



UNIVERSIDAD DE MÁLAGA



E.T.S. INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Realizado por

Pablo Rubia Arias

Tutorizado por

Daniel Garrido Márquez

Departamento

Lenguajes y Ciencias de la Computación

MÁLAGA, 06/2025

Contenido

1. Resumen y palabras clave.....	3
1.1 Resumen del TFG.....	3
1.1.1 Español.....	3
1.1.2 English	4
1.2 Palabras clave.....	5
1.2.1 Español.....	5
1.2.2 English	5
2. Introducción.....	7
2.1 Motivación.....	7
2.2 Objetivos	8
2.3 Estructura de la memoria.....	9
3. Estudio del arte / Tecnologías utilizadas.....	13
3.1 Estado del arte.....	13
3.2 Tecnologías	16
3.2.1 Tecnologías principales.....	16
3.2.2 Herramientas de diseño y prototipado	19
3.2.3 Herramientas de desarrollo y colaboración.....	20
3.2.4 Herramientas de modelado y gestión	21
3.2.4 Herramientas de accesibilidad y pruebas.....	22
4. Metodología de trabajo.....	25
4.1 Metodología	25
4.2 Planificación	26
4.3 Herramientas de gestión.....	27
5. Desarrollo del proyecto.....	29
5.1 Análisis.....	29
5.1.1 Fases del desarrollo.....	29
5.1.2 Requisitos.....	33
5.2 Diseño.....	37
5.2.1 Diseño de la base de datos	37
5.2.2 Diseño de la Interfaz	41
5.2.3 Arquitectura del sistema	43

5.3 Implementación.....	44
5.3.1 Páginas Principales.....	44
5.3.2 Problemas encontrados y soluciones	49
5.3.3 Pruebas y resultados.....	51
6. Análisis económico y viabilidad legal	55
6.1 Costes actuales del sistema.....	55
6.2 Costes en caso de escalar o comercializar	56
6.3 Viabilidad legal de la comercialización.....	57
7. Conclusiones y líneas futuras	59
7.1 Conclusiones	59
7.2 Líneas futuras	61
Bibliografía	63
Documentación de tecnologías utilizadas	63
Herramientas de diseño y prototipado	64
Herramientas de desarrollo y colaboración.....	64
Herramientas de modelado y gestión	64
Herramientas de pruebas y accesibilidad	64
Apéndices.....	65
I. Manual de usuario.....	65
II. Manual de instalación/despliegue	73
II.I De forma remota	73
II.II De forma local.....	73
III. Estructura de la Base de Datos	75
IV. Páginas Principales	80
V. Pruebas WAVE [22].....	89

1. Resumen y palabras clave

1.1 Resumen del TFG

1.1.1 Español

Este Trabajo de Fin de Grado consiste en el desarrollo de una aplicación web avanzada para el análisis estadístico de combates competitivos de Pokémon, centrada en el formato VGC. El sistema recopila y procesa datos de replays públicos de Pokémon Showdown [9] [10], permitiendo a los usuarios consultar rankings de uso, tasas de victoria, tendencias históricas y composiciones de equipos. La aplicación ofrece funcionalidades como filtrado de partidas, guardado de replays, análisis detallado de turnos y un asistente de predicción de jugadas. Además, integra un foro para la discusión y el intercambio de estrategias entre la comunidad. El backend está implementado con Node.js [1] y Google BigQuery [5] para gestionar grandes volúmenes de datos de forma eficiente, mientras que el frontend utiliza React [3] y Material UI para proporcionar una experiencia de usuario moderna y responsiva. El proyecto aborda retos como la integración de datos externos, la visualización interactiva y la gestión de usuarios, proporcionando una herramienta útil tanto para jugadores competitivos como para investigadores del metajuego. La plataforma facilita la toma de decisiones estratégicas y el estudio de tendencias en el competitivo de Pokémon, contribuyendo al avance de la comunidad hispanohablante en este ámbito.

1.1.2 English

This Final Degree Project presents the development of an advanced web platform for statistical analysis of competitive Pokémon battles, focusing on the VGC format. The system collects and processes data from public Pokémon Showdown [9] [10] replays, allowing users to consult usage rankings, win rates, historical trends, and team compositions. The application offers features such as match filtering, replay saving, detailed turn-by-turn analysis, and a move prediction assistant. Additionally, it integrates a forum for community discussion and strategy sharing. The backend is implemented with Node.js [1] and Google BigQuery [5] to efficiently manage large volumes of data, while the frontend uses React [3] and Material UI to provide a modern, responsive user experience. The project addresses challenges such as external data integration, interactive visualization, and user management, providing a valuable tool for both competitive players and metagame researchers. The platform facilitates strategic decision-making and the study of trends in competitive Pokémon, contributing to the advancement of the Spanish-speaking community in this field.

1.2 Palabras clave

1.2.1 Español

- Text Mining
- Estadística
- Frontend
- BigQuery [5]
- RESTful APIs

1.2.2 English

- Text Mining
- Statistics
- Frontend
- BigQuery [5]
- RESTful APIs

2. Introducción

2.1 Motivación

El formato VGC (Video Game Championships) constituye la vertiente oficial del juego competitivo en Pokémon, regulada por The Pokémon Company International. A diferencia del formato "singles" popularizado en comunidades como Smogon [12], el VGC se juega en formato dobles (2v2) y en modalidad Bo3 (mejor de tres) en torneos presenciales y online. Aunque esta disciplina ha crecido significativamente en los últimos años, especialmente con el auge de las retransmisiones en Twitch y YouTube, la disponibilidad de herramientas analíticas especializadas sigue siendo limitada.

Actualmente, la mayoría de las aplicaciones disponibles para el análisis competitivo se centran en mostrar información básica como rankings de uso o estadísticas generales extraídas de Smogon [12] —una base de datos abierta y comunitaria— acompañadas de elementos visuales como gráficos y sprites. Estas soluciones, si bien útiles a nivel informativo, no profundizan en el análisis dinámico de una partida ni ofrecen soporte personalizado al jugador competitivo que busca mejorar su toma de decisiones, su construcción de equipo o su comprensión de los enfrentamientos clave.

A partir de esta carencia surge la motivación principal del proyecto: diseñar y desarrollar una aplicación web capaz de analizar en profundidad partidas reales del formato VGC, extrayendo información directamente desde replays públicas de Pokémon Showdown [9] [10]. El objetivo es ir más allá de las estadísticas convencionales, ofreciendo al usuario un conjunto de herramientas interactivas, visualizaciones detalladas y análisis avanzados que le permitan no sólo observar tendencias, sino también comprender el porqué de ciertos resultados y, en consecuencia, mejorar de forma continua en el entorno competitivo.

2.2 Objetivos

El objetivo principal de este Trabajo de Fin de Grado es crear una aplicación web basada en la nube, en la que el usuario representa un escenario concreto proporcionando varios datos de entrada, como los Pokémon, movimientos, vida, habilidades y cualquier otro factor relevante. La aplicación indica cuál es el mejor movimiento posible basado en estadísticas reales. Todo esto incluye aspectos como la gestión de usuarios, partidas, turnos y otros elementos relacionados.

Las estadísticas se obtienen de partidas públicas accesibles desde Pokémon Showdown, una plataforma dedicada a jugar a cualquier modalidad de Pokémon sin necesidad de videojuegos ni consolas. Es la principal vía de juego tanto para profesionales como para jugadores casuales o principiantes, por lo que constituye la mejor fuente posible de datos. Este proceso se realiza periódicamente para añadir nuevas partidas a la base de datos, incrementando así el número de escenarios y mejorando la precisión de las estadísticas.

El programa tiene en cuenta estas partidas, especialmente las del jugador que gana, y devuelve, de todas esas partidas ganadas, el porcentaje de uso de cada movimiento que ha contribuido a la victoria. Además, muestra un porcentaje de probabilidad de victoria en cada escenario. Un usuario puede seleccionar una partida que ha jugado y analizar en qué turnos ha fallado, pudiendo modificar un turno concreto y simular cuál habría sido el resultado.

La aplicación dispone de un sistema de autenticación que permite a los usuarios ver sus propias estadísticas y revisar sus partidas. Incluye un análisis general de todas las partidas, mostrando, por ejemplo, qué Pokémon o movimiento es el más usado, qué porcentaje de victoria proporciona un determinado Pokémon, etc. Muchas de estas características se visualizan mediante gráficas para facilitar la interpretación. Para ello, se recopilan las batallas según el nombre de usuario, siempre que las repeticiones sean públicas en Pokémon Showdown.

De forma más concreta, la aplicación incluye las siguientes funcionalidades:

- **Mi Perfil:** muestra datos del usuario.
- **Partidas Guardadas:** partidas públicas guardadas por el usuario.
- **Partidas Públicas:** partidas jugadas y públicas.
- **Turn-Assistant:** asistente de jugadas para escenarios concretos introducidos por el usuario, que devuelve consejos basados en estadísticas.
- **Rankings:** rankings de uso y victoria de cada Pokémon.
- **Estadísticas:** estadísticas de jugadores, partidas y equipos.
- **Foro:** espacio donde los jugadores pueden debatir sobre diversos temas.

2.3 Estructura de la memoria

La memoria del presente Trabajo de Fin de Grado se organiza en los siguientes capítulos:

- **Introducción:** motivación personal y objetivos perseguidos.
- **Estudio del arte y tecnologías utilizadas:** Se analizan las principales herramientas y plataformas existentes en el ámbito del análisis competitivo de Pokémon, así como las tecnologías seleccionadas para el desarrollo de la aplicación, justificando su elección.
- **Metodología de trabajo:** Se describe la metodología empleada para la gestión y desarrollo del proyecto, incluyendo la planificación, las herramientas de control de versiones y la organización de las distintas fases.

- **Desarrollo del proyecto:**
 - **Análisis:** Se identifican y describen los requisitos funcionales y no funcionales del sistema, el público objetivo, las limitaciones del contexto actual del competitivo Pokémon y la motivación detrás de la propuesta. Asimismo, se realiza un estudio comparativo de soluciones existentes, detectando carencias que se pretende resolver con este proyecto.
 - **Diseño:** Se presenta la arquitectura general de la aplicación, el diseño de la base de datos y de las APIs, así como los componentes principales del frontend. Se incluyen diagramas, esquemas y prototipos de interfaz, junto con la justificación de las decisiones tomadas tanto a nivel visual como estructural.
 - **Implementación:** Se explica cómo se ha llevado a cabo la construcción de la aplicación, detallando el desarrollo del backend y frontend, la integración con BigQuery [5] y Google Cloud, así como las funcionalidades clave. Se profundiza también en los principales retos técnicos encontrados durante esta fase y las soluciones adoptadas.
- **Pruebas y resultados:** Se exponen los procedimientos de validación y pruebas realizados sobre la plataforma, así como los resultados obtenidos, incluyendo capturas de pantalla y ejemplos de uso.
- **Conclusiones y líneas futuras:** Se resumen los logros alcanzados, las limitaciones detectadas y se proponen posibles mejoras o ampliaciones para el futuro.
- **Bibliografía:** Recoge todas las fuentes consultadas que han servido de referencia para el desarrollo del proyecto.

- **Apéndices:** Incluye información complementaria relevante, como manuales de usuario, listados de código significativos, diagramas de arquitectura y cualquier otro material que amplíe la información del cuerpo principal de la memoria.

3. Estudio del arte / Tecnologías utilizadas

3.1 Estado del arte

En el ámbito del análisis competitivo de Pokémon, las herramientas existentes suelen centrarse en ofrecer estadísticas básicas de uso y popularidad de Pokémon, así como calculadoras de daño. Estas herramientas se apoyan en datos públicos extraídos del simulador **Pokémon Showdown** [9] [10] y se limitan a presentar rankings generales o información individual de cada Pokémon sin entrar en el análisis contextual de partidas completas o el rendimiento estratégico en situaciones reales.

Una de las plataformas más reconocidas es [Pikalytics](#), que recopila estadísticas agregadas del uso de Pokémon en los torneos oficiales y en Showdown [9] [10]. La herramienta es visualmente atractiva y útil como referencia rápida, pero no proporciona ningún tipo de análisis contextual sobre cómo se comportan los Pokémon en batalla, ni permite examinar decisiones o estrategias.

Otra fuente relevante es [Smogon](#) [12], que, aunque está más orientada al juego competitivo no oficial (basado en reglas comunitarias), ofrece abundante información sobre roles, movimientos recomendados y análisis de sets. Sin embargo, su enfoque es más textual y teórico, y tampoco ofrece herramientas interactivas ni análisis basados en datos reales de combate.

Por su parte, [Munchstats](#) también utiliza datos de Showdown [9] [10] y presenta estadísticas desglosadas por formatos, pero nuevamente se trata de una herramienta pasiva: muestra qué Pokémon y combinaciones se ven con más frecuencia, sin ayudar al usuario a entender por qué funcionan ni cómo sacar conclusiones aplicables a su juego.

Otras páginas destacables incluyen:

- [Victory Road](#): publica estadísticas de los torneos oficiales VGC, como los Pokémon más usados, objetos y movimientos. Sin embargo, se limita a eventos concretos y no ofrece herramientas interactivas.
- VGC Stats: (ya cerrado) presenta datos y resultados de torneos oficiales, rankings de jugadores y metagame por torneo, pero no permite explorar turnos, combinaciones ni rendimiento contextualizado.
- Pokémon Global Link (ya cerrado, pero fue relevante durante generaciones anteriores): ofrecía estadísticas generales del entorno competitivo online, pero sin llegar a un nivel de análisis profundo de combates.

Hasta donde alcanza mi conocimiento, no existen herramientas específicas que ofrezcan un análisis profundo y contextualizado de partidas de Pokémon VGC (Video Game Championships), centrado en lo que ocurre en cada turno y el impacto de las decisiones tomadas por los jugadores. Por ello, este proyecto pretende cubrir ese vacío, aportando funcionalidades únicas como:

- **Asistente de turnos**, que permite navegar por las jugadas de un combate y ver qué ocurrió en cada turno y qué decisiones se tomaron.
- **Estadísticas en base a victorias**, analizando qué combinaciones de Pokémon o estrategias tienen más éxito.
- **Estadísticas de equipos y pares**, identificando sinergias reales entre Pokémon.
- **Visualización de acciones detalladas**, con gráficos y filtros avanzados.

Además de herramientas de Pokémon, también es interesante observar el uso de análisis de partidas en otros videojuegos o disciplinas competitivas. Por ejemplo, en ajedrez, plataformas como [Lichess](#) o [Chess.com](#) ofrecen análisis automáticos post-partida, que detectan errores, sugerencias de jugadas mejores, estadísticas de aperturas y rendimiento por fases del juego. Este tipo de enfoque, centrado en el análisis profundo de la toma de decisiones, es el que ha servido como inspiración conceptual para este trabajo.

Del mismo modo, en eSports como *League of Legends* o *Valorant*, existen herramientas como [OP.gg](#) o [Tracker.gg](#) que muestran estadísticas personalizadas, mapas de calor, análisis de rendimiento por rol, etc. Estos ejemplos reflejan la importancia creciente de los datos en la mejora del rendimiento competitivo y justifican la pertinencia de desarrollar algo similar para el entorno del VGC.

Asimismo, en juegos móviles competitivos como *Clash Royale*, existen plataformas como [RoyaleAPI](#) que permiten un análisis detallado de partidas, mazos y rendimiento individual. Estas herramientas ofrecen estadísticas como tasas de victoria, historial de partidas, comparativas entre cartas y sinergias entre mazos, además de funciones avanzadas como predicción de cartas rivales o seguimiento de torneos en tiempo real. Este tipo de análisis automatizado y accesible desde el entorno web refuerza la idea de que el uso inteligente de los datos puede ser un aliado clave en la mejora del rendimiento, incluso en juegos donde las decisiones se toman en cuestión de segundos.

3.2 Tecnologías

3.2.1 Tecnologías principales

3.2.1.1 Node.js

Descripción: Node.js [1] es un entorno de ejecución para JavaScript en el servidor.

Uso en el proyecto: Se ha utilizado como plataforma principal para desarrollar el backend de la aplicación y construir la API REST que gestiona las peticiones entre el cliente y la base de datos. Su enfoque asíncrono permitió un manejo eficiente de múltiples solicitudes, especialmente en operaciones de análisis de datos y obtención de replays.



3.2.1.2 Firebase Authentication

Descripción: Firebase [2] es una plataforma de desarrollo de aplicaciones de Google. Su módulo de autenticación permite implementar login/register de manera segura.

Uso en el proyecto: Se ha empleado para implementar la autenticación de usuarios, gestionando sesiones seguras mediante correo electrónico y contraseña.



3.2.1.3 React

Descripción: React [3] es una biblioteca de JavaScript para la creación de interfaces de usuario interactivas y reactivas.

Uso en el proyecto: Ha sido la base para construir el frontend. Se emplearon componentes funcionales con hooks para crear vistas dinámicas, como la simulación de partidas, visualización de estadísticas, login y rankings.



3.2.1.4 Material UI (MUI)

Descripción: MUI [4] es una librería de componentes de interfaz basada en las directrices de diseño Material Design de Google.

Uso en el proyecto: Se utilizó para el diseño de toda la interfaz gráfica, facilitando una estética moderna, consistente y adaptable a distintos dispositivos. También ayudó a mantener un diseño accesible y funcional sin necesidad de crear estilos desde cero.



3.2.1.5 Google BigQuery

Descripción: BigQuery [5] es un sistema de análisis de datos masivos de Google Cloud, altamente escalable y rápido.

Uso en el proyecto: Se usó para almacenar y consultar de manera eficiente grandes volúmenes de datos sobre partidas Pokémon. Permite el análisis estadístico global, como tasas de victoria, uso de Pokémon, predicción de jugadas y evolución de estrategias.



3.2.1.6 Google Cloud Scheduler

Descripción: Servicio de Google Cloud para programar tareas automáticas en momentos específicos.

Uso en el proyecto: Se configuraron tareas diarias para que se ejecutaran scripts de actualización automática de los datos, como la descarga de nuevas replays desde Showdown [9] [10], de forma regular y sin intervención manual.



3.2.1.7 Google Cloud Run

Descripción: Plataforma de Google para desplegar y escalar aplicaciones basadas en contenedores Docker.

Uso en el proyecto: Se desplegó el proyecto como un contenedor en Cloud Run [7], lo que permite escalar automáticamente según la carga de peticiones, con alta disponibilidad y seguridad gestionada por Google.



3.2.2 Herramientas de diseño y prototipado

3.2.2.1 Figma

Descripción: Plataforma de diseño de interfaces y prototipado colaborativo en tiempo real.

Uso en el proyecto: Se utilizó para diseñar los primeros prototipos de la interfaz gráfica de usuario, permitiendo iterar y validar el diseño antes de implementarlo en React [3]. Se definieron también los flujos de interacción del usuario.



3.2.2.2 Colors

Descripción: Herramienta para la creación de paletas de colores armónicas.

Uso en el proyecto: Se utilizó para definir la identidad visual del proyecto, eligiendo una paleta de colores coherente y accesible que se mantuvo durante toda la implementación del frontend.



3.2.3 Herramientas de desarrollo y colaboración

3.2.3.1 GitHub

Descripción: Plataforma de control de versiones y colaboración en desarrollo de software.

Uso en el proyecto: Se empleó para gestionar el repositorio del proyecto, hacer seguimiento de cambios en el código, gestionar ramas para nuevas funcionalidades y coordinar posibles integraciones futuras.



3.2.3.2 Visual Studio Code (VS Code [18])

Descripción: Entorno de desarrollo ligero y extensible.

Uso en el proyecto: Fue el editor de código principal para desarrollar tanto el backend (Node.js [1]) como el frontend (React [3]). Se complementó con extensiones para linting, control de Git, autocompletado, y herramientas de Firebase [2].



3.2.4 Herramientas de modelado y gestión

3.2.4.1 MagicDraw [19]

Descripción: Herramienta de modelado UML profesional.

Uso en el proyecto: Se utilizó para representar la arquitectura del sistema mediante diagramas de clases, casos de uso, y secuencia, facilitando el diseño estructurado antes de iniciar el desarrollo.



3.2.4.2 Trello [20]

Descripción: Aplicación web para gestión de tareas basada en el método Kanban.

Uso en el proyecto: Permitió planificar y organizar el desarrollo del proyecto por fases, marcando tareas como “Por hacer”, “En progreso” y “Completado”. Facilitó la priorización y control del avance.



3.2.4 Herramientas de accesibilidad y pruebas

3.2.4.1 WAVE [22]

Descripción: Herramienta online para evaluar la accesibilidad web.

Uso en el proyecto: Se utilizó para verificar que la interfaz web cumpliera con los estándares de accesibilidad (WCAG [24]), garantizando una experiencia inclusiva para todos los usuarios.



3.2.4.2 TAW [23]

Descripción: Otra herramienta para análisis de accesibilidad, centrada en el cumplimiento de la normativa española e internacional.

Uso en el proyecto: Se empleó como complemento a WAVE para detectar y corregir posibles errores en contraste, etiquetas de formularios, navegación por teclado y elementos alternativos, aún que su uso fue menor que el de WAVE [22].



3.2.4.3 Postman [21]

Descripción: Plataforma para pruebas de APIs REST.

Uso en el proyecto: Fue esencial durante el desarrollo del backend. Se utilizó para probar rutas de la API, validar peticiones/respuestas y simular distintos escenarios de uso para detectar errores.



POSTMAN

4. Metodología de trabajo

4.1 Metodología

El desarrollo del trabajo se lleva a cabo de forma individual por el alumno, combinando una planificación estructurada con un enfoque iterativo y flexible. Se sigue una metodología basada en principios de desarrollo ágil, adaptados al contexto y necesidades del proyecto. Esto permite dividir el trabajo en fases manejables, priorizar tareas clave y ajustar los objetivos y funcionalidades según los avances y los retos encontrados durante el desarrollo.

El proceso se estructura en ciclos cortos de diseño, implementación y revisión, lo que facilita la incorporación de mejoras continuas y la rápida adaptación a posibles cambios en los requisitos o en la tecnología utilizada. Se emplean herramientas de control de versiones (Git) y gestión de tareas para organizar el trabajo y documentar el progreso.

La metodología empleada ha permitido abordar de manera eficiente las distintas fases del proyecto, desde el análisis inicial y la definición de requisitos, hasta el desarrollo, las pruebas y la puesta en marcha de la plataforma.

4.2 Planificación

Toda la información detallada del desarrollo del proyecto puede consultarse en el repositorio de Trello [20]: <https://trello.com/b/Keke6xD9/pokemon-statistics>

Cronograma de trabajo:

- **30 de septiembre:** Surge la idea inicial del proyecto. Comienzo de la investigación sobre tecnologías relevantes como **MongoDB**, **Postman** [21] y **Node.js** [1].
- **1 – 4 de octubre:** Definición del esquema de información para las replays y creación del primer script para automatizar la recolección de datos.
- **1 – 8 de octubre:** Primer contacto con el tutor académico y propuesta del tema para el Trabajo de Fin de Grado.
- **17 – 22 de octubre:** Migración de la base de datos de **MongoDB** a **BigQuery** [5], alineando el enfoque del proyecto con la línea de TFG disponible.
- **21 de octubre:** Elaboración del primer **diagrama IFML** [15] [16].
- **24 – 31 de octubre:** Desarrollo del esquema básico del proyecto.
- **25 de octubre:** Creación de la primera versión del **diagrama de despliegue**.
- **1 de noviembre:** Segunda iteración del **diagrama IFML** [15] [16], incorporando cambios.
- **18 de noviembre – 6 de diciembre:** Desarrollo de prototipos visuales de la interfaz de usuario mediante **Figma** [13].
- **19 de noviembre:** Selección de la paleta de colores utilizando **Coolers** [14].
- **20 de noviembre:** Comenzando el archivo de recolección de datos.
- **26 de enero:** Redacción del **anteproyecto** del TFG.

- **27 de enero – 27 de febrero:** Mejora continua del archivo de recolección de datos.
- **11 – 24 de febrero:** Proceso de automatización avanzada de la recolección de datos.
- **3 de marzo – 4 de mayo:** Desarrollo intensivo de la aplicación, incluyendo **API**, **IU** (interfaz de usuario) y lógica de análisis.
- **2 de mayo:** Revisión y pruebas con **Postman** [21] para asegurar el correcto funcionamiento de la API.
- **5 – 8 de mayo:** Proceso de **despliegue** de la aplicación.
- **8 – 12 de mayo:** Realización de **tests de accesibilidad** para asegurar la usabilidad del sistema.
- **12 de mayo – 17 junio:** Corrección de bugs y desarrollo de la memoria.

4.3 Herramientas de gestión

Como herramienta de gestión me he limitado a usar Trello [20], ya que, al ser un trabajo realizado de forma individual, no veía necesario el uso de otra más.

5. Desarrollo del proyecto

5.1 Análisis

5.1.1 Fases del desarrollo

El desarrollo del proyecto se ha dividido en varias fases, agrupadas por ámbitos tecnológicos. Cada una de ellas ha abordado tareas específicas que han contribuido de forma progresiva a la implementación y despliegue del sistema completo. A continuación, se describen las principales fases:

5.1.1.1 Backend y Automatización de Datos

En esta fase se desarrolló el núcleo lógico y funcional del proyecto, encargado de recopilar, analizar y almacenar los datos de las partidas.

- **Node.js [1] y Express.js [8]:** Se utilizó Node.js [1] como base del backend, junto con Express [8] para estructurar la API REST encargada de ofrecer acceso a los datos procesados.
- **Script de recolección de datos:** Se desarrolló un script que accede automáticamente a los logs públicos de partidas de Pokémon Showdown [9] [10], procesando su contenido para extraer Pokémon activos, movimientos, habilidades, objetos, efectos de campo, cambios de estado y condiciones del entorno.
- **Cambio de base de datos:** Inicialmente se utilizó MongoDB, pero posteriormente se migró a **Google BigQuery [5]** para aprovechar sus capacidades de análisis a gran escala.

- **Automatización con Google Cloud Scheduler [6]:** Se implementaron tareas programadas para ejecutar el script de importación periódicamente, manteniendo actualizados los datos almacenados. En Cloud Scheduler [6] hay un job llamado `dailyFetchReplays` que llama diariamente a las 12 de la mañana al servicio de Cloud Run [7] llamado `fetchreplaysdaily` que es el que se encarga de obtener las replays de Pokémon Showdown [9] [10], analizarlas y almacenarlas en la base de datos. Ahí se encuentran tres archivos diferentes con distintas funciones: `package.json`, que define las dependencias necesarias para la imagen; `obtainGameDataDeploy.js`, que recibe un replay y la analiza línea a línea para obtener sus datos; y `fetchReplaysDeploy.js`, que recolecta todas las replays de la página y llama a `obtainGameDataDeploy.js` para que las analice y, posteriormente, las almacena en la base de datos. Tanto `fetchReplaysDeploy.js` como `obtainGameDataDeploy.js` se encuentran en el código del proyecto, así como sus versiones iniciales (`fetchReplays.js`, `obtainGameData.js`), que fueron las que se desarrollaron y probaron antes de automatizarlas con Google Cloud Scheduler [6].

5.1.1.2 Frontend e Interfaz de Usuario

Se diseñó e implementó una interfaz amigable que permite a los usuarios explorar y analizar los datos de forma visual e intuitiva.

- **React + MUI (Material UI):** Se desarrolló una aplicación SPA (Single Page Application) con React [3]. La librería MUI [4] permitió crear una interfaz moderna, responsiva y consistente, basada en principios de diseño Material.
- **Diseño visual:** El diseño se estructuró en **Figma** [13], donde se definieron prototipos navegables. Para la identidad visual del proyecto, se empleó **Coolers** [14] para establecer una paleta de colores accesible y coherente.
- **Conexión con la API:** La interfaz se comunicó con el backend mediante llamadas HTTP usando fetch y/o Axios, permitiendo consultar datos filtrados por jugador, Pokémon o resultados.
- **Accesibilidad:** Se realizaron pruebas con herramientas como **WAVE** [22] y **TAW** [23] para garantizar el cumplimiento de los estándares de accesibilidad y mejorar la experiencia para todos los usuarios.

5.1.1.3 Autenticación y Seguridad

Una parte importante del desarrollo consistió en gestionar los accesos y roles de los usuarios.

- **Firestore Authentication:** Se integró el sistema de autenticación de Firestore [2] para permitir que los usuarios puedan iniciar sesión de forma segura. Esto permite gestionar accesos personalizados según los permisos establecidos.

5.1.1.4 Integración y Conectividad

Durante esta fase se integraron todos los componentes desarrollados y se realizaron pruebas para asegurar el correcto funcionamiento del sistema completo.

- **Integración de frontend y backend:** Se conectaron las peticiones del cliente con los endpoints del backend, gestionando estados de carga, errores y visualización dinámica de los datos.
- **Pruebas de la API con Postman [21]:** Se validaron todas las rutas expuestas por la API REST utilizando Postman [21], verificando tanto la estructura de los datos como las respuestas en diferentes escenarios.
- **Control de versiones con GitHub [17]:** Se empleó GitHub [17] para gestionar el repositorio del proyecto, facilitando el trabajo iterativo, la colaboración y el seguimiento del progreso.

5.1.1.5 Despliegue y Producción

Esta fase tuvo como objetivo publicar el sistema en un entorno accesible para su uso real y evaluación.

- **Despliegue:** Se realizó el despliegue de la aplicación y del sistema de recolección mediante **Google Cloud Run [7]**, utilizando contenedores Docker. Además, se configuró un sistema de integración continua para que, cada vez que se sube un cambio a la rama develop, la versión actualizada de la aplicación se despliegue automáticamente.

5.1.1.6 Documentación y Modelado

A lo largo del desarrollo se mantuvo una documentación detallada y se generaron diversos modelos para representar el sistema.

- **Modelado UML con MagicDraw [19]:** Se crearon diagramas de casos de uso y de despliegue mediante la herramienta MagicDraw [19].
- **Gestión del trabajo con Trello [20]:** Se organizó el desarrollo en tareas utilizando tableros Kanban, lo que facilitó el seguimiento del avance y la priorización de actividades.

5.1.2 Requisitos

Durante la fase inicial del análisis, se definieron una serie de requisitos que guiarían el desarrollo de la aplicación. El objetivo principal del proyecto era construir una **aplicación web interactiva** centrada en el análisis de partidas competitivas del videojuego Pokémon, proporcionando herramientas de apoyo estratégico a los usuarios mediante el uso de datos agregados y simulaciones. La aplicación debía incluir capacidades tanto de análisis como de interacción social, priorizando la escalabilidad y el rendimiento.

5.1.2.1 Requisitos funcionales

Los requisitos funcionales hacen referencia a las características específicas que la aplicación debía ofrecer a los usuarios finales. Entre ellos se destacan:

- **Visualización de partidas:** los usuarios deben poder acceder a la información detallada de replays, permitiendo explorar cada turno y acción tomada durante la partida.
- **Análisis turno a turno:** la aplicación debe mostrar la evolución de cada batalla, incluyendo los Pokémon activos, movimientos realizados, efectos de campo, cambios de estadísticas, y estados alterados.
- **Simulación de probabilidades:** la aplicación debe ser capaz de generar simulaciones o cálculos de **porcentajes de victoria esperados** en función del estado actual de la partida y de los Pokémon restantes/revelados.
- **Cálculo de estadísticas agregadas:** se requiere obtener métricas globales como porcentaje de uso de cada Pokémon, tasa de victorias, frecuencia de movimientos y objetos más utilizados, entre otras.
- **Autenticación de usuarios:** los usuarios deben poder registrarse e iniciar sesión de manera segura mediante un sistema de autenticación, para acceder a funcionalidades personalizadas.
- **Interacción entre usuarios (foro/chat):** la aplicación debe permitir a los usuarios participar en un foro o sistema de comentarios para debatir estrategias, compartir ideas o realizar preguntas sobre jugadas y replays.
- **Rankings y comparativas:** se deben ofrecer rankings de Pokémon más usados, equipos con mejor desempeño y otras clasificaciones útiles para la toma de decisiones estratégicas.

5.1.2.2 Requisitos no funcionales

Los requisitos no funcionales definen las cualidades que debe cumplir el sistema en cuanto a rendimiento, seguridad, escalabilidad y experiencia de usuario. Entre los más relevantes se encuentran:

- **Rendimiento en consultas masivas:** dado el gran volumen de partidas y datos analizados, es fundamental que la base de datos responda rápidamente a consultas complejas. Se prevé el uso de sistemas como BigQuery [5] o bases de datos optimizadas para análisis en lote.
- **Escalabilidad:** el sistema debe estar preparado para soportar un crecimiento considerable en el número de usuarios y partidas almacenadas. Se prevé el uso de servicios como **Cloud Run** [7] o plataformas serverless que permitan escalar automáticamente.
- **Seguridad:** el acceso a funcionalidades sensibles (como publicar, comentar, visualizar análisis avanzados) debe estar protegido mediante autenticación y autorización adecuadas. Se contempla el uso de **Firestore Authentication** u otro sistema de autenticación robusto.
- **Accesibilidad:** la interfaz debe ser intuitiva y accesible desde distintos dispositivos y navegadores, permitiendo su uso en entornos con diferentes capacidades técnicas.
- **Portabilidad y despliegue en la nube:** el sistema debe estar diseñado para ser desplegado fácilmente en entornos cloud, permitiendo actualizaciones continuas y disponibilidad global.

5.2 Diseño

5.2.1 Diseño de la base de datos

5.2.1.1 *Firestore*

Firestore se ha utilizado principalmente para almacenar los datos relacionados con los usuarios: autenticación, proveedores de acceso y fechas de uso. Esta base de datos está conectada con **Firestore [2] Authentication**, lo que simplifica considerablemente la gestión de identidad, ya que permite registrar, autenticar y mantener sesiones de manera segura sin necesidad de implementar lógica personalizada del lado del servidor.

Gracias a su integración directa con los servicios de autenticación de Google, se puede gestionar fácilmente el inicio de sesión con distintos proveedores, como correo electrónico o Google, además de obtener metadatos útiles sobre la cuenta.

Este sistema permite controlar fácilmente el acceso de los usuarios a los servicios de la plataforma, asociar sus interacciones con replays y publicaciones del foro, y mantener una experiencia personalizada sin comprometer la seguridad.

Campo	Descripción
Identificador	Correo electrónico
Proveedores	Email/Google
Fecha de creación	Creación de la cuenta
Fecha de acceso	Último acceso
UID de usuario	Único por usuario

5.2.1.2 Google BigQuery

BigQuery [5] actúa como la base de datos analítica del proyecto. Su diseño está enfocado a manejar grandes volúmenes de datos estructurados y realizar consultas complejas de forma rápida y eficiente, características ideales para el análisis de partidas de Pokémon, que puede implicar miles de turnos, eventos y registros por replay.

A continuación, se describen las principales tablas utilizadas en el proyecto.

5.2.1.2.1 forum_topics

Esta tabla contiene los temas creados en el foro. Cada fila representa un hilo o conversación iniciada por un usuario. Se almacena información como el título del tema, su categoría, el autor que lo creó, la fecha de creación y un contador con el número de respuestas, tal y como se muestra en la Figura 19.

Esta estructura permite obtener listados de los temas más populares, recientes o activos, facilitando la organización y navegación del foro.

5.2.1.2.2 forum_messages

Relacionada directamente con `forum_topics`, esta tabla guarda los mensajes individuales de cada hilo. Cada mensaje está vinculado a un tema mediante un ID de referencia y contiene el contenido del mensaje, el autor, y la fecha de publicación, tal y como se muestra en la Figura 20.

Este modelo jerárquico permite construir una vista de conversación tradicional de un foro, donde los mensajes se agrupan en torno a un tema principal, manteniendo además la trazabilidad del usuario que los envió

5.2.1.2.3 saved_replays

Esta tabla almacena los replays que los usuarios han guardado manualmente desde la aplicación. Tiene una estructura centrada en el usuario: cada fila representa una lista de replays guardadas por un usuario concreto, incluyendo su identificador de usuario, las URL originales de las replays, tal y como se muestra en la Figura 21.

De esta manera, los usuarios pueden construir su propio historial de partidas destacadas o interesantes para analizar posteriormente.

5.2.1.2.4 replays

Es la tabla principal del sistema de análisis. Contiene los datos estructurados que se obtienen automáticamente desde la API de Pokémon Showdown [9] [10], procesados y almacenados con una estructura compleja que refleja la evolución de cada partida turno a turno.

Incluye información como:

- Jugadores y equipos utilizados.
- Resultado del combate.
- Pokémon revelados, desmayados y activos en cada momento.
- Acciones realizadas en cada turno (cambios, ataques, activaciones de objetos o habilidades).
- Cambios de estado y estadísticas.
- Condiciones del campo (clima, Trick Room, Tailwind, etc.).

La complejidad de esta tabla radica en su profundidad jerárquica: cada replay contiene un array de turnos, y cada turno a su vez contiene múltiples eventos. Esta estructura permite realizar análisis detallados sobre cómo se desarrollan los combates, identificar patrones de victoria o fracaso, y calcular estadísticas avanzadas como:

- Porcentaje de uso de combinaciones de Pokémon.
- Eficacia de estrategias por apertura o por pareja.
- Análisis de acciones ganadoras o de errores frecuentes.

Todos los detalles se pueden ver en las figuras: Figura 22, Figura 23, Figura 24, Figura 25, Figura 26, Figura 27, Figura 28 y Figura 29.

5.2.2 Diseño de la Interfaz

5.2.2.1 Figma y Coolors

Durante las fases iniciales del proyecto, se consideró fundamental establecer una identidad visual clara y una experiencia de usuario intuitiva. Para ello, se utilizaron herramientas como **Figma** [13], que permitió crear prototipos de las pantallas principales de la aplicación, y **Coolors** [14], que facilitó la elección de una paleta de colores coherente y accesible.

El diseño se centró en la simplicidad y claridad, permitiendo que tanto usuarios experimentados como nuevos pudieran acceder rápidamente a la funcionalidad deseada. Se diseñaron las siguientes vistas principales: perfil, editar perfil, partidas guardadas, partidas públicas, rankings, asistente de turnos y foro. Para ver el diseño, basta con acceder al enlace [PokemonStatistics](#).

Todos los elementos de la interfaz se diseñaron siguiendo principios de diseño responsivo, para asegurar su correcto funcionamiento tanto en dispositivos móviles como en pantallas de escritorio.

Para ver mejor el diseño, se puede cargar el modelo que se encuentra en la carpeta de diagramas del proyecto desde <https://editor.ifmledit.org>.

5.2.3 Arquitectura del sistema

La arquitectura de la aplicación se basa en una separación clara de responsabilidades entre frontend, backend y base de datos. Esta estructura permite una mayor escalabilidad y facilidad de mantenimiento.

- **Frontend:** desarrollado con React [3], gestionando el enrutamiento, estado de la aplicación y renderizado dinámico. La comunicación con el backend se realiza mediante peticiones HTTP a través de fetch o axios.
- **Backend:** construido con Node.js [1] y Express [8], proporciona una API RESTful para gestionar usuarios, replays, estadísticas, rankings, etc. Las rutas están protegidas mediante autenticación con Firebase [2].
- **Base de datos:** se emplea Google BigQuery [5], una base de datos analítica altamente escalable y optimizada para el procesamiento de grandes volúmenes de datos. Su estructura basada en tablas permite almacenar y consultar información compleja como turnos, replays o estadísticas agregadas de forma eficiente.
- **Servicios adicionales:**
 - **Firebase Authentication** [2] para la gestión de sesiones y seguridad.
 - **Cloud Run** [7] (**Google Cloud**) para el despliegue del backend, permitiendo escalar automáticamente sin necesidad de infraestructura dedicada.

- **GitHub** [17] para control de versiones y trabajo colaborativo.

Esta arquitectura modular, resumida en la Figura 3, permite ampliar fácilmente el sistema en el futuro, por ejemplo, integrando nuevos formatos de combate, mejorando el análisis con modelos estadísticos o añadiendo nuevas funcionalidades sociales.

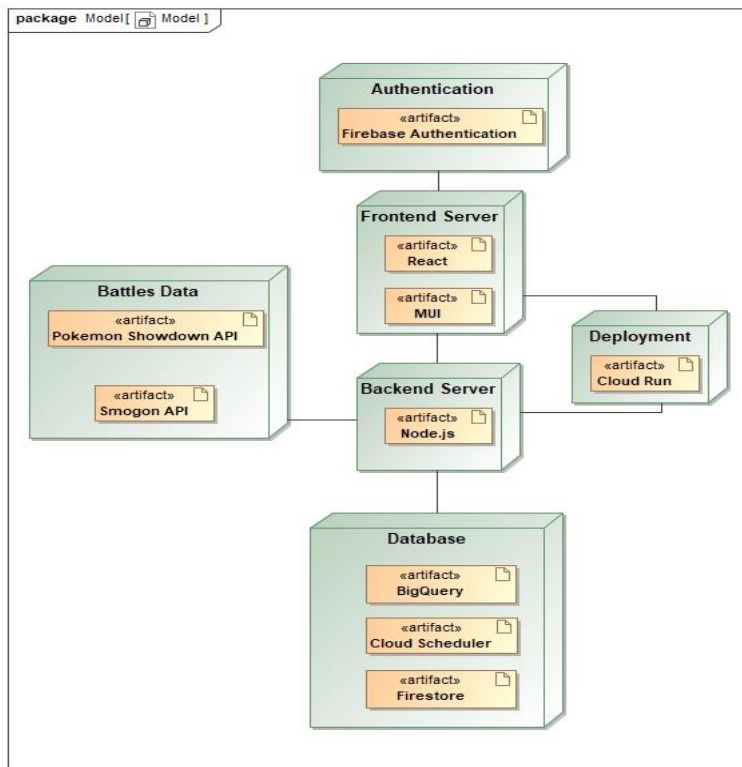


Figura 3. Diagrama de despliegue.

5.3 Implementación

5.3.1 Páginas Principales

Durante esta fase se desarrollaron las funcionalidades principales de la plataforma, centrándose tanto en la experiencia del usuario como en el procesamiento y análisis de los datos. A continuación, se describen en detalle las características implementadas, explicando su lógica, utilidad y la interacción entre ellas. En la Figura 1 se ilustran los casos de uso contemplados para los usuarios del sistema.

5.3.1.1 Home

Pantalla de inicio de la aplicación. Incluye accesos directos a otras secciones y una breve introducción sobre las funcionalidades disponibles, orientando al usuario en su primer contacto con la plataforma (Figura 30).

5.3.1.2 Sign In

Permite a los usuarios acceder a su cuenta mediante credenciales (Figura 31). Se implementaron características como:

- Botón para mostrar/ocultar la contraseña.
- Inicio de sesión a través de correo electrónico o cuenta de Google.
- Enlace directo a la página de registro (Sign Up).

5.3.1.3 Sign Up

Permite el registro de nuevos usuarios (Figura 32). Esta sección incluye:

- Botón para mostrar/ocultar la contraseña.
- Barra que mide la seguridad de la contraseña (mínimo 8 caracteres).
- Registro con correo electrónico o cuenta de Google.
- Enlace directo a la página de inicio de sesión (Sign In).

5.3.1.4 Profile

Muestra la información del perfil de usuario autenticado. Si el usuario no ha iniciado sesión, se redirige automáticamente a la página de inicio de sesión (Figura 33).

Desde esta sección también es posible:

- Cerrar sesión.
- Eliminar la cuenta (con confirmación, mediante la introducción de la contraseña o escribiendo la palabra 'delete' en caso de cuenta de Google).

5.3.1.5 Public Games

Muestra todas las replays públicas almacenadas en la base de datos, organizadas en páginas (Figura 34). Ofrece filtros y opciones de ordenación por nombre de jugador, fecha, resultado, entre otros. Si el usuario ha iniciado sesión, puede guardar partidas en su colección personal. Al hacer clic en una partida, se redirige a su replay original en Pokémon Showdown [9] [10].

5.3.1.6 Saved Games

Muestra las partidas (Figura 37) que el usuario ha guardado previamente desde la sección Public Games (Figura 34). Si no hay partidas guardadas (Figura 35) o si no se ha iniciado sesión (Figura 36), se redirige a dicha sección.

Incluye:

- Filtros y ordenación.
- Eliminación de partidas individuales o de todas en bloque.
- Análisis detallado de una partida con el botón "**Analyze Battle**".
- Posibilidad de seleccionar partidas para generar estadísticas agregadas, activando el botón "**View Analytics**" al seleccionar al menos dos partidas compatibles.

5.3.1.7 Analyze Battle

Permite analizar una partida turno por turno. Se muestran los movimientos realizados, los porcentajes estimados de victoria de cada jugador en cada momento y recomendaciones estratégicas (Figura 38). Esta funcionalidad ha presentado dificultades, ya que las búsquedas requieren una gran especificidad. Con los datos actuales, los resultados son escasos, pero ampliar el rango de coincidencias reduciría la calidad de la recomendación.

5.3.1.8 Battle Analytics

Si el usuario ha iniciado sesión y ha seleccionado al menos dos replays en las que coincidan el nombre y el equipo del jugador, se generan estadísticas detalladas a partir de dichas partidas (Figura 39). Se presentan datos sobre:

- Uso de Pokémon y movimientos.
- Resultados por combate.
- Gráficas agregadas generadas dinámicamente.

5.3.1.9 Rankings

Muestra distintos rankings basados en datos tanto oficiales como propios:

- **Usage Rankings:** basado en estadísticas oficiales de Smogon [12]. Al seleccionar un Pokémon, se muestran datos como uso histórico, habilidades, movimientos, objetos, spreads, tipos Tera y compañeros frecuentes (Figura 40).
- **Winrate Rankings:** basado en datos recopilados por la propia aplicación, ya que no existen fuentes oficiales. La presentación es similar al ranking de uso, e incluye filtros adicionales para refinar los resultados (Figura 41).
- **Teams Rankings y Leads Rankings:** rankings no oficiales que muestran combinaciones de equipos completos (Figura 42) y parejas iniciales de combate (leads) (Figura 43), calculados a partir de las partidas analizadas por el sistema.

5.3.1.10 Turn Assistant

Funcionalidad central del proyecto. Permite simular un turno proporcionando información detallada sobre el estado del combate (Pokémon en campo, HP, condiciones del clima, objetos, etc.) (Figura 44).

Al pulsar "**Analyze**", el sistema devuelve:

- Porcentajes de victoria asociados a cada acción disponible.
- Recomendación sobre la mejor jugada según los datos introducidos.

Esto permite al usuario entrenar y mejorar la toma de decisiones en situaciones de alta complejidad.

5.3.1.11 Forum

Espacio donde los usuarios pueden participar en discusiones, compartir estrategias, resolver dudas o proponer ideas. Se organiza por temas o hilos, y permite respuestas en cadena (Figura 45).

5.3.1.12 Contact/Help

Sección de contacto y ayuda, donde se ofrece información básica sobre el funcionamiento de la plataforma, además de una dirección de correo para sugerencias, dudas o reporte de errores. También se incluyen preguntas frecuentes (FAQ) y enlaces de soporte (Figura 46).

5.3.2 Problemas encontrados y soluciones

A lo largo del desarrollo del proyecto no se han identificado problemas críticos que comprometieran su viabilidad, pero sí se han afrontado diversos desafíos técnicos y de diseño que han influido en el ritmo y alcance de la implementación.

Uno de los aspectos más complejos ha sido el diseño de la base de datos y la estructuración de la información extraída de las partidas públicas. Las replays de Pokémon Showdown [9] [10] no se ofrecen en un formato estructurado, sino como **texto plano con líneas de eventos**, lo que obliga a aplicar un tratamiento especial para convertir esa información en datos útiles.

Para abordar este reto, se ha implementado un sistema de **parsing personalizado línea por línea**, en el que cada tipo de evento del log es interpretado mediante expresiones regulares específicas. El sistema detecta e identifica patrones como:

- Cambios de Pokémon (switch, drag)
- Movimientos realizados (move)
- Daño recibido (-damage)

- Cambios de estado (-status, -curestatus, etc.)
- Activaciones de habilidades y objetos (-activate, -item, -ability)
- Cambios de estadísticas (-boost)
- Climas y efectos de campo (-weather, -fieldstart, -fielddend)

Cada línea es analizada para extraer las entidades implicadas (jugador, Pokémon, acción, objetivo, cantidad de daño, etc.) y traducida a una estructura JSON que refleja el estado de la partida turno a turno.

Además, se han diseñado **mecanismos de actualización incremental del estado de combate**, de forma que se conserva el historial de lo que ha sucedido hasta cada momento (por ejemplo, qué Pokémon ya se han revelado, cuántos siguen activos, si un Pokémon está quemado, etc.).

Para lidiar con la gran variedad de interacciones posibles entre más de 1000 Pokémon, cientos de habilidades y objetos, se ha optado por una estructura de datos lo suficientemente flexible como para permitir extensiones futuras. Las entidades se modelan de forma genérica (por ejemplo, eventos de combate con distintos atributos opcionales) y se ha centralizado la lógica de interpretación por tipo de evento, lo que facilita el mantenimiento y ampliación del sistema.

Otro obstáculo significativo ha sido la imposibilidad de acceder a muchas partidas de alto nivel competitivo debido a que los usuarios pueden marcarlas como privadas. Esta funcionalidad impide que dichas replays se almacenen en la base de datos para su posterior análisis. Se han explorado posibles alternativas para permitir su incorporación de forma privada o local a la colección del usuario, como un sistema de subida manual de archivos .json o .txt, pero no se ha encontrado una solución adecuada dentro del tiempo disponible para el desarrollo del proyecto.

Relacionado con esto, pero no del todo, cabe mencionar que algunas replays públicas no cuentan con el rating de la partida y no se ha encontrado documentación que explique claramente por qué sucede esto.

Asimismo, la implementación del asistente de turnos (Turn Assistant) ha supuesto una de las funcionalidades más exigentes en términos de complejidad. Este módulo requiere no solo simular un estado de partida, sino también evaluar diferentes posibilidades y determinar cuál es la opción más favorable en cada contexto. El diseño de esta herramienta ha implicado la creación de una lógica avanzada de análisis y comparación, lo que ha ralentizado parcialmente el progreso del proyecto. A pesar de ello, se ha logrado construir una base funcional que puede ser ampliada con algoritmos más sofisticados en el futuro.

5.3.3 Pruebas y resultados

Se han realizado pruebas tanto de la API, con Postman [21]; como de accesibilidad, sobre todo con WAVE [22] y también con TAW [23].

Postman [21]

Para la verificación y validación de los endpoints desarrollados en la API del proyecto, se ha utilizado la herramienta Postman [21], ampliamente reconocida en el ámbito del desarrollo backend por su utilidad para realizar pruebas de peticiones HTTP de manera eficiente.

Se ha creado una colección específica en Postman [21] que agrupa todas las pruebas realizadas sobre las distintas funcionalidades implementadas, como la inserción y consulta de partidas, análisis, y gestión de datos en la base de datos. Esta colección incluye tanto peticiones GET como POST, PUT y DELETE, según los métodos habilitados por la API.

Se entrega un archivo `PokemonStatistics API.postman_collection.json` exportado desde Postman [21] que permite importar directamente dicha colección en cualquier entorno local. Este archivo contiene todas las pruebas documentadas y organizadas, facilitando la replicación y validación del funcionamiento del sistema por parte del tribunal evaluador o futuros desarrolladores interesados.

WAVE [22]

WAVE [22] (Web Accessibility Evaluation Tool) es una herramienta desarrollada por WebAIM (Web Accessibility In Mind) con el objetivo de facilitar la evaluación de la accesibilidad de páginas web. Su propósito principal es ayudar a diseñadores, desarrolladores y evaluadores a identificar errores y oportunidades de mejora en el cumplimiento de las pautas de accesibilidad, especialmente las definidas por las WCAG [24] (Web Content Accessibility Guidelines).

WAVE [22] permite analizar una página web directamente desde su sitio web oficial o mediante una extensión del navegador (disponible para Chrome y Firefox), proporcionando un análisis visual de los elementos accesibles e inaccesibles.

En el contexto de este proyecto, WAVE [22] se ha utilizado para verificar la accesibilidad de la aplicación desarrollada, asegurando que se cumplen los principios de perceptibilidad, operabilidad, comprensibilidad y robustez, tal como exigen las normativas actuales de accesibilidad web.

A continuación, se muestran los resultados de todas las páginas de las pruebas realizadas con WAVE [22], con sus respectivas figuras. Mencionar que el warning común de todas las páginas es uno sin importancia que no se puede quitar y que los errores de contraste que aparecen en los rankings son de los colores de las líneas de los grafos, que hay algunos que dan problemas, pero no he encontrado colores suficientemente diferentes al esto que también tengan buen contraste con el fondo: Home (Figura 47), Sign In (Figura 48), Sign Up (Figura 49), Profile (Figura 50), Public Games (Figura 51), Saved Games (Figura 52, Figura 53 y Figura 54), Analyze Battle (Figura 55), Battle Analytics (Figura 56 y Figura 57), Rankings (Figura 58 y Figura 59), Turn Assistant (Figura 60, Figura 61, Figura 62, Figura 63 y Figura 64), Forum (Figura 65 y Figura 66) y Contact/Help (Figura 67).

6. Análisis económico y viabilidad legal

6.1 Costes actuales del sistema

Durante la fase de desarrollo y pruebas, el proyecto ha generado un coste mensual aproximado asociado a distintos servicios de Google Cloud utilizados para su funcionamiento:

- **BigQuery [5]:** almacenamiento de datos y ejecución de consultas sobre partidas.

Coste estimado: **1-2 €/mes.**

- **Cloud Scheduler [6] y Cloud Functions:** automatización del análisis de nuevas partidas y ejecución programada de scripts.

Coste aproximado: **2-3 €/mes.**

- **Artifact Registry:** almacenamiento de artefactos y dependencias para despliegue de funciones. Coste estimado: **1 €/mes.**

Con estos servicios activos, el gasto mensual total se sitúa actualmente entre **4 y 5 €**. En el caso de desplegar y mantener permanentemente la aplicación web y el backend, el coste podría alcanzar los **10 € mensuales**, especialmente si aumenta la actividad o la cantidad de datos procesados.

6.2 Costes en caso de escalar o comercializar

En un escenario donde la aplicación se comercialice y reciba cientos o miles de usuarios diarios, los costes pueden aumentar de forma considerable:

- **BigQuery [5] (lectura y almacenamiento intensivo):** puede escalar hasta decenas de euros mensuales, especialmente si se realizan análisis complejos y frecuentes.
- **Cloud Run [7]:** escalaría con el número de peticiones y tráfico.
- **Base de datos más robusta** (si se necesita un sistema alternativo más económico, como PostgreSQL en Cloud SQL).
- **Mantenimiento, dominio, CDN, autenticación avanzada, etc.:** adicionales si se busca un entorno más profesional.

Un coste estimado en fase de expansión mínima rondaría los **20-50 € mensuales**.

6.3 Viabilidad legal de la comercialización

Este proyecto utiliza fuentes de datos públicos de terceros no oficiales, como:

<https://replay.pokemons showdown.com>

<https://www.smogon.com/stats>

Ambas están relacionadas con la franquicia Pokémon, cuyos derechos pertenecen a Nintendo, Game Freak y Creatures Inc. Por este motivo, **no sería legal comercializar el proyecto de forma directa**, dado que:

- Se emplean nombres, mecánicas y elementos visuales protegidos por copyright.
- No existe licencia oficial para el uso de dicha propiedad intelectual.
- Se basan en datos derivados de plataformas no oficiales.

Comercialización indirecta mediante donaciones

Una posible alternativa es habilitar **métodos de apoyo voluntario** a través de plataformas como:

- **Patreon**
- **Ko-fi**
- **Buy Me a Coffee**
- **PayPal Donations**

Este modelo es empleado por diversas páginas del entorno competitivo de Pokémon, como por ejemplo **Pikalytics**, que también monetiza su actividad mediante **publicidad y donaciones** sin haber sido sancionada por Nintendo hasta la fecha. Esto demuestra que, mientras no se haga uso de contenido visual oficial (como sprites, logotipos o nombres exactos con marca registrada) ni se pretenda replicar o competir directamente con productos oficiales, es posible operar de forma segura dentro de esta “zona gris” legal.

Modelos adicionales

Además de las donaciones, pueden implementarse otros métodos de monetización que no infringen directamente los derechos de autor, como:

- **Publicidad no intrusiva** en la plataforma, al estilo de lo que hacen páginas como Smogon o Pikalytics.
- **Suscripciones premium**, con distintas modalidades:
 - Una suscripción básica para eliminar publicidad.
 - Una suscripción avanzada que desbloquee funcionalidades adicionales, como análisis detallado, reportes personalizados o un límite diario mayor de análisis automáticos.

Este tipo de modelo freemium permite ofrecer valor real a los usuarios sin depender directamente de contenido con derechos protegidos.

Conclusión

La comercialización directa del proyecto, usando elementos oficiales de Pokémon, **no es legal sin licencia expresa**. Sin embargo, existen formas indirectas de obtener beneficios —como donaciones, publicidad o suscripciones premium— que han sido utilizadas por otros sitios similares sin problemas legales hasta el momento. Aun así, se recomienda **evitar el uso de imágenes, logotipos o sprites oficiales**, así como incluir un **disclaimer legal claro** que indique que la plataforma no está afiliada a Nintendo ni a Pokémon Showdown.

7. Conclusiones y líneas futuras

7.1 Conclusiones

El desarrollo de este proyecto ha permitido cumplir con los principales objetivos planteados inicialmente, centrados en la creación de una aplicación web capaz de analizar partidas de Pokémon Showdown [9] [10] para extraer datos relevantes, organizarlos y presentarlos de forma útil y comprensible para el usuario. A través del uso combinado de tecnologías como React [3], Node.js [1], Firebase [2] y Google Cloud, se ha logrado construir una arquitectura funcional, escalable y preparada para futuras ampliaciones.

Uno de los principales logros ha sido el diseño e implementación de un sistema de análisis de partidas que, a partir de datos no estructurados en formato texto, es capaz de reconstruir el flujo del combate turno a turno, registrar los Pokémon revelados, habilidades, objetos, condiciones de estado y eventos relevantes. Este análisis se almacena en una base de datos optimizada y es accesible para su visualización o uso posterior, incluso en herramientas como el Turn Assistant.

Desde el punto de vista del aprendizaje, el proyecto ha supuesto un reto integral en diversas áreas del desarrollo de software. Se ha profundizado en aspectos como el diseño de bases de datos, la gestión de peticiones HTTP, la manipulación de datos no estructurados, la programación asíncrona, el despliegue de contenedores en la nube, la planificación de tareas automáticas y el uso de herramientas de diseño de interfaces y prototipado. Además, se han desarrollado habilidades para trabajar con APIs externas, gestionar autenticación de usuarios y plantear soluciones escalables y reutilizables.

Cabe destacar que la mayoría de las tecnologías utilizadas en el desarrollo del proyecto eran completamente nuevas para mí, lo cual ha representado un reto formativo significativo. Tecnologías como Node.js [1] para el backend, Firebase [2] para la autenticación y los diferentes servicios de Google Cloud Platform (BigQuery [5], Cloud Run [7], Cloud Scheduler [6]) fueron exploradas desde cero. Incluso herramientas más específicas como MUI [4] para React [3] o servicios de despliegue en contenedores fueron utilizadas por primera vez. Tan solo tenía experiencia previa con React [3] en el frontend. Este enfoque de explorar y aprender distintas tecnologías ha sido uno de los ejes del TFG, permitiéndome evaluar sus puntos fuertes y también sus limitaciones. Por ejemplo, BigQuery [5] ha demostrado ser muy potente en el análisis de grandes volúmenes de datos, pero no es adecuado para consultas rápidas como las necesarias en el sistema de chats, lo que ha obligado a replantear partes del diseño.

Esta experiencia me ha proporcionado una visión más crítica y práctica de la selección tecnológica en función de las necesidades concretas del sistema.

Entre las principales limitaciones encontradas, cabe destacar la dificultad de obtener partidas de alto nivel de forma sistemática, debido a la opción de los jugadores de hacerlas privadas, lo que limita la muestra de datos disponible. Asimismo, el desarrollo de funcionalidades complejas como el Turn Assistant ha requerido una cantidad de tiempo considerable, lo cual ha afectado a la implementación de otras características inicialmente previstas.

En resumen, el proyecto ha sido una experiencia muy enriquecedora tanto a nivel técnico como personal, permitiendo aplicar conocimientos adquiridos durante la titulación, enfrentarse a problemas reales de desarrollo y sentar las bases para un sistema con gran potencial de evolución y utilidad dentro del ámbito competitivo de Pokémon.

7.2 Líneas futuras

Con el objetivo de seguir mejorando y ampliando la funcionalidad de la aplicación, se proponen las siguientes líneas de trabajo a futuro:

- **Implementación de una calculadora de daño integrada**, similar a las disponibles en otras plataformas especializadas, que permita simular enfrentamientos entre Pokémon en tiempo real utilizando las estadísticas recogidas.
- **Desarrollo e integración de pruebas automáticas**, incluyendo pruebas unitarias, de integración y pruebas end-to-end (E2E), para garantizar la estabilidad, escalabilidad y correcto funcionamiento del sistema.
- **Traducción a otros idiomas**
- **Creación de una herramienta para la organización y gestión de torneos**, incluyendo funciones como inscripción de jugadores, creación de brackets, seguimiento en tiempo real y recopilación de estadísticas específicas por torneo.
- **Pantalla de creación de equipos**, que ofrezca sugerencias automáticas de Pokémon y estrategias basadas en estadísticas de uso, combinaciones de tipos y sinergias con el equipo actual del usuario.
- **Optimización del backend y del sistema de consultas**, con el fin de mejorar el rendimiento en la obtención, almacenamiento y análisis de datos, especialmente a medida que crece la cantidad de partidas y usuarios.
- **Actualización continua de la aplicación**, incorporando nuevas mecánicas, funciones y datos relevantes conforme se lancen nuevas entregas de juegos de Pokémon y evolucione el metajuego competitivo.

- **Ampliación y mejora de las funcionalidades del asistente de turnos (Turn Assistant)**, incluyendo nuevas recomendaciones basadas en situaciones más complejas y opciones más personalizadas según el estilo de juego del usuario.
- **Integración de contenidos informativos sobre el competitivo**, tales como artículos, guías, vídeos, entrevistas o reglas actualizadas, que sirvan como referencia para jugadores novatos y experimentados.
- **Recopilación y análisis de estadísticas de torneos oficiales**, para complementar las estadísticas de ladder y ofrecer un panorama más completo del entorno competitivo.
- **Incorporación de un sistema de mensajería privada entre jugadores**, que facilite la comunicación directa y segura dentro de la plataforma, fomentando la interacción entre la comunidad.
- **Migración del sistema de chat y almacenamiento de replays a una base de datos especializada**, diseñada para soportar altos volúmenes de datos multimedia de forma más eficiente.
- **Mover estilo a theme.js**, ahora mismo hay algunas cosas en theme.js pero se podrían ajustar muchas cosas más en él, para que quede más claro y siga todo el mismo patrón.
- **Organización en el backend**, ya que ahora mismo se hacen todas las llamadas en el mismo archivo y queda bastante extenso.

Bibliografía

Documentación de tecnologías utilizadas

- [1] Node.js documentation. Disponible en: <https://nodejs.org/en/docs>
- [2] Firebase documentation. Disponible en: <https://firebase.google.com/docs/auth>
- [3] React documentation. Disponible en: <https://react.dev/>
- [4] MUI documentation. Disponible en: <https://mui.com/material-ui/getting-started/overview/>
- [5] BigQuery documentation. Disponible en: <https://cloud.google.com/bigquery/docs/introduction?hl=es>
- [6] Cloud Scheduler documentation. Disponible en: <https://cloud.google.com/scheduler/docs?hl=es-419>
- [7] Cloud Run documentation. Disponible en: <https://cloud.google.com/run/docs/overview/what-is-cloud-run?hl=es-419>
- [8] Express.js documentation. Disponible en: <https://expressjs.com/>
- [9] Pokémon Showdown replay viewer. Disponible en: <https://replay.pokemonshowdown.com>
- [10] Pokémon Showdown API. Disponible en: <https://github.com/smogon/pokemon-showdown-client/blob/master/WEB-API.md>
- [11] Pokémon sprites. Disponible en: <https://play.pokemonshowdown.com/sprites/>
- [12] Smogon – Usage Statistics. Disponible en: <https://www.smogon.com/stats/>

Herramientas de diseño y prototipado

[13] Figma documentation. Disponible en: <https://help.figma.com/>

[14] Coolors. Disponible en: <https://coolors.co/>

[15] IFML – Interaction Flow Modeling Language
OMG (Object Management Group). IFML Specification. Disponible en:
<https://www.omg.org/spec/IFML/>

[16] IFMLEdit – Herramienta de modelado para IFML. Sitio web:
<https://ifmledit.org>

Herramientas de desarrollo y colaboración

[17] GitHub Docs. Disponible en: <https://docs.github.com/>

[18] VS Code documentation. Disponible en: <https://code.visualstudio.com/docs>

Herramientas de modelado y gestión

[19] MagicDraw documentation. Disponible en:
<https://www.nomagic.com/products/magicdraw>

[20] Trello. Disponible en: <https://trello.com/>

Herramientas de pruebas y accesibilidad

[21] Postman Learning Center. Disponible en: <https://learning.postman.com/>

[22] WAVE by WebAIM. Disponible en: <https://wave.webaim.org/>

[23] TAW. Disponible en: <https://www.tawdis.net/>

[24] WCAG 2.1. Disponible en: <https://www.w3.org/TR/WCAG21/>

Apéndices

I. Manual de usuario

Este manual tiene como objetivo guiar al usuario final en el uso de las principales funcionalidades de la aplicación web. Las instrucciones se enfocan en el flujo típico que seguirá un usuario:

- **Registro/inicio de sesión y perfil:**
 - Acceder desde la barra lateral a Sign In o Sign Up y escribir las credenciales (Figura 4).

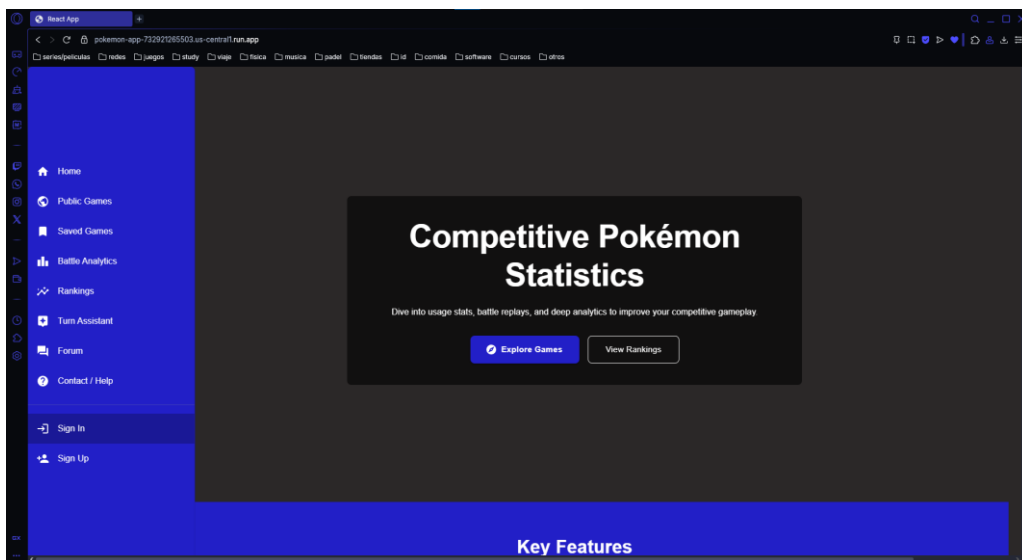


Figura 4. Selección de Sign In en barra lateral desde Home.

- Ahora se mostrará el botón para acceder al perfil donde antes se encontraban Sign In y Sign Up (Figura 5).

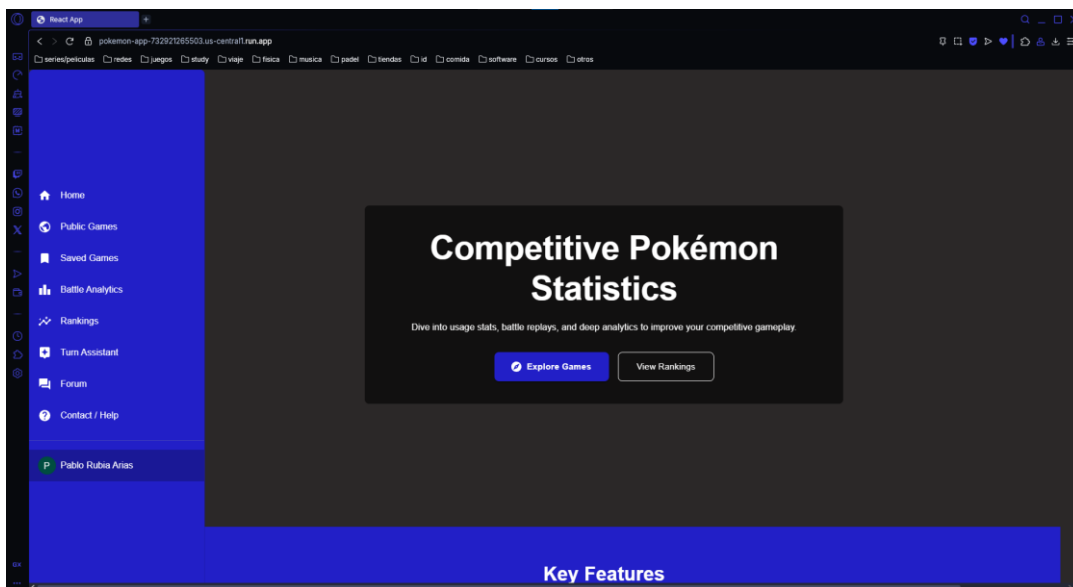


Figura 5. Selección de Profile desde apartado Home.

- Ver detalles de una o varias partidas:
 - Estando autenticado, ir a Public Games y ahí guardar al menos dos partidas pulsando el icono del corazón (Figura 6).

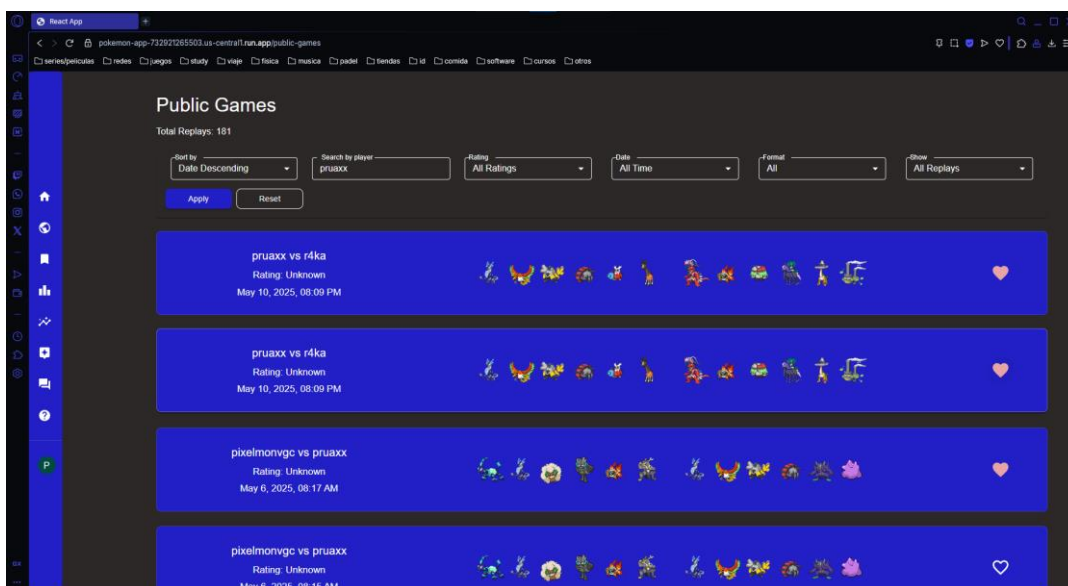


Figura 6. Vista de Public Games con tres partidas guardadas.

- Ir a Saved Games y desde ahí se podrá ver detalles de los turnos de una partida (Analyze Battle, botón central de una partida); seleccionar al menos 2 partidas y pulsar el botón de Battle Analytics en la barra lateral o en la parte inferior de Saved Games; o bien eliminar una partida guardada pulsando el corazón de nuevo (Figura 7).

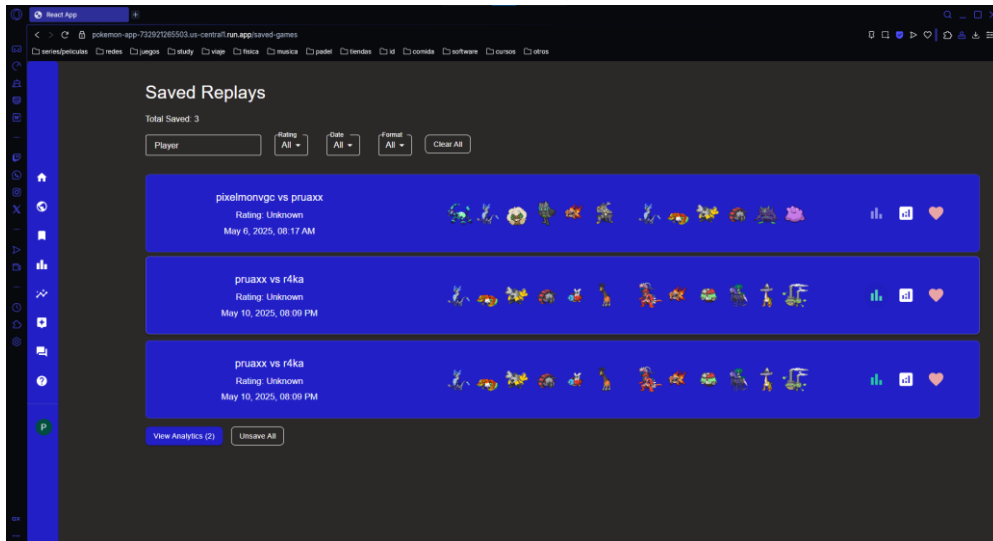


Figura 7. Vista de Saved Games con tres partidas.

- **Ver rankings:**
 - En la barra lateral pulsar la opción de Rankings (Figura 8).

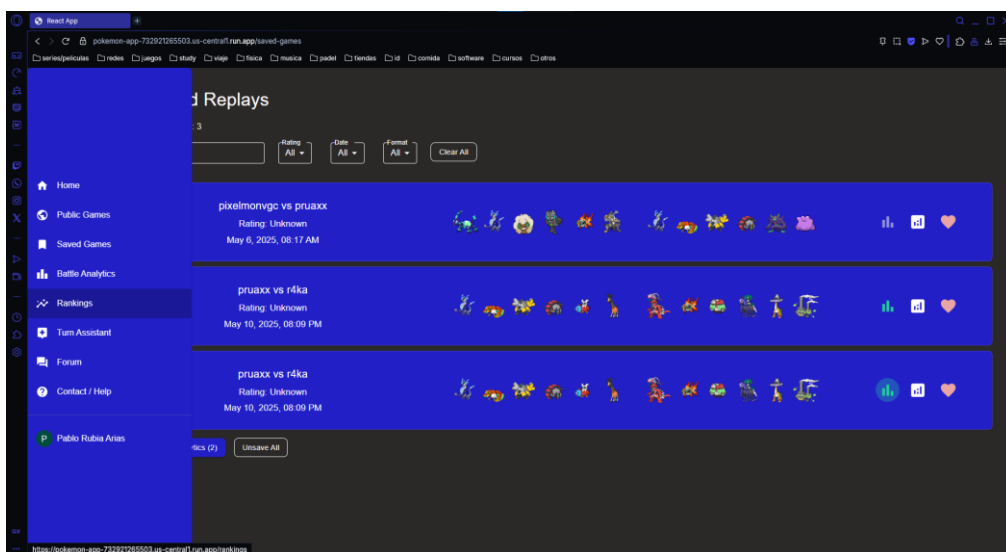


Figura 8. Selección de apartado Rankings.

- Moverse entre los diferentes rankings pulsando los botones de arriba a la derecha: winrate, teams o leads (Figura 9).

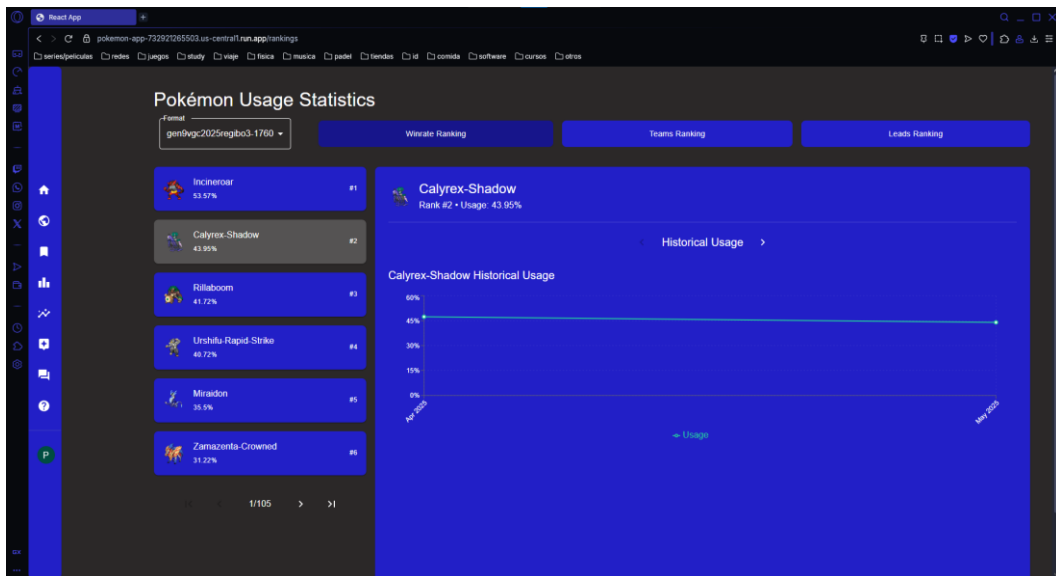


Figura 9. Vista de Rankings de uso.

- **Analizar un turno concreto:**
 - Ir a Turn Assistant desde la barra lateral (Figura 10).

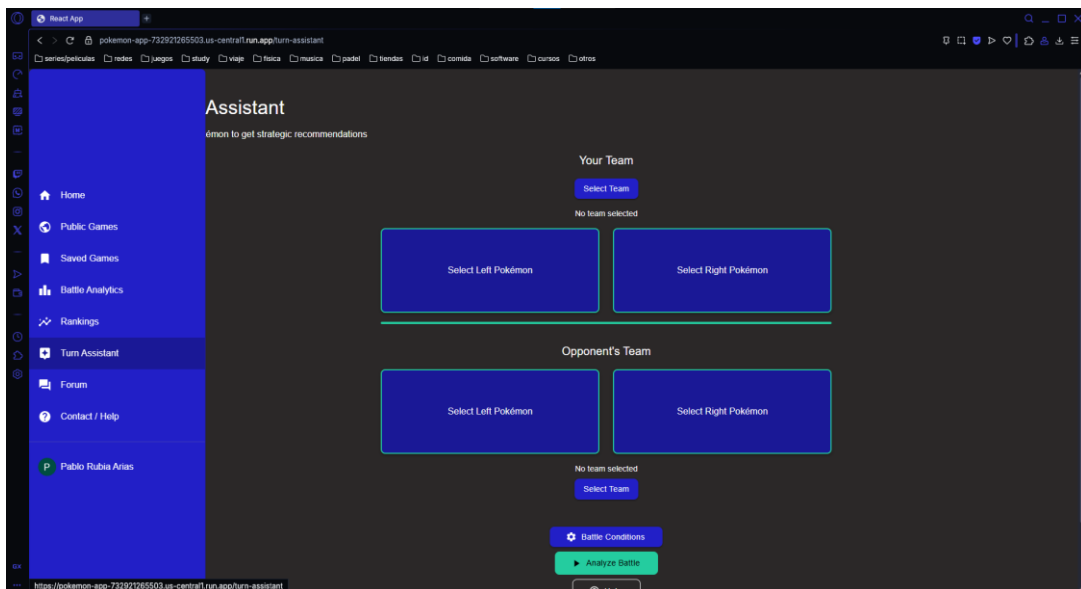


Figura 10. Selección del apartado Turn Assistant.

- Seleccionar los nombres de los cuatro Pokémon activos pulsando en los cuatro cuadrados centrales y buscando el nombre en los selects (Figura 11).

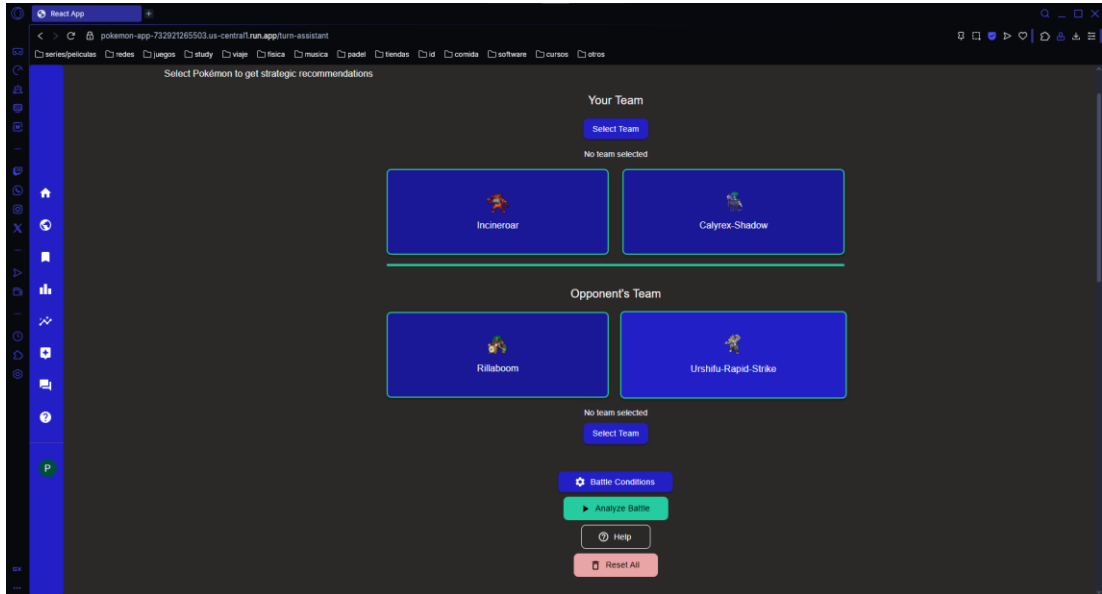


Figura 11. Selección de un escenario de Turn Assistant con cuatro Pokémon.

- Pulsar el botón de Analyze para ver los resultados (Figura 12).

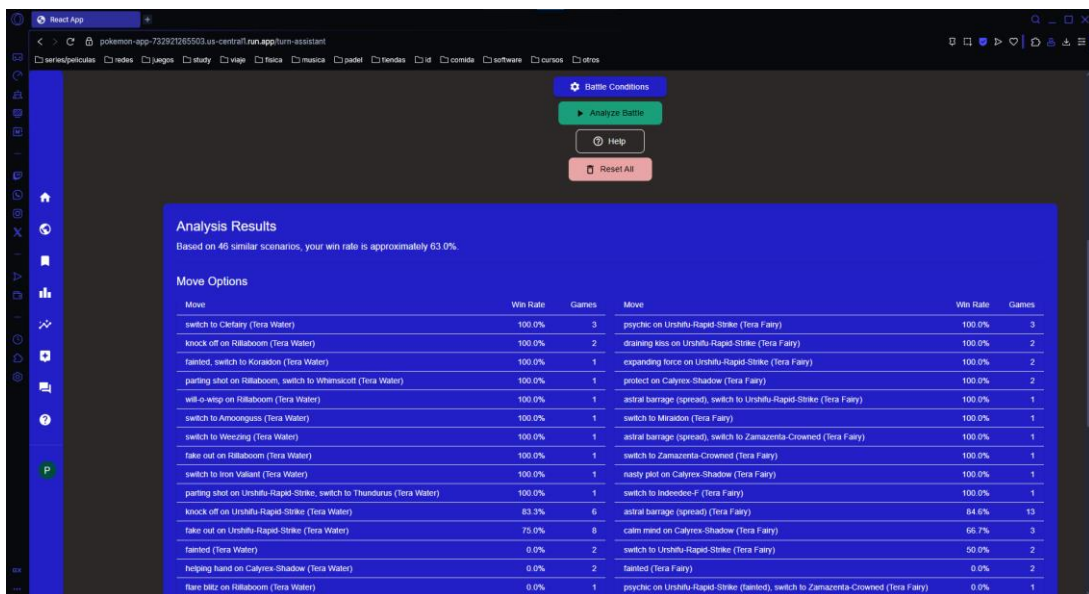


Figura 12. Vista de Turn Assistant una vez pulsado en botón 'Analyze Battle'.

- Opcionalmente, añadir más detalles a la búsqueda de un turno, tanto en los Pokémon activos después de seleccionar un nombre, como en el botón de teams o battle conditions (Figura 13).

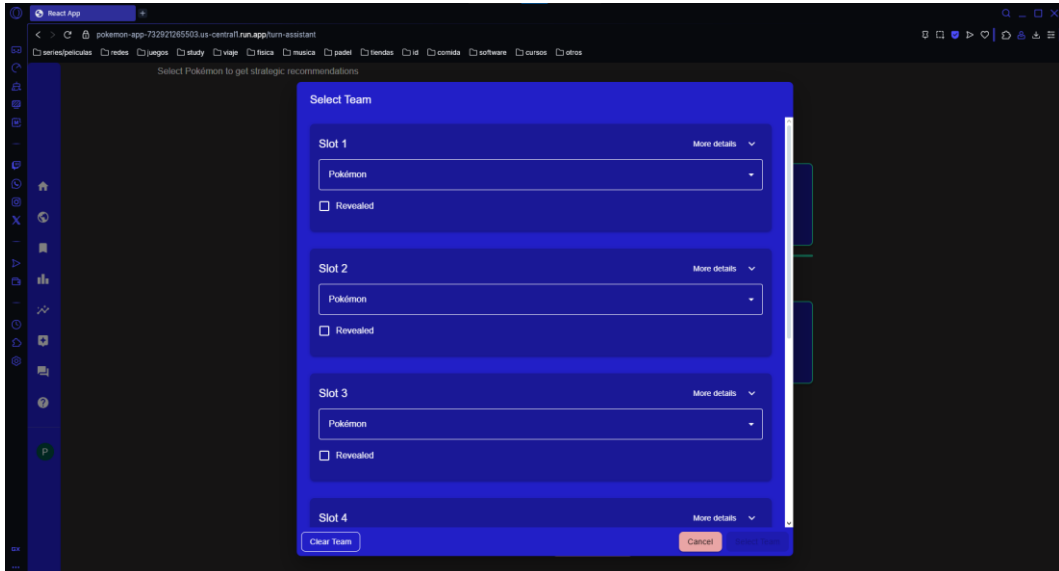


Figura 13. Vista de Turn Assistant en los detalles de un equipo sin datos.

- Adicionalmente se podrá borrar todo lo seleccionado con el botón de clear que se encuentra abajo (Figura 14).

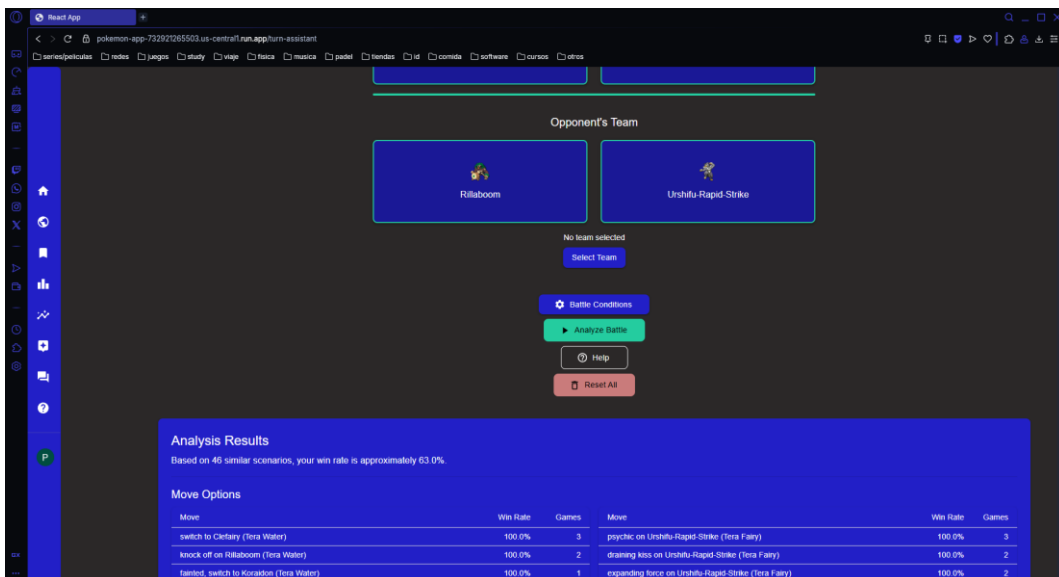


Figura 14. Vista de Turn Assistant mostrando los botones disponibles.

- Escribir en un tópico:
 - Tener la sesión iniciada (Figura 15).

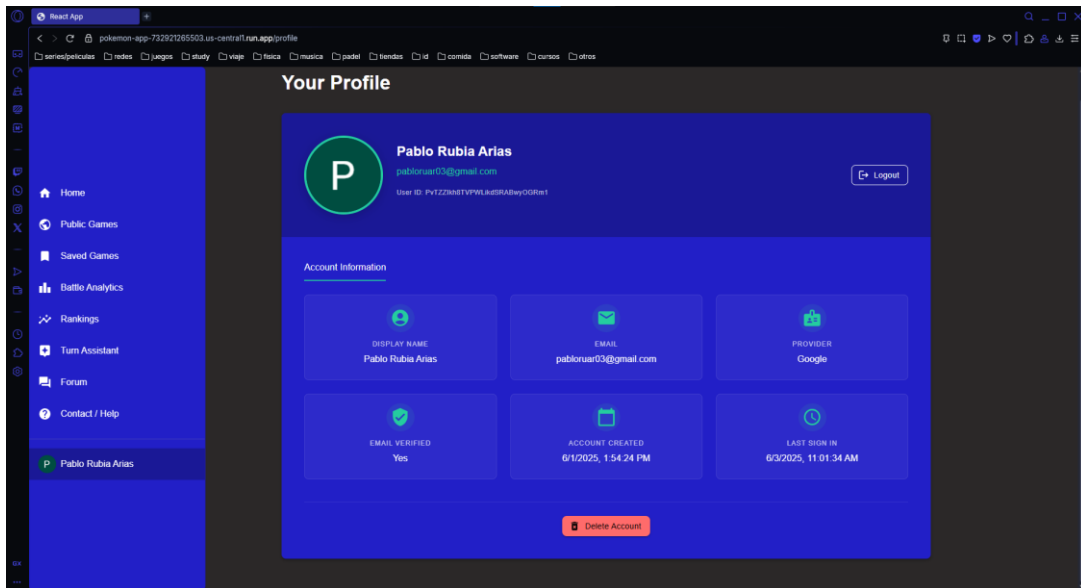


Figura 15. Vista del apartado Profile.

- Ir al apartado Forum de la barra lateral (Figura 16).

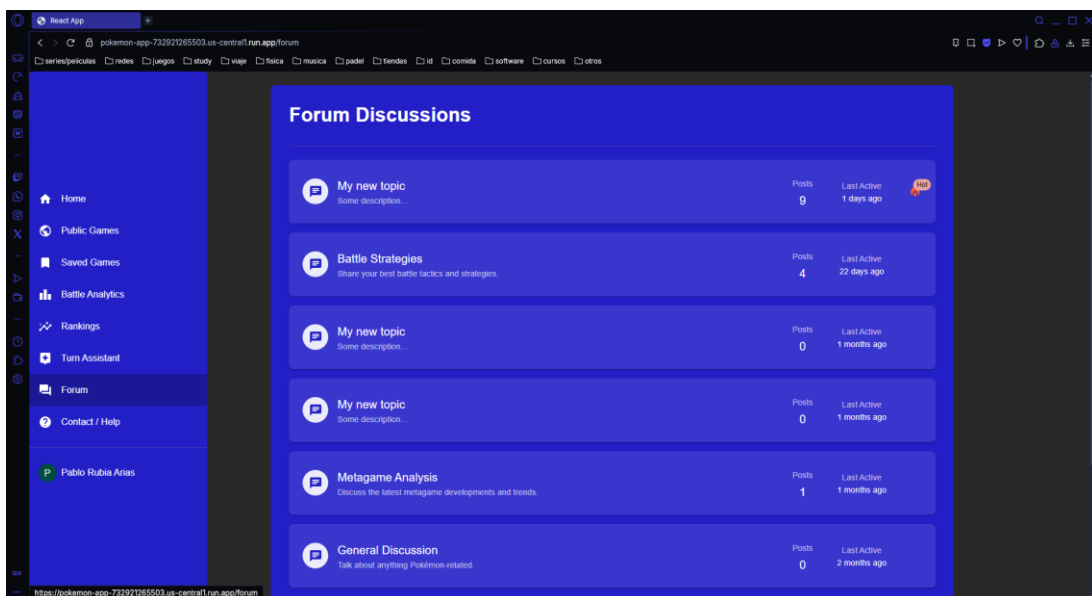


Figura 16. Selección del apartado Forum.

- Elegir un tópico (Figura 17).

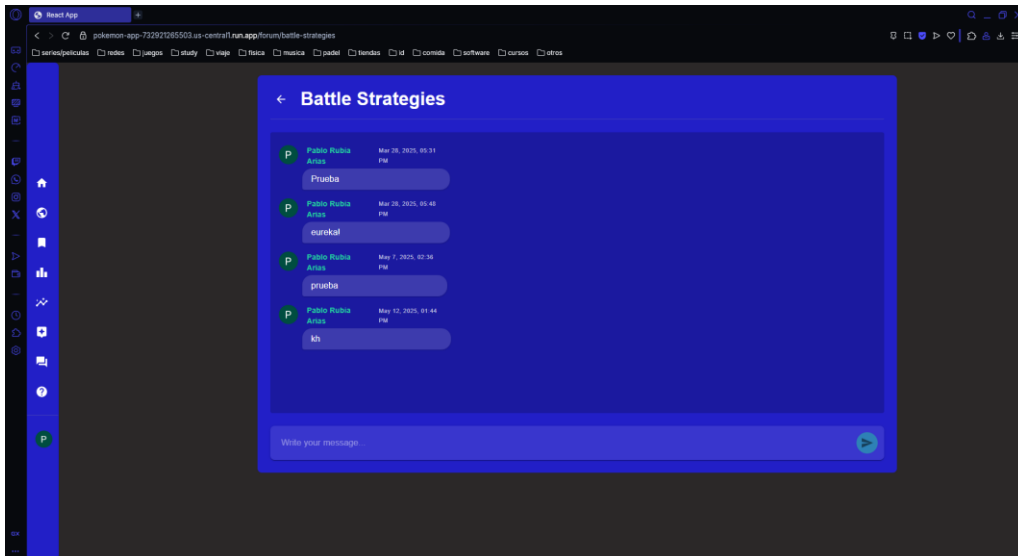


Figura 17. Vista de un Tópico.

- Escribir en el cuadro de texto y pulsar enviar (Figura 18).

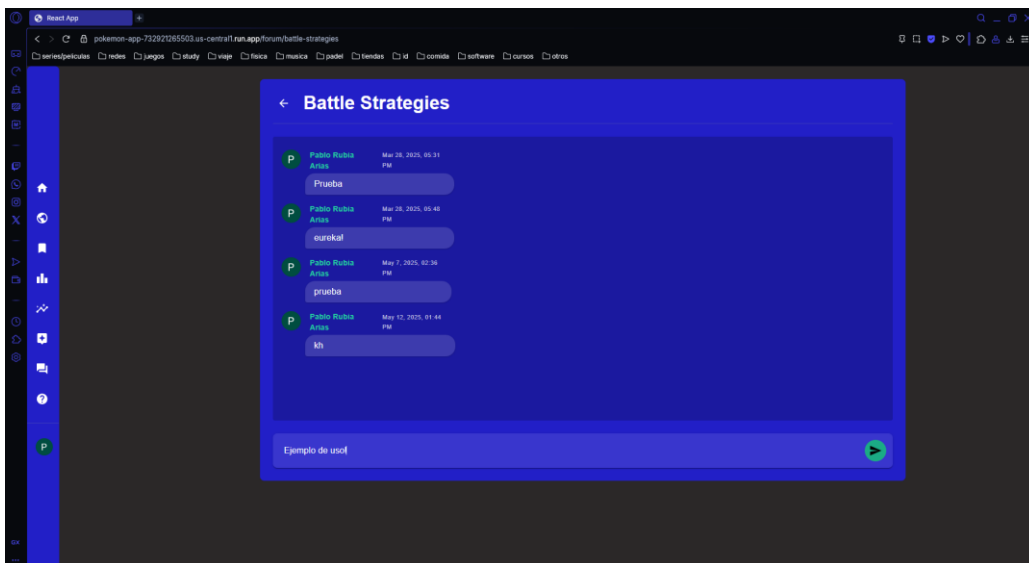


Figura 18. Vista de un chat de un tópico mientras se escribe un mensaje.

II. Manual de instalación/despliegue

II.I De forma remota

Simplemente basta con acceder al enlace del despliegue: <https://pokemon-app-732921265503.us-central1.run.app>

II.II De forma local

Para ejecutar la aplicación de forma local, hay que seguir los siguientes pasos:

- Descargar los archivos del proyecto
- Modificar el código del archivo server.js, en la carpeta backend, en el que hay que descomentar la línea de keyFilename y escribir la ruta en la que se encuentren los archivos del proyecto, referenciando el archivo credentials.json:

```
// Initialize the BigQuery client with explicit credentials
const bigQuery = new BigQuery({
  //keyFilename: "D:/tfg/pokemonStatistics/credentials.json",
});
```

- Para ejecutar el backend:
 - Abrir una nueva terminal en la raíz del proyecto
 - Hacer cd .\backend
 - Hacer node ./server
- Para ejecutar el frontend:
 - Abrir una nueva terminal en la raíz del proyecto
 - Hacer cd .\client
 - Hacer npm install para descargar las dependencias
 - Hacer npm run dev

○ La aplicación se debería haber abierto automáticamente al terminar de ejecutar, si no, ir a la dirección manualmente, que debería ser: <http://localhost:3000>

III. Estructura de la Base de Datos

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags [?]	Description
topic_id	STRING	REQUIRED	-	-	-	-	Unique topic ID
title	STRING	REQUIRED	-	-	-	-	Topic title
description	STRING	NULLABLE	-	-	-	-	Optional description
icon	STRING	NULLABLE	-	-	-	-	Associated visual icon
created_at	TIMESTAMP	REQUIRED	-	-	-	-	Topic creation date
last_active	TIMESTAMP	REQUIRED	-	-	-	-	Date of last message in topic
posts_count	INTEGER	NULLABLE	-	-	0	-	Total number of messages posted

Figura 19. Estructura de la tabla `forum_topics`.

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags [?]	Description
message_id	STRING	REQUIRED	-	-	-	-	Unique message ID
topic_id	STRING	REQUIRED	-	-	-	-	ID of the topic it belongs to
content	STRING	REQUIRED	-	-	-	-	Message text
userId	STRING	REQUIRED	-	-	-	-	ID of the message author
userName	STRING	REQUIRED	-	-	-	-	Name of the author
userAvatar	STRING	NULLABLE	-	-	-	-	URL of the user's avatar
created_at	TIMESTAMP	REQUIRED	-	-	-	-	Message publication date

Figura 20. Estructura de la tabla `forum_messages`.

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags [?]	Description
user_id	STRING	NULLABLE	-	-	-	-	ID of the user who saved the replays
replays_saved	RECORD	REPEATED	-	-	[]	-	List of saved replays
replay_id	STRING	NULLABLE	-	-	-	-	ID of the saved replay

Figura 21. Estructura de la tabla `saved_replays`.

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
replay_id	STRING	REQUIRED	-	-	-	-	Unique replay ID
format	STRING	REQUIRED	-	-	-	-	Competitive format
views	INTEGER	NULLABLE	-	-	-	-	Number of views
rating	INTEGER	NULLABLE	-	-	-	-	Estimated battle rating
player1	STRING	NULLABLE	-	-	-	-	Name of player 1
player2	STRING	NULLABLE	-	-	-	-	Name of player 2
winner	STRING	REQUIRED	-	-	-	-	Battle winner
loser	STRING	REQUIRED	-	-	-	-	Battle loser
date	TIMESTAMP	REQUIRED	-	-	-	-	Battle date
▶ teams	RECORD	NULLABLE	-	-	-	-	Teams of the players
▶ turns	RECORD	REPEATED	-	-	-	-	Battle turns

Figura 22. Estructura general de la tabla replays.

▼ teams	RECORD	NULLABLE	-	-	-	-	Teams of the players
▼ p1	RECORD	REPEATED	-	-	-	-	Player 1's Pokémon
name	STRING	NULLABLE	-	-	-	-	Player1 Pokémon's names
item	STRING	NULLABLE	-	-	-	-	Player1 Pokémon's items
ability	STRING	NULLABLE	-	-	-	-	Player1 Pokémon's abilities
moves	STRING	REPEATED	-	-	-	-	Player1 Pokémon's moves
level	INTEGER	NULLABLE	-	-	-	-	Player1 Pokémon's levels
tera_type	STRING	NULLABLE	-	-	-	-	Player1 Pokémon's tera types
▼ p2	RECORD	REPEATED	-	-	-	-	Player 2's Pokémon
name	STRING	NULLABLE	-	-	-	-	Player2 Pokémon's names
item	STRING	NULLABLE	-	-	-	-	Player2 Pokémon's items
ability	STRING	NULLABLE	-	-	-	-	Player2 Pokémon's abilities
moves	STRING	REPEATED	-	-	-	-	Player2 Pokémon's moves
level	INTEGER	NULLABLE	-	-	-	-	Player2 Pokémon's levels
tera_type	STRING	NULLABLE	-	-	-	-	Player2 Pokémon's tera types

Figura 23. Detalles de los datos de la columna teams de la tabla replays.

▼ turns	RECORD	REPEATED	-	-	-	-	Battle turns
turn_number	INTEGER	NULLABLE	-	-	-	-	Turn number
▶ starts_with	RECORD	NULLABLE	-	-	-	-	Active Pokémon at the start of the turn
▶ ends_with	RECORD	NULLABLE	-	-	-	-	Active Pokémon at the end of the turn
▶ field	RECORD	NULLABLE	-	-	-	-	Terrain type and duration
▶ weather	RECORD	NULLABLE	-	-	-	-	Weather condition and duration
▶ room	RECORD	NULLABLE	-	-	-	-	Active room and duration
▶ screens	RECORD	NULLABLE	-	-	-	-	Active screens and duration
▶ tailwind	RECORD	NULLABLE	-	-	-	-	Tailwind status
▶ spikes	RECORD	NULLABLE	-	-	-	-	Entry hazards
▶ moves_done	RECORD	NULLABLE	-	-	-	-	Moves used by each player
▶ revealed_pokemon	RECORD	NULLABLE	-	-	-	-	Pokémon revealed with status and stats

Figura 24. Detalles de los datos de la columna turns de la tabla replays.

▼ starts_with	RECORD	NULLABLE	-	-	-	-	Active Pokémon at the start of the turn
player1	STRING	REPEATED	-	-	-	-	Player1 active Pokémon's names that starts the turn
player2	STRING	REPEATED	-	-	-	-	Player2 active Pokémon's names that starts the turn
▼ ends_with	RECORD	NULLABLE	-	-	-	-	Active Pokémon at the end of the turn
player1	STRING	REPEATED	-	-	-	-	Player1 active Pokémon's names that ends the turn
player2	STRING	REPEATED	-	-	-	-	Player2 active Pokémon's names that ends the turn
▼ field	RECORD	NULLABLE	-	-	-	-	Terrain type and duration
terrain	STRING	NULLABLE	-	-	-	-	Terrain's name
duration	INTEGER	NULLABLE	-	-	-	-	Terrain's Duration
▼ weather	RECORD	NULLABLE	-	-	-	-	Weather condition and duration
condition	STRING	NULLABLE	-	-	-	-	Weather's name
duration	INTEGER	NULLABLE	-	-	-	-	Weather's duration
▼ room	RECORD	NULLABLE	-	-	-	-	Active room and duration
condition	STRING	NULLABLE	-	-	-	-	Room's name
duration	INTEGER	NULLABLE	-	-	-	-	Room's duration

Figura 25. Detalles de los datos de las columnas starts_with, ends_with, field, weather y room de la tabla replays.

▼ screens	RECORD	NULLABLE	-	-	-	-	Active screens and duration
▼ reflect	RECORD	NULLABLE	-	-	-	-	Reflect type screen
player1	BOOLEAN	NULLABLE	-	-	-	-	Whether Reflect is active on player 1's side
player2	BOOLEAN	NULLABLE	-	-	-	-	Whether Reflect is active on player 2's side
duration1	INTEGER	NULLABLE	-	-	-	-	Remaining turns of Reflect for player 1
duration2	INTEGER	NULLABLE	-	-	-	-	Remaining turns of Reflect for player 2
▼ lightscreen	RECORD	NULLABLE	-	-	-	-	Lightscreen type screen
player1	BOOLEAN	NULLABLE	-	-	-	-	Whether Light Screen is active on player 1's side
player2	BOOLEAN	NULLABLE	-	-	-	-	Whether Light Screen is active on player 2's side
duration1	INTEGER	NULLABLE	-	-	-	-	Remaining turns of Light Screen for player 1
duration2	INTEGER	NULLABLE	-	-	-	-	Remaining turns of Light Screen for player 2
▼ auroraveil	RECORD	NULLABLE	-	-	-	-	Auroraveil type screen
player1	BOOLEAN	NULLABLE	-	-	-	-	Whether Aurora Veil is active on player 1's side
player2	BOOLEAN	NULLABLE	-	-	-	-	Whether Aurora Veil is active on player 2's side
duration1	INTEGER	NULLABLE	-	-	-	-	Remaining turns of Aurora Veil for player 1
duration2	INTEGER	NULLABLE	-	-	-	-	Remaining turns of Aurora Veil for player 2
▼ tailwind	RECORD	NULLABLE	-	-	-	-	Tailwind status
player1	BOOLEAN	NULLABLE	-	-	-	-	Whether Tailwind is active on player 1's side
player2	BOOLEAN	NULLABLE	-	-	-	-	Whether Tailwind is active on player 2's side
duration1	INTEGER	NULLABLE	-	-	-	-	Remaining turns of Tailwind for player 1
duration2	INTEGER	NULLABLE	-	-	-	-	Remaining turns of Tailwind for player 2

Figura 26. Detalles de los datos de las columnas screens y tailwind de la tabla replays.

▼ spikes	RECORD	NULLABLE	-	-	-	-	Entry hazards
▼ player1	RECORD	NULLABLE	-	-	-	-	Entry hazards on player 1's side
spikes	INTEGER	NULLABLE	-	-	-	-	Number of Spikes layers placed (0-3)
toxic_spikes	INTEGER	NULLABLE	-	-	-	-	Number of Toxic Spikes layers placed (0-2)
stealth_rock	BOOLEAN	NULLABLE	-	-	-	-	Whether Stealth Rock is active
sticky_web	BOOLEAN	NULLABLE	-	-	-	-	Whether Sticky Web is active
▼ player2	RECORD	NULLABLE	-	-	-	-	Entry hazards on player 2's side
spikes	INTEGER	NULLABLE	-	-	-	-	Number of Spikes layers placed (0-3)
toxic_spikes	INTEGER	NULLABLE	-	-	-	-	Number of Toxic Spikes layers placed (0-2)
stealth_rock	BOOLEAN	NULLABLE	-	-	-	-	Whether Stealth Rock is active
sticky_web	BOOLEAN	NULLABLE	-	-	-	-	Whether Sticky Web is active
▼ moves_done	RECORD	NULLABLE	-	-	-	-	Moves used by each player
player1	STRING	REPEATED	-	-	-	-	Moves used by player 1 during the turn
player2	STRING	REPEATED	-	-	-	-	Moves used by player 2 during the turn
▼ revealed_pokemon	RECORD	NULLABLE	-	-	-	-	Pokémon revealed with status and stats
▶ player1	RECORD	REPEATED	-	-	-	-	Pokémon revealed from player 1
▶ player2	RECORD	REPEATED	-	-	-	-	Pokémon revealed from player 2

Figura 27. Detalles de los datos de las columnas spikes y moves_done de la tabla replays.

▼ revealed_pokemon	RECORD	NULLABLE	-	-	-	-	Pokémon revealed with status and stats
▼ player1	RECORD	REPEATED	-	-	-	-	Pokémon revealed from player 1
name	STRING	NULLABLE	-	-	-	-	Species name of the Pokémon
nickname	STRING	NULLABLE	-	-	-	-	Nickname used in battle
moves	STRING	REPEATED	-	-	-	-	Known moves of the Pokémon
ability	STRING	NULLABLE	-	-	-	-	Revealed ability
item	STRING	NULLABLE	-	-	-	-	Revealed item
remaining_hp	INTEGER	NULLABLE	-	-	-	-	Remaining HP at last update (0-100)
▶ volatile_status	RECORD	REPEATED	-	-	-	-	Volatile statuses
non_volatile_status	STRING	NULLABLE	-	-	-	-	Persistent status condition
▶ stats	RECORD	NULLABLE	-	-	-	-	Stat boosts/reductions
▶ tera	RECORD	NULLABLE	-	-	-	-	Terastallization status
fighting_status	STRING	NULLABLE	-	-	-	-	Fighting status
consecutive_protect_counter	INTEGER	NULLABLE	-	-	-	-	Number of consecutive Protect-type moves attempted
ability_suppressed	BOOLEAN	NULLABLE	-	-	-	-	Whether the ability is currently suppressed
last_move_used	STRING	NULLABLE	-	-	-	-	Last move used by this Pokémon
mimic_used	BOOLEAN	NULLABLE	-	-	-	-	Whether Mimic was used
copied_move	STRING	NULLABLE	-	-	-	-	Move copied by Mimic
transformed	BOOLEAN	NULLABLE	-	-	-	-	Whether the Pokémon is under Transform

Figura 28. Detalles de los datos de la columna revealed_pokemon de la tabla replays.

▼ volatile_status	RECORD	REPEATED	-	-	-	-	Volatile statuses
name	STRING	NULLABLE	-	-	-	-	Name of the volatile status
turn_counter	INTEGER	NULLABLE	-	-	-	-	Duration counter for the volatile status
non_volatile_status	STRING	NULLABLE	-	-	-	-	Persistent status condition
▼ stats	RECORD	NULLABLE	-	-	-	-	Stat boosts/reductions
atk	INTEGER	NULLABLE	-	-	-	-	Attack stat stage
def	INTEGER	NULLABLE	-	-	-	-	Defense stat stage
spa	INTEGER	NULLABLE	-	-	-	-	Special Attack stat stage
spd	INTEGER	NULLABLE	-	-	-	-	Special Defense stat stage
spe	INTEGER	NULLABLE	-	-	-	-	Speed stat stage
acc	INTEGER	NULLABLE	-	-	-	-	Accuracy stage
eva	INTEGER	NULLABLE	-	-	-	-	Evasion stage
▼ tera	RECORD	NULLABLE	-	-	-	-	Terastallization status
type	STRING	NULLABLE	-	-	-	-	Tera type
active	BOOLEAN	NULLABLE	-	-	-	-	Whether Terastallization is active

Figura 29. Detalles de los datos de las columnas volatile_status, stats y tera dentro de revealed_pokemon de la tabla replays.

IV. Páginas Principales

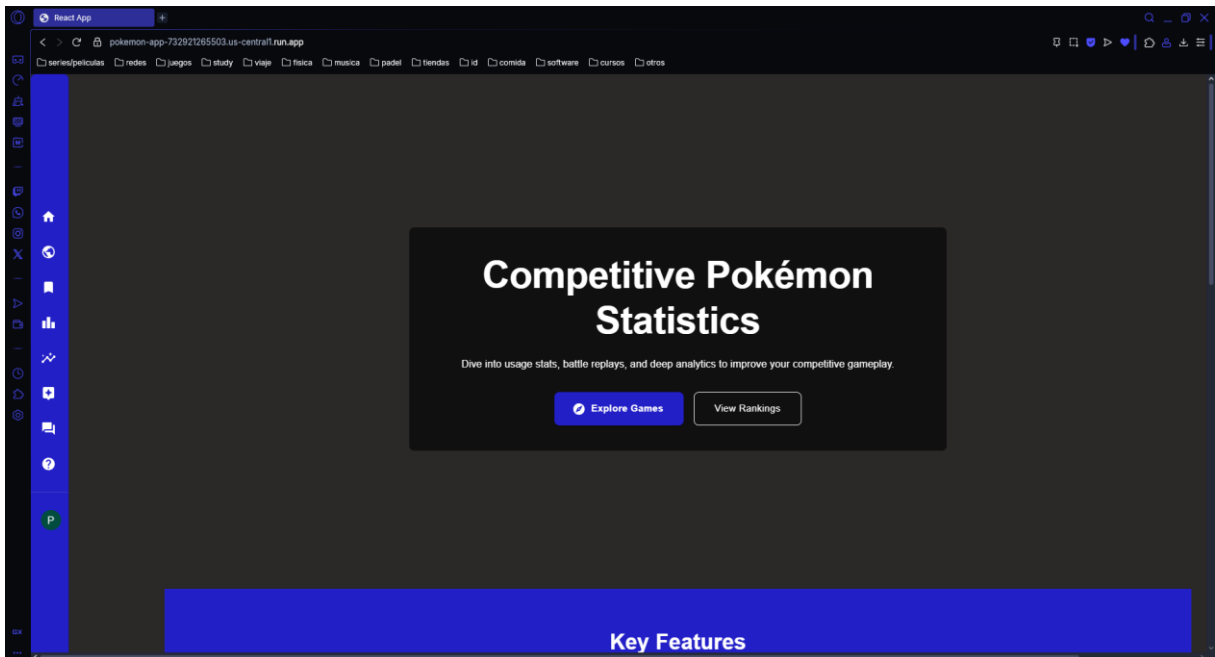


Figura 30. Sección Home.

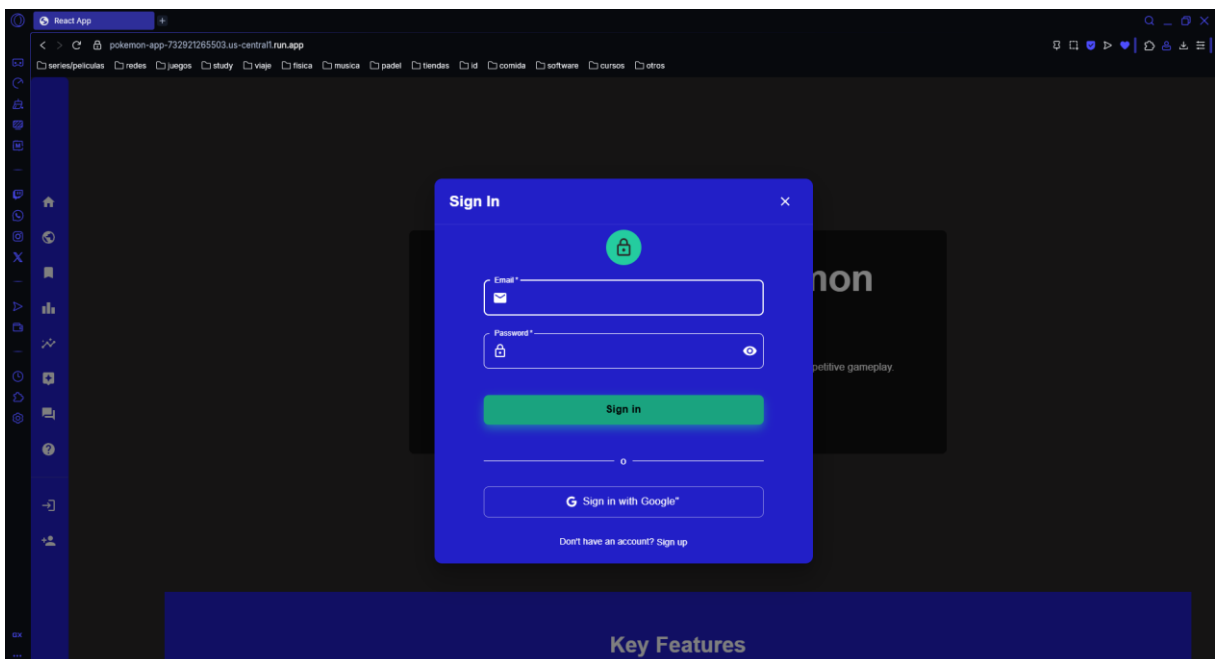


Figura 31. Sección Sign In.

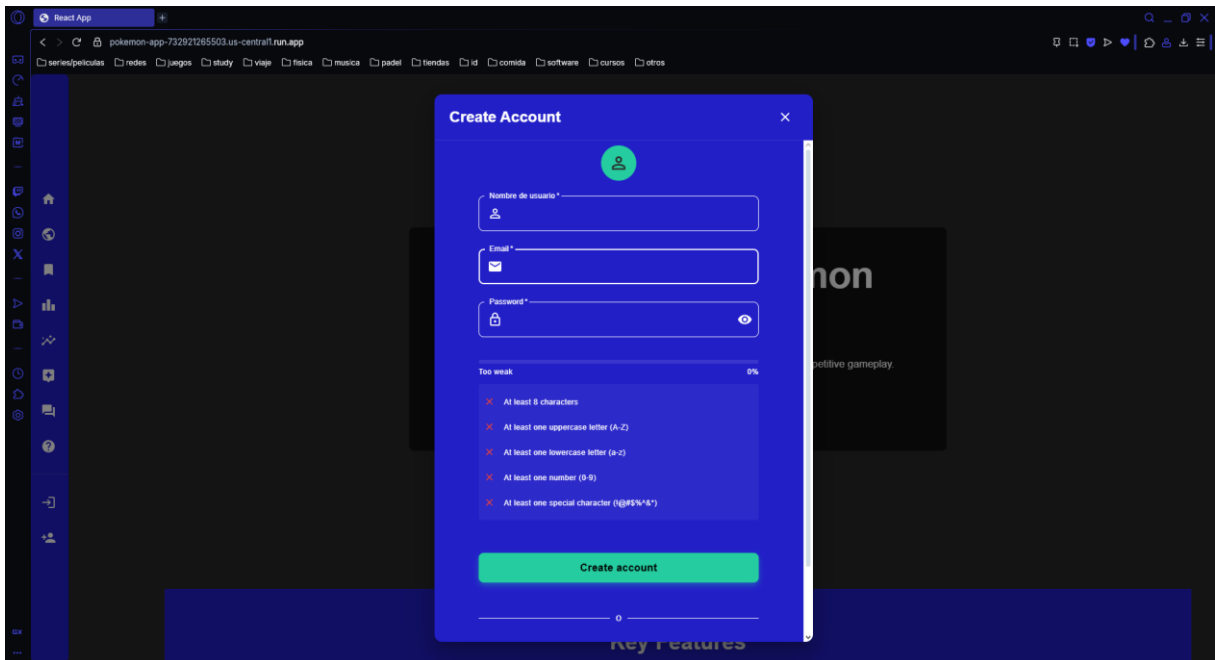


Figura 32. Sección Sign Up.

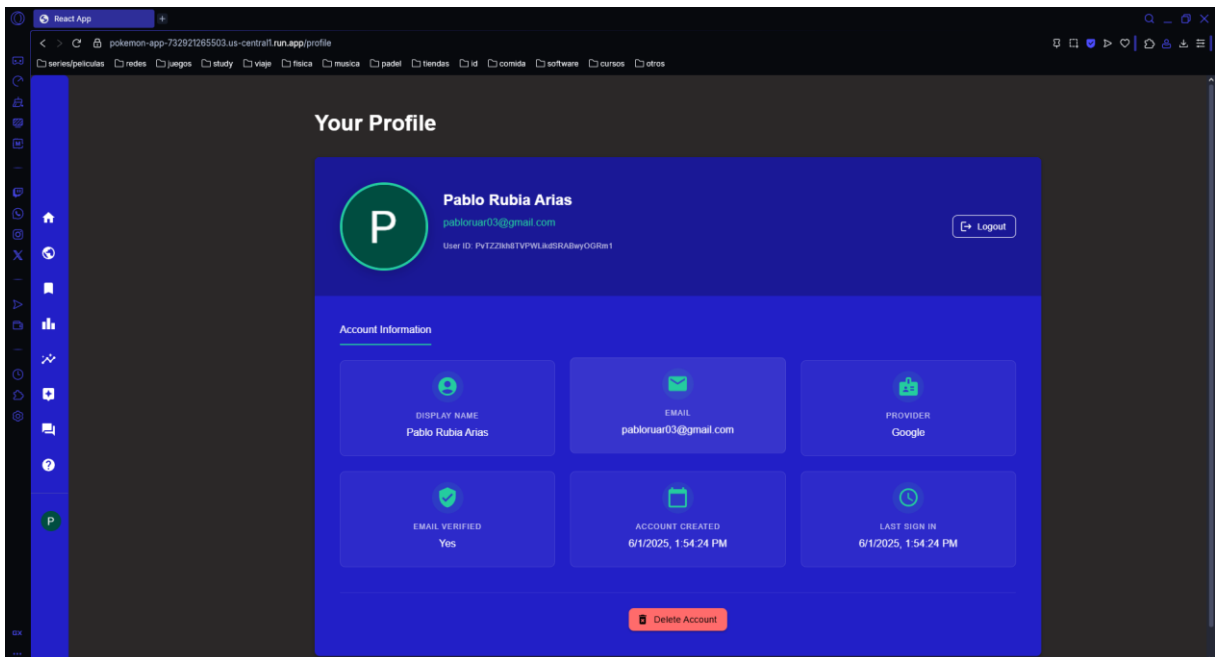


Figura 33. Sección Profile.

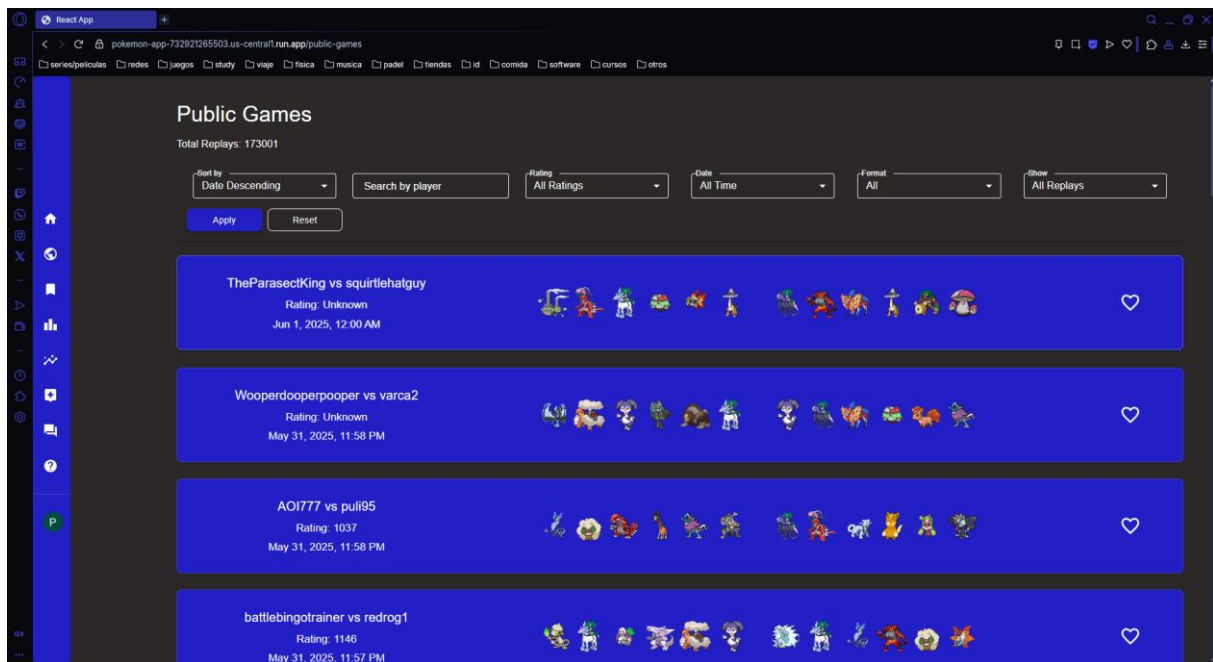


Figura 34. Sección Public Games.

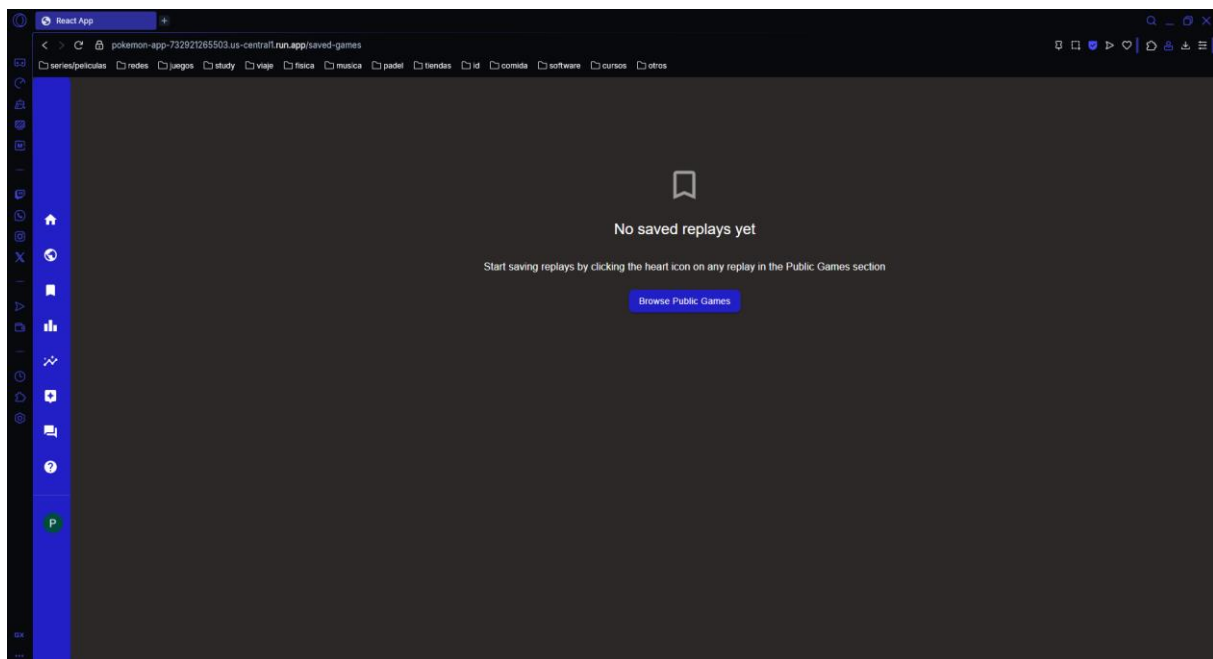


Figura 35. Sección Saved Games si no se ha guardado ninguna partida.

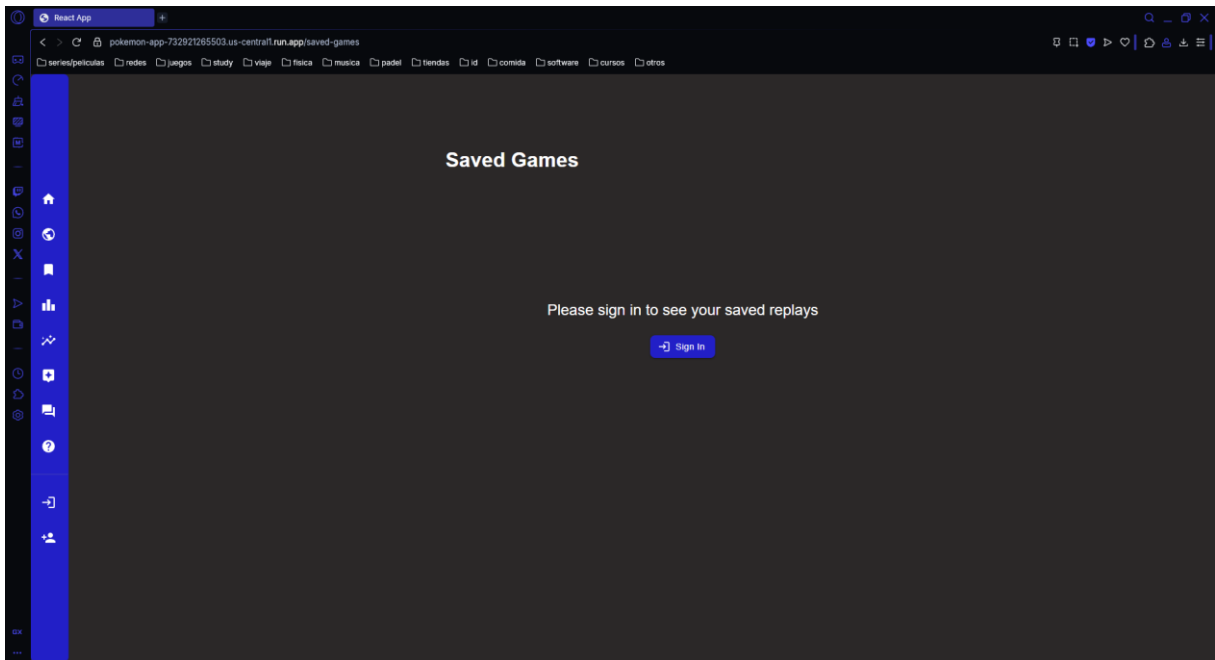


Figura 36. Sección Saved Games si no se ha iniciado sesión.

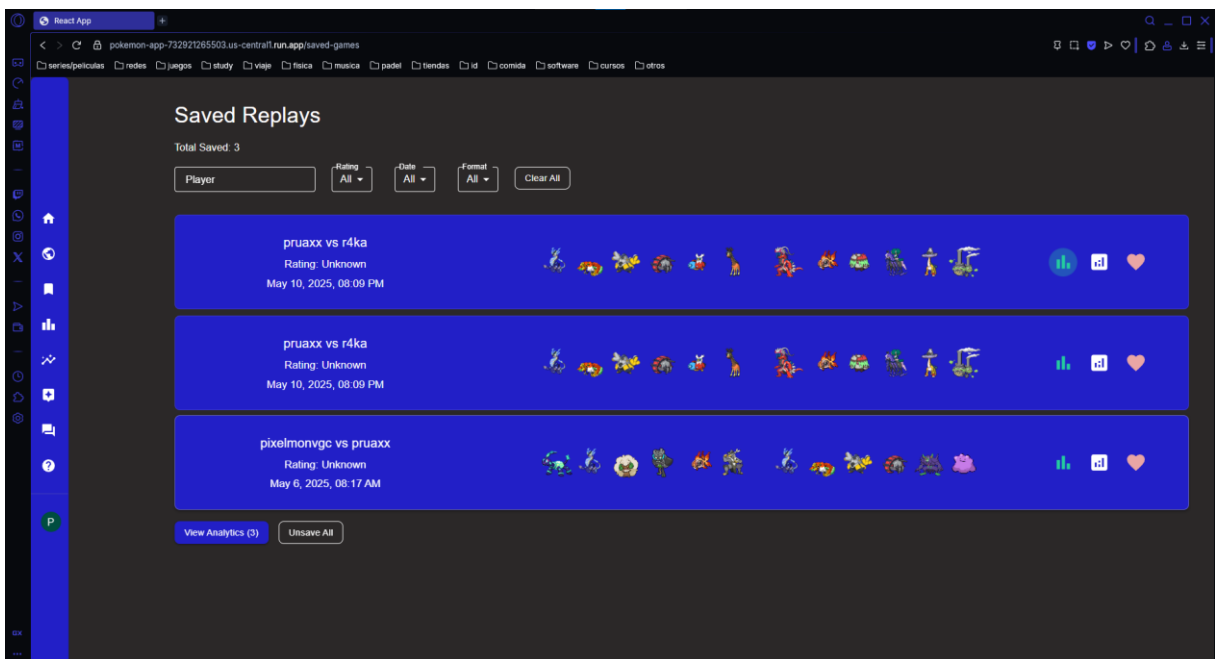


Figura 37. Sección Saved Games.

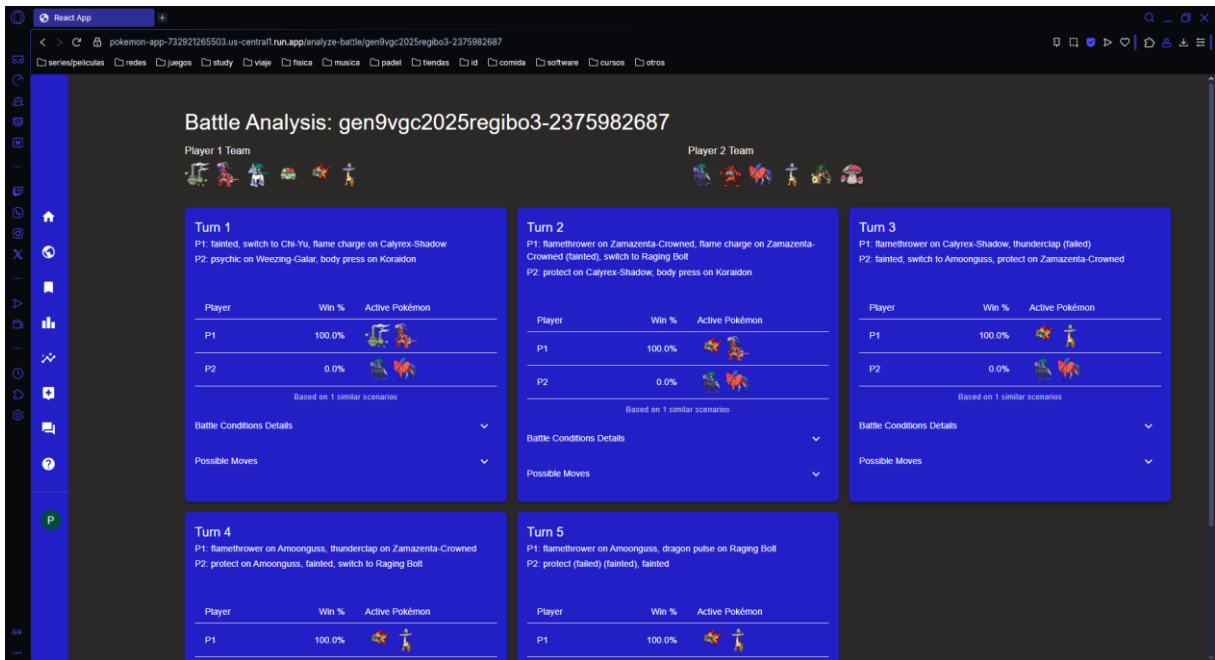


Figura 38. Sección Analyze Battle.

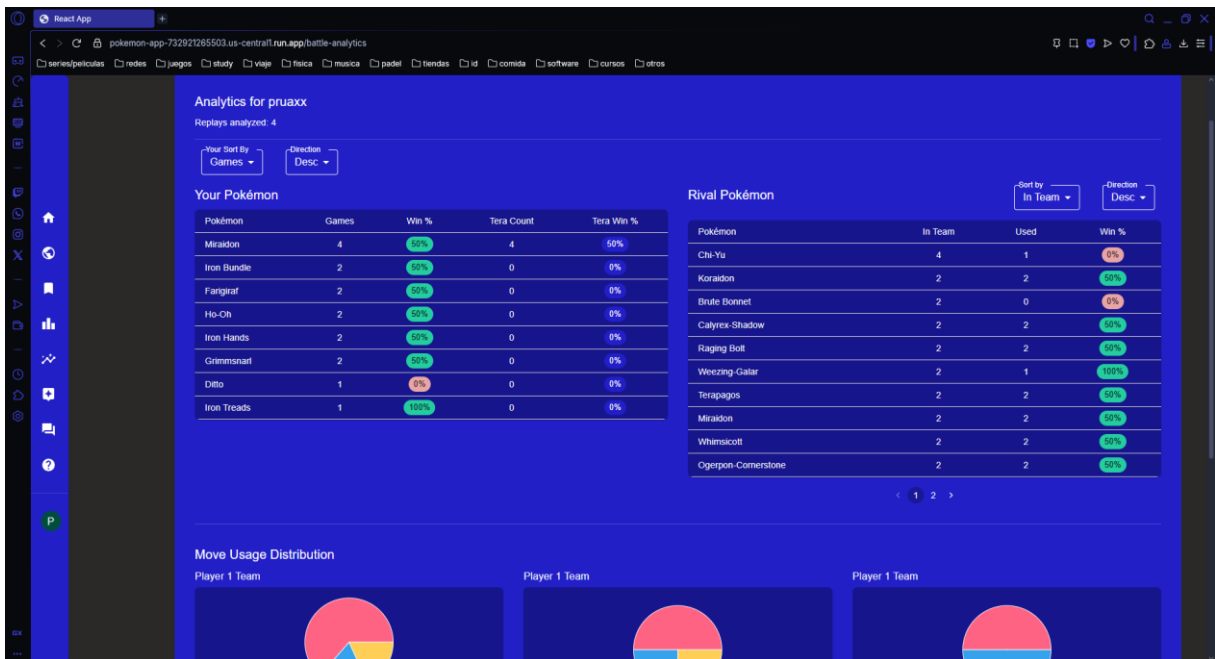


Figura 39. Sección Battle Analytics.

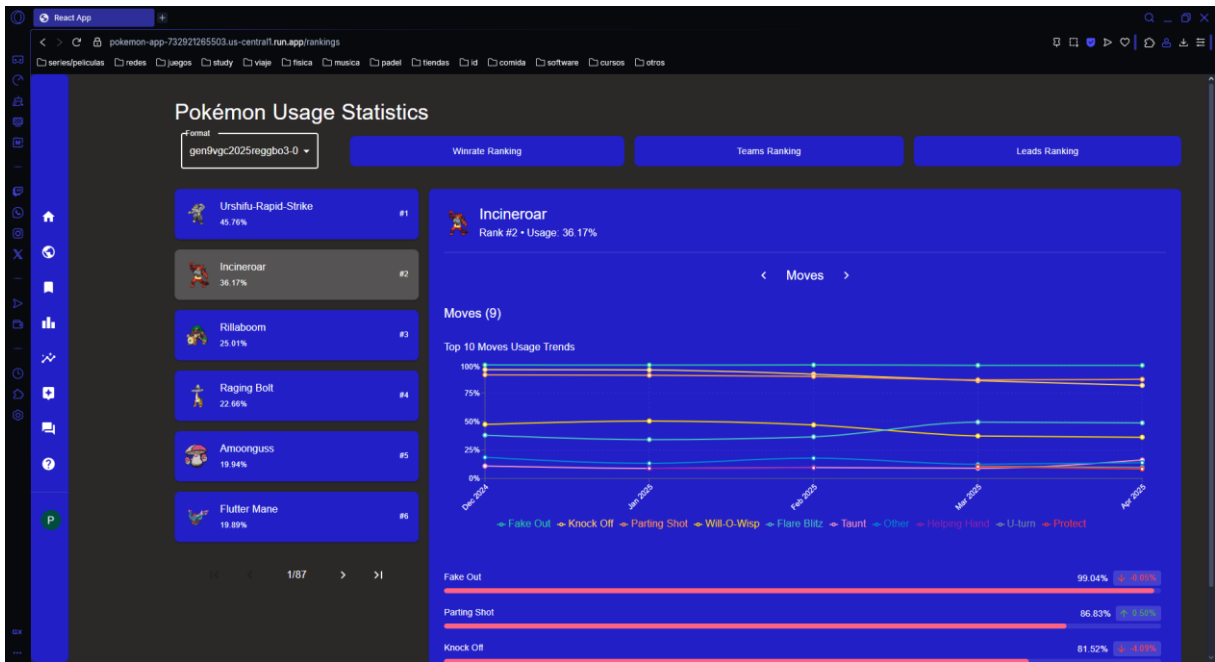


Figura 40. Sección Rankings de uso.

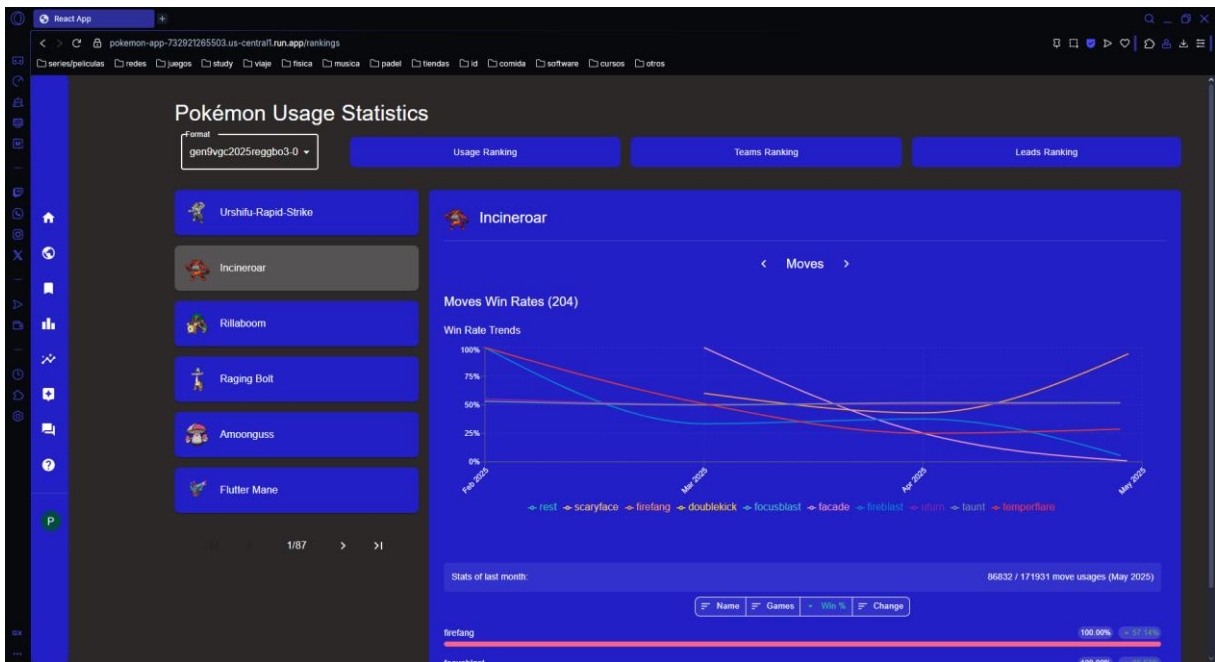


Figura 41. Sección Rankings de victoria.

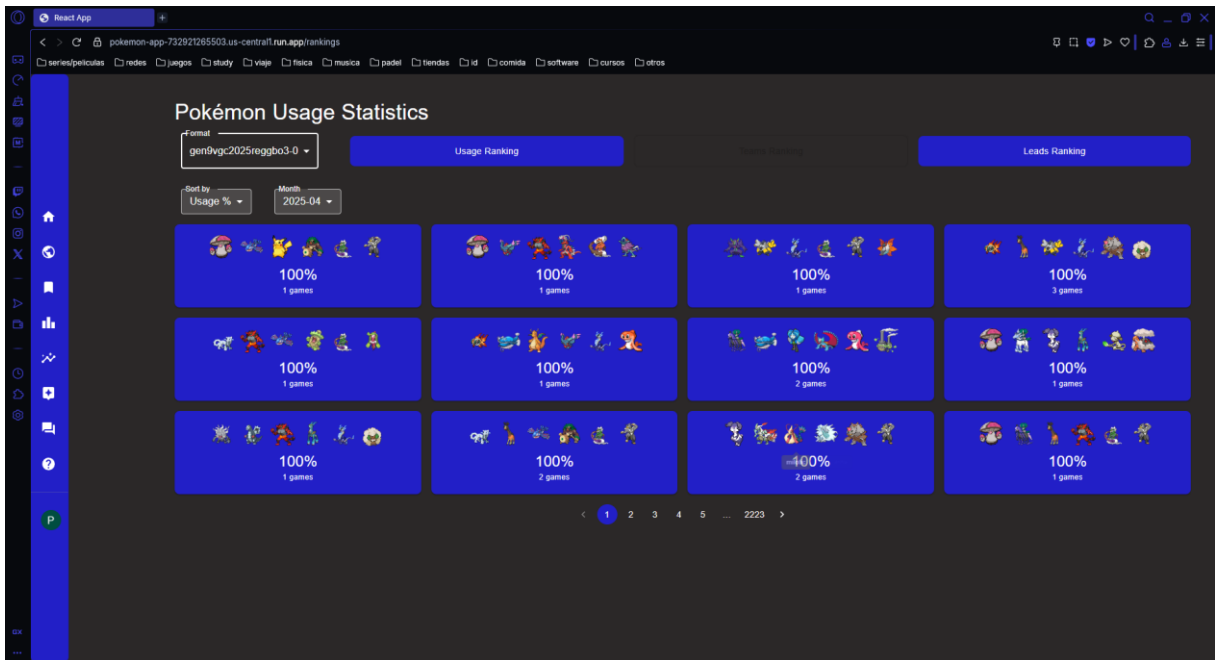


Figura 42. Sección Rankings de equipos.

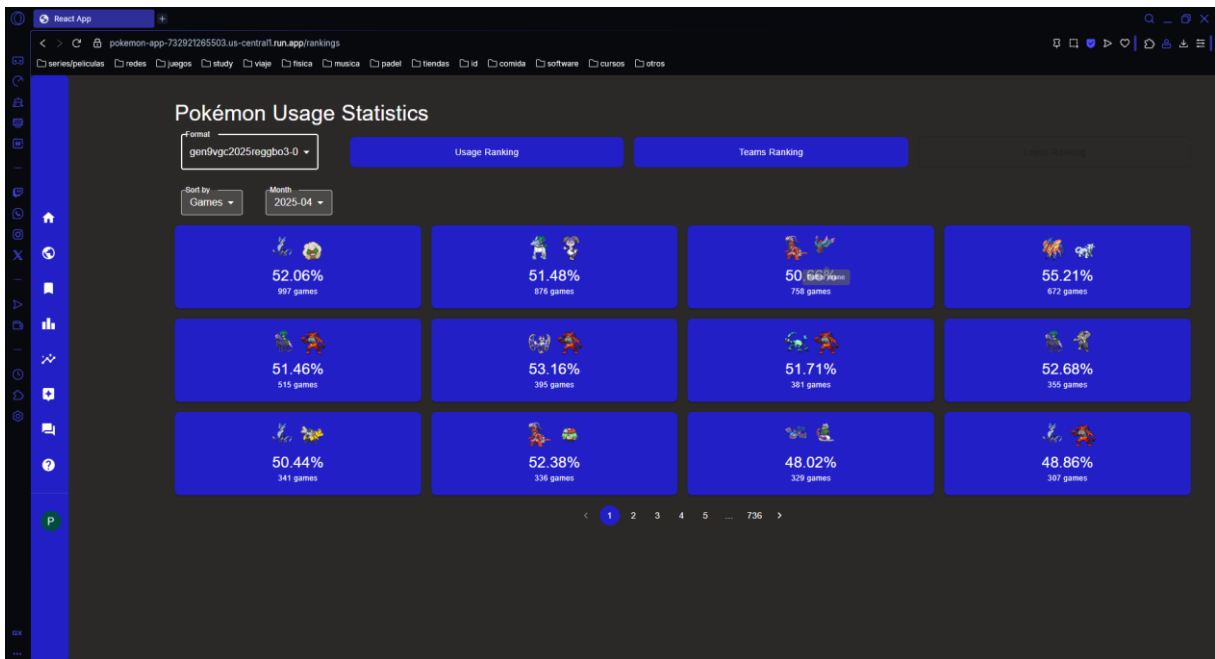


Figura 43. Sección Rankings de leads.

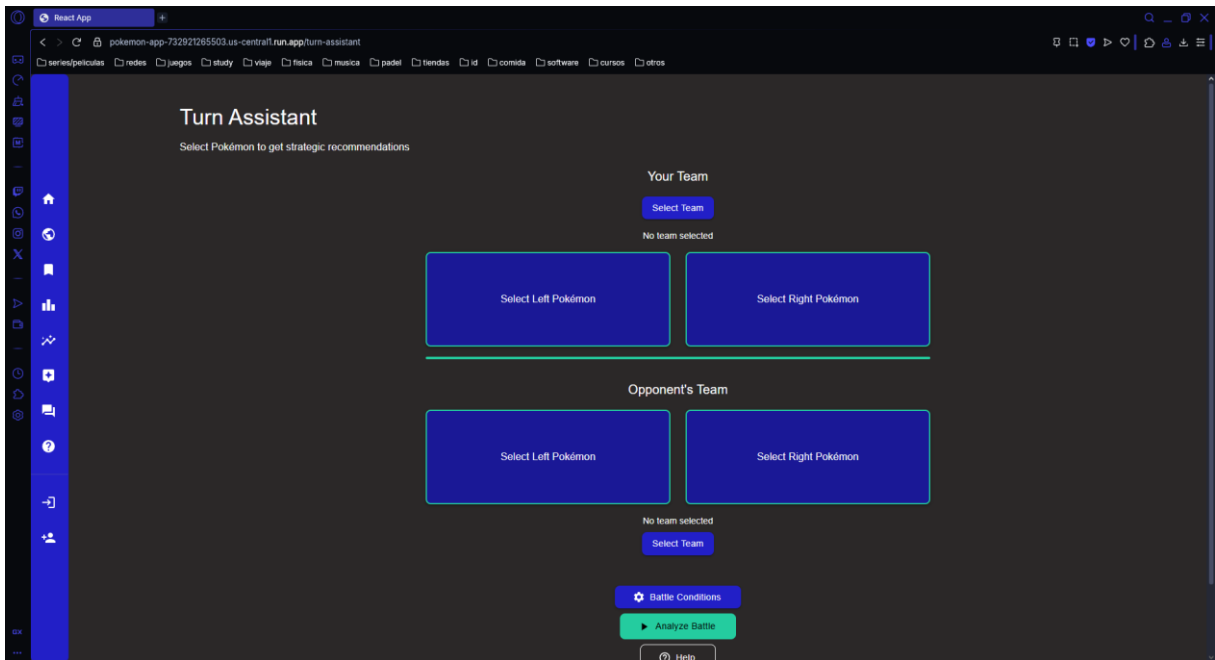


Figura 44. Sección Turn Assistant.

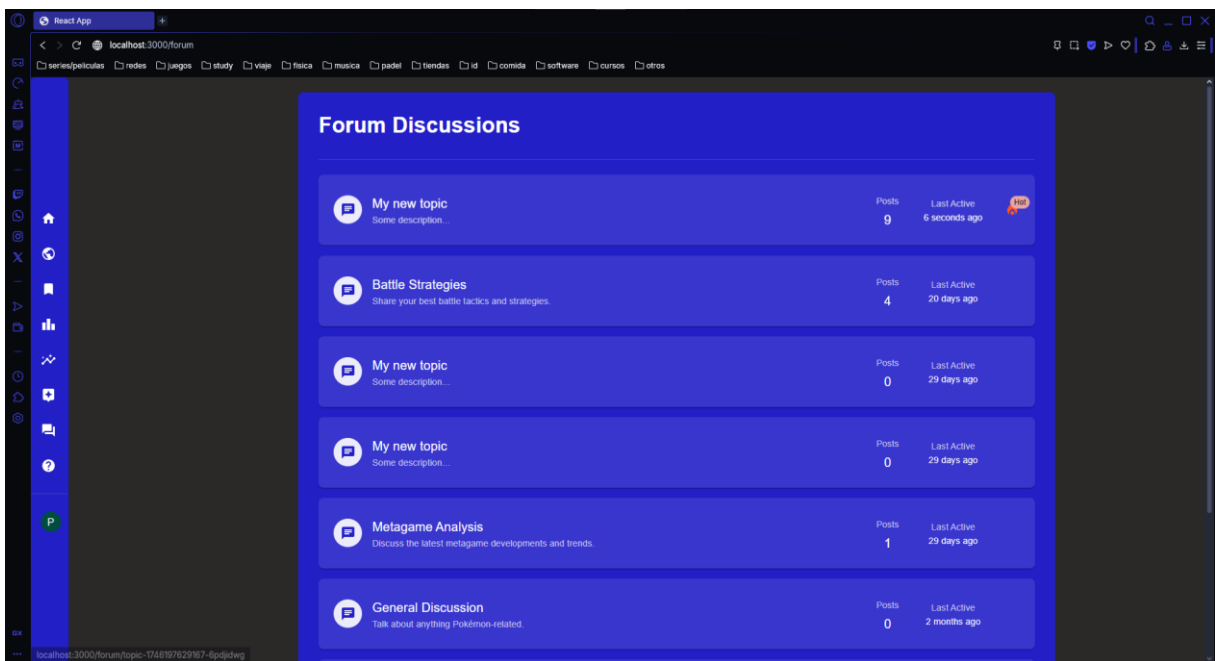


Figura 45. Sección Forum.

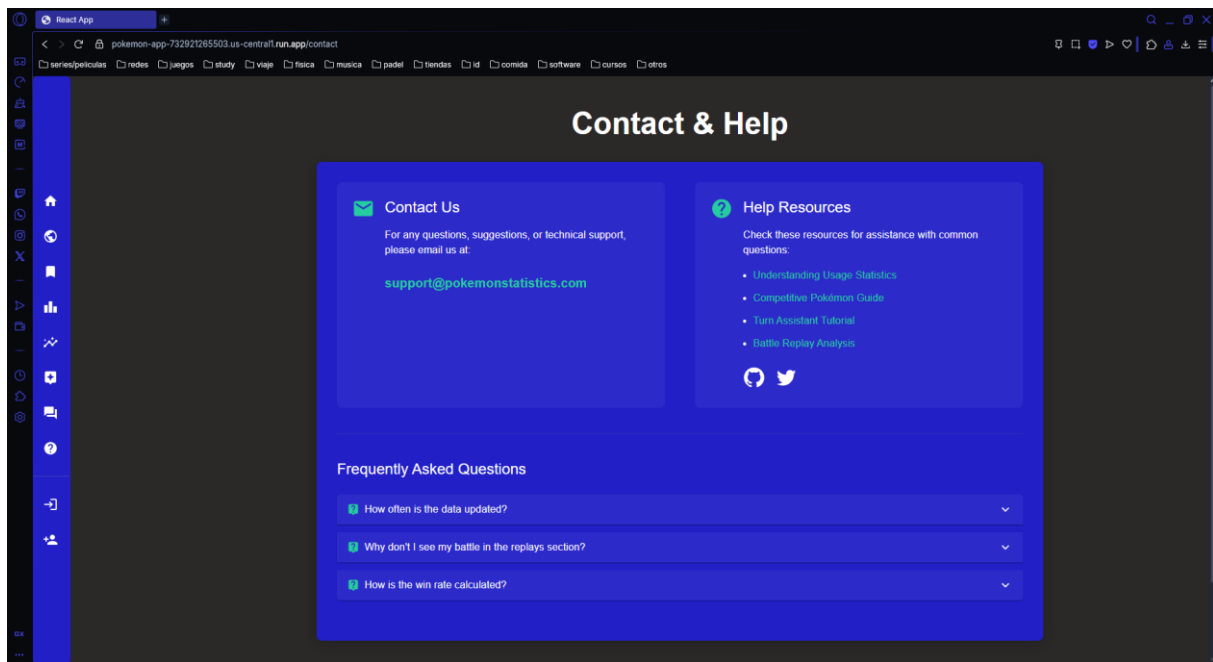


Figura 46. Sección Contact.

V. Pruebas WAVE [22]

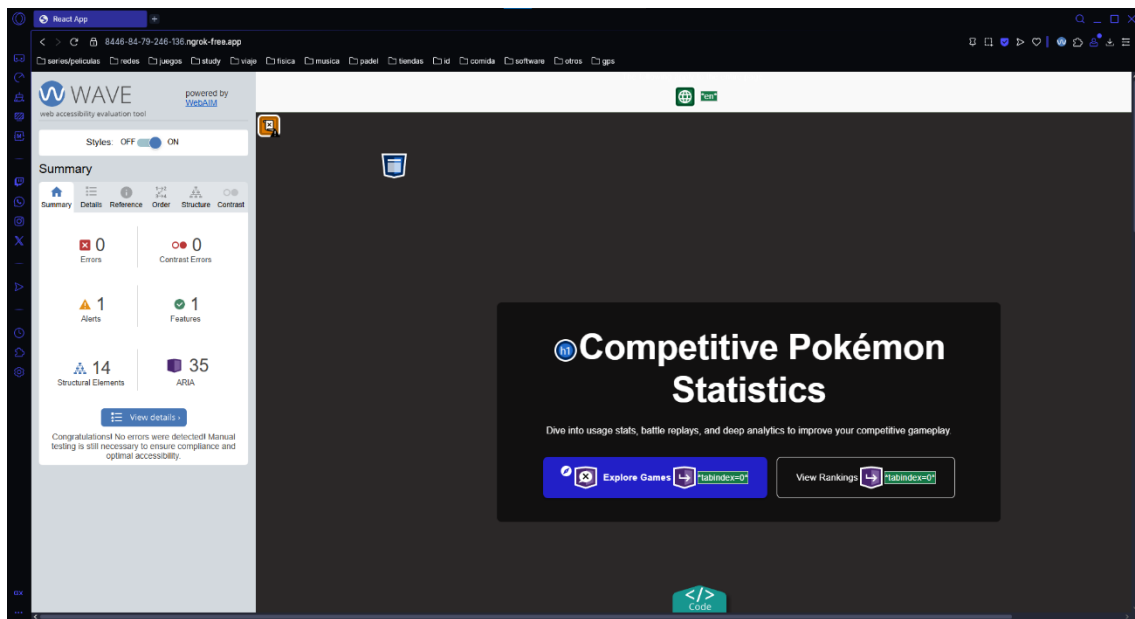


Figura 47. Prueba de WAVE [22] en pestaña home.

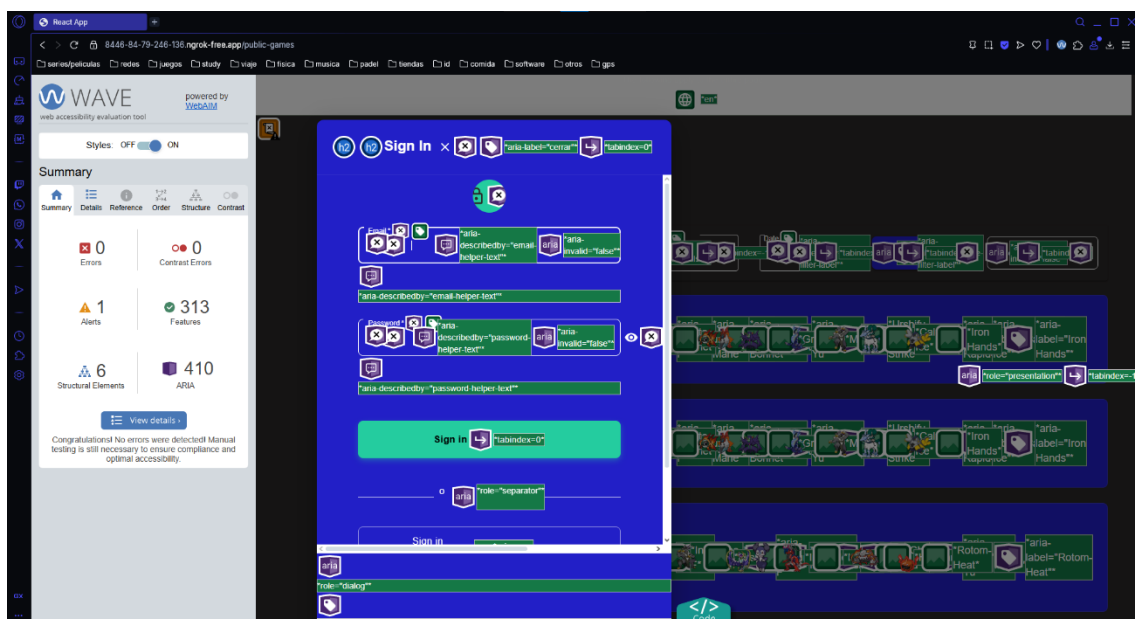


Figura 48. Prueba de WAVE [22] en pestaña sign in.

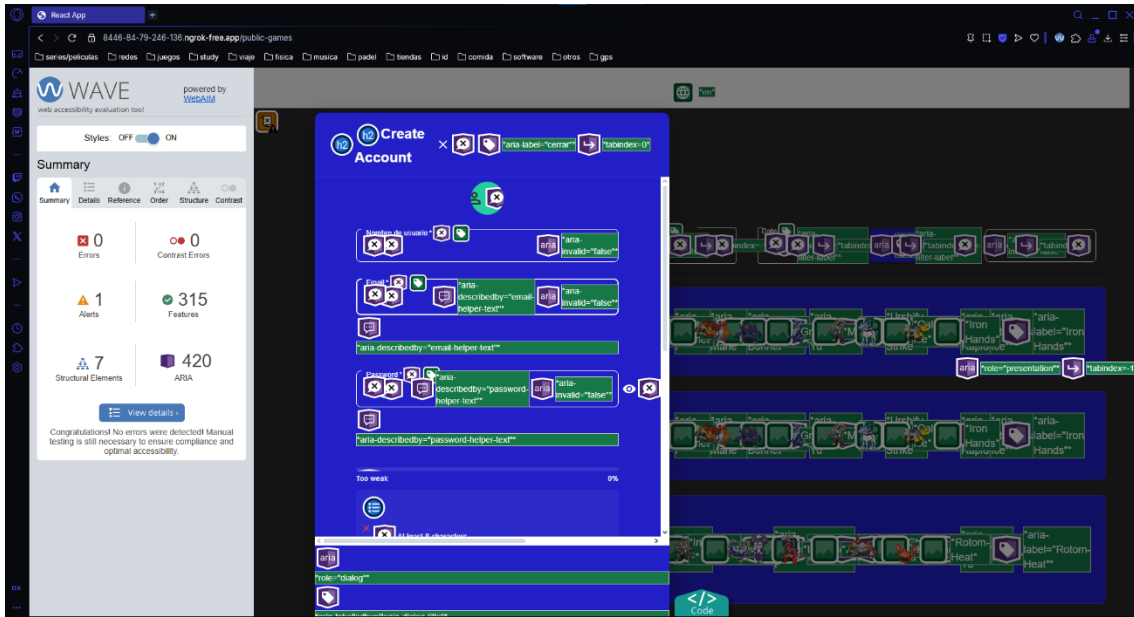


Figura 49. Prueba de WAVE [22] en pestaña sign up.

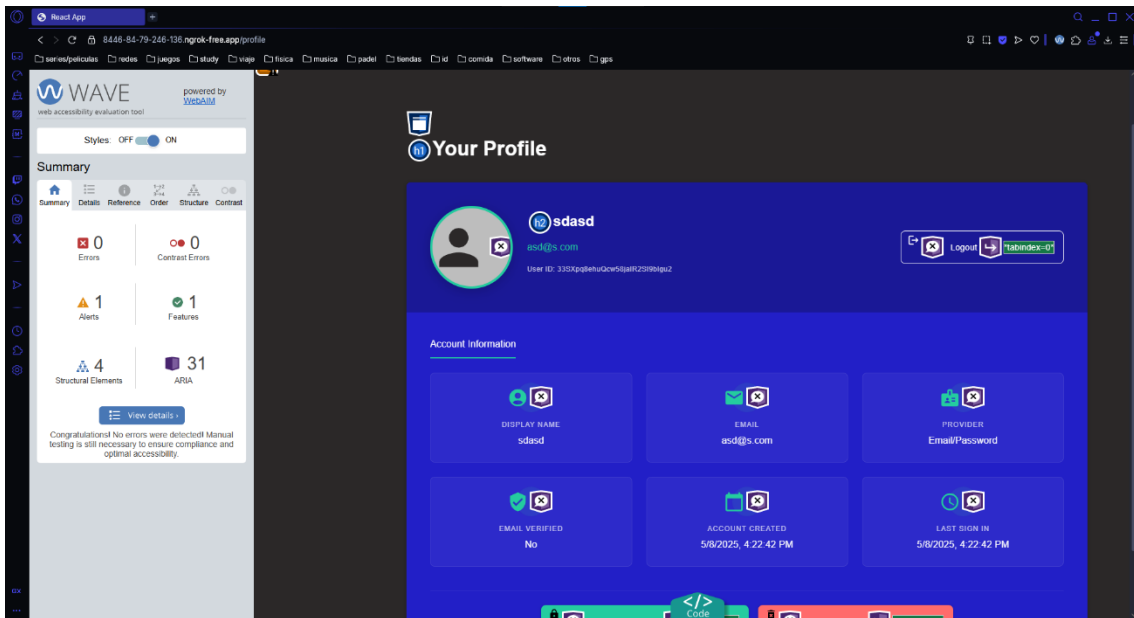


Figura 50. Prueba de WAVE [22] en pestaña profile.

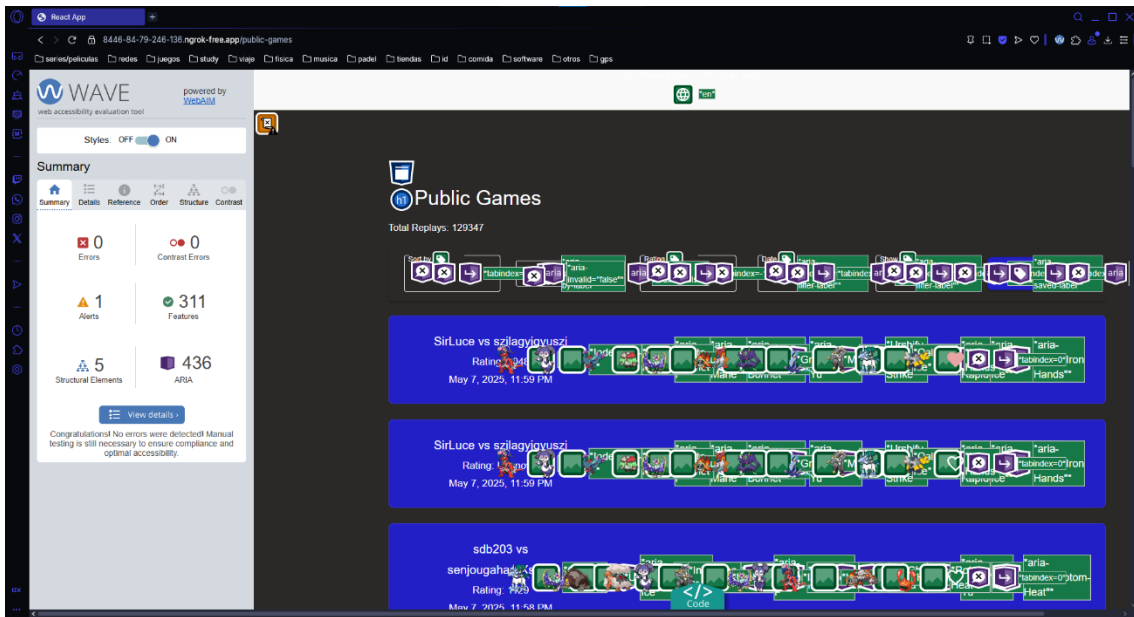


Figura 51. Prueba de WAVE [22] en pestaña public games.

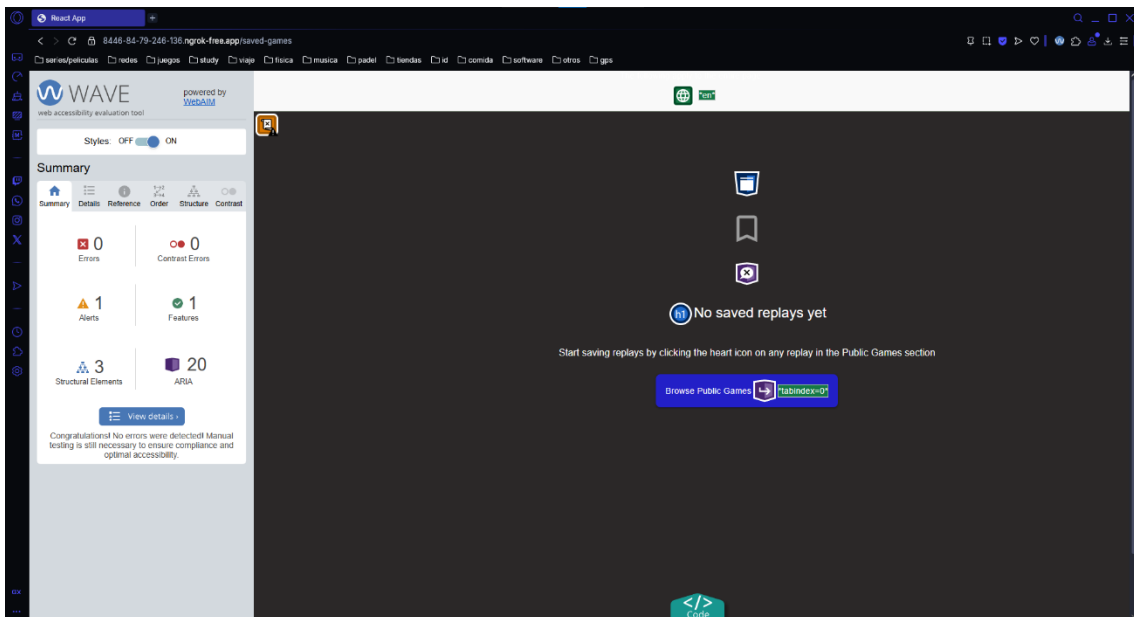


Figura 52. Prueba de WAVE [22] en pestaña saved games cuando no hay replays guardadas.

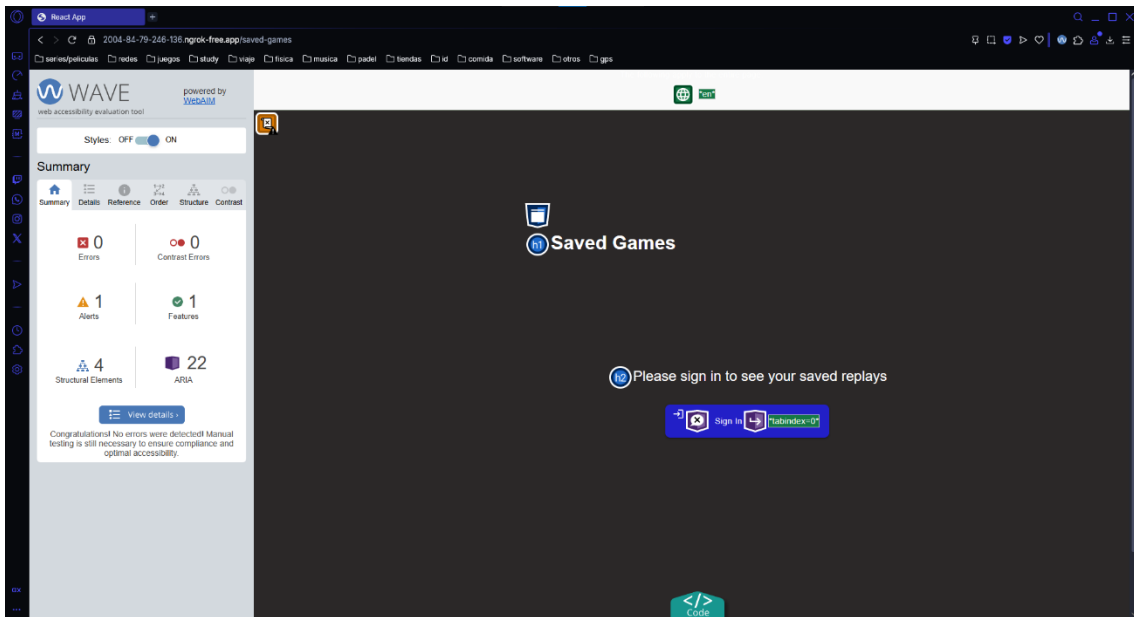


Figura 53. Prueba de WAVE [22] en pestaña saved games cuando no se ha iniciado sesión.

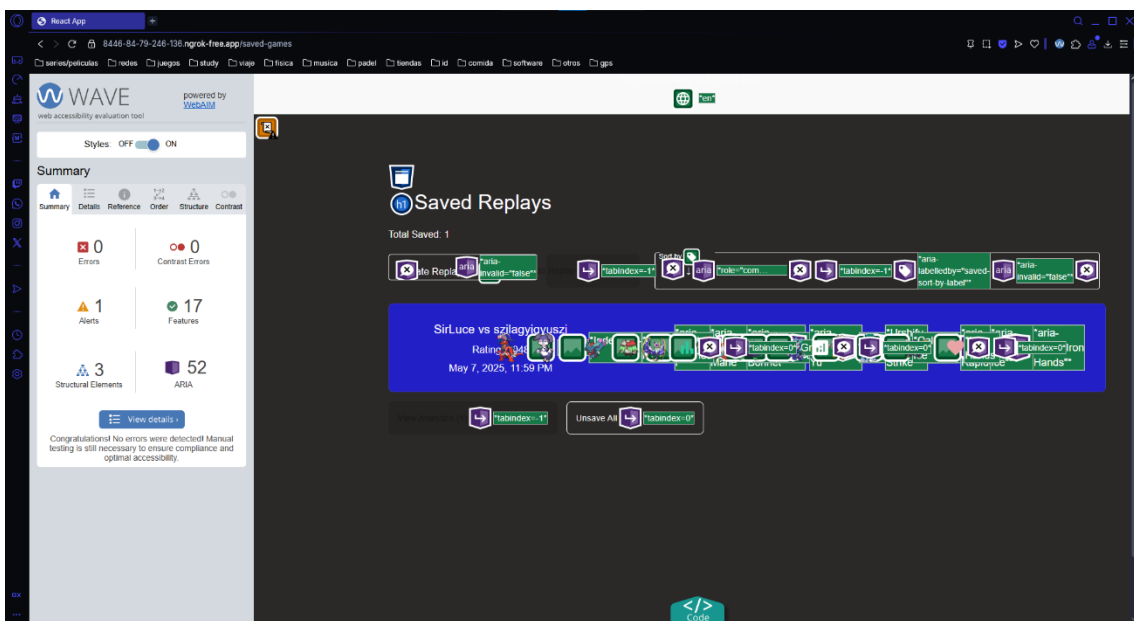


Figura 54. Prueba de WAVE [22] en pestaña saved games cuando se ha iniciado sesión y se ha guardado una replay.

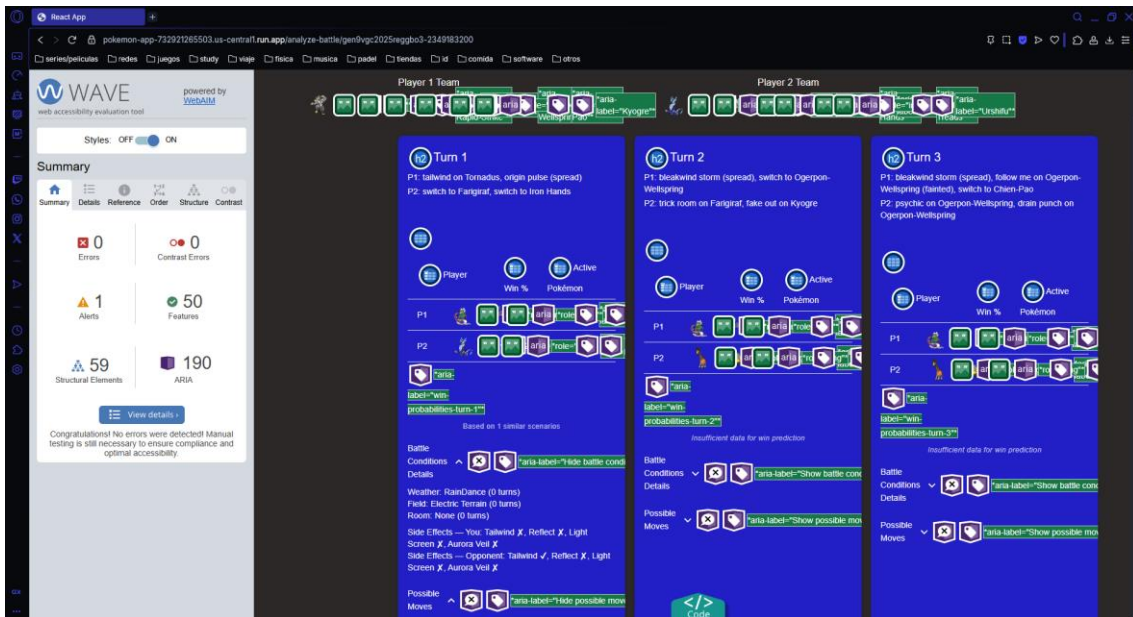


Figura 55. Prueba de WAVE [22] en pestaña analyze battle.

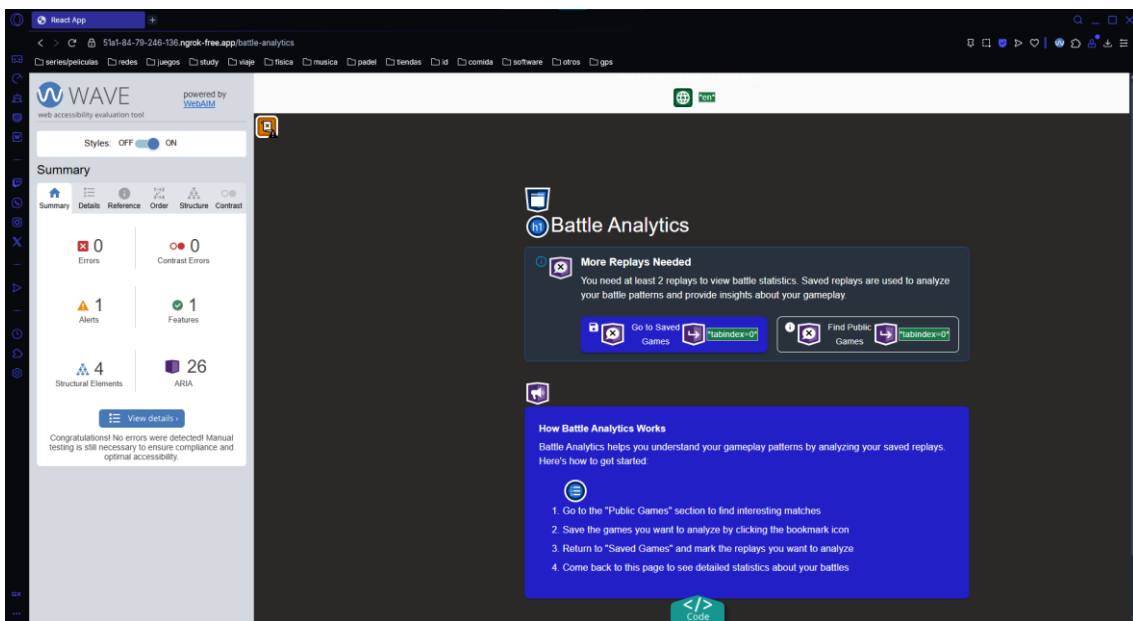


Figura 56. Prueba de WAVE [22] en pestaña battle analytics cuando no se han seleccionado 2 o más partidas.

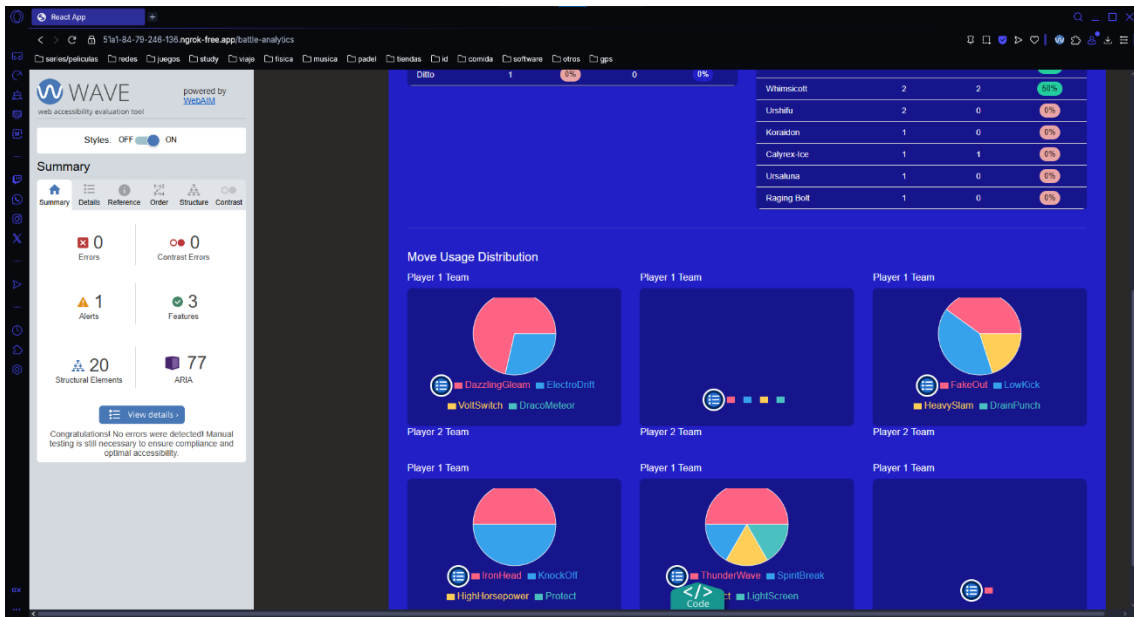


Figura 57. Prueba de WAVE [22] en pestaña battle analytics cuando se han seleccionado 2 o más partidas.

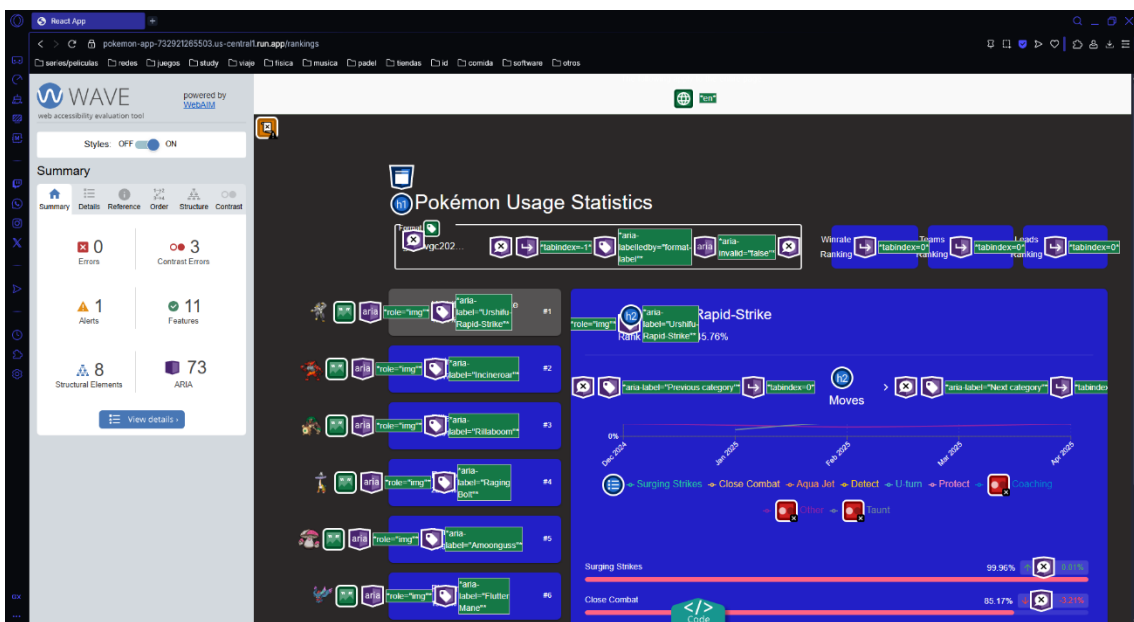


Figura 58. Prueba de WAVE [22] en pestaña usage rankings.

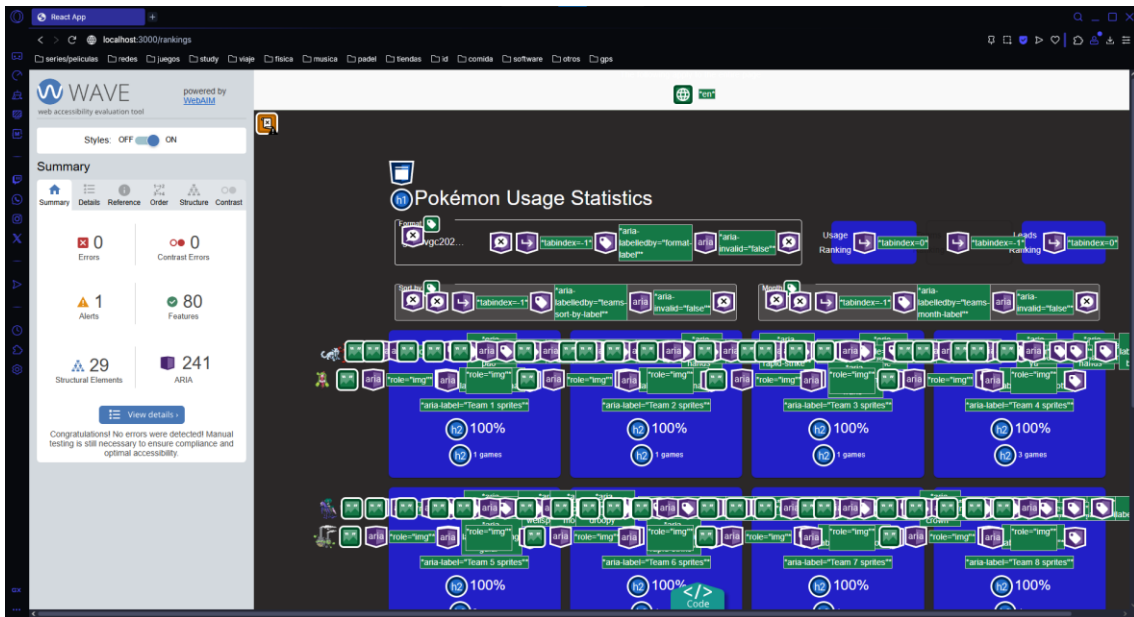


Figura 59. Prueba de WAVE [22] en pestaña teams rankings.

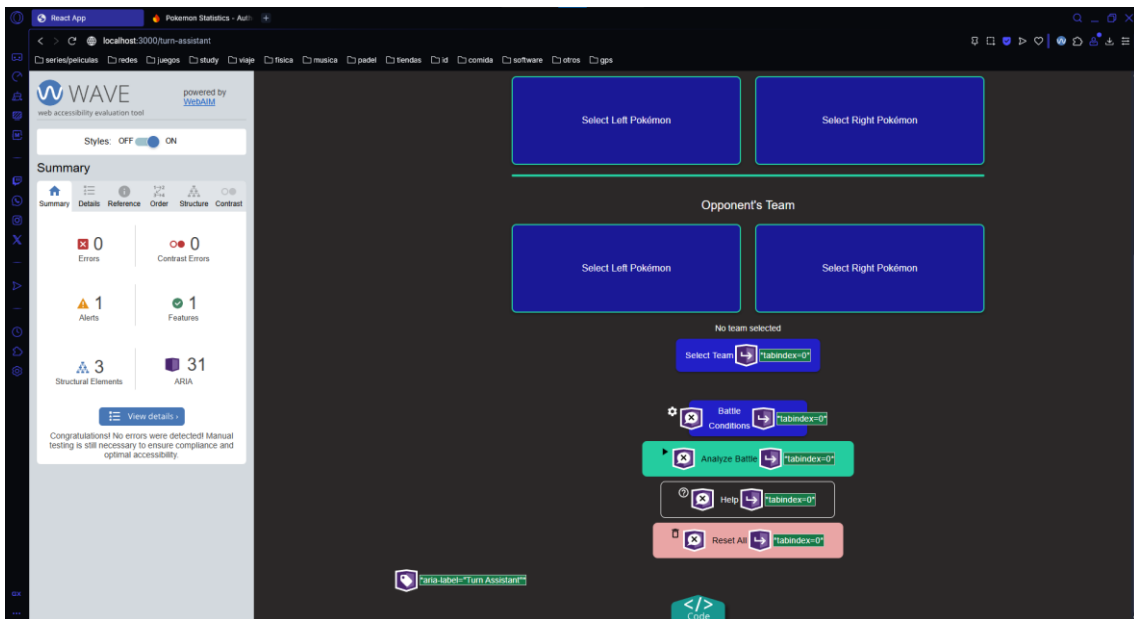


Figura 60. Prueba de WAVE [22] en pestaña turn assistant.



Figura 61. Prueba de WAVE [22] en pestaña turn assistant cuando se elige un Pokémon y sus detalles.

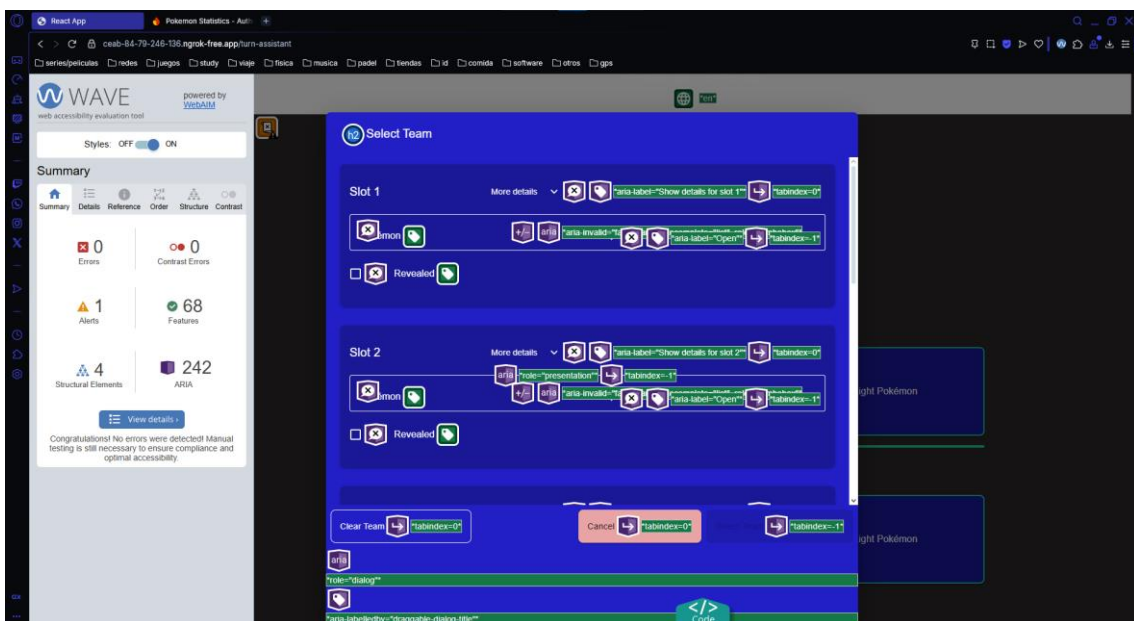


Figura 62. Prueba de WAVE [22] en pestaña turn assistant cuando se seleccionan los Pokémon y sus detalles de un equipo.

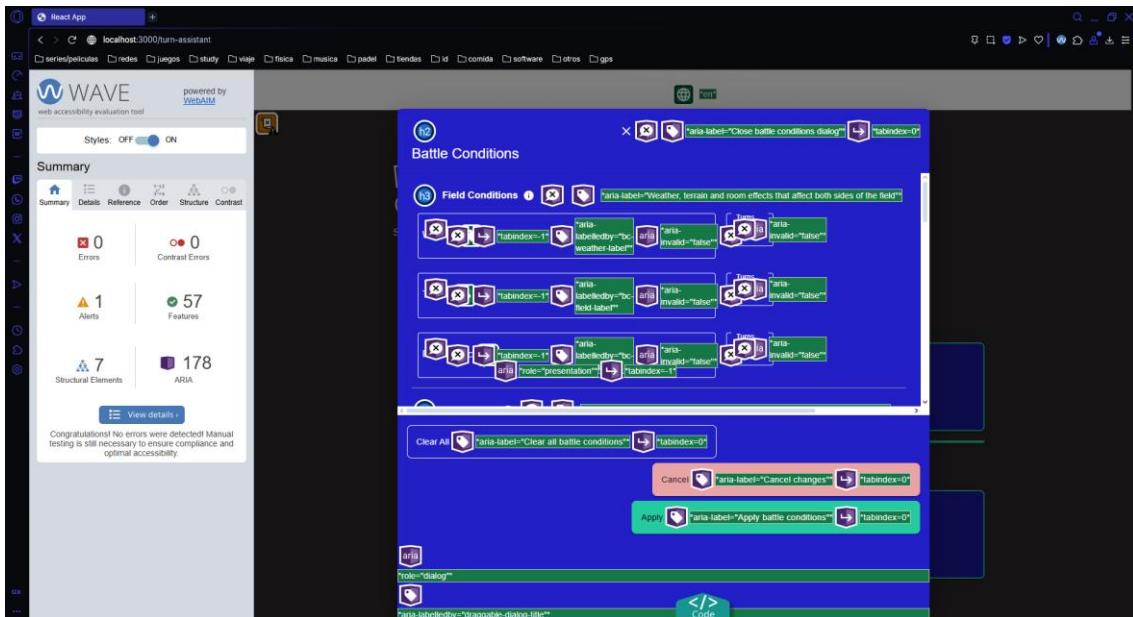


Figura 63. Prueba de WAVE [22] en pestaña turn assistant cuando se seleccionan las condiciones de la partida.

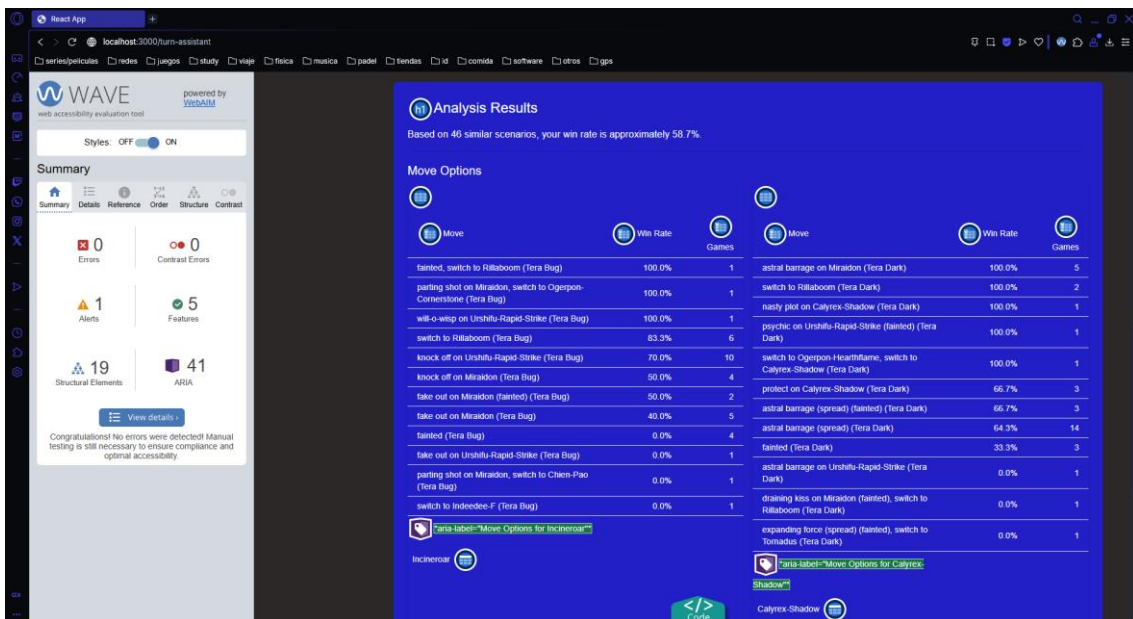


Figura 64. Prueba de WAVE [22] en pestaña turn assistant cuando se ven los resultados del análisis.

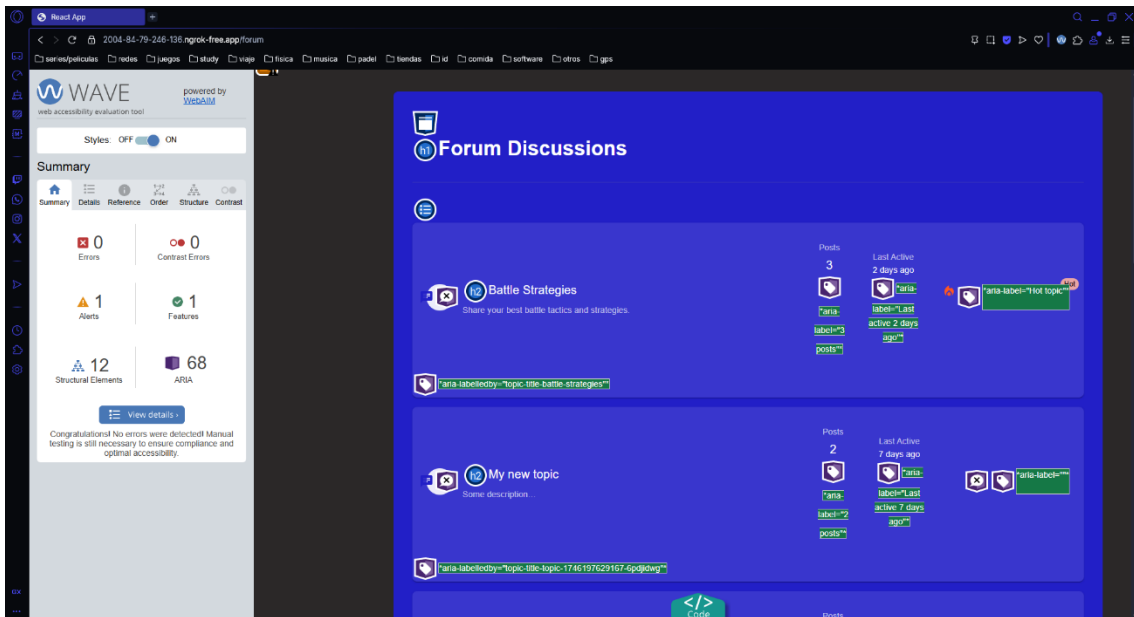


Figura 65. Prueba de WAVE [22] en pestaña forum.

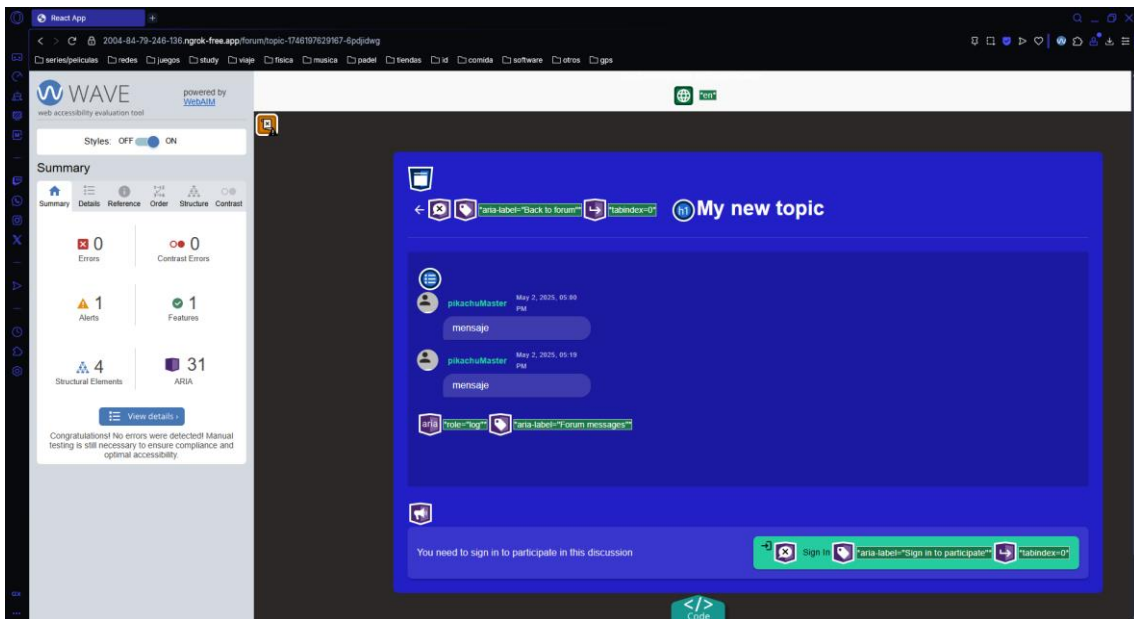


Figura 66. Prueba de WAVE [22] en pestaña forum al entrar en un t3pico.

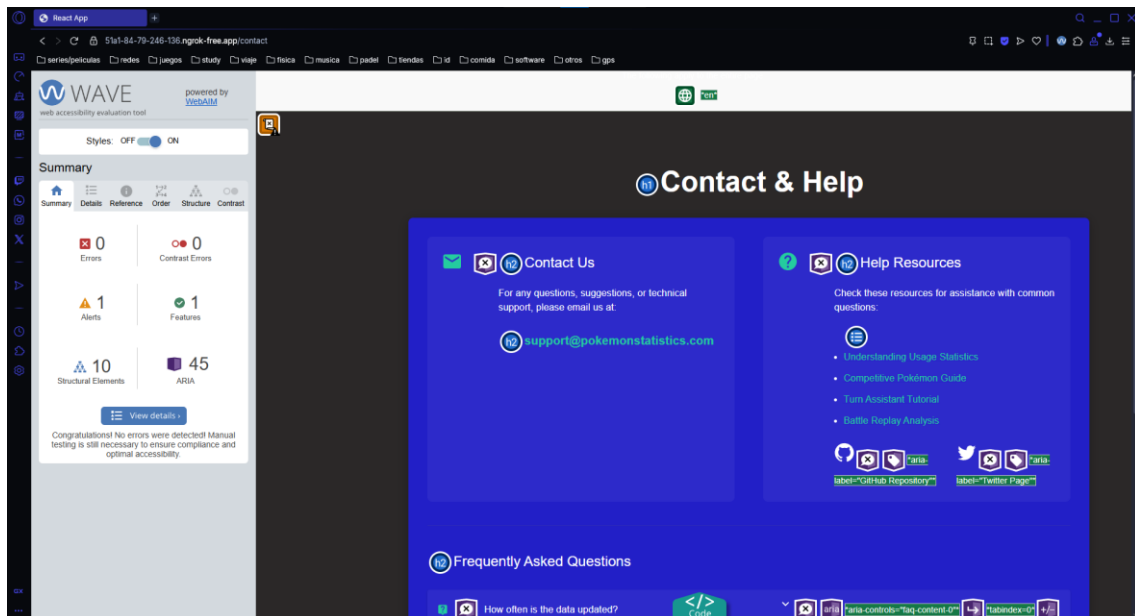


Figura 67. Prueba de WAVE [22] en pestaña de contacto.



UNIVERSIDAD
DE MÁLAGA | **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática

Bulevar Louis Pasteur, 35

Campus de Teatinos

29071 Málaga