
- Clave foránea: se aplica a una columna de una tabla, y apunta a la clave primaria de otra tabla, indicando que dicha tabla contiene información relacionada con el registro cuyo valor de la clave foránea coincida con el de la primaria.

La relación entre tablas [12] puede ser de múltiples tipos, dependiendo de si las relaciones son unidireccionales o bidireccionales:

- Relación *one to one*: cada registro de la tabla foránea está asociado a un único registro en la tabla primaria.
- Relación *one to many*: cada registro de la tabla foránea está asociado a varios registros en la tabla primaria.
- Relación *many to many*: múltiples registros de la tabla foránea están asociados a múltiples registros en la tabla primaria.
- Relación *many to one*: múltiples registros de la tabla foránea pueden asociarse a un único registro de la tabla primaria.

2.3 HTML, CSS Y JAVASCRIPT [13]

HTML, CSS y JavaScript son las tres tecnologías fundamentales para el desarrollo del frontend de aplicaciones web.

- HTML (*HyperText Markup Language*) es el lenguaje que define la estructura del contenido en una página web. Permite organizar la información en elementos, actuando como el esqueleto del sitio.
- CSS (*Cascading Style Sheets*) se utiliza para aplicar estilos a los elementos definidos en HTML. A través de CSS se controla la apariencia visual de la interfaz.
- JavaScript, por su parte, es un lenguaje de programación que permite dotar a la página de interactividad. Con JavaScript se pueden capturar acciones del usuario, modificar elementos en tiempo real, realizar peticiones a servidores y actualizar el contenido sin necesidad de recargar la página.

En conjunto, estas tecnologías permiten construir interfaces web completas: HTML define el contenido, CSS lo personaliza y JavaScript lo hace interactivo y funcional.

2.4 REACT [14]

React es una biblioteca de JavaScript desarrollada por Meta que se utiliza para construir interfaces de usuario dinámicas y eficientes. Su principal característica es la capacidad de dividir la interfaz en componentes reutilizables.

En lugar de escribir instrucciones paso a paso para modificar el DOM (*Document Object Model*), el desarrollador describe cómo debería verse la interfaz en cada momento, y React la actualiza de forma eficiente cuando cambian los datos.

Además, React permite gestionar el estado de la aplicación y reaccionar automáticamente a cambios en los datos, lo que lo convierte en una herramienta potente para desarrollar interfaces interactivas, escalables y mantenibles.

Capítulo 3. ESTADO DE LA CUESTIÓN

Este capítulo pretende definir el estado del arte en el sector de la analítica web. Hay infinidad de herramientas de analítica de datos, y cada una está diseñada para hacer frente a distintos tipos de datos (web, videojuegos...). Esta herramienta pretende reunir los mejores aspectos de todas las opciones disponibles, y para ello se han tomado tres como referencia: Matomo, Google Analytics y Plausible.

3.0 *GOOGLE ANALYTICS*

Google Analytics es una herramienta de marketing digital lanzada por Google en el año 2005, y que facilita a sus usuarios información agrupada del tráfico en sus páginas web. Su logo se muestra en la Figura 9:



Figura 9 Logo de Google Analytics [15]

La versión inicial contenía un número limitado de funciones, pero el lanzamiento completo de la aplicación fue en 2014 con la actualización de “Universal Analytics”, que ampliaba la información disponible. Se observa un panel de ejemplo en la Figura 10.

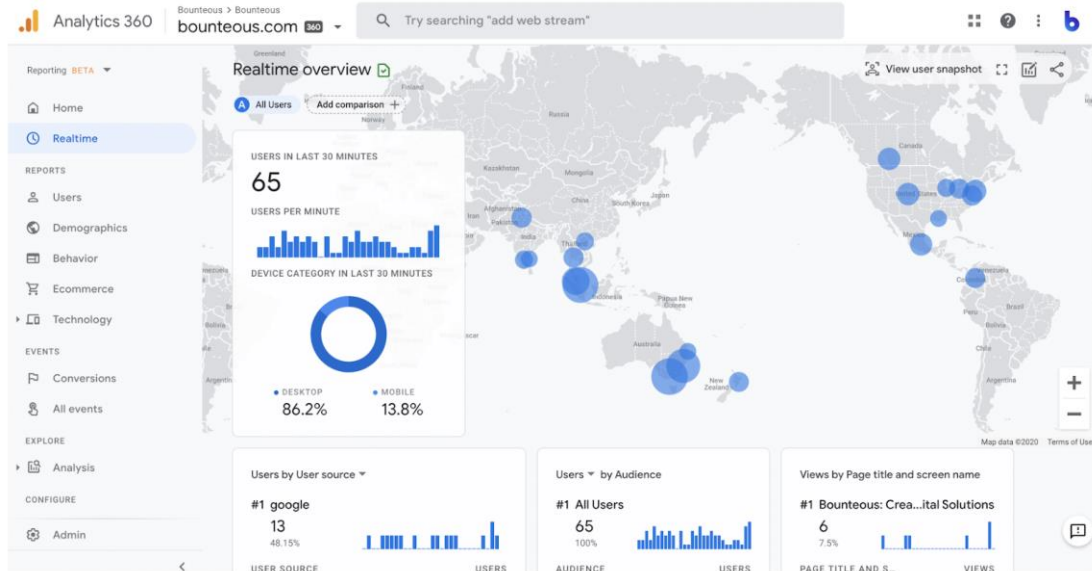


Figura 10 Panel de Google Analytics [16]

Incluso antes de la actualización, Google Analytics siempre ha sido una herramienta con una amplia variedad de funciones, y que es fácil de implementar en el código JavaScript. Desde siempre ha sido la herramienta de análisis web por excelencia, monopolizando el servicio. Posteriormente, se lanzó una nueva actualización: Google Analytics 4 [17], que incluía las siguientes novedades:

- Integración de análisis web y API en un único entorno.
- Con la reducción de uso de las cookies, los métodos de recogida de datos se orientan hacia:
 - o Fuentes de terceros.
 - o Cookies existentes.
 - o Modelos predictivos de *machine learning* con la información proveniente de las fuentes anteriores.
- Análisis *CrossDevice*, es decir, continuar el seguimiento de un usuario a través de diferentes dispositivos.
- Nueva interfaz, con informes actualizados y usabilidad renovada.

- Aumento de la facilidad para monitorizar eventos personalizados.

Cabe mencionar que, al contrario que la mayor parte de sus competidores, Google Analytics es una herramienta gratuita, lo cual le otorga una gran ventaja competitiva, y contribuye a recuperar usuarios desde una serie de cambios que se realizaron, que no gustaron al público y permitió la popularización de una de sus principales alternativas: Matomo Analytics.

3.1 MATOMO ANALYTICS

Matomo Analytics es una aplicación web creada por un grupo internacional de diseñadores. Fue creada en 2007 y, hasta 2018, era conocida como Piwik. Su logo es el siguiente (Figura 11):



Figura 11 Logo de Matomo Analytics [18]

La motivación para crear Matomo fue ofrecer a los usuarios una alternativa de código abierto a Google Analytics, que además respetara la privacidad de los datos. A continuación, un ejemplo de panel en la Figura 12

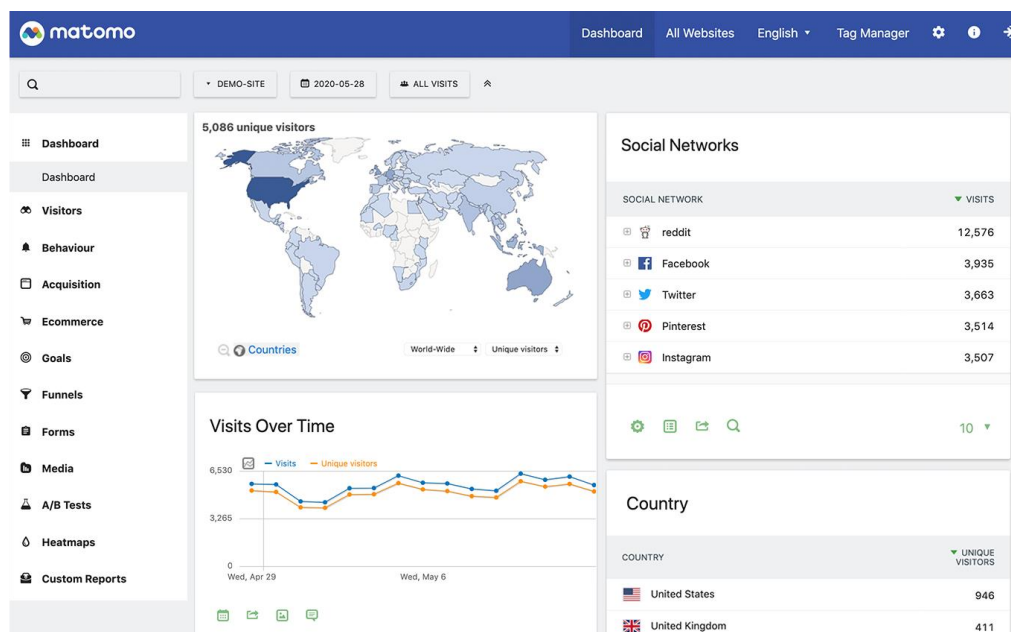


Figura 12 Panel de Matomo Analytics [19]

Matomo, en cambio, tiene múltiples planes de pago, que se ajustan a tus necesidades y al número de visitas. Es, con diferencia, la segunda herramienta de análisis web más usada, y permite importar datos de Google Analytics. Destaca por su interfaz fácil de usar, y su amplio control de los datos, pero su desventaja más importante es la limitación de la granularidad de los datos ante un gran número de visitas.

3.2 PLAUSIBLE

Por último, y siendo con diferencia la opción menos popular de las tres, queda Plausible, que es una herramienta alojada en Europa, relativamente nueva, habiendo sido lanzada al mercado en 2019, y que pretende crear una alternativa de código abierto y que respete la privacidad de los usuarios, manteniendo sus datos en sus propias manos. En la Figura 13 se representa su logo:



Figura 13 Logo de Plausible Analytics [20]

En la Figura 14 se muestra cómo Plausible Analytics representa la información en forma de un panel.

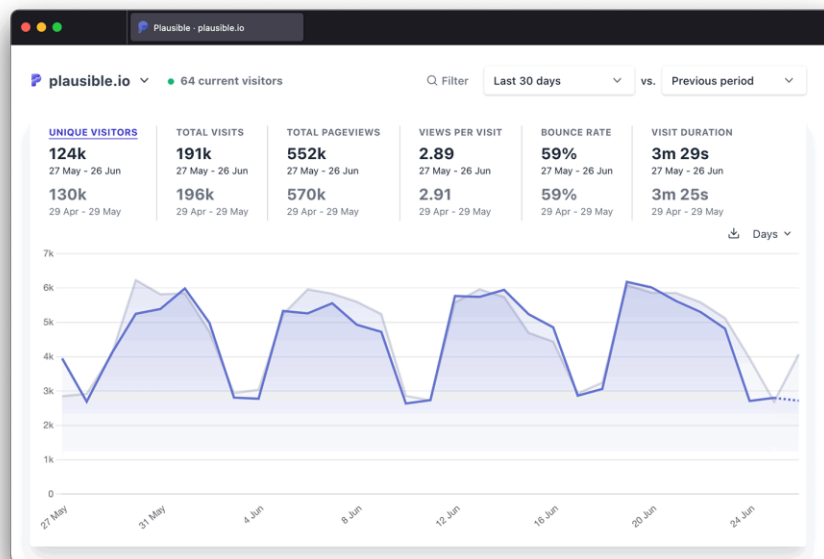


Figura 14 Panel de Plausible Analytics [21]

Su *marketing* está basado en ser la opción más simple y sencilla de todas, comparándose tanto con Google Analytics Figura 3 y Matomo, y ofreciendo un rango más limitado de funciones. Esta herramienta también ofrece múltiples planes dependiendo del número de visitas y de las necesidades y funciones que sean necesarias.

3.3 COMPARATIVA

Para decidir cuáles serán las funciones que implementar en el Trabajo de Fin de Grado, se ha hecho un estudio de las características de las 3 herramientas presentadas anteriormente, y se han elegido las opciones mejor valoradas tanto por las empresas como por los particulares que usarán la aplicación.

Las funciones más relevantes que de las tres aplicaciones, y que serán implementadas en el Trabajo de Fin de Grado se muestran en la Tabla 2. La tabla recopila todas las funciones que incluirá el Trabajo de Fin de Grado, e indica cuáles de las otras aplicaciones también las incorporan. Hay tanto funciones técnicas (embudos, cohortes de usuarios...) como características de la herramienta (código abierto, script ligero, facilidad de uso).

A la hora de decidir las funciones a implementar en el trabajo, se ha pensado en crear una cuarta alternativa que reúna las funciones más demandadas, mezclando los mejores aspectos de las 3 herramientas tomadas en consideración, y manteniendo la accesibilidad y la sencillez.

Tabla 2 Comparativa de herramientas

FUNCIÓN	DESCRIPCIÓN	GOOGLE ANALYTICS	MATOMO	PLAUSIBLE	TFG
Actualizaciones de datos en tiempo real	Observa el flujo de visitas a tu sitio web en tiempo real. Obtén una vista detallada de tus visitantes, las páginas que han visitado y los objetivos que han activado.	X	X	X	X
Seguimiento de eventos	Mida cualquier interacción de los usuarios en sus sitios web y aplicaciones.	X	X	X	X
Embudos	Un embudo es una serie de acciones que espera que realicen sus visitantes a medida que avanzan en su sitio web. Al definir un embudo, puede mejorar sus tasas de conversión, ventas e ingresos, ya que puede determinar dónde pierde a sus visitantes mientras navegan por su sitio web.	X	X	X	X
Cohortes de usuarios	Las cohortes de usuarios le permiten ver y comparar el comportamiento de diferentes cohortes a lo largo del tiempo para comprender qué tan bien retiene e involucra a sus visitantes.	X	X	X	X
Análisis de búsqueda del sitio	Realice un seguimiento de las búsquedas realizadas en su motor de búsqueda interno.		X		X
Alertas personalizadas	Con Alertas Personalizadas puedes monitorear tu negocio y recibir alertas en tu teléfono o por correo electrónico cuando alguno de tus Indicadores Clave de Desempeño aumente o disminuya.		X		X
Exploración retrospectiva de rutas	Examina los datos que se registraron antes de que se produjera una conversión u otro evento clave para analizar diferencias, tendencias o patrones en las rutas que los usuarios siguieron antes de realizar el evento en cuestión.	X			X
Código abierto	El código de la aplicación es público, lo cual permite la modificación y la colaboración de los usuarios.		X	X	X
Script ligero	El tamaño del script afecta al tiempo de procesamiento de la página web/aplicación.			X	X
Fácil de usar	Se prioriza la simplicidad y el entendimiento de los outputs de la aplicación.			X	X

Capítulo 4. DEFINICIÓN DEL TRABAJO

Este capítulo cubre el proceso de trabajo que se ha seguido para desarrollar la herramienta, además de definir su valor para potenciales clientes o inversores.

4.0 JUSTIFICACIÓN

La sostenibilidad económica es uno de los principales retos a los que se enfrentan los proyectos digitales emergentes. En un mercado cada vez más saturado, el acceso a herramientas tecnológicas de calidad suele estar condicionado por modelos de suscripción o licencias de pago, que resultan inasumibles para muchos particulares, *startups* o pequeñas empresas.

Este proyecto surge como respuesta a esa necesidad: ofrecer una solución de analítica web potente, gratuita y accesible, que permita a cualquier usuario obtener información clave sobre el comportamiento de los visitantes en su sitio web sin depender de plataformas de terceros ni asumir costes recurrentes. A continuación, se detalla por qué esta herramienta representa una alternativa económica real frente a las opciones existentes en el mercado.

4.0.1 LIMITACIONES ECONÓMICAS DE LAS HERRAMIENTAS ACTUALES

Actualmente, las herramientas más populares del mercado (Google Analytics, Matomo y Plausible) presentan barreras económicas importantes. Google Analytics, si bien es gratuito, ha generado controversia por el uso que hace de los datos personales de los usuarios, lo que ha llevado a muchas organizaciones a buscar alternativas más respetuosas con la privacidad. Por otro lado, tanto Matomo como Plausible ofrecen versiones gratuitas muy limitadas y se apoyan en planes de pago que escalan en función del tráfico, el número de usuarios o los informes generados.

Este modelo de negocio penaliza directamente a aquellos proyectos que están en fase de crecimiento, como startups, pequeños negocios, desarrolladores individuales o asociaciones sin ánimo de lucro. Para ellos, asumir estos costes mensuales desde el inicio puede suponer un freno importante, impidiendo el acceso a herramientas de analítica avanzadas que sí están al alcance de empresas con mayores recursos.

4.0.2 VALOR DIFERENCIAL DE LA HERRAMIENTA DESARROLLADA

La solución planteada en este TFG parte de una premisa clara: ofrecer una alternativa gratuita, abierta y funcional que cubra las necesidades esenciales de análisis web. Esta herramienta permite:

1. Registrar visitas e interacciones de los usuarios.
2. Visualizar datos clave en paneles simples e intuitivos.
3. Mantener el control local sobre los datos sin cederlos a terceros.

Además, al estar basada en tecnologías libres y ampliamente conocidas (Python, Flask, HTML/CSS/JS, SQLAlchemy), la herramienta es fácil de desplegar, escalar y personalizar. No depende de licencias, ni de servidores propietarios, ni de cuotas. Esto aporta una ventaja económica inmediata, pero también una oportunidad a largo plazo: permite crecer sin cambiar de plataforma ni aumentar costes.

4.0.3 POR QUÉ INVERTIR O APOSTAR POR ESTA SOLUCIÓN

Porque resuelve un problema real con una solución funcional, clara y económicamente viable.

Desde el punto de vista comercial, representa una propuesta de valor que combina tres elementos clave:

1. Coste cero para su adopción e implantación inicial.
2. Escalabilidad sin depender de tarifas externas.
3. Adaptación total a las necesidades del cliente o usuario.

Es una solución pensada para un mercado desatendido: el de los pequeños proyectos que no pueden pagar Matomo ni Plausible, pero que tampoco quieren depender de Google. Por eso, representa una oportunidad para facilitar el acceso a la analítica web, y supone una alternativa real para aquellos que quieren controlar sus datos, analizar el comportamiento de sus usuarios y tomar decisiones informadas sin hipotecar su presupuesto.

4.1 OBJETIVOS

El proyecto pretende diseñar un servidor que proporcione una API y un servicio similar a Matomo o Google Analytics. El proyecto implica crear una plataforma para rastrear y analizar datos de tráfico de manejo de distintas aplicaciones. Esto requiere diseñar una API RESTful, elegir tecnologías como una base de datos y un lenguaje de programación, y garantizar la escalabilidad y la seguridad del sistema.

Los objetivos del proyecto se han fijado teniendo en cuenta la motivación, explicada en la sección anterior, pero se especifica más sobre las características técnicas de la herramienta.

Las principales metas son:

- Desarrollo de un sistema capaz de medir el tráfico de los usuarios en una página web o videojuego, a través de incorporación de código.
- Creación de un panel web accesible, fácil de entender y que resuma la información de forma útil para el cliente incorporando granularidad.
- Diseño de un script optimizado, sin redundancia en el código, y que ralentice lo menos posible la plataforma sobre la que mide.
- Diseño de una base de datos relacional, con relaciones bien definidas entre las variables y las tablas.
- Implementación correcta en el ámbito del apps o webs, haciendo las llamadas pertinentes a la API, y mostrando las estadísticas en tiempo real en el panel web.

4.2 METODOLOGÍA

El desarrollo del proyecto se ha llevado a cabo siguiendo una metodología incremental y estructurada en tres etapas fundamentales, que reflejan la arquitectura lógica de una aplicación web moderna: persistencia de datos, construcción de una interfaz de programación de aplicaciones (API) y diseño de una interfaz de usuario interactiva.

En primer lugar, se diseñó e implementó la base de datos, utilizando Python, en específico la librería SQLAlchemy. Esta herramienta, basada en el paradigma ORM (Object-Relational Mapping), permitió definir las entidades y relaciones del modelo de datos mediante clases y objetos, facilitando así una integración coherente con el resto del sistema.

Una vez establecida la base de datos, se procedió al desarrollo de una API REST utilizando el framework Flask, también en Python. Esta API cumple la función de servir de puente entre la base de datos y la interfaz de usuario, permitiendo acceder a los datos de forma estructurada y segura mediante peticiones HTTP. En una primera fase, se implementaron las rutas básicas para consultar la información esencial del sistema. A medida que avanzaba el desarrollo de la aplicación, y especialmente durante la construcción del panel web, se fueron incorporando de manera progresiva nuevas rutas que respondían a las necesidades específicas de visualización y análisis.

La fase final del desarrollo se centró en la implementación del panel web, diseñado para visualizar métricas clave y facilitar la interpretación de los datos recogidos. Aunque inicialmente se consideró el uso de bibliotecas modernas como React, se optó finalmente por una solución basada en HTML, CSS y JavaScript. Esta elección se justificó por la mayor flexibilidad y control que ofrecen estas tecnologías para la personalización del diseño y la experiencia de usuario. Durante el desarrollo del panel, se adoptó una estrategia iterativa: a medida que se diseñaban nuevas secciones o funcionalidades de la interfaz, se identificaban las necesidades de datos y se ampliaba la API con nuevos endpoints que proporcionaran la información necesaria. De este modo, la integración entre frontend y backend se mantuvo alineada en todo momento.

4.3 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

El desglose de tareas que se ha seguido a lo largo del curso aparece representado en el gráfico siguiente (Figura 15):

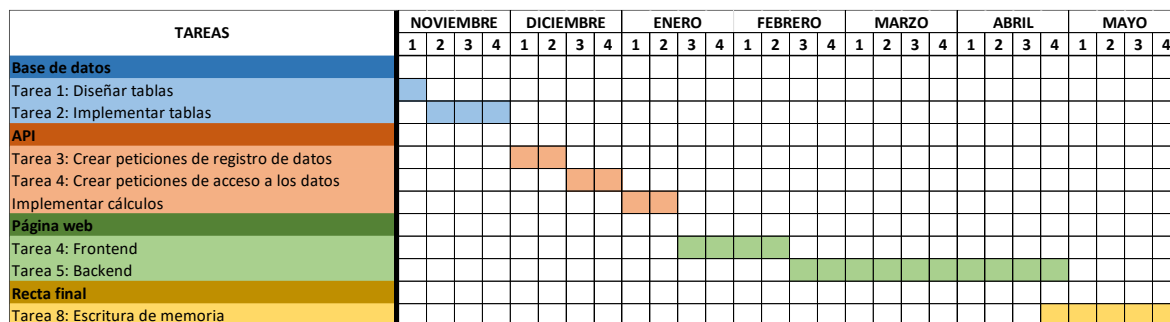


Figura 15 Cronograma de tareas

El proceso de diseño se divide en las siguientes partes: creación de la base de datos, creación de la API y diseño de la página web.

En primer lugar, es necesario realizar una planificación de qué tablas harán falta, y la relación entre los atributos de todas las tablas. Es muy importante que este primer paso se haga bien, ya que el resto de la aplicación se apoya sobre los datos almacenados.

La segunda tarea realizada es la programación de la API REST, que se divide en tres subtareas: el diseño de las peticiones cuya función es registrar datos en la base de datos, la creación de peticiones destinadas a acceder a los datos, que serán usadas por la página web, y los cálculos internos que se deben hacer con los datos, como el ratio de rebote.

Para terminar la herramienta, es necesario crear la página web, en la que estarán disponibles todas las estadísticas recogidas y debidamente procesadas. Para ello, habrá que crear el frontend, usando HTML y CSS, y posteriormente el backend, usando JavaScript. Se dedicarán dos meses a ello, ya que es la parte visual de la aplicación, y debe estar totalmente pulida para garantizar la efectividad del trabajo.

Ya habiendo diseñado la herramienta, se ha realizado una simulación sobre una página web de *captación*.

Por último, se ha dedicado el mes de mayo a escribir lo restante del documento final, ya que se irá documentando a medida que se realicen los pasos indicados en el cronograma.

El proceso seguido se muestra a continuación en la Figura 16:

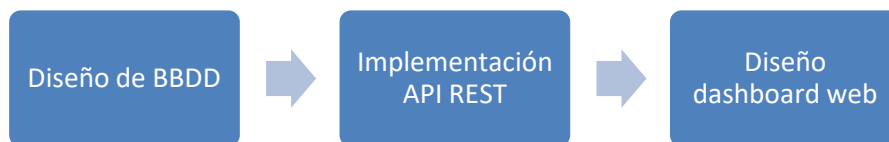


Figura 16 Proceso seguido

Capítulo 5. SISTEMA/MODELO DESARROLLADO

En este capítulo se va a explicar el sistema desarrollado y que consta de tres módulos:

- Una base de datos, que almacena la información que se registra sobre los usuarios, de forma estructurada y relacional. Dicha BBDD se encuentra especificada en los archivos *models.py* y *database.py*.
- Una API REST, que permite grabar la información en la base de datos, y conectarla después con el panel. Ésta se describe en el fichero *app.py*.
- Un panel web, que transforma la información para mostrarla al cliente de manera organizada y lo más sencilla posible. Este panel comprende los archivos *tfh.html*, *tfg.css* y *tfg.js*.

Una vez explicada la estructura general del proyecto, y antes de explicar en detalle cada una de las partes que lo forman, se procede a exponer un caso de uso.

5.0 DISEÑO DE BASE DE DATOS

Se ha decidido implementar una base de datos relacional, y para ello se ha utilizado la librería *SQLAlchemy*, que es un ORM (*Object-Relational Mapper*), es decir, una librería que permite crear y manipular bases de datos sin necesidad de utilizar lenguaje SQL.

Dicha librería permite almacenar la base de datos de forma remota, o en un fichero *.sql*. Se ha escogido la primera opción, ya que aporta simplicidad y flexibilidad, permitiendo guardar copias de seguridad en local. La segunda opción es más apta para situaciones en las que el servidor se encuentra distribuido en múltiples ordenadores, y un cambio en la BBDD en uno de ellos debe afectar también al resto.

SQLAlchemy también presenta una adición llamada *Flask-SQLAlchemy*. Se entrará en detalle sobre *Flask* en la siguiente sección, sobre la API REST, pero esta combinación permite

precisaamente conectar la base de datos con un servidor, lo cual es exactamente lo que se pretende hacer en el proyecto.

El esquema general de la base de datos se muestra en la Figura 17:

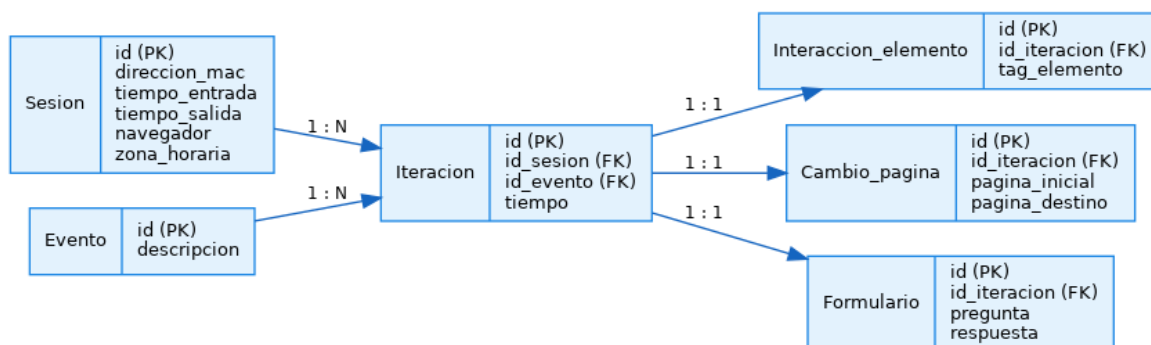


Figura 17 Diagrama de la base de datos

Una vez vista la estructura general, se explicará cada tabla en detalle, junto con su motivación.

Es importante mencionar que, al ser una base de datos relacional, todas las tablas poseen una columna con un identificador numérico único, que separará unívocamente cada registro. No se explicará el identificador propio de la tabla en su sección; no obstante, sí se hará alusión a posibles relaciones de una tabla con otra a través de su identificador.

5.0.1 TABLA DE SESIONES

Una sesión es una visita única por parte de un usuario, y un mismo usuario puede registrar múltiples sesiones. Es el elemento central de nuestro sistema y por ello la tabla de sesión aporta información sobre cada visita que recibe la aplicación o página web. La información que se registra de cada sesión aparece en la Tabla 3:

Tabla 3 Campos de tabla de sesión

<i>Campo</i>	<i>Descripción</i>	<i>Tipo de dato</i>	<i>Ejemplo</i>
Dirección MAC	Dirección física asociada al dispositivo con el que se accede.	String	00:aa:bb:cc:dd:24
Tiempo de entrada a la página	Marca de tiempo en la que el usuario entra a la página en el formato ISO 8601.	Datetime	2025-06-03 18:23:10.833926
Tiempo de salida de la página	Marca de tiempo en la que el usuario sale de la página en el formato ISO 8601.	Datetime	2025-06-03 18:28:10.833926
Navegador	Aplicación de navegación usada para acceder a la página.	String	Chrome, Firefox, Edge
Zona horaria	Zona horaria configurada en el dispositivo con el que se accede a la página.	String	UTC, UTC+1

En primer lugar, la dirección MAC es necesaria para poder identificar a un mismo usuario que entra múltiples veces a la página en momentos distintos. Esto funciona porque la MAC es la dirección física del dispositivo, por lo que no cambia al desplazarse el usuario o cambiar de red wifi, como sí lo haría la dirección IP, por ejemplo.

Cabe mencionar que en el caso de que un mismo usuario entrase con 2 dispositivos diferentes a la página web, no sería posible inferir que se trata de la misma persona, ya que distintos dispositivos tienen distintas direcciones MAC, pero eso se sale del alcance del proyecto.

El tiempo de entrada y tiempo de salida son absolutamente necesarios, ya que aportan información acerca del tiempo que el usuario le dedica a la página, pero, sobre todo, permiten inferir en qué ventana de tiempo tuvo lugar la sesión, lo cual facilita en gran medida la asignación de eventos a sus respectivas sesiones.

Por último, el navegador y la zona horaria son campos que permiten conocer más a fondo al usuario, y discernir información sobre su comportamiento. La zona horaria es esencial para averiguar en qué momento del día se accede a la página web, además de dar una pista esencial sobre la zona geográfica en la que se encuentra el usuario. Esto permite al administrador de la página tomar decisiones de manera informada sobre las futuras actualizaciones por desarrollar.

5.0.2 TABLA DE ITERACIONES

En la base de datos, una iteración se entiende como una acción que realiza el usuario; es decir, esta tabla recoge todas y cada una de las interacciones que el usuario realiza dentro de la página web, ya sea hacer clic en un botón o rellenar un formulario.

Los campos que contiene esta tabla se muestran a continuación, en la Tabla 4:

Tabla 4 Campos de la tabla de iteraciones

<i>Campo</i>	<i>Descripción</i>	<i>Tipo de dato</i>	<i>Ejemplo</i>
ID de sesión	Identificador de la sesión a la que pertenece la iteración.	<i>Integer</i>	1, 2, 3, 4...
ID de evento	Identificador del tipo de evento al que pertenece la iteración.	<i>Integer</i>	1, 2, 3, 4...
Tiempo	Marca de tiempo en la que sucede la iteración, en el formato ISO 8601.	<i>Datetime</i>	2025-06-03 18:28:10.833926

Como se puede deducir, la función de esta tabla es relacionar la sesión que realiza la acción con el evento que sucede.

Tanto el ID de sesión como el ID de evento son claves foráneas que apuntan a los identificadores numéricos de dichas tablas, que han sido comentadas anteriormente.

Además, esta tabla contiene el momento exacto en el que se realiza la acción. Por lo tanto, esta tabla recoge toda la información común a cada uno de los diferentes tipos de eventos, dejando los parámetros específicos de cada evento como el elemento pulsado, o la página a la que se cambia, para sus respectivas tablas.

5.0.3 TABLA DE INTERACCIONES CON ELEMENTOS

Esta es una de las tres tablas que modelan un evento. En este caso, se trata de cualquier interacción que un usuario pueda tener con cualquier elemento presente en la página.

Cabe mencionar que los elementos pertinentes para esta tabla son aquellos que requieren una interacción simple por parte del usuario, como un botón, un enlace o un *checkbox*. Aquellos inputs del usuario que requieran introducir texto, como una caja de comentarios o una pregunta abierta, se registrarán en la tabla de formularios, la cual se explicará más adelante.

Entonces, los campos que contiene esta tabla (Tabla 5) son:

Tabla 5 Campos de la tabla de interacción con elemento

<i>Campo</i>	<i>Descripción</i>	<i>Tipo de dato</i>	<i>Ejemplo</i>
ID de iteración	Identificador de la iteración a la que pertenece la interacción.	<i>Integer</i>	1, 2, 3, 4...
Tag de elemento	Nombre que el administrador de la página asigna a cada elemento.	<i>String</i>	btn_1, info5, skip_page

El identificador de la iteración es necesario para poder relacionar la interacción con la sesión, y de esa forma con el usuario que la realiza. Esto es esencial para posteriormente hacer un análisis de la sesión de un usuario, y poder filtrar los elementos con los que ha interactuado.

En cuanto al tag del elemento, se presentan 2 opciones de implementación.

La primera de ellas consiste en crear una tabla separada de elementos, en la que, de forma similar al proceso de la tabla de eventos, se asigne un identificador numérico único a cada elemento, y se referencie dicho identificador desde esta tabla, mediante una clave foránea. Esto tiene dos desventajas principales:

- Es necesario definir los elementos antes de integrar el código en el JavaScript de la página web, ya que, de lo contrario, el identificador a introducir en esta tabla estaría por definir.
- Se aumenta la complejidad de la base de datos, lo cual puede no ser un problema para la persona que la desarrolla, pero sí para el dueño de un negocio, que ha subcontratado la creación de su página web, y no tiene los conocimientos necesarios para implantar la aplicación de analítica.

La segunda opción, y la que se ha elegido, es simplemente guardar un *tag* para identificar el elemento, es decir, una cadena de caracteres. De esta forma se reduce mucho la complejidad del registro de las interacciones con elementos, pero se deja en manos del usuario la correcta implementación en el JavaScript, y la responsabilidad de mantener el mismo *tag* para el posterior análisis.

5.0.4 TABLA DE CAMBIOS DE PÁGINA

La tabla de cambio de página se comporta de la misma manera que la tabla de interacciones con elementos. Describe el movimiento de los usuarios a través de las distintas subsecciones que componen la página web completa.

Tiene los siguientes campos, descritos en la Tabla 6 :

Tabla 6 Campos de la tabla de cambio de página

<i>Campo</i>	<i>Descripción</i>	<i>Tipo de dato</i>	<i>Ejemplo</i>
ID de iteración	Identificador de la iteración a la que pertenece la interacción.	<i>Integer</i>	1, 2, 3, 4...
Página inicial	Nombre que el administrador asigna a la página en la que se encontraba el usuario.	<i>String</i>	start, checkout, exit
Página de destino	Nombre que el administrador asigna a la página a la que se desplaza el usuario.	<i>String</i>	start, checkout, exit

De nuevo, es necesario conectar el cambio de página con la iteración correspondiente, no solo para tener la capacidad de asociarla al usuario que la realiza, sino también para saber en qué momento exacto se realizó.

La unión de esta tabla con la de iteraciones permite, mediante el uso del *timestamp* de la tabla de iteraciones, reconstruir el flujo completo de cada usuario dentro de la página, lo cual aporta una gran cantidad de información sobre qué rutas son preferidas por los usuarios y cuáles de ellas necesitan una mejora.

De manera análoga a la tabla de interacción con elementos, el administrador tiene la responsabilidad de aportar coherencia en los nombres de las páginas, asegurándose de que cada una de ellas tiene el mismo nombre cada vez que se accede a ella desde cualquier otra.

5.0.5 TABLA DE FORMULARIOS

La última tabla que compone la base de datos es la tabla de formularios. Ésta registra los *inputs* más elaborados del usuario: los que contienen texto. Es necesario separar estos *inputs* de las interacciones con elementos de la tabla mencionada previamente porque el procesamiento de texto es, con diferencia, mucho más complicado que el procesamiento de eventos simples.

Los campos que contiene esta tabla son los siguientes (Tabla 7):

Tabla 7 Campos de la tabla de formularios

<i>Campo</i>	<i>Descripción</i>	<i>Tipo de dato</i>	<i>Ejemplo</i>
ID de iteración	Identificador de la iteración a la que pertenece la interacción.	<i>Integer</i>	1, 2, 3, 4...
Pregunta	Texto de la pregunta que se le hace al usuario.	<i>String</i>	Puntúa el servicio entre el 1 y el 10.
Respuesta	Respuesta que aporta el usuario, ya sea una pregunta abierta o de opción múltiple.	<i>String</i>	Estoy completamente satisfecho.

De nuevo, el identificador de iteración permite unir la respuesta con el usuario que la proporciona. El campo de la pregunta, de forma análoga a los elementos explicados anteriormente, se podría gestionar con una tabla auxiliar, que recoja las preguntas que contiene la página. No obstante, dicho método presenta el mismo problema: la necesidad de insertar con anterioridad las preguntas en la base de datos, por lo que se opta por dejar abierta la especificación al administrador de la página web.

5.0.6 TABLA DE EVENTOS

La tabla de eventos es completamente auxiliar, y está predefinida en la base de datos. Permite asociar un tipo de evento a un identificador numérico, para después hacer referencia a dicho identificador desde las otras tablas. Es importante mencionar que la persona que implemente la aplicación de analítica web sobre la que trata este proyecto, tampoco podrá añadir ni eliminar registros de esta tabla. Esta tabla deberá ser definida por el administrador antes de utilizar el sistema.

Las asignaciones de eventos se muestran en la Tabla 8:

Tabla 8 Asignaciones de eventos

<i>Identificador</i>	<i>Tipo de evento</i>
1	Interacción con un elemento
2	Cambio de página
3	Interacción con un formulario

Se ha escogido manejar los eventos de esta manera con el fin de aportar escalabilidad al sistema, de forma que, si en un futuro se deseara lanzar una actualización sobre la aplicación para permitir el registro de eventos adicionales, únicamente hubiera que incluir el mapeo en esta tabla para que fueran reconocidos por la aplicación.

5.1 SERVIDOR Y COMUNICACIONES

La API REST se ha creado con una funcionalidad lo más directa posible, con el objetivo de obtener los datos realizando *fetch* desde el JavaScript, minimizando al máximo las transformaciones a realizar.

Para desarrollarla, se ha elegido utilizar la librería *Flask*, que es un *framework* web con un estilo de programación minimalista en comparación con sus principales competidores, como *Django*. Además, existe una librería conjunta [22] con *SQLAlchemy* (que ha sido usada para el diseño de la base de datos), llamada *Flask-SQLAlchemy*, la cual combina los funcionamientos de ambas, permitiendo integrar una base de datos SQL dentro de la aplicación *Flask*.

A continuación, se describen las rutas que componen la API, agrupadas según su funcionalidad.

5.1.1 PETICIONES DE SESIÓN

Las peticiones que se muestran en la Tabla 9, acceden a la tabla de sesión, escribiendo, leyendo datos y filtrando para obtener su información básica.

Tabla 9 Peticiones de sesión

<i>Ruta</i>	<i>Descripción</i>
POST /sesiones { direccion_mac: <direccion_mac>, navegador: <navegador>, zona_horaria: <zona_horaria> }	Registra una sesión en la tabla de sesiones. El ID de sesión se genera automáticamente, incrementándose en una unidad consecutivamente con cada sesión. El tiempo de entrada se fija en el momento de la petición, y el tiempo de salida se deja en blanco.
PUT /sesiones/<ID>/finalizar	Añade a una sesión existente <ID> el tiempo de salida, cuando el usuario abandona la pág.
GET /sesiones	Devuelve la información sobre todas las sesiones registradas.
GET /sesiones/usuario { direccion_mac: <direccion_mac> }	Devuelve toda la información disponible sobre todas las sesiones del usuario identificado por <direccion_mac>.
GET /sesiones/direcciones-mac	Devuelve las direcciones MAC únicas almacenadas en la tabla de sesiones.
GET /sesiones/tiempo/direcciones-mac { direccion_mac: <direccion_mac> }	Devuelve todas las tuplas de sesión y tiempo de entrada para el usuario identificado por <direccion_mac>.

Es importante mencionar que, la segunda petición de la Tabla 9, la que tiene como ruta */sesiones/<ID>/finalizar*, tiene una utilidad especial. Cuando un usuario entra a la página web, es necesario registrar su sesión inmediatamente, ya que los eventos y acciones que realicen requieren ser relacionados a la tabla de sesión mediante el identificador, que se crea en el momento de registro de la sesión. No obstante, en la entrada del usuario, no se sabe cuándo va a abandonar la página, entonces se debe dejar ese campo en blanco. Es por esto que se crea la petición en cuestión, que tiene como objetivo completar la información de cada sesión en el momento de salida de la página web, registrando dicho tiempo.

5.1.2 PETICIONES DE ITERACIÓN

Como se ha explicado anteriormente, una iteración es cualquier evento que realiza el usuario, ya sea:

- Interacciones con elementos.
- Cambios de página.
- Relleno de formularios.

Todas estas iteraciones tienen peticiones asociadas, que se muestran a continuación en la Tabla 10:

Tabla 10 Peticiones de iteración

<i>Ruta</i>	<i>Descripción</i>
POST /iteraciones/elementos { id_sesion: <id_sesion>, tag_elemento: <tag_elemento> }	Registra la interacción con el elemento <tag_elemento> por parte del usuario cuya sesión es <id_sesion>.

POST /iteraciones/cambios-pagina { id_sesion: <id_sesion>, pagina_inicial: <pagina_inicial>, pagina_destino: <pagina_destino> }	Registra un cambio de página, de <pagina_inicial> a <pagina_destino>, asociándolo a la sesión identificada por <id_sesion>.
POST /iteraciones/formularios { id_sesion: <id_sesion>, pregunta: <pregunta>, respuesta: <respuesta> }	Registra una respuesta <respuesta> a la pregunta <pregunta>, por parte del usuario cuya sesión es <id_sesion>.
GET /iteraciones	Devuelve la información sobre todas las iteraciones de todos los usuarios.
GET /iteraciones/elementos	Devuelve todas las interacciones con elementos de todos los usuarios.
GET /iteraciones/cambios-pagina	Devuelve todos los cambios de página de todos los usuarios.
GET /iteraciones/formularios	Devuelve todas las interacciones con formularios de todos los usuarios.
GET /iteraciones/sesion {	Devuelve toda la información sobre las iteraciones asociadas a <id_sesion>.

```
id_sesion: <id_sesion>
}
```

Adicionalmente, hay una petición asociada con las iteraciones, que afecta a la tabla de eventos, representada en la Tabla 11:

Tabla 11 Peticiones de evento

<i>Ruta</i>	<i>Descripción</i>
POST /eventos { descripcion: <descripcion> }	Crea una entrada en la tabla de eventos, explicando qué tipo de evento es en <descripcion>.

Cabe mencionar que esta petición debe utilizarse únicamente por el creador de la aplicación, y no por la persona que la implementa en su página web.

5.1.3 PETICIONES DE ESTADÍSTICA

Por último, se tienen las peticiones cuya finalidad es devolver una medida probabilística o parámetro que tiene como objetivo aportar información sobre la página web, de cara a ser usado en el panel.

Se emplean dos tipos de peticiones:

- Las destinadas a medidas simples.
- Las destinadas a generar gráficos temporales o sectoriales.

En primer lugar, en la Tabla 12, se detallan las medidas simples:

Tabla 12 Peticiones de medidas

<i>Ruta</i>	<i>Descripción</i>
GET /estadisticas/visitas-totales	Devuelve la suma total de visitas entre todos los usuarios.
GET /estadisticas/visitantes-unicos	Devuelve una lista con las direcciones MAC únicas registradas.
GET /estadisticas/visitas/promedio-por-usuario	Devuelve la media aritmética de visitas por parte de cada usuario.
GET /estadisticas/tiempo/promedio-por-usuario	Devuelve el tiempo medio que los usuarios pasan en la página web.
GET /estadisticas/tasa-rebote	Calcula y devuelve la tasa de rebote de la página web.
GET /estadisticas/elemento-mas-iteraciones	Devuelve el elemento registrado con mayor número de iteraciones.
GET /estadisticas/paginas/promedio-por-sesion	Devuelve el número medio de subsecciones dentro de la página web que visita un usuario.

GET /estadisticas/iteraciones/promedio- por-usuario	Devuelve el número medio de iteraciones conjuntas por usuario.
---	--

Y, a continuación, se muestran las peticiones destinadas a realizar gráficos en la Tabla 13:

Tabla 13 Peticiones de gráficos

<i>Ruta</i>	<i>Descripción</i>
GET /estadisticas/visitas/ultimos-30-dias { direccion_mac: <direccion_mac> }	Devuelve las visitas diarias de los últimos 30 días. Admite el parámetro <direccion_mac>, que, si está presente, devuelve únicamente el número de visitas de ese usuario. En caso de estar vacío, devuelve las visitas totales por día.
GET /estadisticas/visitas/acumuladas { direccion_mac: <direccion_mac> }	Devuelve las visitas acumuladas de los últimos 30 días. Admite el parámetro <direccion_mac>, que, si está presente, devuelve únicamente el número de visitas de ese usuario. En caso de estar vacío, devuelve las visitas totales acumuladas por día.
GET /estadisticas/tiempo/promedio-por-dia	Devuelve el tiempo promedio que los usuarios pasan en la página web durante cada día.

GET /estadisticas/navegadores/porcentajes	Devuelve los porcentajes de uso de cada navegador con el que los usuarios acceden a la página.
GET estadisticas/zonas-horarias/porcentajes	Devuelve los porcentajes de usuarios que provienen de cada zona horaria desde la que se ha accedido a la página.

5.2 WEB DE VISUALIZACIÓN

Habiéndose explicado en detalle tanto la API REST como la base de datos, falta una tercera pieza crucial, y es que todo el análisis de datos debe ser expuesto de forma comprensible para un usuario común. Esta es la función del panel interactivo, simplificar la información obtenida, y mostrarla de forma que cualquier persona sea capaz de interpretarla con facilidad.

Esto es una tarea difícil, y que requiere una planificación intencional de los diferentes paneles, y para ello se ha elegido la combinación de HTML, CSS y JavaScript, que es la que ofrece un mayor nivel de personalización.

La web de visualización se compone de tres archivos interrelacionados: *tfg.html*, *tfg.css* y *tfg.js*. Dentro de este último, hay múltiples funciones auxiliares como *cargarDI()*, que muestra en pantalla toda la información del panel de tráfico y adquisición (Figura 21), *actualizarGráficoVisitas()*, que vuelve a realizar el *fetch* para incorporar al gráfico nuevos datos, o *formatearFecha()*, que transforma las fechas del formato ISO a otro más sencillo de interpretar por el administrador.

Se han separado las métricas obtenidas en cuatro paneles diferentes:

- Tráfico y adquisición.
- Interacciones y conversión.
- Panel histórico.
- Información de flujo.

Nada más abrir la aplicación, se mostrará directamente el panel de tráfico y adquisición, que será explicado más adelante y se muestra en la Figura 18. Se puede acceder al resto de paneles a través de una barra de opciones (Figura 18, recuadro naranja) localizada en la parte superior de la pantalla. Además, hay un icono de usuario (Figura 18, recuadro azul), que corresponde a un filtro que se abordará en la siguiente sección.

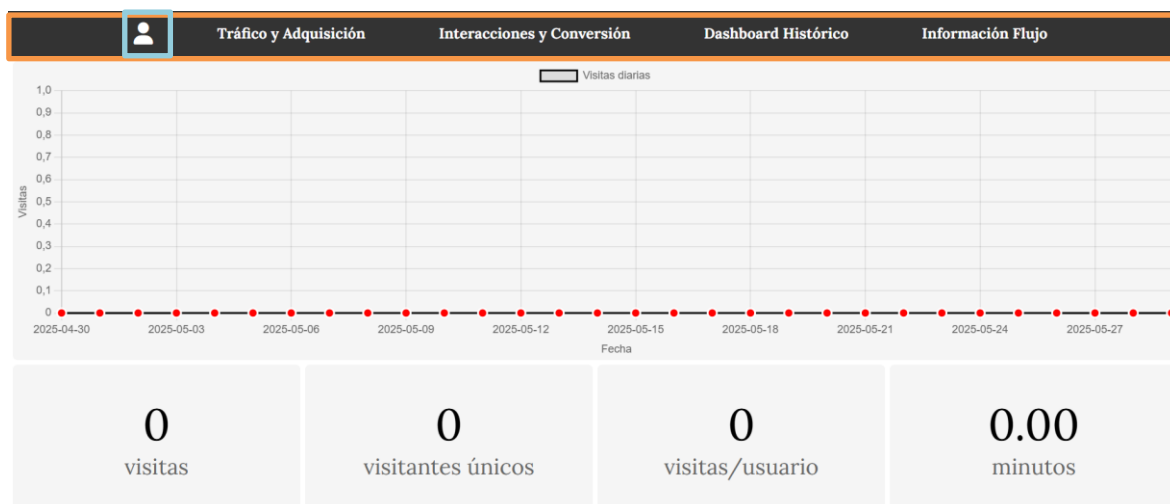


Figura 18 Página inicial

5.2.1 FILTRO DE USUARIOS

Hay cierta información en los paneles que tiene valor al ser filtrada por usuario, es decir, pese a que es interesante observar la evolución de las visitas totales, también hay situaciones en las que se quiere monitorizar la evolución de las visitas de un usuario en concreto. Es por esto, que existe una pestaña de filtrado por usuario, que afecta a:

- El gráfico temporal de visitas. (Figura 21, recuadro verde)
- El gráfico de visitas acumuladas. (Figura 25, recuadro amarillo)
- El gráfico de tiempo medio. (Figura 25, recuadro morado)

No tiene sentido filtrar el resto de las métricas, ya que su finalidad es agrupar el comportamiento de múltiples usuarios, y, por lo tanto, perderían su valor de calcularlas para un único visitante.

Para acceder a dicha pestaña se debe pulsar el botón de arriba a la izquierda con el icono del usuario, como se muestra en la Figura 19:

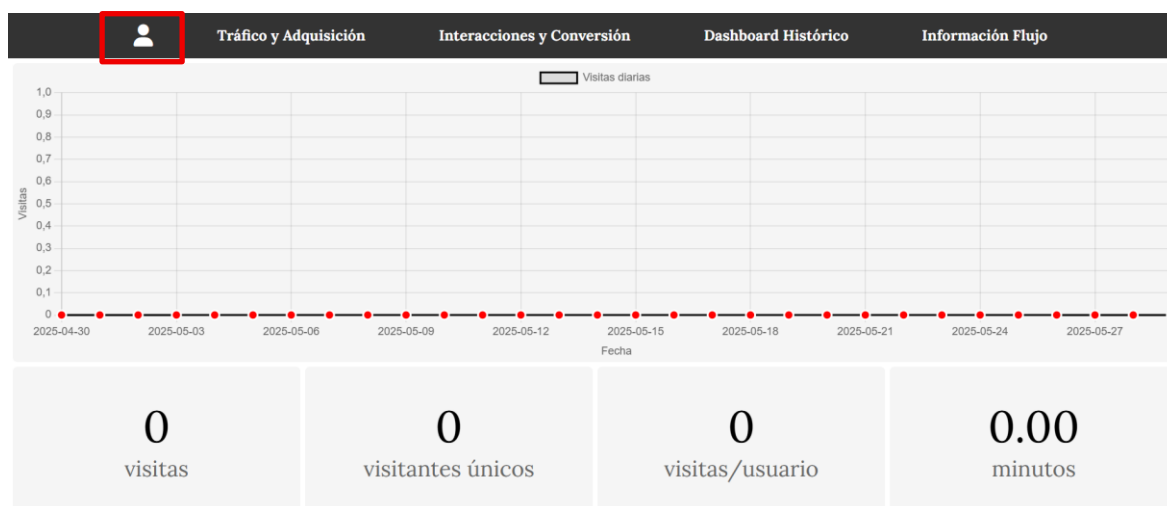


Figura 19 Botón de filtrado de usuario

Al hacer click en el icono (Figura 19, recuadro rojo), aparecerá un desplegable (Figura 20, recuadro verde), en el que se puede seleccionar la dirección MAC del usuario para el que se quiere ver la información. Para esto, se ha utilizado la petición GET `/sesiones/direcciones-mac` (Tabla 9 Peticiones de sesión).

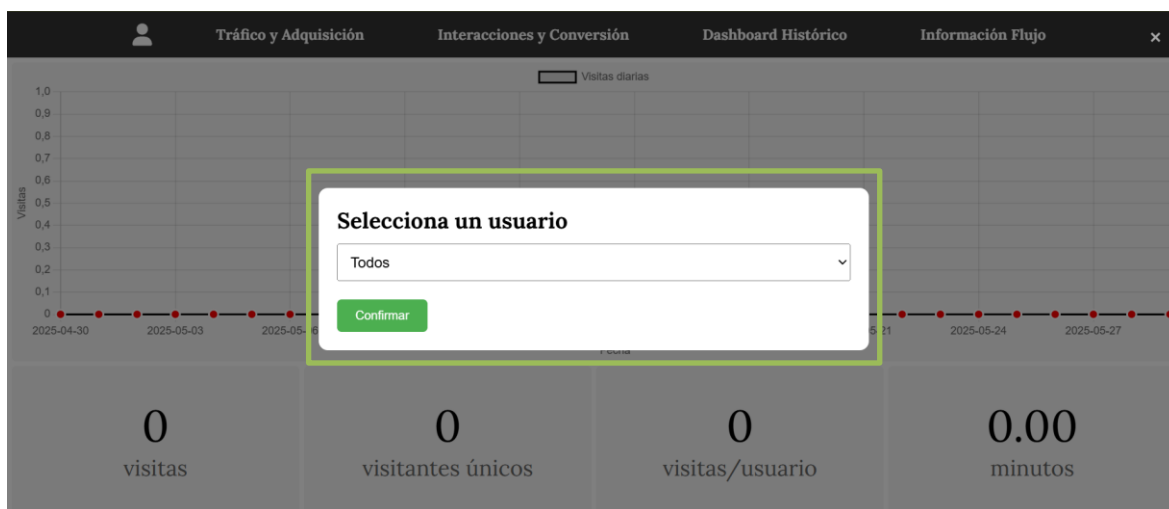


Figura 20 Desplegable de usuarios

5.2.2 PANEL DE TRÁFICO Y ADQUISICIÓN

El primer panel que aparece al abrir la aplicación web es el de tráfico y adquisición, el cual se centra en analizar la página web de forma global, permitiendo visualizar métricas principalmente centradas en las visitas.

El panel se muestra en la Figura 21:

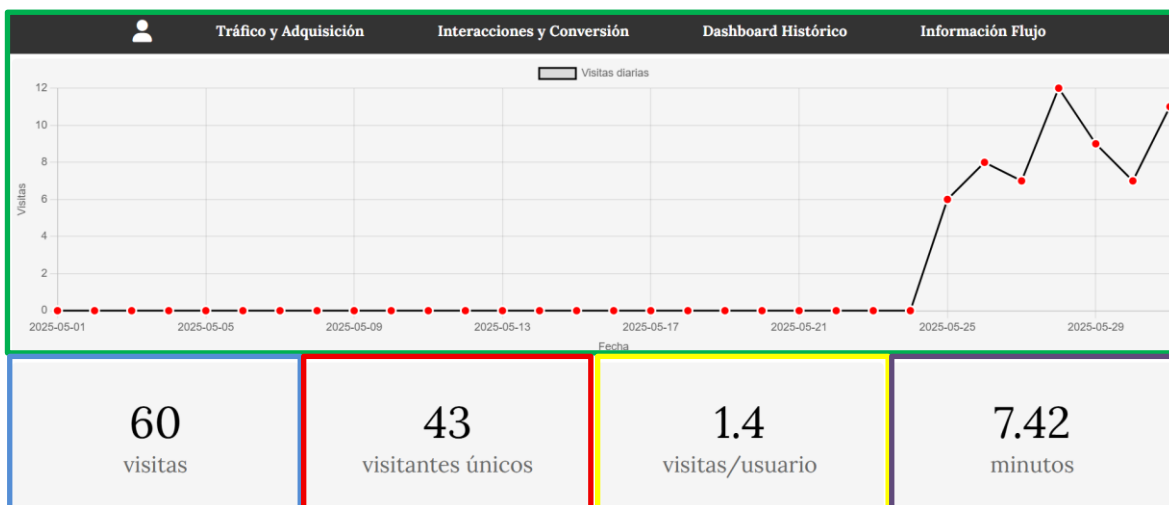


Figura 21 Panel de tráfico y adquisición

Como se puede observar, el principal contenedor de información, que ocupa 2/3 de la pantalla, es un gráfico que muestra las visitas globales cada día, por el último mes (Figura 21, recuadro verde). Dicho gráfico es interactivo, y permite hacer *zoom* para mostrar las visitas de un periodo de tiempo más reciente, como se muestra en la Figura 22. Para generarlo, se hace uso de la petición *GET /estadisticas/visitas/ultimos-30-dias*.

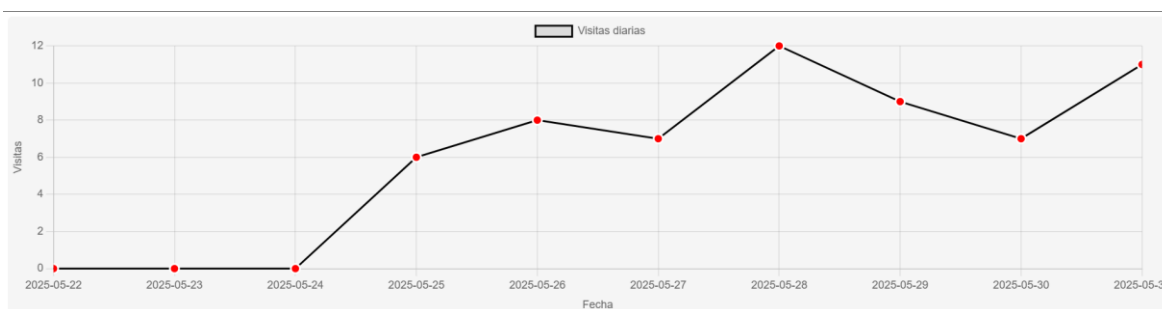


Figura 22 Gráfico de visitas ampliado

Además, en este panel se muestra una serie de métricas que complementan la información del gráfico, las cuales son:

- Número de visitas (Figura 21, recuadro azul) (*GET /estadisticas/visitas/totales*)
- Número de visitantes (Figura 21, recuadro rojo (*GET /estadisticas/visitantes-unicos*)).
- Número medio de visitas por usuario (Figura 21, recuadro amarillo) (*GET /estadisticas/visitas/promedio-por-usuario*).
- Tiempo medio por usuario (Figura 21, recuadro morado) (*GET /estadisticas/tiempo/promedio-por-usuario*).

5.2.3 PANEL DE INTERACCIONES Y CONVERSIÓN

El segundo panel tiene como objetivo aportar información sobre cómo los usuarios se comportan una vez acceden a la página web. Es por esto que las métricas escogidas pretenden discernir qué áreas de la página web requieren actualizaciones o mejoras.

La Figura 23 muestra el aspecto del panel:



Figura 23 Panel de interacciones y conversión

En primer lugar, aparecen 4 métricas, que son:

- Elemento con mayor número de interacciones (Figura 23, recuadro azul) (*GET /estadisticas/elemento-mas-iteraciones*).
- Tasa de rebote (Figura 23, recuadro rojo) (*GET /estadisticas/tasa-rebote*).
- Número medio de páginas visitadas por usuario (Figura 23, recuadro amarillo) (*GET /estadisticas/paginas/promedio-por-sesion*).
- Número medio de interacciones por usuario (Figura 23, recuadro morado) (*GET /estadisticas/iteraciones/promedio-por-usuario*).

Todas estas métricas definen en conjunto cómo los usuarios interactúan con la página web y sus secciones internas. Entre todas ellas, la más importante es la tasa de rebote, que representa el porcentaje de usuarios que abandonan la página web tras haber visitado únicamente la página inicial, es decir, sin haber cambiado de sección. Este porcentaje ayuda a discernir si es necesario modificar la página inicial para lograr captar la atención de los usuarios.

En la parte inferior se puede observar un gráfico de flujo (Figura 23, recuadro verde), más concretamente uno con el estilo de *Sankey*. Éste representa los cambios de página que

realizan los usuarios, calculando los porcentajes para crear una imagen general del movimiento de los visitantes dentro de la página. Para generarlo, se usa la petición *GET /estadisticas/datos-flujo*. Pasando el ratón por encima del gráfico, aparece el porcentaje de usuarios que realiza ese movimiento, como se muestra en la Figura 24, en el recuadro rojo:



Figura 24 Porcentaje de usuarios que van de /cta a /exit

5.2.4 PANEL HISTÓRICO

Este panel no contiene métricas, ya que pretende analizar la distribución de usuarios, además de aportar información general a lo largo del tiempo. Es por esto que se ha decidido dividirlo en 4 gráficos, que son:

- Gráfico de sectores por navegador (Figura 25, recuadro azul) (*GET /estadisticas/navegadores/porcentajes*).
- Gráfico de sectores por zona horaria (Figura 25, recuadro rojo) (*GET /estadisticas/zonas-horarias/porcentajes*).
- Gráfico temporal de visitas acumuladas (Figura 25, recuadro amarillo) (*GET /estadisticas/visitas/acumuladas*).
- Gráfico temporal de tiempo medio en la página (Figura 25, recuadro morado) (*GET /estadisticas/tiempo/promedio-por-dia*).

El aspecto del panel aparece en la Figura 25:

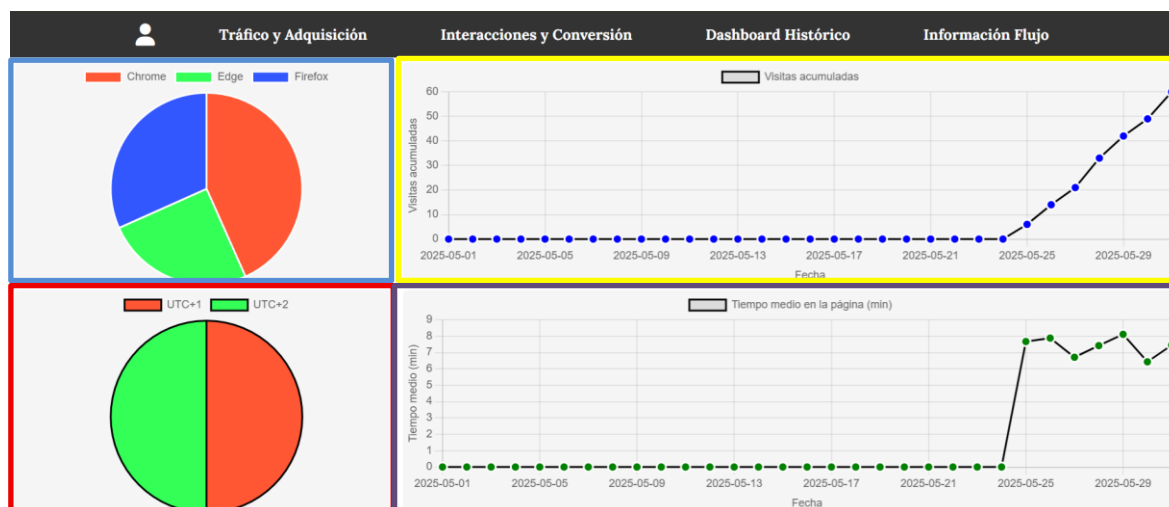


Figura 25 Panel histórico

En primer lugar, el gráfico de sectores de navegador permite discernir qué navegador o navegadores son los más utilizados, para así adaptar la página web al mayor número de usuarios posible. Pasando el ratón por encima de uno de los sectores, aparece el porcentaje de usuarios, como se muestra en la Figura 26, en el recuadro rojo:

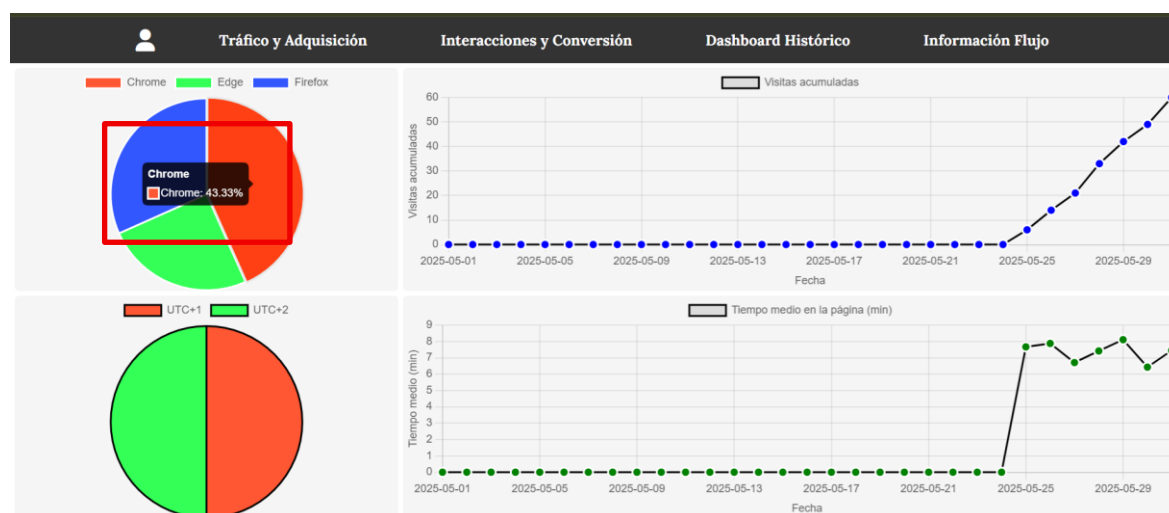


Figura 26 Porcentaje de usuarios que usan Google Chrome

El gráfico de sectores de zonas horarias funciona de la misma forma, pasando el ratón por encima de los sectores para ver el porcentaje. Éste permite al administrador de la página

saber a qué hora entra el mayor número de usuarios en la página, optimizando las campañas de marketing durante las horas punta.

5.2.5 PANEL DE INFORMACIÓN DE FLUJO

Este panel proporciona una vista detallada del comportamiento de cada usuario individual durante su sesión en el sitio o aplicación. A través del seguimiento secuencial de eventos como interacciones, cambios de página o respuestas a formularios, se obtiene una representación precisa de cómo los usuarios navegan, interactúan y completan acciones.

El panel tiene el siguiente aspecto (Figura 27):

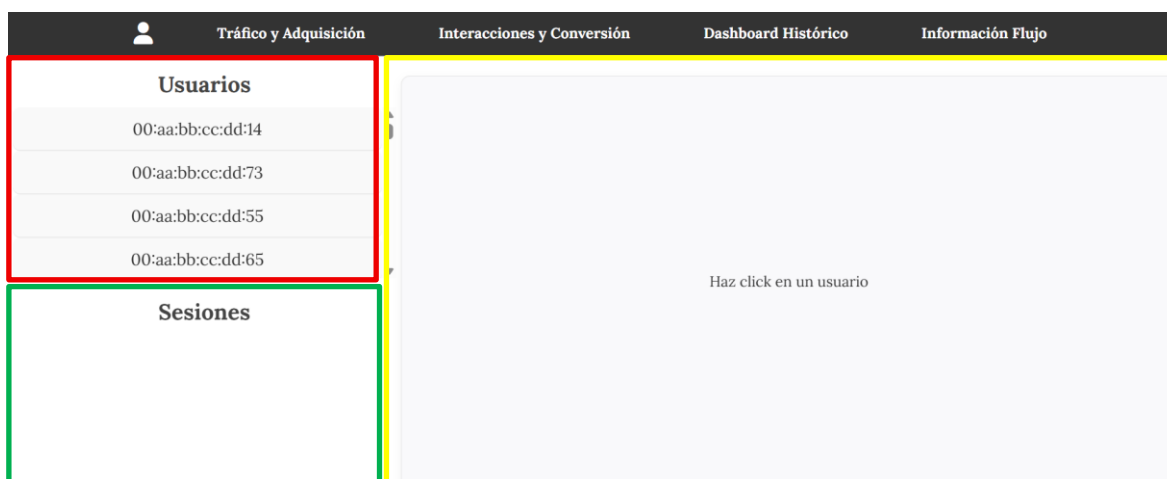


Figura 27 Vista inicial del panel de información de flujo

Se presentan dos listas, una de usuarios (Figura 27, recuadro rojo) (*GET /sesiones/tiempo/direcciones-mac*), que son identificados por su dirección MAC, y otra de sesiones en la parte inferior (Figura 27, recuadro verde) (*GET /sesiones/usuario*), que empieza vacía. A la derecha, se observa el panel de información (Figura 27, recuadro amarillo), en el que se visualiza el flujo de acciones de cada usuario (*GET /iteraciones/sesion*).

Cuando se hace *click* en uno de los usuarios, aparecen todas las sesiones asociadas a su dirección MAC en la lista inferior. Además, en el panel de información se especifica la fecha

y hora en la que se inició cada una de las sesiones, con la finalidad de distinguir unas de otras. Al hacer *click* en una de las sesiones, aparecen las acciones tomadas dentro de la sesión en el panel de información.

Las acciones aparecen especificadas con las siguientes frases:

- Se ha cambiado de la página X a Y.
- Se ha interactuado con Z.
- Se ha respondido a la pregunta A con B.

En la Figura 28 (recuadro verde), se muestra el panel de información con las acciones de la sesión 15 del usuario con dirección MAC "00:aa:bb:cc:dd:12":



Figura 28 Panel de información

Las acciones están ordenadas cronológicamente, permitiendo seguir de cerca el razonamiento del visitante.

5.3 CASO DE USO

Habiéndose explicado en detalle cada uno de los módulos del sistema, se propone ahora un flujo de usuario con el fin de ejemplificar las acciones que realiza la plataforma ante una visita a la página web. Esto se muestra de manera gráfica en la Figura 29:

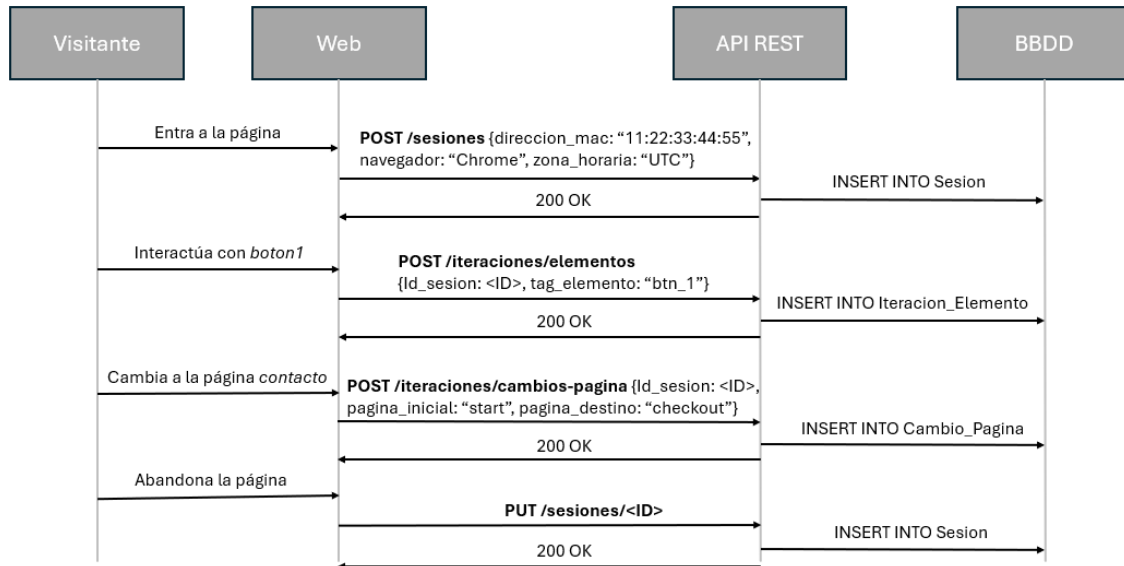


Figura 29 Ejemplo de flujo de usuario

En este caso, el usuario entra en la página web, lo cual provoca una llamada a la API, por medio de HTTP, la cual se comunica con SQL con la BBDD para almacenar el registro. Posteriormente, el visitante interactúa con un elemento y cambia de página; acciones que quedan registradas de la misma manera, con las peticiones HTTP que se observan en la Figura 29. Por último, y como se ha explicado previamente en este capítulo, al abandonar la página, se realiza una última llamada a la API, la cual manipula la base de datos para completar el registro de sesión inicial.

Capítulo 6. ANÁLISIS DE RESULTADOS

Para poder probar el sistema realizado, se ha preparado una simulación de la implementación de la herramienta desarrollada en una página web de captación. Esta simulación tiene como objetivo demostrar la capacidad de la herramienta para capturar y analizar el flujo de usuarios, sus interacciones y decisiones a lo largo del proceso. Para ello, se ha utilizado un archivo de Python, llamado *script_simulacion.py*, el cuál crea un archivo *database.db*, con las siguientes reglas:

- El número total de visitas debe estar en torno a 1000.
- La tendencia de las visitas debe ser creciente.
- Debe haber usuarios que no completen el flujo completo de la página, es decir, que la abandonen previamente.
- Cada visitante debe interactuar con los elementos y formularios que haya en la página en la que se encuentre en cada momento.

La página web seleccionada para esta simulación corresponde a un flujo de captación típico, diseñado para guiar a nuevos usuarios en el primer contacto con un producto o servicio digital. El árbol de páginas y elementos se puede observar en la Figura 30:

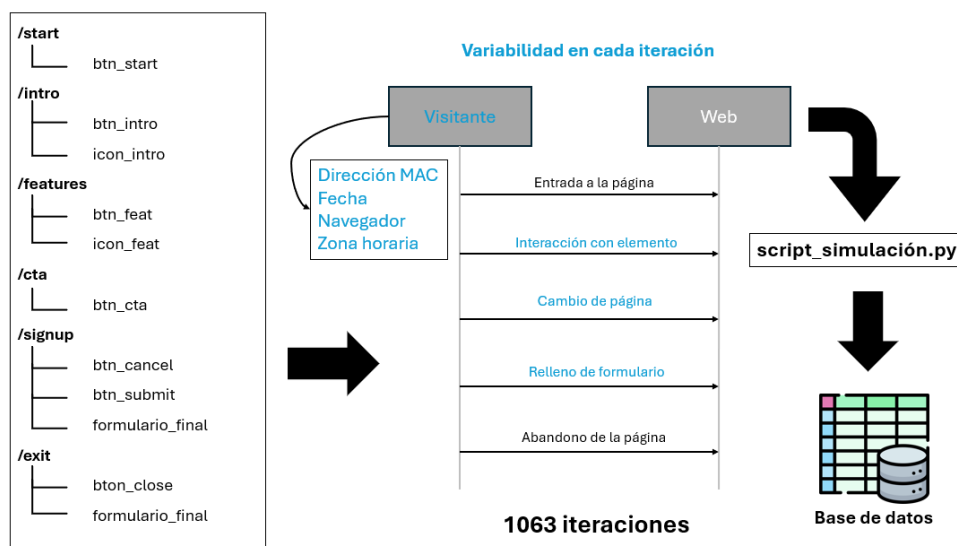


Figura 30 Esquema de simulación [4]

El flujo comienza en la página /start, que actúa como pantalla de bienvenida e introducción. Desde allí, el usuario puede avanzar a /intro, donde se presenta una visión general del producto, y luego a /features, que muestra las funcionalidades principales de forma interactiva. Finalmente, la página /cta invita al usuario a realizar una acción concreta, como registrarse (/signup) o abandonar el proceso (/exit).

Los datos escritos en azul en la Figura 30, son aquellos que varían entre diferentes visitantes dentro de la simulación. La simulación permite comprobar cómo la herramienta captura datos valiosos que facilitan el análisis del comportamiento del usuario en esta página de captación, proporcionando información para mejorar la experiencia de usuario y optimizar el proceso de conversión.

Se asume que se ha implementado el proyecto hace 40 días, y ahora ya hay suficiente información recopilada para extraer valor de ella y tomar las decisiones pertinentes. Entonces, se analiza la información proporcionada por la aplicación.

6.0 PANEL DE TRÁFICO Y ADQUISICIÓN

El panel de tráfico y adquisición contiene la siguiente información, mostrada en la Figura 31:



Figura 31 Panel de tráfico y adquisición en la implementación

En primer lugar, se observa una clara tendencia creciente en el gráfico temporal de visitas, pese a las fluctuaciones. Eso es una buena señal y significa que la página está atrayendo popularidad, y puede significar que la estrategia de promoción está funcionando. Esta es la hipótesis más probable, ya que un aumento de popularidad por otros medios se manifiesta en muchos casos a través de un pico repentino en el número de visitas, lo cual no es el caso para la página de *captación*.

No obstante, al analizar las métricas disponibles, se ve una discrepancia entre el número total de visitas y el de visitantes. Aproximadamente se encuentran 10 veces más visitas que visitantes únicos, lo cual es coherente con las 11.81 visitas por usuario. Esto demuestra que, pese a tener un número muy alto de visitas, realmente no se están captando tantos clientes como parece, lo cual es complicado de discernir sin una aplicación de análisis como la que se ha desarrollado en este proyecto.

Por último, llama la atención los siete minutos y medio que los usuarios permanecen en la página. Esto es una buena señal, y significa que los visitantes están siguiendo el proceso completo, y dedicándole el tiempo suficiente a la página, pero se contradice con el hecho de que éstos vuelven una media de diez veces a visitar la página de nuevo.

6.1 PANEL DE INTERACCIONES Y CONVERSIÓN

Al acceder al segundo panel de la aplicación, se observa lo siguiente (Figura 32):



Figura 32 Panel de interacciones y conversión en la implementación

En primer lugar, se observa que el elemento con mayor número de interacciones es uno llamado "btn_start", con un total de 562. Dado que es un elemento que es necesario pulsar para iniciar el proceso, se podría considerar eliminarlo del análisis, para así obtener el elemento con el que más interactúan los usuarios por voluntad propia.

También se observa que la tasa de rebote alcanza casi el 15%. Esto no es alarmante, ya que simplemente significa que 1 de cada 10 usuarios que entran a la página, la abandonan sin visitar ninguna otra subsección. Este número no es para nada elevado, y entra dentro de los

márgenes habituales [23], por lo que no es necesario rediseñar la página inicial para captar la atención de los visitantes de manera óptima.

El gráfico de flujo muestra que el proceso que los usuarios siguen en la página comprende 5 pasos, que son los siguientes:

1. /start
2. /intro
3. /features
4. /cta
5. /signup o /exit

Y el panel registra que los usuarios visitan una media de 3.79 secciones, lo cual significa que el usuario promedio abandona el proceso en la sección /cta. Esto es una buena señal, ya que se mantiene a los visitantes durante la mayor parte del proceso.

Por último, los visitantes interactúan de media con casi 5 elementos, lo cuál puede ser un área de mejora, dependiendo del número total de elementos interactivos que contenga la página, y de cuál sea su función.

6.2 PANEL HISTÓRICO

El panel histórico, utilizando los datos de la implementación en la página web de captación, muestra la siguiente información, en la Figura 33:

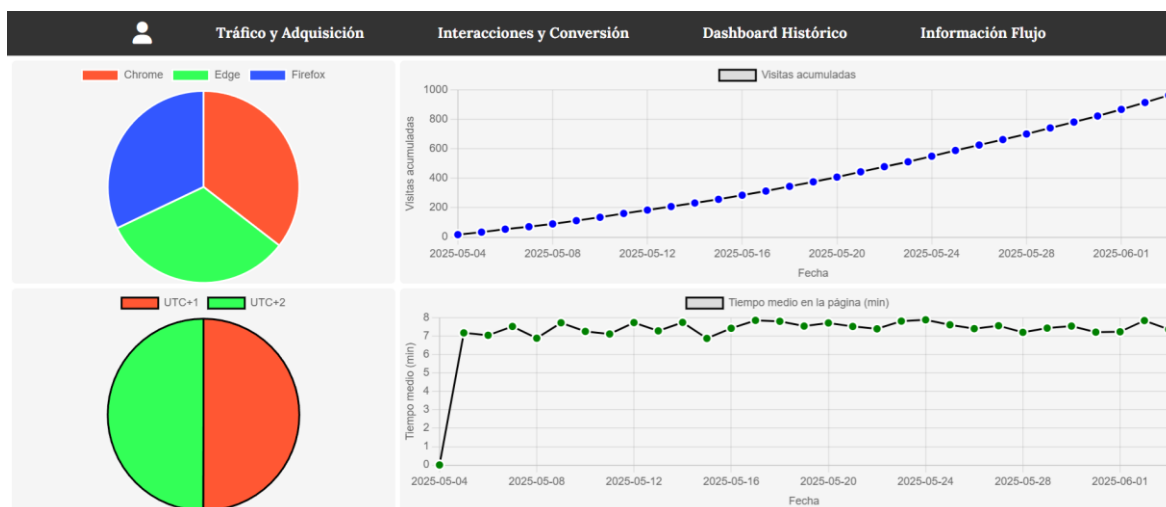


Figura 33 Panel histórico en la implementación

El primer gráfico, de sectores segregando por navegador, muestra el uso de tres navegadores, con estos porcentajes:

- Chrome: 35.47%
- Edge: 32.46%
- Firefox: 32.08%

Se observa que son todos similares, sin encontrar una predominancia de ninguno de ellos. Esto dificulta el proceso de adaptación, ya que, de haber uno con una frecuencia claramente superior al resto, la decisión correcta hubiera pasado por adaptar la página a las fortalezas de dicho navegador, pero en este caso no es posible.

Analizando el segundo gráfico de sectores, por zona horaria, se obtiene una información sumamente relevante. Todos los visitantes de la página están en zonas horarias muy similares, lo cual facilita completamente el priorizar las horas de mantenimiento durante sus horas de sueño, y decidir qué anuncios se muestran en qué horas.

Acerca de las visitas acumuladas, se ve una curva con tendencia creciente, y cuya pendiente también aumenta. Esto es coherente con el gráfico de visitas de la Figura 31, y confirma el aumento en popularidad de la página de *captación*.

Por último, el gráfico temporal del tiempo medio que los usuarios permanecen en la página muestra una tendencia estable, con fluctuaciones relativamente pequeñas, y que se mantiene en torno a los 7 minutos. Esto es aceptable, pero se pueden tomar medidas que contribuyan a mantener a los visitantes dentro de la página durante un mayor tiempo.

6.3 PANEL DE INFORMACIÓN DE FLUJO

En cuanto al panel de información de flujo, su utilidad principal es el análisis de casos concretos, para intentar inferir dónde pueden estar surgiendo los principales problemas que afectan a los usuarios.

Como ejemplo, se observa que, los usuarios que hacen click en el elemento "btn_submit", suelen responder "Sí" a la pregunta final, sobre si el proceso ha sido claro, y los que no lo hacen, suelen responder "No". Se proporciona un ejemplo de cada caso en la Figura 34 y en la Figura 35:



Figura 34 Ejemplo de flujo de usuario 1



Figura 35 Ejemplo de flujo de usuario 2

6.4 DIAGNÓSTICO FINAL

Tras haber analizado las medidas de la aplicación al completo, se procede a extraer las conclusiones y recomendaciones pertinentes.

En la Tabla 14, se muestran las principales áreas de mejora, incluyendo posibles soluciones para implementar:

Tabla 14 Áreas de mejora

<i>Área de mejora</i>	<i>Posible solución</i>
Reducida cantidad de usuarios únicos.	Implementar una campaña de captación dirigida (SEO, redes sociales, colaboración con otras plataformas) para aumentar el alcance y atraer nuevos usuarios a la página.
Repetidas visitas de un usuario sin completar el proceso.	Añadir funcionalidad de guardado de progreso, que permita continuar desde donde se dejó.
Tiempo de permanencia elevado que puede indicar falta de entendimiento.	Simplificar y estructurar el contenido en pasos breves y visuales.
Dependencia del icono informativo para comprender la página.	Incluir directamente la información explicativa en la interfaz, eliminando la necesidad de hacer clic para entender conceptos clave.
Interfaz poco adaptada a los horarios del público objetivo.	Programar contenidos y actualizaciones en los horarios de mayor actividad.

Toda la información que hace posible la detección y resolución de las áreas de mejora es invisible a un análisis básico, y requiere de aplicaciones como la desarrollada en este proyecto para ser detectada y solucionada.

El código para ejecutar el servidor, y abrir el dashboard web con los datos de la página web de captación se ha subido a un repositorio de GitHub, en el siguiente enlace: <https://github.com/miguelsbeato/TFG>.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

7.0 GRADO DE CUMPLIMIENTO DE LOS OBJETIVOS

Este Trabajo Fin de Grado ha abordado el diseño y desarrollo de una plataforma de analítica de datos centrada en la privacidad, con el objetivo de ofrecer una alternativa accesible y eficiente a soluciones comerciales como Google Analytics o Matomo. El resultado ha sido la creación de un sistema funcional y escalable que permite medir, almacenar y visualizar el tráfico de usuarios en aplicaciones web, mediante el uso de una API REST y un panel de control visual.

A continuación, se comenta el grado de cumplimiento sobre cada objetivo planteado en el capítulo 4 en la Tabla 15:

Tabla 15 Grado de cumplimiento de objetivos

<i>Objetivo</i>	<i>Grado de cumplimiento</i>
Desarrollo de un sistema de medición de tráfico	Se ha desarrollado un sistema capaz de registrar y analizar el comportamiento de los usuarios en páginas web y videojuegos.
Creación de un panel accesible	Se ha diseñado e implementado un panel de control web que permite visualizar las estadísticas de tráfico de forma clara, accesible y con distintos niveles de detalle.
Diseño de un script optimizado y ligero	El script cliente desarrollado es eficiente, está libre de redundancias innecesarias y tiene un impacto mínimo en el rendimiento de la página web o aplicación donde se integra.

Creación de una base de datos correctamente estructurada	El sistema se apoya en una base de datos relacional con un modelo bien definido, que garantiza integridad, consistencia y escalabilidad. Las relaciones entre tablas permiten organizar y consultar la información de forma eficiente, lo que facilita tanto el análisis como futuras ampliaciones del sistema.
Implementación en el ámbito del videojuego	Este objetivo no se ha cumplido en su forma original, ya que no se ha realizado la integración directa con un videojuego o aplicación externa. En su lugar, se ha optado por simular la implantación en un entorno web realista, específicamente una página de captación, lo que ha permitido validar la funcionalidad del sistema en condiciones similares.

7.1 PRÓXIMOS PASOS

Pese a que el proyecto ha alcanzado la mayoría de los objetivos propuestos, se identifican varias líneas de mejora que permitirían ampliar su funcionalidad y reafirmar su utilidad práctica. A continuación, se proponen tres desarrollos prioritarios que podrían marcar una evolución natural y estratégica del sistema:

- **En la BBDD, se propone:**
 - **Sistema de autenticación y control de acceso:** se propone implementar un sistema de autenticación que permita gestionar usuarios con distintos roles como administrador, editor o lector. Esto aumentaría la seguridad del sistema y permitiría su uso en contextos colaborativos o en organizaciones donde varias personas necesiten acceso a los datos con distintos niveles de permiso.

- **Sistema de usuarios:** se propone crear una tabla adicional, que permita clasificar la información de distintas páginas web, implementando un sistema de usuarios, y permitiendo que cada uno de ellos sea propietario de distintas páginas.
- **En el panel web, se propone:**
 - **Permitir exportación de flujos a un archivo externo:** se propone modificar la información presente en el último panel (Figura 27), mostrándola en forma de tabla, añadiendo la posibilidad de filtrar la información, y exportando un archivo .csv.

Capítulo 8. BIBLIOGRAFÍA

- [1] “Analítica Web: definición, ventajas y mejores herramientas - NDS.” Accessed: May 26, 2025. [Online]. Available: <https://nuclio.school/blog/analitica-web/>
- [2] “user experience - Explorar - Tendencias de Google.” Accessed: Jun. 01, 2025. [Online]. Available: <https://trends.google.es/trends/explore?date=all&q=user%20experience&hl=es-419>
- [3] “Principales desafíos para Análisis integrado y cómo superarlos.” Accessed: Jun. 01, 2025. [Online]. Available: <https://www.revealbi.io/es/blog/top-challenges-for-embedded-analytics>
- [4] “Iconos vectoriales y stickers - PNG, SVG, EPS, PSD y CSS.” Accessed: Jun. 08, 2025. [Online]. Available: <https://www.flaticon.es/>
- [5] “Iconos Administrador gratis - Flaticon.” Accessed: Jun. 08, 2025. [Online]. Available: <https://www.flaticon.es/iconos-gratis/administrador> title="administrador iconos">Administrador iconos creados por Eucalyp - Flaticon
- [6] “Iconos Hoja-de-calculo gratis - Flaticon.” Accessed: Jun. 08, 2025. [Online]. Available: <https://www.flaticon.es/iconos-gratis/hoja-de-calculo> title="hoja de cálculo iconos">Hoja de cálculo iconos creados por Freepik - Flaticon
- [7] “Iconos Servidor gratis - Flaticon.” Accessed: Jun. 08, 2025. [Online]. Available: <https://www.flaticon.es/iconos-gratis/servidor> title="servidor iconos">Servidor iconos creados por juicy_fish - Flaticon

- [8] “¿Qué es HTTP? | Cloudflare.” Accessed: May 24, 2025. [Online]. Available: <https://www.cloudflare.com/es-es/learning/ddos/glossary/hypertext-transfer-protocol-http/>
- [9] “Cliente, servidor y protocolo HTTP – e-SOLUCIONES-TIC.” Accessed: Jun. 01, 2025. [Online]. Available: <https://www.e-soluciones-tic.com/cliente-servidor-y-protocolo-http/>
- [10] “¿Qué es una API de REST?” Accessed: May 25, 2025. [Online]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- [11] “Qué es una base de datos relacional | Oracle España.” Accessed: May 25, 2025. [Online]. Available: <https://www.oracle.com/es/database/what-is-a-relational-database/>
- [12] “Relationships in SQL – One-to-One, One-to-Many, Many-to-Many | GeeksforGeeks.” Accessed: May 25, 2025. [Online]. Available: <https://www.geeksforgeeks.org/relationships-in-sql-one-to-one-one-to-many-many-to-many/>
- [13] “Entendiendo el lenguaje HTML - El blog de dinahosting.” Accessed: Jun. 03, 2025. [Online]. Available: <https://dinahosting.com/blog/entendiendo-la-base-del-desarrollo-web-html-css-y-javascript/>
- [14] “Inicio rápido – React.” Accessed: Jun. 03, 2025. [Online]. Available: <https://es.react.dev/learn>
- [15] “Google Analytics Logo - símbolo, significado logotipo, historia, PNG.” Accessed: Jun. 01, 2025. [Online]. Available: <https://1000marcas.net/google-analytics-logo/>
- [16] “Google Analytics - Wikipedia, la enciclopedia libre.” Accessed: Nov. 02, 2024. [Online]. Available: https://es.wikipedia.org/wiki/Google_Analytics

- [17] “▷ Google Analytics 4 | Novedades y Configuración de GA4 | UCM.” Accessed: May 26, 2025. [Online]. Available: <https://www.mastermarketingdigitalucm.com/google-analytics-4/>
- [18] “Privacy-first Google Analytics Alternative - App & Web Analytics - Matomo.” Accessed: Jun. 01, 2025. [Online]. Available: <https://matomo.org/>
- [19] “Privacy-first Google Analytics Alternative - App & Web Analytics - Matomo.” Accessed: Jun. 01, 2025. [Online]. Available: <https://matomo.org/>
- [20] “What’s new in Plausible Analytics | Plausible Analytics.” Accessed: Jun. 01, 2025. [Online]. Available: <https://plausible.io/changelog>
- [21] “How to offer a white label web analytics dashboard powered by Plausible | Plausible Analytics.” Accessed: Jun. 01, 2025. [Online]. Available: <https://plausible.io/white-label-web-analytics>
- [22] “Flask-SQLAlchemy — Flask-SQLAlchemy Documentation (3.1.x).” Accessed: Jun. 06, 2025. [Online]. Available: <https://flask-sqlalchemy.readthedocs.io/en/stable/>
- [23] “¿Qué es la Tasa de Rebote y Qué Bounce Rate es buena?” Accessed: May 31, 2025. [Online]. Available: https://es.semrush.com/blog/que-es-la-tasa-de-rebote/?g_network=g&g_keyword=&g_acctid=951-454-0426&g_campaign=ES_SRCH_DSA_Blog_ES&g_keywordid=dsa-2229731902743&g_adtype=search&g_adid=678247170660&g_campaignid=19249322774&g_adgroupid=157746394769&kw=&cmp=ES_SRCH_DSA_Blog_ES&label=dsa_pagefeed&Network=g&Device=c&utm_content=678247170660&kwid=dsa-2229731902743&cmpid=19249322774&agpid=157746394769&BU=Core&extid=109498516666&adpos=&matchtype=&gad_source=1&gad_campaignid=19249322774&gbraid=0AAAAADiv3HSshBehmkxC6bI1R4Fu527Le&gclid=Cj0KCQjw0erBBhDTARIsAKO8iqS0H88XgIJP2CaKAmMpU2WXZi2j5QE7QIYtmqUKoqE0yy8DyfNQVwaAveiEALw_wcB

- [24] “Objetivos y metas de desarrollo sostenible - Desarrollo Sostenible.” Accessed: Jun. 01, 2025. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [25] “Crecimiento económico - Desarrollo Sostenible.” Accessed: Jun. 01, 2025. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/economic-growth/>
- [26] “Reducir las desigualdades entre países y dentro de ellos - Desarrollo Sostenible.” Accessed: Jun. 01, 2025. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/inequality/>

ANEXO I - ALINEACIÓN CON LOS ODS

Este Trabajo Fin de Grado no solo busca ofrecer una solución técnica útil, sino también alinearse con una forma de entender la tecnología como algo que puede tener un impacto positivo más allá de lo puramente funcional. En ese sentido, el proyecto se relaciona con varios de los Objetivos de Desarrollo Sostenible (ODS) establecidos por la Agenda 2030 de la ONU, que marcan una hoja de ruta para avanzar hacia un mundo más justo, equitativo y sostenible [24].

La herramienta desarrollada pretende ser accesible, ética y respetuosa con la privacidad, pero también útil para personas que normalmente no tendrían acceso a este tipo de soluciones. A continuación, explico cómo se relaciona este trabajo con algunos ODS concretos:

8.0 ODS 8 - TRABAJO DECENTE Y CRECIMIENTO ECONÓMICO [25]

Este proyecto ofrece una alternativa gratuita a soluciones comerciales como Google Analytics, lo que ayuda a reducir barreras de entrada para pequeños negocios o profesionales autónomos. Cualquier persona con una web o un proyecto digital puede usar la herramienta para entender mejor a sus usuarios sin pagar por herramientas externas, fomentando así el emprendimiento digital con menos recursos.

8.1 ODS 10 - REDUCCIÓN DE LAS DESIGUALDADES [26]

La herramienta promueve la equidad digital al ofrecer una solución gratuita y accesible, sin barreras técnicas ni económicas, permitiendo que colectivos con menos recursos, como pequeñas ONG, proyectos educativos o emprendedores locales, puedan beneficiarse de la analítica web sin depender de grandes plataformas comerciales.