
Petapa *ontrack*: desarrollo de una plataforma digital informativa sobre los tiempos de espera aproximados en las atracciones del IRTRA
Petapa

Karen Jimena Hernández Ortega

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Petapa *ontrack*: desarrollo de una plataforma digital informativa sobre los tiempos de espera aproximados en las atracciones del IRTRA Petapa

Trabajo de graduación presentado por
Karen Jimena Hernández Ortega
para optar al grado académico de Licenciada en Ingeniería en Ciencia
de la Computación y Tecnologías de la Información


Guatemala, 2025

Vo.Bo.:

(f) _____
MSc.Kareen Anasilvia Salazar

Tribunal Examinador:

(f) _____
MSc.Kareen Anasilvia Salazar

(f) _____

PhD.Gabriel Antonio Barrientos Rodriguez

Fecha de aprobación: Guatemala, noviembre del 2025.

Lista de figuras	7
Lista de cuadros	8
Resumen	9
Abstract	10
1. Introducción	11
2. Justificación	13
3. Objetivos	14
3.1. Objetivo general	14
3.2. Objetivos específicos	14
4. Marco teórico	15
4.1. Psicología de las filas de espera	15
4.2. Teoría de colas	16
4.2.1. Sistemas de colas	16
4.2.2. Sistema de cola de una sola estación	17
4.2.3. Modelo de cola M M 1	17
4.2.4. Supuestos del proceso de llegadas Poissonianas	18
4.3. Ciencia de datos y aprendizaje automático	18
4.3.1. Ciclo de vida de la ciencia de datos	18
4.3.2. Aprendizaje supervisado	20
4.4. Diseño centrado en el usuario	21
4.4.1. ISO 9241-210, principios del diseño centrado en el usuario	21
4.4.2. Experiencia de usuario (UX)	22
4.4.3. Interfaz de usuario (UI)	22
4.4.4. Principios visuales	23
4.5. Usabilidad	24
4.5.1. ISO 9241-11	24
4.5.2. Evaluación de la usabilidad	24
4.6. Aplicación web	27
4.6.1. Historia y evolución de las aplicaciones web	27
4.6.2. Guías de accesibilidad web y WCAG 2.2	28

4.6.3. Enfoque <i>mobile-first</i>	28
4.6.4. Modelo cliente-servidor	28
4.6.5. Computación en la nube	29
5. Metodología	30
5.1. Enfoque mixto	30
5.2. Fase de investigación y diseño	30
5.2.1. Investigación exploratoria	30
5.2.2. Diseño y prototipado	35
5.3. Procesamiento y predicción de datos	42
5.3.1. Captura de datos	42
5.3.2. Almacenamiento	43
5.3.3. Preprocesado	44
5.3.4. Cálculo del tiempo de espera	44
5.3.5. Análisis de datos	46
5.3.6. Selección de <i>features</i>	46
5.3.7. Desarrollo de modelos	46
5.4. Implementación del sistema	47
5.4.1. <i>Frontend</i>	47
5.4.2. <i>Backend</i>	49
5.5. Evaluación de la plataforma	51
5.5.1. Datos de la muestra	51
6. Resultados	54
6.1. Evaluación de usabilidad y diseño	54
6.1.1. Pruebas de usabilidad	54
6.1.2. Cuestionarios	55
6.1.3. Entrevistas estructuradas	57
6.1.4. Heurísticas de Nielsen	58
6.2. Modelos de <i>machine learning</i>	67
6.3. Sistema <i>backend</i>	68
7. Discusión de resultados	70
8. Conclusiones	73
9. Recomendaciones	74
Bibliografía	78
Anexos	79
A. Fase exploratoria	79
A.1. Encuesta exploratoria	79
A.2. Guion de entrevistas	79
A.3. Consentimiento informado	80
A.4. Pruebas de usabilidad	81
A.5. Entrevista estructurada	81
B. Procesamiento y predicción de datos	82
B.1. Unión y enriquecimiento de datos	82
B.2. Implementación de factor beta	83
B.3. Implementación modelo M M 1	85
B.4. Asistencia total por año	86
B.5. Asistencia por día	88

C. Despliegue e infraestructura	90
C.1. Configuración inicial EC2	90
C.2. Configuración de IP elástica	91
C.3. Conexión segura entre el servidor y PocketBase	91
C.4. Transferencia de modelos y <i>datasets</i> al servidor	92
C.5. Integración con PocketBase	93
C.6. Configuración de <i>cronjobs</i>	94
D. Evaluación de la plataforma	95
D.1. Datos demográficos y consentimiento informado	95
D.2. Evaluación de usabilidad – SUS, <i>UX-Lite</i> , NPS	96
D.3. Entrevista estructurada	97

Lista de figuras

4.1. Sistema de colas	16
4.2. Etapas típicas en el ciclo de vida de los datos	19
4.3. Ciclo de vida de UX, análisis, diseño, implementación y evaluación	22
4.4. Esquema básico de una aplicación web	29
5.1. Hallazgos sobre la satisfacción y percepción de los visitantes	31
5.2. Hallazgos de la gestión y comparación de la satisfacción	31
5.3. Hallazgos sobre los comportamientos digitales de los visitantes	31
5.4. Hallazgos sobre la interacción móvil y consulta de tiempos de espera	32
5.5. <i>User persona 1</i> Mamá planificadora	34
5.6. <i>User persona 2</i> Joven sin responsabilidades	35
5.7. <i>User persona 3</i> Adulto sin hijos	35
5.8. Prototipo baja fidelidad basado en listas	36
5.9. Prototipo baja fidelidad basado en mapa interactivo	37
5.10. Pantallas de la primera versión del prototipo de alta fidelidad	38
5.11. Pantallas de la segunda versión del prototipo de alta fidelidad	39
5.12. Flujo metodológico del proceso de prueba	39
5.13. Pantallas de la tercera versión del prototipo de alta fidelidad	41
5.14. Pantallas de la cuarta versión del prototipo de alta fidelidad	42
5.15. Estructura de los archivos <code>.xls</code> proveídos	43
5.16. Estructura de directorios por año	43
5.17. Resultados demográficos y consentimiento de los participantes	52
6.1. Indicadores de carga	58
6.2. Correspondencia de atracciones	59
6.3. Correspondencia de datos	59
6.4. Correspondencia de ubicaciones	59
6.5. Navegación para regresar	60
6.6. Logos, fuentes e imágenes del IRTRA Petapa	61
6.7. Atracciones que cumplen requisitos	62
6.8. Uso de listas visuales	63
6.9. Distintos tipos de navegación	64
6.10. Diseño con elementos gráficos innecesarios	65
6.11. Producto final sin elementos gráficos innecesarios	65
6.12. Búsqueda sin resultados	66
6.13. Página sin resultados	66
6.14. Guía de uso inicial	67

6.15. Soporte y preguntas frecuentes	67
6.16. Arquitectura final del sistema <i>backend</i>	69
B.1. Asistencia total por año (Parte 1)	86
B.2. Asistencia total por año (Parte 2)	87
B.3. Asistencia total por año (Parte 3)	87
B.4. Asistencia por día (Parte 1)	88
B.5. Asistencia por día (Parte 2)	88
B.6. Asistencia por día (Parte 3)	89
C.1. Configuración inicial de máquina en Amazon EC2	90
C.2. Configuración de IP elástica	91
C.3. Transferencia de modelos entrenados al servidor	92
C.4. Carga de <i>datasets</i>	92
C.5. Script utilizado para la integración con PocketBase	93
C.6. Configuración de <i>cronjobs</i> en el servidor EC2	94

Lista de cuadros

5.1. Ejemplo de estructura del manual de codificación	33
5.2. Hallazgos de la encuesta A/B sobre diseño preferido	37
5.3. Bitácora de tareas utilizada en el proyecto	39
5.4. Tiempos promedios por tarea en la prueba de usabilidad	40
5.5. Hallazgos de la escala SUS aplicada a los participantes	40
5.6. Resultados de la comparación entre la nueva propuesta (1) y la propuesta anterior (2)	41
5.7. Plan de evaluación de la plataforma final	53
6.1. Resumen general de resultados por tarea en la prueba de usabilidad	55
6.2. Resultados de la escala SUS aplicada a los participantes	56
6.3. Resultados de la escala <i>UX-Lite</i> aplicada a los participantes	56
6.4. Respuestas individuales de los participantes en la escala NPS	57
6.5. Resultados generales del <i>Net Promoter Score</i> (NPS)	57
6.6. Resultados de los modelos de <i>machine learning</i> por atracción	68

Los parques de diversiones están diseñados para el entretenimiento de los visitantes. Sin embargo, la espera prolongada en las atracciones es una parte inevitable de la experiencia. En IRTRA Petapa, uno de los parques más visitados de Guatemala, esta situación representa una oportunidad de mejora, ya que, según observaciones y encuestas autorizadas por el IRTRA, los usuarios manifiestan insatisfacción por la falta de información sobre las filas y muestran una alta disposición a utilizar herramientas digitales. Por ello, el presente estudio propone el diseño y desarrollo de una plataforma digital que permite visualizar los tiempos aproximados de espera por atracción.

La solución incluye el diseño y desarrollo de una interfaz que cumple con los estándares de usabilidad propuestos en las heurísticas de Nielsen, un sistema *backend* y un modelo de aprendizaje automático (*machine learning*) entrenado con datos históricos del parque. Todo el desarrollo es guiado mediante un enfoque metodológico mixto que combina análisis cualitativo y cuantitativo, lo que permite flexibilidad y mejoras continuas durante el proceso.

Los resultados muestran un desempeño predictivo satisfactorio y altos niveles de usabilidad según las evaluaciones con los instrumentos *System Usability Scale* (SUS), *Net Promoter Score* (NPS) y *UX-Lite*.

Amusement parks are designed for the entertainment of visitors. However, long waiting times at attractions are an inevitable part of the experience. At IRTRA Petapa, one of the most visited parks in Guatemala, this situation represents an opportunity for improvement, since, according to observations and surveys authorized by IRTRA, users express dissatisfaction with the lack of information about lines and show a high willingness to use digital tools. Therefore, this study proposes the design and development of a digital platform that allows users to visualize the approximate waiting times for each attraction.

The solution includes the design and development of an interface that meets the usability standards proposed in Nielsen's heuristics, a *backend* system, and a machine learning model trained with the park's historical data. The entire development is guided by a mixed methodological approach that combines qualitative and quantitative analysis, allowing for flexibility and continuous improvement throughout the process.

The results show satisfactory predictive performance and high levels of usability according to evaluations using the System Usability Scale (SUS), Net Promoter Score (NPS), and UX-Lite instruments.

CAPÍTULO 1

Introducción

En los parques de atracciones, las filas de espera son inevitables en la experiencia del visitante. Sin embargo, la manera en que los usuarios perciben ese tiempo puede influir significativamente en su satisfacción general [1].

En 1985, David Maister escribió *The Psychology of Waiting Lines*, en donde propuso la “primera ley del servicio”, definida como:

$$\text{Satisfacción} = \text{Percepción} - \text{Expectativa}$$

Esto significa que, si el cliente espera cierto nivel de servicio y percibe que el servicio recibido fue mejor, estará satisfecho. Si percibe el mismo nivel de siempre, pero esperaba uno mejor, se sentirá insatisfecho. Aplicando esta ley en un parque de atracciones, la satisfacción del visitante depende en gran medida de la precisión del tiempo de espera estimado, ya que una predicción inexacta o nula del tiempo real puede disminuir la calidad percibida del servicio afectando negativamente la experiencia del usuario [1, 2].

Los parques de atracciones modernos reconocen que la satisfacción de los usuarios no depende únicamente del servicio brindado, sino también de las emociones generadas durante la experiencia en la visita. Según Bigné et al., “la satisfacción del consumidor puede definirse como un estado cognitivo-afectivo resultante de evaluaciones cognitivas, así como de las emociones que estas evaluaciones evocan”[3, p. 835].

Actualmente, parques como Walt Disney World han implementado soluciones tecnológicas que mejoran la experiencia de los usuarios, como la aplicación *My Disney Experience* que brinda información en tiempo real sobre los tiempos de espera, mejorando la experiencia del visitante[1]. En contraste, el Instituto de Recreación de los Trabajadores de la Empresa Privada de Guatemala (IRTRA), específicamente el parque IRTRA Petapa, uno de los parques más famosos del país, no cuenta con una herramienta similar. Esto representa una oportunidad para el desarrollo de una plataforma digital que proporcione esta información a los visitantes.

El presente trabajo constituye la culminación de un proceso de investigación cuyo objetivo es diseñar y desarrollar una solución digital que permita a los visitantes del IRTRA Petapa conocer los tiempos aproximados de espera en las atracciones. Esta plataforma busca reducir la incertidumbre y la frustración asociadas a las filas, promoviendo una mejor planificación del recorrido dentro del parque. Además, al cumplir con los estándares de usabilidad propuestos en las heurísticas de Nielsen y alinearse con la identidad visual del IRTRA Petapa, la plataforma se consolida como una

herramienta de valor para el parque. A futuro, esta solución podría evolucionar para incorporar nuevas funcionalidades que contribuyan a optimizar aún más la experiencia del visitante.

Diversos estudios y observaciones en campo evidencian que las esperas prolongadas en filas dentro de los parques de atracciones, cuando no se proporciona información adecuada, reducen significativamente la satisfacción del visitante [1, 2], lo que hace necesaria la intervención con herramientas digitales [4]. Estas tecnologías permiten actualizar datos de forma automática, son personalizables y pueden ser accesibles desde dispositivos móviles, lo que representa una ventaja ante las soluciones tradicionales, como lo son la señalización física o el personal informativo [4]. De hecho, existe evidencia de que más del 80 % de los asistentes a parques de atracciones utilizan dispositivos móviles durante su visita para acceder a información en línea [4].

Según Toro, las tecnologías de información (TI) correctamente aplicadas no solo aumentan la eficiencia de una empresa, sino que también permiten a éstas mejorar la forma en que crean y entregan valor a sus usuarios [5]. Estas soluciones no deben limitarse a replicar procesos tradicionales, sino que deben transformar la experiencia y la interacción del usuario con los servicios ofrecidos [6].

Si bien la presente investigación se enfoca en la experiencia del usuario durante los tiempos de espera en parques de atracciones y existen estudios internacionales que abordan el uso de plataformas digitales en estos entornos [1, 4], existe poca evidencia en la literatura sobre el diseño y desarrollo de herramientas digitales que informen y orienten a los visitantes en espacios recreativos en Guatemala.

En consecuencia, resulta indispensable el estudio del uso de herramientas digitales que permitan informar a los usuarios en contextos recreativos, ya que no solo responde a una necesidad identificada, sino que también como una vía para mejorar la experiencia del visitante buscando soluciones innovadoras y adaptadas a sus expectativas [1, 2, 4].

Los resultados que presenta este trabajo son relevantes no solo para el campo de la interacción humano-computador (HCI, por sus siglas en inglés), sino también para áreas como la ingeniería de software, el desarrollo web y la ciencia de datos, ya que contribuyen a la literatura existente y sientan las bases para el diseño y desarrollo de futuras herramientas digitales orientadas a mejorar la experiencia del usuario en entornos recreativos, tanto en Guatemala como en contextos internacionales.

3.1 Objetivo general

Desarrollar una plataforma digital para el IRTRA Petapa que brinde información sobre los tiempos de espera aproximados en sus atracciones.

3.2 Objetivos específicos

- Diseñar una interfaz de usuario basada en listas visuales alineada con los principios de usabilidad propuestos en las heurísticas de Nielsen y con los estándares de marca del IRTRA Petapa.
- Implementar un modelo de *machine learning* que estime los tiempos de espera de los visitantes en las atracciones, basado en datos históricos.
- Implementar un sistema *backend* que permita conectar el modelo de predicción con la interfaz de usuario.

4.1 Psicología de las filas de espera

Estudios previos han logrado identificar una serie de principios que permiten comprender cómo los individuos interpretan y experimentan las esperas, así como estrategias efectivas para gestionar dichas percepciones con el fin de mejorar la satisfacción general del usuario [7, 8]. Estos principios y estrategias incluyen:

- **El tiempo desocupado se percibe como más largo que el ocupado:** involucrar al usuario mediante actividades interactivas e informativas mientras espera.
- **Las esperas previas al inicio del servicio se perciben como más largas que aquellas que ocurren durante el proceso:** hacer que el usuario se sienta dentro del servicio para percibir un menor tiempo de espera.
- **La ansiedad incrementa la percepción del tiempo de espera:** la falta de control y de información clara sobre el proceso puede generar incertidumbre, aumentando la duración subjetiva del tiempo de espera.
- **Las esperas inciertas o indefinidas se perciben como más largas que aquellas que tienen un límite conocido:** Proveer tiempos estimados claros y realistas.
- **Las esperas no explicadas generan mayor frustración que aquellas acompañadas de una justificación:** la comunicación efectiva y la explicación de los motivos de la demora permiten gestionar mejor las expectativas.
- **Las situaciones de inequidad percibida durante la espera incrementan la insatisfacción:** es fundamental establecer mecanismos visibles de orden y prioridad que transmitan equidad y respeto por el tiempo de cada usuario.
- **La disposición a esperar se relaciona con el valor percibido del servicio:** los usuarios están dispuestos a tolerar mayores tiempos de espera si consideran que el servicio o producto lo justifica.

- **La espera en solitario tiende a percibirse como más prolongada que la espera en compañía:** fomentar el sentido de comunidad o interacción social en el entorno de espera puede generar una percepción menor del tiempo de espera.

En este sentido, investigaciones recientes han determinado que la experiencia de espera no solo está definida por su duración, sino por cómo se recuerda [1, 3].

Existe una lista exhaustiva de consideraciones psicológicas implicadas en la aceptación por parte de los clientes del tiempo de espera. No obstante, mediante la información proporcionada por [7, 8], se puede obtener una comprensión de la complejidad psicológica de las filas de espera y un acercamiento de cómo se puede gestionar la experiencia psicológica de la espera.

4.2 Teoría de colas

La teoría de colas tiene como objetivo principal modelar sistemas de espera en los que existe un conjunto de servidores con cierta capacidad, a los que llegan clientes que demandan un servicio, esperan si están ocupados y abandonan el sistema tras recibirlo [9].

4.2.1 Sistemas de colas

Los sistemas de colas típicos están compuestos por una fuente de entrada, una cola y un mecanismo de servicio:

- **Fuente de entrada/Llegadas:** genera los clientes que llegan al sistema. Puede ser finita o infinita y los clientes llegan siguiendo un patrón estadístico.
- **Cola:** los clientes esperan turno hasta ser atendidos.
- **Mecanismo de servicio:** atiende a los clientes en orden según la disciplina de la cola.
- **Salida:** los clientes ya atendidos dejan el sistema.

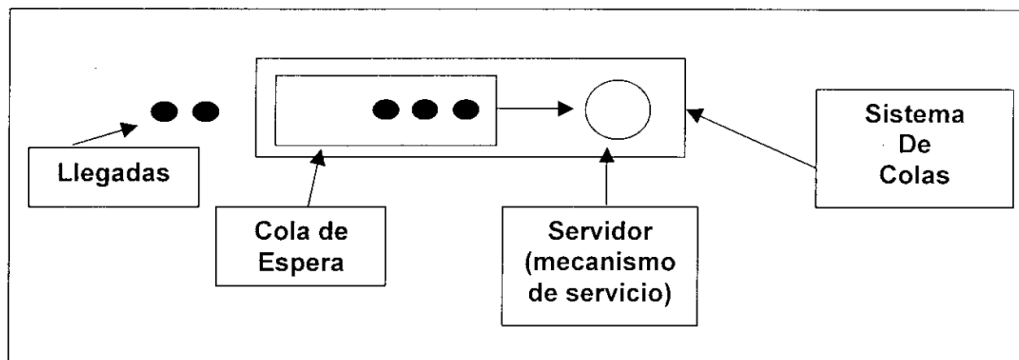


Figura 4.1: Sistema de colas [10]

Para describir un sistema de colas se usa la notación descrita por Kendall en [11]:

$$A|S|c|K|F|d$$

- **A**: Distribución de arribos
- **S**: Distribución del tiempo de servicio
- **c**: número de servidores
- **K**: Capacidad máxima del sistema.
- **F**: = Número de individuos en la fuente.
- **d**: disciplina de servicio (FIFO, LIFO, SIRO, etc.)

4.2.2 Sistema de cola de una sola estación

En el modelo de cola de una sola estación, los clientes llegan a un único punto de servicio y, si éste está ocupado, forman una fila de espera hasta que el servidor quede libre; tras recibir el servicio, abandonan el sistema [9].

Este modelo consta de dos supuestos clave:

- **Tiempos entre arribos iid**: los intervalos entre las llegadas son variables aleatorias independientes e idénticamente distribuidas.
- **Tiempos de servicio iid y *work-conserving***: los tiempos de servicio siguen una misma distribución, los servidores son estocásticamente idénticos y no permanecen ociosos si hay clientes esperando.

Dentro de los modelos de una sola estación, el **M|M|1** ha sido tradicionalmente el más abordado en investigaciones, dado que representa el caso fundamental y más simple de la teoría de colas [12].

4.2.3 Modelo de cola M|M|1

En el modelo **M|M|1**, las llegadas de clientes siguen un proceso de Poisson con tasa λ , es decir, los intervalos entre llegadas son variables aleatorias con distribución exponencial. De igual forma, el tiempo de servicio del único servidor a cada cliente es exponencial con parámetro μ . Cuando el servidor está ocupado, los clientes forman una única fila FIFO y esperan hasta ser atendidos[13].

Condición de estabilidad

Para que el sistema no crezca indefinidamente, la tasa de llegadas debe ser menor que la tasa de servicio:

$$\rho = \frac{\lambda}{\mu} < 1$$

donde ρ es el *factor de utilización*, es decir, la fracción de tiempo en que el servidor está ocupado.

Medidas de desempeño

Las métricas principales del modelo M|M|1 son:

- Número esperado de clientes en el sistema:

$$L = \frac{\rho}{1 - \rho}$$

- Número esperado de clientes en la cola:

$$L_q = \frac{\rho^2}{1 - \rho} = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

- Tiempo promedio en el sistema:

$$W = \frac{L}{\lambda} = \frac{1}{\mu - \lambda}$$

- Tiempo promedio en la cola:

$$W_q = \frac{L_q}{\lambda} = \frac{\lambda}{\mu(\mu - \lambda)}$$

[10]

4.2.4 Supuestos del proceso de llegadas Poissonianas

Según Green [14], para que sea razonable modelar las llegadas mediante un proceso de Poisson, se deben cumplir estas tres propiedades fundamentales:

1. Los clientes llegan uno a la vez. Aunque puedan ocurrir excepciones, esto no suele alterar significativamente la validez del modelo.
2. La probabilidad de que llegue un cliente en un instante no depende de cuándo hayan llegado otros clientes.
3. La probabilidad de que un cliente llegue en un instante dado es independiente de dicho instante.

Cuando estas condiciones son razonablemente válidas, es posible emplear el proceso de Poisson estándar como un buen modelo si lo restringimos a períodos breves durante los cuales la tasa de arribos λ se mantiene aproximadamente constante.

4.3 Ciencia de datos y aprendizaje automático

La ciencia de datos se refiere a macroprocesos que contienen una serie de subprocesos complejos, con el fin de convertir los datos en patrones útiles para la toma de decisiones. El aprendizaje automático es una rama de la inteligencia artificial que se encarga de generar algoritmos que tienen la capacidad de aprender y no de tener que programarlos de manera explícita. Como explica Sandoval [15], el aprendizaje automático se divide en dos tipos fundamentales: supervisado y no supervisado[16].

4.3.1 Ciclo de vida de la ciencia de datos

Los avances en tecnología e informática han permitido que cantidades significativas de datos con diversas características sean capturadas y almacenadas en bases de datos. Por otra parte, el poder de

cómputo también ha tenido un gran avance en los últimos años, lo que ha permitido que grandes cantidades de información digitalizada sean fácilmente capturadas, almacenadas y procesadas.

La extracción de conocimiento a partir de datos es un proceso iterativo que se desarrolla a través de seis etapas principales en el ciclo de vida de los datos: captura, almacenamiento, preprocesado, análisis, visualización y publicación [17]. Cada fase se explica más a detalle a continuación.

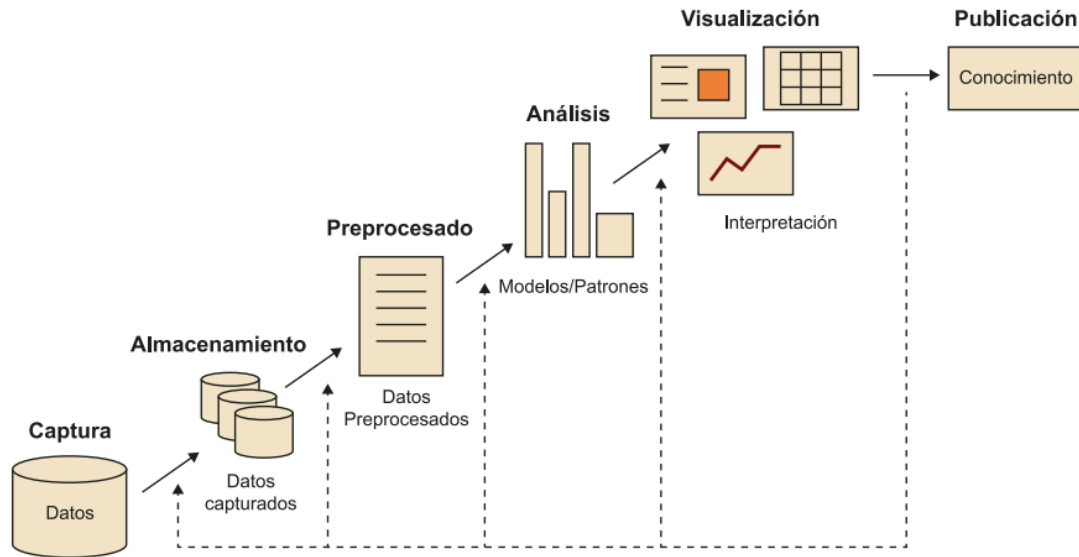


Figura 4.2: Etapas típicas en el ciclo de vida de los datos [17]

4.3.1.1 Captura

Es la primera fase del ciclo de vida y se refiere a la adquisición de datos generados por un proceso. Existen dos mecanismos principales:

- **Creación:** implica la implementación de rutinas que generan datos en tiempo real desde un sistema accesible. Ejemplos incluyen sensores o formularios.
- **Extracción:** se usa cuando no se tiene acceso directo al sistema que genera los datos. Incluye el uso de APIs, repositorios abiertos, *scraping web* o manipulación de URLs.

4.3.1.2 Almacenamiento

Los datos capturados deben almacenarse adecuadamente para su posterior uso. Existen varias formas:

- **Ficheros simples:** almacenamiento en estructuras básicas como archivos planos.
- **Bases de datos:** relacionales o no relacionales.
- **Data warehouses y datamarts:** integran datos desde múltiples fuentes para facilitar análisis específicos.

4.3.1.3 Preprocesado

Esta etapa prepara los datos para el análisis mediante:

- **Integración:** combinación de múltiples fuentes de datos.
- **Selección:** filtrado de datos relevantes.
- **Reducción:** disminución del volumen o dimensiones, manteniendo la integridad informativa.
- **Conversión:** transformación del formato o escalas, normalización, discretización, etc.
- **Limpieza:** eliminación de errores, datos faltantes, valores extremos e inconsistencias.

4.3.1.4 Análisis

Esta fase se dedica a modelar y extraer conocimiento:

- **Estadístico descriptivo e inferencial:** resumen o inferencia sobre poblaciones.
- **Extracción de características:** creación de nuevas variables o atributos.
- **Modelos supervisados:** aprendizaje con etiquetas conocidas.
- **Modelos no supervisados:** agrupamiento o identificación de estructuras ocultas.

4.3.1.5 Visualización

Permite representar los datos y sus análisis de manera gráfica para facilitar la comprensión.

4.3.1.6 Publicación

Consiste en documentar y compartir los resultados para su reutilización.

4.3.2 Aprendizaje supervisado

El aprendizaje supervisado es cuando se entrena un algoritmo proporcionándole las características o preguntas y las respuestas o etiquetas, para posteriormente hacer una predicción conociendo las características.

En este tipo de aprendizaje, hay dos algoritmos: el de clasificación y el de regresión. [15]

4.3.2.1 Algoritmos de clasificación

En este caso, el algoritmo nos indica a qué grupo pertenece el elemento en el estudio; esto lo hace por medio de una búsqueda de patrones en los datos que se le proveen para posteriormente clasificarlos.

4.3.2.2 Algoritmos de regresión

Este tipo de algoritmos espera un número. No lo ubica en un grupo, sino que devuelve un valor específico.

4.3.2.3 Métricas de evaluación

La elección de la métrica para evaluar un modelo es de suma importancia, ya que con esta es posible explicar la relación y el objetivo principal del fenómeno.

Con el fin de presentar una visión estructurada de las métricas más utilizadas en las principales técnicas de aprendizaje supervisado, a continuación se describen las categorías de regresión y clasificación junto con sus indicadores de desempeño habituales [18].

- **Regresión**
 - Error absoluto medio (MAE)
 - Raíz del error cuadrático medio (RMSE)
 - Coeficiente de determinación (R^2)
- **Clasificación**
 - Exactitud
 - Área bajo la curva ROC (AUC)
 - F1-*score*
 - Precisión
 - *recall* (sensibilidad)

4.4 Diseño centrado en el usuario

Es posible definir la experiencia del usuario como la sensación, sentimiento, respuesta emocional, valoración y satisfacción del usuario respecto a un producto, resultado de la interacción con el producto y con su proveedor [19].

4.4.1 ISO 9241-210, principios del diseño centrado en el usuario

La extensión de esta norma, la ISO 9241-210, determina una serie de principios que orientan el diseño centrado en el usuario. Estos principios aseguran que los sistemas desarrollados sean realmente efectivos desde la perspectiva del usuario final. Los principios son:

- El diseño está basado en una comprensión explícita de los usuarios, las tareas y los entornos.
- Los usuarios están involucrados en el diseño y el desarrollo.
- El diseño está dirigido y refinado por evaluaciones centradas en los usuarios.

- El proceso es iterativo.
- El diseño está dirigido a toda la experiencia del usuario.
- El equipo de diseño incluye habilidades y perspectivas multidisciplinarias[20].

4.4.2 Experiencia de usuario (UX)

El diseño centrado en el usuario ha evolucionado desde un enfoque meramente ingenieril, enfocado únicamente en el rendimiento funcional, hacia una definición más amplia que incorpora aspectos emocionales, sensoriales y contextuales, lo que hoy se conoce como experiencia de usuario (UX)[21].

Según The UX Book [21], la UX no puede diseñarse directamente, sino que ocurre como resultado de la interacción del usuario con el sistema en un contexto determinado.

Para estructurar el diseño UX, los autores proponen un **ciclo de vida compuesto por cuatro actividades fundamentales**:

1. **Analizar:** comprender el trabajo y las necesidades del usuario.
2. **Diseñar:** crear conceptos de diseño de interacción y definir comportamiento y apariencia.
3. **Implementar:** realizar prototipos funcionales a partir de los diseños.
4. **Evaluar:** verificar, probar y refinar el diseño con base en las metas de UX y el contexto de uso.

Este ciclo se representa como una rueda, permitiendo la *iteración constante entre etapas*, moviéndose hacia atrás en el proceso cuando sea necesario para refinar decisiones.

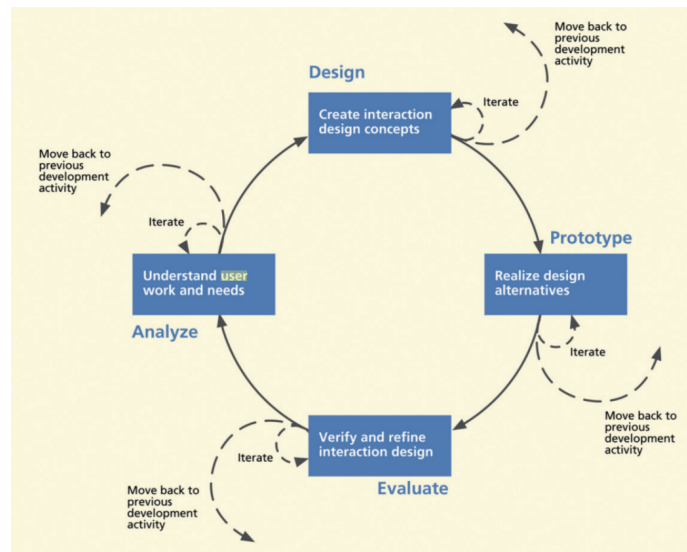


Figura 4.3: Ciclo de vida de UX, análisis, diseño, implementación y evaluación[21]

4.4.3 Interfaz de usuario (UI)

La interfaz de usuario (UI) puede entenderse como el espacio de comunicación que media la interacción entre una persona (usuario o desarrollador) y un sistema informático. No es únicamente

una colección de elementos visuales, sino un mecanismo comunicacional que permite representar acciones, tareas y procesos mediante objetos gráficos, metáforas e íconos que facilitan la comprensión del sistema por parte del usuario[22].

En este sentido, el diseño de interfaz de usuario debe garantizar una serie de cualidades fundamentales para que la experiencia del usuario sea efectiva, agradable y funcional.

En la propuesta de [22], se establece un sistema de principios que guían el diseño de la interfaz de usuario, aplicable a diferentes tipos de aplicaciones y metodologías de desarrollo de software. Los principios son:

- **El proceso de diseño de interfaz de usuario es un proceso complejo:** la interfaz es un sistema visual, interactivo e hipermedial que afecta directamente la comunicación usuario-sistema.
- **Todo proceso de diseño es preventivo-proyectivo:** anticipa necesidades y condiciones de uso mediante el análisis del contexto y los usuarios.
- **El proceso de diseño de interfaz de usuario es dialéctico:** el proceso evoluciona mediante contradicciones internas, integrando múltiples disciplinas.
- **Su carácter colaborativo y multidisciplinar garantiza la calidad del resultado final:** involucra a diversos actores para mejorar la calidad del diseño.
- **Todo proceso de diseño de interfaz debe ser flexible:** puede aplicarse a cualquier tipo de sistema y contexto.
- **El proceso de diseño de interfaz de usuario humaniza la tecnología:** busca transparencia e intuición en la interacción para lograr una mejor experiencia de usuario.
- **Todo proceso de diseño de interfaz de usuario debe ser ordenado, lógico, optimizado e incremental:** sigue metodologías estructuradas, asegurando coherencia en cada fase del desarrollo.

4.4.4 Principios visuales

En el diseño de interfaces, existen ciertos principios visuales que orientan la organización de los elementos gráficos para facilitar la comprensión y navegación del usuario. Hartson y Pyla [23] introducen el concepto de *meta-principios* del diseño de interfaz, que agrupan principios esenciales para una experiencia visual efectiva. Los principios son:

- **Consistencia:** implica mantener un lenguaje visual uniforme a lo largo de la interfaz. Esto incluye repetir patrones de diseño abarcando elementos como disposición, tipografía, color, íconos, entre otros, para que el usuario pueda anticipar comportamientos y reconocer funcionalidades sin necesidad de reaprender. La consistencia puede ser *interna*, cuando ocurre dentro de una misma aplicación, o *externa*, cuando se alinea con estándares usados en otras plataformas familiares al usuario.
- **Jerarquía:** es la percepción e interpretación de la importancia relativa de los objetos en una interfaz. Es influenciada por la posición, tamaño, color y tratamiento visual de los elementos, determinando así el orden en que el usuario interpreta la información. La falta de jerarquía sucede cuando todos los elementos llaman la atención del usuario de la misma manera. Debido a esto, es necesario crear diferenciación entre los elementos mediante contraste visual, proximidad y tratamiento gráfico, guiando así a los usuarios sobre qué acciones pueden tomar.

- **Personalidad:** hace referencia a las impresiones que los usuarios forman basándose en su apariencia y comportamiento visual. Se compone de tres aspectos del diseño: el diseño visceral, que se relaciona con la apariencia inicial; el diseño conductual, que abarca el placer y la efectividad durante el uso; y el diseño reflexivo, relacionado con la autoimagen, la satisfacción personal y los recuerdos. Para definir la personalidad visual, es necesario emplear *user personas*, que son perfiles ficticios basados en investigación sobre usuarios reales, incluyendo datos demográficos, patrones de comportamiento, motivaciones y objetivos.

4.5 Usabilidad

La usabilidad en sistemas de software surgió a partir del desarrollo del Internet como red de comunicación. Este concepto fue formalizado por Jakob Nielsen, reconocido como el “padre de la usabilidad”[24].

La usabilidad puede definirse como un atributo de calidad importante para el éxito de un producto, servicio o sistema[20]. También como las técnicas que ayudan a los seres humanos a realizar tareas en entornos gráficos de ordenador[25].

Una aplicación usable es la que permite al usuario centrarse en su tarea, no en la aplicación. De esta forma, un elemento esencial para que los sistemas sean más usables es *HCI* (Human-Computer Interaction), una disciplina cuyo propósito es garantizar que los usuarios se sientan cómodos al interactuar con una herramienta de software[25].

4.5.1 ISO 9241-11

El estándar internacional ISO 9241-11 define la usabilidad como el “grado al cual un producto puede ser utilizado por usuarios específicos para alcanzar metas específicas con eficacia (*effectiveness*), eficiencia (*efficiency*) y satisfacción, en un contexto especificado de uso”[25, p. 42].

Según este estándar, las dimensiones de la usabilidad se fundamentan en[25]:

- Eficacia (*effectiveness*): la exactitud y la cantidad con el cual los usuarios alcanzan metas especificadas.
- Eficiencia (*efficiency*): los recursos gastados con relación a la certeza con cual usuarios logran las metas.
- Satisfacción: la comodidad y la aceptabilidad del uso.

4.5.2 Evaluación de la usabilidad

Para la evaluación de la usabilidad existen varias propuestas de métodos, los cuales utilizan determinados medios y técnicas que intentan medir diferentes aspectos relacionados con esta[24]. Estos se suelen agrupar en tres grandes familias:

4.5.2.1 Métodos de inspección

En los métodos de inspección se realizan evaluaciones sin usuario, realizadas por expertos que revisan la interfaz frente a un conjunto de criterios o reglas.

Heurísticas de usabilidad de Nielsen

Jakob Nielsen propuso un conjunto de 10 principios heurísticos ampliamente utilizados para evaluar la usabilidad de interfaces. Estas heurísticas no son reglas estrictas, sino lineamientos generales[26]:

- Visibilidad del estado del sistema.
- Concordancia entre el sistema y el mundo real.
- Control y libertad del usuario.
- Consistencia y estándares.
- Prevención de errores.
- Reconocer en lugar de recordar.
- Flexibilidad y eficiencia de uso.
- Diseño estético y minimalista.
- Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores.
- Ayuda y documentación.

4.5.2.2 Métodos de indagación

En los métodos de indagación, se recogen opiniones y reacciones de usuarios.

System Usability Scale

El cuestionario *System Usability Scale* (SUS) creado por John Brooke en 1986 ha sido utilizado ampliamente como una herramienta rápida y confiable para evaluar subjetivamente la usabilidad percibida de sistemas interactivos[27].

El cuestionario está compuesto por diez ítems, donde los ítems impares están redactados positivamente y los pares negativamente:

1. Creo que me gustaría usar este sistema con frecuencia.
2. Encontré el sistema innecesariamente complejo.
3. Consideré que el sistema era fácil de usar.
4. Creo que necesitaría el apoyo de una persona técnica para poder usar este sistema.
5. Considero que las diversas funciones en este sistema están bien integradas.
6. Pienso que hay demasiada inconsistencia en este sistema.
7. Imagino que la mayoría de las personas aprenderían a usar este sistema muy rápidamente.
8. Encontré el sistema muy incómodo de usar.
9. Me sentí muy seguro usando el sistema.

10. Necesité aprender muchas cosas antes de poder comenzar a usar este sistema.

Para aplicar este cuestionario, cada enunciado se muestra a los participantes en una escala Likert de cinco niveles, donde 1 equivale a “Totalmente en desacuerdo” y 5 a “Totalmente de acuerdo”. Si alguien omite una pregunta, se le asigna el valor intermedio de 3.

Una vez finalizado, se calcula para cada ítem una puntuación entre 0 y 4. En los ítems positivos (1, 3, 5, 7 y 9) se extrae 1 del número; en los ítems negativos (2, 4, 6, 8 y 10) se sustrae el valor seleccionado de 5. La suma de todas estas puntuaciones se multiplica por 2.5, obteniéndose así la puntuación total del SUS, que va de 0 a 100 en saltos de 2.5 puntos.

Finalmente, ese resultado se compara con una tabla de referencia que lo clasifica en categorías cualitativas, de modo que sea más sencillo interpretar el grado de usabilidad percibido[27].

UX-Lite

Consiste en un cuestionario breve que permite obtener una medida confiable de la experiencia de usuario con un mínimo de tiempo y esfuerzo. Surge como una evolución del *UMUX-Lite*, manteniendo la validez psicométrica del instrumento original, pero simplificando su aplicación.

Consiste en dos ítems que evalúan dos dimensiones clave de la experiencia de usuario:

- **Facilidad de uso (*ease*):** “Este producto es fácil de usar”.
- **Utilidad percibida (*usefulness*):** “Las funciones de este producto satisfacen mis necesidades”.

A diferencia del *UMUX-Lite*, que utiliza una escala de siete puntos, el *UX-Lite* usa una escala de cinco puntos y reformula en un lenguaje más accesible el ítem de adecuación funcional, lo que facilita su comprensión por parte de los usuarios[28].

El resultado se calcula mediante el método de interpolación utilizado por [29]:

$$UX-Lite = \left(\frac{(Ease + Useful)}{2} - 1 \right) \times \frac{100}{4}$$

donde *Ease* corresponde al ítem “Esta plataforma es fácil de usar” y *Useful* al ítem “Las funciones de esta plataforma satisfacen mis necesidades”.

Net Promoter Score (NPS)

El *Net Promoter Score* es un indicador de lealtad que clasifica a los encuestados según su intención de recomendación en una escala de 0 a 10 [30]. Las respuestas se agrupan en *Promotores* (9–10), *Pasivos* (7–8) y *Detractores* (0–6). El puntaje se calcula restando el porcentaje de detractores al porcentaje de promotores, de acuerdo con la formulación original y su discusión en la literatura [30, 31]:

$$NPS = \left(\frac{\text{Promotores}}{N} - \frac{\text{Detractores}}{N} \right) \times 100$$

donde *N* es el número total de participantes.

4.5.2.3 Tests con usuarios

En los tests con usuarios, estos ejecutan tareas representativas mientras se registra su comportamiento.

Think aloud

Se les solicita a los usuarios, y de forma individual, que expresen en voz alta y libremente sus pensamientos, sentimientos y opiniones sobre cualquier aspecto del sistema o prototipo que están utilizando, con el fin de conocer aspectos relacionados con las actividades cognitivas de los usuarios potenciales del sistema[24].

Pruebas A/B

Las pruebas A/B constituyen una técnica de experimentación controlada que permite comparar dos variantes de un mismo diseño con el fin de evaluar cuál ofrece una mejor experiencia de usuario[32].

4.6 Aplicación web

Una aplicación web es una aplicación cliente/servidor, donde tanto el cliente como el servidor y el protocolo mediante el que se comunican (HyperText Transfer Protocol (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones[33].

4.6.1 Historia y evolución de las aplicaciones web

La historia de las aplicaciones web se relaciona al desarrollo de los estándares que dieron forma a la World Wide Web. En 1995 se publicó el borrador de HTML 3.0, que amplió significativamente las capacidades de estructuración del contenido en línea. Posteriormente, en 1999, el W3C formalizó el estándar de HTML 4.01 y se publicaron las primeras Web Content Accessibility Guidelines (WCAG 1.0), que marcaron el inicio de la accesibilidad digital.

En el año 2000, se consolidó XHTML 1.0. En esa misma década, tecnologías complementarias como las Cascading Style Sheets (CSS) fueron ganando popularidad, alcanzando una adopción mayoritaria hacia 2005, cuando más del 50 % de los sitios web ya las empleaban de manera consistente. Este avance permitió separar la presentación del contenido y dio paso a diseños más sofisticados y adaptables.

Otro hito importante se produjo en 2005 con la popularización de AJAX (Asynchronous JavaScript and XML), que revolucionó el desarrollo web al permitir la creación de interfaces mucho más dinámicas e interactivas. Finalmente, en diciembre de 2007, el W3C presentó las WCAG 2.0, que reforzaron y ampliaron las directrices de accesibilidad, respondiendo a las nuevas necesidades de la web contemporánea[34].

4.6.2 Guías de accesibilidad web y WCAG 2.2

Las guías de accesibilidad web o WCAG (Web Content Accessibility Guidelines) son un estándar del World Wide Web Consortium (W3C), publicado como «W3C Recommendation» el 5 de octubre de 2023, con el objetivo de proporcionar un estándar compartido para la accesibilidad del contenido web que satisfaga las necesidades de personas, organizaciones y gobiernos a nivel internacional.

La versión 2.2, la cual es la última versión publicada, se organiza bajo los cuatro principios fundamentales del estándar: perceptible, operable, comprensible y robusto. Estos principios establecen que la información y los componentes de la interfaz deben ser presentados de manera que los usuarios puedan percibirlos con independencia de sus capacidades sensoriales; que los elementos interactivos deben ser utilizables sin requerir interacciones que resulten imposibles para personas con limitaciones motoras o cognitivas; que la información y la operación de la interfaz deben ser comprensibles y predecibles; y que el contenido debe ser lo suficientemente robusto para que pueda ser interpretado por una amplia variedad de agentes de usuario, incluidas tecnologías de asistencia[35].

4.6.3 Enfoque *mobile-first*

El enfoque *mobile-first* surge como respuesta al crecimiento del uso de dispositivos móviles como principal medio de acceso a internet. Este modelo propone que el diseño y desarrollo de una plataforma web debe comenzar priorizando al usuario móvil, concentrándose en lo esencial del producto y en las funcionalidades que resultan más relevantes en contextos de movilidad. De este modo, se asegura que la experiencia en pantallas reducidas sea clara, eficiente y usable, para luego escalar progresivamente a dispositivos de mayor tamaño[36].

La diferencia principal entre *mobile-first* con el diseño adaptativo tradicional, consiste en que el con *mobile-first*, primero se define el contenido y funcionalidades necesarias en pantallas pequeñas, y posteriormente se amplía la experiencia para pantallas grandes, el diseño tradicional frecuentemente parte de una versión de escritorio y la reduce.

De acuerdo con la propuesta metodológica de Martínez et al. [36], un diseño de calidad basado en el enfoque *mobile-first* debe regirse por ciertos principios, entre ellos: la identificación del contenido esencial; evitar la sobrecarga en pantallas pequeñas; la simplificación de la navegación a un desplazamiento unidireccional; la priorización de la ligereza de las páginas para reducir tiempos de carga; la utilización de gráficos vectoriales escalables (*SVG*); el uso de unidades relativas como *em* o porcentajes; y la construcción de hojas de estilo desde la visualización móvil para luego ampliar la experiencia a dispositivos de mayor tamaño.

4.6.4 Modelo cliente-servidor

El modelo cliente-servidor es un modelo arquitectónico de red en la que cada ordenador o proceso en la red es cliente o servidor. Lo que implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores).

Esta arquitectura facilita la distribución de aplicaciones con separación de funciones, concurrencia y escalabilidad[33].

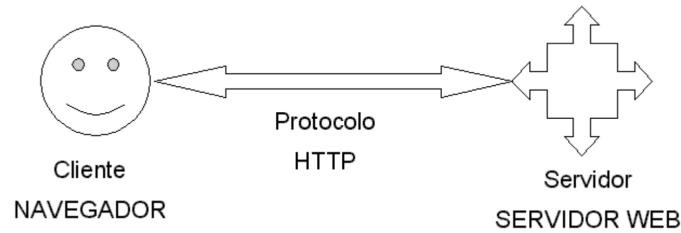


Figura 4.4: Esquema básico de una aplicación web [33]

4.6.5 Computación en la nube

La computación en la nube es un modelo de provisión de recursos informáticos, como servidores, almacenamiento, bases de datos, redes y software, a través de Internet[37]. Según el Instituto Nacional de Estándares y Tecnología (NIST), se compone de cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue:

- **Características esenciales:** autoservicio bajo demanda, acceso desde múltiples dispositivos, recursos compartidos, escalabilidad rápida y servicio medido con transparencia en costos[37].
- **Modelos de servicio:**
 - *SaaS*: aplicaciones listas para usar vía web.
 - *PaaS*: entornos para desplegar aplicaciones sin gestionar infraestructura.
 - *IaaS*: recursos básicos de cómputo y almacenamiento bajo control del usuario.
- **Modelos de despliegue:** privada (exclusiva de una organización), comunitaria (compartida por varias), pública (para el público en general) e híbrida (combinación flexible de las anteriores).

5.1 Enfoque mixto

Este proyecto adoptó un diseño metodológico mixto, que utilizó estrategias **cuantitativas** y **cualitativas** para responder de manera integral a los objetivos de investigación[38].

Para la parte **cuantitativa** se incluyó tanto la validación de los modelos predictivos como la evaluación de la usabilidad de la plataforma. Desde la perspectiva **cualitativa**, se recolectó información mediante encuestas y entrevistas, posteriormente analizada con el método de análisis temático.

La validez de la investigación se aseguró mediante **triangulación**, la cual consiste en la combinación de múltiples métodos y fuentes de datos[39].

Finalmente, la metodología integró tanto **evaluaciones formativas** como **sumativas**. Las formativas se realizaron durante el desarrollo, con el fin de guiar el diseño y mejorar los prototipos en cada iteración. Posteriormente, las sumativas se llevaron a cabo con la versión final[40].

5.2 Fase de investigación y diseño

Esta fase integró la investigación y el diseño de la plataforma, con el fin de sustentar su desarrollo y garantizar que las decisiones tomadas respondieran a las necesidades de los usuarios.

5.2.1 Investigación exploratoria

Esta fase comprendió la fase exploratoria del proyecto, en donde se realizaron encuestas y entrevistas, con el fin de comprender mejor la necesidad de los usuarios.

5.2.1.1 Encuestas

Con el objetivo de comprender la problemática y obtener una visión general de la experiencia de los visitantes respecto a los tiempos de espera, se diseñó y distribuyó una encuesta mediante Google Forms dirigida a personas guatemaltecas que han asistido al IRTRA Petapa.

El cuestionario presentado en el Anexo A.1 el cual fue contestado por 44 personas, incluyó nueve preguntas orientadas a medir la satisfacción general, la percepción del tiempo de espera y la disposición a utilizar herramientas digitales para obtener esta información.

Los hallazgos formativos presentados en las Figura 5.1 a Figura 5.4, mostraron que la mayoría de los participantes calificó los tiempos de espera como *tolerables* o *molestos*. Por otra parte, la satisfacción general con el parque se ubicó principalmente entre valores de 7 y 9 en una escala de 1 a 10. Asimismo, se observó que la satisfacción específica con la gestión de los tiempos de espera fue más baja en comparación con la satisfacción general, evidenciando un área de mejora clara. De manera complementaria, el 100% de los encuestados reportó utilizar su teléfono durante la visita, y más del 70% manifestó interés en poder consultar los tiempos de espera desde el propio dispositivo.

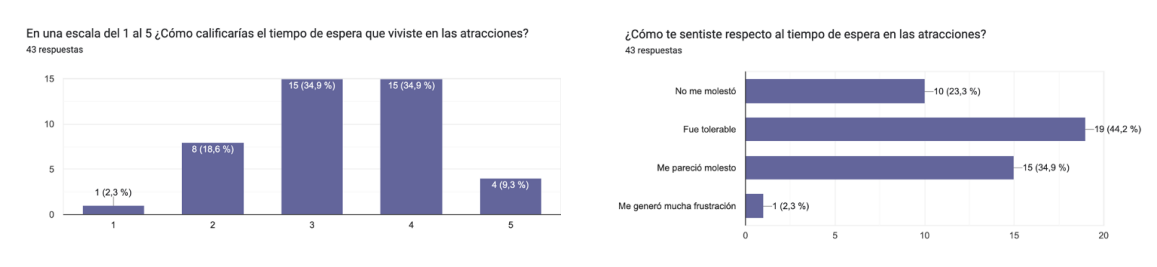


Figura 5.1: Hallazgos sobre la satisfacción y percepción de los visitantes

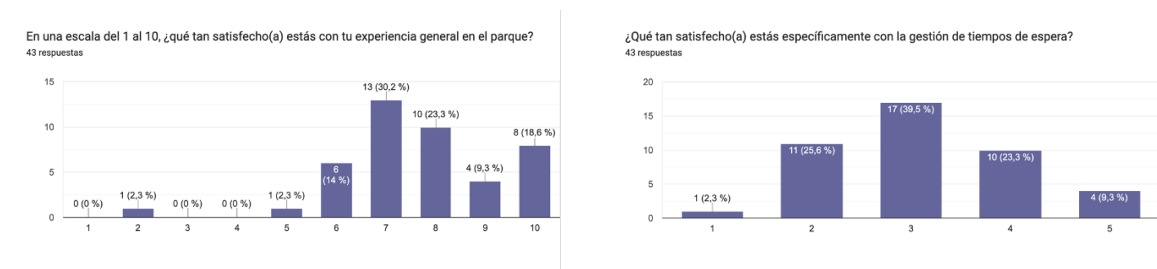


Figura 5.2: Hallazgos de la gestión y comparación de la satisfacción

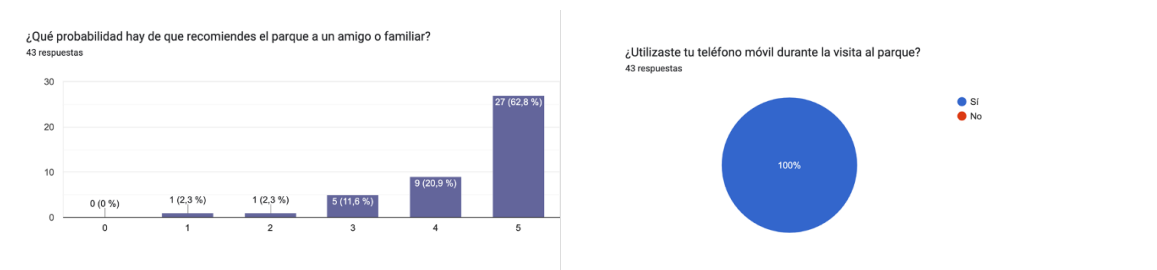


Figura 5.3: Hallazgos sobre los comportamientos digitales de los visitantes

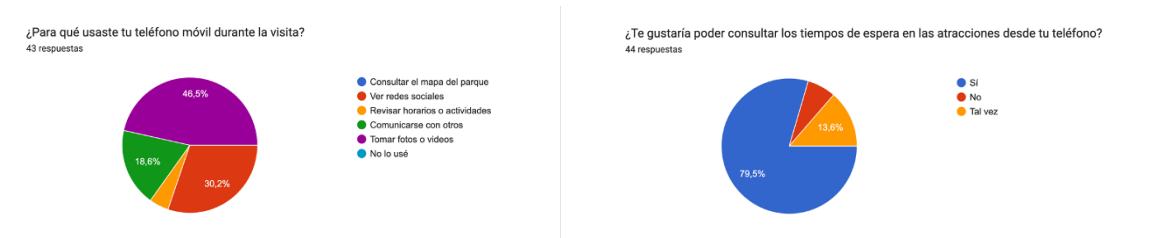


Figura 5.4: Hallazgos sobre la interacción móvil y consulta de tiempos de espera

5.2.1.2 Entrevistas

Posteriormente, se realizaron entrevistas estructuradas a seis personas de distintas edades, con el objetivo de profundizar en sus necesidades y expectativas respecto a la experiencia en los parques de diversiones. La entrevista se organizó en tres secciones clave: **perfil**, **experiencia general** y **escenarios**.

En la sección de perfil se consultó la edad y ocupación de cada participante; en la sección de experiencia general se les preguntó sobre visitas recientes a parques de diversiones, tanto nacionales como internacionales, la compañía habitual durante dichas visitas, experiencias tanto positivas como negativas y, específicamente, sobre sus visitas al IRTRA Petapa y las esperas en fila que experimentaron. Finalmente, en la sección de escenarios se plantearon situaciones hipotéticas, con el fin de explorar las decisiones que tomarían. El guion de la entrevista completa se presenta en el Anexo A.2.

Los hallazgos formativos de las entrevistas que guiaron el trabajo futuro mostraron lo siguiente:

Perfil de los participantes

Los entrevistados fueron personas de diferentes edades y ocupaciones. Todos han visitado el IRTRA Petapa, y varios también han asistido a parques internacionales como Disney.

Experiencia general

En cuanto a los aspectos positivos, se destacó el valor de la convivencia familiar y la posibilidad de disfrutar atracciones junto a hijos y amigos. Sin embargo, las principales experiencias negativas estuvieron relacionadas con filas largas, especialmente en días festivos o en temporada alta. Algunos participantes señalaron que, en comparación con parques internacionales, el IRTRA no ha mostrado avances significativos en la gestión de las colas.

Estrategias durante la espera

Los entrevistados mencionaron que, durante las filas, suelen recurrir a estrategias como conversar, consumir alimentos o utilizar el teléfono móvil. En el caso de quienes asistieron con niños pequeños, se mencionó la necesidad de distraerlos con juegos o dispositivos electrónicos para evitar el aburrimiento. En situaciones de espera excesiva, algunos participantes admitieron abandonar la fila y buscar otras atracciones.

Escenarios hipotéticos

Cuando se planteó un aumento de 45 minutos en el tiempo de espera, varios participantes afirmaron que optarían por retirarse de la fila, mientras que otros permanecerían únicamente si se les brindaba información clara y actualizada.

Ante el escenario de elegir entre una fila larga y otra aparentemente más corta, la mayoría indicó que preferiría cambiar de atracción si la atracción les llamaba la atención.

Necesidades

Demostraron necesidad por contar con información en tiempo real acerca de los tiempos de espera, ya sea mediante pantallas distribuidas en el parque o a través de una herramienta digital oficial. Asimismo, se sugirió la implementación de modalidades de acceso preferencial, similares a los sistemas de *fast pass* en parques internacionales. En cuanto a la infraestructura, se destacó la necesidad de contar con áreas de sombra, ventilación y elementos de distracción que disminuyan la incomodidad durante la espera.

5.2.1.3 Manual de codificación

Las entrevistas fueron transcritas y analizadas cualitativamente, elaborando un Manual de codificación (*codebook*) una herramienta esencial para la exploración de datos, la identificación de patrones y el desarrollo de ideas fundamentadas [41]. Este proceso permitió clasificar las respuestas en 11 códigos diferentes: *Disfrute en los parques*, *Conversación en tiempos de espera*, *Negatividad sobre la espera*, *Uso del celular*, *Evitar filas largas*, *Necesidad de información*, *Interacciones familiares*, *Mención de comida*, *Sin filas*, *Evaluación de condiciones* y *Mención de tiempos en días de alta demanda*. Cada código incluyó definición, categoría, ejemplos textuales y un color distintivo, utilizado para marcar su aparición en las transcripciones, véase Tabla 5.1.

Tabla 5.1: Ejemplo de estructura del manual de codificación

Código	Definición	Categoría	Ejemplos
Disfrute en los parques	Experiencias emocionales positivas sobre ir a parques de diversiones.	Experiencia en parque	“Estuvo alegre”, “Experiencia agradable”.
Conversación en tiempos de espera	Pláticas entre acompañantes para sobrellevar la espera.	Manejo del tiempo de espera	“Tuvimos que ponernos a platicar”, “Platicar con familia”.
Negatividad sobre la espera	Emociones negativas como frustración o ansiedad durante la espera.	Tiempo de espera	“Fue complicado esperar tanto”, “Me desesperé”.

5.2.1.4 *User persona*

A partir de los hallazgos formativos obtenidos en las encuestas, manual de codificación y entrevistas, se desarrollaron tres perfiles de usuario arquetípicos presentados en las Figura 5.5, Figura 5.6 y

Figura 5.7, los cuales describen intereses, metas, necesidades, frustraciones y rasgos de personalidad para cada perfil. Estos perfiles permitieron detallar esta información para guiar el diseño futuro de la plataforma:

- **Carla, 42 años.** Representa una mamá médica planificadora, responsable y organizada, que busca experiencias familiares agradables y libres de desorganización.
- **Luis, 22 años.** Simboliza a un joven estudiante sin responsabilidades y sociable, que busca maximizar la cantidad de atracciones disfrutadas en un día.
- **Carlos, 36 años.** Representa a un adulto que es diseñador gráfico freelance sin hijos, independiente y exigente con su tiempo, que valora experiencias cómodas y organizadas.

CARLA
Edad: 42 años
Profesión: Médica

DESCRIPCIÓN
Carla es una médica con dos hijas, es una persona responsable y busca experiencias familiares agradables. Le frustra la desorganización y las largas filas sin información clara, por lo que prefiere herramientas tecnológicas que le permitan planificar y evitar el estrés durante sus salidas familiares.

CARACTERÍSTICAS

- Paciente
- Intolerante al caos o desinformación.
- Familiar, empática y protectora.

INTERESES

- Usar apps de organización o planificación.
- Ver documentales, series o películas en casa.
- Paseos familiares.

METAS

- Disfrutar salidas familiares sin estrés ni contratiempos.
- Mantener a sus hijos entretenidos y tranquilos durante todo el día.
- Aprovechar el tiempo al máximo sin pérdidas innecesarias en filas.

NECESIDADES

- Áreas de descanso familiares
- Información anticipada para tomar mejores decisiones.
- Herramientas tecnológicas para aprovechar mejor el tiempo.

FRUSTRACIONES

- Largas filas sin alternativas.
- Mantener a sus hijos entretenidos y tranquilos durante todo el día.
- Aprovechar el tiempo al máximo sin pérdidas innecesarias en filas.

Figura 5.5: User persona 1 | Mamá planificadora

Figura 5.6: *User persona 2* | Joven sin responsabilidadesFigura 5.7: *User persona 3* | Adulto sin hijos

5.2.2 Diseño y prototipado

Esta etapa abordó el diseño iterativo de la plataforma mediante prototipos de baja y alta fidelidad.

5.2.2.1 Prototipos de baja fidelidad

Se elaboraron dos prototipos en baja fidelidad con el fin de explorar las preferencias de los usuarios:

Prototipo basado en listas

El primer prototipo presentado en la Figura 5.8, constó de una lista con las atracciones ordenadas según el tiempo de espera, destacando este dato como elemento principal.

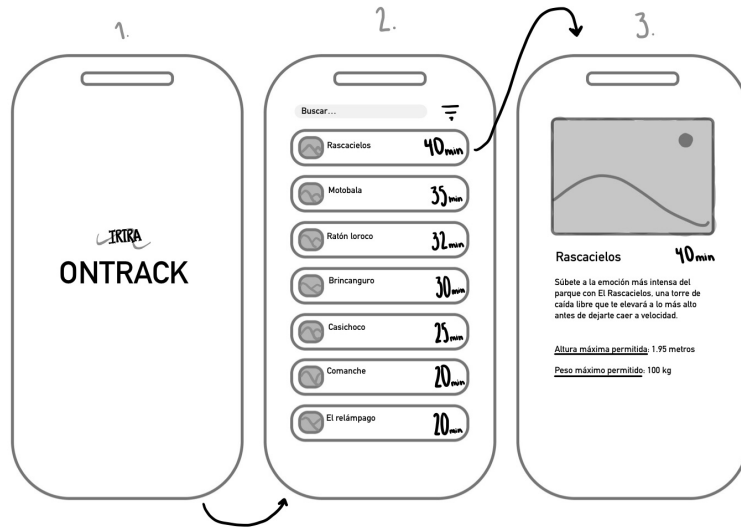


Figura 5.8: Prototipo baja fidelidad basado en listas

Prototipado basado en un mapa interactivo

El segundo prototipo presentado en la Figura 5.9, constó de un mapa interactivo priorizando la ubicación geográfica de las atracciones, con el tiempo de espera como información complementaria.

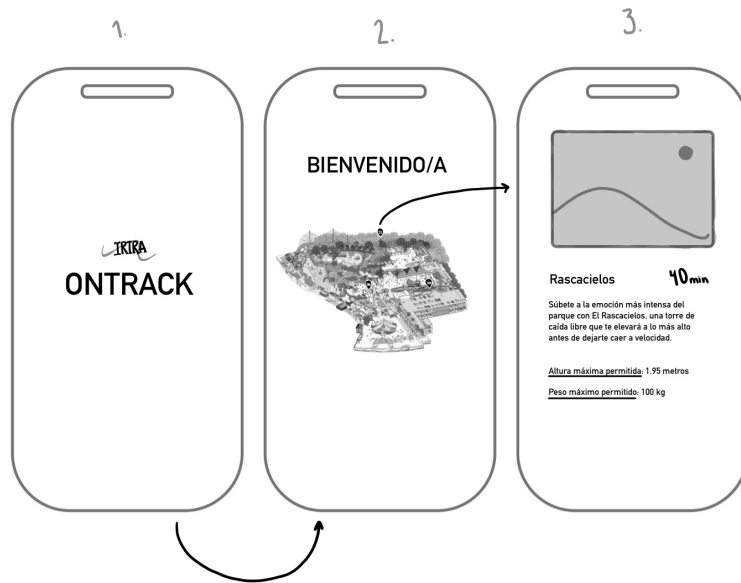


Figura 5.9: Prototipo baja fidelidad basado en mapa interactivo

Ambos prototipos incluyeron pantallas adicionales con la descripción y especificaciones de cada atracción.

Para evaluar estas opciones, se llevó a cabo un *test A/B* mediante una encuesta distribuida en redes sociales (Instagram y WhatsApp). La pregunta principal planteada a los participantes fue:

“Si pudiera ver los tiempos de espera aproximados de los juegos en el IRTRA Petapa, ¿cuál de estos dos diseños preferiría para hacerlo?”

El test A/B fue compartido mediante redes sociales (Instagram y WhatsApp) y participaron 232 personas.

De acuerdo con la evidencia formativa presentada en la Tabla 5.2, la **opción basada en listas** recibió el 69.82% de los votos, mientras que la **opción basada en un mapa interactivo** obtuvo el 30.17%, lo que posteriormente guió el diseño de la plataforma.

Tabla 5.2: Hallazgos de la encuesta A/B sobre diseño preferido

Plataforma	Opción	Votos
Instagram	Arriba	115
Instagram	Abajo	58
WhatsApp	Arriba	47
WhatsApp	Abajo	12

5.2.2.2 Prototipos de alta fidelidad

El prototipado de alta fidelidad se desarrolló en cuatro iteraciones sucesivas, alineado con el enfoque cíclico de diseño propuesto por Hartson y Pyla en *The UX Book* [21]. En cada iteración

se incorporaron ajustes tanto de navegación como de contenido en base a los hallazgos formativos obtenidos. El IRTRA Petapa proporcionó manuales de marca, logotipos y un mapa del parque, que se adaptaron y estandarizaron en formatos .svg y .png para garantizar una coherencia visual en la plataforma.

Versión 1

Con base en la evidencia formativa obtenida en los prototipos de baja fidelidad, la primera versión del prototipo de alta fidelidad se basó en un diseño con una lista de atracciones ordenada con tiempos de espera de mayor a menor. Cada atracción contaba con una pantalla de detalle que incluía información relevante como tiempo de espera, altura mínima, peso máximo y restricciones, así como la opción de visualizar la ubicación en el mapa.



Figura 5.10: Pantallas de la primera versión del prototipo de alta fidelidad

Versión 2

La segunda versión del prototipo se desarrolló tras haber tenido reuniones de retroalimentación con una experta en interacción humano-computador. Donde señaló la necesidad de mejorar la legibilidad y el contraste visual de los elementos. Con base en esto, se eliminaron gradientes innecesarios, se ajustaron colores de fondo para garantizar mayor contraste y se rediseñó el fondo de las tarjetas de atracciones. Asimismo, la lista de atracciones se reordenó de menor a mayor tiempo de espera, con el fin de favorecer una navegación más intuitiva.



Figura 5.11: Pantallas de la segunda versión del prototipo de alta fidelidad

En esta versión se realizó un proceso de prueba formativa con el fin de determinar la usabilidad del prototipo.

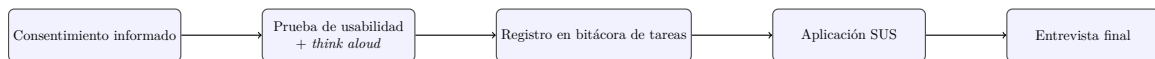


Figura 5.12: Flujo metodológico del proceso de prueba

Este proceso de prueba contó con los siguientes pasos: Antes de iniciar la evaluación, cada participante firmó el **consentimiento informado** (Anexo A.3). Posteriormente, la versión fue sometida a **pruebas de usabilidad estructuradas**, donde los usuarios completaron tareas en la plataforma, expresando en voz alta sus pensamientos mediante el método *think aloud* [24]. El guion completo de tareas y las instrucciones aparecen en el Anexo A.4.

Todas las actividades fueron documentadas en una **bitácora de tareas**, donde se anotó el tiempo empleado, el estado de finalización, la cantidad de errores y las observaciones de cada tarea.

Tabla 5.3: Bitácora de tareas utilizada en el proyecto

Persona	# Tarea	Duración	Completada	# Errores	Observaciones

Los tiempos promedio obtenidos (Tabla 5.4) mostraron que las tareas fueron completadas en menos de un minuto en todos los casos. Lo que indica que las interacciones básicas del sistema resultaron rápidas y comprensibles para los participantes. No obstante, se identificaron ciertas dificultades en tareas específicas, principalmente al aplicar filtros y al revisar los detalles de atracciones.

Tabla 5.4: Tiempos promedios por tarea en la prueba de usabilidad

Tarea	Tiempo promedio
T1	00:00:55
T2	00:00:52
T3	00:00:44
T4	00:00:24
T5	00:00:18

Después de completar las tareas, se aplicó el *System Usability Scale (SUS)* [27], con el fin de obtener una medida estandarizada sobre la facilidad de uso y la satisfacción general. El puntaje obtenido fue de 85.31, lo que, según el umbral propuesto por [42, 43], se clasifica dentro de la categoría *excellent*. A partir del benchmark descrito por Lewis [44], este resultado corresponde a una calificación *A+*. Por lo que, en general, los participantes percibieron el sistema como altamente usable (véase Tabla 5.5).

Tabla 5.5: Hallazgos de la escala SUS aplicada a los participantes

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Raw SUS	SUS Score
4	1	5	1	4	1	5	1	5	1	38	95.0
3	1	4	1	3	1	5	1	5	1	35	87.5
5	2	4	2	5	1	3	2	3	1	32	80.0
5	1	5	1	4	1	5	1	5	1	39	97.5
5	1	5	2	5	1	5	1	5	1	39	97.5
4	1	5	1	5	2	4	1	4	1	36	90.0
1	2	3	1	2	3	4	1	3	4	22	55.0
4	2	5	1	4	3	4	2	4	1	32	80.0
										34.1	85.31

Finalmente, se llevó a cabo una entrevista estructurada (Anexo A.5) para complementar los resultados cuantitativos del SUS con aspectos cualitativos de la experiencia. Los participantes señalaron principalmente problemas de distribución visual, tipografía poco legible, descripciones insuficientes y falta de jerarquía clara. También identificaron limitaciones en el mapa (sin *zoom*, sin ícono de ubicación y con pocos puntos de referencia) y dificultades con la visibilidad e identificación de algunos botones. Entre las mejoras sugeridas se incluyó incorporar filtros más específicos, un sistema de pines o colores con leyenda visual y mayor *feedback* en los elementos interactivos.

Es importante resaltar que estos hallazgos son **resultados formativos**, pues fueron utilizados para identificar fortalezas y áreas de mejora en esta etapa de desarrollo.

Versión 3

En la tercera versión del prototipo se rediseñaron las cartas de cada atracción, debido a algunos comentarios de los usuarios respecto a la inconsistencia en los tamaños de las imágenes que acompañaban a cada juego. Para validar si esta modificación era pertinente, se realizó un *test A/B* en el que

se comparó la nueva propuesta con la anterior, con el fin de determinar si los cambios aportaban a una mejor experiencia visual y de uso.

Los resultados de la votación (véase Tabla 5.6) demostraron preferencia por la versión anterior, lo que evidenció que los ajustes no necesariamente fueron percibidos como una mejora por la mayoría de los participantes.

Tabla 5.6: Resultados de la comparación entre la nueva propuesta (1) y la propuesta anterior (2)

Propuesta	Votos
Nueva propuesta	150
Propuesta anterior	179



Figura 5.13: Pantallas de la tercera versión del prototipo de alta fidelidad

Versión 4

En la última versión del prototipo se integraron los comentarios obtenidos en las iteraciones previas, esta última versión se enfocó en perfeccionar la usabilidad. Por lo que, se aseguró de contar con un contraste accesible y consistente entre los textos y los fondos, validado mediante la herramienta *Color Contrast Checker*¹, con el fin de cumplir criterios de legibilidad y accesibilidad. De igual forma, se agregó información realista en la pantalla de detalles de la atracción, proporcionada por el personal del IRTRA Petapa, para simular de mejor forma la experiencia final del usuario.

Finalmente, en el mapa se añadieron nuevas ubicaciones de referencia, como puntos de comida y servicios, con el propósito de facilitar la orientación dentro del parque y apoyar al visitante en la toma de decisiones al momento de desplazarse.

¹<https://color-contrast.incluid.dev/?fg=ffffff&bg=346494>



Figura 5.14: Pantallas de la cuarta versión del prototipo de alta fidelidad

5.3 Procesamiento y predicción de datos

Esta sección describe el proceso de captura, limpieza y análisis de datos, así como el desarrollo de los modelos de predicción.

5.3.1 Captura de datos

Los datos históricos fueron proveídos por el IRTRA Petapa, la información fue entregada en formato .xls, donde cada archivo representaba un día de apertura del parque en el período de 2022-2024 e incluía columnas con el nombre de la atracción, el horario, los ciclos y la asistencia registrada por cada hora, tal como se muestra en la Figura 5.15.

REPORTE A HORA DE ASISTENCIA A LOS RUCOS
Del 01/01/2022 al 01/01/2022

	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	
México	0	0	0	0	0	0	0	0	0	0	0
Caracas Guayana	0	0	0	0	0	0	0	0	0	0	0
Bahamas Whetl	0	0	41	0	88	0	102	0	100	0	99
Bahama Bahamas	0	0	0	0	0	0	0	0	0	0	0
Arizono	0	0	0	0	0	0	0	0	0	0	0
Esca Ribn	0	0	0	0	0	0	0	0	0	0	0
Dreano	0	0	14	221	13	193	14	218	214	13	207
Caracasmi Antiguo	0	0	0	0	0	0	0	0	0	0	0
Tera Andino	0	0	0	0	0	0	0	0	0	0	0
Cinamde	0	0	8	111	0	111	0	172	0	163	0
Dein	0	0	0	0	0	0	0	0	0	0	0
Bañta Lorena	0	0	70	210	72	191	77	222	76	214	75
Bermitio	0	0	72	240	72	230	72	240	72	230	72
Banacion	0	0	10	172	10	176	0	160	0	160	0
Treano Sulu	0	0	70	237	143	301	166	460	161	609	164
Banagran	0	0	10	155	11	248	11	250	11	247	11
Mein Bala	0	0	24	283	24	276	24	284	24	280	24
Carroy	0	0	8	42	10	72	10	98	11	113	10
Bermitio	0	0	2	41	2	81	0	103	0	96	0
Biri Magda	0	0	4	45	0	48	0	88	0	83	0
Paia Neve	0	0	11	48	0	54	0	66	0	72	0
San Volabera	0	0	0	0	0	0	0	0	0	0	0
Prado	0	0	0	0	0	0	0	0	0	0	0
Sid De Mi Barria	0	0	4	33	12	134	3	192	3	189	3
TIFA	0	0	2	167	2	163	2	161	2	160	2
Carusel	0	0	8	120	7	132	5	152	6	240	3
Treano	0	0	3	143	3	143	3	151	3	151	3
El Revatano	0	0	7	146	7	143	7	146	7	145	7
Guerra Pirata	0	0	5	51	5	100	0	146	5	123	5
Pais Sulu	0	0	9	11	11	120	11	127	11	130	11
El Bol mague	0	0	20	138	20	280	20	278	20	283	20
Cachibano	0	0	6	70	6	100	6	91	6	114	6
Totál:	0	0	428	2887	582	3888	588	4876	582	4254	4847

Figura 5.15: Estructura de los archivos .xls proveídos

5.3.2 Almacenamiento

Cada archivo fue almacenado en hojas de cálculo con extensión .xls, organizadas en directorios por año. Dentro de cada carpeta se incluyó un subdirectorio llamado "sucio" para los datos originales antes del proceso de limpieza, véase Figura 5.16.

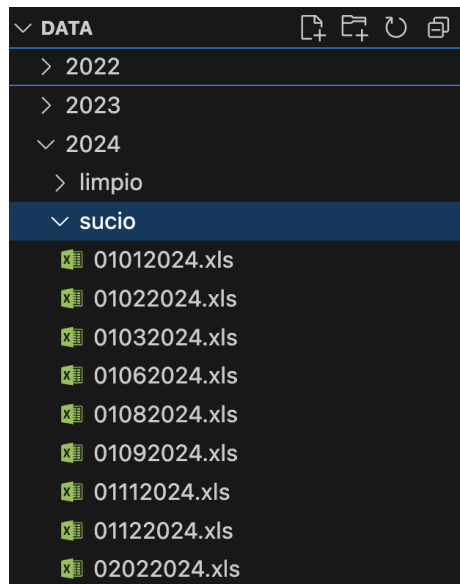


Figura 5.16: Estructura de directorios por año

5.3.3 Preprocesado

Dado que los datos fueron entregados en múltiples archivos `.xls`, uno por cada día de operación, fue necesario realizar un proceso de limpieza y unificación. Para esto, se implementó un script en Python el cual se encuentra disponible en el repositorio del módulo de datos ². El procedimiento realizado para cada archivo fue el siguiente:

- Se eliminaron los juegos y arcadas no relevantes como: *Carros Gnomos*, *Aviones*, *Carroussel Antiguo*, *Tren Antiguo*, *Delta*, *Los Voladores* y *Frisbi*, manteniendo únicamente los datos de las atracciones mecánicas que continúan en operación.
- Se renombraron los nombres y columnas para asegurar consistencia y unicidad en los nombres.
- Las asistencias y ciclos fueron reorganizadas en intervalos horarios de 9:00 a 18:00 horas.
- A cada registro se le agregó la fecha exacta, un indicador de si correspondía a un día festivo nacional en Guatemala, y condiciones climáticas obtenidas mediante la API de *World Weather Online*.
- Se generaron variables como la asistencia promedio por hora y el número promedio de ciclos por hora.

Finalmente, cada archivo fue almacenado en un directorio, con el nombre del año al que pertenecía.

5.3.4 Cálculo del tiempo de espera

Dado que los registros proporcionados por el IRTRA Petapa no incluían una variable de tiempo de espera, se optó por estimarla a través de la teoría de colas empleando el modelo $M|M|1$. El uso de este modelo se justifica porque cada atracción puede representarse como un sistema de cola de un solo servidor, donde los usuarios ingresan de acuerdo con un proceso de Poisson con tasa λ y los tiempos de servicio (ciclos de operación) siguen una distribución exponencial con parámetro μ [13]. Por lo que, aunque varias personas ingresan simultáneamente en un ciclo, la atracción se comporta como un único servidor efectivo que procesa un grupo por vez, lo cual es consistente con la notación de Kendall $M|M|1$ [45].

Además de su fundamento teórico, este planteamiento ha sido puesto en práctica en un estudio reciente, donde el modelo $M|M|1$ sirvió para estimar tiempos de espera y analizar el flujo de usuarios en atracciones [46]. No obstante, es importante subrayar que no es posible expresar el tiempo de espera de cada atracción con una sola fórmula matemática, ya que los visitantes interactúan entre sí y con el entorno, lo que introduce variaciones y limita la precisión absoluta de cualquier modelo [47].

5.3.4.1 Procedimiento computacional

Antes de realizar el cálculo del tiempo de espera para cada atracción, se consolidaron los `.csv` limpios de 2022–2024, se estandarizaron las fechas y los nombres de juegos, se derivaron variables de calendario (`day`, `month`, `day_of_week`), se generó la característica `temporada_alta` y se exportó el conjunto unificado a un `.csv` llamado `all_data` (Anexo B.1).

El procedimiento se llevó a cabo en dos fases principales:

²https://github.com/Kojimena/petapaontrack_ds/blob/main/reader.py

Uso del modelo M|M|1 por atracción

El conjunto de datos unificado `all_data` se agrupó por *juego* para tener un *dataframe* por cada atracción. En cada uno se organizaron las columnas horarias de *ciclos* y *asistencia* por hora $h \in \{09:00, \dots, 18:00\}$.

Luego, se implementó una función `mm1_perhour` que, para cada hora h y atracción, calculó:

$$\lambda_h = \frac{\text{asistencia}_h}{60}, \quad \mu_h^{nom} = \frac{\text{ciclos}_h \times \text{capacidad_por_ciclo}}{60}.$$

Sin embargo, dado que las atracciones no siempre operan a su capacidad máxima teórica, se incorporó un factor de eficiencia β , estimado a partir de los datos históricos (percentil 80 de la razón λ_h/μ_h^{nom} por atracción), proceso realizado en Anexo B.2. El ajuste aplicado se alinea con un estudio realizado en parques temáticos que afirma que la capacidad nominal suele fijarse en torno al 80 % de la asistencia mayor proyectada; esto debido a que los ciclos de carga/descarga y la variabilidad de la demanda reducen la capacidad real respecto a la nominal [47]. De esta forma, la tasa de servicio efectiva utilizada en el modelo fue:

$$\mu_h = \mu_h^{nom} \times \beta,$$

ajustando la capacidad a la fracción realmente alcanzada en operación.

Con estas tasas por minuto se derivaron la utilización $\rho_h = \lambda_h/\mu_h$ y el tiempo medio de espera en cola $W_{q,h}$ bajo condiciones de estabilidad ($\mu_h > \lambda_h$):

$$W_{q,h} = \frac{\lambda_h}{\mu_h(\mu_h - \lambda_h)}.$$

Para horas sin operación ni demanda, se fijó $W_{q,h}=0$; para horas con demanda pero sin servicio y para saturación, en lugar de asignar un valor infinito, se aplicó el tope máximo predefinido del juego. Esta decisión se tomó ya que, desde el punto de vista computacional, es necesario reemplazar o eliminar dichos valores para evitar fallos durante el entrenamiento del modelo; de hecho, [48] sugiere explícitamente acotar las observaciones atípicas dentro de un rango razonable como una forma válida de tratar valores atípicos. Por otra parte, en contextos recreativos, los visitantes no esperan indefinidamente; de hecho, la teoría de colas introduce los conceptos de balking (rechazo a entrar a la cola cuando está excesivamente larga) y reneging (abandono de la cola tras esperar un tiempo prolongado sin servicio) lo que implica que siempre hay un límite finito de paciencia del cliente[49].

La función generó, por cada hora, columnas de salida con λ_h , μ_h , ρ_h , $W_{q,h}$ y el *estado* de la hora (*closed*, *off service*, *stable*, *saturated*).

Exportación por juego

Se escribieron 25 archivos `.csv`, uno por atracción, con los datos históricos (2022,2023,2024), para su uso posterior en los modelos de *machine learning*.

La parametrización de `capacidad_por_ciclo` y del tope de espera w^{\max} se definió por juego y fue información brindada por el personal del IRTRA Petapa (véanse los diccionarios `CAPACIDADES_JUEGOS` y `MAX_TIME_JUEGOS` en el Anexo B.3).

5.3.5 Análisis de datos

Para cada archivo .csv por atracción, se realizó un análisis exploratorio de datos (EDA). Este proceso incluyó la detección de valores faltantes, la exploración de la distribución de variables relevantes y la identificación de relaciones mediante histogramas, diagramas de caja y matrices de correlación.

En el análisis se observó que la asistencia al parque tuvo un crecimiento significativo entre 2022 y 2023; las atracciones más demandadas, como *Tronco Splash* y *Ratón Loroco*, mostraron asistencia considerablemente superior a juegos de menor capacidad y para menores, como *Bici Mágica* o *Convoy* véase Anexo B.4.

También se determinó que los días con más afluencia en el parque son los domingos, seguidos de los sábados; véase Anexo B.5.

5.3.6 Selección de *features*

Con base en el análisis realizado, se definió un conjunto de variables predictoras compuesto por:

- `day_of_week`: día de la semana.
- `hora`: hora
- `month`: mes
- `day`: día
- `temperatura_max`: temperatura máxima de ese día.
- `es_festivo`: si el día es festivo
- `temporada_alta`: si el día está en temporada alta.
- `asistencia_h`: asistencia por hora
- `ciclos_h`: ciclos por hora

5.3.7 Desarrollo de modelos

Una vez obtenido el conjunto de entrenamiento, se codificaron las variables categóricas.

Para la variable objetivo se utilizó el tiempo de espera (W_q), calculado anteriormente mediante el modelo de colas $M|M|1$, mientras que como variables de entrada se seleccionaron tanto **categóricas** (día de la semana, hora, mes, día y condiciones del cielo) como **numéricas** (temperatura máxima, asistencia por hora, festivo y ciclos).

Los datos fueron transformados a un formato largo, para poder disponer de múltiples registros por hora y así mejorar la granularidad del entrenamiento.

Asimismo, se aplicaron procedimientos de limpieza, eliminando registros asociados a atracciones cerradas. Los valores faltantes fueron transformados mediante la mediana y se aplicó detección de valores atípicos con el criterio IQR, eliminándose únicamente cuando representaban menos del 5% del conjunto.

Esto se decidió para no afectar el entrenamiento de los modelos con outliers, dejando únicamente la tarea de mostrar si el juego se encuentra cerrado o abierto en la parte administrativa de la plataforma.

Los datos se dividieron en conjuntos de entrenamiento y prueba (80/20) para validar el desempeño sin sesgo[50].

La evaluación se realizó con las siguientes métricas: **error absoluto medio** (MAE), **raíz del error cuadrático medio** (RMSE) y **coeficiente de determinación** (R^2), ya que permiten reflejar la precisión en minutos de espera según los cálculos previos realizados [51].

Posteriormente, se implementó un *Random Forest Regressor* como modelo de predicción por su capacidad de capturar relaciones no lineales y su robustez frente a valores atípicos y estructuras heterogéneas [50]. Todo el proceso de modelado y evaluación se implementó utilizando la biblioteca `scikit-learn`³, que facilitó la codificación de variables, la división de los datos y la aplicación de los modelos de regresión. Este proceso puede consultarse en el repositorio del proyecto⁴.

Finalmente, los modelos entrenados con mejores resultados fueron exportados y almacenados en formato `.pkl`, lo que permitió su integración en el sistema de predicción.

5.4 Implementación del sistema

La aplicación desarrollada se dividió en dos componentes principales: *frontend* y *backend*, los cuales fueron integrados con el fin de consultar y administrar los tiempos de espera y la información de las atracciones del IRTRA Petapa.

La arquitectura utilizada para el sistema fue un **modelo cliente-servidor**, debido a que es una arquitectura ampliamente utilizada en el desarrollo de aplicaciones distribuidas [33].

Se decidió utilizar este modelo, ya que este esquema es beneficioso para la separación de responsabilidades y la escalabilidad del sistema [33]. En el caso de este proyecto:

- El *frontend* actúa como *cliente*.
- El *backend* funciona como *servidor*.

5.4.1 *Frontend*

El desarrollo se realizó utilizando el *framework* **Next.js** basado en React. Este *framework* ofrece distintas ventajas para la construcción de aplicaciones web modernas, entre estas: el renderizado híbrido (*server-side rendering* y *static generation*), la optimización automática de rutas y una mayor eficiencia en el tiempo de carga de las páginas, lo que es de suma importancia para que el usuario acceda de forma rápida y accesible a la información.⁵ Todo el código del *frontend* del proyecto puede consultarse en el repositorio público⁶.

³<https://scikit-learn.org/stable/>

⁴https://github.com/Kojimena/petapaontrack_ds/tree/main/data_analysis/eda

⁵<https://nextjs.org/docs/app>

⁶<https://github.com/Kojimena/petapaontrack>

5.4.1.1 Componentes

El *frontend* se diseñó de forma modular, siguiendo el paradigma de desarrollo basado en componentes. Lo que permitió construir una interfaz escalable, donde elementos fueron reutilizados y combinados de manera flexible[52].

Los componentes realizados fueron los siguientes:

- *Header*: barra superior de navegación que contiene el título y facilita la navegación.
- *Footer*: sección inferior con logo institucional.
- *Loader*: componente de retroalimentación visual que indica al usuario que la información está siendo cargada.
- *GameList*: lista dinámica que renderiza todas las atracciones disponibles, ordenadas según distintos criterios.
- *SearchBar*: barra de búsqueda que permite filtrar atracciones.
- *Filters*: conjunto de controles para aplicar filtros.
- *GameCard*: componente encargado de mostrar la información detallada de una atracción específica.
- *Game*: tarjeta visual que resume datos clave de cada atracción.
- *Map*: componente interactivo basado en SVG que permite visualizar la ubicación de las atracciones y puntos de referencia dentro del parque.
- *FaqsComponent*: sección de preguntas frecuentes que organiza y muestra respuestas a dudas comunes de los usuarios.
- *FloatingButton*: botón flotante de soporte.
- *PinComponent*: marcador visual en el mapa que señala la ubicación de un lugar.
- *WalkThrough*: componente que guía paso a paso al usuario en el uso inicial de la aplicación.

5.4.1.2 Páginas

Las páginas se organizaron siguiendo el modelo de enrutamiento de Next.js⁷, el cual permite crear rutas a partir de la estructura de directorios dentro de la carpeta `app`, con el fin de separar las responsabilidades de cada vista.

En este proyecto se implementaron las siguientes páginas:

- **Inicio** (`page.jsx`): página principal de la aplicación que despliega el listado general de atracciones y sus tiempos de espera.
- **Detalles de atracción** (`game/[id]/page.jsx`): página dinámica que carga la información específica de cada juego, incluyendo tiempo de espera y restricciones.
- **Layout general** (`layout.jsx`): componente que define la estructura de las páginas.

⁷<https://nextjs.org/docs/app>

- **Mapa por atracción** (`map/[id]/page.jsx`): página dinámica con el mapa del parque mostrando puntos de referencia cercanos y la atracción seleccionada.
- **Preguntas frecuentes** (`faqs/page.jsx`): vista informativa que responde a las dudas más comunes de los usuarios sobre el uso de la aplicación.
- **Listado de juegos** (`games/page.jsx`): página que presenta todas las atracciones disponibles.

5.4.1.3 Despliegue

El despliegue se realizó en **Netlify**, ya que es una plataforma que cuenta con integración continua desde el repositorio y con soporte nativo para proyectos basados en Node.js.

Se configuró la rama principal como origen de despliegue, se definió el comando de construcción (`npm run build`) y se estableció el directorio de publicación correspondiente. Asimismo, se declararon variables de entorno dentro del panel de Netlify. La configuración se documentó siguiendo las guías oficiales de Netlify.⁸

5.4.2 Backend

El *backend* constituye el punto central de lógica en la aplicación, ya que es el responsable de administrar los datos de cada atracción y proveerlos al *frontend* mediante una API. Sin embargo, es importante aclarar que la creación de un *backend* desde cero no era el objetivo principal de este proyecto, sino más bien implementar un sistema *backend* que permitiera conectar el modelo de predicción de cada atracción con la interfaz. Debido a esto, se optó por utilizar una solución ligera y ya estructurada que permitiera centrar los esfuerzos en el análisis de datos y la experiencia de usuario, sin sacrificar la robustez ni escalabilidad de la plataforma.

5.4.2.1 Tecnologías

Para la construcción del *backend* se evaluaron distintas alternativas de sistemas ligeros de base de datos con administración integrada, entre las que destacan:

- **Supabase**: plataforma de código abierto que ofrece una base de datos relacional (PostgreSQL) con autenticación, almacenamiento y API en tiempo real ⁹.
- **Firebase**: solución de Google orientada a aplicaciones web y móviles, que integra base de datos en tiempo real, autenticación y almacenamiento de archivos en la nube ¹⁰.

No obstante, se eligió **PocketBase**, una solución compacta y autosuficiente que integra una base de datos (*SQLite*), autenticación, un panel de administración web, una API REST y soporte en tiempo real dentro de un único ejecutable. ¹¹.

La decisión de utilizar esta herramienta se basó en la poca cantidad de tablas/colecciones que contiene la plataforma, la facilidad de despliegue de la herramienta y el hecho de que ofrece una API REST y en tiempo real lista para consumir desde el *frontend*.

⁸<https://docs.netlify.com/>

⁹<https://supabase.com/docs>

¹⁰<https://firebase.google.com/docs>

¹¹<https://pocketbase.io/docs/>

5.4.2.2 Base de datos

PocketBase sigue un modelo relacional simplificado y permite definir colecciones que funcionan como tablas. En este proyecto se diseñó una colección principal *games* que almacena los datos para cada atracción:

- **image**: campo de tipo imagen para almacenar el icono de cada atracción.
- **name**: campo de tipo texto para el nombre de la atracción.
- **description**: campo de tipo texto enriquecido para la descripción de la atracción.
- **restrictions**: campo de tipo texto enriquecido para las restricciones de la atracción.
- **inner_color**: campo de tipo texto con el color interno de cada tarjeta en la lista de atracciones.
- **outer_color**: campo de tipo texto con el color externo de cada tarjeta, utilizado también en la página de detalles.
- **time**: campo numérico que representa el tiempo de espera de la atracción.
- **rules_height**: campo de tipo JSON con las reglas de altura para cada juego.

5.4.2.3 Administración del *backend* y manejo de colecciones

A partir de la estructura de datos descrita anteriormente, **PocketBase** integra un panel de administración que permite gestionar la información almacenada en la base de datos, creando, editando y eliminando registros de las colecciones definidas sin modificar directamente el código ni interactuar manualmente con la base de datos.

Asimismo, este panel permite supervisar el flujo automatizado de actualización de los tiempos de espera, de modo que la persona encargada del sistema (usuario administrador) pueda verificar que las predicciones se reflejen correctamente y, cuando es necesario, realizar ajustes manuales sobre los datos.

5.4.2.4 Despliegue e infraestructura

El *backend* se desplegó en una instancia **Amazon EC2**, debido a la flexibilidad de AWS para garantizar escalabilidad y disponibilidad.¹²

Para este proceso se seleccionó una máquina virtual con Ubuntu Server bajo el plan *free tier*, específicamente una instancia **t3.micro**.

En la configuración inicial de la instancia se establecieron diferentes reglas de seguridad en los *security groups*, habilitando únicamente los puertos necesarios (SSH, HTTP y HTTPS) tal como se muestra en el Anexo C.1. Asimismo, se generó y descargó una llave de acceso en formato **.pem**, para conectarse al servidor mediante un cliente llamado Termius¹³.

Finalmente, se configuró una dirección IP elástica (*Elastic IP*), con el objetivo de mantener una dirección estática, véase el Anexo C.2.

¹²<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

¹³<https://termius.com/index.html>

Una vez creada la infraestructura, se procedió al despliegue de **PocketBase**, descargando la versión correspondiente y ejecutándola ¹⁴. Se configuró **Nginx** para asegurar la accesibilidad externa como *reverse proxy*, asignando un dominio propio y redirigiendo las solicitudes hacia el servicio de PocketBase en el puerto interno. Esta configuración se almacenó en el archivo `/etc/nginx/sites-available/pocketbase`, enlazado posteriormente en `sites-enabled`, se incluye el script completo en el Anexo C.3. Finalmente, se instaló y configuró **Certbot** ¹⁵, habilitando certificados SSL y garantizando el acceso seguro mediante HTTPS.

5.4.2.5 Automatización

Con el fin de mantener la información actualizada, se configuró un **cronjob** en el servidor EC2.

El flujo completo de la automatización se estructuró de la siguiente manera:

1. Los modelos entrenados (`.joblib`) y los datos de entrada por atracción (`.csv`) fueron transferidos al servidor y organizados en directorios específicos (`models/` y `data/by_game/`), como se muestra en el Anexo C.4.
2. Cada script en Python fue diseñado para cargar un modelo, procesar los datos del día y generar la predicción correspondiente al tiempo de espera.
3. Se implementó un archivo auxiliar (`pb_helpers.py`) encargado de la comunicación con la base de datos **PocketBase**, autenticando la conexión y actualizando el campo `time` de la colección **Games** (Anexo C.5).
4. Finalmente, se programaron los *cronjobs* en Linux (Anexo C.6), de modo que cada atracción ejecutara su script asociado en intervalos de una hora, asegurando que la información estuviera siempre sincronizada con la base de datos.

5.5 Evaluación de la plataforma

La última etapa consistió en evaluar la usabilidad, experiencia de usuario y la satisfacción de la versión final ya desarrollada de Petapa OnTrack en condiciones cercanas a un uso real dentro del parque IRTRA Petapa. Para esto, se reclutaron diez participantes de diferentes edades y géneros que aceptaron participar de manera voluntaria en la evaluación.

5.5.1 Datos de la muestra

La muestra estuvo distribuida equitativamente por género (50 % femenino y 50 % masculino). En cuanto a la edad, la mayoría se concentró en el rango de 18 a 24 años (70 %), seguido por el grupo de 25 a 34 años (20 %), mientras que un 10 % correspondió a personas mayores de 45 años. No se registraron participantes en el rango de 35 a 44 años.

Respecto al nivel educativo, el 50 % indicó contar con estudios universitarios, un 30 % con nivel diversificado y un 20 % con educación secundaria; no se reportaron participantes con estudios de primaria ni de posgrado.

¹⁴<https://pocketbase.io/docs/>

¹⁵<https://certbot.eff.org/>

En relación con la ocupación, el 60 % de los participantes se identificó como trabajadores, un 30 % como estudiantes y un 10 % como desempleados.

El consentimiento informado fue aceptado por el 100 % de los participantes, y todos autorizaron la grabación de pantalla, audio o video durante la prueba de usabilidad.

En la Figura 5.17 se presentan las características demográficas de los participantes.

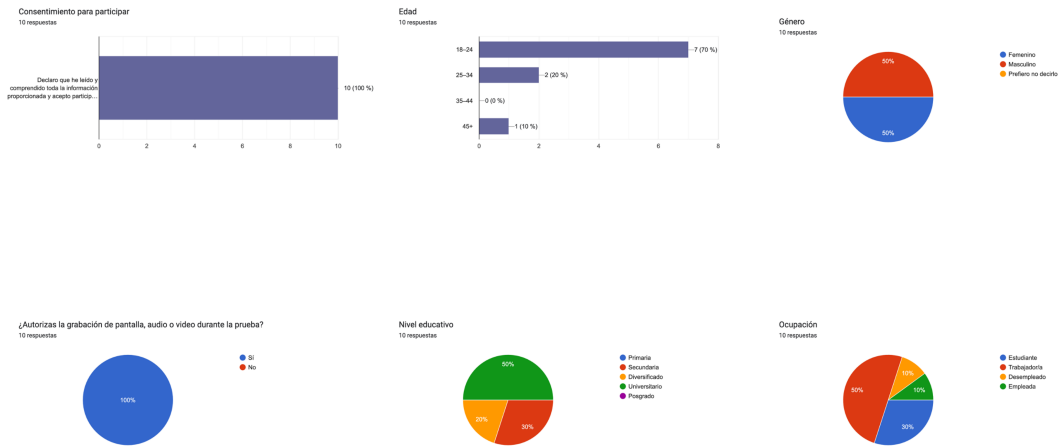


Figura 5.17: Resultados demográficos y consentimiento de los participantes

Como se muestra en la Tabla 5.7, la evaluación se estructuró en cuatro actividades principales con sus respectivas duraciones y requisitos.

Tabla 5.7: Plan de evaluación de la plataforma final

Actividad	Duración	Requisitos
Formulario demográfico y consentimiento informado (Anexo D.1)	5 minutos	<ul style="list-style-type: none"> ■ Dispositivo móvil ■ Navegador web
Pruebas de usabilidad con <i>think aloud</i>	10 minutos	<ul style="list-style-type: none"> ■ Dispositivo móvil ■ Navegador web
Cuestionarios SUS, <i>UX-Lite</i> y NPS (Anexo D.2)	10 minutos	<ul style="list-style-type: none"> ■ Dispositivo móvil ■ Navegador web
Entrevista estructurada (Anexo D.3)	10 minutos	<ul style="list-style-type: none"> ■ Dispositivo móvil ■ Grabación presencial ■ Micrófono

6.1 Evaluación de usabilidad y diseño

En esta sección se presentan los resultados de la evaluación del diseño, la usabilidad, la experiencia de usuario y el grado de recomendabilidad de la plataforma. Los resultados corresponden a la participación de diez personas que interactuaron con la plataforma.

6.1.1 Pruebas de usabilidad

En esta sección se presentan los resultados obtenidos de los participantes en las siguientes pruebas de usabilidad realizadas:

1. Solo tienes 30 minutos antes de irte del parque. ¿Qué atracciones podrías visitar?
2. Tienes 2 horas disponibles. ¿Cuántas atracciones visitarías? Haz tu itinerario considerando los tiempos de espera.
3. Imagina que a tu hermano menor no le gusta esperar. ¿Cómo encontrarías las atracciones con menor tiempo de espera que podrían adaptarse a él?
4. Tu hermano mide 1.40 m y quiere subirse al *Relámpago*. ¿Cómo verificarías si puede ingresar a esa atracción?
5. Estás intentando ver el tiempo de espera de *Tronco Splash*, pero la plataforma no carga la información. ¿Qué harías para reportar este problema?
6. Llegaste un miércoles a las 10:00 am y el parque está cerrado. Busca en las preguntas frecuentes los días y horarios de apertura.

Tabla 6.1: Resumen general de resultados por tarea en la prueba de usabilidad

Tarea	Tiempo promedio (s)	Tasa de completación (%)	Número de errores
1	26.90	100.00	0
2	24.00	100.00	0
3	32.40	90.00	2
4	30.70	100.00	0
5	16.60	100.00	1
6	20.20	100.00	0

La tarea con menor tiempo promedio fue la 5, con 16 segundos, seguida de la 6 con 20 segundos. Las 2 y 1 registraron tiempos de 24 y 27 segundos, respectivamente, mientras que las 4 y 3 presentaron los mayores promedios, con 31 y 32 segundos.

En cuanto a la tasa de completación, cinco de las seis tareas fueron finalizadas por el 100% de los participantes, mientras que la tarea 3 obtuvo una tasa del 90%. Con respecto al número total de errores, se registraron tres en total: dos en la tarea 3 y uno en la tarea 5.

En la Tabla 6.1 se detallan los tiempos promedio registrados por tarea durante la prueba de usabilidad.

6.1.2 Cuestionarios

En esta sección se presentan los cuestionarios realizados a los participantes de las pruebas de usabilidad.

6.1.2.1 SUS

En este apartado se presentan los resultados obtenidos a partir de la aplicación de la escala SUS a los participantes. La Tabla 6.2 muestra los puntajes individuales obtenidos para cada participante, así como el valor total calculado de la escala.

Tabla 6.2: Resultados de la escala SUS aplicada a los participantes

Participantes	Preguntas										Raw SUS	SUS Score
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
1	5	2	4	3	3	2	3	3	4	3	26	65.0
2	5	1	5	2	5	2	5	1	5	1	38	95.0
3	3	3	3	3	3	2	2	3	3	3	20	50.0
4	1	3	5	1	5	1	5	1	5	1	34	85.0
5	4	3	4	5	2	2	1	4	1	5	13	32.5
6	5	4	5	1	5	2	5	1	5	1	36	90.0
7	2	1	5	1	5	1	5	1	5	1	37	92.5
8	5	1	5	3	5	1	5	2	5	5	33	82.5
9	5	1	5	3	5	1	4	1	5	1	37	92.5
10	5	1	5	1	4	1	5	1	5	1	39	97.5
Promedio											31.3	78.25

El puntaje final promedio obtenido fue de **78.25**. El valor más bajo registrado fue de 32.5, mientras que el más alto alcanzó 97.5.

6.1.2.2 Escala *UX-Lite*

En este apartado se presentan los resultados obtenidos a partir de la aplicación de la escala *UX-Lite* a los participantes de la prueba de usabilidad. La Tabla 6.3 muestra los puntajes individuales obtenidos en los dos ítems de la escala, evaluados en una escala de 1 a 5, así como el valor total transformado a una escala de 0 a 100.

Tabla 6.3: Resultados de la escala *UX-Lite* aplicada a los participantes

Participante	Fácil de usar	Satisface necesidades	<i>UX-Lite</i> (0–100)
1	3	4	62.5
2	5	5	100.0
3	2	3	37.5
4	5	5	100.0
5	1	2	12.5
6	5	5	100.0
7	5	3	75.0
8	5	5	100.0
9	5	5	100.0
10	5	5	100.0
Promedio general			78.75

El puntaje promedio general obtenido fue de **78.75 / 100**. El valor más bajo registrado fue de 37.5, mientras que el más alto, 100.

6.1.2.3 *Net Promoter Score* (NPS)

En este apartado se presentan los resultados obtenidos a partir de la aplicación del *Net Promoter Score* (NPS) a los participantes. La Tabla 6.4 muestra las respuestas individuales registradas en una escala de 0 a 10.

Tabla 6.4: Respuestas individuales de los participantes en la escala NPS

Participante	Puntaje (0–10)
1	8
2	10
3	4
4	10
5	8
6	10
7	7
8	10
9	10
10	10

Tabla 6.5: Resultados generales del *Net Promoter Score* (NPS)

Categoría	Cantidad de participantes
Promotores (9–10)	6
Pasivos (7–8)	3
Detractores (0–6)	1
NPS final	50.0

En la Tabla 6.5 se resumen los resultados del cálculo realizado del *Net Promoter Score* (NPS), en donde se identificaron a **seis promotores**, **tres pasivos** y **un detractor**. El puntaje final obtenido fue de **50.0**.

6.1.3 Entrevistas estructuradas

En esta sección se presentan los resultados obtenidos a partir de las entrevistas estructuradas realizadas con los participantes.

En cuanto a los aspectos positivos, cuatro personas mencionaron la visualización de los tiempos de espera como la característica más útil, mientras que tres resaltaron la facilidad de uso y simplicidad de la interfaz. Asimismo, dos participantes resaltaron que les gustó el mapa interactivo y los íconos de las atracciones en la pantalla de inicio.

Sobre la experiencia de uso, la mayoría (7 de 10) afirmó sentirse cómoda y confiada al interactuar con la aplicación, aunque tres personas mencionaron haber tenido confusión inicial al comprender la navegación. Respecto a la utilidad, todos los participantes coincidieron en que la aplicación cumplía su propósito de informar los tiempos de espera.

Con respecto a las dificultades, dos participantes mencionaron la carga lenta del mapa, y tres expresaron la necesidad de contar con más instrucciones o claridad en ciertas funciones.

Finalmente, en las sugerencias de mejora, se propuso incluir la venta de tickets en línea, el listado de restaurantes y la opción de cambio de idioma.

6.1.4 Heurísticas de Nielsen

Esta sección presenta los resultados del análisis heurístico aplicado a la versión final de la plataforma, con base en los diez principios propuestos por Nielsen.

6.1.4.1 Visibilidad del estado del sistema

La plataforma muestra los tiempos de espera, con indicadores de carga en caso de actualización de datos (véase la Figura 6.1).



Figura 6.1: Indicadores de carga

6.1.4.2 Correspondencia entre el sistema y el mundo real

Los datos mostrados en la plataforma corresponden con la terminología oficial del IRTRA Petapa (véanse las Figuras 6.2, 6.3 y 6.4).



Figura 6.2: Correspondencia de atracciones



Figura 6.3: Correspondencia de datos



Figura 6.4: Correspondencia de ubicaciones

6.1.4.3 Control y libertad del usuario

Se implementaron opciones de navegación que permiten volver fácilmente a la pantalla principal y cancelar acciones (véase la Figura 6.5).



Figura 6.5: Navegación para regresar

6.1.4.4 Consistencia y estándares

Se respetaron los lineamientos de diseño del IRTRA Petapa (véase la Figura 6.6).

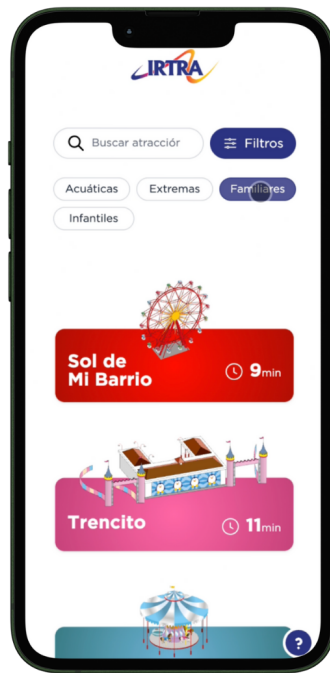


Figura 6.6: Logos, fuentes e imágenes del IRTRA Petapa

6.1.4.5 Prevención de errores

En el apartado de búsqueda y filtrado solo se muestran las atracciones que cumplen con los requisitos establecidos, evitando que el usuario seleccione opciones no disponibles (véase la Figura 6.7).



Figura 6.7: Atracciones que cumplen requisitos

6.1.4.6 Reconocimiento en lugar de recuerdo

El sistema utiliza un sistema de listas visuales, reduciendo la necesidad de memorizar pasos (véase la Figura 6.8).



Figura 6.8: Uso de listas visuales

6.1.4.7 Flexibilidad y eficiencia de uso

La navegación puede realizarse de distintas formas, ya sea desplazándose (*scroll*), utilizando la barra de búsqueda o aplicando filtros sobre las atracciones (véase la Figura 6.9).



Figura 6.9: Distintos tipos de navegación

6.1.4.8 Diseño estético y minimalista

Se eliminaron elementos gráficos innecesarios y gradientes excesivos, logrando una interfaz más clara y enfocada en la información principal (véanse las Figuras 6.10 y 6.11).



Figura 6.10: Diseño con elementos gráficos innecesarios



Figura 6.11: Producto final sin elementos gráficos innecesarios

6.1.4.9 Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de errores

La plataforma presenta mensajes claros en situaciones de error, como páginas no encontradas (404), búsquedas o filtros sin resultados y atracciones inexistentes (véanse las Figuras 6.12 y 6.13).



Figura 6.12: Búsqueda sin resultados



Figura 6.13: Página sin resultados

6.1.4.10 Ayuda y documentación

Se incluyen opciones de ayuda mediante un botón de soporte, una sección de preguntas frecuentes y un Guía de uso (*walkthrough*) inicial que orienta al usuario en las funciones principales (véanse las Figuras 6.14 y 6.15).

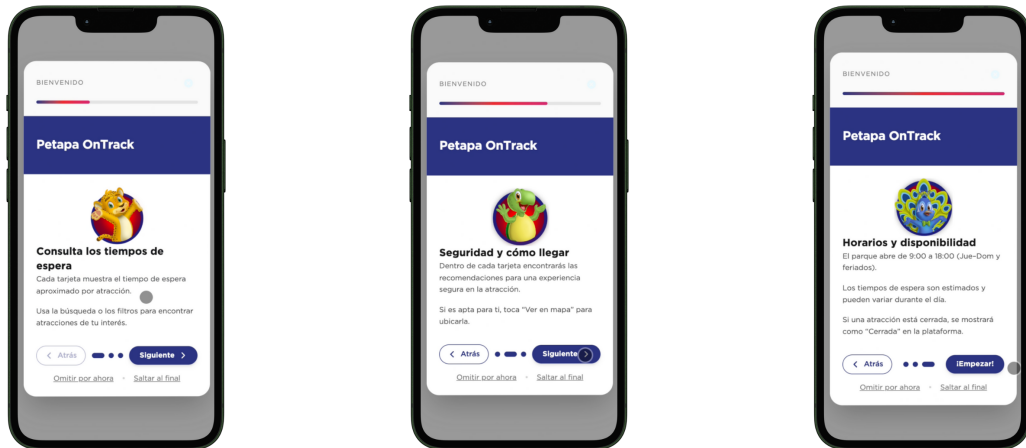


Figura 6.14: Guía de uso inicial



Figura 6.15: Soporte y preguntas frecuentes

6.2 Modelos de *machine learning*

En esta sección se presentan los resultados de los distintos modelos de *machine learning* desarrollados y entrenados de forma independiente para cada atracción, utilizando el algoritmo de *Random*

Forest. La Tabla 6.6 resume los valores de las métricas de error medio absoluto (MAE, por sus siglas en inglés), raíz del error cuadrático medio (RMSE, por sus siglas en inglés) y coeficiente de determinación (R^2) correspondientes a cada modelo.

Tabla 6.6: Resultados de los modelos de *machine learning* por atracción

Atracción	MAE	RMSE	R^2
Ballon Wheel	5.41	8.61	0.869
Bici Mágica	6.82	10.41	0.866
Brincanguro	4.86	8.94	0.844
Bumperazo	8.83	11.63	0.800
Bumpercitos	9.43	13.07	0.713
Carrusel	5.63	9.87	0.822
Casichoco	6.10	9.34	0.845
Comanche	13.84	19.15	0.755
Convoy	7.18	11.65	0.779
Dragón	6.80	10.64	0.776
Revoloteo	5.33	9.87	0.820
Faro Saltarín	7.15	10.38	0.804
Guerra Pirata	6.28	9.75	0.828
Loco Bus	6.91	11.42	0.764
Motobala	15.83	20.47	0.629
Polo Norte	7.26	10.11	0.819
Rascacielos	12.64	17.18	0.754
Ratón Loroco	16.10	23.85	0.579
Relámpago	13.66	19.59	0.709
Remolino	18.40	25.04	0.515
Samba Ballon	0.07	0.09	0.800
Sol de mi Barrio	5.39	10.06	0.889
Tifón	10.85	16.81	0.802
Trencito	8.16	12.70	0.864
Tronco Splash	14.09	21.73	0.606

Los valores de MAE se encuentran en un rango que va desde 0.07 hasta 18.40, mientras que los valores de RMSE varían entre 0.09 y 25.04. En cuanto al coeficiente de determinación (R^2), los resultados se sitúan entre 0.515 y 0.889.

6.3 Sistema *backend*

En esta sección se presentan los resultados del sistema *backend*. La Figura 6.16 muestra la arquitectura final implementada, la cual contiene una instancia AWS EC2 que aloja un PocketBase que contiene un API REST y una base de datos y *cron jobs* que ejecutan el *script* predictivo en Python. Cada tarea programada carga el `modelo (.pk1)`, lee datos históricos en `.csv` y variables externas como el clima y genera la predicción de tiempo de espera por atracción. El resultado se actualiza en PocketBase mediante solicitudes HTTP POST/PUT, mientras que el *frontend* consulta los datos a través de HTTP GET sobre la API REST usando HTTPS.

Figura 6.16: Arquitectura final del sistema *backend*

Discusión de resultados

En la presente investigación se evaluó el diseño de la interfaz de la plataforma mediante el análisis de la usabilidad, la experiencia de usuario y el grado de recomendabilidad, así como el desempeño de los modelos de *machine learning*, utilizando métricas de error medio absoluto (MAE, por sus siglas en inglés), raíz del error cuadrático medio (RMSE, por sus siglas en inglés) y coeficiente de determinación (R^2), y del sistema *backend*, mediante la verificación de la correcta integración entre los modelos de *machine learning* y la interfaz de usuario.

Se analizaron los resultados de manera cuantitativa por medio de las pruebas de usabilidad, cuestionarios y las métricas de desempeño de los modelos de *machine learning*, así como de forma cualitativa a través de las entrevistas estructuradas y la evaluación heurística.

En cuanto a los resultados de la evaluación de la usabilidad y del diseño, los datos obtenidos muestran un desempeño positivo de la plataforma, ya que de las seis tareas propuestas, **cinco** fueron completadas exitosamente por el 100 % de los participantes, mientras que la tarea tres, la cual consistía en buscar atracciones que se adaptaran a un hermano menor, presentó una tasa de completación ligeramente menor, del 90 %. Esta diferencia se debió a las dificultades que algunos usuarios experimentaron al intentar encontrar los filtros relacionados con las atracciones infantiles, lo que puede ser una oportunidad de mejora en la visibilidad y organización de los filtros en la interfaz.

Asimismo, el tiempo promedio de ejecución se mantuvo en un rango de **16.60 a 32.40** segundos. De igual forma, se registró un número mínimo de errores: dos en la tarea tres, la cual consistía en encontrar las atracciones para un hermano menor, y uno en la tarea cinco, en donde un participante no identificó inmediatamente la opción para reportar fallos del sistema. Estos errores sugieren áreas de mejora en cuanto a la visibilidad de los elementos en la interfaz. No obstante, los resultados en general reflejan que los usuarios lograron comprender la navegación y las funciones principales del sistema de forma eficiente, lo cual coincide con lo señalado por Nielsen y Budiu, quienes afirman que, al final, si los usuarios no pueden completar la tarea, todo lo demás resulta irrelevante [53].

Respecto al cuestionario *System Usability Scale* (SUS), el puntaje promedio obtenido fue de **78.25** puntos. Este valor, de acuerdo con la escala adjetiva desarrollada a partir de una amplia base de datos de resultados del SUS, se ubica dentro del rango “*Bueno*” [54]. Es importante señalar que las puntuaciones del SUS son significativas únicamente cuando se comparan con un conjunto de referencia de software similar. En este caso, la comparación se realizó con una población representativa de aplicaciones de uso general en la industria, lo que indica que los usuarios perciben la plataforma desarrollada como un sistema con una usabilidad superior al promedio [55].

Al comparar los resultados formativos obtenidos con la aplicación del SUS en la fase de investigación y diseño, es importante considerar el impacto que pudo tener la composición de los grupos de participantes. En la fase de investigación y diseño, los usuarios evaluados tenían un grado de familiaridad previa con la persona encargada de llevar a cabo las pruebas, lo cual pudo haber generado sesgos positivos asociados al deseo de complacer al facilitador. Por el contrario, en la evaluación final de la plataforma, los participantes no tenían ningún tipo de vínculo previo con la persona responsable de la prueba, reduciendo dicho efecto y favoreciendo valoraciones potencialmente más críticas [56].

Asimismo, estos resultados se relacionan con el puntaje obtenido en el *Net Promoter Score* (NPS), donde el valor final obtenido fue de **50.0**, con seis promotores, tres pasivos y un detractor. Este puntaje se considera favorable en términos de lealtad hacia la plataforma, indicando que la mayoría de los usuarios recomendarían su uso a otros visitantes, ya que un valor superior a 0, es considerado bueno y un valor mayor a 50 excepcional [57]. Esta coherencia entre las métricas de usabilidad (SUS) y de satisfacción (NPS) constituye una triangulación que respalda la validez y confiabilidad de los datos obtenidos, confirmando que la experiencia de uso fue evaluada de manera favorable [55].

De manera complementaria, la escala *UX-Lite* reportó un puntaje promedio de **78.75** sobre 100, lo que indica una experiencia de usuario positiva en términos de facilidad de uso y satisfacción general. Según Lewis y Sauro [58], las puntuaciones de *UX-Lite* pueden interpretarse mediante una escala curvada basada en percentiles, donde valores entre 70 y 79 se corresponden con una calificación de “B”, es decir, una experiencia por encima del promedio.

Por otra parte, los resultados cualitativos obtenidos en las entrevistas respaldan los hallazgos cuantitativos, al señalar que los usuarios valoraron positivamente la simplicidad visual, la coherencia del diseño y la utilidad de la aplicación.

Finalmente, la aplicación de la evaluación heurística permitió validar que la plataforma cumple con las diez heurísticas de Nielsen, evidenciando un diseño claro, coherente y centrado en la experiencia del usuario.

El **88 %** de los modelos desarrollados tuvo un desempeño satisfactorio, con valores de R^2 iguales o superiores a 0.6, lo que respalda su capacidad predictiva. Este resultado se encuentra dentro del rango aceptable propuesto por Ozili [59], quien considera valores de R^2 entre 0.50 y 0.99 como adecuados en la investigación social.

Sin embargo, las atracciones **Ratón Loroco** ($R^2 = 0.579$) y **Remolino** ($R^2 = 0.515$) presentaron un desempeño inferior. Este comportamiento puede atribuirse a la alta proporción de valores atípicos en los datos de estas atracciones (superior al 20 %), lo que incrementa el ruido en los datos y reduce la capacidad del modelo para generalizar patrones [60].

En cuanto a las métricas MAE y RMSE, en la mayoría de las atracciones, los valores de MAE se encuentran entre 5 y 9, mientras que los RMSE entre 8 y 13, siendo un nivel de error moderado. Como es esperable, el RMSE tuvo valores ligeramente mayores que el MAE, dado que penaliza con mayor severidad los errores de gran magnitud [61].

A pesar de esto, atracciones como la **Motobala**, **Ratón Loroco** y **Remolino** presentan los mayores errores, con RMSE superiores a 20. Aunque el algoritmo Random Forest es robusto ante valores atípicos y datos heterogéneos [62], la magnitud de estos errores demuestra que para estas atracciones, no hubo una identificación correcta de patrones.

Cabe mencionar que a pesar de que la mayoría de los modelos tuvieron un desempeño satisfactorio, el tiempo de espera utilizado como variable dependiente no fue un dato medido directamente, sino calculado a partir de las variables de ciclos y asistencia, lo que puede introducir cierto grado de error. De esta forma, parte de las desviaciones observadas en las métricas de error podrían atribuirse a la forma en que se estimó el tiempo de espera, más que a fallas en el modelo predictivo.

Sin embargo, a partir de la visita de campo realizada al parque, se observó que las atracciones que mostraron mayores tiempos de espera en los modelos coincidieron con aquellas que, en la práctica, presentaban filas más largas y mayor afluencia de visitantes.

Las pruebas realizadas en la plataforma confirmaron que el *backend* procesa correctamente las peticiones, almacena los resultados y los devuelve a la interfaz correctamente.

De igual forma, las predicciones realizadas por medio de la automatización con *cronjobs* se ejecutan cada hora, lo que garantiza que los datos mostrados en la interfaz se mantengan actualizados, se reduzca la intervención manual y contribuya a que el sistema sea eficiente.

Además, el uso de PocketBase como sistema de gestión de base de datos permitió una administración eficiente y segura de la información, ya que facilitó la creación de colecciones personalizadas y una API REST lista para utilizar desde el *frontend*, reduciendo significativamente el tiempo de desarrollo. Asimismo, el panel administrativo integrado, permitió supervisar las actualizaciones generadas automáticamente por los *cronjobs*, lo que permitió tener consistencia y precisión de los datos presentados en la plataforma.

Finalmente, la implementación en un entorno en la nube mediante una instancia Amazon AWS EC2 garantizó la disponibilidad continua del servicio, además de ofrecer la posibilidad de escalar los recursos de manera flexible en caso de que fuese necesario. Asimismo, se aseguró una conectividad confiable y segura entre el servidor y la base de datos mediante el uso del protocolo HTTPS y certificados SSL.

El presente trabajo tuvo como propósito desarrollar una plataforma digital para mostrar los tiempos de espera aproximados en las atracciones de IRTRA Petapa; este fue alcanzado con éxito, ya que se desarrolló una herramienta funcional que integra una interfaz, un modelo predictivo y un *backend*.

Se validó que la interfaz diseñada cumpliera con los principios de usabilidad establecidos en las heurísticas de Nielsen. Asimismo, las pruebas realizadas con usuarios mediante los instrumentos *System Usability Scale* (SUS), *Net Promoter Score* (NPS) y *UX-Lite* demostraron que los diez participantes entrevistados para la evaluación de usabilidad y diseño lograron utilizar la plataforma sin dificultad, comprender la información de la misma y valorar positivamente la experiencia, reflejando altos niveles de satisfacción y usabilidad percibida.

Se determinó que el 88% de los modelos desarrollados tuvieron un desempeño satisfactorio al estimar los tiempos de espera de los visitantes en las atracciones, con valores de R^2 iguales o superiores a 0.6. Las métricas de error mostraron valores de error medio absoluto (MAE, por sus siglas en inglés) entre 5 y 9 minutos y de raíz del error cuadrático medio (RMSE, por sus siglas en inglés) entre 8 y 13 minutos, considerados márgenes moderados y aceptables para este tipo de estimaciones. No obstante, cabe mencionar que la robustez del modelo está limitada por los datos disponibles al momento del estudio y por los cálculos realizados con base en la teoría de colas, por lo que una mayor granularidad de los datos podría mejorar aún más los resultados.

Finalmente, se comprobó que la infraestructura técnica implementada en la construcción del sistema *backend* permite la comunicación entre la interfaz y el modelo de *machine learning*. Además, la arquitectura implementada permite la escalabilidad futura, facilitando la incorporación de nuevas funcionalidades y la integración con otros servicios.

Recomendaciones

Se recomienda que el IRTRA explore la adopción de tecnologías basadas en el internet de las cosas (IoT, por sus siglas en inglés), para optimizar la recolección, el análisis de datos y la mejora de los modelos de *machine learning*.

Se sugiere implementar mecanismos de supervisión que permitan validar la estabilidad y confiabilidad de la plataforma frente a picos de demanda.

Por último, se recomienda realizar ajustes en el módulo de filtros de la interfaz con el propósito de fortalecer la usabilidad general de la plataforma y optimizar la experiencia del usuario. Asimismo, se sugiere incorporar nuevos elementos funcionales que proporcionen información relevante, como opciones de gestión en línea y accesos a servicios del parque.

Bibliografía

- [1] E. C. Daniels, J. B. Burley, T. Machemer, and P. Nieratko, “Theme park queue line perception,” *International Journal of Cultural Heritage*, vol. 2, pp. 105–118, 2017.
- [2] M. M. Davis and J. Heineke, “How disconfirmation, perception and actual waiting times impact customer satisfaction,” *international Journal of Service industry Management*, vol. 9, no. 1, pp. 64–73, 1998.
- [3] E. Bigné, L. Andreu, and J. Gnoth, “The theme park experience: An analysis of pleasure, arousal and satisfaction,” *Tourism Management*, vol. 26, no. 6, pp. 833–844, 2005.
- [4] E. Sciarretta, A. Carriero, and G. Marinensi, “Optimizing user experience in amusement parks and enhancing their active role in urban spaces through new technology,” in *Human-Computer Interaction. Design Practice in Contemporary Societies: Thematic Area, HCI 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part III 21*, pp. 270–281, Springer, 2019.
- [5] V. M. Toro, “Transformación digital de una empresa tradicional,” *Revista Sistemas*, vol. 48, no. 155, pp. 24–37, 2019.
- [6] I. Svetel, T. Kosić, and M. Pejanović, “Digital vs. traditional design process,” in *PT2018 Proceedings*, Innovation Center, University of Belgrade, 2018.
- [7] D. H. Maister, “La psicología de las colas de espera,” in *The Service Encounter* (J. Czepiel, ed.), Lexington Books, 1984. Versión en español del caso HBS 9-684-064.
- [8] D. A. Norman, “The psychology of waiting lines,” *Excerpt of*, vol. 3, 2008.
- [9] G. F. P. Siqueiros, “Introducción a la teoría de colas y su simulación,” 2013.
- [10] J. M. Pereda Ramos, “Introducción a la teoría de colas,” 2004.
- [11] D. G. Kendall, “Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain,” *The Annals of Mathematical Statistics*, pp. 338–354, 1953.
- [12] Á. D. Beriso Castillo *et al.*, “Gestión de las líneas de espera en el caso real de un centro óptico en cartagena,” 2013.
- [13] S. d. l. Fuente Fernández, “Teoría de colas.” Notas de curso, Departamento de Economía Aplicada, Universidad Autónoma de Madrid, n.d. Accedido: 2025-07-31.

- [14] L. Green, "Queueing theory and modeling," in *Handbook of Healthcare Delivery Systems* (Y. Yih, ed.), London: Taylor & Francis, 2011.
- [15] L. J. Sandoval Serrano *et al.*, "Algoritmos de aprendizaje automático para análisis y predicción de datos," *Revista Tecnológica*; no. 11, 2018.
- [16] F. Crespo, T. Alves, M. Soto, *et al.*, "Ciencia de datos, inteligencia artificial, y sus impactos sobre la sociedad," *Observatorio Económico*, no. 169, pp. 9–11, 2022.
- [17] L. S. Maté, D. O. P. Trenard, M. C. González, and I. G. Hormigo, *Introducción al ciclo de vida de los datos*. Universitat Oberta de Catalunya, 2019.
- [18] C. Miller, T. Portlock, D. M. Nyaga, and J. M. O'Sullivan, "A review of model evaluation metrics for machine learning in genetics and genomics," *Frontiers in Bioinformatics*, vol. 4, p. 1457619, 2024.
- [19] Y. Hassan Montero and F. J. Martín Fernández, "La experiencia del usuario," *No solo usabilidad*, no. 4, 2005.
- [20] Centro de Estudios de la Banca, "Políticas y mecanismos de la banca cubana: análisis y propuestas," *Revista Cubana de Información en Ciencias de la Salud*, vol. 24, no. 2, pp. 15–25, 2024. Disponible en el documento "POLITICAS BANCARIAS (PDF)".
- [21] R. Hartson and P. S. Pyla, *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier, 2012.
- [22] S. M. Labrada, "Principios del proceso de diseño de interfaz de usuario," *Revista cubana de transformación digital*, vol. 1, no. 3, pp. 143–155, 2020.
- [23] T. Schlatter and D. Levinson, *Visual usability: Principles and practices for designing digital applications*. Newnes, 2013.
- [24] L. Perurena Cancio and M. Moráquez Bergues, "Usabilidad de los sitios web, los métodos y las técnicas para la evaluación," *Revista Cubana de Información en Ciencias de la Salud (ACIMED)*, vol. 24, no. 2, pp. 176–194, 2013.
- [25] H. J. B. Ferreras, "Aplicación de la usabilidad al proceso de desarrollo de páginas web," Master's thesis, UNIVERSIDAD POLITECNICA DE MADRID, 2021. Tesis de maestría.
- [26] J. Nielsen, "Ten usability heuristics," 2005.
- [27] J. R. Lewis and J. Sauro, "The factor structure of the system usability scale," in *International conference on human centered design*, pp. 94–103, Springer, 2009.
- [28] J. Lewis and J. Sauro, "Measuring ux: From the umux-lite to the ux-lite." <https://measuringu.com/from-umux-lite-to-ux-lite/>, Sept. 2021. Consultado el 8 de septiembre de 2025.
- [29] J. Lewis, "Measuring perceived usability: Sus, umux, and csuq ratings for four everyday products," *International Journal of Human-Computer Interaction*, vol. 35, pp. 1404–1419, 01 2019.
- [30] F. F. Reichheld, "The one number you need to grow," *Harvard Business Review*, vol. 81, no. 12, pp. 46–54, 2003.
- [31] D. B. Grisaffe, "Questions about the ultimate question: Conceptual considerations in evaluating reichheld's net promoter score (nps)," *Journal of Consumer Satisfaction, Dissatisfaction and Complaining Behavior*, vol. 20, pp. 36–53, 2007.
- [32] F. Quin, D. Weyns, M. Galster, and C. C. Silva, "A/b testing: A systematic literature review," *Journal of Systems and Software*, vol. 211, p. 112011, 2024.
- [33] S. Luján-Mora, "Programación en internet: clientes web," *Programación en Internet*, 2001.

- [34] A. Chen, “Web evolution,” Master’s thesis, The University of Manchester, United Kingdom, Sept. 2008.
- [35] World Wide Web Consortium (W3C), “Web content accessibility guidelines (wcag) 2.2,” W3C Recommendation REC-WCAG22-20241212, W3C, Dec. 2024.
- [36] M. R. Martínez, R. A. Rodríguez, P. M. Vera, *et al.*, “Guía metodológica para el desarrollo de sitios web móviles adaptativos con el enfoque mobile first,” 2016.
- [37] P. Mell and T. Grance, “The nist definition of cloud computing,” Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, MD, Sept. 2011. U.S. Department of Commerce, Computer Security Division, Information Technology Laboratory.
- [38] R. Hernández-Sampieri and C. Mendoza, “Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta,” 2020.
- [39] R. Naidu-Valentine, “Triangulation in mixed method research design,” *ResearchGate*, 2024.
- [40] A. Kendrick, “Formative vs. summative evaluations,” July 2019. Consultado el 27 de septiembre de 2025.
- [41] M. J. Ritchie, K. L. Drummond, B. N. Smith, J. L. Sullivan, and S. J. Landes, “Development of a qualitative data analysis codebook informed by the i-parihs framework,” *Implementation science communications*, vol. 3, no. 1, p. 98, 2022.
- [42] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the system usability scale,” *International Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.
- [43] A. Bangor, P. T. Kortum, and J. T. Miller, “Determining what individual sus scores mean: Adding an adjective rating scale,” *Journal of Usability Studies*, vol. 4, no. 3, pp. 114–123, 2009.
- [44] J. R. Lewis, “The system usability scale: Past, present, and future,” *International Journal of Human-Computer Interaction*, vol. 34, no. 7, pp. 577–590, 2018.
- [45] R. Barbosa and A. Rojas, “Teoría de colas de espera: Modelo integral de aplicación para la toma de decisiones,” *Ingeniería y Desarrollo*, no. 1, 1995.
- [46] O. Prat Font, “Prediction of queue waiting times,” 2023. Trabajo de fin de grado en Ingeniería de Datos.
- [47] I. Beloiu and G. Szekely, “Theme park queuing systems: guest satisfaction, a comparative study,” 2018.
- [48] NVIDIA, “Dealing with outliers using three robust linear regression models.” <https://developer.nvidia.com/blog/dealing-with-outliers-using-three-robust-linear-regression-models/>, 2022. Accedido: 19 septiembre 2025.
- [49] A. Narayana Shankaranarayanan and A. B. Suresh Babu, “Digital queuing in amusement parks-possibilities and challenges,” 2020.
- [50] M. Stojiljković, “Split your dataset with scikit-learn’s train_test_split().” <https://realpython.com/train-test-split-python-data/>, January 2025. Real Python.
- [51] D. Chicco, M. Warrens, and G. Jurman, “The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation,” *PeerJ Computer Science*, vol. 7, p. e623, 07 2021.
- [52] L. Prajapati, “Day 10: Mastering react component composition for reusability — next.js,” 2025. Consultado en junio 2025.

- [53] J. Nielsen and R. Budi, "Success rate: The simplest usability metric," Feb. 2001. [Online; accessed 10-October-2025].
- [54] M. Schrepp, J. Kollmorgen, and J. Thomaschewski, "A comparison of sus, umux-lite, and ueq-s," *UXPA Journal*, pp. 86–104, 2024. [Online].
- [55] J. Sauro, "How to interpret a sus score," 2024. Consultado el 17 de octubre de 2025.
- [56] R. Molich, C. Wilson, C. M. Barnum, D. Cooley, S. Krug, C. LaRoche, B. A. Martin, J. Patrowicz, and B. Traynor, "How professionals moderate usability tests," *Journal of Usability Studies*, vol. 15, pp. 184–209, Aug. 2020.
- [57] E. B. Seufert, "Freemium metrics," in *Freemium Economics: Leveraging Analytics and User Segmentation to Drive Revenue*, pp. 83–113, Amsterdam: Elsevier, 2014.
- [58] J. R. Lewis and J. Sauro, "How to score and interpret the ux-lite," August 2024. MeasuringU. Accessed October 2025.
- [59] P. K. Ozili, "The acceptable r-square in empirical modelling for social science research," *Social Research Methodology and Publishing Results*, June 2022. Available at SSRN: <https://ssrn.com/abstract=4128165> or <http://dx.doi.org/10.2139/ssrn.4128165>.
- [60] S. Hia, H. Kuswanto, and D. Prastyo, *Robustness of Support Vector Regression and Random Forest Models: A Simulation Study*, pp. 465–479. Lecture Notes on Data Engineering and Communications Technologies, Germany: Springer Science and Business Media Deutschland GmbH, 2023. Publisher Copyright: © 2023, The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd.
- [61] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?—arguments against avoiding rmse in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.
- [62] H. P. Limbong, M. A. Lubis, and M. Furqan, "House price prediction analysis using linear regression and random forest algorithms," *Journal of Artificial Intelligence and Engineering Applications*, vol. 4, p. 1929, June 2025. Published 15th June 2025.

A.1 Encuesta exploratoria

1. En una escala del 1 al 5, ¿cómo calificaría el tiempo de espera que experimentó en las atracciones?
2. ¿Cómo se sintió respecto al tiempo de espera en las atracciones?
3. En una escala del 1 al 10, ¿qué tan satisfecho(a) estuvo con su experiencia general en el parque?
4. ¿Qué tan satisfecho(a) estuvo específicamente con la gestión de tiempos de espera?
5. ¿Qué probabilidad hay de que recomiende el parque a un amigo o familiar?
6. ¿Utilizó su teléfono móvil durante la visita al parque?
7. ¿Para qué usó su teléfono móvil durante la visita?
8. ¿Le gustaría poder consultar los tiempos de espera en las atracciones desde su teléfono?
9. ¿Qué mejoraría sobre los tiempos de espera en el parque?

A.2 Guion de entrevistas

Perfil

1. ¿Cuál es su edad y ocupación?

Experiencia general

1. ¿Ha visitado recientemente algún parque de diversiones? ¿Cuál fue el último que visitó?

2. ¿Con quién suele ir cuando los visita (familia, hijos, pareja, amigos)?
3. ¿Podría describirme una visita que recuerde especialmente, sea buena o mala?
4. ¿Ha visitado parques fuera del país? ¿Qué diferencias notó con parques nacionales?
5. ¿Ha visitado el IRTRA Petapa?
 - a) Sí
 - 1) ¿Cuándo fue la última vez que visitó el IRTRA Petapa?
 - 2) ¿Recuerda haber esperado en fila para subir a alguna atracción?
 - 3) ¿Cómo se sintió durante las esperas más largas? (aburrido, frustrado, tranquilo, etc.)
 - 4) ¿Qué hizo durante esos tiempos de espera?
 - b) No
Mismas preguntas pero con un parque que haya visitado.

Escenarios

1. Imagine que está con sus hijos/pareja/amigos haciendo fila para una atracción, y luego un empleado del parque les dice que el tiempo de espera aumentará 45 minutos.
 - a) ¿Qué haría usted en ese momento?
 - b) ¿Qué información necesitaría para tomar una mejor decisión?
2. Ahora imagine que hay otra atracción cercana con una fila más corta, pero no sabe cuánto tarda.
 - a) ¿Preferiría arriesgarse o quedarse donde está?
 - b) ¿Qué información le ayudaría a decidir?

A.3 Consentimiento informado

¿Qué harás durante esta prueba?

- Realizarás tareas específicas dentro del sistema.
- Al finalizar, se te hará una entrevista breve con preguntas sobre tu experiencia y percepción del diseño.
- La actividad tendrá una duración de entre 30 y 60 minutos.
- Se podrá grabar video y/o audio para fines de análisis, únicamente si estás de acuerdo.

Confidencialidad

- Tu identidad será protegida y los datos se utilizarán únicamente con fines académicos.
- Podrás retirarte en cualquier momento sin ninguna consecuencia.
- La participación no implica ningún riesgo ni beneficio económico.

Consentimiento para participar

1. Nombre completo
2. Correo electrónico
3. ¿Autorizas la grabación de pantalla, audio o video durante la prueba? (Sí / No)
4. Declaro que he leído y comprendido toda la información proporcionada y acepto participar voluntariamente en la prueba de usabilidad y entrevista.

A.4 Pruebas de usabilidad

1. Solo tienes 30 minutos antes de irte del parque. ¿Qué atracciones podrías visitar?
2. Tienes 2 horas. ¿Cuántas atracciones visitarías? Haz tu itinerario considerando los tiempos de espera.
3. Imagina que tienes un hermano menor al que no le gusta esperar y necesitas encontrar los juegos que se adapten a él con menor tiempo de espera.
4. Tu hermano menor mide 140cm y quiere subirse al relámpago, ¿cómo sabrías si se puede subir a dicha atracción?
5. No estás seguro en dónde se encuentra “El relámpago” en el parque, ¿cómo te guiarías por medio del sistema?

A.5 Entrevista estructurada

1. ¿Qué fue lo que más te gustó del diseño o funcionamiento de la aplicación?
2. ¿Cómo te sentiste al usar la aplicación durante la prueba?
3. ¿Consideras que la aplicación te resultó útil o resolvió la necesidad para la que fue creada?
¿Por qué?
4. ¿Hubo algo que te costó entender o que te resultó complicado durante el uso?
5. Si pudieras cambiar, agregar o eliminar algo de la aplicación, ¿qué sería?

Procesamiento y predicción de datos

B.1 Unión y enriquecimiento de datos

Listing B.1: Unión y enriquecimiento diario de datos

```
1 import pandas as pd
2 import numpy as np
3 import glob
4 import os
5
6 def cargar_con_origen(file):
7     df = pd.read_csv(file)
8     df['source_file'] = os.path.basename(file)
9     return df
10
11 file_2022 = sorted(glob.glob("./data/2022/limpio/*.csv"))
12 file_2023 = sorted(glob.glob("./data/2023/limpio/*.csv"))
13 file_2024 = sorted(glob.glob("./data/2024/limpio/*.csv"))
14
15 all_data = pd.concat([cargar_con_origen(file) for file in file_2022 + file_2023 +
16                       file_2024], ignore_index=True)
17
18 # %%
19 # put date in first column
20 all_data.insert(0, 'date', all_data['fecha'])
21 # remove 'fecha' column
22 all_data.drop(columns=['fecha'], inplace=True)
23
24 # %%
25 all_data['date'] = pd.to_datetime(all_data['date'], format='%Y-%m-%d')
26
27 all_data['day'] = all_data['date'].dt.day
28 all_data['month'] = all_data['date'].dt.month
29 all_data['year'] = all_data['date'].dt.year
30 all_data['day_of_week'] = all_data['date'].dt.day_name()
31
32 # %%
33 def es_temporada_alta(fecha):
```

```

33     """
34     Determina si una fecha está en temporada alta
35     1 = Temporada alta, 0 = Temporada normal
36     """
37     mes = fecha.month
38     dia = fecha.day
39
40     # Temporada Navideña (Octubre, Noviembre, Diciembre)
41     if mes in [11, 12]:
42         return 1
43
44     # Semana Santa ( marzo-abril)
45     if mes == 3 and dia >= 15: # Segunda quincena de marzo
46         return 1
47     if mes == 4 and dia <= 15: # Primera quincena de abril
48         return 1
49
50     # Vacaciones de medio año
51     if mes == 6 and dia >= 15: # Segunda quincena de junio
52         return 1
53
54     return 0
55
56 all_data['temporada_alta'] = all_data['date'].apply(es_temporada_alta)
57
58 print("Distribución de temporada alta:")
59 print(all_data['temporada_alta'].value_counts())
60 print(f"\nPorcentaje de temporada alta: {(all_data['temporada_alta'].sum() / len(
61     all_data) * 100) :.1f}%")
62
63 # %%
64 all_data
65
66 # %%
67 #export to csv
68 all_data.to_csv('all_data.csv', index=False)

```

B.2 Implementación de factor beta

Listing B.2: Implementación de factor beta

```

1 import argparse
2 import pandas as pd
3 import numpy as np
4 import re
5 import sys
6 from pathlib import Path
7
8 CAPACIDADES_JUEGOS = {
9     "Balloon Wheel": 24,
10    "Samba Balloon": 32,
11    "Loco Bus": 24,
12    "Dragon": 20,
13    "Comanche": 24,
14    "Ratón Loroco": 4,
15    "Remolino": 6,
16    "Rascacielos": 18,
17    "Tronco Splash": 4,
18    "Bumperazo": 60,
19    "Brincanguro": 24,
20    "Moto Bala": 12,
21    "Convoy": 20,
22    "Bumpercitos": 28,

```

```

23     "Bici_Magica": 12,
24     "Polo_Norte": 12,
25     "Sol_de_Mi_Barrío": 108,
26     "Tifón": 16,
27     "Carrusel": 52,
28     "Trencito": 74,
29     "El_Revoloteo": 24,
30     "Guerra_Pirata": 24,
31     "Faro_Saltarin": 16,
32     "El_Relámpago": 9,
33     "Casichoco": 40
34 }
35
36 def detect_hour_pairs(columns):
37     """Devuelve lista de (hora, col_asistencia, col_ciclos) presentes en el DataFrame
38     """
39     hour_re = re.compile(r"^\d{1,2}:\d{2}\s+")
40     asist = [c for c in columns if re.search(r"^\d{1,2}:\d{2}\s+asistencia$", c)]
41     cicl = [c for c in columns if re.search(r"^\d{1,2}:\d{2}\s+ciclos$", c)]
42     hours = sorted({hour_re.match(c).group(1) for c in asist + cicl if hour_re.match(
43         c)})
44     return [(h, f"{h}_asistencia", f"{h}_ciclos") for h in hours
45             if f"{h}_asistencia" in columns and f"{h}_ciclos" in columns]
46
47 def build_long(df, caps):
48     """Expande a formato largo con lambda, ciclos, mu_nom y beta_req."""
49     rows = []
50     pairs = detect_hour_pairs(df.columns)
51     for _, row in df.iterrows():
52         juego = row.get("juego", None)
53         if juego not in caps:
54             continue
55         cap_nom = caps[juego]
56         for h, a_col, c_col in pairs:
57             lam_h = row.get(a_col, np.nan)
58             ciclos = row.get(c_col, np.nan)
59             if pd.isna(lam_h) or pd.isna(ciclos):
60                 continue
61             try:
62                 lam_h = float(lam_h)
63                 ciclos = float(ciclos)
64             except Exception:
65                 continue
66             mu_nom = ciclos * cap_nom
67             if mu_nom <= 0:
68                 beta_req = np.nan if lam_h == 0 else np.inf
69             else:
70                 beta_req = lam_h / mu_nom
71             rows.append({
72                 "juego": juego,
73                 "hour": h,
74                 "lambda_pax_h": lam_h,
75                 "ciclos_h": ciclos,
76                 "cap_nom": cap_nom,
77                 "mu_nom_pax_h": mu_nom,
78                 "beta_req": beta_req
79             })
80     return pd.DataFrame(rows)
81
82 def main():
83     ap = argparse.ArgumentParser(description="Calcula percentiles de beta_req por
84     juego (p50, p75, p80, p90, p95).")
85     ap.add_argument("--in", dest="inp", default="all_data.csv", help="CSV de
86     entrada")
87     ap.add_argument("--out", dest="outp", default="betas_por_juego.csv", help="CSV de
88     salida")
89     args = ap.parse_args()

```

```

86 inp = Path(args.inp)
87 outp = Path(args.outp)
88 if not inp.exists():
89     print(f"[ERROR] No encuentro {inp.resolve()}", file=sys.stderr)
90     sys.exit(1)
91
92 df = pd.read_csv(inp)
93 long = build_long(df, CAPACIDADES_JUEGOS)
94
95 valid = long.replace([np.inf, -np.inf], np.nan).dropna(subset=["beta_req"]).copy()
96
97 if valid.empty:
98     cols = ["juego", "p50", "p75", "p80", "p90", "p95"]
99     pd.DataFrame(columns=cols).to_csv(outp, index=False)
100     print(f"[WARN] No hay observaciones válidas. Archivo vacío escrito en {outp}")
101     return
102
103 stats = (valid.groupby("juego")["beta_req"]
104         .quantile([0.50, 0.75, 0.80, 0.90, 0.95])
105         .unstack()
106         .rename(columns={0.50: "p50", 0.75: "p75", 0.80: "p80", 0.90: "p90",
107                        0.95: "p95"})
108         .reset_index())
109
110 cols = ["juego", "p50", "p75", "p80", "p90", "p95"]
111 stats[cols].sort_values("juego").to_csv(outp, index=False)
112 print(f"Betas guardados en {outp}")
113
114 if __name__ == "__main__":
115     main()

```

B.3 Implementación modelo M|M|1

Listing B.3: Implementación modelo M|M|1

```

1 import numpy as np
2 import pandas as pd
3
4 HORAS = [f"{h}:00" for h in range(9, 19)] # 9..18
5
6 def mm1_perhour(df, capacidad_por_ciclo, max_espera_min=60, juego_name=None):
7     df = df.copy()
8     beta = get_beta(juego_name) if juego_name is not None else 0.80
9
10    for h in HORAS:
11        col_cic = f"{h}_ciclos"
12        col_asist = f"{h}_asistencia"
13        if col_cic not in df.columns or col_asist not in df.columns:
14            continue
15
16        lam = df[col_asist].astype(float).to_numpy() / 60.0
17        mu = (df[col_cic].astype(float).to_numpy() * (float(capacidad_por_ciclo) *
18            beta)) / 60.0
19
20        rho = np.full(len(df), np.nan)
21        Wq = np.full(len(df), np.nan)
22        state = np.array([""] * len(df), dtype=object)
23
24        closed = (mu == 0) & (lam == 0)
25        off_service = (mu == 0) & (lam > 0)
26        stable = (mu > lam) & (mu > 0)
27        saturated = (mu <= lam) & (mu > 0)

```

```

27
28     rho[closed] = 0
29     Wq[closed] = 0
30     state[closed] = "Closed"
31
32     rho[off_service] = np.inf
33     Wq[off_service] = max_espera_min
34     state[off_service] = "Off_service"
35
36     rho[stable] = lam[stable] / mu[stable]
37     Wq[stable] = lam[stable] / (mu[stable] * (mu[stable] - lam[stable]))
38     state[stable] = "stable"
39
40     rho[saturated] = lam[saturated] / mu[saturated]
41     Wq[saturated] = max_espera_min
42     state[saturated] = "saturated"
43
44     df[f"lambda_{h}/(min)"] = lam.round(4)
45     df[f"mu_{h}/(min)"] = mu.round(4)
46     df[f"rho_{h}"] = np.clip(rho, None, 10)
47     df[f"Wq_{h}(min)"] = np.minimum(Wq, max_espera_min)
48     df[f"state_{h}"] = state
49     return df

```

B.4 Asistencia total por año

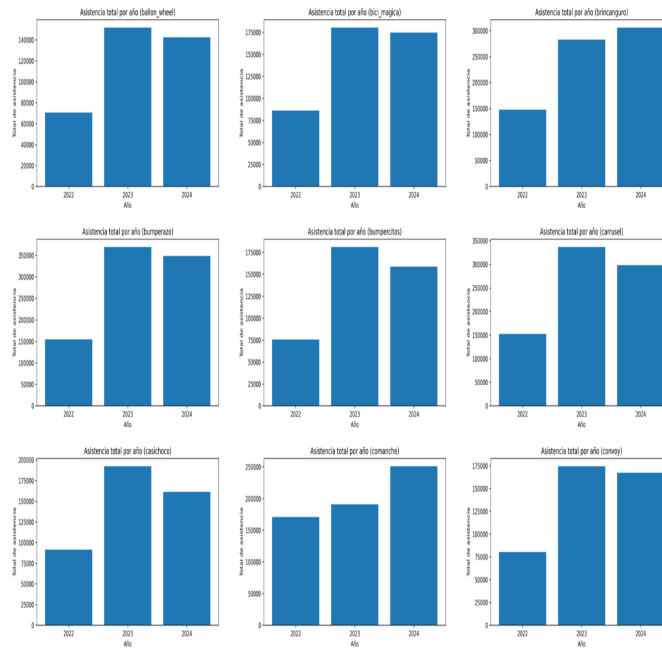


Figura B.1: Asistencia total por año (Parte 1)

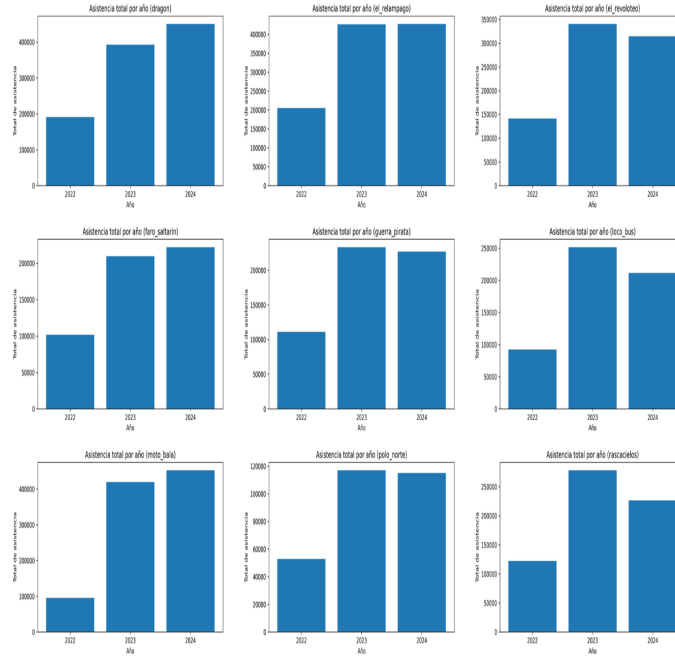


Figura B.2: Asistencia total por año (Parte 2)

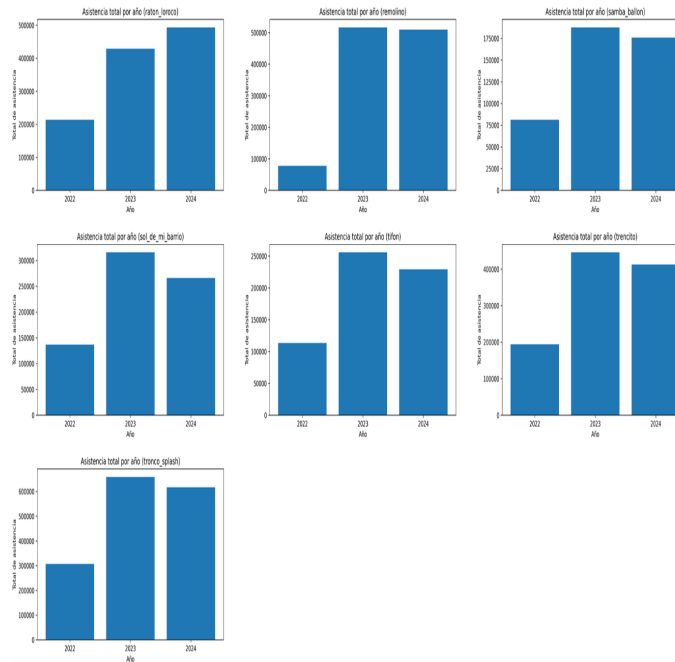


Figura B.3: Asistencia total por año (Parte 3)

B.5 Asistencia por día

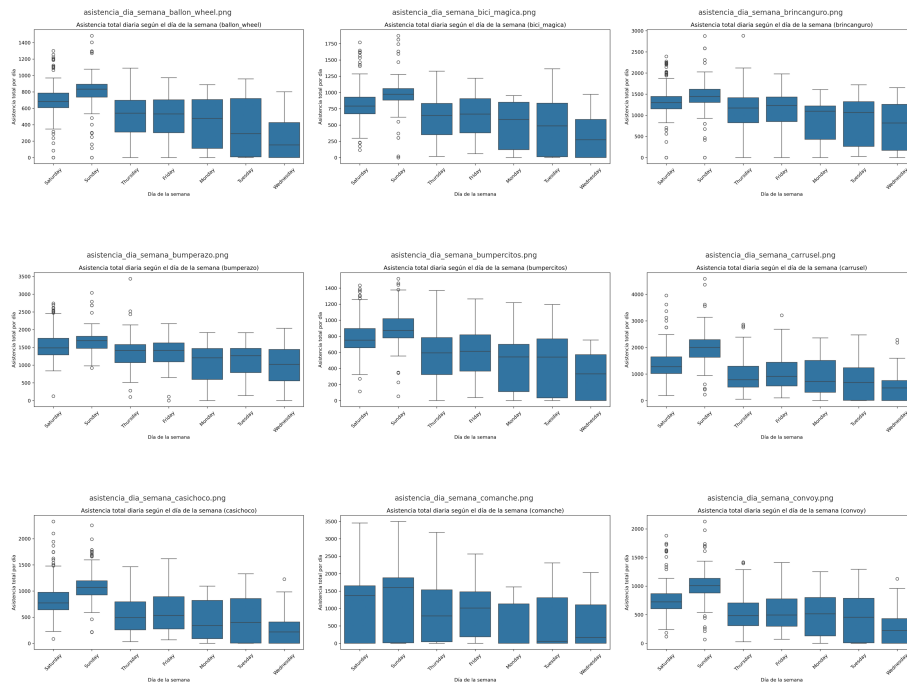


Figura B.4: Asistencia por día (Parte 1)

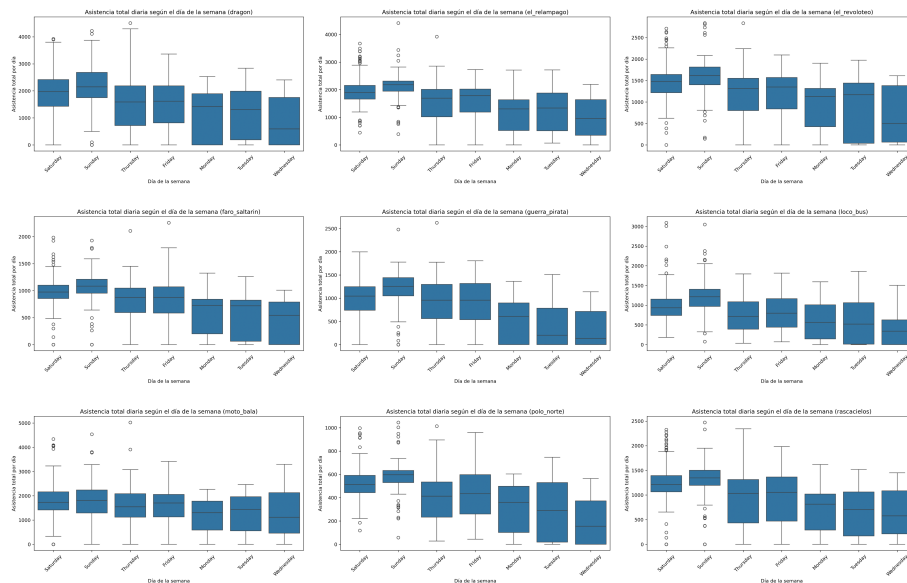


Figura B.5: Asistencia por día (Parte 2)

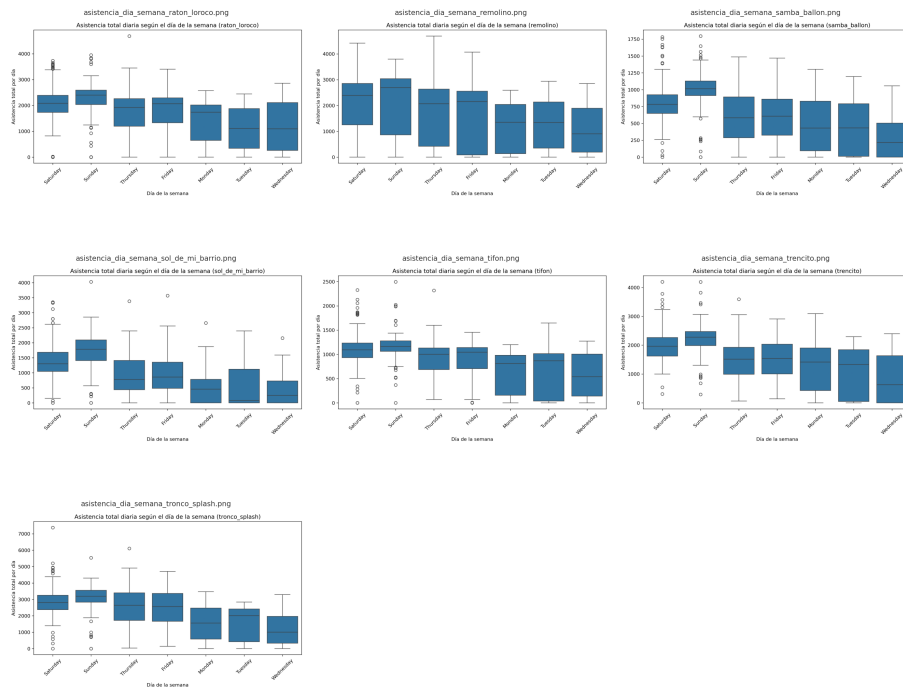


Figura B.6: Asistencia por día (Parte 3)

Despliegue e infraestructura

C.1 Configuración inicial EC2

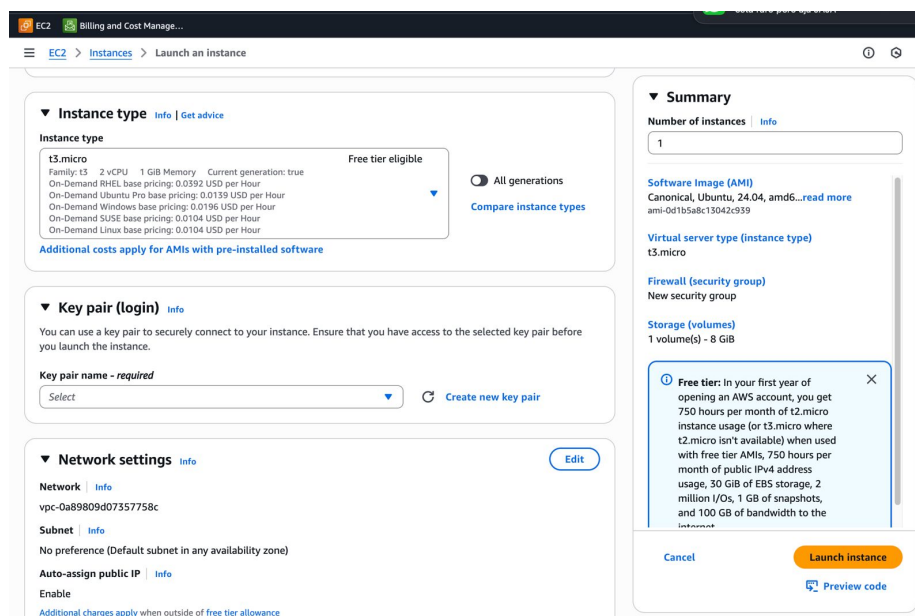


Figura C.1: Configuración inicial de máquina en Amazon EC2

C.2 Configuración de IP elástica

EC2 > Elastic IP addresses > 18.189.150.57 > Associate Elastic IP address

Associate Elastic IP address
Choose the instance or network interface to associate to this Elastic IP address (18.189.150.57)

Elastic IP address: 18.189.150.57

Resource type
Choose the type of resource with which to associate the Elastic IP address.

Instance
 Network interface

Warning: If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. [Learn more](#)

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

Instance
i-0b105c0539fbd3c3

Private IP address
The private IP address with which to associate the Elastic IP address.
172.31.34.96

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
 Allow this Elastic IP address to be reassociated

Cancel Associate

Figura C.2: Configuración de IP elástica

C.3 Conexión segura entre el servidor y PocketBase

Listing C.1: Nginx como reverse proxy para PocketBase con TLS

```

1 server {
2     server_name petapaontrack.kojimena.com;
3
4     client_max_body_size 20m;
5
6     location / {
7         proxy_pass          http://127.0.0.1:8090;
8         proxy_http_version 1.1;
9         proxy_set_header    Host                $host;
10        proxy_set_header    X-Real-IP          $remote_addr;
11        proxy_set_header    X-Forwarded-For    $proxy_add_x_forwarded_for;
12        proxy_set_header    X-Forwarded-Proto $scheme;
13        proxy_set_header    Connection        "";
14        proxy_read_timeout   360s;
15    }
16
17    # TLS gestionado por Certbot
18    listen 443 ssl;                                # managed by Certbot
19    listen [::]:443 ssl ipv6only=on;                # managed by Certbot
20    ssl_certificate      /etc/letsencrypt/live/petapaontrack.kojimena.com/fullchain.
21    pem; # managed by Certbot
22    ssl_certificate_key  /etc/letsencrypt/live/petapaontrack.kojimena.com/privkey.pem;
23    # managed by Certbot
24    include              /etc/letsencrypt/options-ssl-nginx.conf;
25    # managed by Certbot
26    ssl_dhparam          /etc/letsencrypt/ssl-dhparams.pem;
27    # managed by Certbot

```

```

24 }
25
26 # Redireccion HTTP -> HTTPS y manejo 404
27 server {
28     if ($host = petapaontrack.kojimena.com) {
29         return 301 https://$host$request_uri; # managed by Certbot
30     }
31
32     listen 80;
33     listen [::]:80;
34     server_name petapaontrack.kojimena.com;
35     return 404; # managed by Certbot
36 }

```

C.4 Transferencia de modelos y *datasets* al servidor

```

(base) jime@MacBook-Pro-de-Jime-2 petapaontrack_ds % scp -i /Users/jime/10mo semestre/graduacion/petapaontrack.pem' data_
analysis/models/*.joblib ubuntu@18.189.150.57:~/predicciones/data_analysis/models/

ballon_wheel.joblib          100% 59MB  5.5MB/s  00:10
bici_magica.joblib          100% 56MB  5.1MB/s  00:11
brincanguero.joblib        100% 51MB  5.9MB/s  00:08
bumperazo.joblib           100% 60MB  5.7MB/s  00:10
bumpercitos.joblib         100% 62MB  5.6MB/s  00:11
carrusel.joblib            100% 58MB  5.5MB/s  00:10
casichoco.joblib           100% 59MB  5.8MB/s  00:10
comanche.joblib            100% 43MB  6.8MB/s  00:06
convoy.joblib               0% 255KB 228.4KB/s 04:35 ETA

```

Figura C.3: Transferencia de modelos entrenados al servidor

```

(base) jime@MacBook-Pro-de-Jime-2 petapaontrack_ds % scp -i /Users/jime/10mo semestre/graduacion/petapaontrack.pem' data_
by_game/*.csv ubuntu@18.189.150.57:~/predicciones/data/by_game/

ballon_wheel.csv           100% 400KB 874.9KB/s 00:00
bici_magica.csv            100% 405KB  1.4MB/s  00:00
brincanguero.csv          100% 401KB  1.7MB/s  00:00
bumperazo.csv             100% 407KB  1.7MB/s  00:00
bumpercitos.csv           100% 427KB  1.8MB/s  00:00
carrusel.csv              100% 436KB  1.8MB/s  00:00
casichoco.csv             100% 403KB  1.8MB/s  00:00
comanche.csv              100% 354KB  1.6MB/s  00:00
convoy.csv                100% 416KB  1.8MB/s  00:00
dragon.csv                 100% 381KB  1.9MB/s  00:00
el_relampago.csv          100% 402KB 908.0KB/s 00:00
el_revoloteo.csv          100% 400KB  1.3MB/s  00:00
faro_saltarin.csv         100% 402KB  2.0MB/s  00:00
guerra_pirata.csv         100% 394KB  1.6MB/s  00:00

```

Figura C.4: Carga de *datasets*

C.5 Integración con PocketBase

```
GNU nano 7.2 /home/ubuntu/predicciones/scripts/pb_helpers.py *
import os, json, urllib.parse, requests
from dotenv import load_dotenv

load_dotenv()
PB_URL = (os.getenv("PB_URL") or "").rstrip("/")
PB_EMAIL = os.getenv("PB_EMAIL")
PB_PASSWORD = os.getenv("PB_PASSWORD")
PB_AUTH_TYPE = os.getenv("PB_AUTH_TYPE", "admin") # admin | user
PB_GAMES_KEY_FIELD = os.getenv("PB_GAMES_KEY_FIELD", "name")
PB_TIME_MODE = os.getenv("PB_TIME_MODE", "numeric") # numeric | map (no usaremos map aquí)

class PocketBaseClient:
    def __init__(self):
        if not PB_URL or not PB_EMAIL or not PB_PASSWORD:
            raise RuntimeError("Faltan PB_URL / PB_EMAIL / PB_PASSWORD en .env")
        self.token = None

    def _headers(self):
        if not self.token:
            self.login()
        return {"Authorization": f"Bearer {self.token}"}

    def login(self):
        if PB_AUTH_TYPE == "admin":
            ep = "/api/admins/auth-with-password"
        else:
            ep = "/api/collections/users/auth-with-password"
        url = f"{PB_URL}{ep}"
        r = requests.post(url, json={"identity": PB_EMAIL, "password": PB_PASSWORD}, timeout=20)
        r.raise_for_status()
        self.token = r.json()["token"]

    def get_game_by_key(self, key: str):
        filt = f'{"PB_GAMES_KEY_FIELD"}={key}'
        url = f"{PB_URL}/api/collections/games/records?filter={urllib.parse.quote(filt)}&perPage=1&page=1"
        r = requests.get(url, headers=self._headers(), timeout=20)
        r.raise_for_status()
        items = r.json().get("items", [])
        if not items:
```

Figura C.5: Script utilizado para la integración con PocketBase

C.6 Configuración de *cronjobs*

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 * * * * /home/ubuntu/predicciones/scripts/run_rascacielos.sh
1 * * * * /home/ubuntu/predicciones/scripts/run_ballon_wheel.sh
2 * * * * /home/ubuntu/predicciones/scripts/run_bici_magica.sh
3 * * * * /home/ubuntu/predicciones/scripts/run_brincanguro.sh
4 * * * * /home/ubuntu/predicciones/scripts/run_bumperazo.sh
5 * * * * /home/ubuntu/predicciones/scripts/run_bumpercitos.sh
6 * * * * /home/ubuntu/predicciones/scripts/run_carrusel.sh
7 * * * * /home/ubuntu/predicciones/scripts/run_casichoco.sh
8 * * * * /home/ubuntu/predicciones/scripts/run_comanche.sh
9  * * * * /home/ubuntu/predicciones/scripts/run_dragon.sh
10 * * * * /home/ubuntu/predicciones/scripts/run_relampago.sh
11 * * * * /home/ubuntu/predicciones/scripts/run_el_revoloteo.sh
12 * * * * /home/ubuntu/predicciones/scripts/run_faro_saltarin.sh
13 * * * * /home/ubuntu/predicciones/scripts/run_guerra_pirata.sh
14 * * * * /home/ubuntu/predicciones/scripts/run_loco_bus.sh
15 * * * * /home/ubuntu/predicciones/scripts/run_moto_bala.sh
16 * * * * /home/ubuntu/predicciones/scripts/run_polo_norte.sh
17 * * * * /home/ubuntu/predicciones/scripts/run_raton_loroco.sh
18 * * * * /home/ubuntu/predicciones/scripts/run_remolino.sh
19 * * * * /home/ubuntu/predicciones/scripts/run_samba_ballon.sh
20 * * * * /home/ubuntu/predicciones/scripts/run_sol_de_mi_barrio.sh
21 * * * * /home/ubuntu/predicciones/scripts/run_tifon.sh
22 * * * * /home/ubuntu/predicciones/scripts/run_trencito.sh
23 * * * * /home/ubuntu/predicciones/scripts/run_tronco_splash.sh
24 * * * * /home/ubuntu/predicciones/scripts/run_convoy.sh
```

Figura C.6: Configuración de *cronjobs* en el servidor EC2

D.1 Datos demográficos y consentimiento informado

¿Qué harás durante esta prueba?

- Realizarás tareas específicas dentro del sistema.
- Al finalizar, se te hará una entrevista breve con preguntas sobre tu experiencia y percepción del diseño.
- La actividad tendrá una duración de entre 30 y 60 minutos.
- Se podrá grabar video y/o audio para fines de análisis, únicamente si estás de acuerdo.

Confidencialidad

- Tu identidad será protegida y los datos se utilizarán únicamente con fines académicos.
- Podrás retirarte en cualquier momento sin ninguna consecuencia.
- La participación no implica ningún riesgo ni beneficio económico.

Consentimiento para participar

1. Nombre completo
2. Edad
3. Género

4. Nivel educativo
5. Ocupación
6. Correo electrónico
7. ¿Autorizas la grabación de pantalla, audio o video durante la prueba? (Sí / No)
8. Declaro que he leído y comprendido toda la información proporcionada y acepto participar voluntariamente en la prueba de usabilidad y entrevista.

D.2 Evaluación de usabilidad – SUS, *UX-Lite*, NPS

System Usability Scale (SUS)

1. Creo que me gustaría usar este sistema con frecuencia. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
2. Encontré el sistema innecesariamente complejo. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
3. Me pareció que el sistema era fácil de usar. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
4. Creo que necesitaría la ayuda de una persona con conocimientos técnicos para poder usar este sistema. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
5. Considero que las distintas funciones del sistema estaban bien integradas. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
6. Me pareció que había demasiada inconsistencia en el sistema. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
7. Imagino que la mayoría de las personas aprenderían a utilizar este sistema rápidamente. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
8. Encontré el sistema muy engorroso de usar. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
9. Me sentí muy seguro/a usando el sistema. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
10. Necesité aprender muchas cosas antes de poder utilizar el sistema. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)

UX-Lite

1. Esta plataforma es fácil de usar. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)
2. Las funciones de esta plataforma satisfacen mis necesidades. Escala: 1 (Totalmente en desacuerdo) – 5 (Totalmente de acuerdo)

Net Promoter Score (NPS)

- En una escala de 0 a 10, ¿qué tan probable es que recomiendes esta plataforma a un amigo o familiar? Escala: 0 (Nada probable) – 10 (Muy probable)

D.3 Entrevista estructurada

1. ¿Qué fue lo que más te gustó del diseño o funcionamiento de la aplicación?
2. ¿Cómo te sentiste al usar la aplicación durante la prueba?
3. ¿Consideras que la aplicación te resultó útil o resolvió la necesidad para la que fue creada?
¿Por qué?
4. ¿Hubo algo que te costó entender o que te resultó complicado durante el uso?
5. Si pudieras cambiar, agregar o eliminar algo de la aplicación, ¿qué sería?