



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

IMPLEMENTACIÓN DE UN SISTEMA EXPERTO
PARA LA SELECCIÓN ESTRATÉGICA DE
CAMPEONES EN LEAGUE OF LEGENDS

T E S I S

QUE PARA OPTAR POR EL GRADO DE:
Licenciado en Ciencias Computacionales

PRESENTA:

Gustavo Olguín Chávez

ASESORES:

Dra. Anilu Franco Arcega

Dr. Luis Heriberto García Islas



Ciudad del Conocimiento, Mineral de la Reforma, 2025



Mineral de la Reforma, Hgo., a 7 de mayo de 2025

Número de control: ICBI-D/752/2025
 Asunto: Autorización de impresión.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Con Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio, le comunico que el Jurado asignado al egresado de la Licenciatura en Ciencias Computaciones **Gustavo Olguín Chávez**, quien presenta el trabajo de titulación **“Implementación de un sistema experto para la selección estratégica de campeones en League of Legends”**, ha decidido, después de revisar fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; dicho trabajo en la reunión de sinodales, **autorizar la impresión del mismo**, una vez realizadas las correcciones acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente: Dr. Manuel Alejandro Ojeda Misses

Secretario: Dr. Aldo Márquez Grajales

Vocal: Dra. Anilú Franco Árcega

Suplente: Dr. Luis Heriberto García Islas

Sin otro particular por el momento, reciba un cordial saludo.

Atentamente
 “Amor, Orden y Progreso”

Mtro. Gabriel Vergara Rodríguez
 Director del ICBI



GVR/YCC

Ciudad del Conocimiento, Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
 Teléfono: 771 71 720 00 Ext. 40001
 direccion_icbi@uaeh.edu.mx,
 vergarar@uaeh.edu.mx



Agradezco profundamente a mi madre, a mis abuelos y a mis amigos por su amor, sus enseñanzas y su apoyo incondicional. Han sido siempre mi mayor ejemplo y una compañía invaluable en cada paso del camino. También quiero agradecer a KOI, equipo de esports que me transmitió la pasión por este mundo, pues gracias a ellos nació en mí el interés que inspiró este trabajo académico y que ha llenado mi vida de entusiasmo y motivación.



Resumen

Esta investigación aborda el desarrollo de un sistema experto basado en técnicas de minería de datos mediante la obtención de reglas de asociación, enfocado en el videojuego competitivo League of Legends (LoL). El proyecto surge como una necesidad de proporcionar herramientas de apoyo a jugadores casuales, facilitando la toma de decisiones durante la selección de campeones, runas e ítems.

La investigación se desarrolló mediante la integración de la metodología en cascada junto con Knowledge Discovery in Databases (KDD), lo que permitió estructurar tanto el desarrollo del sistema web como el análisis de datos. Para la implementación se utilizaron tecnologías como React en el frontend, Django en el backend y PostgreSQL como gestor de base de datos, además de herramientas especializadas para el análisis de datos como Pandas, NumPy y Power BI.

El sistema se implementó en un ambiente web que permite a los usuarios, mediante un sencillo formulario, obtener recomendaciones personalizadas de campeones, junto con sugerencias de runas e ítems óptimos para cada selección. La plataforma está diseñada para proporcionar información relevante y útil durante la fase de selección de campeones donde el tiempo es limitado.

La validación del sistema mediante la escala SUS (System Usability Scale) permitió obtener un promedio total de 68.25, lo que lo clasifica como un sistema “Acceptable”. Por otra parte las pruebas de validación a las recomendaciones del sistema, permitieron obtener resultados favorables, al recibir un 80 % de aprobación por parte de jugadores de League of Legends, que realizaron pruebas con las sugerencias del sistema en partidas, por ultimo la validación de las reglas obtenidas mediante las métricas de el algoritmo Apriori, demuestra que las reglas tienen un alto nivel de efectividad, al tener una probabilidad del 85 % de cumplirse, lo que les permite obtener una alta confianza de poder cumplir con lo requerido. Se identificó como área de mejora la percepción de los usuarios sobre la veracidad de las recomendaciones, por lo que es necesario optimizar los algoritmos de análisis de datos, mejorar la base de conocimientos e implementar actualizaciones constantes. Este proyecto sienta las bases para futuras investigaciones en la aplicación de técnicas de minería de datos en el ámbito de los videojuegos.

Abstract

This research addresses the development of an expert system based on data mining techniques through the extraction of association rules, focused on the competitive video game League of Legends (LoL). The project arises from the need to provide support tools for casual players, facilitating decision-making during the selection of champions, runes, and items.

The research was carried out through the integration of the waterfall methodology along with Knowledge Discovery in Databases (KDD), which made it possible to structure both the development of the web system and the data analysis. For the implementation, technologies such as React on the frontend, Django on the backend, and PostgreSQL as the database manager were used, in addition to specialized tools for data analysis such as Pandas, NumPy, and Power BI.

The system was implemented in a web environment that allows users, through a simple form, to obtain personalized champion recommendations, along with suggestions for optimal runes and items for each selection. The platform is designed to provide relevant and useful information during the champion selection phase, where time is limited.

The validation of the system using the SUS (System Usability Scale) yielded an overall average of 68.25, which classifies it as an “Acceptable” system. On the other hand, validation tests of the system’s recommendations produced favorable results, receiving 80 % approval from League of Legends players who tested the system’s suggestions in matches. Lastly, the validation of the rules obtained through the metrics of the Apriori algorithm demonstrates that the rules have a high level of effectiveness, with an 85 % probability of being fulfilled, which allows them to obtain a high degree of confidence in meeting the requirements. An area for improvement was identified in users’ perception of the accuracy of the recommendations, so it is necessary to optimize the data analysis algorithms, improve the knowledge base, and implement constant updates. This project lays the groundwork for future research into the application of data mining techniques in the field of video games.

Índice general

| | |
|---------------------------------------------------------------|-----------|
| Índice de figuras | xI |
| Índice de tablas | xv |
| Introducción | 1 |
| Planteamiento del problema | 2 |
| Justificación | 3 |
| Propuesta de solución | 4 |
| Objetivos | 4 |
| Delimitaciones y limitaciones | 5 |
| 1. Marco teórico | 7 |
| 1.1. Sistemas | 7 |
| 1.1.1. Sistemas expertos | 8 |
| 1.1.2. Componentes de un sistema experto | 8 |
| 1.1.3. Tipos de sistemas expertos | 11 |
| 1.1.4. Etapas de desarrollo de un sistema experto | 12 |
| 1.1.5. Aplicaciones y ejemplos de sistemas expertos | 13 |
| 1.2. Minería de datos | 14 |
| 1.2.1. Modelos (Predictivos y descriptivos) | 15 |
| 1.2.1.1. Modelos predictivos | 16 |
| 1.2.1.2. Modelos descriptivos | 17 |
| 1.2.2. Reglas de asociación | 18 |
| 1.2.2.1. Métricas clave | 18 |
| 1.2.3. Algoritmos principales | 20 |
| 1.2.3.1. Apriori | 20 |
| 1.2.3.2. FP-Growth | 20 |
| 1.2.4. ECLAT | 21 |
| 1.2.5. Discusión | 21 |
| 1.3. Área de aplicación | 21 |
| 1.3.1. Mecánicas básicas del juego | 22 |
| 1.3.2. Progresión y economía dentro del juego | 26 |
| 1.3.3. ¿Cómo empezar a jugar? | 29 |

| | |
|----------------------------------------------------------------------------------|-----------|
| 2. Marco metodológico | 35 |
| 2.1. Metodologías de desarrollo | 35 |
| 2.1.1. Metodologías tradicionales | 35 |
| 2.1.1.1. Modelo en cascada (Waterfall) | 36 |
| 2.1.1.2. Modelo en espiral | 37 |
| 2.1.2. Metodologías Ágiles | 38 |
| 2.1.2.1. SCRUM | 39 |
| 2.1.2.2. Kanban | 40 |
| 2.2. Metodologías de análisis de datos | 42 |
| 2.2.1. Metodología KDD | 42 |
| 2.2.2. Metodología SEMMA | 44 |
| 2.2.3. Discusión | 46 |
| 3. Marco tecnológico | 47 |
| 3.1. Herramientas de desarrollo para la página web | 48 |
| 3.1.1. Diseño y prototipado - Figma | 48 |
| 3.1.2. Desarrollo del Frontend - React | 49 |
| 3.1.2.1. Requerimientos | 49 |
| 3.1.3. Desarrollo del Backend - Django | 50 |
| 3.1.3.1. Arquitectura Modelo-Vista-Template (MVT) | 50 |
| 3.1.3.2. Otros módulos de Django | 51 |
| 3.2. Herramientas de desarrollo para la base de datos | 51 |
| 3.2.1. PostgreSQL | 52 |
| 3.2.2. Arquitectura | 52 |
| 3.3. Herramientas de desarrollo para el análisis de datos | 54 |
| 3.3.1. Librerías de Python para análisis de datos | 54 |
| 3.3.1.1. Librería requests | 54 |
| 3.3.1.2. Librería Pandas | 55 |
| 3.3.1.3. Librería pyarrow | 57 |
| 3.3.1.4. NumPy | 58 |
| 3.3.1.5. Matplotlib | 59 |
| 3.3.2. Data Warehousing | 61 |
| 3.3.2.1. Dremio | 61 |
| 3.3.3. Visualización de Datos | 63 |
| 3.3.3.1. Power BI | 63 |
| 3.3.4. Técnicas de Minería de Datos | 64 |
| 3.3.4.1. Biblioteca Apyori | 64 |
| 3.3.4.2. Biblioteca Scikit-learn | 64 |
| 4. Estado del arte | 67 |
| 4.1. eSports, un enfoque basado en inteligencia artificial | 67 |
| 4.2. En busca de patrones de comportamiento entre jugadores de League of Legends | 69 |

| | | |
|-----------|-------------------------------------------------------------------------------------------------------------------|------------|
| 4.3. | Sistema de recomendación de League of Legends mediante agrupación y redes de neuronas | 71 |
| 4.4. | Data mining techniques in video games | 72 |
| 4.5. | Diseño de una herramienta de análisis de rendimiento de jugadores para equipos de alto nivel de eSports | 74 |
| 4.6. | Discusión | 75 |
| 5. | Aplicación de metodologías de desarrollo | 79 |
| 5.1. | Análisis de requisitos | 79 |
| 5.1.1. | Alcance | 79 |
| 5.1.2. | Problemáticas actuales | 80 |
| 5.1.3. | Requisitos funcionales | 80 |
| 5.1.4. | Requisitos no funcionales | 80 |
| 5.1.5. | Restricciones | 81 |
| 5.1.6. | Metas del sistema | 81 |
| 5.2. | Diseño | 82 |
| 5.2.1. | Diseño de la arquitectura del sistema | 82 |
| 5.2.2. | Diseño de la base de datos | 83 |
| 5.2.3. | Diseño de la interfaz de usuario (UI) | 84 |
| 5.3. | Implementación | 86 |
| 5.3.1. | Selección | 89 |
| 5.3.2. | Preprocesamiento y limpieza | 89 |
| 5.3.3. | Transformación y reducción | 90 |
| 5.3.4. | Minería de datos | 90 |
| 5.3.4.1. | Visualización mediante Dremio y Power BI | 90 |
| 5.3.4.2. | Reglas de asociación | 95 |
| 5.3.5. | Interpretación y evaluación | 97 |
| 5.3.6. | Implementación de la página web | 97 |
| 5.3.6.1. | Implementación de la base de datos | 98 |
| 5.3.6.2. | Implementación del Backend | 99 |
| 5.3.6.3. | Implementación del Frontend | 101 |
| 5.4. | Verificación | 102 |
| 5.4.1. | Verificación de la API | 102 |
| 5.4.2. | Verificación de la integración con el frontend | 102 |
| 5.4.3. | Pruebas de funcionamiento de las recomendaciones | 106 |
| 6. | Resultados y validación | 115 |
| 6.1. | Validación SUS (System Usability Scale) | 115 |
| 6.1.1. | Descripción de la metodología | 115 |
| 6.1.1.1. | Cuestionario | 116 |
| 6.1.1.2. | Obtención de resultados | 117 |
| 6.1.2. | Descripción de la muestra | 118 |
| 6.1.2.1. | Relación con los videojuegos | 118 |
| 6.1.2.2. | Experiencia con League of Legends | 120 |

ÍNDICE GENERAL

| | | |
|---------------------|-----------------------------------------------------------|------------|
| 6.1.3. | Aplicación del cuestionario | 122 |
| 6.1.4. | Resultados y análisis | 130 |
| 6.1.4.1. | Resultado de validación | 131 |
| 6.2. | Validación de la efectividad de las sugerencias | 131 |
| 6.2.1. | Validación cualitativa | 131 |
| 6.2.1.1. | Cuestionario | 132 |
| 6.2.1.2. | Descripción de la muestra | 133 |
| 6.2.1.3. | Aplicación del cuestionario | 134 |
| 6.2.1.4. | Resultados de la validación | 139 |
| 6.2.2. | Validación cuantitativa | 140 |
| 6.3. | Discusión | 141 |
| Conclusiones | | 143 |
| A. Anexos | | 147 |
| A.1. | Manual de Usuario | 147 |
| A.1.1. | Alcance del sistema | 147 |
| A.1.2. | Audiencia | 148 |
| A.1.3. | Requisitos del sistema | 148 |
| A.1.4. | Módulos Principales | 149 |
| A.1.4.1. | Landing Page | 149 |
| A.1.4.2. | Cuestionario de características de jugador | 151 |
| A.1.4.3. | Sección de recomendaciones | 154 |
| A.1.4.4. | Detalles sobre la recomendación | 155 |
| A.1.5. | Solución de problemas | 156 |
| A.1.5.1. | Errores comunes | 156 |
| A.1.5.2. | Preguntas frecuentes (FAQ) | 157 |
| A.1.5.3. | Contacto de soporte | 157 |
| Referencias | | 159 |

Índice de figuras

| | |
|---------------------------------------------------------------------------------------------|-----|
| 1.1. Estructura de un sistema experto | 9 |
| 1.2. Ciclo de desarrollo de un sistema experto | 13 |
| 1.3. Zonas del mapa y principales estructuras | 23 |
| 1.4. Mapa de League of Legends | 25 |
| 1.5. Distribución del oro otorgado por diferentes tipos de súbditos. | 27 |
| 1.6. Distribución del oro otorgado por los monstruos de la jungla. | 28 |
| 1.7. Cliente del LoL | 30 |
| 1.8. Sala de espera para jugar una partida | 31 |
| 1.9. Menú de selección de campeones | 32 |
| 1.10. Selección de runas para el personaje | 33 |
| 2.1. Esquema del modelo en cascada para el desarrollo de software. | 36 |
| 2.2. Estructura del modelo Espiral | 38 |
| 2.3. Ciclo de desarrollo de la metodología Scrum. | 39 |
| 2.4. Representación de la metodología Kanban | 41 |
| 2.5. Etapas de la metodología KDD | 43 |
| 2.6. Fases de la metodología SEMMA | 45 |
| 3.1. Arquitectura de un servidor PostgreSQL | 52 |
| 3.2. Gráfico creado mediante matplotlib | 61 |
| 5.1. Diagrama Entidad-Relación de la base de datos. | 84 |
| 5.2. Diseño inicial de Home Page. | 85 |
| 5.3. Diseño inicial del formulario. | 87 |
| 5.4. Diseño inicial de sugerencias. | 88 |
| 5.5. Diseño inicial de detalles y estadísticas. | 88 |
| 5.6. Nueva estructura del conjunto de datos. | 91 |
| 5.7. Desempeño de los campeones según la dificultad y rol. | 92 |
| 5.8. Desempeño de los campeones que ganaron una partida. | 93 |
| 5.9. Runas más usadas por campeón, según el rol. | 94 |
| 5.10. Ítems más usados por campeón, según el rol. | 95 |
| 5.11. Prueba con Postman de la API de sugerencias. | 103 |
| 5.12. Primera pregunta del formulario para identificar características del jugador. | 104 |

ÍNDICE DE FIGURAS

| | |
|---------------------------------------------------------------------------------------------------------------------|-----|
| 5.13. Solicitud y respuesta del servidor usando como parámetros la información del formulario. | 105 |
| 5.14. Sugerencias obtenidas con base en las características del jugador. | 105 |
| 5.15. Detalles de la sugerencia. | 106 |
| 5.16. Resultados de la prueba de recomendaciones uno. | 107 |
| 5.17. Resultados de la prueba de recomendaciones dos. | 108 |
| 5.18. Resultados de la prueba de recomendaciones tres. | 109 |
| 5.19. Resultados de la prueba de recomendaciones cuatro. | 110 |
| 5.20. Resultados de la prueba de recomendaciones cinco. | 111 |
| | |
| 6.1. Escala de usabilidad SUS | 117 |
| 6.2. Resultados de la prueba: ¿Juegas videojuegos? | 118 |
| 6.3. Resultados de la prueba: ¿Con qué frecuencia juegas videojuegos? | 119 |
| 6.4. Resultados de la prueba: Frecuencia de juego por género | 120 |
| 6.5. Resultados de la prueba: ¿Has jugado League of Legends? | 121 |
| 6.6. Resultados de la prueba: Si has jugado League of Legends, ¿cuánto tiempo llevas jugando? | 121 |
| 6.7. Creo que me gustaría usar este sistema de recomendación de personajes con frecuencia. | 123 |
| 6.8. Encontré el sistema innecesariamente complejo para elegir un personaje adecuado. | 123 |
| 6.9. Pensé que el sistema hacía fácil encontrar un personaje para jugar. | 124 |
| 6.10. Creo que necesitaría el apoyo de otra persona para entender cómo usar este sistema de recomendación. | 125 |
| 6.11. Encontré que las diferentes funciones del sistema estaban bien integradas. | 126 |
| 6.12. Imagino que la mayoría de los jugadores aprenderían a usar este sistema de recomendación rápidamente. | 127 |
| 6.13. Pensé que había demasiada inconsistencia en las recomendaciones de personajes. | 128 |
| 6.14. Encontré el sistema complicado o poco intuitivo para elegir un personaje. | 129 |
| 6.15. Me sentí seguro al usar el sistema para encontrar un personaje adecuado para mí. | 129 |
| 6.16. Necesité aprender muchas cosas antes de poder aprovechar bien este sistema de recomendación. | 130 |
| 6.17. Distribución del Nivel de Experiencia | 133 |
| 6.18. Distribución del Estilo de Juego | 134 |
| 6.19. ¿Qué tanto cree que el campeón recomendado se adaptó a su estilo de juego? | 135 |
| 6.20. ¿Qué tan satisfecho quedó con el rendimiento del campeón recomendado durante la partida? | 136 |
| 6.21. ¿Considera que el sistema de recomendación le ayudó a mejorar su desempeño en la partida? | 137 |
| 6.22. ¿Qué tan adecuados le parecieron los ítems y runas sugeridos por el sistema? | 138 |

| | |
|---------------------------------------------------------------------|-----|
| 6.23. ¿Le gustaría seguir usando el sistema en el futuro? | 139 |
| A.2. Hero Section | 150 |
| A.1. Header | 150 |
| A.3. Objetivo del Sistema | 151 |
| A.4. Funcionamiento del sistema | 151 |
| A.5. Pregunta para elegir Rol | 152 |
| A.6. Pregunta para elegir Estilo de juego | 153 |
| A.7. Pregunta para seleccionar nivel de experiencia | 154 |
| A.8. Sección de Recomendaciones | 155 |
| A.9. Detalles sobre un campeón | 156 |

Índice de tablas

| | |
|---------------------------------------------------------------------------------|-----|
| 1.1. Categorías de runas en League of Legends y sus características principales | 26 |
| 1.2. Tipos de objetos en *League of Legends* y su costo promedio. | 29 |
| 4.1. Comparativa de estudios sobre análisis de datos en eSports | 77 |
| 5.1. Descripción de las variables del conjunto de datos. | 112 |
| 5.2. Tabla de campos del modelo de campeones. | 113 |
| 6.1. Promedio por pregunta en escala de 1 a 5 | 140 |

Introducción

En la actualidad, los sistemas informáticos se han convertido en una parte fundamental de la sociedad, permitiendo la automatización de tareas, el procesamiento de grandes volúmenes de información y la toma de decisiones más informada. Dentro de este vasto campo, los sistemas expertos y la minería de datos tienen una gran relevancia y aplicabilidad para la resolución de problemas.

Los sistemas expertos son programas informáticos diseñados para emular el conocimiento y la toma de decisiones de expertos humanos en un campo específico. Estos sistemas se componen de diversos elementos, como una base de conocimientos que almacena el conocimiento del experto, un motor de inferencia que procesa esa información y genera conclusiones, y una interfaz de usuario que facilita la interacción. Los sistemas expertos se han aplicado en una amplia variedad de campos, desde la medicina hasta la ingeniería, y se han convertido en herramientas necesarias para resolver problemas complejos de manera más sencilla. Por otra parte, la minería de datos tiene como objetivo la extracción de conocimiento a partir de grandes conjuntos de datos. Esta disciplina involucra el uso de modelos predictivos, que permiten hacer predicciones sobre eventos futuros, y modelos descriptivos, que ayudan a comprender patrones y tendencias en los datos. Dentro de los modelos descriptivos se hace uso de reglas de asociación, que revelan relaciones entre diferentes elementos de un conjunto de datos. Algunos de los algoritmos más destacados en este ámbito son Apriori, FP-Growth y ECLAT, cada uno con sus propias características y aplicaciones.

Aunado a lo anterior, las metodologías de desarrollo son parte fundamental en el desarrollo de sistemas expertos y soluciones de minería de datos. Las metodologías tradicionales, como el modelo en cascada, han sido ampliamente utilizadas en el desarrollo de sistemas computacionales a lo largo de los años. Estas metodologías se caracterizan por un enfoque secuencial y bien definido, donde cada fase del desarrollo (análisis, diseño, implementación, pruebas, mantenimiento) se completa antes de pasar a la siguiente. Por otro lado, las metodologías ágiles, como Scrum o Kanban, se han convertido en enfoques cada vez más populares en el desarrollo de software. Estas metodologías se caracterizan por un enfoque iterativo y colaborativo, donde los requisitos y las soluciones evolucionan a través de la colaboración entre equipos multifuncionales.

Además de las metodologías tradicionales y ágiles, también se puede considerar la metodología de análisis de datos (KDD, Knowledge Discovery in Databases) como una metodología para el desarrollo de soluciones basadas en minería de datos. KDD proporciona un marco estructurado que incluye pasos como la selección, limpieza, transformación, minería de datos y evaluación de los resultados.

En este trabajo, se analizan estas dos áreas, utilizando como área de aplicación el videojuego League of Legends. Este juego, es uno de los más populares a nivel mundial, presenta una gran cantidad de datos y mecánicas que lo convierten en un escenario ideal para la aplicación de sistemas expertos y técnicas de minería de datos.

League of Legends es un videojuego de estrategia en tiempo real (MOBA) en el que dos equipos de cinco jugadores se enfrentan en un mapa. Cada jugador controla un personaje, llamado *campeón*, con habilidades y estilos de juego únicos. El objetivo del juego es destruir la estructura principal del equipo contrario, conocida como "Nexo". Para lograr esto, los jugadores deben recorrer el mapa, recolectar recursos, mejorar a sus campeones y trabajar en equipo para derrotar a los oponentes. La progresión y economía del juego se basa en la adquisición de objetos y habilidades que mejoran las capacidades de los campeones. Los jugadores pueden obtener monedas (oro) a través de diversas actividades, como matar a los enemigos, destruir estructuras y completar objetivos. Estos recursos se utilizan para comprar mejoras permanentes o temporales para sus personajes.

Planteamiento del problema

En League of Legends (LoL), los jugadores casuales frecuentemente se enfrentan al desafío de seleccionar no solo el campeón más adecuado, sino también los ítems y las runas idóneas para cada partida. Esta toma de decisiones se complica debido a varios factores:

- La lista de campeones disponibles (más de 150).
- La variedad de ítems y combinaciones posibles.
- El complejo sistema de runas y sus efectos en el juego.
- Las frecuentes actualizaciones que alteran el balance del juego.
- La dificultad para adaptar las elecciones al nivel de habilidad individual y al rol deseado.

Los jugadores casuales normalmente carecen del tiempo y experiencia necesarios para mantenerse al día con las mejores estrategias para cada situación del juego. La LEC (League of Legends EMEA Championship) proporciona una gran fuente de datos sobre selecciones exitosas de campeones, construcciones de ítems y configuraciones de runas. Sin embargo, existe una brecha entre esta información de alto nivel y su aplicación práctica para jugadores de diferentes niveles de habilidad y estilos de juego. En este contexto, surge la necesidad de desarrollar una herramienta que pueda:

- Identificar patrones de éxito en la selección de campeones, ítems y runas.
- Proporcionar recomendaciones personalizadas y accesibles para jugadores casuales.

Justificación

La elección de desarrollar un sistema experto basado en técnicas de minería de datos para procesar partidas de la LEC y proporcionar recomendaciones personalizadas a jugadores casuales de League of Legends surge a la creciente popularidad del juego y la necesidad de mejorar la experiencia a jugadores casuales de este videojuego. Este proyecto aprovecha la oportunidad de aplicar técnicas de análisis de datos en el contexto de los videojuegos, con el potencial de crear una herramienta útil para una amplia base de jugadores.

El desafío central que se busca abordar es la brecha de conocimiento y orientación personalizada que enfrentan los jugadores casuales, quienes a menudo se ven abrumados por la complejidad del juego y gran cantidad de información disponible. La investigación propone desarrollar un sistema experto capaz de analizar datos de alto nivel de la LEC, traduciéndolos en recomendaciones prácticas y accesibles. Los beneficiarios directos de este desarrollo serán los jugadores casuales de League of Legends, quienes recibirán sugerencias personalizadas adaptadas a las necesidades específicas de cada jugador, considerando su nivel de habilidad, rol y estilo de juego.

Desde una perspectiva técnica, este proyecto representa un avance significativo en la aplicación de minería de datos y sistemas expertos en el campo de los videojuegos competitivos. El proceso de analizar registros oficiales de partidas, estadísticas de jugadores y equipos, y otros recursos relacionados con la LEC, utilizando la API oficial de RIOT GAMES, permitirá identificar patrones ocultos y relaciones significativas entre los elementos clave del juego. Esta base de conocimientos, procesada y presentada a través de una interfaz web intuitiva, ofrecerá a los jugadores casuales una ventana única al mundo del juego profesional, con recomendaciones directamente aplicables a su experiencia de juego.

El sistema propuesto tiene el potencial de convertirse en un modelo para futuros desarrollos en otros juegos y géneros. Además demuestra cómo la aplicación de técnicas de minería de datos puede beneficiar a usuarios de todos los niveles de habilidad, transformando la manera en que los jugadores interactúan con los juegos competitivos.

Propuesta de solución

Este estudio se enfoca en abordar esta problemática mediante el desarrollo de un sistema experto basado en datos del competitivo de LoL, que pueda ofrecer sugerencias personalizadas como:

- El campeón más adecuado según su nivel de habilidad y rol preferido.
- La selección de ítems ideal para el campeón seleccionado.
- La configuración de runas más efectiva de acuerdo a las características del rol preferido y campeón sugerido

Al ofrecer estas recomendaciones específicas, la herramienta tiene como objetivo simplificar el proceso de toma de decisiones, permitiendo a los jugadores casuales beneficiarse de estrategias probadas en el juego competitivo, adaptadas a su contexto individual.

En este contexto, la pregunta central que guía esta investigación es: ¿Cómo se puede desarrollar un sistema experto basado en técnicas de minería de datos que procese efectivamente las partidas de la LEC para proporcionar recomendaciones personalizadas y útiles a jugadores casuales de League of Legends? Adicionalmente, surgen preguntas secundarias como: ¿Qué técnicas de minería de datos son más efectivas para identificar patrones relevantes en los datos de la LEC? ¿Qué características de la interfaz web son necesarias para hacer que las recomendaciones sean fácilmente comprensibles por los usuarios?

Objetivo General

Desarrollar un sistema experto basado en técnicas de minería de datos que procese las partidas de la LEC (League of Legends EMEA Championship), con el fin de proporcionar recomendaciones personalizadas a jugadores casuales de League of Legends, considerando el nivel de habilidad, rol y estilo de juego en un ambiente web.

Objetivos Específicos

- Analizar registros oficiales de partidas, estadísticas de jugadores y equipos, y otros recursos relacionados con la LEC, utilizando la API oficial de RIOT GAMES, para identificar los datos relevantes para el sistema experto.
- Aplicar técnicas de preprocesamiento a los datos recopilados, implementando estrategias de limpieza, transformación, estandarización y normalización de los mismos, con el fin de prepararlos para su análisis mediante técnicas de minería de datos.
- Emplear técnicas de minería de datos, para examinar los conjuntos de datos y revelar patrones ocultos, relaciones y conocimientos valiosos, ejecutando librerías basadas en python
- Identificar relaciones significativas entre los elementos clave del juego (campeones, objetos, runas y estadísticas de rendimiento), llevando a cabo un proceso de interpretación de resultados, que permitan generar una base de conocimientos para el sistema experto.
- Diseñar la interfaz del sistema experto, generando prototipos a través del software Figma, que permita modelar el funcionamiento del sistema experto
- Implementar el sistema experto en la web con el lenguaje Python flask, que permita emitir recomendaciones en diferentes aspectos del juego.

Delimitaciones y limitaciones

El estudio se centrará exclusivamente en los datos de partidas y estadísticas de la LEC (League of Legends EMEA Championship) del año 2024 y no abarca otras ligas o torneos de League of Legends. Los datos utilizados estarán limitados a los disponibles públicamente o proporcionados por las organizaciones oficiales. Además, el estudio se enfocará en los factores relacionados con el juego mismo, excluyendo aspectos externos como el entorno de los jugadores y/o factores psicológicos.

Marco teórico

En este capítulo se abordan de manera estructurada los temas fundamentales relacionados con este proyecto. Como primer punto a tratar, se presenta un análisis general de los sistemas, sentando las bases para comprender los sistemas expertos. Posteriormente, se aborda el campo de la minería de datos, explorando conceptos básicos y modelos predictivos y descriptivos. Esto servirá como pauta para entender el modelo descriptivo y los algoritmos de reglas de asociación, elementos clave en este proyecto.

Posteriormente, se analiza el área de aplicación específica del proyecto: el videojuego League of Legends (LoL). Aquí, se presentan los conceptos básicos para comprender el funcionamiento y las mecánicas del juego, contextualizando así la relevancia del presente estudio.

1.1. Sistemas

Un sistema en computación es un conjunto de elementos físicos y lógicos diseñados con el fin de guardar, procesar y mostrar información. Como elementos físicos se hace referencia a todo el hardware, mientras que el software representa los elementos lógicos (16).

Estos sistemas están compuestos por tres elementos clave para su funcionamiento:

- **Hardware:** Abarca todas las partes físicas del sistema, incluyendo elementos como placa base, memoria RAM, dispositivos de almacenamiento y periféricos de entrada/salida (59).
- **Software:** Engloba todos los programas o aplicaciones que un usuario utiliza para realizar ciertas tareas en específico, como editores de texto, navegadores web, reproductores de música o videojuegos (16).

- Personal informático: Implica a todos los usuarios que están involucrados en el uso de estos sistemas, desde un desarrollador informático hasta el usuario final, quien interactúa con el sistema para realizar sus actividades (16).

El software puede clasificarse de distintas maneras según el tipo de función que cumpla:

- Software de sistema: Proporciona una interfaz entre el usuario y el hardware, otorgando un entorno para la ejecución de programas y administración del sistema. Incluye compiladores, enlazadores, intérpretes y herramientas de programación (54).
- Sistemas operativos: Administran los recursos del hardware y controlan la ejecución de los programas cargados en el sistema. Ejemplos conocidos son Windows, macOS, Linux y Android (48).
- Software de aplicación: Diseñado para realizar tareas específicas enfocadas en el usuario final. Engloba una gran variedad de aplicaciones como software de productividad, administración de bases de datos, herramientas de programación o sistemas de inteligencia artificial (59).

Dentro de esta gran variedad de sistemas, los sistemas expertos emergen como una herramienta avanzada de la inteligencia artificial, que permite resolver problemas mediante el conocimiento adquirido de un experto humano.

1.1.1. Sistemas expertos

Según León Quitarar (88), "Los sistemas expertos son el primer resultado operacional de la Inteligencia Artificial, pues logran resolver problemas a través del conocimiento y raciocinio de igual forma que lo hace el experto humano".

Un sistema experto es una aplicación de la inteligencia artificial que trata de imitar el comportamiento, raciocinio y toma de decisiones como lo haría un humano experto en el tema (103), con el fin de resolver problemas específicos de esa área de manera precisa y satisfactoria (88). Se basa en el conocimiento adquirido de una persona que domina el área o tema a tratar; toda esta información se almacena en una base de conocimientos. Posteriormente, el usuario proporciona los datos más relevantes de su consulta y el sistema proporciona una solución, ofreciendo resultados precisos como lo haría un humano (103).

1.1.2. Componentes de un sistema experto

La estructura interna de un sistema experto cuenta con distintos elementos que garantizan su correcto funcionamiento. Uno de ellos es la ya mencionada base de cono-

cimientos. Otros elementos de esta arquitectura son el motor de inferencia, la base de hechos, la interfaz de usuario o el módulo explicativo (53).

En la Figura 1.1, se puede observar detalladamente cuál es el funcionamiento de un sistema experto, iniciando del lado del experto, quien es el encargado del proceso de adquisición de conocimiento, hasta el usuario final, en la etapa de utilización del conocimiento.

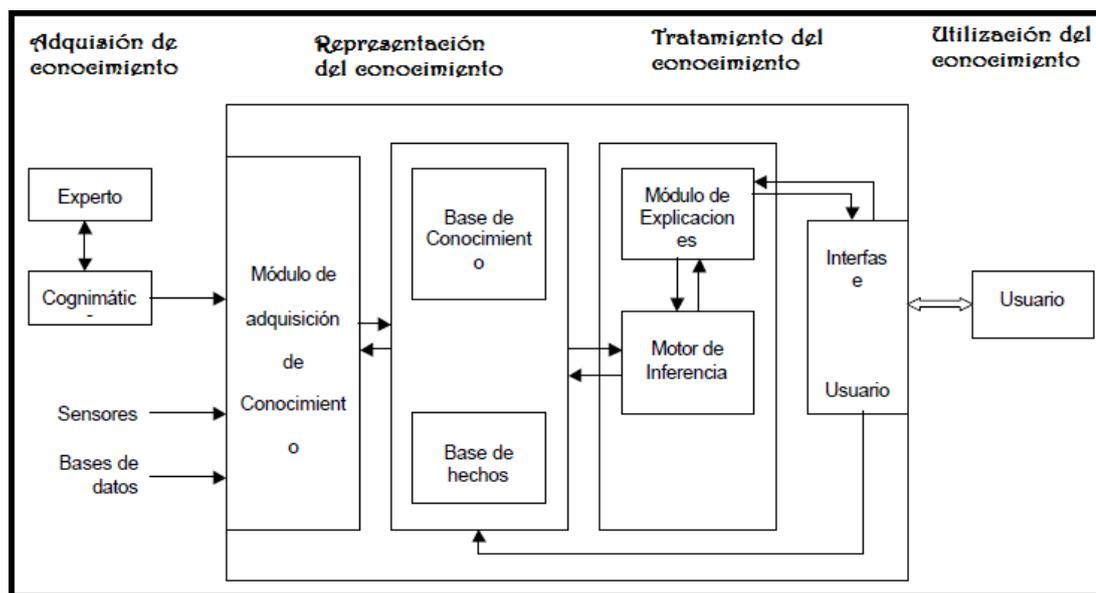


Figura 1.1: Estructura de un sistema experto

Base de conocimientos

Memoria en la que se encuentra recogido todo el conocimiento específico (experiencia), todo en relación al área o dominio en el que se pretende que funcione (53). Este conocimiento está basado en hechos, objetos, sucesos y situaciones relacionadas al contexto que aborda el sistema experto. Por otra parte, este conocimiento puede ser expresado en un conjunto de reglas que se sirven de hechos para obtener información que no se encuentra almacenada de forma explícita (88).

Base de datos o de hechos

Es un sistema de almacenamiento para todos aquellos datos recibidos en el proceso inicial de la resolución de un problema. Contiene información sobre el problema en el que se trabaja, por ejemplo, las respuestas que el usuario da a las preguntas que el sistema le pueda generar (53).

Motor de inferencias

Es el encargado de controlar el proceso de razonamiento del sistema para generar una solución. Utiliza tanto la base de hechos como la de conocimientos para generar una respuesta clara y concreta según las necesidades del usuario (53). León Quitarar (88) afirma que la estrategia de control puede ser de encadenamiento progresivo o de encadenamiento regresivo.

En el caso del encadenamiento progresivo, se efectúa el siguiente proceso para generar una respuesta ante la problemática:

- Evalúa las condiciones de todas las reglas con respecto a la base de datos, identificando las que sean más adecuadas para la resolución del problema.
- Si se encuentra una regla que sea aplicable, entonces se ejecuta su correspondiente acción y ese hecho generado es añadido a la base de datos; en caso contrario, se termina el proceso sin éxito.
- Una vez llegado al objetivo, se puede decir que el problema fue resuelto; en caso contrario, se vuelve al primer paso.

En el encadenamiento regresivo, el usuario comienza declarando una expresión E y el objetivo del sistema será establecer la verdad de esa expresión. Para ello deberá seguir los siguientes pasos:

- Buscar la expresión E en las reglas establecidas; estas nos darán la certeza de la veracidad de la expresión.
- En caso de encontrar varias reglas que se puedan aplicar, se elige una y se verifica su condición C en la base de datos.
- Si C es verdadera, eso significa que la expresión E es veraz y se resolverá el problema.
- Si C es falsa, se descarta esa regla y se elige otra para su pertinente verificación.
- Si C no es ni verdadera ni falsa, se cataloga como desconocida; por lo tanto, ahora se le considera como subobjetivo, es decir, C ahora es la expresión E y se regresa al primer paso.

Componente de adquisición

Permite la creación y edición de elementos de la base de conocimientos, además de servir como una herramienta que permite la supervisión y mantenimiento. El formato que suele utilizar este tipo de sistemas es similar a un editor de textos (53), y debería contar con las siguientes características:

- Capacidad para mostrar de forma transparente toda la información recogida en la base de conocimientos.
- Disponibilidad de un método para introducir el conocimiento en la base.
- Comprobación de la sintaxis de todas las unidades de conocimiento.

Componente explicativo

Le permite al usuario obtener una justificación de cuál ha sido la secuencia que ha seguido el sistema experto hasta obtener un resultado. Esto genera confianza en que la solución es factible y sigue un proceso seguro y confiable (53).

Interfaz de usuario

Es una de las partes más importantes, ya que será clave para que el usuario pueda describir su problema y, posteriormente, visualizar la solución generada por el motor de inferencias (88), todo esto de manera fácil e intuitiva, sin que el usuario necesariamente conozca el proceso que hay detrás para recibir la solución a su cuestionamiento.

1.1.3. Tipos de sistemas expertos

Existen varios puntos de vista de algunos autores sobre cuál podría ser una clasificación de los sistemas, algunos los clasifican según el tipo de problema a resolver. Según Castillo (14), se pueden clasificar en dos:

- **Problemas esencialmente deterministas:** Son conocidos como los sistemas expertos basados en reglas, porque sus conclusiones son a partir de un razonamiento lógico basándose en un conjunto de reglas.
- **Problemas esencialmente estocásticos:** Aquellos que utilizan medidas como la incertidumbre o la probabilidad para calcular la incertidumbre asociada a las conclusiones. Los sistemas expertos que usan la probabilidad como una medida de incertidumbre se les conoce como sistemas expertos probabilísticos.

Por otro lado, Badaro (8) propone que se pueden clasificar en cuatro según su funcionamiento

- **Basados en reglas previamente establecidas:** Resuelven problemas aplicando reglas establecidas y, posteriormente, generan y aplican nuevas reglas basadas en la problemática para llegar a una conclusión.
- **Basados en casos:** Obtienen la solución a nuevos problemas basándose en la solución a problemas anteriores. Es un método comúnmente usado por las personas, ya que buscan resolver una problemática a partir de experiencias pasadas y que

puede ser implementado en los sistemas expertos como una poderosa herramienta que busque la solución con base en experiencias previas.

- **Basados en redes bayesianas:** Basados en modelos bayesianos o probabilísticos que utilizan un gráfico acíclico dirigido (DAG), para representar variables aleatorias y sus dependencias condicionales.
- **Sistemas expertos difusos:** Emplean modelos de lógica difusa, donde se emplea un modelo matemático de conjuntos difusos para simular el razonamiento humano, permitiendo a la computadora tratar de imitar el comportamiento y la lógica de un humano y no la de una computadora.

Sistemas expertos difusos

Emplean modelos de lógica difusa, donde se emplea un modelo matemático de conjuntos difusos para simular el razonamiento humano, permitiendo a la computadora tratar de imitar el comportamiento y la lógica de un humano y no la de una computadora.

1.1.4. Etapas de desarrollo de un sistema experto

Castillo, en su libro "Sistemas expertos y modelos de redes probabilísticas"[\(14\)](#), sugiere a desarrolladores los pasos a seguir para el desarrollo de un sistema experto.

En la Figura 1.2, se muestra el ciclo a seguir para la construcción de un sistema experto, iniciando el planteamiento del problema hasta la implementación y mantenimiento, pasando por la elección de la herramienta de desarrollo, diseño del sistema y prototipos.

Las fases del ciclo de desarrollo de un sistema experto se pueden describir de manera general de la siguiente forma:

Planteamiento del problema: Definición y delimitación del problema, para establecer claramente los objetivos y restricciones del sistema.

Encontrar expertos humanos: La búsqueda de expertos humanos con conocimientos especializados en la problemática, que puedan proporcionar información y conocimiento relevantes para desarrollar una solución eficaz.

Diseño de un sistema experto: Planificación y diseño de las estructuras para almacenar el conocimiento, el motor de inferencias y la interfaz de usuario.

Elección de la herramienta de desarrollo: Elección de herramientas adecuadas según las necesidades del proyecto. Para sistemas simples (menos de 100 reglas), se

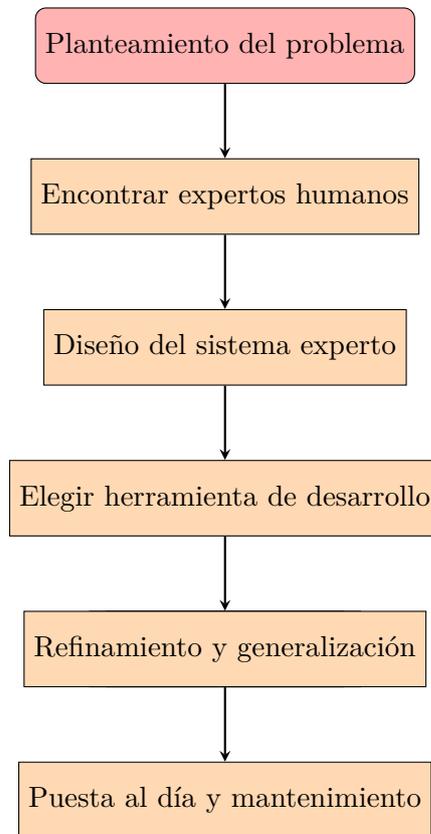


Figura 1.2: Ciclo de desarrollo de un sistema experto

recomiendan herramientas más manejables y fáciles de usar.

Desarrollo y pruebas de prototipo: Evaluación continua del prototipo mediante pruebas, identificando y solucionando posibles fallas hasta obtener un prototipo satisfactorio.

Refinamiento y generalización: Corrección de errores detectados y se añaden nuevas funcionalidades no incluidas en el diseño inicial.

Mantenimiento y puesta al día: Se solucionan problemas reportados por los usuarios, se corrigen errores, se actualiza el sistema con nuevos avances y se evalúa constantemente hasta obtener un sistema completo y sin fallas.

1.1.5. Aplicaciones y ejemplos de sistemas expertos

Los avances tecnológicos en los sistemas expertos han generado una clara evolución en su funcionamiento, funciones y procesamiento. Estos se aplican en diversas áreas

1. MARCO TEÓRICO

como medicina, economía, recursos humanos y soporte técnico, entre otras, mejorando significativamente la productividad en estos campos (91). Algunas de las aplicaciones más notables son:

- **Diagnóstico médico:** Analizan síntomas de un paciente y los comparan con una base de conocimientos para identificar enfermedades o condiciones médicas. Esto ayuda a los médicos a realizar diagnósticos más precisos y rápidos.
- **Soporte técnico:** Proporcionan respuestas y soluciones a los usuarios con problemas comunes o frecuentes. Ayuda a la empresa a minimizar el tiempo de respuesta y mejorar la experiencia de usuario.
- **Finanzas:** Pueden analizar datos históricos, tendencias de mercado y factores económicos que ofrezcan recomendaciones de cuándo comprar o vender acciones.
- **Logística:** Pueden considerar variables como la distancia, el tiempo de entrega, costo de combustible y restricciones de tráfico para sugerir cuál es la mejor ruta para realizar una entrega.

También se puede ver aplicado en situaciones de la vida cotidiana sin saber que es un sistema experto

Asistentes virtuales

Los asistentes de voz virtuales como Google Assistant o Amazon Alexa permiten dar órdenes para realizar tareas. Mediante técnicas de procesamiento de lenguaje e inteligencia artificial, el sistema puede interpretar lo que el usuario está diciendo y realizar tareas específicas (58).

Sistemas de detección de fraudes financieros

Analizan grandes volúmenes de datos para detectar patrones o comportamientos sospechosos que puedan indicar actividades ilegales. Gracias a esto, los usuarios pueden realizar transacciones de forma segura (21).

Todos estos ejemplos y aplicaciones demuestran que los sistemas expertos están en nuestro día a día, facilitando la vida de sus usuarios y teniendo gran importancia en el impacto de la experiencia de usuario.

1.2. Minería de datos

En los últimos años, ha ganado una gran relevancia debido a las posibilidades que ofrece en diversos sectores de la industria. Esta tecnología permite un análisis profundo

de datos, revelando información relevante para los usuarios.

Aunque la minería de datos no es una aplicación nueva, su uso se ha incrementado significativamente con los avances en hardware y software especializado, capaces de manejar tareas computacionalmente intensivas.

Existen varias definiciones de minería de datos, cada una enfatizando aspectos particulares:

- Amazon AWS (4): Un método analítico que explora y procesa volúmenes masivos de información con el propósito de revelar tendencias y relaciones ocultas.
- Microsoft (74): Una metodología que aplica análisis matemático para identificar tendencias y patrones significativos dentro de conjuntos de datos.
- SAS (92): Un procedimiento para detectar irregularidades, patrones recurrentes o relaciones significativas en grandes repositorios de información.

De estas definiciones se pueden identificar algunos puntos clave:

- Se aplica a grandes conjuntos de datos.
- Utiliza técnicas avanzadas de análisis.
- Busca encontrar patrones, anomalías o correlaciones ocultas.

La identificación de patrones y tendencias en grandes volúmenes de datos es fundamental para la creación de modelos predictivos y descriptivos. Estos modelos son esenciales para la toma de decisiones informadas y estratégicas.

1.2.1. Modelos (Predictivos y descriptivos)

Según Espino (39), un modelo predictivo es un modelo estadístico que utiliza datos existentes para predecir datos futuros. Su objetivo es evaluar la probabilidad de que en una muestra específica se exhiba un comportamiento determinado. También sugiere que cuenta con los siguientes elementos:

- Conjunto de entrenamiento: Muestras con atributos y comportamientos conocidos.
- Conjunto de prueba: Muestras con atributos similares, pero sin conocer su comportamiento.

Mientras que los descriptivos los define como aquellos que cuantifican las relaciones entre los datos, por ejemplo, clasificar clientes o contactos en grupos. Su objetivo es identificar las relaciones que existen entre los atributos de los datos.

1.2.1.1. Modelos predictivos

Como se mencionó anteriormente, los modelos predictivos se enfocan en inferir datos que aún no se conocen. Existen diversos algoritmos que permiten analizar los datos y mostrar la información necesaria para realizar una predicción.

Entre las técnicas predictivas se pueden encontrar:

- Clasificación
- Regresión

Clasificación

Permite asignar un elemento a un grupo o clase, esto con base en un análisis de todos los elementos o casos en los que se conozca la clase a la que pertenecen. Así, se genera un mecanismo que establece la pertenencia a clases según los valores de distintas variables, permitiendo un grado de influencia o discriminación de estas (69).

Se hace uso de técnicas como algoritmos matemáticos, sistemas expertos, sistemas de conocimiento e inducción de reglas.

Algunas de las técnicas de clasificación más usadas son:

- Árboles de decisión
- Naive Bayes
- Redes neuronales

Regresión

Se basa en el establecimiento de una ecuación matemática como un modelo que permita representar las interacciones entre las variables (39). Esta técnica permite obtener información sobre el comportamiento de las variables en distintas situaciones, por ejemplo, cómo una variable dependiente cambia cuando una independiente lo hace (39).

Entre los modelos de regresión más representativos están:

- Regresión lineal
- Regresión logística
- Regresión polinomial

Cada uno de estos modelos predictivos se utiliza dependiendo del estado y la aplicación de los datos, pero el fin común es poder hacer una predicción de un evento con base en los datos existentes.

1.2.1.2. Modelos descriptivos

Consisten en realizar un análisis de todo el histórico de los datos con el fin de buscar patrones y relaciones, para describir las características más sobresalientes. Permiten visualizar, detectar, calcular e identificar comportamientos ocultos, lo que facilita una toma de decisiones más informada (50).

Gutiérrez (51) afirma que algunas de las técnicas que se pueden encontrar son:

- Agrupamiento
- Reglas de asociación
- Análisis estadístico

Agrupamiento

Son técnicas que buscan identificar patrones y estructuras en los datos sin tener una etiqueta previa, con el objetivo de agrupar instancias que sean similares (86). Estas técnicas pueden ejecutarse sin tener conocimientos previos sobre los grupos y sus características, además de que no hay un objetivo predefinido, ya que no se sabe cuántos grupos se buscarán (57).

Algunos ejemplos son:

- K-means
- Agrupamiento jerárquico
- DBSCAN

Reglas de asociación

Establecen relaciones o correlaciones entre distintas acciones o sucesos que supuestamente son independientes. Con esto se puede identificar cómo la ocurrencia de una acción o acontecimiento puede generar la aparición de otros (69). Se busca entender las relaciones entre diferentes eventos o elementos, lo que puede proporcionar información relevante para la toma de decisiones.

Los algoritmos más comunes son:

- Apriori
- FP-Growth
- ECLAT

Análisis estadístico

Consiste en analizar, resumir y presentar los datos de una muestra o población, con el objetivo de describirlos de manera comprensible para poder realizar un mejor análisis. Permite representar la información de forma clara, facilitando su interpretación (83).

Se pueden encontrar las siguientes técnicas:

- Medidas de tendencia central
- Medidas de variabilidad
- Distribuciones

Los modelos descriptivos mencionados tienen el fin de realizar un análisis de todos los datos que permita describir sus características y realizar una mejor presentación e interpretación, con el fin de obtener patrones ocultos y tomar decisiones informadas.

En el desarrollo de este proyecto se hará uso de las reglas de asociación para realizar un análisis de los datos y encontrar las relaciones más importantes. A continuación, se hará una descripción detallada de estas técnicas.

1.2.2. Reglas de asociación

Las reglas de asociación buscan encontrar ítems con semejanzas en un conjunto de datos, estableciendo reglas que indiquen dependencias entre los ítems. Es una técnica que surgió con el objetivo de realizar un análisis de las ventas en mercados, también denominado "Market Basket Analysis", y consta de un análisis a un conjunto de ítems que un cliente compra, para obtener información acerca de la compra de productos (75). Actualmente, esta técnica es usada en procesos de descubrimiento del conocimiento, que permiten encontrar tendencias y patrones de comportamiento en los datos (85).

Una regla de asociación se puede expresar como $X \Rightarrow Y$, donde X y Y son conjuntos de elementos. Estos elementos son atributos que permiten crear expresiones lógicas como conjunciones, disyunciones y negaciones.

Si un conjunto de datos contiene el elemento X , es probable que también contenga el elemento Y (46).

1.2.2.1. Métricas clave

Las métricas clave son parámetros de entrada para algoritmos de inducción de reglas de asociación; una de sus aplicaciones puede ser comprobar la fiabilidad de las reglas. Las métricas más utilizadas por algoritmos de inducción de reglas de asociación son soporte (Support), confianza (Confidence), elevación (Lift) y convicción (Conviction) (85).

Soporte

Número de transacciones que ocurren en una regla en relación con los datos, o dicho de otra manera, es la frecuencia con la que los términos de una regla de asociación se

encuentran en los datos.

Para cualquier conjunto de elementos $X \subseteq R$, existe una propiedad fundamental que permite cuantificar la frecuencia de aparición de lo que se denomina *soporte*. Molina (46) define la Ecuación 1.1, en donde se considera R como un conjunto universal y r como una base binaria definida sobre R . Sea $X \subseteq R$ un subconjunto de ítems. Se dice que X concuerda con una línea $t \in r$ si $X \subseteq t$. El conjunto de líneas de r con las que X coincide se denota como $M(X, r)$. Por ejemplo, $M(X, r) = \{t \in r \mid X \subseteq t\}$. El *soporte* de X en r , denotado como $sop(X, r)$, es:

$$sop(X, r) = \frac{|M(X, r)|}{|r|} \quad (1.1)$$

Confianza

Medida que representa un valor de correspondencia entre los ítems que componen una regla (85). La Ecuación 1.2 describe que es la probabilidad de que una línea elegida de manera aleatoria dentro de r coincida con X y también con Y (46).

Se define como:

$$conf(X \Rightarrow Y, r) = \frac{|M(X \cup Y, r)|}{|M(X, r)|} \quad (1.2)$$

Elevación

Métrica utilizada para medir el grado de dependencia entre los términos de una regla. La Ecuación 1.3 muestra las variables que permiten obtener la métrica de elevación.

$$lift(X \Rightarrow Y, r) = \frac{conf(X \Rightarrow Y, r)}{sop(Y, r)} \quad (1.3)$$

Convicción

Pinho (85) la define como una medida estadística que evalúa el grado en que el término antecedente influye en la ocurrencia del término consecuente dentro de una regla de asociación. Una característica importante de la convicción es que es una medida unidireccional, lo que significa que el resultado de convicción puede variar en un rango que va desde 1 hasta el infinito positivo. Cuanto más alto sea el valor de convicción, mayor será la probabilidad de que ocurra el término consecuente cuando se presenta el término antecedente de la regla. La Ecuación 1.4 describe esa característica de forma teórica.

$$conviction(X \rightarrow B) = \frac{sop(X) \times (N - sop(X, Y))}{sop(X) - sop(X, Y)} \quad (1.4)$$

1.2.3. Algoritmos principales

Dentro de los algoritmos de reglas de asociación, existen diversos algoritmos que se han desarrollado para abordar diferentes aspectos del análisis de datos. Estos algoritmos son fundamentales para descubrir patrones, asociaciones y tendencias en grandes conjuntos de datos. Esta sección se centra en tres algoritmos principales que han demostrado ser efectivos y ampliamente utilizados en la práctica: Apriori, FP-Growth y ECLAT.

1.2.3.1. Apriori

Uno de los primeros algoritmos desarrollados para la búsqueda de reglas de asociación (17). Se basa en el principio de *clausura descendente del soporte de itemsets* (45).

Según el sitio web *Ciencia de datos*"(17), el algoritmo consta de dos etapas:

- Identificar aquellos itemsets que ocurren con una frecuencia por encima de un límite predeterminado.
- Convertir los itemsets frecuentes en reglas de asociación.

1.2.3.2. FP-Growth

Según Pinho (85), es un algoritmo creado con el fin de mejorar las deficiencias en el rendimiento del algoritmo Apriori. Está basado en una estructura de datos llamada FP-Tree (Árbol de Patrones Frecuentes), en la cual cada nodo representa un elemento y su recuento actual; además, cada rama representa una asociación diferente (87). A diferencia de Apriori, permite extraer reglas a partir de itemsets frecuentes sin generar candidatos para cada tamaño (17) y sin verificar si superan un determinado umbral.

La estructura consiste en un nodo raíz que contiene una etiqueta de valor nulo, conjuntos de subárboles (hijos del nodo raíz) y una tabla cabecera que almacena información sobre los ítems frecuentes (85). El árbol se construye leyendo un conjunto de datos y asignando a cada transacción una ruta del árbol FP. Al tener diferentes transacciones, se pueden obtener varios elementos en común por lo que sus rutas se superponen, lo que puede significar mayor compresión en la estructura FP. Si el tamaño del árbol FP es lo suficientemente pequeño para almacenarse en memoria principal, permitirá extraer conjuntos de elementos frecuentes directamente desde la memoria, sin la necesidad de realizar pases repetidos sobre los datos almacenados en disco (87).

1.2.4. ECLAT

El algoritmo Equivalence Class Clustering and bottom-up Lattice Traversal (ECLAT) es una versión basada en Apriori con la diferencia de que es más eficiente y escalable, debido a que funciona de manera vertical, lo que permite que sea más rápido que Apriori (47). Consta de dos fases: en la primera se calcula el soporte de los ítems que aparecen en la tabla; posteriormente, para cada ítem que supere el umbral de soporte, se extraen ramas de árbol donde aparece el ítem y se reajusta el soporte de los ítems que aparezcan en las ramas. Después se genera un nuevo árbol considerando las ramas extraídas y se seleccionan los itemsets que superen el umbral de soporte mínimo. Con los itemsets frecuentes se generan todas las reglas posibles y se seleccionan las que superen el umbral de confianza mínimo (27).

1.2.5. Discusión

Para la generación de reglas de asociación, se opta por el algoritmo Apriori debido a las características del proyecto, ya que el conjunto de datos cuenta con variables que se adaptan mejor a este algoritmo. Por ejemplo, otros algoritmos requieren que todos los datos sean categóricos o numéricos, lo que dificulta la etapa de preprocesamiento y del análisis de los datos.

En cuanto al análisis y detección de relaciones entre los datos, se decantará por el algoritmo K-Means para el clustering de campeones, permitiendo agruparlos basados en sus atributos fundamentales como ataque, defensa, magia y dificultad.

La integración de estos diversos algoritmos y técnicas analíticas permite obtener datos ocultos de los datos, abarcando desde la clasificación de campeones hasta la predicción de resultados y el análisis de estrategias de equipamiento. Al combinar las características que ofrece el algoritmo Apriori en la generación de reglas de asociación con las estadísticas derivadas de los algoritmos de clustering, visualización y predicción, se puede desarrollar un sistema capaz de proporcionar un análisis profundo con recomendaciones más acertadas en el complejo ecosistema de League of Legends.

1.3. Área de aplicación

League of Legends es un juego de estrategia por equipos, constituidos por dos equipos de cinco campeones que se enfrentan para ver quién destruye la base enemiga antes que el otro. El juego cuenta con 140 campeones que permiten realizar jugadas espectaculares, asesinar rivales y derribar torretas con el objetivo de lograr la victoria (90).

Breve historia de League of Legends

League of Legends (LoL) fue lanzado en 2009 de la mano de la empresa Riot Games (71). La idea surgió cuando Steve “Guinsoo” Feak y Steve “Pendragon” Mescon, desarrolladores de mods para el videojuego Warcraft 3, se reunieron en 2005 en el estudio de Riot Games con el fin de crear un MOBA autónomo independiente al videojuego Warcraft 3, algo que era muy común en la época. Tras 4 años de desarrollo, se presentó una beta en abril de 2009, en la que se presentaban 19 campeones y una jugabilidad muy limitada. Hasta octubre de ese mismo año se lanzó de manera completa con 40 campeones (12). A lo largo de su historia, ha sufrido diversos cambios que han significado una gran evolución desde la primera versión del juego. Una de las más importantes fue la adquisición de la empresa china Tencent Holdings Limited a Riot Games, lo que implicó un cambio en el desarrollo y funcionalidades del juego. Hasta el día de hoy, LoL es uno de los videojuegos más populares, albergando jugadores de distintas edades, experiencias y estilos de juego (65). Esto gracias a constantes actualizaciones que permiten agregar nuevas estrategias de fiabilidad y añadir nuevos campeones con habilidades únicas, lo que permite una nueva experiencia del juego. A día de hoy se cuentan con más de 150 campeones disponibles para jugar (90).

Género y tipo de juego (MOBA - Multiplayer Online Battle Arena)

El subgénero de videojuegos Multiplayer Online Battle Arena, mejor conocido como MOBA, es un juego de estrategia en tiempo real en donde equipos de normalmente 5 jugadores compiten entre sí en un campo de batalla con el fin de destruir la base enemiga antes que el otro equipo (61). Es uno de los géneros más populares; por ejemplo, el juego League of Legends tiene más de 117 millones de jugadores jugando mensualmente. Otro de los juegos más populares es Dota 2, el cual es la competencia directa de League of Legends en cuanto a usuarios y ganancias generadas, contando mensualmente con más de 500,000 jugadores activos, que aunque es mucho menor que LoL, siguen siendo grandes cifras para un videojuego (38).

1.3.1. Mecánicas básicas del juego

Para entender mejor los conceptos básicos del juego, es necesario conocer el mapa; esto permitirá conocer mejor los elementos y roles dentro del juego. En la Figura 1.3 se puede observar cómo es el mapa de forma general, señalándose las distintas zonas del mapa y principales estructuras. Movistar Esports (5) define como principales estructuras del juego a 3 elementos:

- Nexo
- Inhibidores
- Torres



Figura 1.3: Zonas del mapa y principales estructuras

Nexo

El objetivo del juego que permite ganar una partida es destruir el nexo enemigo. El nexo es la estructura más importante del mapa; para llegar a ella es necesario destruir objetivos llamados inhibidores, que fungirán como una especie de defensa del nexo.

Inhibidores

Son tres estructuras que se encuentran dentro de la base del equipo, están protegidas por torretas y la destrucción de uno de ellos da acceso al nexo. Como característica principal, estos pueden regenerarse pasado un tiempo.

Torres

Son estructuras defensivas que disparan a todo lo que se acerque y protegen al nexo e inhibidores. Son divididas en dos según su ubicación: torres y torretas. Son la resistencia principal del juego e impiden que los jugadores avancen por cualquier lugar del mapa en el lado enemigo. Cada equipo tiene 11 torres ubicadas en las distintas líneas del mapa, 3 por línea y dos encargadas de defender el nexo.

El juego ofrece una variedad de personajes denominados campeones que, con diferentes habilidades y estilos de juego, permitirán al jugador elegir cuál será la forma en

que quieren jugar.

Campeones

El término campeón se refiere a un ser ficticio el cual es invocado a un mapa para conformar equipo con un grupo de 3 o 5 jugadores, controlados por otros jugadores. Cada uno cuenta con un kit de habilidades distintas entre cada campeón, con diferentes estadísticas de daño, vida, defensa, velocidad o mejora. Estas habilidades se determinan dependiendo de los atributos de cada campeón (41).

Estos están divididos en 6 roles:

- Asesino
- Luchador
- Mago
- Tirador
- Soporte
- Tanque

Cada uno de los personajes cuenta con sus propias habilidades que deben ir desbloqueando progresivamente cuando suben de nivel mientras avanza la partida.

En la Figura 1.4 se observan las tres calles o líneas principales: la Top Lane, Mid Lane, Bot Lane y también una cuarta interconectada llamada la jungla. Cada jugador debe elegir en qué posición quiere jugar.

Calle superior (Top Lane)

Se caracteriza por contar con campeones tipo tanque, es decir, aquellos que cuentan con mucha vida, aguanten mucho daño gracias a su defensa y puedan quitar daño una vez tengan el equipo necesario. Esta línea se caracteriza por la paciencia, ya que cada jugador debe tener cuidado e intentar cometer los mínimos errores posibles al atacar o defenderse del enemigo, ya que el combate es 1 contra 1.

Jungla

Entre las diferentes calles, hay zonas interconectadas entre sí; estas zonas son conocidas como *la jungla*. El jugador que decida ir en esa posición se le conocerá como *el jungla*, una de las posiciones más difíciles debido a que en este rol se tiene que estar pendiente de todo el mapa y además ayudar a sus compañeros. El jungla debe ir eliminando a los monstruos que aparecen en esa zona, permitiéndole adquirir experiencia y



Figura 1.4: Mapa de League of Legends

oro.

Calle central (Mid Lane)

Es una de las posiciones más sacrificadas a la hora de jugarse, ya que implica un buen posicionamiento y visión, debido a que se pueden recibir ataques desde cualquier línea sin que lo sepas. Por eso es necesario tener una buena atención al mapa y una buena comunicación con el equipo. Hay que controlar todas las zonas cercanas para saber si se acerca un enemigo o no.

Calle inferior para dos posiciones: el carry y el support

En esta se ubican dos roles, el ADC (Attack Damage Carry) y el support (apoyo), suelen considerarse las posiciones más fáciles para los jugadores novatos debido a que la partida se desarrolla durante un largo tiempo con un 2 contra 2. El carry suele ser el personaje que más daño va a hacer, pero cuenta con una vitalidad, defensa y resistencia bajas respecto a los demás campeones; en cambio, genera mucho daño. Suele estar acompañado por el support o apoyo, quien se encarga de apoyar al tirador durante toda la partida. El objetivo del support no es golpear o sacar oro, sino ayudar al resto. Los mejores campeones para este rol son los magos o tanques, ya que pueden servir para curar, establecer barreras, recibir mucho daño o inmovilizar al enemigo (11).

Una de las formas para que un campeón reciba mejoras iniciales de acuerdo al rol, y que pueden significar ganar un enfrentamiento al enemigo, es el uso de las runas, ítems que son elegidos antes de iniciar una partida y que pueden hacer que el campeón aumente sus estadísticas iniciales.

Runas

Una runa es una mejora que el jugador elige para sus campeones antes de iniciar una partida, permitiendo aumentos en las estadísticas de los campeones. Cada runa otorga una mejora en una categoría específica y estas se acumulan para obtener mejores beneficios (42).

Existen varias categorías enfocadas a diferentes estadísticas del campeón en específico; por ejemplo, vida, daño o capacidad de absorber daño. En la Tabla 1.1 se mencionan los 5 tipos de runas y cuál es su característica principal. Para el jugador es importante conocer qué runas elegir porque de eso depende tener un buen inicio de partida y llevar una ventaja sobre el enemigo.

| Categoría de Runa | Característica Principal |
|--------------------|-------------------------------------------------|
| Precisión | Aumento de velocidad de ataque. |
| Dominación | Aumento de daño de ráfaga. |
| Brujería | Incremento de poder de habilidad y daño mágico. |
| Valor | Aumento de vida y resistencia. |
| Inspiración | Acceso a herramientas y objetos especiales. |

Tabla 1.1: Categorías de runas en League of Legends y sus características principales

1.3.2. Progresión y economía dentro del juego

Durante el transcurso de una partida, mientras más tiempo pase, el campeón necesitará mejoras que ayudarán a aumentar las estadísticas del personaje. Estas mejoras son compradas en la tienda ubicada en la base del campeón, mediante la moneda del juego conocida como oro, la cual puede ser obtenida realizando diferentes actividades dentro de la partida.

Oro

El oro es la moneda con la que se comercia dentro del juego. Se utiliza para comprar

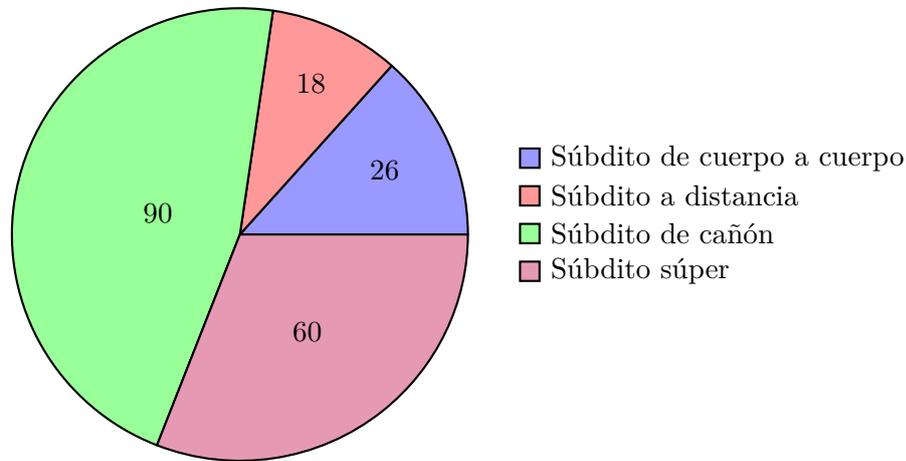


Figura 1.5: Distribución del oro otorgado por diferentes tipos de súbditos.

objetos en la tienda, que permiten mejorar estadísticas y habilidades de los campeones, para que aumenten su poder a lo largo de una partida (80).

Una de las mecánicas importantes dentro de las partidas es la progresión de niveles y economía de los personajes. Existen diversos métodos que permiten generar oro o experiencia para subir de nivel, entre los cuales están:

- Derribar torres
- Matar enemigos
- Eliminar súbditos
- Asesinar monstruos de la jungla

Súbditos

Los súbditos son unidades aliadas o enemigas que se generan automáticamente con el objetivo de atacar a las unidades y torres enemigas que estén presentes en los carriles que les son asignados. Los hay de distintos tipos y aparecen en determinadas situaciones a lo largo de la partida (64). Son la forma más práctica de conseguir oro y subir de nivel debido a que constantemente hay oleadas que permiten eliminarlos. En la Figura 1.5 se describen los distintos tipos de súbditos y cuál es el beneficio en oro por eliminarlos, siendo los súbditos de cañón los que más oro dan, pero estos solo aparecen cada 3 oleadas.

Monstruos de la jungla

En el caso del rol de jungla, la situación es diferente debido a que no tiene la posibilidad de eliminar súbditos. La forma de generar oro es a partir de asesinar a los

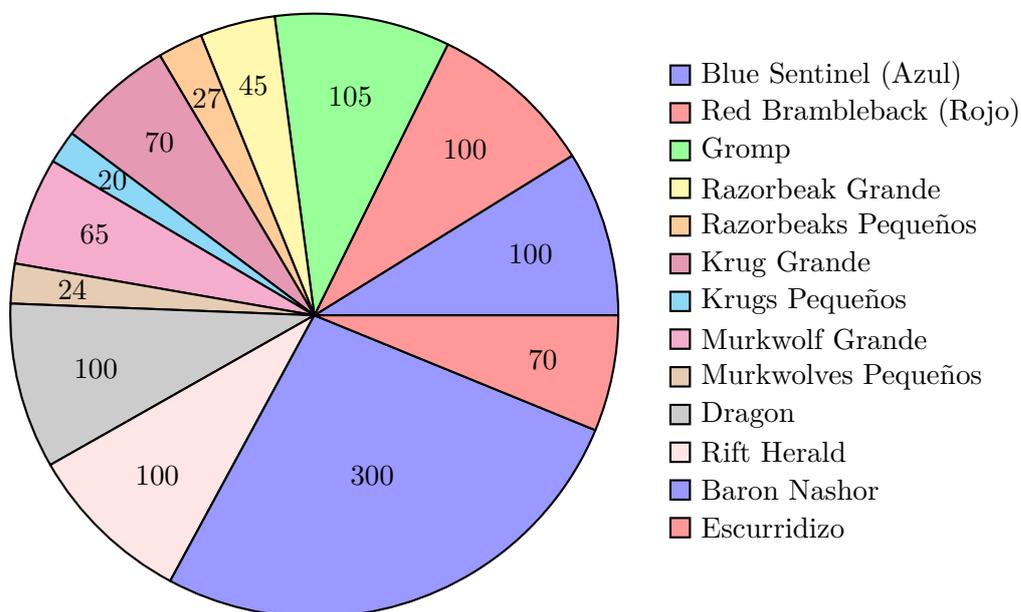


Figura 1.6: Distribución del oro otorgado por los monstruos de la jungla.

diferentes monstruos de la jungla que, dependiendo del nivel de dificultad, darán más o menos oro (79). Esto no solo aplica a los junglas, sino que conforme avance la partida los demás jugadores tendrán la posibilidad de generar un extra de oro al matar estos monstruos sin afectar la economía del jungla. En la Figura 1.6 se describe cuál es el oro otorgado por los diferentes enemigos de la jungla, siendo el Barón Nashor el que más oro da al jugador.

Objetos

El oro permite comprar diversos objetos en la tienda que aumentan las habilidades del jugador para poder ser mejor que el enemigo y ganar los enfrentamientos. El inventario del jugador puede llevar hasta seis objetos y una baratija a la vez, lo que hace que la elección de los artículos sea un factor importante en el desempeño de un campeón (81). En la Tabla 1.2 se describen los diferentes tipos de objetos y un costo promedio. Los objetos consumibles o las botas son los más baratos e ideales para el inicio de la partida por su bajo costo, pero conforme esta avance, lo recomendable es adquirir mejores objetos de acuerdo al campeón y rol durante la partida.

Tabla 1.2: Tipos de objetos en *League of Legends* y su costo promedio.

| Tipo de Objeto | Costo Promedio |
|-------------------------------------|----------------------|
| Botas | 300 de oro |
| Consumibles | 50 - 100 de oro |
| Objetos de Daño Físico | 2,800 - 3,500 de oro |
| Objetos de Daño Mágico | 2,800 - 3,500 de oro |
| Objetos de Defensa | 2,500 - 3,800 de oro |
| Objetos de Utilidad | 2,000 - 3,500 de oro |
| Objetos de Asesino | 2,800 - 3,300 de oro |
| Objetos de Apoyo | 2,000 - 3,200 de oro |
| Objetos de Enfriamiento y Movilidad | 2,800 - 3,500 de oro |
| Objetos Legendarios | 3,000 - 3,600 de oro |
| Objetos Míticos | 3,000 - 3,500 de oro |

1.3.3. ¿Cómo empezar a jugar?

Para comenzar a jugar es necesario contar con algunas consideraciones que permitan tener un mejor contexto sobre el juego y cómo comenzar en este.

Instalación

Es necesario tener en consideración las especificaciones mínimas para poder instalar el juego y tener una buena experiencia, sin que esta se vea afectada por factores como el rendimiento.

Se requiere un equipo con sistema operativo Windows 10 o versiones superiores, o MacOS con versiones superiores a la 10.12 (89), con los siguientes requisitos mínimos para su buen funcionamiento:

- RAM: 2 GB
- Almacenamiento disponible: 16 GB
- Memoria gráfica: 1 GB

Una vez considerados estos requisitos, el enlace [1.1](#) dirige a la página oficial de League of Legends, donde se podrá descargar la última versión del juego, tanto para

1. MARCO TEÓRICO

Windows como para MacOS.

Enlace 1.1: Página oficial para descargar League of Legends

1 <https://www.leagueoflegends.com/es-es/download/>

Cómo jugar una partida de LoL

Una vez instalado el juego, se iniciará un menú conocido como el *Ciente del LoL*, donde se encuentran diversas opciones como ver personajes, objetos comprados, los diferentes modos de juego disponibles y noticias relacionadas al videojuego. La Figura 1.7 muestra algunas de las opciones que se pueden encontrar dentro del cliente del LoL.

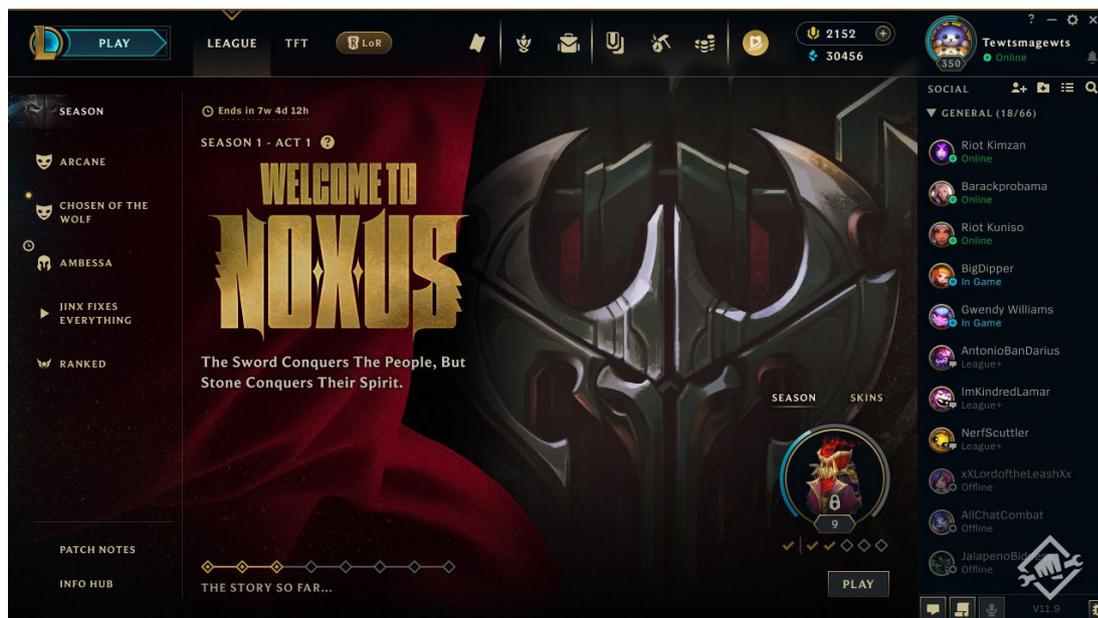


Figura 1.7: Cliente del LoL

Para jugar una partida basta con presionar el botón de *Jugar*, que automáticamente redirige al jugador a una sala de espera donde puede invitar o ver a los integrantes de su equipo, así como realizar algunos ajustes, como elegir el rol que desea jugar. La Figura 1.8 muestra los elementos más destacados de la sala de espera. Posteriormente, el juego enviará una notificación sobre la disponibilidad para iniciar una partida, a la que el jugador debe responder si acepta o no unirse e iniciar el juego.



Figura 1.8: Sala de espera para jugar una partida

Una vez aceptada la invitación para unirse a una partida, el jugador entrará a una sección donde podrá elegir un personaje de los más de 150 disponibles. La Figura 1.9 detalla el funcionamiento de la selección de un campeón, mediante un sistema de turnos en el que cada jugador cuenta con un tiempo determinado para elegir un personaje que se adapte a su rol y posibles necesidades dentro de la partida.

1. MARCO TEÓRICO

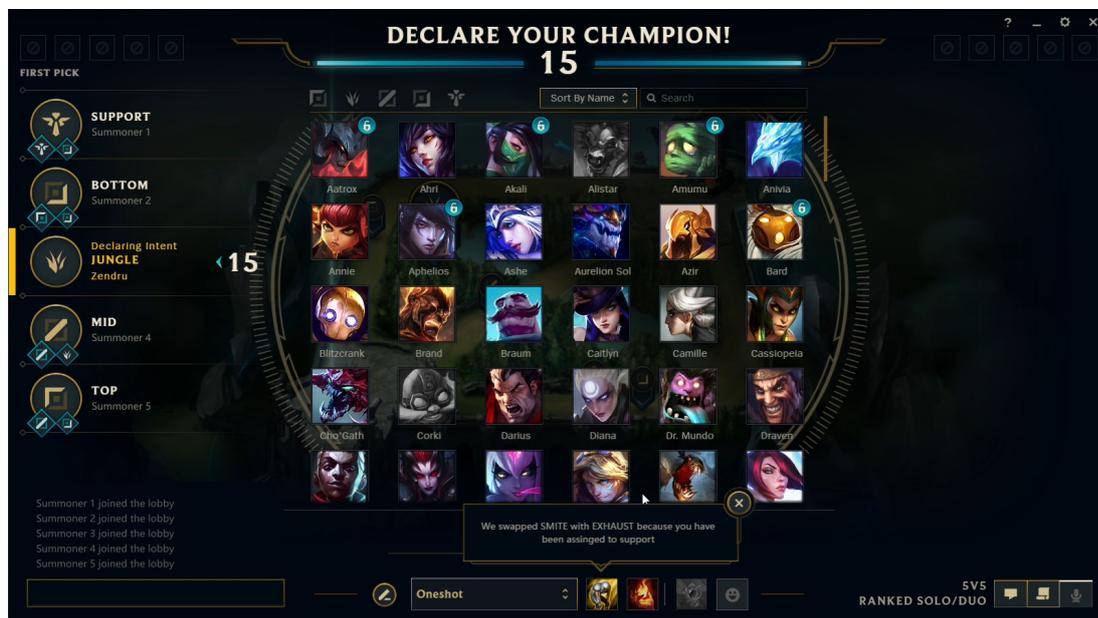


Figura 1.9: Menú de selección de campeones

Por último, antes de entrar a la partida, es necesario configurar las runas que usará el personaje. Esta opción también está disponible dentro de la sección de selección de campeones. La Figura 1.10 muestra algunas de las posibles runas que se pueden elegir para el personaje seleccionado, en donde será necesario conocer su funcionamiento y las posibles mejoras que puedan ofrecer en partida.

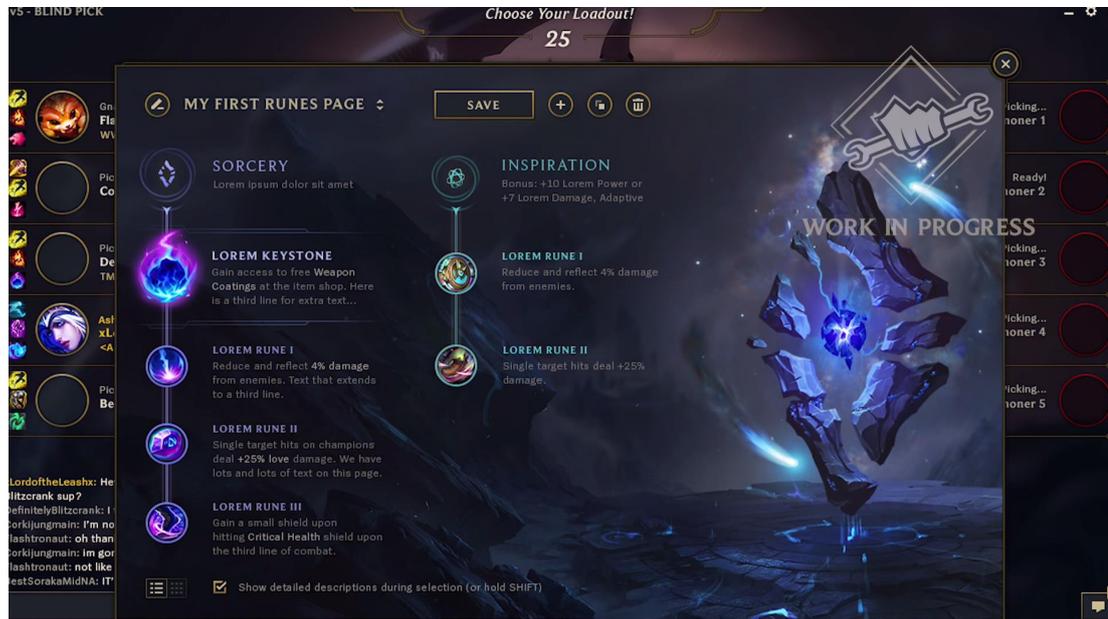


Figura 1.10: Selección de runas para el personaje

Ultimados todos los detalles, es momento de dar inicio a la partida y buscar destruir el objetivo final, conocido como *Nexo*.

Marco metodológico

En este capítulo se realiza un análisis comparativo de las metodologías de desarrollo, tanto tradicionales como ágiles, destacando sus características más notables. El capítulo concluye con una descripción detallada de las metodologías de análisis de datos Knowledge Discovery in Databases (KDD) y Sample, Explore, Modify, Model and Assess (SEMMA).

2.1. Metodologías de desarrollo

Una metodología de desarrollo permite estructurar, planificar y contar con un control en el proceso de desarrollo de software (68). Existen las tradicionales como el modelo en cascada o en espiral, que se enfocan en seguir un proceso secuencial durante todo el proceso de desarrollo. Por otro lado, en los últimos años han estado en auge las metodologías ágiles, debido a que se enfocan en un desarrollo más dinámico, sin ser tan riguroso como una metodología tradicional.

2.1.1. Metodologías tradicionales

Las metodologías tienen como característica principal el tener una estructura establecida, lineal y nada flexible ante los cambios. Cuentan con roles, actividades y artefactos ya preestablecidos, requieren de una documentación exhaustiva, clara y detallada del proyecto. Sin embargo, a día de hoy las metodologías tradicionales siguen siendo usadas debido a que se cuenta con buenas prácticas, control y seguimiento de cada aspecto que las conforma (104). Algunas de las metodologías más conocidas son el modelo en cascada *Waterfall* o el modelo en espiral. Son metodologías con una estructura muy lineal pero bastante funcional para diversos proyectos que así lo requieran.

2.1.1.1. Modelo en cascada (Waterfall)

Es una metodología de desarrollo secuencial, muy utilizada en el desarrollo de software. Consta de varias etapas que deben ejecutarse una tras otra. Recibe ese nombre debido a su estructura en fases, ya que deben seguirse una después de otra siguiendo un orden en concreto y estricto (31).

En la Figura 2.1 se muestran las diferentes fases que se deben seguir para poder completar esta metodología de manera exitosa. Consta de cinco fases, iniciando con el análisis de requisitos, después fases de desarrollo como diseño, implementación y verificación, y por último la etapa de mantenimiento.

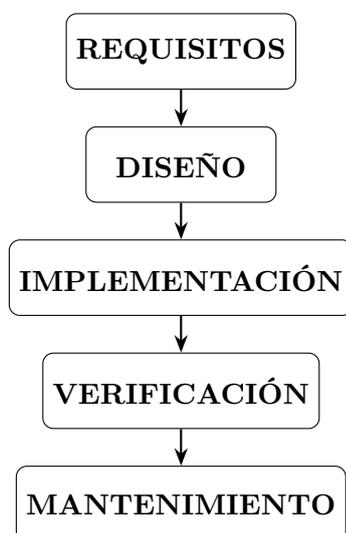


Figura 2.1: Esquema del modelo en cascada para el desarrollo de software.

Análisis de Requisitos

Se especifican los requisitos que debe ofrecer el producto, y la problemática que resuelve el sistema (55). Se analiza qué servicios, restricciones y metas del sistema existen (31). De manera general, son los requisitos del sistema y del proyecto.

Diseño

Se presenta una solución a la problemática, teniendo en cuenta el análisis de requisitos realizado en la fase anterior, con el fin de tener un plan de diseño detallado que incluya tareas y acciones específicas. Como puede ser el diseño de interfaces, el entorno de desarrollo, frameworks, librerías y todo aquello que se pueda requerir para el desarrollo del proyecto (55).

Implementación

Consta de tres etapas en donde se contempla todo el desarrollo del sistema:

- Programación del software: Proceso de creación de la aplicación o programa.
- Búsqueda y corrección de errores: Pruebas unitarias que buscan detectar cualquier error y poder ser corregido.
- Resultado final: Producto final que debe ser probado en la siguiente fase.

Verificación

Se prueba el sistema completo para asegurar que cumple con todos los requisitos y expectativas definidas en la fase de análisis (31). Si el producto cumple con lo requerido, está listo para su lanzamiento ante el usuario final.

Mantenimiento

Entrega del producto al cliente y el posterior mantenimiento del sistema con parches o actualizaciones para corregir errores que no fueron descubiertos en etapas anteriores, mejorar el sistema y adaptarlo a las necesidades del cliente (31).

A pesar de ser una metodología que puede llegar a ser muy lineal y poco flexible, permite llevar un orden y una secuencia en el desarrollo de un software, que para cierto tipo de proyectos puede ser una opción más viable para su uso.

2.1.1.2. Modelo en espiral

Modelo de proceso de software evolutivo donde se une la construcción de prototipos con aspectos controlados y sistemáticos de un modelo lineal y secuencial. Se basa en un desarrollo rápido de versiones incrementales del software que no se basan en fases definidas y separadas para crear un sistema (84).

En la Figura 2.2 se muestra que el modelo en espiral consta de cuatro etapas con un proceso iterativo hasta culminar con el desarrollo del software.

Definición de objetivos

Se definen los objetivos específicos, para poder identificar las limitaciones del proceso y del software, además de la gestión e identificación de riesgos. Se valoran los objetivos que deben vincularse a cada uno de los pasos del desarrollo del software (60).



Figura 2.2: Estructura del modelo Espiral

Valoración de las alternativas

Se identifican los riesgos que pueden llegar a presentarse durante el proceso del desarrollo del software. Posteriormente se gestionan estrategias más efectivas que permitan minimizar el impacto de estos fallos en el proyecto (84).

Desarrollo, verificación y validación

Desarrollo del software en donde, a manera de evitar futuros riesgos, se presentan estrategias de desarrollo, presentadas durante el proceso de desarrollo. El código es probado varias veces hasta alcanzar el resultado deseado y que no presente un riesgo para otros pasos de desarrollo (60).

Planificar

Se revisa el proyecto para verificar si los objetivos de un ciclo se han podido cumplir y, posteriormente, se toma la decisión de continuar con un ciclo posterior de la espiral (84). En caso de continuar, se desarrollan los planes para la siguiente fase. En caso contrario, se buscan soluciones para identificar los fallos del ciclo (60).

El modelo en espiral no establece un orden de forma estricta, se puede combinar en cualquier momento con otros métodos de procedimiento, por lo que las fases no tienen que presentarse en cada ciclo.

2.1.2. Metodologías Ágiles

El desarrollo ágil de software es una metodología de gestión de proyectos adaptativa, que permite llevar proyectos de desarrollo de software que se adapten a cambios y evolucionen en forma conjunta con el software (9). Son flexibles, pueden ser modificadas

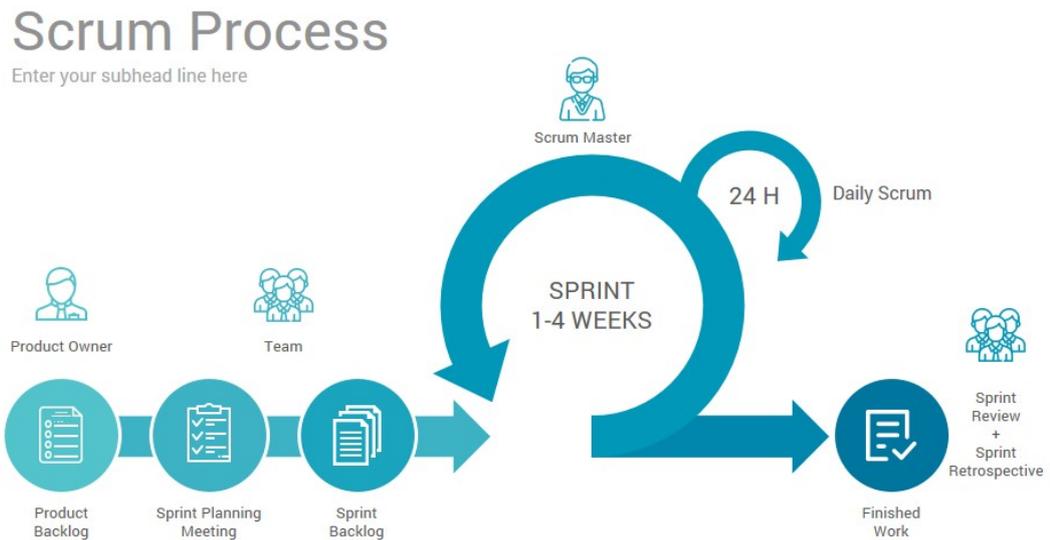


Figura 2.3: Ciclo de desarrollo de la metodología Scrum.

para que se ajusten a la realidad de cada equipo y proyecto, ya que se subdividen en proyectos más pequeños, donde son tratados de manera independiente y se desarrolla un subconjunto de características durante un periodo corto (77).

En los últimos años, la aplicación de este tipo de metodologías ha estado en auge, debido a los beneficios generados con su uso. De las más populares son *SCRUM* y *Kanban*

2.1.2.1. SCRUM

Según Lévano (67) es una metodología o marco de trabajo para la gestión de proyectos grandes y complejos. Se basa en ejecutar el proyecto en iteraciones de entre dos y cuatro semanas, esto es conocido como "Sprints", con una duración fija "Timebox".

En la Figura 2.3 se muestra el ciclo de vida de SCRUM, consta de fases de análisis y planificación, etapas para los Sprints, análisis y revisión de lo realizado para pasar a la siguiente fase del proyecto.

Roles de trabajo

Los equipos en Scrum están divididos en tres roles

- Scrum master: Responsable de asegurar los procesos

- Dueño de producto: Se encarga de maximizar el valor del producto
- Equipo: Encargados de realizar el trabajo

Ceremonias

Antes y después de llevar a cabo los sprints, se realizan una fase de análisis y planeación, llamadas ceremonias (reuniones), se realizan 6 en total contando a los Sprints, con el fin de asegurar el cumplimiento de objetivos (9).

- Reunión de planificación de la entrega
- Reunión de planificación del sprint
- El Sprint
- Reunión diaria
- Reunión de revisión
- Reunión de retrospectiva

Artefactos

Scrum utiliza 4 herramientas aplicables a los procesos anteriores llamadas artefactos

- Backlog de producto: Lista de lo que requiere el software
- Backlog de Sprint: Lista de tareas necesarias para convertir en backlog de producto
- Scrum taskboard: Tablero que permite una gestión de los ítems de trabajos pendientes, en curso y terminados de un mismo Sprint
- Diagrama de Burndown: Gráfico que permite observar y analizar el progreso de los ítems del backlog de producto

Cada uno de estos elementos, roles, ceremonias y artefactos, son parte fundamental de SCRUM, lo que permite una gestión dinámica de los proyectos, principalmente aquellos que sean grandes y complejos, ya que esta metodología permite dividir la carga de trabajo en fases más sencillas de trabajar.

2.1.2.2. Kanban

Metodología visual que divide el trabajo en bloques, permitiendo visualizar el flujo de trabajo del proceso de desarrollo de software (67). El trabajo se muestra en forma de tablero organizado en columnas, cada una de estas representa una etapa del trabajo (6).

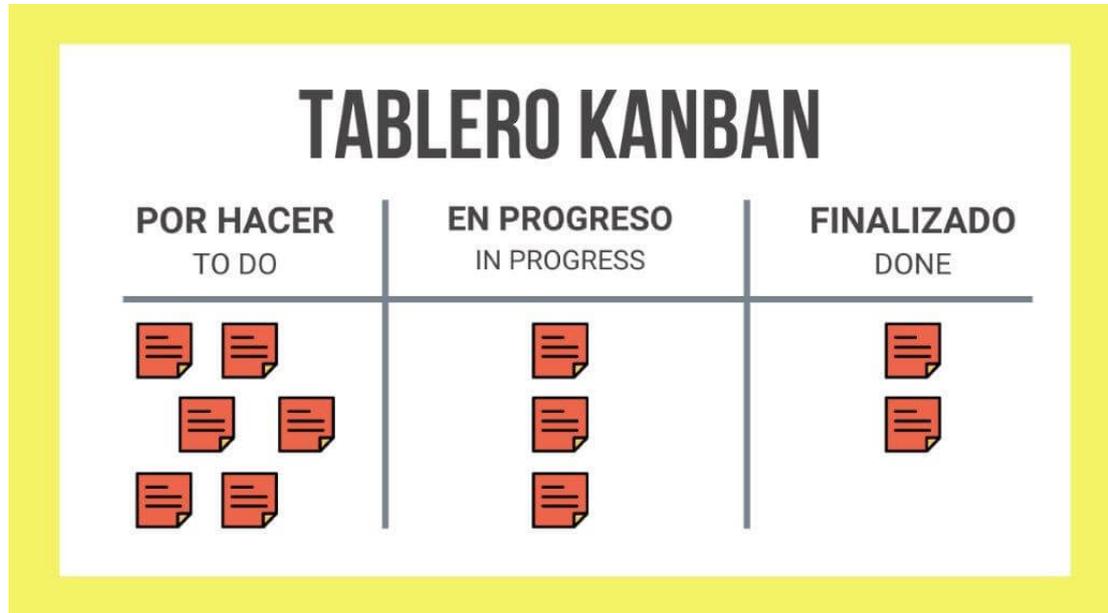


Figura 2.4: Representación de la metodología Kanban

Los principios de Kanban

Esta metodología presenta 4 principios que ayudarán a seguir una buena implementación de la metodología.

- Se adapta a cualquier proceso o flujo de trabajo actual, ya que es bastante flexible como para adaptarse a las prácticas centrales del equipo de trabajo
- Se enfoca en los cambios progresivos y de manera continua para lograr que los procesos evolucionen correctamente con el tiempo, ya que si se hacen cambios bruscos o grandes, esto puede afectar el funcionamiento del sistema.
- No hay roles integrados y puede funcionar con la estructura y procesos actuales del equipo.
- Se alienta a los miembros del equipo a participar, proponer nuevas ideas y emprender nuevas iniciativas de trabajo que permitan mejorar el flujo de trabajo.

Funcionamiento

Esta metodología cuenta con un funcionamiento sencillo que permite una fácil adaptación a cualquier flujo de trabajo. En la Figura 2.4 se muestra un ejemplo de cómo es representada esta metodología, mediante un tablero en donde se divide el flujo de trabajo en columnas.

Lo primero es crear un tablero en blanco, posteriormente dividirlo en diferentes columnas dependiendo del número de etapas que desee añadir el equipo de trabajo. Algunas de las columnas más comunes son:

- Nuevo
- Listo o prioritario
- En proceso
- En espera
- Finalizado

Se puede dividir el tablero en el número de columnas que se deseen según la organización del flujo de trabajo y del equipo.

Posteriormente agregar a cada columna las tareas correspondientes, son representadas mediante tarjetas y se puede agregar información como descripciones, contexto o solo una palabra que indique cuál es la tarea en específico. Conforme vaya avanzando el flujo de trabajo, se mueven las tareas por las distintas columnas hasta culminar con ella (6).

2.2. Metodologías de análisis de datos

El proceso de análisis de datos cuenta con metodologías que permiten un mejor análisis, procesamiento y obtención de conclusiones, a partir de una serie de fases que permiten obtener mejores resultados a partir de un proceso más ordenado.

La metodología Knowledge Discovery in Databases (KDD) es de las más usadas en el análisis de datos para poder identificar patrones novedosos, útiles y entendibles para el usuario, aunque también hay otras metodologías como la Sample, Explore, Modify, Model and Assess (SEMMA), metodología parecida a KDD, pero con otro enfoque sobre el análisis de los datos.

2.2.1. Metodología KDD

Proceso en el que se combinan descubrimiento y análisis, que consiste en extraer patrones en forma de reglas o funciones a partir de los datos (100). Proceso centrado en el usuario compuesto por varias etapas, con la capacidad de ser interactivo y guiado por las decisiones del usuario (78).

Según Timaran-Pereira (100) KDD cuenta con 5 fases que permiten la selección, procesamiento, análisis e interpretación de los datos. Las etapas son las siguientes:

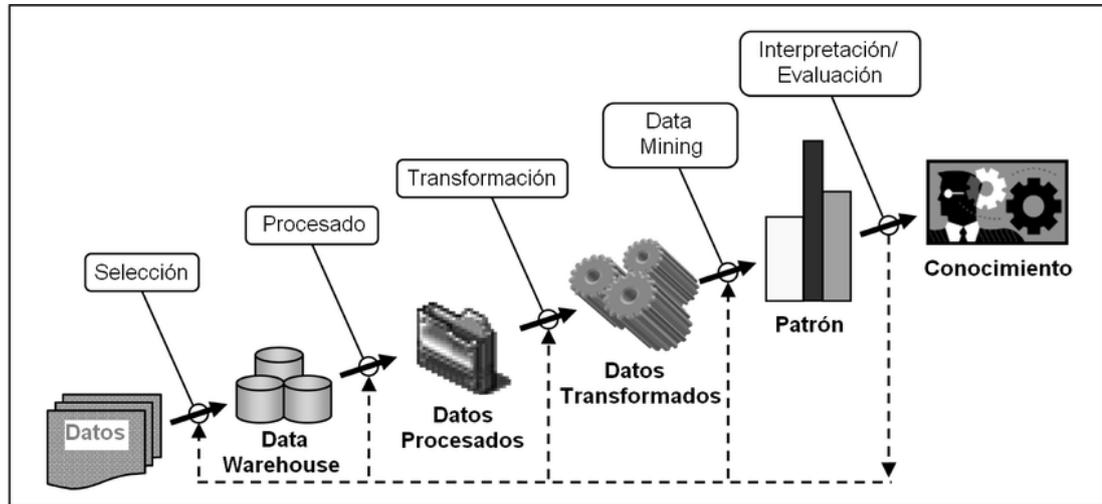


Figura 2.5: Etapas de la metodología KDD

- Selección
- Preprocesamiento/Limpieza
- Transformación/Reducción
- Minería de datos
- Interpretación/Evaluación

En la Figura 2.5 se muestra a detalle cuáles son las etapas de KDD y cómo son aplicadas a los datos.

Selección

Se seleccionan los datos más relevantes para el análisis, desde una o varias fuentes, con el objetivo de crear un almacén de datos que contenga la información que será utilizada para etapas futuras (30).

Preprocesamiento/Limpieza

Se realiza un análisis de la calidad de los datos, y un preprocesamiento que implica técnicas de limpieza como remoción de datos ruidosos, datos nulos, datos duplicados y aplicación de técnicas estadísticas para su reemplazo (100). Con el objetivo de mejorar la calidad del almacén de datos y tratar de mostrar el conjunto de datos de la forma más apta para siguientes fases (30).

Transformación/Reducción

Se buscan características que sean útiles para representar los datos, utilizando métodos de reducción de dimensiones o de transformación, para simplificar una base de datos de manera vertical u horizontal, o sea la eliminación de atributos que no sean necesarios o redundantes en el caso horizontal y en vertical se busca eliminar tuplas que sean idénticas o la discretización de valores continuos como la edad pero en formato de rango de edades (100).

Minería de datos

Selección y aplicación de métodos o algoritmos para la búsqueda de patrones en los datos. Para obtener conocimiento que pueda ser utilizado para la toma de decisiones informadas (100).

Interpretación/Evaluación

Se interpretan los patrones descubiertos y en algunos casos se retorna a etapas anteriores para más iteraciones del análisis de los datos. Puede incluir visualización de los patrones extraídos, remoción de patrones redundantes o irrelevantes y traducción de los patrones en términos entendibles para el usuario (100).

Es un proceso muy completo y efectivo para el análisis de los datos, ya que cada una de las fases está enfocada en los datos y principalmente en el usuario, para que sea adaptable a sus necesidades y pueda obtener conocimiento destacable que pueda aplicar en sus necesidades.

2.2.2. Metodología SEMMA

Sample, Explore, Modify, Model and Assess, más conocida como SEMMA, es una metodología enfocada en la minería de datos, cuenta con herramientas enfocadas en el desarrollo de un modelo de minería de datos enfocado al descubrimiento de patrones de negocio ocultos (20). Está conformada por 5 fases las cuales están definidas por las siglas de su acrónimo como se muestra en la Figura 2.6.

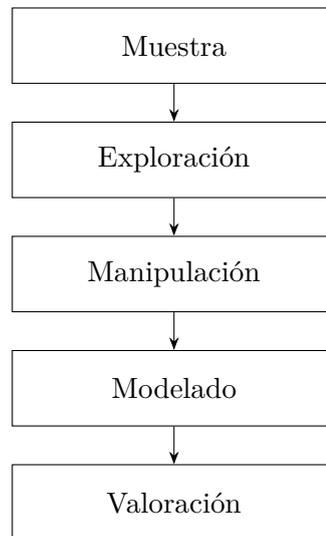


Figura 2.6: Fases de la metodología SEMMA

Muestreo

Extracción y selección de una población muestral representativa, ya que si no es así se invalida todo el modelo y los resultados no serán admisibles (40).

Exploración

Exploración de la información disponible con el objetivo de simplificar el problema y optimizar la eficiencia del modelo. Se suelen utilizar herramientas de visualización o técnicas estadísticas que ayuden a visualizar las relaciones entre variables.

Manipulación

Manipulación de los datos, con el fin de dar el formato adecuado a los datos para que puedan ser introducidos al modelo deseado, con el propósito de identificar las relaciones entre las variables explicativas y las variables objeto, permitiendo inferir un valor de estas con un nivel de confianza adecuado (20).

Modelado

Aplicación de diversas técnicas de minería de datos, según el tipo de tarea deseada, por ejemplo métodos estadísticos, métodos de agrupamiento y análisis de regresión o redes neuronales (40).

Valoración

Valoración de resultados mediante el análisis de bondad del modelo o modelos contrastados, con otros métodos estadísticos o con nuevas poblaciones muestrales (20).

2.2.3. Discusión

Tras un análisis de las diversas metodologías de desarrollo de software y análisis de datos presentadas, se empleará el modelo en cascada (Waterfall) para el desarrollo del software y la metodología Knowledge Discovery in Databases (KDD) para el análisis de datos en este proyecto. Esta elección se fundamenta en las características y ventajas de ambas metodologías en el desarrollo del proyecto.

El modelo en cascada cuenta con una estructura más lineal, la cual es más adecuada para proyectos de desarrollo de software. El enfoque secuencial del modelo en cascada facilita la gestión de los recursos y del tiempo, con plazos definidos y recursos limitados. Además, considerando que el proyecto será llevado a cabo por un solo desarrollador, la estructura más lineal del modelo en cascada es más óptima que las metodologías ágiles, las cuales están diseñadas para equipos más grandes y proyectos con mayor dinamismo.

Por otro lado, la elección de la metodología KDD para el análisis de datos se basa en que los procesos son más fáciles de adaptar al proyecto. KDD ofrece fases más completas que abarcan desde la selección inicial de los datos hasta la interpretación final de los resultados, proporcionando un mejor manejo de los datos para cada etapa del análisis. A pesar de tener fases bien definidas, KDD ofrece la flexibilidad necesaria para permitir iteraciones y cambios continuamente. Esta característica puede permitir que hallazgos iniciales conduzcan a nuevas preguntas o enfoques, requiriendo ajustes en el proceso de análisis. Otro aspecto destacable de KDD es su compatibilidad con una amplia gama de herramientas y técnicas de minería de datos. Esto proporciona la flexibilidad necesaria para aplicar técnicas de análisis de datos, permitiendo adaptar el enfoque analítico a las necesidades específicas de la presente investigación. Esta versatilidad es crucial en un proyecto académico, donde la profundidad y sofisticación del análisis son factores determinantes para la calidad de los resultados.

La sinergia entre el modelo en cascada para el desarrollo del software y KDD para el análisis de datos proporciona un desarrollo más estructurado y completo para el proyecto. Esta combinación permite abordar de manera sistemática tanto los aspectos de desarrollo de software como los de análisis de datos, permitiendo mantener un control sobre cada fase del proyecto. La claridad y estructura que ofrecen ambas metodologías facilitarán la documentación del proceso de desarrollo, además de proporcionar la flexibilidad necesaria para profundizar en aspectos específicos del análisis de datos, mientras se mantiene un enfoque estructurado en el desarrollo del software.

Marco tecnológico

En este capítulo se presenta un análisis detallado de las herramientas de desarrollo y tecnologías empleadas en la realización de este proyecto, en donde se analizarán los aspectos del desarrollo de la página web, bases de datos y el análisis de datos, cada una con sus herramientas y enfoques específicos.

Para el desarrollo web, la fase de diseño y prototipado se realizará con el software Figma. Esta herramienta facilitará la creación de prototipos funcionales que permitan una visualización clara y un funcionamiento preciso de la interfaz de usuario antes de pasar a la etapa de desarrollo. Para la implementación del frontend, se opta por el uso del framework React, que proporcionará las herramientas necesarias para construir interfaces dinámicas y responsivas. Esto mejorará notablemente la experiencia del usuario final, garantizando una interacción fluida con la plataforma. En el backend, el framework Django proporcionará una base sólida para desarrollar las funcionalidades del sistema, gestionar bases de datos y la integración del frontend y backend. En el caso de la base de datos, se opta por PostgreSQL como sistema de gestión de bases de datos. Proporcionará una estructura de datos confiable y escalable, ideal para manejar consultas complejas y grandes volúmenes de información. Esta elección permitirá una gestión eficiente de los datos y garantizará la integridad de la información en la aplicación.

En el proceso inicial de análisis de datos, se decantará por el uso de librerías de Python especializadas en el análisis de datos, que permitirán la extracción, limpieza y normalización de datos de manera más eficiente. Además del uso de herramientas de almacenamiento de datos como Dremio, permitiendo una integración con herramientas de visualización como Power BI. Esto fomentará análisis exploratorios detallados y la identificación de patrones que sentarán las bases para técnicas de análisis más complejas. En una fase posterior del análisis de datos se aplicarán técnicas de minería de datos, con librerías de Python que descubrirán patrones ocultos y relaciones más complejas dentro de los datos.

3.1. Herramientas de desarrollo para la página web

El apartado de desarrollo web es parte fundamental del proyecto, ya que es la parte funcional que el usuario final usará y es indispensable que esta sea intuitiva, funcional y fluida con el objetivo de que los jugadores puedan obtener la información que requieren de una forma rápida y sencilla.

Una de las primeras fases del desarrollo web es el análisis y diseño de las funcionalidades con las que debería contar el sistema; para ello se suelen utilizar herramientas de diseño o prototipado como Figma.

3.1.1. Diseño y prototipado - Figma

Figma es un software de diseño que permite a los usuarios crear interfaces de usuario (UI), prototipos interactivos, además de la posibilidad de colaborar en proyectos con otros usuarios de forma simultánea desde cualquier lugar (15).

Fue lanzada en 2016 como una herramienta novedosa para los diseñadores, ya que contaba con herramientas gratuitas además de la interacción y almacenamiento en la nube, dejando atrás a software especializado como Sketch o Adobe XD (37).

Es una herramienta muy completa que ofrece demasiadas opciones para crear un diseño de alta calidad, además de poder hacerlo funcional. El sitio web "HackBoss" (32) menciona las siguientes características de Figma como las más reconocidas:

Prototipado interactivo

Permite a los diseñadores probar la funcionalidad y experiencia de usuario antes de empezar el desarrollo e implementación final. Figma permite simular la navegación e interacciones de cada elemento como si fuera una web real.

Interfaz intuitiva

Ofrece una interfaz intuitiva y versátil, lo que genera una curva de aprendizaje mínima para aquellos usuarios que sean nuevos en este tipo de software, ya que ofrece herramientas y funciones para crear diseños con una alta facilidad.

Colaboración en tiempo real

Permite trabajar simultáneamente en un mismo diseño a varios miembros de un equipo, y estos pueden realizar cambios, comentar y ver las actualizaciones en tiempo real, lo que garantiza una mejor comunicación y procesos de desarrollo más ágiles.

Diseño responsivo y consistente

Ofrece herramientas para crear diseños responsivos, o sea que sean capaces de adaptarse a cualquier resolución de diversos dispositivos, además de opciones que permiten mantener una consistencia en todo el proyecto como la gestión de colores, tipografías y estilos de botones.

3.1.2. Desarrollo del Frontend - React

React es una biblioteca de JavaScript, que fue desarrollada por Facebook, con el fin de crear interfaces de usuario interactivas y eficientes, a partir de la creación de interfaces declarativas, o sea que se describe cómo debería presentarse una interfaz en un estado determinado y React se encargará de actualizar la interfaz cuando los datos cambian (76). Se basa en un paradigma llamado programación de componentes; un componente es una pieza con la que el usuario interactúa, estos componentes se crean a partir de archivos JSX, que permiten escribir HTML dentro de objetos de JavaScript. Las modificaciones que aplica React primero son aplicadas en una representación virtual del DOM, de forma que esos cambios se apliquen posteriormente al DOM real con los cambios necesarios (19).

3.1.2.1. Requerimientos

Para poder usar React es necesaria la instalación de diversas herramientas de desarrollo que permiten su correcto funcionamiento, a continuación se realiza una corta descripción de lo necesario para el correcto funcionamiento de React.

Node.js

Es un entorno de ejecución para JavaScript, es gratuito, de código abierto y multi-plataforma; este permite a los desarrolladores poder crear servidores, aplicaciones web, scripts o comandos. También incluye npm, el administrador de paquetes de Node, y npx, el ejecutor de paquetes de Node; estos últimos necesarios para el correcto funcionamiento de React y poder acceder a bibliotecas o herramientas externas a React (18).

Gitbash

Bash es una shell (aplicación de terminal que se utiliza como una interfaz mediante el uso de comandos) popular en Linux y macOS; la alternativa comúnmente usada en Windows es Gitbash, un paquete que instala Bash, algunas de sus utilidades más comunes y Git. Gitbash es una aplicación que ofrece una capa de emulación para una experiencia de líneas de comando para Git (7).

3.1.3. Desarrollo del Backend - Django

Es un framework de desarrollo para el lenguaje Python, muy utilizado para el desarrollo de páginas web, conocido por ser de código abierto y gratuito, lo que ha generado una gran comunidad de desarrolladores que constantemente comparten mejoras o recursos para este framework (43).

Django es muy utilizado por su estructura que permite un desarrollo más fácil, ya que este agrupa diferentes funcionalidades como autenticación, recuperación de información de una base de datos o la administración de cookies, en diferentes módulos reutilizables, lo que permite escribir código de manera más eficiente y con tiempos de desarrollo más cortos (98).

3.1.3.1. Arquitectura Modelo-Vista-Template (MVT)

Django administra las solicitudes entre cliente-servidor y servidor-cliente mediante la arquitectura Modelo-Vista-Template (MVT).

Modelo

Los modelos actúan como un enlace entre la base de datos y el código del lado del servidor, estos modelos contienen las operaciones y consultas necesarias para poder interactuar con la base de datos. Django convierte las tablas de la base de datos en clases u objetos en el código Python, esto es denominado como mapeo-objeto-relacional. Esto quiere decir que cada modelo se asigna a una tabla de base de datos y este tiene atributos que representan los campos de la base de datos.

Vistas

Con ayuda de los modelos, las vistas procesan las solicitudes provenientes del lado del cliente. Pueden tomar una solicitud como entrada y devolver una respuesta, que según el caso puede ser un código de error, una imagen o cualquier tipo de dato. Su funcionamiento es mediante un mapeador o despachador de URL, que ayuda a asignar funciones de vista a sus URL. Por ejemplo, si se desea que los usuarios de la página web vean una lista de algo en específico, se configura la ruta URL con las especificaciones de lo que se solicita y después el marco web Django envía los datos solicitados como respuesta.

Plantilla

Administran la presentación de la página web en el navegador, esto significa que se pueden utilizar estas plantillas para escribir en un estilo similar a HTML pero mediante Django. Los elementos que se pueden encontrar en estas plantillas son elementos estáti-

cos de salida HTML, como imágenes, botones y encabezados, y por otro lado la sintaxis especial para insertar contenido o datos dinámicos que cambian con cada solicitud.

3.1.3.2. Otros módulos de Django

A pesar de que el estándar en Django sea la arquitectura MVT, es posible usar otros módulos que permiten mejorar el desarrollo de manera eficiente, además permitiendo reducir tiempos en la construcción de funcionalidades similares a las de estos módulos.

Formularios

Muchos sitios web tienen implementados formularios que permiten ya sea el registro de usuarios o recabar información, es por ello que Django proporciona herramientas y bibliotecas que permiten aplicar esta funcionalidad de manera fácil y eficiente, permitiendo simplificar y automatizar el procesamiento de formularios, además de hacerlo de forma más segura que hacerlo de forma propia.

Autenticación de usuarios

La gran mayoría de los sitios web cuentan con sistemas de autenticación y autorización de usuarios; Django cuenta con herramientas que permiten la creación de usuarios, gestión de opciones según el rol o tipo de usuario o sesiones de usuario basadas en cookies.

Administración del sitio

Ofrece una página especial para la administración de su sitio, en donde pueden crear, editar o ver los modelos de datos del sitio.

3.2. Herramientas de desarrollo para la base de datos

Una base de datos es parte fundamental de un sistema actualmente, ya que el almacenamiento y uso de los datos de forma óptima y eficiente es una de las bases de un buen sistema. PostgreSQL es un sistema de base de datos que ofrece distintas funciones especializadas en escalabilidad y grandes flujos de datos, utilizada en aplicaciones móviles, web y de análisis de datos.

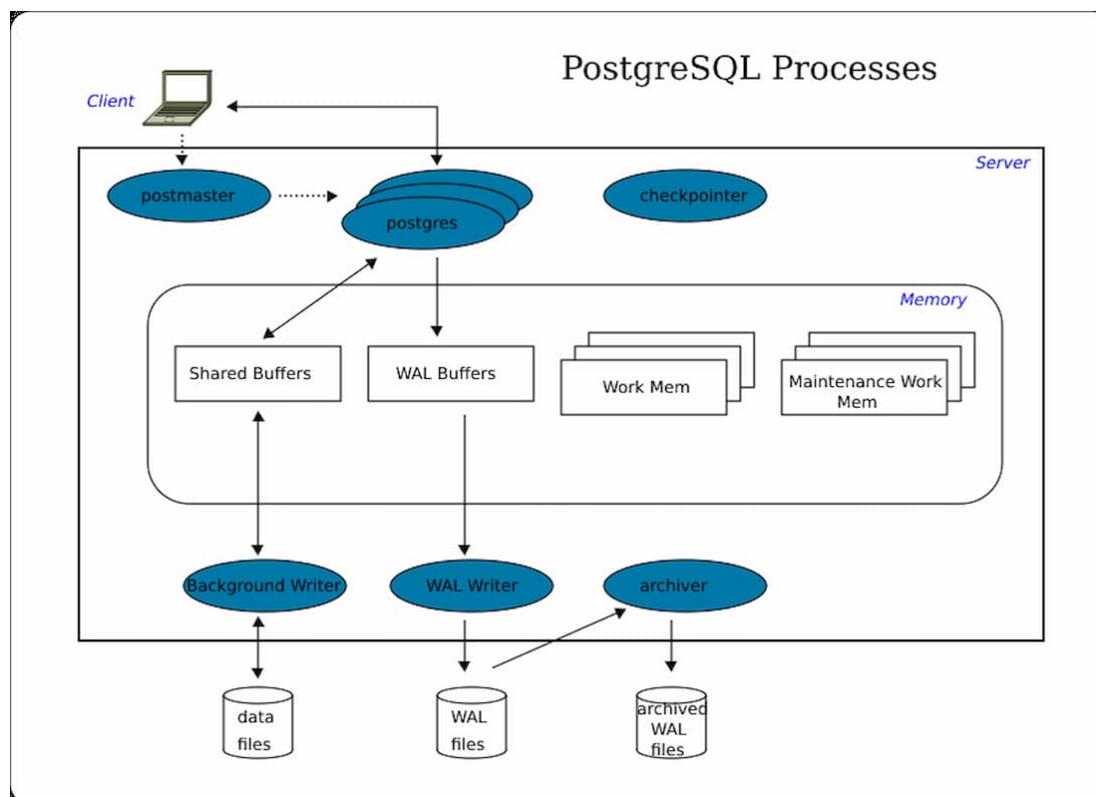


Figura 3.1: Arquitectura de un servidor PostgreSQL

3.2.1. PostgreSQL

PostgreSQL es un sistema de código abierto enfocado en la gestión de base de datos relacional de objetos conocida como ORDBMS (99), que proporciona soporte a funciones de SQL, como claves foráneas o subconsultas, además de funciones que son escalables reservadas para las cargas de trabajo de datos.

3.2.2. Arquitectura

PostgreSQL utiliza una arquitectura cliente/servidor, lo que permite que tanto el servidor como el cliente estén alojados en diferentes hosts y puedan tener comunicación entre ellos; el servidor gestiona los archivos de la base de datos, acepta y ejecuta conexiones simultáneas a la base de datos desde los sistemas del cliente (63).

Consiste en una memoria compartida, procesos ejecutándose en segundo plano y un directorio de datos. En la Figura 3.1 se muestra cada uno de los procesos con los que cuenta un servidor de PostgreSQL, a continuación se realiza una descripción de cada uno de estos elementos.

Memoria Compartida

Reservada para el almacenamiento caché del registro de transacciones y de la base de datos, sus principales elementos son:

- **Buffers compartidos:** Ayudan a minimizar las operaciones de entrada y salida del disco del servidor, esto permite acceder a los datos lo más rápido posible, mediante lotes de datos agrupados en memoria, lo que ayuda a minimizar tiempos de espera.
- **Buffers WAL:** Almacenan los cambios en la base de datos de manera temporal, indispensables para recuperar datos durante procesos de copias de seguridad y recuperación.
- **Memoria de trabajo:** Utilizada para operaciones de mapa de bits, ordenación, uniones de fusión, y uniones de hash, para escribir datos en archivos temporales.
- **Memoria de trabajo de mantenimiento:** Utilizada para operaciones de bases de datos.

Procesos en segundo plano

Cada proceso en segundo plano es único e integral para la gestión del servidor.

- **Proceso de punto de control:** Permite la creación de puntos de control, que son esenciales cuando la base de datos se bloquea, y estos se miden obteniendo la diferencia entre el último punto de control y el tiempo de parada de PostgreSQL.
- **Escritor de WAL:** Escribe y descarga los datos de la WAL en el buffer de la WAL, en el almacenamiento de la base de datos.
- **Archivador:** Copia los archivos de registro de la WAL a un directorio especificado.
- **Registrador/Recolector de registros:** Escribe un buffer WAL en el archivo WAL.

Directorio de datos

PostgreSQL está conformado por varias bases de datos, y juntas forman un clúster de bases de datos. Cuando se crea una base de datos, se realiza a través de las bases de datos de plantilla, que consisten en tablas del catálogo del sistema. Los datos que son necesarios para el clúster se almacenan en un directorio de bases de datos, conocido como PGDATA, el cual consta de varios subdirectorios:

- **Global:** Formado por las tablas del clúster.
- **Base:** Ubicación física del espacio de tablas por defecto; dentro se almacenan los catálogos del sistema.
- **PID:** Es el ID del proceso postmaster actual.

- PG-VERSION: Subdirectorio donde se almacena la información de la versión de la base de datos.
- PG-NOTIFY: Subdirectorio que consiste en los datos de estado de LISTEN/NOTIFY, archivos útiles para la resolución de problemas.

3.3. Herramientas de desarrollo para el análisis de datos

El análisis de datos requiere de diferentes herramientas que permitan obtener la información más relevante de nuestro conjunto de datos, iniciando desde procesos de aplicación de algoritmos para obtener datos y posteriormente prepararlos para el análisis mediante algoritmos de limpieza o normalización. También se pueden utilizar herramientas de visualización que permitan identificar de manera más intuitiva y visual los patrones más evidentes y después poder obtenerlos de forma más precisa con algoritmos enfocados en el análisis de datos. Python cuenta con diversas librerías enfocadas al análisis de datos, lo cual facilita su aplicación y obtener mejores resultados de forma más rápida; además, se puede complementar con otros software como Dremio o Power BI, para el almacenamiento y visualización de los datos.

3.3.1. Librerías de Python para análisis de datos

Uno de los lenguajes más enfocados en el análisis de datos y que cuenta con diversas herramientas o algoritmos que permiten obtener patrones o conocimiento relevante de una forma más sencilla es Python; cuenta con librerías que permiten obtener datos desde una API, manejar esos datos para almacenarlos en diferentes formas, como bases de datos, archivos .CSV o JSON, y poder aplicarles diferentes algoritmos para poder quitar o agregar información, con tal de que los datos estén preparados para la aplicación de algoritmos de minería de datos.

3.3.1.1. Librería requests

La librería Python Requests permite realizar solicitudes HTTP, facilitando la ejecución de solicitudes y operaciones HTTP hacia un servidor web, mediante una URL determinada; además, permite enviar información adicional al servidor a través de diferentes parámetros, encabezados y la codificación de las respuestas que genera el servidor. Se pueden realizar peticiones GET o POST, para pasar cabeceras en la petición o simulaciones de envío de datos de un formulario; además, se pueden enviar o recibir un JSON o gestionar códigos de respuestas a la petición HTTP que se realiza al servidor (44).

Instalación

La forma de instalación es muy sencilla, mediante un gestor de paquetes como pip, que ya viene por defecto en la instalación de Python, como se muestra en el Código 3.1

Enlace 3.1: Instalación de librería Requests mediante pip

```
$> pip install requests
```

Una vez puesto el comando, se instalará todo lo necesario para poder utilizar esta librería; posteriormente es necesario poner en el inicio del programa `import requests` para poder utilizar todas las funciones de la librería (62).

Ejemplo de uso petición GET y POST

Para realizar una petición GET, se invoca a la función `get()`, en donde se indicará la URL a la que se realizará la petición. En el fragmento de Código 3.2 se muestra un ejemplo básico de uso para una petición GET.

Enlace 3.2: Ejemplo de petición GET

```
import requests
resp = requests.get('https://www.example.com/')
```

En el caso de una petición POST, es similar a una petición GET, pero ahora se llama a la función `post()` e indicar en un parámetro `data`, un diccionario de datos con la información del cuerpo de la petición; posteriormente, request se encargará de codificar esos datos antes de realizar la petición y enviarlos. En el fragmento de Código 3.3 se muestra cómo se puede aplicar una petición POST.

Enlace 3.3: Ejemplo de petición POST

```
import requests
auth_data = {'email': 'example@mail.com', 'pass': '1234'}
resp = requests.post('https://mipagina.xyz/login/', data=auth_data)
```

3.3.1.2. Librería Pandas

Pandas es una biblioteca de software abierto, diseñada para el análisis y manipulación de los datos en el lenguaje Python, es ampliamente usada por su flexibilidad, características y facilidad de uso. Permite cargar, manipular, alinear o fusionar datos (24).

Funcionamiento

3. MARCO TECNOLÓGICO

Su funcionamiento se basa en tres estructuras de datos o arrays, siendo la primera, las series, una estructura unidimensional, como los arrays de cualquier lenguaje de programación. Después están los DataFrames (Tablas de datos bidimensionales), donde las columnas contienen valores de una variable y las filas contienen conjuntos de valores de cada columna. Por último también trabaja con Paneles o estructuras tridimensionales como un cubo de datos (56).

Instalación

Según el sitio web datacamp (23) la forma de instalar pandas es sencilla, solo hay que colocar un comando en la línea de comando del sistema para poder instalar todas las dependencias necesarias para poder utilizarlo, como se muestra en el Código 3.4.

Enlace 3.4: Instalación de la biblioteca Pandas mediante pip

```
$> pip install pandas
```

Ejemplo de uso - Convertir un CSV a JSON

Un ejemplo básico de las funcionalidades de Pandas es la importación de un archivo de tipo CSV para que se convierta en un JSON, el primer paso es importar la librería Pandas, como se muestra en el Código 3.5

Enlace 3.5: Importar Pandas a los proyectos de Python

```
import pandas as pd
```

Posteriormente cargar el archivo CSV, el cual será procesado para convertirse en un DataFrame por el algoritmo, como se muestra en el Código 3.6.

Enlace 3.6: Importar un archivo CSV con Pandas

```
csv_file = 'archivo.csv'  
df = pd.read_csv(csv_file)
```

Después el algoritmo se encarga de convertir ese DataFrame en un archivo de tipo JSON, el Fragmento 3.7 muestra el código que permite convertir el DataFrame en un JSON

Enlace 3.7: Convertir DataFrame en JSON

```
json_data = df.to_json(orient='records', lines=True)
```

Por último el usuario elige el nombre del archivo y se crea un mensaje para indicar que el archivo fue convertido correctamente, en el Código 3.8 se muestra la parte final del algoritmo para guardar el archivo creado

Enlace 3.8: Personalización de nombre del archivo JSON creado

```
with open('archivo.json', 'w') as json_file:
    json_file.write(json_data)
print("Archivo CSV convertido a JSON con éxito.")
```

3.3.1.3. Librería pyarrow

PyArrow es un enlace para poder usar Apache Arrow en Python, este es un formato de datos en columnas en memoria, que se usa para transferir datos de forma eficaz (72). Especifica un formato de memoria en columnas estandarizado, para datos planos o jerárquicos, organizado para operaciones analíticas en hardware más moderno y eficiente (93).

Instalación

Al igual que otras librerías en Python, esta librería se instala mediante el gestor de paquetes pip, mediante un comando en la línea de comandos del sistema, en el Código 3.9 se especifica el comando para instalarla.

Enlace 3.9: Instalar Pyarrow

```
$> pip install pyarrow
```

Usos de Pyarrow

- Mejoras en la conversión de pandas DataFrame a Spark DataFrame y viceversa, por ejemplo archivos de tipo PARQUET.
- Uso con Pandas UDF (UDF Vectorizado).

Ejemplo de uso - Convertir CSV a PARQUET

Uno de los ejemplos de uso más comunes es convertir un formato de datos como CSV al formato PARQUET; para poder aplicar este ejemplo es necesario importar la librería pyarrow especificando el formato a importar y exportar, como se muestra en el Código 3.10.

Enlace 3.10: Importar Pyarrow

```
import pyarrow.csv as pv
import pyarrow.parquet as pq
```

Posteriormente importar el archivo CSV que se desea convertir; después el algoritmo convertirá esos datos en una tabla Arrow, en el Código 3.11 se describe mejor el funcionamiento.

Enlace 3.11: Cargar archivo CSV mediante Pyarrow

```
csv_file = 'archivo.csv'
csv_table = pv.read_csv(csv_file)
```

Finalmente se especifica cuál es el nombre del archivo de formato parquet que será generado y se convertirá el archivo CSV a PARQUET; por último se genera un mensaje de confirmación sobre la conversión del archivo, como se muestra en el fragmento de Código 3.12.

Enlace 3.12: Convertir CSV a PARQUET

```
parquet_file = 'archivo.parquet'
pq.write_table(csv_table, parquet_file)
print(f"Archivo CSV convertido a Parquet: {parquet_file}")
```

3.3.1.4. NumPy

Es una librería que se especializa en el cálculo y análisis de datos, lo que la hace especial es la implementación de una nueva clase de objetos llamados arrays, que permiten representar colecciones de datos de un mismo tipo en diferentes dimensiones y funciones para la manipulación de los datos. Además, el procesamiento de los datos es mucho más rápido que otras funciones básicas de Python (3).

Arrays

Son estructuras de datos de un mismo tipo que están organizadas en tablas o cuadrículas de dimensiones distintas. La estructura de datos ndarray o n-dimensional es una estructura que tiene como característica que todos los datos son homogéneos, o sea, que son del mismo tipo (25).

La forma de crear un array es mediante la función `np.array()`, a partir de una lista o tupla ya definida. El número de dimensiones dependerá de las listas o tuplas que estén anidadas en la lista.

- Vector: Array de una dimensión que consta de una lista de valores.
- Matriz: Lista de listas de valores conocido como array de dos dimensiones.
- Cubo: Generado a partir de una lista de listas de listas, conocido como array de tres dimensiones.

No hay un límite en el número de dimensiones de un array, todo depende del número de listas que estén anidadas.

Instalación

La instalación es como cualquier biblioteca de Python, solo se necesita un gestor de paquetes como pip; posteriormente en la línea de comandos colocar el Comando [3.13](#) y se instalarán todas las dependencias para el correcto funcionamiento de NumPy ([29](#)).

Enlace 3.13: Instalar NumPy

```
$> pip install numpy
```

Ejemplo de uso

Importar la biblioteca al archivo de Python mediante el Código [3.14](#)

Enlace 3.14: Importar NumPy

```
import numpy as np
```

Posteriormente el Código [3.15](#) describe cómo es el funcionamiento de NumPy, primero se define un conjunto de datos, en este caso, una lista con tres números, después con la función `np.array()` tomando como parámetro la lista definida anteriormente. Obteniendo como resultado un array que puede ser utilizado para la manipulación y procesamiento de datos.

Enlace 3.15: Ejemplo de uso de NumPy

```
l = [5,12,16]
```

```
np.array(l)
```

```
Out[10]: array([ 5, 12, 16])
```

3.3.1.5. Matplotlib

Según el sitio web Aprende con Alf ([2](#)), Matplotlib es una biblioteca especializada en la visualización de datos que permite crear gráficos de dos dimensiones como:

- Diagramas de barras
- Histogramas
- Mapas de color
- Diagramas de dispersión o puntos
- Diagramas de líneas.

Está construido sobre la librería NumPy, lo que permite trabajar con vectores y arrays de mejor manera. Uno de los complementos más usados es `pyplot`, este módulo permite realizar funciones sencillas para la edición y visualización de gráficos, como la creación de figuras, trazado de líneas o inserción de texto (26).

Instalación

El Comando 3.16 describe la forma de instalar `matplotlib`, a través de un gestor de paquetes, mediante un comando de instalación (28).

Enlace 3.16: Instalar `matplotlib`

```
$> pip install matplotlib
```

Ejemplo de uso

En este ejemplo de uso se creará un gráfico que contendrá 3 puntos con distintas posiciones. El primer paso es importar la biblioteca `matplotlib` junto con el módulo `pyplot`, mediante el Código 3.17.

Enlace 3.17: Importar `matplotlib` junto al módulo `pyplot`

```
import matplotlib.pyplot as plt
```

Posteriormente, con la función `subplots()`, se podrá crear la figura y sus ejes; además, se definirá la ubicación de los puntos, tal y como lo describe el Fragmento 3.18 de código.

Enlace 3.18: Crear la figura y los ejes

```
fig, ax = plt.subplots()  
ax.scatter(x=[1, 2, 3], y=[3, 2, 1])
```

Por último, mediante la función `savefig()`, se podrá guardar el gráfico generado en un formato de imagen, en este caso PNG, y mediante `show()` mostrará el gráfico generado. En el Código 3.19 se describe la parte final de este proceso.

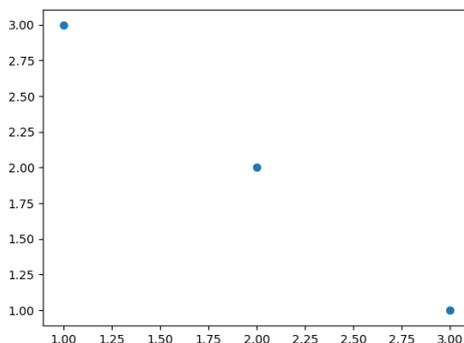


Figura 3.2: Gráfico creado mediante matplotlib

Enlace 3.19: Funciones para guardar y visualizar gráficos

```
plt.savefig('diagrama-dispersion.png')  
plt.show()
```

Como resultado, se obtiene un gráfico con un eje x, y con tres puntos en diferentes ubicaciones, tal y como se muestra en la Figura 3.2.

3.3.2. Data Warehousing

Un data warehouse es un sistema que permite la gestión de los datos, especialmente diseñado para realizar tareas de analítica de datos, ya que estos suelen estar diseñados para realizar consultas y tareas de análisis de datos, enfocados en grandes cantidades de datos (82).

Dremio es una plataforma que proporciona este tipo de servicios de data warehouse, facilitando a los analistas de datos una herramienta sencilla que permite unificar muchos datos de diferentes fuentes, además de que es una herramienta bastante flexible y con un costo menor a otras plataformas.

3.3.2.1. Dremio

Dremio es una plataforma abierta de lakehouse, basada en estándares que son impulsados por la comunidad como Apache Iceberg o Apache Arrow, permitiendo explorar y visualizar datos con tiempos de respuesta más eficientes que otras plataformas.

Puede ser utilizada con bases de datos relacionales como Apache Hadoop, MongoDB, Amazon S3, es compatible con SQL, además que permite realizar consultas mediante una interfaz web (35).

Arquitectura

La arquitectura de Dremio está basada en 4 capas que trabajan en conjunto, permitiendo mejorar la eficiencia y rendimiento en el procesamiento de consultas (101).

Capa de acceso a datos

Permite acceder a los datos desde distintas fuentes, como bases de datos, API REST o la conexión con otros servicios de almacenamiento de datos mediante los controladores de Dremio ODBC/JDBC.

Capa semántica

Encargada de unificar y transformar los datos, permitiendo obtener una vista unificada de los datos, además de poder aplicar transformaciones y consultas de datos de forma intuitiva.

Capa de aceleración de consultas

Acelera y mejora el rendimiento en la aplicación de consultas a los datos, utilizando Data Reflections, que permiten almacenar datos preprocesados en memoria para mejorar el rendimiento en posteriores consultas relacionadas.

Capa de presentación

Es la interfaz de usuario donde de forma intuitiva se pueden cargar datos, realizar consultas e implementar otras herramientas externas de visualización de datos.

Instalación

Dremio puede ser instalado en cualquier sistema operativo para computadoras, también puede ser desplegado en servicios de Cloud como Amazon Web Services. Además ofrece la posibilidad de acceder a todas sus funcionalidades mediante su despliegue a través de herramientas de contenedores como el software Docker.

Docker

Permite crear, probar e implementar aplicaciones de forma más sencilla, mediante el empaquetado del software en unidades estandarizadas conocidas como contenedores, permitiendo implementar y ajustar la escala de aplicaciones rápidamente. Funciona como una máquina virtual, pero virtualiza el sistema operativo de un servidor, proporcionando comandos sencillos para poder crear, iniciar o detener contenedores (97).

Implementar Dremio mediante Docker

Según la documentación oficial de Dremio (34), para instalarlo mediante Docker al iniciar Docker, se tendrá acceso a una línea de comandos, la cual permitirá desplegar Docker de forma sencilla, mediante el Código 3.20 se podrá acceder a Dremio de forma sencilla.

Enlace 3.20: Desplegar Dremio con Docker

```
docker run -p 9047:9047 -p 31010:31010 \  
-p 45678:45678 -p 32010:32010 -e \  
DREMIO_JAVA_SERVER_EXTRA_OPTS=\  
-Dpaths.dist=file:///opt/dremio/data/dist \  
dremio/dremio-oss
```

Una vez completada la implementación solo basta con iniciar el contenedor y acceder mediante el navegador a <http://localhost:9047> para poder obtener acceso a la aplicación web de Dremio.

3.3.3. Visualización de Datos

Una parte importante del análisis de datos es la visualización de ellos mediante gráficos y la aplicación de distintas técnicas estadísticas, permitiendo obtener mejor información de una forma más visual, dando un mejor entendimiento de los datos y cómo se relacionan entre sí, antes de pasar a la etapa de aplicación de minería de datos en donde se podrán obtener las relaciones de forma más precisa.

Power BI es una potente herramienta de visualización de datos, permitiendo la integración de data warehouse como Dremio, unificando lo mejor de estas herramientas: el procesamiento eficaz de los datos y la visualización de los mismos.

3.3.3.1. Power BI

Es un conjunto unificado de servicios de software, aplicaciones y conectores que permiten convertir orígenes de datos sin relación entre sí en información que sea atractiva visualmente, interactiva y accesible (73).

Permite hacer cálculos y gráficos de forma intuitiva; además, estos cálculos se pueden reutilizar según lo necesite el usuario, facilitando el acceso a los datos y librarse de posibles errores humanos. Por otro lado, la interacción con tablas y gráficos admite al usuario poder manejarlas y modificarlas de manera específica para cada caso, según sus necesidades, de forma sencilla (10).

Implementación con Dremio

La implementación con Power BI es muy sencilla, debido a las herramientas de integración que tiene Power BI por defecto. Al crear un nuevo proyecto, se mostrará la opción de añadir datos desde diferentes fuentes o servicios; dentro de esos servicios viene Dremio. Al seleccionar esta opción, pedirá ingresar datos como el dominio del servidor y las claves de acceso configuradas al iniciar Dremio por primera vez. Una vez realizada la conexión, el siguiente paso es elegir cuál es el conjunto de datos que se desea importar; por último, cargará todos los datos y permitirá aplicar todas las herramientas de visualización que ofrece Dremio (33).

3.3.4. Técnicas de Minería de Datos

La etapa de minería de datos consiste en aplicar diversos algoritmos al conjunto de datos, que permitan la obtención de conocimiento que pueda ser utilizado para descubrir patrones ocultos. Dependiendo del resultado que se desea obtener, se aplicarán diversos algoritmos según el tipo de modelo y método que se desee aplicar a los datos, por ejemplo para obtener reglas de asociación o clustering.

3.3.4.1. Biblioteca Apyori

Biblioteca creada específicamente para la implementación del algoritmo Apriori en Python, con herramientas que permiten leer y manipular datos, generar reglas de asociación y evaluar su calidad. Es muy utilizada ya que ayuda a obtener información sobre las relaciones estructuradas entre diferentes elementos involucrados de manera sencilla (22).

Instalación

Su instalación es sencilla, solo basta con poner en la línea de comandos del proyecto el Código 3.21, y se instalarán todas las dependencias necesarias para poder usarlo (36).

Enlace 3.21: Instalar Apyori

```
$> pip install apyori
```

3.3.4.2. Biblioteca Scikit-learn

Es una de las librerías más completas y especializadas en algoritmos de minería de datos, contando con algoritmos de clasificación, regresión y clustering. Es compatible con otras librerías especializadas en el análisis de datos, como Pandas o NumPy (95).

Es bastante conocida por las funciones que ofrece, como su robusto preprocesamiento de datos, ideal para procesar datos antes de aplicar los algoritmos que ofrece.

También cuenta con evaluaciones de la capacidad predictiva de los modelos a través de validación cruzada y curvas de aprendizaje (94).

Instalación

La forma de instalación es muy sencilla, en la línea de comandos colocar el siguiente Comando 3.22, esto permitirá instalar las siguientes dependencias:

- NumPy
- SciPy
- threadpoolctl

Dependencias necesarias para poder aplicar las distintas funcionalidades que puede proveer esta biblioteca (1).

Enlace 3.22: Instalar scikit-learn

```
$> pip install scikit-learn
```


Estado del arte

La aplicación de técnicas de minería de datos o uso de inteligencia artificial en el campo de los videojuegos no es algo nuevo; diversas investigaciones han demostrado cómo se pueden implementar soluciones que permitan obtener conocimiento que facilite la toma de decisiones informadas. Estas investigaciones principalmente se enfocan en el uso de la inteligencia artificial en videojuegos multijugador competitivos, especialmente en el área de los deportes electrónicos, con el fin de realizar un análisis sobre los comportamientos y estadísticas de jugadores profesionales.

4.1. eSports, un enfoque basado en inteligencia artificial

Este estudio realizado por Torres (102) hace un análisis al videojuego Counter-Strike: Global Offensive (CS:GO), con el objetivo de encontrar reglas, patrones y estadísticas que permitan comprender el comportamiento de jugadores profesionales en diferentes situaciones.

La metodología utilizada para el análisis de datos fue CRISP-DM, una metodología que permite, mediante una serie de pasos, realizar un proceso de selección de datos para la aplicación de un preprocesamiento y algoritmos de minería de datos que permitan realizar una evaluación e interpretación de los datos obtenidos.

El origen de los datos proviene de 400 grabaciones de partidas profesionales de la ESL Pro League, obtenidas a través de la página HLTV, una página que permite obtener información y estadísticas sobre las competiciones del CS:GO. Las grabaciones fueron procesadas por la librería demoInfo, que permite procesar archivos de tipo .DEM. En un primer preprocesamiento y limpieza de datos se obtuvieron un total de 9,202 registros con 25 atributos, en donde se incluyen datos como dinero disponible, supervivientes por bando, tiempo de rondas, rondas ganadas, entre otras que describen las principales mecánicas del videojuego.

4. ESTADO DEL ARTE

Una vez obtenidos los datos que servirán para realizar el análisis estadístico mediante la herramienta Weka, se obtuvo cuáles fueron las partidas jugadas en cada mapa durante 2019, en donde se obtuvo que el mapa de vértigo es el menos jugado con solo 7.24 % del total de todos los mapas jugados. Posteriormente, se obtuvo que en el mapa de Dust 2 existe un desbalance en el juego con respecto al rol de cada equipo, donde se encontró que el lado de Counter-Terrorist consigue un número mayor de victorias en las rondas que el bando Terrorist.

En otra etapa de la aplicación de algoritmos y técnicas de minería de datos, se hizo uso del programa RapidMiner, el cual está basado en algunas de las características y funcionalidades más notables de Weka. Los algoritmos aplicados con este software fueron:

- NaiveBayes
- Logistic Regression
- Generalized Linear Model
- Fast Large Margin
- Deep Learning
- Decision Tree
- Random Forest
- Random Forest

El objetivo de aplicar estos algoritmos fue realizar comparaciones de resultados entre ellos.

De la aplicación de estos algoritmos se obtuvo que, en caso de conseguir la primera eliminación, te permite obtener un 74.8 % de probabilidades de ganar la ronda. Además, que la activación de la bomba es otro de los factores importantes para la victoria.

Derivado de la comparación de resultados obtenidos al aplicar estos algoritmos, se obtuvo que el algoritmo Gradient Boosted Trees fue el que mejor precisión obtuvo con un 98.7 %. Por otro lado, NaiveBayes fue el más rápido pero con una precisión más baja, siendo de 54.1 %. Por último, el que tuvo un rendimiento más equilibrado entre la velocidad y precisión con un 95.8 %.

Del análisis de este estudio se pudo obtener información sobre qué técnicas de minería de datos pueden ser aplicadas en los videojuegos, específicamente en el campo de los deportes electrónicos. Además, da a conocer los resultados obtenidos y cómo podría ayudar a los jugadores en sus partidas a tomar mejores decisiones y conocer mejor el juego.

4.2. En busca de patrones de comportamiento entre jugadores de League of Legends

El trabajo de Mateos (70) se enfoca en analizar los patrones de comportamiento de jugadores profesionales de League of Legends, utilizando los modelos ocultos de Márkov, en donde se busca estudiar las secuencias de eventos y decisiones que pueden significar una victoria o derrota.

El conjunto de datos obtenido fue a partir de la plataforma Kaggle, una plataforma dedicada a almacenar y ofrecer datos públicos con el fin de realizar tareas de análisis de datos, diseñada para expertos en el análisis de datos o quienes deseen experimentar. Del conjunto de datos obtenidos se registraron un total de 7,620 partidas profesionales.

El conjunto de datos se enfoca en los eventos que ocurren en el juego, por ejemplo las torres destruidas o dragones eliminados; cada uno de estos eventos contiene información sobre el período de tiempo en el que sucedieron, además de información sobre el oro total con el que contaba cada equipo y cuál era su diferencia sobre el equipo enemigo, además de información sobre qué equipos ganaron o perdieron una partida. Posteriormente se realizó un preprocesamiento de los datos, en donde primero se eliminaron aquellas partidas que contaban con un sistema antiguo de la generación de dragones, con el fin de solo dejar las partidas con el sistema más actual. Después, los datos fueron divididos en 8 eventos que ocurren continuamente en una partida. Los eventos elegidos fueron:

- Dragón elemental asesinado
- Torre exterior destruida
- Torre interior destruida
- Inhibidor destruido
- Torre de nexo destruida
- Herald
- Dragón anciano
- Barón Nashor

Cada uno de estos eventos fue tratado a nivel de equipo, o sea, que se toma en cuenta qué equipo realizó el evento correspondiente. Otros de los atributos obtenidos fueron la diferencia de oro por minuto entre cada equipo y si esos equipos resultaban ganadores o perdedores en una partida.

Una vez hecho el preprocesamiento, en el apartado del análisis de los datos se utilizó como base el lenguaje R, un lenguaje de programación enfocado en el análisis estadístico, el cual permitió aplicar los siguientes algoritmos para el análisis de datos para entrenar y aplicar los modelos ocultos de Márkov:

- dplyr: Manipulación de data frames
- tidyr: Organización de datos
- purrr: Manejo de funciones y vectores
- TraMineR: Análisis de datos secuenciales
- seqHMM: Entrenamiento de modelos ocultos de Márkov
- ggplot2: Visualización
- igraph: Análisis de redes y grafos

Como resultado de la aplicación de los modelos de Márkov, se identificó que un factor importante que puede definir el futuro de una partida es el control temprano de la Bot Lane, ya que esta permitirá tener un acceso temprano a objetivos del juego como dragones y movilidad por el mapa. También demostró que los jugadores que resultaban ganadores seguían una secuencia clara respecto a los eventos del juego, en donde se priorizan primero los objetivos como dragones y después las torretas, ya que otorgan una diferencia de oro considerable contra equipos que siguen un ciclo de eventos sin valor estratégico. Otro punto a destacar es que se pudo identificar mejor qué equipo tiene más probabilidad de ganar, según su posición en el mapa, ya sea lado rojo o azul. Por un lado, los equipos del lado azul tienen más probabilidades de ganar con partidas que sean más rápidas, mientras que los del lado rojo aumentan su probabilidad conforme una partida aumente su duración; esto se debe a que tienen una posición que les permite eliminar más objetivos como el Barón Nashor o el Dragón Anciano, los cuales aparecen en un punto muy avanzado de la partida, permitiendo ganar más oro y mejorar sus estadísticas de personajes, lo que hace evidente una clara ventaja sobre el equipo enemigo.

Del análisis de esta investigación se pudo identificar cómo es el manejo de los datos para el caso específico del videojuego League of Legends, enfocándose principalmente en los eventos que ocurren en el juego, más que en las estadísticas de los personajes u objetos usados. Además de identificar algunos algoritmos que permiten el preprocesamiento y análisis de los datos. Esto permitió observar cuáles eran los comportamientos o secuencias de jugadores profesionales ante eventos del juego como la aparición de monstruos de la jungla o la pronta destrucción de las torres enemigas.

4.3. Sistema de recomendación de League of Legends mediante agrupación y redes de neuronas

La investigación de Herreros (52) se enfocó en el diseño de un sistema de recomendación de estrategias, basadas en los patrones de juego de los usuarios, con el fin de mejorar su desempeño mediante técnicas de agrupación como K-medias y Affinity Propagation.

Los datos fueron obtenidos mediante la API de desarrollo de RIOT GAMES, la cual provee estadísticas detalladas de partidas y jugadores. Los datos se obtienen con base en el *accountId* del jugador, lo que permite obtener información individual sobre todas sus partidas, como:

- Asesinatos
- Asistencias
- Daño realizado

Además de información a nivel colectivo, en donde se obtiene:

- Torres destruidas
- Dragones eliminados
- Barones Nashor eliminados

Posteriormente, mediante el uso del software Weka, se aplicaron las técnicas de agrupamiento Affinity Propagation y K-medias, con el objetivo de identificar patrones similares entre partidas, logrando obtener en diferentes clusters los estilos de juego más comunes. Por ejemplo, jugadores que priorizan el daño directo al enemigo o aquellos que se enfocan en apoyo mediante control de visión o del mapa. Estos clusters sirvieron para entrenar un modelo de clasificación mediante algoritmos de predicción como Multilayer Perceptron (MLP) y J48, con el fin de obtener el mejor modelo según su precisión en la predicción de futuras partidas de un jugador, generando estrategias específicas basadas en su estilo de juego. Algunas de las estrategias obtenidas son:

- Estrategias de control de visión y del mapa.
- Estrategias de alta agresividad y combate directo.
- Estrategias enfocadas en la destrucción de torres.
- Estrategias de resistencia y peleas en equipo.
- Estrategias de apoyo.

Cada estrategia detalla las acciones que debe realizar el jugador según su estilo y efectividad en cada tipo de partida, lo que puede ayudar a los jugadores a tomar mejores decisiones acorde al estado del juego.

Por último, estas estrategias fueron implementadas en un sistema desarrollado en C# y Java, que permite al usuario conectar su cuenta de juego para que el sistema obtenga sus estadísticas de partidas. Esto sirve para que el modelo de predicción sugiera las estrategias más adecuadas, mediante estadísticas y gráficas que muestran el estilo de juego del usuario y la tasa de éxito para cada estrategia, facilitando al jugador identificar la estrategia que mejor se adapte a sus necesidades.

Esta investigación permitió identificar el manejo de datos a través de la API de desarrolladores de RIOT GAMES para obtener información detallada sobre partidas. Además, destacó los elementos más importantes para el análisis de datos, como estadísticas individuales o en equipo en League of Legends. El análisis y la aplicación de diferentes algoritmos permitió conocer cómo se relacionan los datos para generar clusters con características definidas, como estilos de juego, y cómo integrar esta información en un sistema que apoye a cualquier jugador a analizar sus partidas y recibir recomendaciones estratégicas personalizadas.

4.4. Data mining techniques in video games

El trabajo de Lora (66) aborda un análisis sobre el comportamiento de los jugadores en videojuegos multijugador del género MMORPG, específicamente *The Exiled Realm of Arborea* (Tera) y *World of Warcraft* (WoW), dos títulos reconocidos por reunir gran parte de la comunidad gamer debido a sus mecánicas y contenido constante. El objetivo es analizar mediante estrategias de medición y telemetría el comportamiento de los jugadores para mejorar mecánicas, ajustar la dificultad y personalizar la experiencia del usuario.

El método aplicado para la obtención de los datos fue la telemetría, un proceso de recolección de datos en tiempo real que captura la interacción del jugador con el juego. Lora identificó tres métodos principales para la recopilación de datos:

- **Eventos:** Registro de acciones específicas realizadas por el jugador.
- **Frecuencia:** Muestreo de datos en intervalos de tiempo predefinidos.
- **Manual:** Activación y desactivación de registros bajo condiciones específicas.

Estos métodos permiten obtener métricas para evaluar el comportamiento de los jugadores, la dificultad y el impacto del juego en la experiencia general.

En conjunto con la telemetría, este estudio aplica la metodología CRISP-DM para el análisis de datos, compuesta por las siguientes fases:

- **Comprensión del negocio:** Identificación de objetivos y necesidades del análisis.
- **Comprensión de los datos:** Procesamiento y análisis de las fuentes de datos recopiladas.
- **Preparación de los datos:** Limpieza y transformación de los datos.
- **Modelado:** Aplicación de técnicas de aprendizaje automático para extraer patrones.
- **Evaluación:** Análisis del desempeño de los modelos generados.
- **Despliegue:** Integración de los resultados en el sistema.

Una vez obtenidos los datos, se utilizó la herramienta TetrisAnalytics, una plataforma que analiza el comportamiento de jugadores en Tetris. Se generaron registros de partidas, como movimientos del jugador, acciones realizadas y tiempo de juego. Estos datos se procesaron con el software Weka, aplicando algoritmos como:

- **Clustering con K-medias:** Agrupación de jugadores según estilos de juego.
- **Máquinas de soporte vectorial (SVM):** Predicción del comportamiento del jugador.
- **Interpolación de datos:** Homogeneización de muestras de partidas de distinta duración.

Como resultado, se obtuvo una clasificación de jugadores en cada videojuego, detallando su comportamiento y perfiles. Se identificaron seis tipos de jugadores:

- **Élite:** Jugadores con puntuaciones altas en combate y logros, pero bajas habilidades en minería y plantación.
- **Atrasados:** Jugadores con bajo desempeño en todas las métricas.
- **Jugadores promedio:** Rendimiento medio en la mayoría de métricas.
- **Dependientes:** Jugadores con muchas interacciones sociales, buena puntuación en combate y economía.
- **Trabajador 1 y Trabajador 2:** Enfocados en habilidades económicas como minería y plantación.

Se identificó que los jugadores con mayor interacción social y mejor desempeño eran más propensos a continuar en el juego, mientras que aquellos con bajo rendimiento tenían mayor riesgo de abandono. Estos videojuegos sirvieron como base para aplicar técnicas de minería de datos y telemetría empleadas en el análisis del comportamiento del jugador, permitiendo obtener una clasificación según tiempo jugado, nivel de habilidad y eventos ocurridos. Esta investigación aporta avances en el análisis de datos aplicados a videojuegos multijugador, con el fin de obtener información detallada sobre los jugadores, su comportamiento y su relación con estilos de juego o niveles de habilidad.

4.5. Diseño de una herramienta de análisis de rendimiento de jugadores para equipos de alto nivel de eSports

El trabajo de González (49) consistió en el diseño y desarrollo de una herramienta de análisis de datos orientada a mejorar el rendimiento de jugadores de eSports, específicamente para el videojuego League of Legends (LoL). El objetivo principal fue proporcionar visualizaciones del rendimiento de los jugadores, desde perspectivas generales hasta análisis detallados, facilitando la toma de decisiones para optimizar el desempeño en dos momentos clave:

- Durante las partidas (fases de elección y descansos).
- Análisis post-partida para comprender resultados.

Los datos fueron extraídos mediante la plataforma `pandascore.co`, especializada en recopilar y estructurar información de distintos videojuegos, facilitando así su tratamiento posterior. Con estos datos se diseñó un Data Mart optimizado para el análisis del rendimiento de jugadores, diseñado para maximizar la eficiencia de las consultas. El modelo incluye:

- **Dimensión Campeón:** Identificadores, nombres, roles y posicionamiento en el meta actual.
- **Dimensión Versión de Juego:** Identificadores, versión del juego y región de la partida.
- **Dimensión Jugador:** Datos personales (nombre, rol, ranking), equipo y métricas de rendimiento (KDA medio, Performance Score).
- **Tabla de Hechos - Partidas Jugadas:** Duración, KDA (kills, deaths, assists), daño infligido, oro obtenido, CS (Creep Score), línea jugada, puntuación de visión, objetivos conseguidos e indicador de rendimiento.

Estos parámetros permiten realizar un análisis detallado del rendimiento de un jugador en partidas profesionales de League of Legends, trabajando con aproximadamente 1500 registros con diversas métricas y estadísticas. Para el análisis de datos se emplearon técnicas de Machine Learning con librerías de Python como Pandas y Scikit-Learn, enfocándose en identificar patrones de juego y rendimiento mediante:

- Análisis de dispersión: Evaluación de la distribución temporal de eventos.
- Cálculo de promedios por agrupación: Determinación de la duración media de partidas por equipo.
- Medidas de tendencia central y agrupación: Identificación de los equipos más exitosos y sus características comunes.

Además, se utilizaron herramientas de visualización como Power BI, facilitando la presentación de métricas clave para el análisis.

El sistema desarrollado cuenta con funcionalidades que permiten a los entrenadores acceder a información relevante sobre sus equipos de forma intuitiva, entre ellas:

- Autenticación: Inicio de sesión, registro de nuevos usuarios, creación de clubes y equipos.
- Menú principal: Acceso a estadísticas de partidas, jugadores, tendencias, información de otros jugadores y configuración de cuenta.
- Tendencias: Comparativas entre tendencias globales y del equipo, como el top 5 de campeones más jugados y los campeones con mayor porcentaje de victorias.
- Partidas: Listado de partidas recientes con estadísticas generales.
- Estadísticas detalladas: Información completa de partidas específicas.
- Jugadores: Estadísticas individuales.

Este proyecto permitió extraer información valiosa sobre el rendimiento de los jugadores, la efectividad de sus estrategias y áreas de mejora. Así, fue posible identificar patrones de juego, las mejores tácticas de cada equipo y comparativas de desempeño, brindando a los entrenadores una herramienta accesible y comprensible para tomar decisiones informadas durante el desarrollo de partidas profesionales. Este trabajo sienta las bases para el desarrollo de sistemas que integren la extracción de datos y técnicas de minería para la visualización detallada de partidas profesionales de LoL.

4.6. Discusión

Las investigaciones analizadas han contribuido a establecer un contexto sobre el uso de técnicas de minería de datos e inteligencia artificial aplicadas a videojuegos. A

4. ESTADO DEL ARTE

partir de estas se recopiló un conjunto de elementos clave para el desarrollo de proyectos relacionados con minería de datos en eSports. La Tabla 4.1 sintetiza los factores más relevantes de cada estudio, destacando sus aportes y metodologías, que han servido de base para el desarrollo de soluciones en esta área.

Tabla 4.1: Comparativa de estudios sobre análisis de datos en eSports

| Estudio | Objetivo | Juegos Analizados | Metodología | Algoritmos |
|------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|---------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| eSports, un enfoque basado en inteligencia artificial, Torres (102) | Analizar comportamientos de jugadores profesionales e identificar patrones | CS:GO | CRISP-DM | NaiveBayes, Logistic Regression, GLM, Fast Large Margin, Deep Learning, Decision Tree, Random Forest, Gradient Boosted Trees |
| En busca de patrones de comportamiento entre jugadores de League of Legends, Mateos (70) | Identificar patrones de comportamiento y secuencias de eventos para la victoria | League of Legends | Análisis secuencial | Modelos ocultos de Márkov, dplyr, tidyr, purrr, TraMineR, seqHMM, ggplot2, igraph |
| Sistema de recomendación de League of Legends mediante agrupación y redes de neuronas, Herreros (52) | Diseñar un sistema de recomendación de estrategias basado en patrones de juego | League of Legends | Agrupación y clasificación | K-medias, Affinity Propagation, Multilayer Perceptron (MLP), J48 |
| Data mining techniques in video games, Lora (66) | Analizar comportamiento de jugadores para mejorar mecánicas y experiencia | MMORPG (Tera y WoW) | CRISP-DM, Telemetría | K-means, SVM, Interpolación de datos |
| Diseño de una herramienta de análisis de rendimiento de jugadores para equipos de alto nivel de eSports, González (49) | Diseñar herramienta de análisis para mejorar rendimiento de equipos profesionales | League of Legends | Data Mart, visualización de datos | Análisis de Dispersión, Promedios por Agrupación, Medidas de Tendencia Central |

Aplicación de metodologías de desarrollo

La metodología en cascada permitió el desarrollo del sistema experto junto con la metodología en cascada para el análisis de datos, permitiendo obtener una sinergia para complementar el apartado del desarrollo de un sistema experto en un ambiente web, además del análisis de datos mediante técnicas de minería de datos. Llevando de manera estructurada una serie de procesos que permitieron llevar a cabo el desarrollo del sistema con los requerimientos necesarios para su correcto funcionamiento.

5.1. Análisis de requisitos

El desarrollo de un sistema experto que ayude a la selección estratégica de campeones en el videojuego League of Legends (LoL) responde a la creciente necesidad de los jugadores de herramientas que les permitan obtener más información y ayudas sobre el juego y sus elementos, ya que, al ser un videojuego complejo, los usuarios con conocimientos menores tienden a tener una mala experiencia de juego frente a usuarios con mayor experiencia y conocimientos básicos, lo que limita su mejora. El objetivo principal del sistema experto es brindar recomendaciones a los jugadores de LoL a partir de su rol dentro del juego, estilo de juego y nivel de habilidad. Esto permite obtener las variables más influyentes en un usuario. El sistema experto tendrá como base todos los datos recabados y procesados a partir de una competición profesional de LoL de alto nivel, conocida como League of Legends EMEA Championship (LEC). Todos los datos obtenidos de esta competición son proporcionados por la empresa RIOT GAMES, mediante su API de desarrolladores que permite obtener información del juego, como datos, estadísticas generales o imágenes de los campeones o información sobre las estadísticas de una partida de LoL.

5.1.1. Alcance

El sistema abarcará la recomendación de los campeones más aptos de acuerdo con las características de cada jugador, obtenidas a partir de un simple formulario que

permita identificar las variables más importantes para el funcionamiento del sistema experto. De esta manera, las recomendaciones se adaptarán a las elecciones del jugador. Además, cada recomendación de campeón contendrá información sobre las runas e ítems más aptos para el campeón y las características del usuario.

5.1.2. Problemáticas actuales

LoL es un videojuego muy complejo y profundo, ya que cuenta con demasiadas mecánicas que son necesarias de aprender o tener conocimientos básicos para poder jugar, lo cual es un problema para cierto sector de jugadores que son muy nuevos y no tienen ningún tipo de experiencia en el juego, o jugadores que tienen cierta experiencia pero necesitan ayudas para actualizar sus conocimientos o mejorar sus habilidades. La raíz de la problemática en la que se basa el sistema experto es el sistema de selección de un campeón dentro de una partida, además de elegir las runas e ítems más óptimos para dicho campeón, ya que el tiempo para elegir es muy limitado y son demasiadas opciones que pueden llegar a confundir al jugador, impidiéndole optar por un personaje que se adapte más a sus necesidades y obligándose a elegir una opción rápida sin tener un fundamento en sus decisiones.

5.1.3. Requisitos funcionales

El sistema contará con varios módulos interconectados que permitirán al usuario navegar en la página y obtener las recomendaciones de forma sencilla y rápida. El primer módulo será para la recopilación de la información del usuario, en donde, mediante un formulario de tres preguntas sobre el rol, estilo de juego y experiencia, el usuario tendrá la oportunidad de elegir las opciones que más se adapten a él. El segundo módulo es el apartado de recomendaciones, en donde, mediante reglas de asociación precargadas, se mostrarán los cinco campeones más adecuados con información sobre las runas e ítems más óptimos para el campeón. Estas sugerencias estarán basadas en las respuestas recibidas del formulario y, mediante la base de conocimientos y el motor de inferencias del sistema, se mostrarán las opciones más relevantes. Por último, el módulo de información detallada, en donde se mostrará a detalle la información del campeón elegido, así como datos de partidas anteriores y estadísticas relevantes.

Cada uno de estos módulos interconectados fungirá como el sistema experto que permita obtener recomendaciones personalizadas de forma fácil y rápida, permitiendo a los usuarios que estén dentro de una partida elegir lo que más se adapte a sus necesidades en un corto tiempo.

5.1.4. Requisitos no funcionales

En el apartado de requisitos no funcionales, es importante que el sistema sea intuitivo y fácil de usar para cualquier tipo de usuario, permitiendo que sea comprensible

para usuarios con nula, poca o mayor experiencia con este tipo de sistemas. Por otro lado, es primordial el rendimiento del sistema, ya que es necesario que el funcionamiento de todos los componentes sea rápido y optimizado y las sugerencias sean mostradas de forma inmediata. Por último, es importante garantizar que a futuro el sistema sea capaz de adaptarse a las nuevas necesidades de los jugadores y a los cambios constantes en el juego, mediante actualizaciones constantes y rápidas que permitan mejorar el sistema.

5.1.5. Restricciones

Una de las principales restricciones del sistema es la constante actualización de los datos y adaptaciones conforme a la evolución del juego, ya que suelen haber cambios dentro del juego como estadísticas de campeones, personajes nuevos o mecánicas de los ítems y runas que han sido agregadas, modificadas o eliminadas, lo que puede afectar en la precisión de las sugerencias debido a que hay periodos de tiempo en los que la competición de donde se obtienen los datos está detenida por cuestiones de descanso o el inicio de otras competiciones de mayor importancia o jerarquía dentro del entorno competitivo de LoL, lo que puede significar que en esos periodos la información dentro del sistema no esté actualizada.

Otra de las restricciones más importantes es la capacidad de los recursos computacionales con la que se cuenta, debido a que la aplicación de técnicas de minería de datos, en específico a un gran volumen de datos, requiere de recursos computacionales con un alto rendimiento, que permitan obtener el conocimiento requerido de forma correcta y en poco tiempo.

5.1.6. Metas del sistema

La principal meta a corto plazo es la implementación de un sistema que permita a los jugadores obtener información sobre la elección de un personaje y objetos para ese personaje, basado en las características de los jugadores, mediante una interfaz intuitiva y fluida, con el fin de ayudar a los jugadores a tomar decisiones informadas.

A mediano plazo, se planean actualizaciones constantes con el fin de adaptar los datos conforme a las actualizaciones del juego, además de mejorar el sistema de elecciones mediante la implementación de mejores algoritmos de minería de datos que permitan obtener información más detallada y precisa, contar con apartados de estadísticas más detalladas de cada campeón y una sección para poder buscar a cualquier personaje del juego y obtener información básica del campeón sin necesidad de realizar el proceso de selección mediante el formulario del sistema experto.

A largo plazo, la idea es implementar algoritmos de *machine learning* que permitan un mejor análisis de los datos de forma automática y en tiempo real, de forma que el

conocimiento obtenido sea en base a las mecánicas más actualizadas del juego, además de implementar guías y tutoriales que permitan al usuario conocer mejor ese campeón.

5.2. Diseño

El diseño del sistema experto que permita la selección estratégica de campeones, runas e ítems, basado en las características del jugador, está basado en una arquitectura modular cliente-servidor. La elección de este diseño se fundamenta en el análisis de requisitos, en donde se especifica que es necesario un sistema estructurado que permita la recolección de la información del usuario y utilizarla para mostrar las recomendaciones de forma precisa y rápida según las necesidades del usuario.

5.2.1. Diseño de la arquitectura del sistema

La arquitectura del sistema tiene un diseño modular en donde cada componente se encarga de una función en específico del sistema, de forma que cada módulo está interconectado con otros componentes, permitiendo un diseño estructurado y principalmente intuitivo para los usuarios.

Módulo de recopilación de información

El primer módulo consta de un formulario de tres preguntas en donde se busca obtener las características más importantes de un jugador en el juego LoL.

- **Rol:** Permite obtener información sobre cuál es la posición preferida del usuario en una partida de LoL, a fin de solo sugerir personajes que tengan un mejor desempeño en esas posiciones, teniendo como opciones: Top Lane, Jungla, Mid Lane, Soporte y AD Carry.
- **Estilo de juego:** Describe cuál es el estilo de juego del jugador frente a otros jugadores dentro de una partida de LoL, con el objetivo de mostrar personajes que se adapten mejor a ese estilo. Las opciones son: Defensivo, Balanceado, Agresivo.
- **Experiencia:** Permite obtener información acerca del nivel de experiencia de un jugador, para poder mostrar recomendaciones de mayor o menor dificultad. Las opciones a elegir son: Principiante, Intermedio y Experto.

El objetivo de este formulario es obtener las variables que permitan al sistema experto mostrar la sugerencia más adecuada, permitiendo adaptarse a las características de cada jugador.

Módulo de recomendaciones

Este módulo está encargado de mostrar las cinco recomendaciones más relevantes y que se adapten a las características de cada jugador según las respuestas del formulario. Cada sugerencia está basada en reglas precargadas, que fueron obtenidas a partir de la aplicación de técnicas de minería de datos como el algoritmo Apriori, que permite la obtención de reglas de asociación de un conjunto de datos, además de una etapa de filtrado y selección a partir de algoritmos de *clustering*, a fin de obtener las reglas mejor valoradas.

Módulo de información detallada

Módulo que tiene como objetivo mostrar la información más relevante de cada sugerencia, con detalles como información básica del campeón, runas e ítems adecuados para ese campeón y estadísticas sobre partidas anteriores donde ese personaje haya tenido una participación. Este módulo le permitirá al jugador obtener más información sobre el campeón sugerido, a fin de asegurarse de que puede serle de utilidad en una partida.

5.2.2. Diseño de la base de datos

La estructura de la base de datos busca almacenar la menor información externa posible, es decir, solo las reglas de asociación, debido a que mucha de la información, como imágenes, estadísticas o datos en específico, se puede obtener directamente desde la API de RIOT GAMES, la cual proporciona cualquier tipo de dato respecto al juego. Esto beneficiará al rendimiento de la base de datos y del sistema, al contar únicamente con la información necesaria y acceder a una fuente especializada en la información requerida. Además, permite mantener los datos actualizados en caso de cambios en estadísticas o información general.

Tomando en cuenta lo anterior, la estructura para la base de datos se muestra en la Figura 5.1, en donde se observa un modelo de entidad-relación. Se cuentan con las siguientes cuatro tablas:

- usuario_sesión
- usuario_información
- recomendación
- reglas

Esta estructura contempla, en un futuro, almacenar información de inicio de sesión de los usuarios, lo cual permitirá crear relaciones para almacenar los datos recabados al responder el formulario y, posteriormente, realizar una recomendación que será guardada en la tabla de recomendación, relacionando las tablas de reglas y de información de sesión.

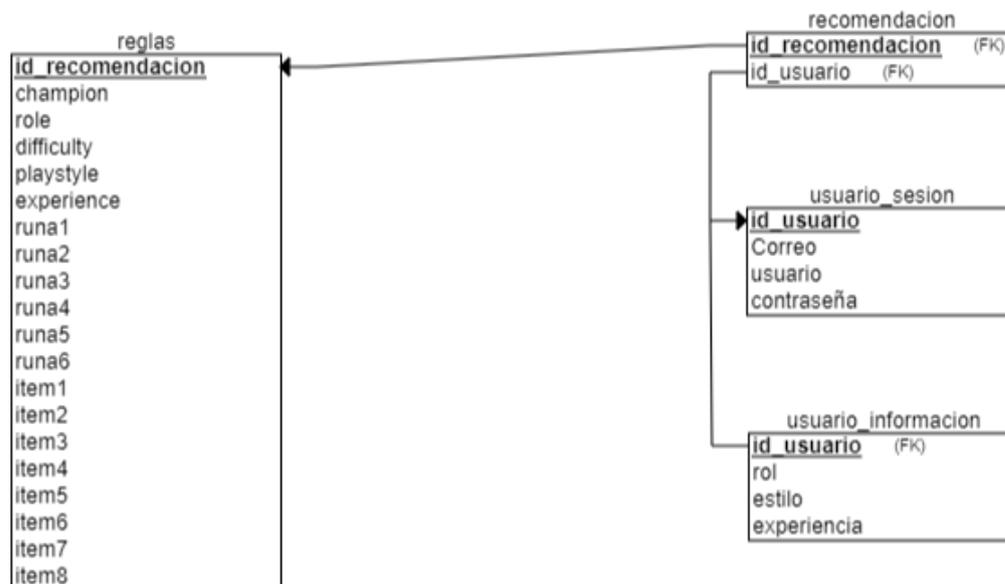


Figura 5.1: Diagrama Entidad-Relación de la base de datos.

5.2.3. Diseño de la interfaz de usuario (UI)

Este es uno de los apartados más importantes, ya que representa la parte del sistema con la que el usuario interactúa. Por ello, el diseño debe enfocarse en ofrecer una buena experiencia de usuario, que sea intuitiva y atractiva para los jugadores de LoL. Se hace uso de elementos iconográficos clásicos del juego, como íconos de campeones, runas e ítems, o imágenes alusivas a personajes del juego, con el objetivo de adentrar al jugador en un ambiente familiar.

Home-Page

La Home-Page será la primera presentación del sistema, en donde se mostrará información alusiva a las funcionalidades del sistema mediante textos descriptivos, videos que muestran el funcionamiento general o imágenes que ilustran las principales características. En la Figura 5.2 se muestra el diseño inicial de la home-page, en donde predominan colores oscuros y llamativos, alusivos a los colores del juego, además de personajes conocidos, de tal manera que el sistema sea atractivo y logre involucrar a los jugadores en un entorno visual similar al del juego.

Formulario sobre características del jugador

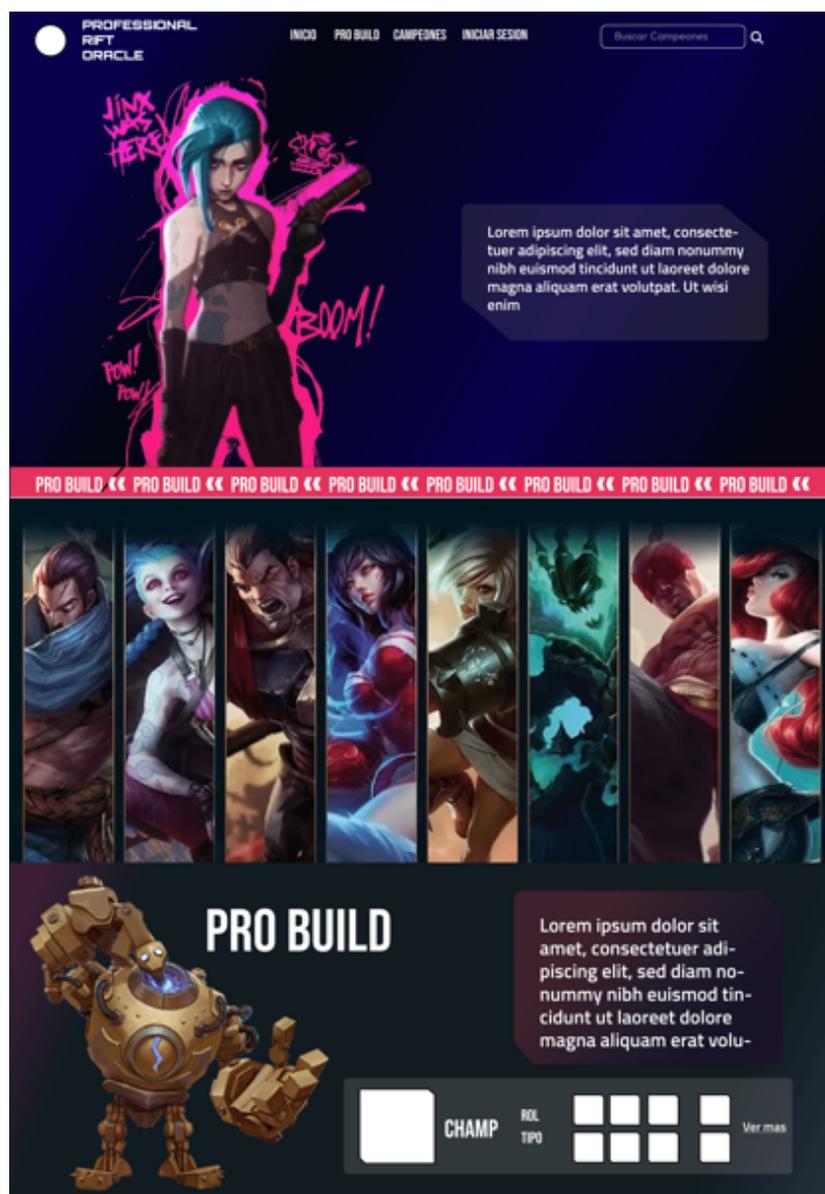


Figura 5.2: Diseño inicial de Home Page.

Es importante evitar que el usuario se sienta saturado o indeciso al responder este tipo de formularios. Un formulario con demasiadas preguntas o con opciones confusas puede provocar que el usuario abandone la página. Por ello, en la parte funcional se ha implementado un formulario por etapas: al responder cada pregunta, el sistema dirigirá automáticamente a la siguiente, evitando así la sobrecarga de información. Las opciones ofrecen descripciones detalladas y concisas, que permiten al usuario tomar decisiones más informadas. En cuanto al diseño, se continúa con el uso de colores oscuros y detalles llamativos que generan un contraste atractivo. En la Figura 5.3 se aprecia un ejemplo detallado de una pregunta del formulario con todos los elementos mencionados anteriormente.

Sugerencias

Este es el siguiente paso en el proceso de elección de campeón, en donde, de acuerdo con las respuestas del formulario, se mostrarán los cinco campeones más adecuados según las características del jugador. Este componente estará estructurado en cinco secciones donde se mostrará la imagen alusiva al campeón, sus datos más importantes como nombre y rol, además de los ítems y runas más adecuados. Otro apartado de este componente es la visualización de las respuestas del formulario, lo cual permitirá al jugador modificar sus respuestas en caso de querer elegir otras opciones que se adecuen mejor a lo que busca.

La Figura 5.4 muestra la estructura anteriormente mencionada del componente, en donde se detallan mejor los elementos y su distribución, buscando una presentación agradable para los usuarios.

Detalles y estadísticas

Este apartado busca mostrar al jugador información extra de alguno de los campeones sugeridos. Además de mostrar información como el rol, habilidades, runas e ítems recomendados, también se presentarán datos de partidas donde el campeón haya sido utilizado. Esto permitirá al jugador tener un mejor contexto sobre su uso en el competitivo de alto nivel y cómo podría adaptarse a su estilo de juego. En la Figura 5.5 se muestra la distribución de los elementos y los posibles datos a visualizar.

5.3. Implementación

El paso de implementación es un proceso complejo en donde se conjunta el proceso de análisis de datos junto con el sistema experto, en donde el primer paso es la obtención de todos los datos necesarios para realizar el proceso de análisis de los datos y aplicar las diferentes técnicas de minería de datos con el fin de obtener todo el conocimiento necesario para poder iniciar el proceso de desarrollo del sistema experto y el entorno

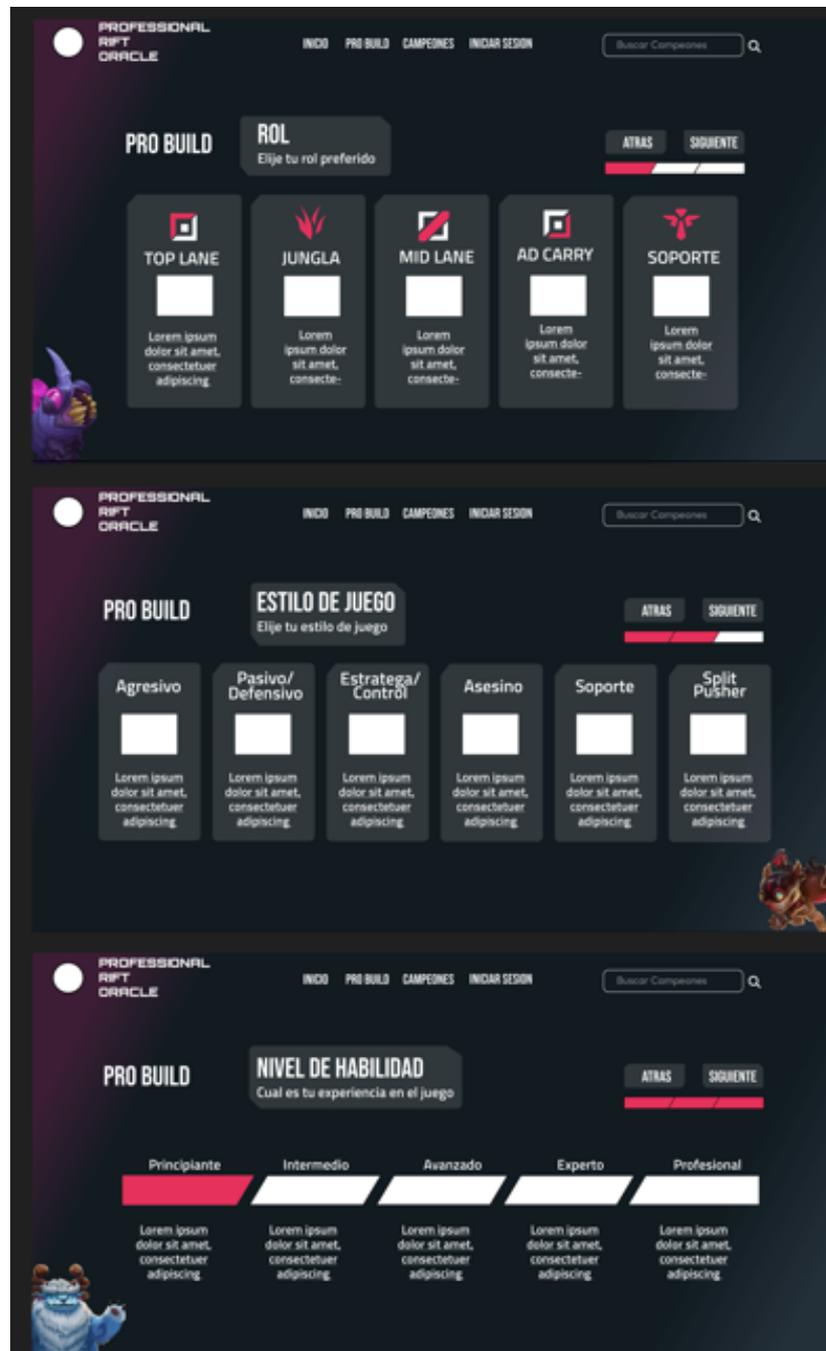


Figura 5.3: Diseño inicial del formulario.

5. APLICACIÓN DE METODOLOGÍAS DE DESARROLLO

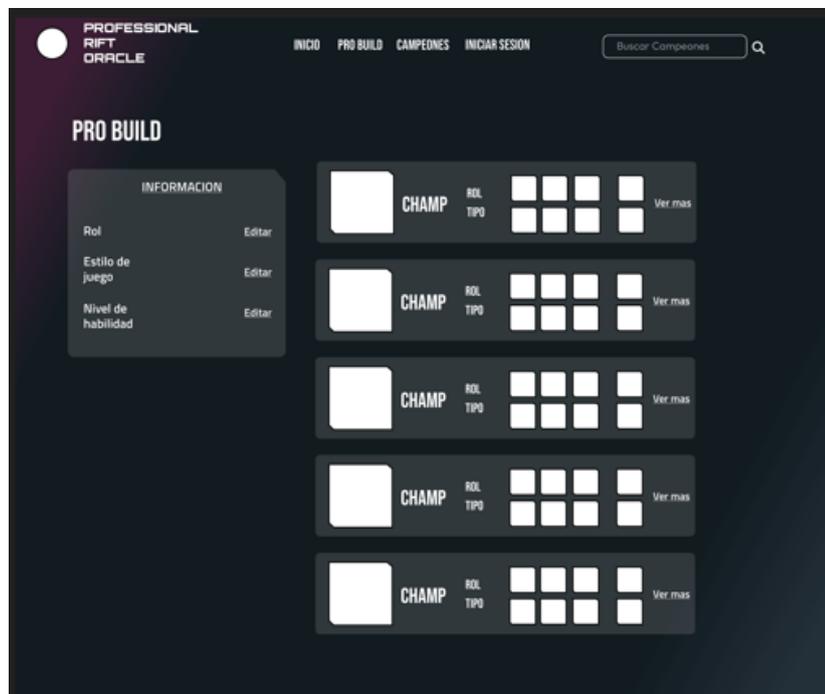


Figura 5.4: Diseño inicial de sugerencias.



Figura 5.5: Diseño inicial de detalles y estadísticas.

web donde se ejecutará todo el desarrollo.

Aquí es donde se observa la unión de la metodología KDD junto con el modelo en cascada, en donde se implementan algoritmos para el proceso de análisis de datos y desarrollo web para el desarrollo del sistema experto.

5.3.1. Selección

Los datos obtenidos fueron proporcionados por la empresa distribidora del juego, RIOT GAMES, a través de su API de desarrollo, en donde se pueden obtener datos de partidas, información de campeones como roles o estadísticas básicas.

Como variables importantes para el análisis de datos, se consideraron en un principio un total de 30 variables, en donde se toman en cuenta datos sobre campeones como rol o el tipo de campeón, y estadísticas de cada campeón antes de iniciar una partida, como daño, velocidad o resistencia. Por otra parte, también se toman datos de cada campeón al terminar una partida, en donde se obtienen datos sobre su rendimiento en partida, como el total de asesinatos, muertes y asistencias que obtuvo el campeón. También se toman en cuenta las runas e ítems seleccionados, y estadísticas de cada campeón pero al terminar una partida, lo que permite diferenciar estadísticas antes y después de cada partida. Todas las variables seleccionadas se pueden observar en la Tabla 5.1, en donde se muestra a detalle cada una de las variables utilizadas y una descripción sobre la variable. En total se obtuvieron 2940 datos, correspondientes a los diferentes torneos jugados en la LEC en el año 2024, teniendo un total de 28 variables.

5.3.2. Preprocesamiento y limpieza

Al obtener todos los datos requeridos, algunos campos contenían valores nulos o que eran muy dispares respecto a los otros datos, por ejemplo, valores en 0 o con valores muy elevados respecto a la media de todos los datos, lo que podría significar problemas en el análisis estadístico. Como solución al problema de los valores nulos, se procedió a realizar un método de imputación de valores faltantes, en donde cada variable con un valor nulo se convertía en "N/A", lo que indicaba que para ese dato en específico no aplicaba dicha variable. Por otro lado, para los valores atípicos se decidió eliminar los datos en concreto con ese tipo de deficiencias.

Estos dos procesos fueron realizados mediante un script de Python junto con la librería de Pandas, que permite añadir, editar o eliminar datos en concreto. El proceso de imputación inicia cuando se definen, mediante un diccionario, aquellas columnas con posibles valores nulos. En este caso, incluye "tag2" y los nombres de las columnas de ítems ("item1" a "item8"), tal como se muestra en el Código 5.1.

Enlace 5.1: Crear diccionario de posibles datos con valores nulos

```
columns_to_fill = ["tag2"] + [f"item{i}" for i in range(1, 9)]
```

Luego, en el fragmento de Código 5.2 se describe cómo utilizar "fillna('N/A')" para reemplazar los valores nulos en las columnas listadas en el diccionario con "N/A".

Enlace 5.2: Imputar el valor de N/A a valores nulos

```
df[columns_to_fill] = df[columns_to_fill].fillna('N/A')
```

De esta forma se pueden imputar datos mediante el uso de la librería Pandas y poder limpiar y preprocesar mejor los datos.

5.3.3. Transformación y reducción

El conjunto de datos obtenido estaba albergado en una sola tabla, lo que para futuras consultas o procesamientos podría significar problemas de rendimiento y visualización de los datos. Para ello, se realizó una reestructuración de los datos a fin de obtener un mejor acceso y manejo de los mismos.

El primer paso fue realizar una evaluación de la tabla que contenía todos los datos con el fin de identificar las variables y cómo se relacionaban unas con otras. Derivado de ese análisis, se creó una base de datos en donde se obtuvieron las tablas mostradas en la Figura 5.6. Se puede observar cómo se hicieron tablas específicas para apartados como los campeones, en donde se almacenan todos los campeones en una misma tabla y después se obtienen relaciones entre esa tabla y otras de las que dependa el campeón, como la tabla de estadísticas.

La nueva estructura de la base de datos permite obtener mejores resultados en el procesamiento de consultas y visualización de los datos.

5.3.4. Minería de datos

El proceso de minería de datos consistió en varios pasos hasta llegar a los resultados finales, desde la visualización de los datos hasta la creación de las reglas de asociación.

5.3.4.1. Visualización mediante Dremio y Power BI

Para poder obtener un mejor entendimiento de cómo se relacionan los datos y cuál puede ser su comportamiento ante diversos cambios en las variables, se implementó un servidor de Dremio para almacenar todos los datos obtenidos de las anteriores etapas de limpieza y transformación. En Dremio se creó un Data Lake, que permitía el procesamiento de los datos con un mejor rendimiento que otras herramientas similares. Posteriormente, se realizó la conexión con Power BI para modificar el comportamiento

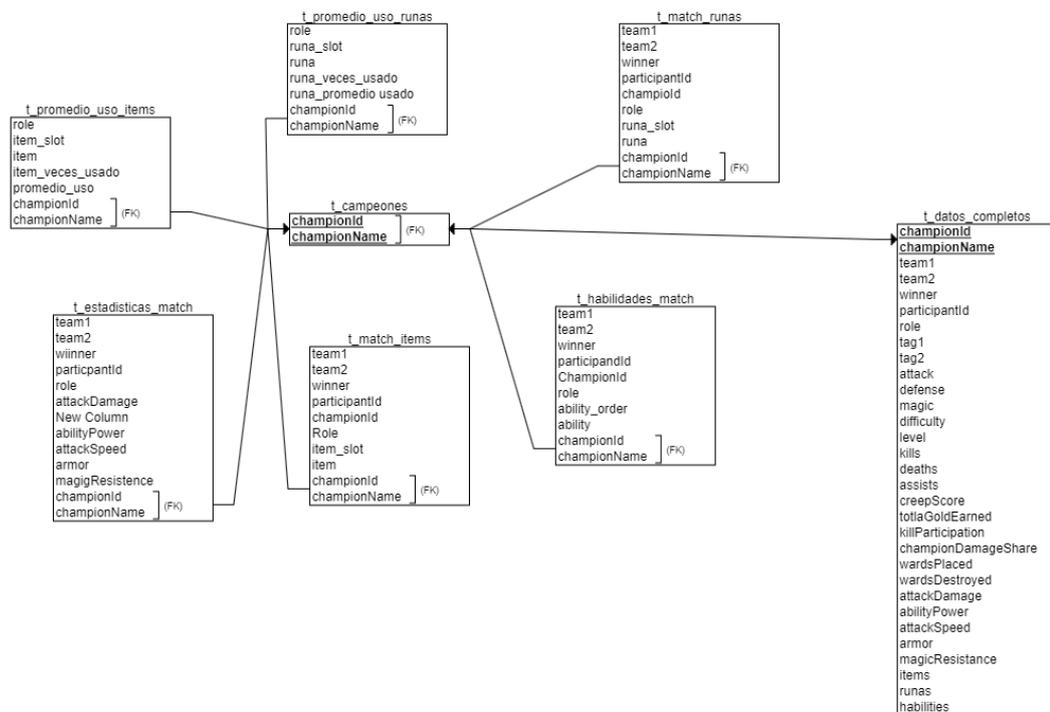


Figura 5.6: Nueva estructura del conjunto de datos.

5. APLICACIÓN DE METODOLOGÍAS DE DESARROLLO

de los datos y visualizarlos mediante elementos gráficos como tablas o gráficas.

En Power BI se realizaron diversas pruebas para medir el comportamiento de los datos, en donde se buscaba observar lo siguiente:

- Desempeño de los campeones según la dificultad y el rol.
- Desempeño de los campeones que ganaron una partida.
- Runas más usadas por campeón, según el rol.
- Ítems más usados por campeón, según el rol.

Con el objetivo de observar el comportamiento de las variables de campeón, ítems y runas frente a otras variables categóricas como el rol o la dificultad.

En el primer caso, se obtuvo como resultado la Figura 5.7. Se observa cómo, al cambiar las variables de rol y campeón, se muestra la cantidad de asesinatos y muertes del campeón según su dificultad y rol, además de estadísticas generales como su defensa o ataque.

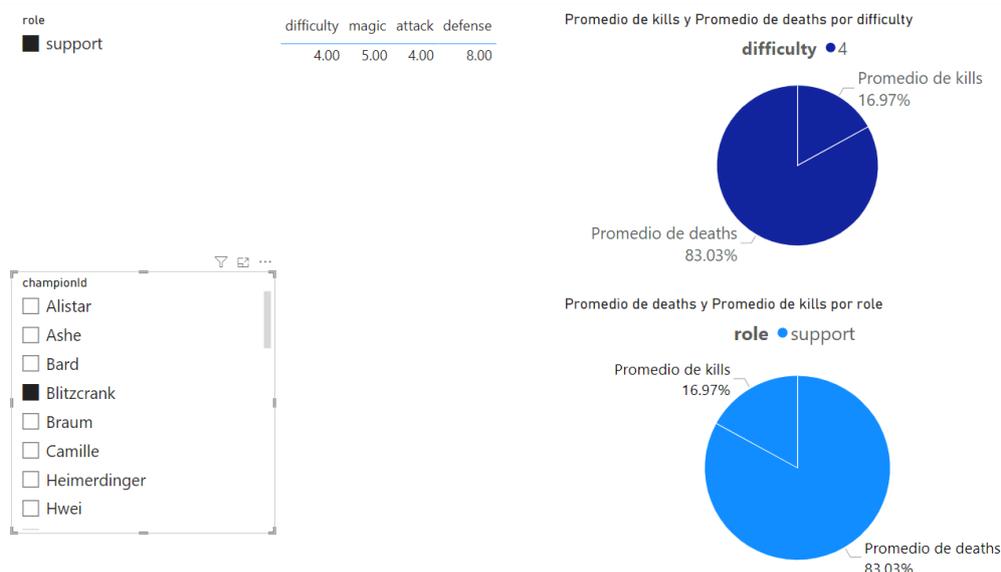


Figura 5.7: Desempeño de los campeones según la dificultad y rol.

En el segundo caso, se buscó identificar aquellos campeones que hubieran ganado algún partido y obtener el número de asesinatos y muertes de ese campeón en concreto, tal como se muestra en la Figura 5.8.

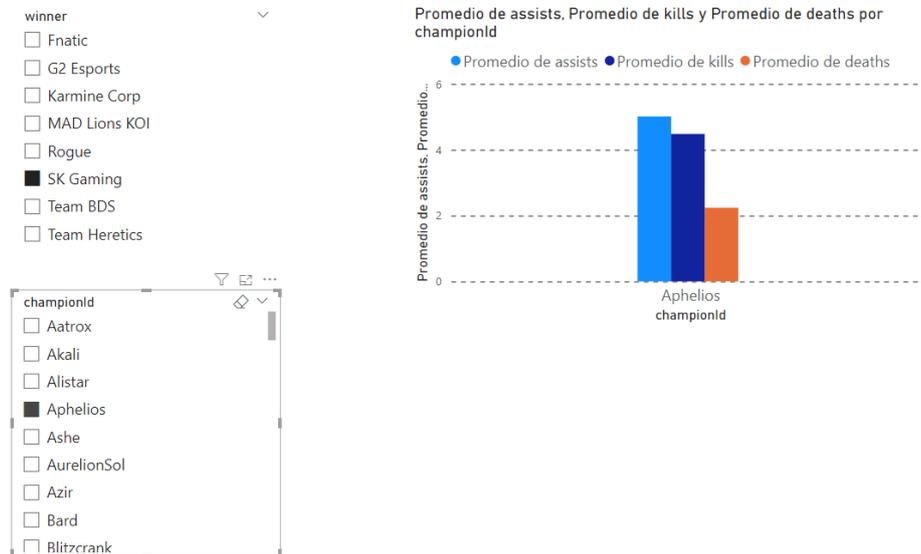


Figura 5.8: Desempeño de los campeones que ganaron una partida.

En el tercer caso del análisis, se buscó obtener una comparación de porcentajes de uso de las distintas runas, según el campeón y rol seleccionado, permitiendo generar una gráfica comparativa que describe qué runas son más usadas. En la Figura 5.9, se muestran los resultados obtenidos de una prueba realizada con un campeón en específico.

5. APLICACIÓN DE METODOLOGÍAS DE DESARROLLO

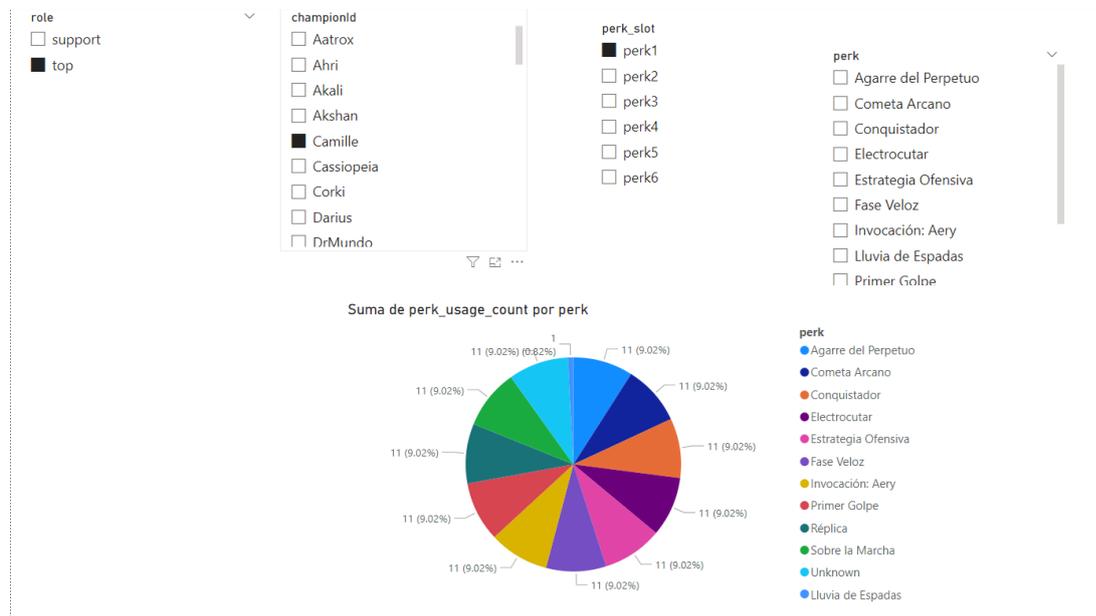


Figura 5.9: Runas más usadas por campeón, según el rol.

Por último, en el cuarto caso del análisis, se buscó obtener una comparativa de uso de los ítems más usados por campeón, según el rol seleccionado. El resultado se muestra en la Figura 5.10, donde se presenta una gráfica de los ítems más usados por un campeón en específico.

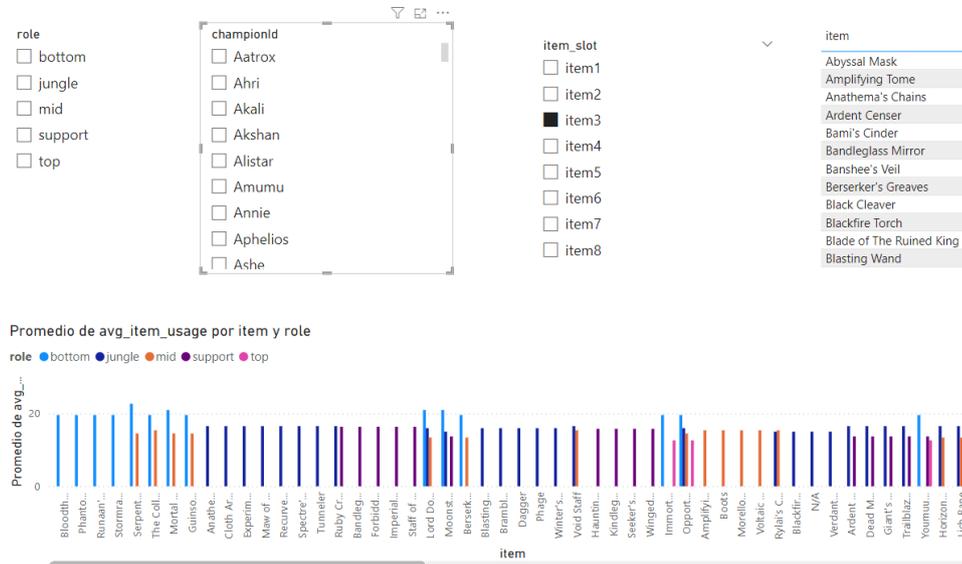


Figura 5.10: Ítems más usados por campeón, según el rol.

Todo este proceso de análisis permitió observar el comportamiento de las variables ante el cambio en diversos factores. Esto ayudó a determinar mejor las variables más importantes y cuáles son sus posibles relaciones con el resto. Con esta información se pudo definir mejor el proceso de creación de las reglas de asociación y cuál sería el impacto de las variables en su respectiva creación.

5.3.4.2. Reglas de asociación

Para la creación de reglas de asociación se optó por utilizar el algoritmo Apriori, debido a las características de los datos, lo que facilita más la aplicación del algoritmo y la creación de reglas de forma más eficiente que otros algoritmos de su tipo. Para ello se utilizó la biblioteca de Python 'apyori', que permite la ejecución del algoritmo de forma rápida y segura.

Se utilizó un script que emplea las bibliotecas 'pandas' para la manipulación de datos, 'apyori' para aplicar el algoritmo Apriori, 'sqlite3' para interactuar con la base de datos SQLite, y 'json' y 'os' para manejar archivos JSON y rutas de archivos.

Este script permite importar una base de datos y realizar una consulta para seleccionar diferentes campos como campeones, rol, dificultad, ítems y runas, así como estadísticas de rendimiento (kills, deaths, assists y participación en kills).

Una vez cargados los datos, se procede a categorizar a cada campeón en uno de tres

5. APLICACIÓN DE METODOLOGÍAS DE DESARROLLO

estilos de juego (agresivo, balanceado o defensivo) basándose en su participación en asesinatos, asistencias, asesinatos y muertes; permitiendo categorizar a los campeones con valores más bajos como defensivos, a los de valores muy altos como agresivos y los de estadísticas intermedias como balanceados.

Posteriormente, se clasifica a cada campeón en “experto”, “intermedio” o “principiante” según su nivel de dificultad. Después, se crea una lista que combina ítems y runas para cada campeón, según su porcentaje de uso en cada rol.

Se agrupan los datos por campeón, rol, dificultad, estilo de juego y nivel de experiencia, y luego se aplica el algoritmo Apriori a los ‘dataframes’ generados. Además, se configuran parámetros como el soporte mínimo, la confianza mínima, la elevación y la convicción. Para el caso de nuestros datos, se configuraron de la siguiente manera:

- Soporte mínimo = 0.1
- Confianza mínima = 0.85
- Elevación = 1.2
- Convicción = 2

Posteriormente, al conjunto de reglas obtenidas se les aplicó el algoritmo de ‘clustering’ K-means, con el objetivo de obtener las reglas más representativas y eliminar aquellas que sean redundantes o con valores atípicos. Esto se logró mediante un script de Python con la biblioteca ‘sklearn’, que provee herramientas para aplicar este algoritmo a grandes conjuntos de datos de forma más eficaz.

Finalmente, las reglas obtenidas a través del algoritmo de ‘clustering’ se almacenan en un archivo JSON, con la finalidad de facilitar su visualización e implementación en el sistema experto. En el fragmento de código 5.3 se muestra un ejemplo de una de las reglas obtenidas.

Enlace 5.3: Regla de asociación obtenida con Apriori

```
1 {
2   "model": "probuild.recommendation",
3   "pk": 32,
4   "fields": {
5     "champion": "Azir",
6     "role": "mid",
7     "difficulty": 9.0,
8     "playstyle": "agresivo",
```

```
9     "experience": "experto",
10    "item1": "Stealth Ward",
11    "item2": "Mercury's Treads",
12    "item3": "Dark Seal",
13    "item4": "Liandry's Torment",
14    "item5": "Nashor's Tooth",
15    "item6": "Void Staff",
16    "item7": "Needlessly Large Rod",
17    "item8": "Statikk Shiv",
18    "runa1": "Agarre del Perpetuo",
19    "runa2": "Demolicion",
20    "runa3": "Acondicionamiento",
21    "runa4": "Crecimiento Excesivo",
22    "runa5": "Concentracion Profunda",
23    "runa6": "Leyenda: Celeridad"
24  }
25 }
```

5.3.5. Interpretación y evaluación

En el proceso de minería de datos se generaron un total de 175 reglas de asociación que identifican características clave de cada campeón, como su rol, estilo de juego, dificultad y nivel de experiencia necesario. Estas reglas no solo formarán la base de conocimientos para el sistema experto, sino que también permitirán personalizar las recomendaciones a los jugadores. Al ingresar información sobre su rol, tipo de juego y nivel de experiencia, los jugadores recibirán al menos cinco opciones de campeones, junto con información detallada sobre los ítems y runas más adecuados. A futuro, se contempla extender el conjunto de datos e integrar nuevas métricas de rendimiento que ayuden a mejorar y obtener reglas más precisas y acertadas.

5.3.6. Implementación de la página web

El proceso de desarrollo de la página web es el siguiente paso del análisis de los datos, ya que en la página web es donde se podrán implementar todas las reglas obtenidas y, mediante una interfaz intuitiva y eficiente, el usuario podrá hacer uso del sistema experto.

El desarrollo de la página web consta de la implementación del frontend, backend y una base de datos, mediante diversas tecnologías de desarrollo como React para la creación de las interfaces y funcionalidades de usuario, y Django para la implementación del sistema experto, integración con la base de datos y funciones del lado del servidor. De tal forma que se puedan integrar todas las funcionalidades requeridas para el sistema.

5.3.6.1. Implementación de la base de datos

Para el almacenamiento y gestión de la información, se implementó una base de datos con el gestor PostgreSQL, que permitirá un acceso fiable así como un manejo eficiente de los datos.

Configuración del servidor

Es necesario configurar el servidor con especificaciones sobre las direcciones y puertos que usará la base de datos para comunicarse con el sistema, además de opciones de rendimiento que permitan un mejor manejo de la información con un buen desempeño. Para el caso de las configuraciones de conexión, se accede al archivo `pg_hba.conf`, el cual contiene algunas configuraciones básicas del servidor, y será necesario modificar algunas de estas opciones de acuerdo con los requerimientos del sistema. El Código 5.4 muestra un ejemplo de la configuración de la conexión de la base de datos.

Enlace 5.4: Configuración de la conexión del servidor de la base de datos

```
# Acceso local
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Permitir acceso desde la red
host all all 192.168.1.0/24 md5
```

Posteriormente, desde el archivo `postgresql.conf` se podrán realizar modificaciones a las especificaciones de rendimiento del servidor. Algunas de las más importantes son configuraciones de memoria, como el tamaño de memoria caché o el número de procesos simultáneos que puede ejecutar el servidor. En el Código 5.5 se observa un ejemplo de algunas de las configuraciones de rendimiento realizadas.

Enlace 5.5: Configuración de rendimiento de la base de datos

```
# Memoria compartida para PostgreSQL
shared_buffers = '1GB'
# Memoria para operaciones de ordenamiento y hash
work_mem = '64MB'
# Memoria para operaciones de mantenimiento
maintenance_work_mem = '512MB'
# Memoria cache del sistema operativo
effective_cache_size = '2GB'
```

Creación de la base de datos

Es un proceso sencillo en el que solo se creará la base de datos en donde se almacenarán todos los datos del sistema. Mediante el Código 5.6 se creará la base de datos que será utilizada para almacenar las reglas de asociación e información como datos de usuario e inicio de sesión. No es necesaria la creación de tablas directamente desde el gestor de base de datos, ya que Django ofrece una herramienta de migración y gestión de bases de datos mediante la configuración de módulos, lo cual permite crear, modificar o eliminar información de forma más sencilla sin manipular directamente la base de datos.

Enlace 5.6: Creación de la base de datos

```
CREATE DATABASE probuild;
```

De esta forma, se configuran tanto las especificaciones del servidor como la creación de una base de datos que posteriormente será modificada mediante las herramientas de desarrollo de Django.

5.3.6.2. Implementación del Backend

En este proceso se desarrollaron los modelos y métodos que permitieron la implementación de las funcionalidades principales del sistema, así como la conexión con la base de datos para poder crear las APIs con el fin de obtener y enviar información.

Conexión con la base de datos

Para conectar PostgreSQL con Django es necesario instalar un controlador que permita la conexión entre PostgreSQL y Python. El controlador más popular es `psycopg2`, y la forma de instalación es mediante un gestor de paquetes como `pip`. En el Código 5.7 se muestra cómo instalar el controlador.

Enlace 5.7: Instalar controlador psycopg2

```
$> pip install psycopg2
```

Una vez instalado, se procede a configurar el archivo `settings.py` con la información de nuestro servidor de base de datos, como el nombre de la base de datos, usuario, contraseña, host y puerto que utiliza el servidor. Una vez configurada esta información, se realizan las migraciones iniciales con el fin de crear todas las tablas necesarias que Django requiere para su funcionamiento, además de funcionar como prueba para comprobar si la conexión entre la base de datos y Django es exitosa. Estas migraciones se ejecutan con el Código 5.8 en la línea de comandos.

Enlace 5.8: Migraciones iniciales de Django a PostgreSQL

```
$> python manage.py migrate
```

Una vez hechas las migraciones, en la base de datos se podrán observar nuevas tablas relacionadas con la seguridad del sistema, administración de usuarios e información.

Método para cargar reglas de asociación

Para cargar las reglas de asociación se creó un modelo que especifica cuáles son los campos que contienen nuestras reglas de asociación. La Tabla 5.2 describe los campos específicos que se utilizan en la creación del modelo, además de su respectiva configuración en la base de datos.

Posteriormente, se creó un método que permite cargar las reglas de asociación desde un archivo JSON y enviar esa información a la base de datos. La forma de ejecutar ese método es mediante el Código 5.9, que se coloca en la línea de comandos. Una vez ejecutado, se cargarán todas las reglas en la base de datos.

Enlace 5.9: Comando para cargar reglas de asociación a la base de datos

```
$> python manage.py load_fixtures
```

De esta forma se cargarán las reglas de asociación creadas en el proceso de minería de datos y se podrá acceder a ellas mediante la base de datos.

Vista para el motor de inferencias

Este método permitirá obtener la información que el usuario envía desde el formulario, y mostrará las reglas que se adapten mejor a sus necesidades según las reglas de asociación.

Primero se configuraron los parámetros que debería tener la solicitud para obtener las sugerencias, los cuales son:

- rol
- estilo
- experiencia

Posteriormente se configura el cuerpo de la respuesta, en donde se especifica el nombre del campeón, runas e ítems, según las reglas de asociación.

Una vez creada la vista, se configura la ruta de la API desde donde se podrá obtener la información. En el archivo `urls.py` se coloca la ruta y la vista asociada, con la cual se podrá acceder a las sugerencias. En el Código 5.10 se muestra la ruta usada para acceder a la información de las reglas de asociación.

Enlace 5.10: Ruta para acceder a las sugerencias

```
path('expert-system/', views.get_recommendations, name='get_recommendations'),
```

5.3.6.3. Implementación del Frontend

Para la implementación del frontend se hizo uso del framework React, el cual permite el desarrollo mediante componentes o módulos, lo que posibilita un desarrollo más ágil y mejora el rendimiento de la página web. En total se crearon cinco componentes destinados a cada elemento de la página web:

- **HomePage:** Página inicial del sitio web, en donde se muestra información general de las funcionalidades del sistema.
- **Formulario:** Tres componentes destinados a ser el formulario que recabará las características de juego del jugador, en donde se pregunta el rol, estilo de juego y experiencia del jugador.
- **Sugerencias:** Módulo en donde se mostrarán las cinco sugerencias más adecuadas a las respuestas obtenidas del formulario, mostrando la información del campeón y las respuestas dadas en el formulario.
- **Detalles del campeón:** Mostrará información a detalle de un campeón en específico, en donde se podrá observar información básica como las habilidades, rol o tipo de campeón, además de mostrar las runas e ítems más adecuados para el campeón.
- **Integración de información:** Componente destinado a recabar y almacenar en caché las respuestas que el usuario proporciona en el formulario, y posteriormente poder usar esa información para que el sistema experto gestione y muestre las sugerencias.

Implementación de APIs

El sistema hace uso de dos APIs que permiten obtener toda la información acerca del juego y sobre las recomendaciones que ofrece el sistema:

- **API del sistema experto:** Permite obtener las sugerencias basadas en las características del jugador. La información que se puede obtener incluye el nombre del campeón, rol, ítems y runas.
- **Data Dragon API:** Ofrece toda la información correspondiente al juego, como información del campeón, imágenes de ítems, runas o campeones y estadísticas.

5.4. Verificación

En esta etapa se realizan pruebas al sistema con el fin de validar su funcionamiento en cada uno de los componentes y su integración con el backend y la base de datos. En caso de no cumplir con lo requerido, se regresará a la etapa de implementación para corregir o mejorar la funcionalidad en específico.

5.4.1. Verificación de la API

Esta verificación permitirá obtener información sobre el comportamiento del sistema al momento de ingresar o solicitar información. Para realizar pruebas se utilizó el software Postman, una herramienta que permite realizar pruebas a APIs mediante solicitudes de tipo request, con el fin de obtener información sobre el funcionamiento o estado de rendimiento de la API.

En este caso se realizaron pruebas a la API que permite obtener las sugerencias. La Figura 5.11 muestra una prueba de esta API, en donde se realiza una solicitud de tipo GET a la ruta de la API; a esta ruta se le agregan los parámetros *rol*, *playstyle* y *experience* con el fin de obtener las sugerencias más adecuadas según esos parámetros. Como resultado, en el cuerpo de la respuesta se obtiene un JSON con las cinco sugerencias más adecuadas a la configuración de los parámetros, en donde cada sugerencia muestra el nombre del campeón, runas e ítems que pueden ser usados.

La verificación del estado de las APIs permite determinar que el funcionamiento es el adecuado según las necesidades del sistema.

5.4.2. Verificación de la integración con el frontend

Se realizaron pruebas que permitieron obtener información sobre el comportamiento de los componentes y la integración de la API, y buscar posibles mejoras, tanto en la

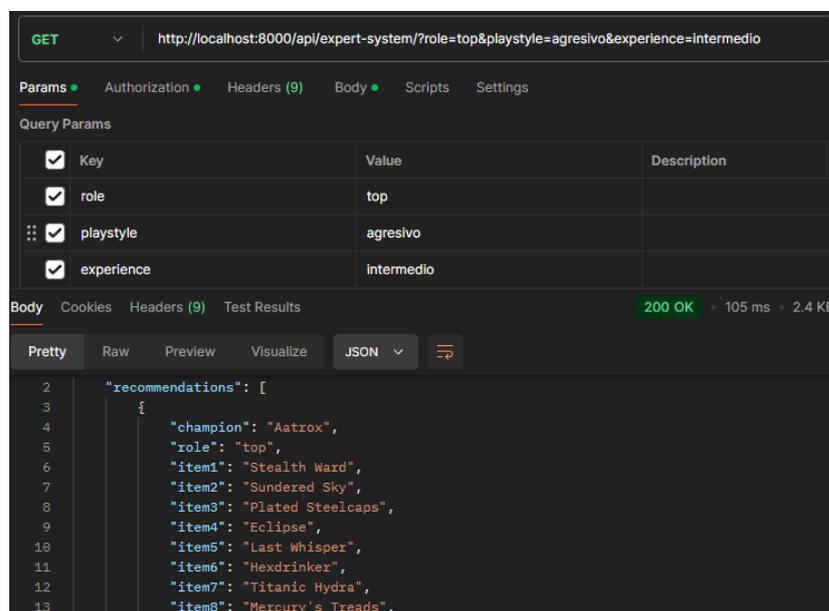


Figura 5.11: Prueba con Postman de la API de sugerencias.

experiencia de usuario como en la parte funcional del sistema.

Entre las principales pruebas realizadas están:

- Funcionamiento del formulario.
- Mostrar sugerencias.
- Mostrar detalles de sugerencias.

A continuación, se realiza una pequeña descripción del proceso de verificación y los resultados obtenidos.

Funcionamiento del formulario

Para la sección del formulario, fue importante identificar si la información recopilada de las respuestas a las preguntas se enviaba correctamente al servidor y si esta solicitud resultaba exitosa, cuál era el mensaje que enviaría el servidor; en este caso, cinco recomendaciones diferentes basadas en las reglas de asociación cargadas en la base de datos.

Para ello se utilizaron las herramientas de desarrollador que ofrece cualquier navegador web por defecto. Esto permitió identificar posibles errores en consola, el comportamiento del sistema al contestar las preguntas y la información sobre las solicitudes a la API. La Figura 5.12 muestra la interfaz de una de las preguntas del cuestionario para identificar en este caso el rol del jugador; cuando el usuario selecciona una opción

5. APLICACIÓN DE METODOLOGÍAS DE DESARROLLO

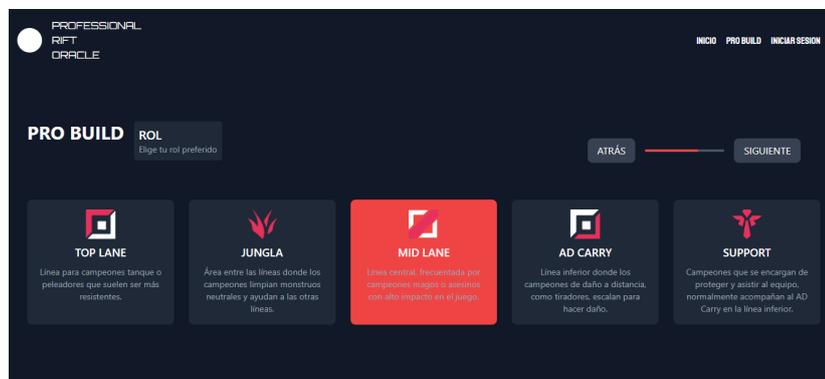


Figura 5.12: Primera pregunta del formulario para identificar características del jugador.

y presiona el botón de siguiente, el sistema almacenará su respuesta y posteriormente será usada para realizar la solicitud de las sugerencias.

El Código 5.11 muestra los console logs que permiten mostrar cuáles son los datos que recaba el sistema para poder enviar la solicitud a la API.

Enlace 5.11: Console logs de la información recibida del formulario

```
console.log("Role:", selectedRole);  
console.log("Playstyle:", selectedPlaystyle);  
console.log("Skill_Level:", selectedSkillLevel);
```

Posteriormente, con la opción Red de las herramientas de desarrollador, se pudo observar cuáles fueron las diferentes solicitudes de entrada y salida una vez completadas y enviadas las respuestas del formulario. La Figura 5.13 describe cuál fue la solicitud y parámetros enviados al servidor, además de cuál fue la respuesta del servidor a esa solicitud; en este caso, un JSON que contiene toda la información referente a los campeones y objetos obtenidos a partir de las reglas de asociación.

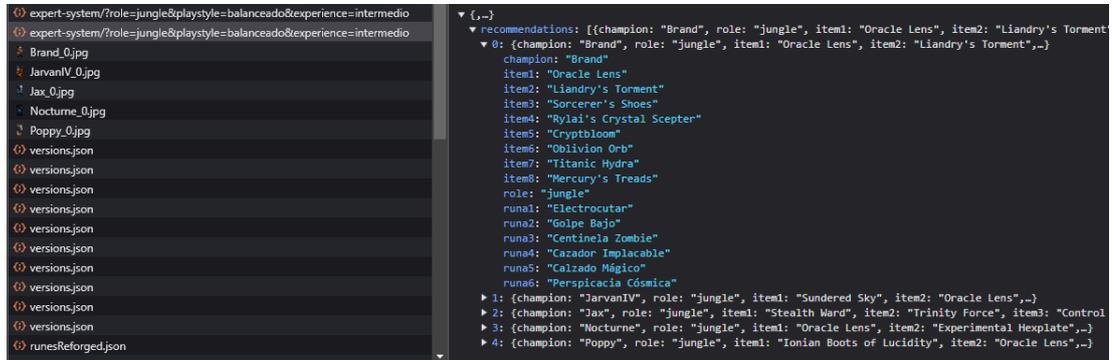


Figura 5.13: Solicitud y respuesta del servidor usando como parámetros la información del formulario.

Como resultado de la información obtenida, la Figura 5.14 muestra las cinco sugerencias obtenidas de la solicitud al servidor, donde se puede observar una imagen alusiva al personaje, el nombre, el rol y las runas que podrían funcionar mejor con ese campeón.

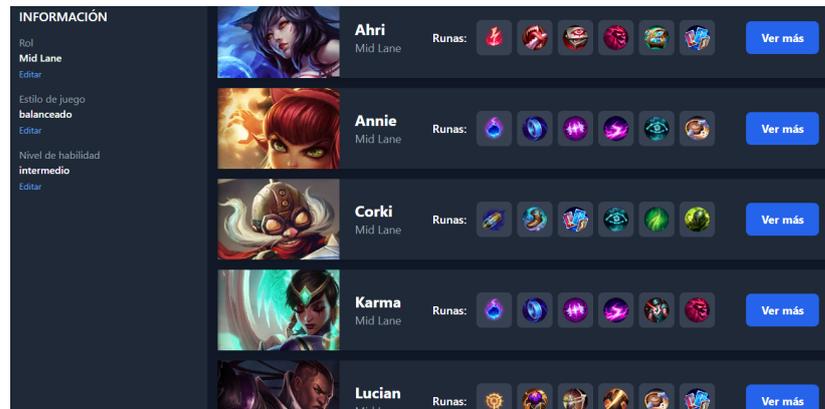


Figura 5.14: Sugerencias obtenidas con base en las características del jugador.

Por último, si se desea ver más detalles acerca del campeón, el botón de ver más permite abrir otra sección para conocer mejor las características del personaje. La Figura 5.15 describe qué información adicional se ofrece, como las habilidades, tipo de campeón y las runas e ítems sugeridos para esa sugerencia.

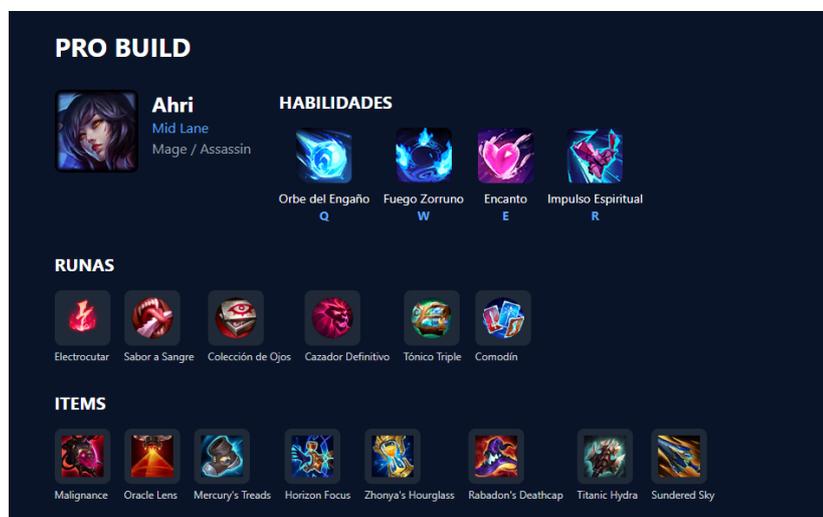


Figura 5.15: Detalles de la sugerencia.

Cada una de estas pruebas y métodos para verificar la funcionalidad del sistema permitieron observar el comportamiento del sistema experto, a fin de encontrar posibles errores y oportunidades de mejora, obteniendo como resultado el correcto funcionamiento de las funcionalidades implementadas.

5.4.3. Pruebas de funcionamiento de las recomendaciones

Es importante observar el comportamiento del sistema en diversas situaciones. En este caso, se realizaron distintas pruebas sobre las recomendaciones obtenidas, con el fin de demostrar que las sugerencias se adaptan a las características de los jugadores. Para ello, se realizaron cinco pruebas que consisten en responder diferentes formularios con características variadas del jugador, buscando obtener resultados diversos en las sugerencias. Esto permite evidenciar que las recomendaciones se ajustan a distintos parámetros de configuración o perfiles de jugador.

Prueba uno

Para la prueba uno, las respuestas del formulario de características del jugador son las siguientes:

- Rol: Top lane
- Estilo de juego: Balanceado
- Nivel de experiencia: Principiante

Derivada de esta configuración, la Figura 5.16 muestra que se obtuvieron resultados variados en la recomendación de personajes, ítems y runas, siendo estas las opciones que el sistema considera más adecuadas para las características del jugador mencionadas.

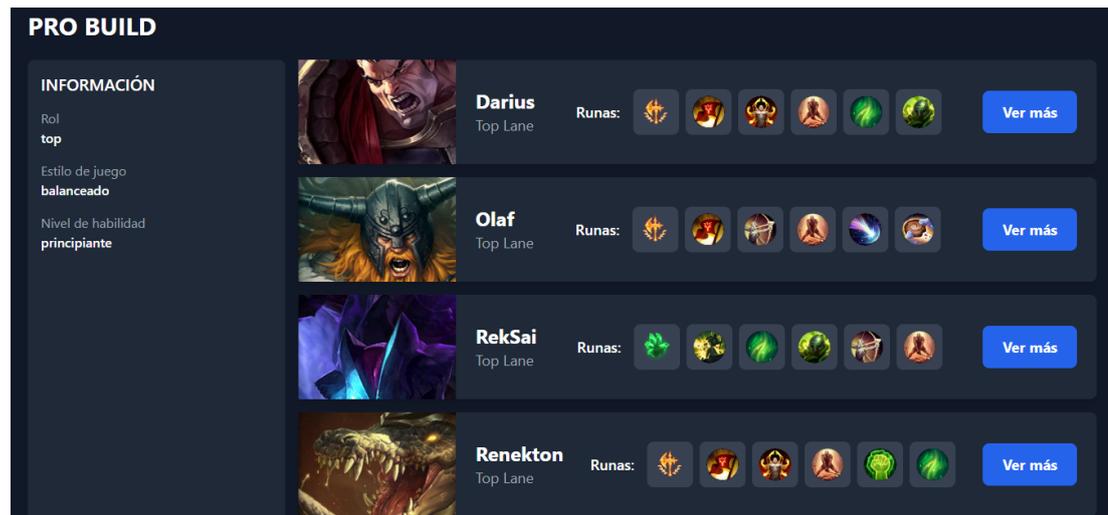


Figura 5.16: Resultados de la prueba de recomendaciones uno.

Prueba dos

Para la prueba dos, las respuestas del formulario variaron en el estilo de juego y nivel de experiencia, buscando demostrar que las recomendaciones se adaptan a diferentes estilos y niveles para un mismo rol. La configuración de parámetros del sistema experto utilizada fue la siguiente:

- Rol: Top lane
- Estilo de juego: Agresivo
- Nivel de experiencia: Experto

La Figura 5.17 comprueba que se obtuvieron sugerencias diferentes a las de la prueba uno, lo que demuestra que el sistema adapta las recomendaciones a las características específicas del jugador, sin importar el nivel de experiencia o estilo de juego.

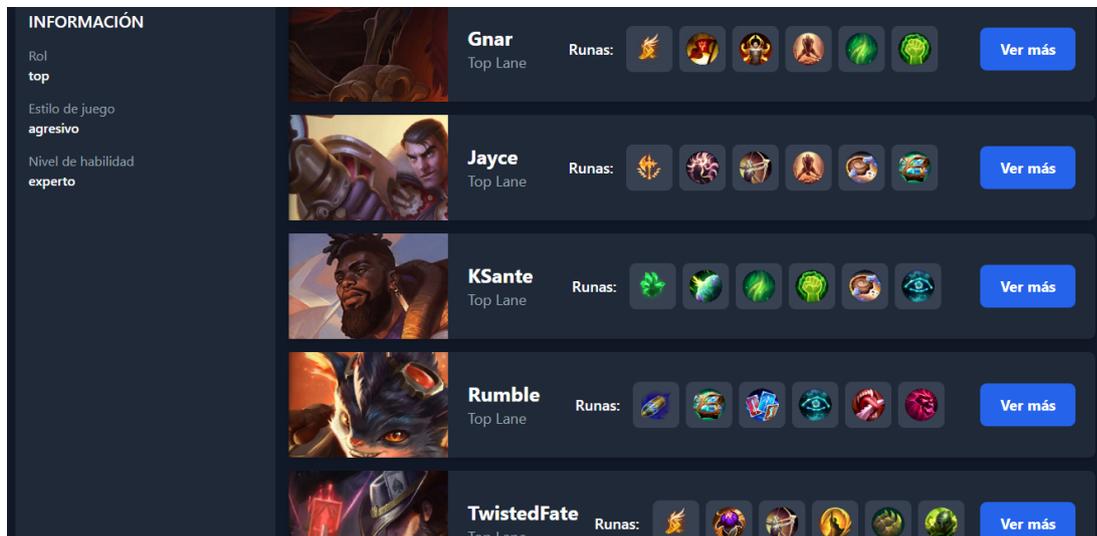


Figura 5.17: Resultados de la prueba de recomendaciones dos.

Prueba tres

En esta prueba se varió el rol preferido, buscando obtener personajes diferentes o runas distintas a las obtenidas en pruebas anteriores.

- Rol: Support
- Estilo de juego: Balanceado
- Nivel de experiencia: Intermedio

Esta configuración pone a prueba el sistema para el rol de support. La Figura 5.18 muestra las recomendaciones obtenidas para esta configuración.

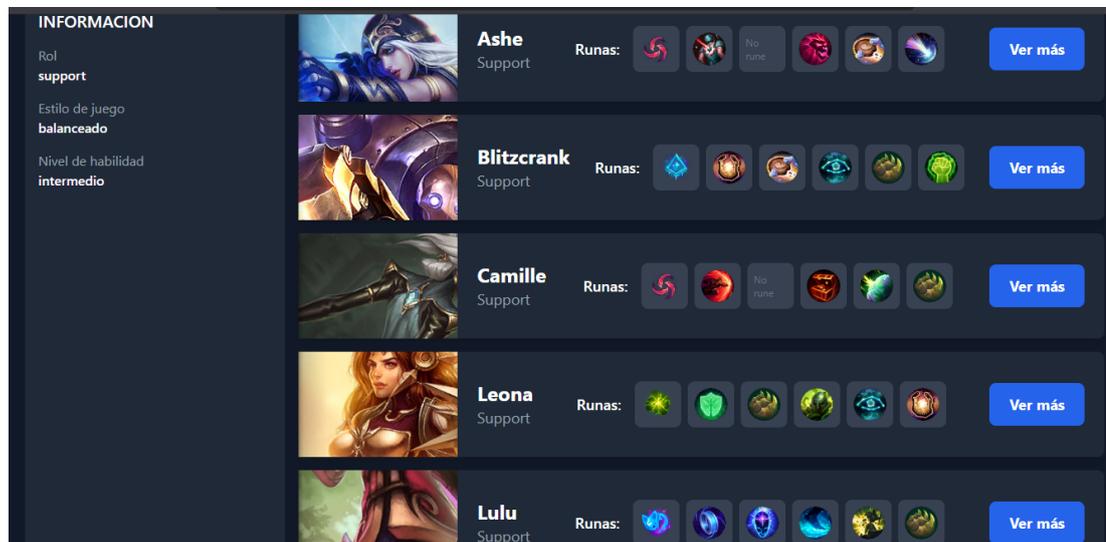


Figura 5.18: Resultados de la prueba de recomendaciones tres.

Prueba cuatro

En la prueba cuatro, se mantiene el mismo rol de support, pero con variación en el nivel de experiencia. Las respuestas para el formulario fueron las siguientes:

- Rol: Support
- Estilo de juego: Balanceado
- Nivel de experiencia: Principiante

Para este caso, el nivel de experiencia es el de un jugador principiante, por lo que se buscan opciones fáciles de jugar. La Figura 5.19 muestra las opciones más adecuadas para un jugador principiante en el rol de support.

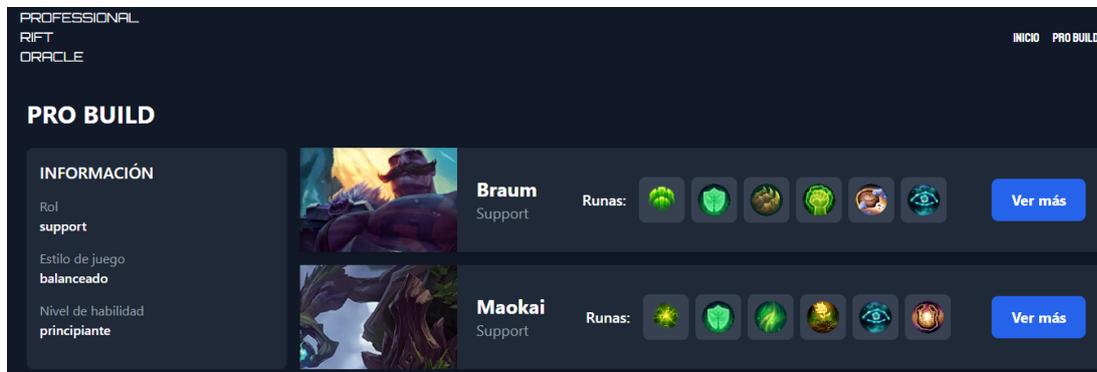


Figura 5.19: Resultados de la prueba de recomendaciones cuatro.

Prueba cinco

Finalmente, en la última prueba, se busca una configuración totalmente diferente, con características de un jugador experimentado en una de las líneas más complejas del juego.

- Rol: Mid lane
- Estilo de juego: Defensivo
- Nivel de experiencia: Experto

A partir de esta prueba, la Figura 5.20 muestra los resultados que obtendría un jugador altamente experimentado al usar el sistema de recomendación, con runas e ítems que se adaptan a sus necesidades.

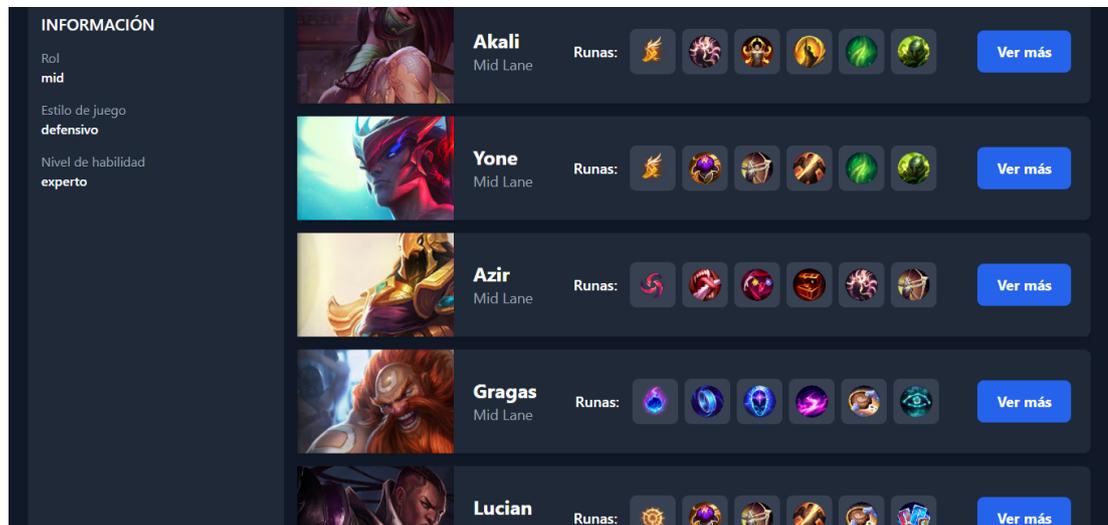


Figura 5.20: Resultados de la prueba de recomendaciones cinco.

Con las pruebas realizadas, se demuestra que el sistema puede adaptarse perfectamente a las necesidades de los jugadores, dependiendo de sus características dentro del juego, ofreciendo diversas opciones adecuadas para cada perfil en una partida de *League of Legends*.

5. APLICACIÓN DE METODOLOGÍAS DE DESARROLLO

| Variable | Descripción |
|--------------------------|-------------------------------------------------|
| team1 | Nombre del equipo 1 |
| team2 | Nombre del equipo 2 |
| winner | Equipo ganador |
| championId | ID del campeón seleccionado |
| role | Rol del campeón (e.g., jungla, soporte) |
| tag1 | Primera etiqueta de características del campeón |
| tag2 | Segunda etiqueta de características del campeón |
| attack | Valor de ataque del campeón |
| defense | Valor de defensa del campeón |
| magic | Poder mágico del campeón |
| difficulty | Nivel de dificultad del campeón |
| item1, item2, ..., item8 | IDs de los ítems equipados |
| perk1, perk2, ..., perk6 | Runas seleccionadas por el jugador |
| participantId | ID del participante en la partida |
| level | Nivel alcanzado por el campeón |
| kills | Número de asesinatos del jugador |
| deaths | Número de muertes del jugador |
| assists | Número de asistencias del jugador |
| creepScore | Puntuación de súbditos derrotados |
| totalGoldEarned | Total de oro ganado en la partida |
| killParticipation | Participación en asesinatos del equipo |
| championDamageShare | Porcentaje de daño total hecho por el campeón |
| wardsPlaced | Número de centinelas colocadas |
| wardsDestroyed | Número de centinelas destruidas |
| attackDamage | Daño de ataque del campeón |
| abilityPower | Poder de habilidad del campeón |
| attackSpeed | Velocidad de ataque del campeón |
| armor | Armadura del campeón |
| magicResistance | Resistencia mágica del campeón |
| abilities | Habilidades seleccionadas durante la partida |

Tabla 5.1: Descripción de las variables del conjunto de datos.

| Campo | Tipo de dato | Configuraciones |
|--------------|---------------------|-------------------------------|
| champion | CharField | max_length=100, db_index=True |
| role | CharField | max_length=50, db_index=True |
| difficulty | FloatField | - |
| playstyle | CharField | max_length=50 |
| experience | CharField | max_length=50 |
| item1 | CharField | max_length=255 |
| item2 | CharField | max_length=255 |
| item3 | CharField | max_length=255 |
| item4 | CharField | max_length=255 |
| item5 | CharField | max_length=255 |
| item6 | CharField | max_length=255 |
| item7 | CharField | max_length=255 |
| item8 | CharField | max_length=255 |
| runa1 | CharField | max_length=255 |
| runa2 | CharField | max_length=255 |
| runa3 | CharField | max_length=255 |
| runa4 | CharField | max_length=255 |
| runa5 | CharField | max_length=255 |
| runa6 | CharField | max_length=255 |

Tabla 5.2: Tabla de campos del modelo de campeones.

Resultados y validación

Tras terminar con el proceso de desarrollo, el proceso de validación es clave al implementar diversos procedimientos que permitan identificar de forma clara si las recomendaciones son viables u obtener detalles sobre la percepción del sistema y las recomendaciones ante un público reducido, que esté relacionado con el área de aplicación que aborda el desarrollo, permitiendo obtener datos que ayuden a identificar si un sistema es apto para su despliegue final.

6.1. Validación SUS (System Usability Scale)

La aplicación de una metodología de validación es parte fundamental para la evolución de un sistema, ya que permite integrar a los usuarios en un proceso de pruebas y validación, que ayuda a los desarrolladores a identificar los aspectos clave en los que un sistema puede mejorar. La validación SUS (System Usability Scale) es parte clave del proceso de validación de este proyecto, al ser aplicada en una muestra que permitió determinar cuáles son las oportunidades de mejora y qué aspectos son aplicados de forma efectiva.

6.1.1. Descripción de la metodología

La escala SUS (System Usability Scale) es un método que permite aplicar una evaluación de la usabilidad de un sistema, lo que permite medir la facilidad de uso de un sistema o servicio (96).

Desarrollada por John Brooke en 1986, cuando trabajaba para una empresa de desarrollo y manufactura de microcomputadoras. Esta metodología permite evaluar tres puntos importantes que destacan en la usabilidad de un sistema.

- Eficacia: Indica en qué grado los usuarios pueden alcanzar con éxito sus objetivos.

6. RESULTADOS Y VALIDACIÓN

- Eficiencia: Mide cuál es el esfuerzo necesario que se necesita para cumplir con esos objetivos.
- Satisfacción: Conocer si hacer uso del sistema fue satisfactorio.

A partir de estos puntos, este método de validación funciona a partir de la aplicación de un cuestionario con enunciados predefinidos, lo que permite calcular el resultado final de forma más sencilla (13).

6.1.1.1. Cuestionario

La escala SUS consta de diez enunciados predefinidos que pueden ser adaptados a cualquier contexto, sistema, producto o servicio. Se aplica a usuarios que tengan algún tipo de experiencia de uso con el sistema evaluado. Cada enunciado se puntúa con la escala Likert.

- Totalmente en desacuerdo: 1 punto
- En desacuerdo: 2 puntos
- Neutral / Ni de acuerdo ni en desacuerdo: 3 puntos
- De acuerdo: 4 puntos
- Totalmente de acuerdo: 5 puntos

Para el sistema de recomendación de campeones en League of Legends, los enunciados fueron adaptados de acuerdo con el contexto del área de aplicación, lo que permitió identificar de mejor manera los aspectos más destacados del sistema.

1. Creo que me gustaría usar este sistema de recomendación de personajes con frecuencia.
2. Encontré el sistema innecesariamente complejo para elegir un personaje adecuado.
3. Pensé que el sistema hacía fácil encontrar un personaje para jugar.
4. Creo que necesitaría el apoyo de otra persona para entender cómo usar este sistema de recomendación.
5. Encontré que las diferentes funciones del sistema estaban bien integradas.
6. Pensé que había demasiada inconsistencia en las recomendaciones de personajes.
7. Imagino que la mayoría de los jugadores aprenderían a usar este sistema de recomendación rápidamente.
8. Encontré el sistema complicado o poco intuitivo para elegir un personaje.

9. Me sentí seguro al usar el sistema para encontrar un personaje adecuado para mí.
10. Necesité aprender muchas cosas antes de poder aprovechar bien este sistema de recomendación.

Una vez aplicado el cuestionario de validación SUS, es necesario aplicar una serie de operaciones matemáticas que permitan identificar el SUS total del sistema.

6.1.1.2. Obtención de resultados

Para obtener la métrica SUS, es necesario realizar algunos cálculos sencillos por cada participante para, al final, obtener el SUS total del sistema. Este proceso consta de:

- Sumar todas las respuestas de los enunciados que son impares (1, 3, 5, 7, 9) y restarles 5.
- Sumar todas las respuestas de los enunciados que son pares (2, 4, 6, 8, 10) y restarles 25.
- Sumar los dos resultados y multiplicar por 2.5.

Una vez obtenido el SUS de cada participante, es necesario obtener un promedio de todos los resultados y así obtener el SUS total del sistema. Ese resultado es la puntuación total del sistema en un rango de 0 a 100, donde una puntuación promedio mayor a 68 % indica que es un sistema aceptable con una buena usabilidad percibida por parte de los participantes. La Gráfica 6.1 muestra cuáles son las puntuaciones y rangos en los que puede ser catalogado un sistema según su SUS.

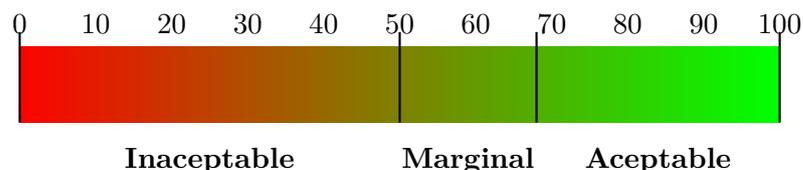


Figura 6.1: Escala de usabilidad SUS

A raíz de esto, se aplicó la prueba de validación con una muestra poblacional relacionada con los videojuegos y, en algunos casos, con League of Legends, lo que permitió identificar la perspectiva de usabilidad en usuarios con experiencia en el juego y potenciales jugadores casuales.

6.1.2. Descripción de la muestra

Esta prueba de validación se aplicó a una muestra poblacional de 30 personas, en la que se recabó información sobre cómo es su relación con los videojuegos, en específico si conocían y habían jugado League of Legends. A partir de eso, se pudo realizar un análisis sobre el tipo de usuario al que se le aplicó esta prueba.

Derivado de este análisis, se encontró que la muestra poblacional a la que se le aplicó la prueba de validación SUS para el sistema de recomendación estratégica de campeones de League of Legends se encuentra en un rango de edad de 18 a 25 años, siendo un total de 23 personas del género masculino y 7 del género femenino.

6.1.2.1. Relación con los videojuegos

Todos los participantes tienen o han tenido algún tipo de experiencia o acercamiento con los videojuegos. La Gráfica 6.2 muestra que un 53.3% de los participantes juega videojuegos con cierta frecuencia, mientras que el resto juega ocasionalmente o ha tenido algún acercamiento en algún momento.

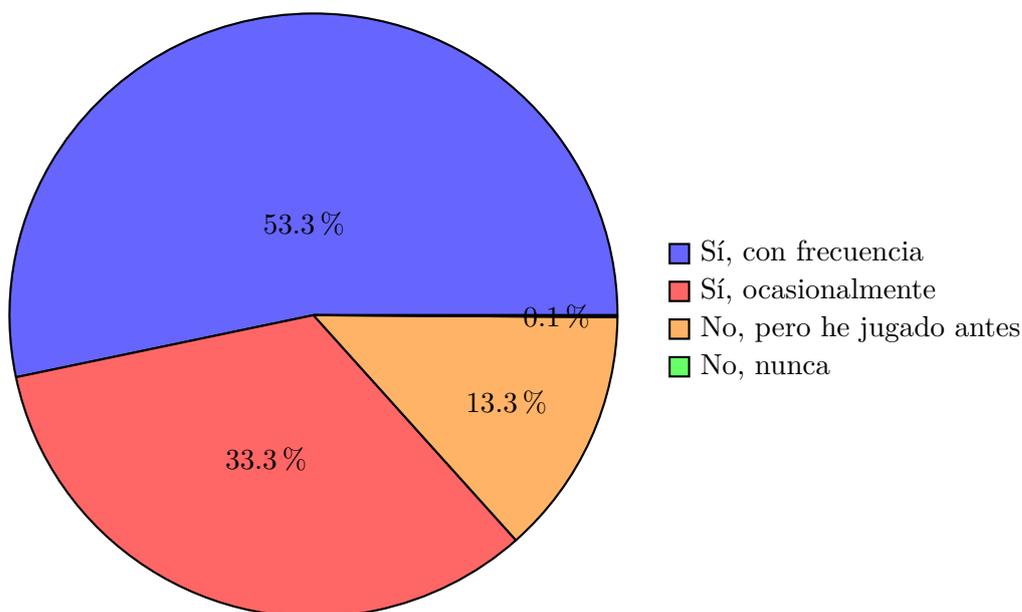


Figura 6.2: Resultados de la prueba: ¿Juegas videojuegos?

En un análisis más detallado, la Gráfica 6.3 describe la frecuencia con la que suelen jugar videojuegos, siendo una gran mayoría —por arriba del 50%— la que juega todos los días o al menos una vez a la semana, lo que implica que los participantes tienen algún tipo de conocimiento avanzado sobre videojuegos.

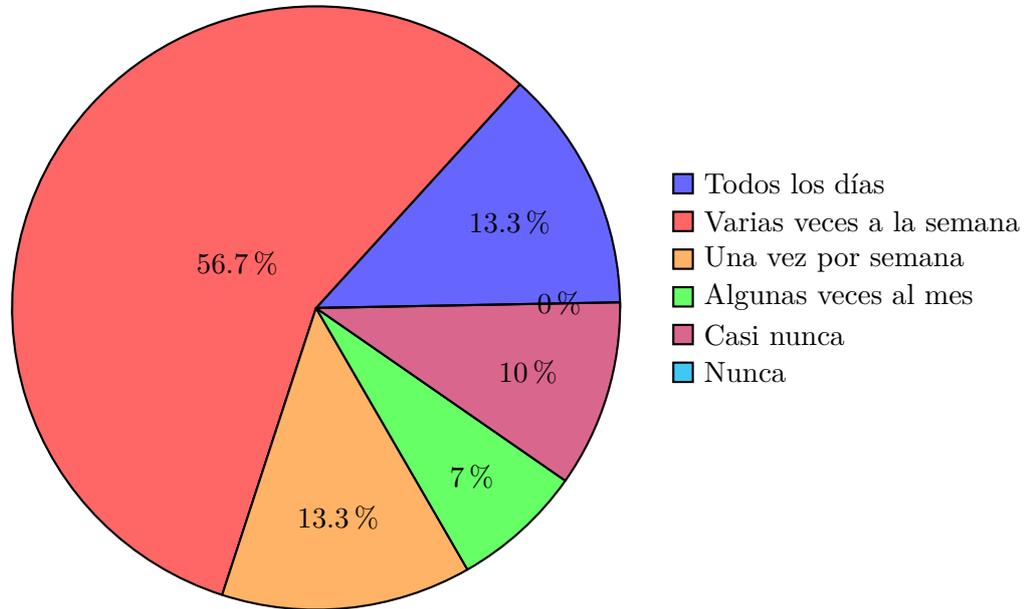


Figura 6.3: Resultados de la prueba: ¿Con qué frecuencia juegas videojuegos?

Entre los géneros que suelen ser jugados se encuentran los shooters, battle royale, RPG o de mundo abierto y juegos de celular o casuales. La Gráfica 6.4 también muestra que solo siete participantes juegan o han tenido algún acercamiento a los juegos MOBA, entre los cuales se incluye League of Legends.

¿Cuál(es) de estos géneros de videojuegos juegas con más frecuencia?

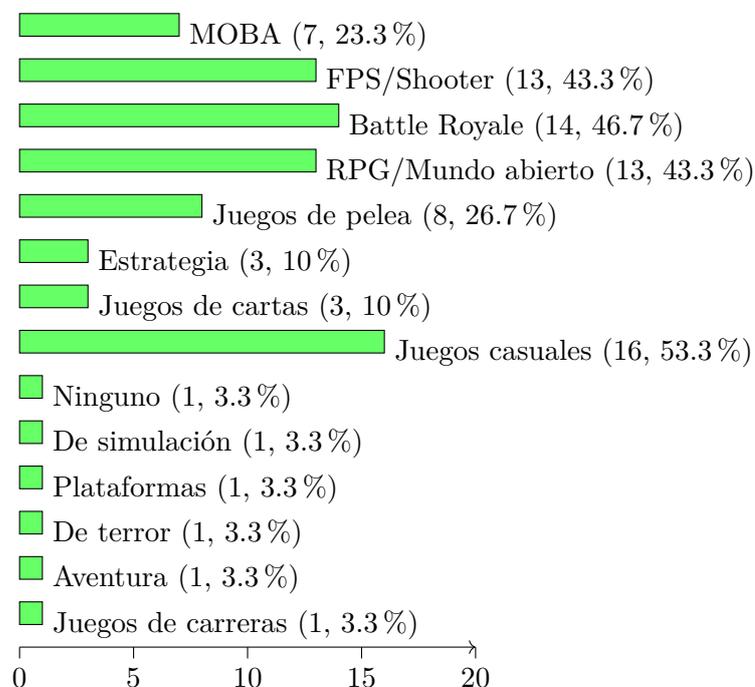


Figura 6.4: Resultados de la prueba: Frecuencia de juego por género

En este análisis inicial se infiere que los participantes no juegan o no han jugado League of Legends, lo que puede implicar que la gran mayoría de los participantes, al no tener experiencia con el videojuego, tengan una percepción sesgada sobre el sistema de recomendación debido al desconocimiento de los elementos y mecánicas básicas del juego. Por lo tanto, se les puede clasificar como jugadores casuales, quienes pueden iniciar en el juego sin ningún tipo de conocimiento y usar esta herramienta como una ayuda extra para empezar a jugar.

6.1.2.2. Experiencia con League of Legends

La gran mayoría de los usuarios participantes no ha tenido experiencia jugando League of Legends, tal como lo muestran la Gráfica 6.5 y la Gráfica 6.6, donde solo un 40 % ha jugado en algún momento el juego, siendo un 27 % quienes lo han jugado menos de 6 meses y el resto más de un año.

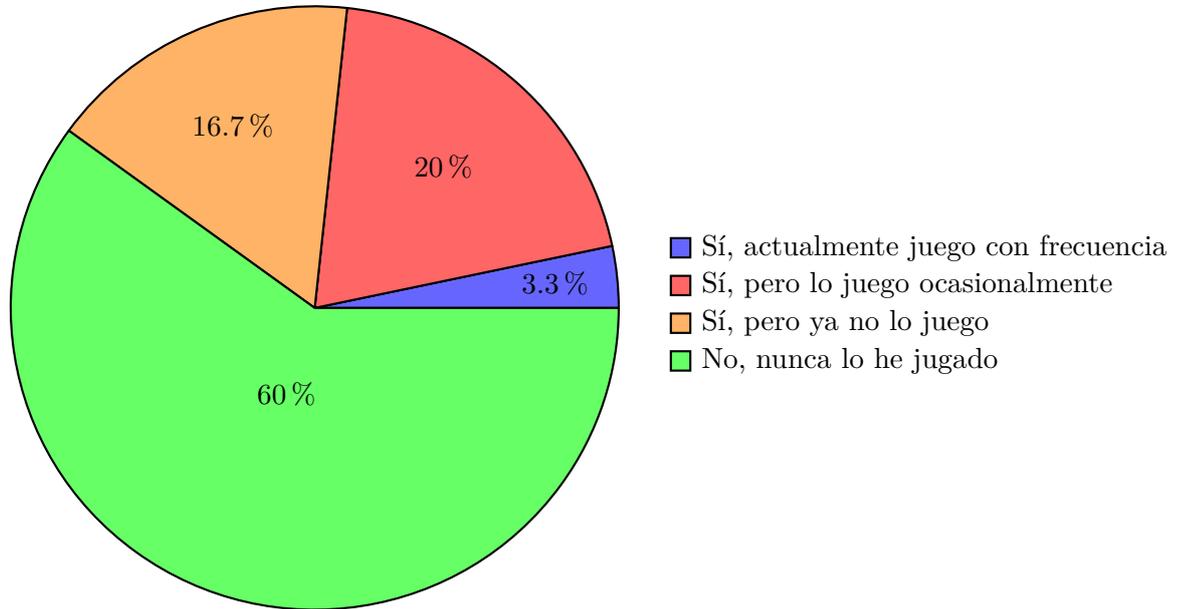


Figura 6.5: Resultados de la prueba: ¿Has jugado League of Legends?

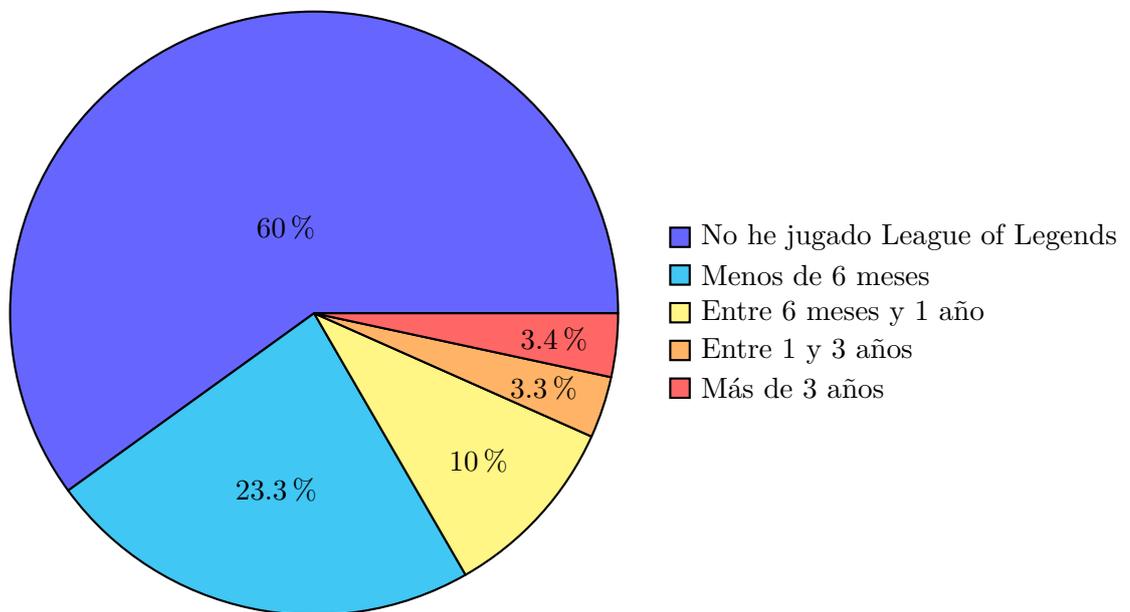


Figura 6.6: Resultados de la prueba: Si has jugado League of Legends, ¿cuánto tiempo llevas jugando?

6. RESULTADOS Y VALIDACIÓN

A partir de esto, la gran mayoría de los participantes son clasificados como jugadores casuales que han tenido poca o nula experiencia en el videojuego, y para quienes esta herramienta puede servir de ayuda para la selección estratégica de algún personaje que se adapte a sus características como jugadores de otros videojuegos y que se puedan adaptar a League of Legends.

6.1.3. Aplicación del cuestionario

Posteriormente, los participantes realizaron el cuestionario con las 10 sentencias de la prueba de validación SUS, adaptada al contexto del sistema de recomendación de personajes, y eligieron la opción que más se adaptaba a su percepción del sistema de recomendación de campeones.

La primera sentencia le pregunta al usuario si le gustaría usar este sistema de recomendación de campeones con frecuencia, a lo que el 68 % respondió de manera positiva, estando “Totalmente de acuerdo” o “De acuerdo” con la afirmación. Mientras que el resto de participantes se mantuvo neutral o en desacuerdo, la Gráfica 6.7 muestra que más de la mitad tuvo una aceptación positiva respecto a esta sentencia.

Por otro lado, la Gráfica 6.8 describe que solo el 10 % de los participantes notaron innecesariamente complejo el sistema para elegir un personaje adecuado, mientras que un 73 % afirma que no lo perciben de esa forma y que el sistema no es complejo para elegir un campeón.

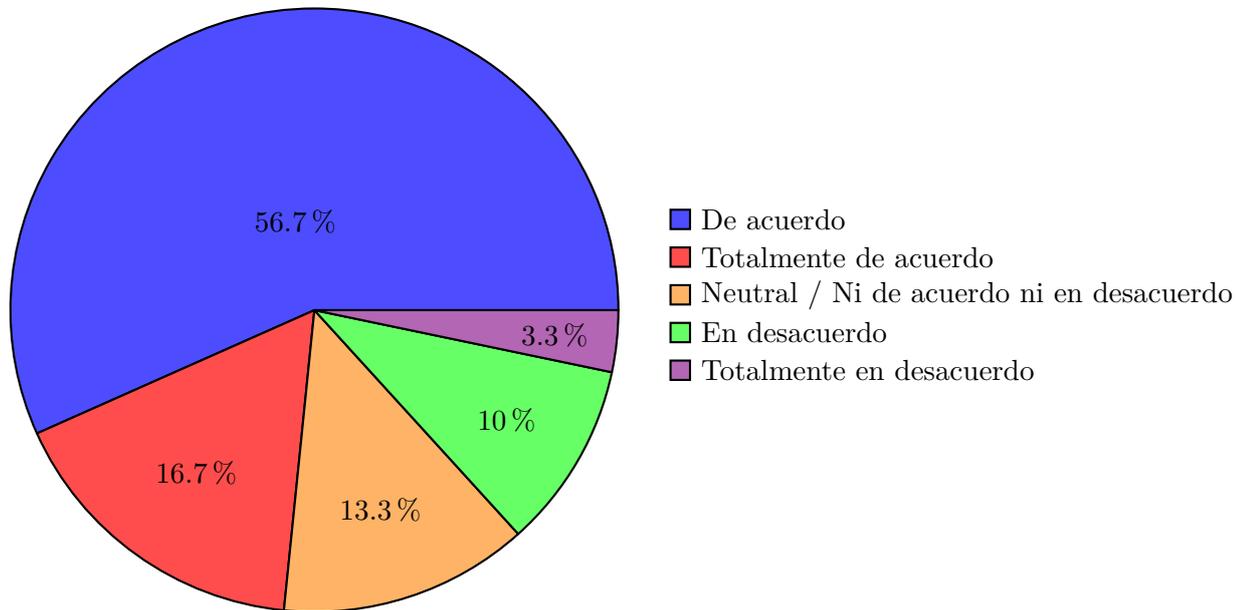


Figura 6.7: Creo que me gustaría usar este sistema de recomendación de personajes con frecuencia.

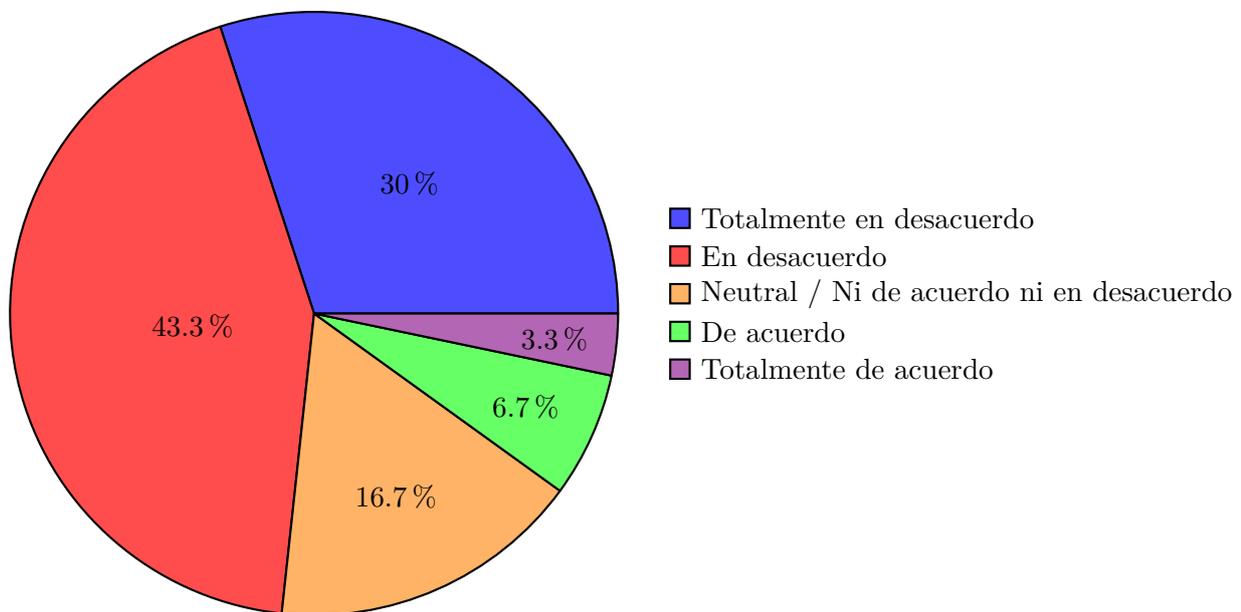


Figura 6.8: Encontré el sistema innecesariamente complejo para elegir un personaje adecuado.

6. RESULTADOS Y VALIDACIÓN

En la percepción sobre si el usuario pensaba que el sistema hacía fácil encontrar un personaje para jugar, un 77% de los participantes respondió que el sistema sí permite encontrar un personaje de forma fácil. Relacionado a esto, la siguiente afirmación era acerca de si el participante cree que necesitaría el apoyo de otra persona para entender cómo usar este sistema de recomendación. La Gráfica 6.9 describe que la gran mayoría de los participantes no cree necesitar ayuda para hacer uso de este sistema, lo que indica que este sistema simplifica el proceso de elección de un personaje de forma sencilla y sin ayuda de personas externas. Por otro lado, la Gráfica 6.10 muestra que un 16.7% de los participantes reconocen necesitar apoyo para usar el sistema; esto puede deberse a diferentes factores, como el desconocimiento de algunos aspectos del juego, por lo que el proceso de elección para algunas personas puede ser algo difícil.

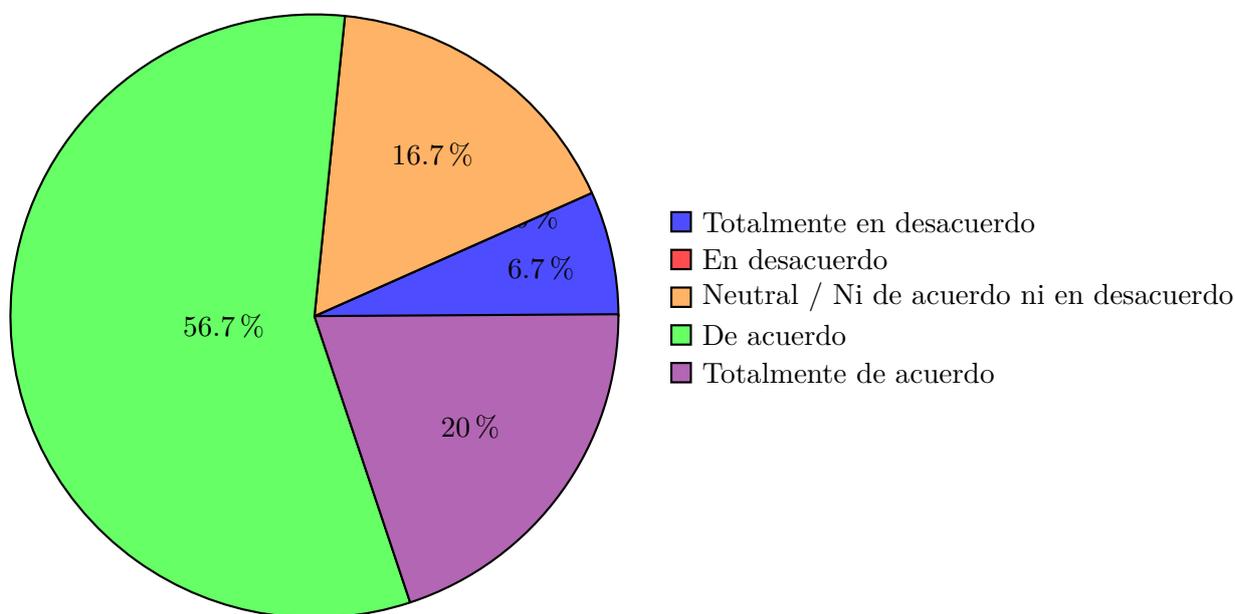


Figura 6.9: Pensé que el sistema hacía fácil encontrar un personaje para jugar.

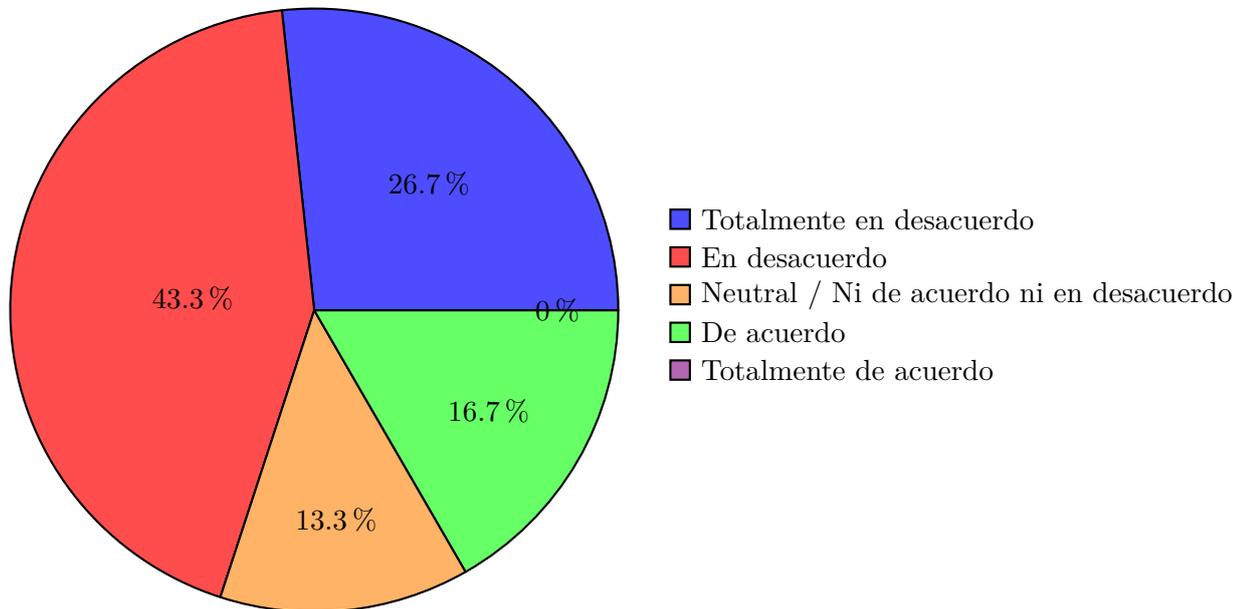


Figura 6.10: Creo que necesitaría el apoyo de otra persona para entender cómo usar este sistema de recomendación.

La Gráfica 6.11 indica que la percepción sobre la integración de las funcionalidades en el sistema es buena, al contar con un 90% que están “De acuerdo” o “Totalmente de acuerdo”, demostrando que las funcionalidades en el sistema están bien aplicadas y pueden solucionar la problemática principal de buena forma. Otro factor que puede indicar que las funcionalidades están implementadas de buena forma es el aspecto de qué tan rápido podrían aprender los usuarios a usar las diferentes funcionalidades del sistema. La Gráfica 6.12 indica que un sólido 80% de los participantes aprenderían a usar este sistema de recomendación rápidamente.

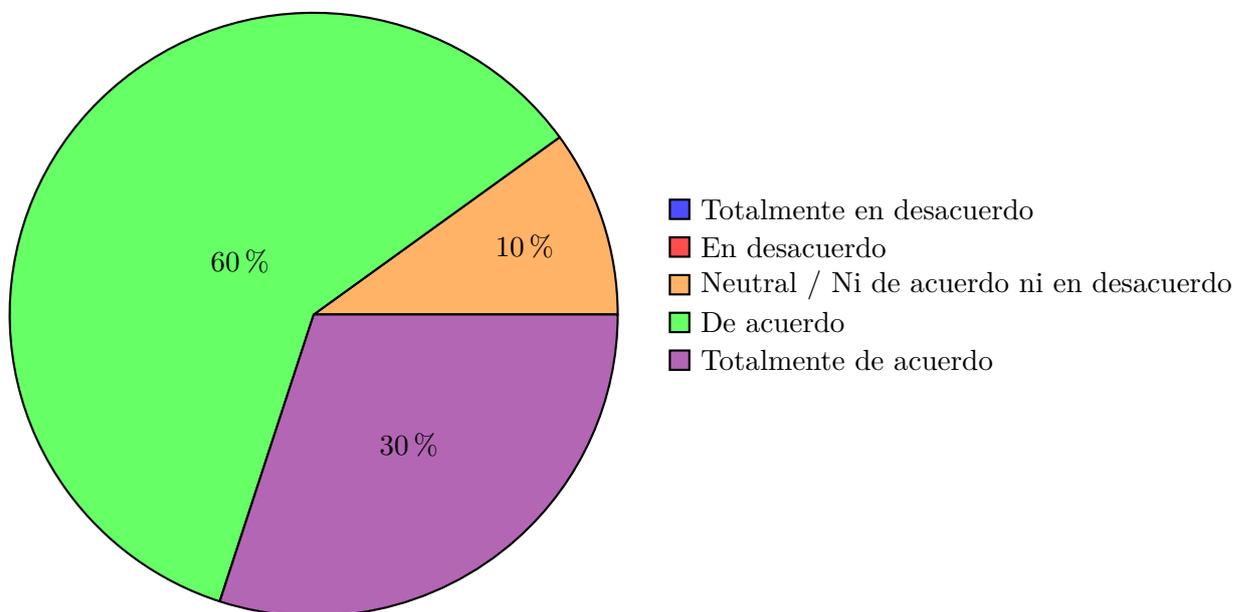


Figura 6.11: Encontré que las diferentes funciones del sistema estaban bien integradas.

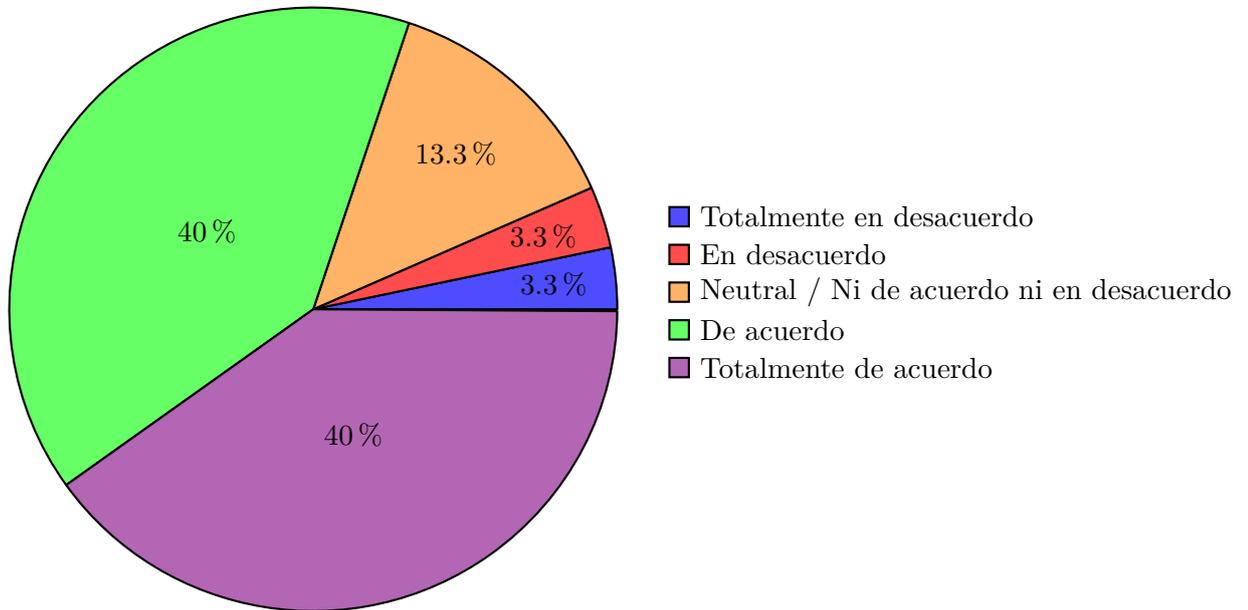


Figura 6.12: Imagino que la mayoría de los jugadores aprenderían a usar este sistema de recomendación rápidamente.

Una de las sentencias con respuestas más variadas es sobre si el usuario pensaba que había demasiada inconsistencia en las recomendaciones de campeones. La Gráfica 6.13 muestra que un 20% identificó inconsistencias sobre las recomendaciones de los campeones y un 27% se mantuvo neutral, mientras que poco más de la mitad respondió que no notaban esa inconsistencia. Esto significa que en un futuro será necesario realizar cambios en este aspecto, con el objetivo de que la gran mayoría de los usuarios tengan más confianza en las recomendaciones.

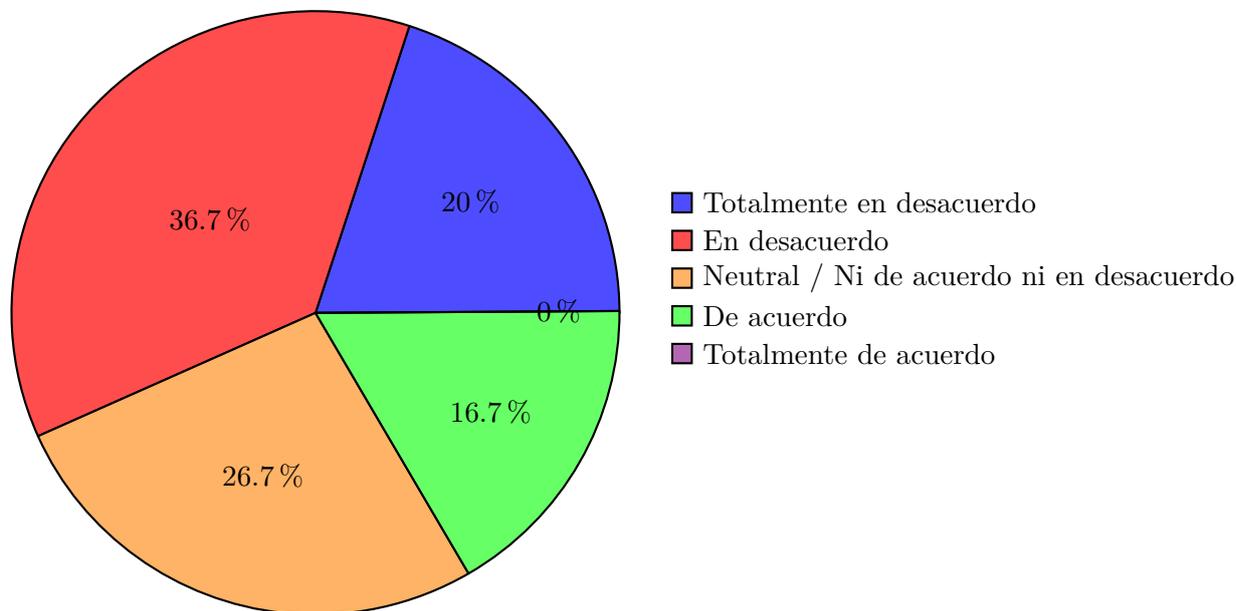


Figura 6.13: Pensé que había demasiada inconsistencia en las recomendaciones de personajes.

Uno de los aspectos clave del sistema es ser intuitivo para cualquier tipo de jugador. La Gráfica 6.14 confirma que este aspecto se cumple de forma positiva, al tener un 77% que afirma que el sistema no es complicado o poco intuitivo para elegir un campeón. Relacionado con la sentencia anterior, la Gráfica 6.15 identifica cómo ese mismo porcentaje se siente seguro al usar el sistema para encontrar un campeón adecuado a sus necesidades, siendo un 77% de aquellos que sí lo perciben de esta forma.

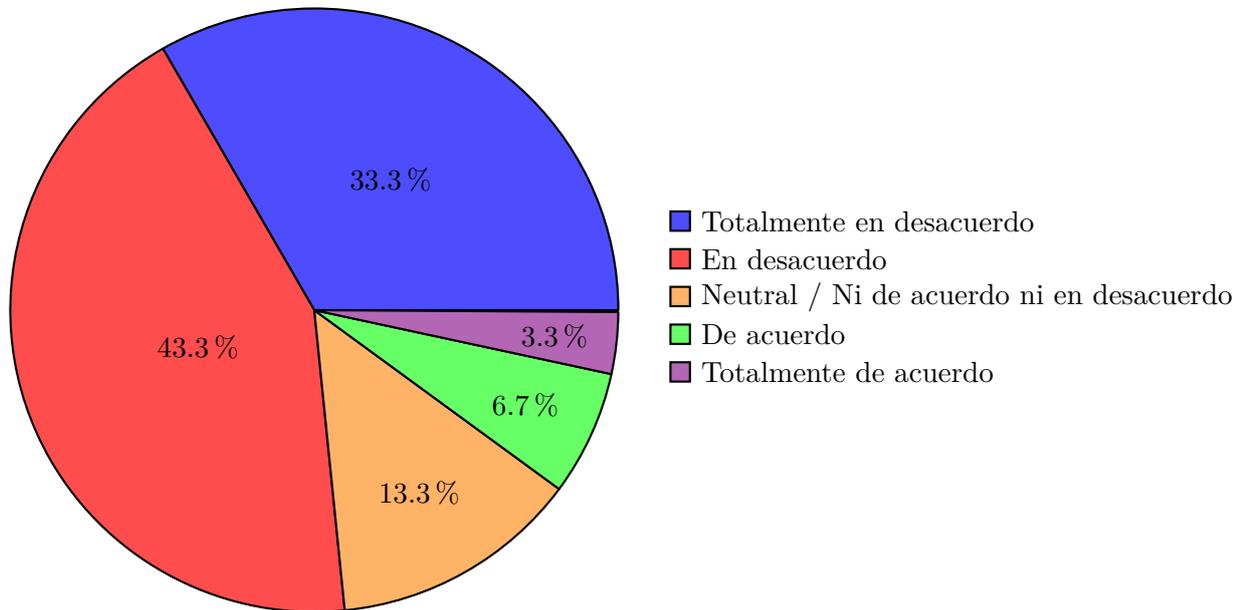


Figura 6.14: Encontré el sistema complicado o poco intuitivo para elegir un personaje.

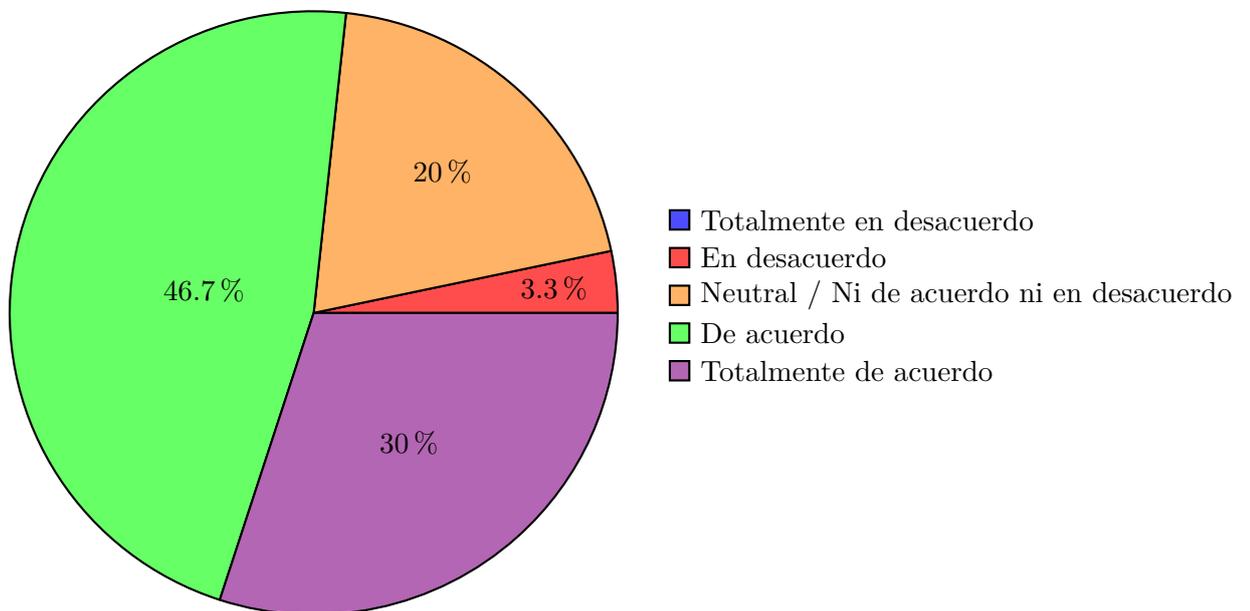


Figura 6.15: Me sentí seguro al usar el sistema para encontrar un personaje adecuado para mí.

Por último, en el aspecto de si los usuarios necesitan aprender muchas cosas antes

6. RESULTADOS Y VALIDACIÓN

de poder aprovechar bien este sistema de recomendación, la Gráfica 6.16 muestra que un gran sector de los participantes sí piensa que necesitan conocer más cosas para aprovechar las funcionalidades del sistema de buena forma, siendo solo un 53 % quienes sienten tener los conocimientos necesarios para sacarle el máximo provecho, siendo el porcentaje restante aquellos que tal vez no tengan la suficiente experiencia y consideren que el juego es complejo, por lo que necesitan conocer mejor las mecánicas de League of Legends para poder usar este tipo de sistemas de forma que les pueda beneficiar.

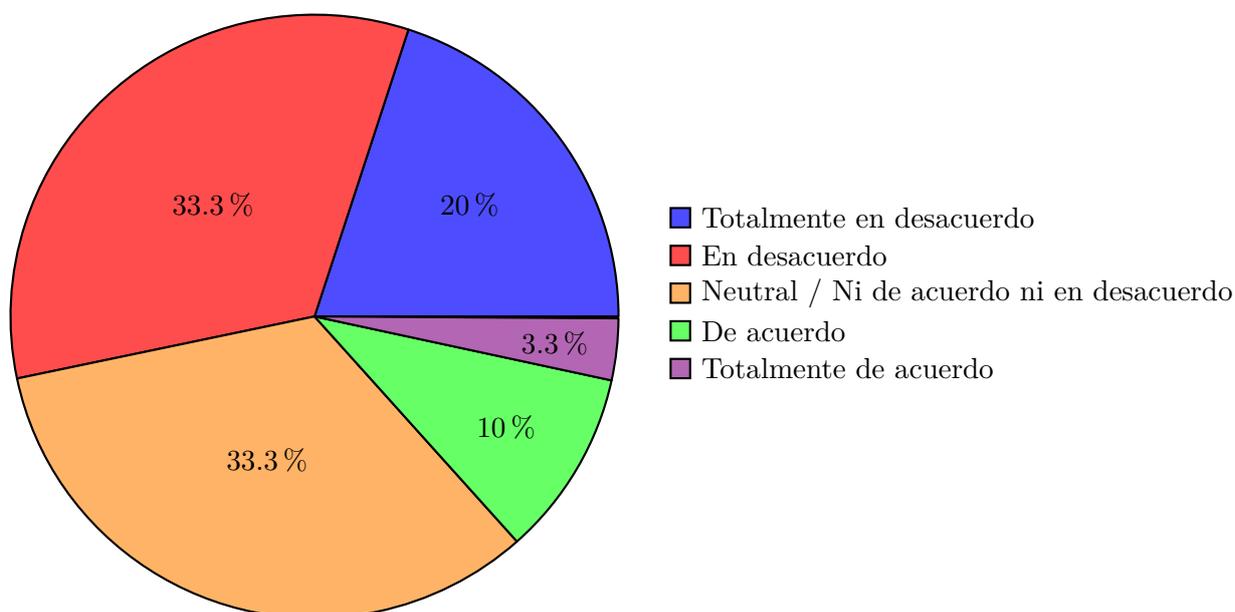


Figura 6.16: Necesité aprender muchas cosas antes de poder aprovechar bien este sistema de recomendación.

A partir de esto, se puede realizar un análisis rápido en el que se infiere que los usuarios perciben el sistema como fácil de usar, el cual cuenta con buenas funcionalidades, pero que no pueden ser bien aprovechadas debido al desconocimiento de algunos aspectos del juego. Por otra parte, es necesario trabajar en la veracidad de las recomendaciones del sistema, ya que se percibe que tal vez no sean del todo confiables, por lo que será necesario mejorar el proceso del manejo de información con respecto a las sugerencias y cómo son obtenidas.

6.1.4. Resultados y análisis

Obtener el promedio total de SUS permitirá reconocer la usabilidad del sistema en diversos aspectos, tanto de usabilidad, integración de funcionalidades y confianza en el

sistema por parte de los usuarios. Una vez obtenidas todas las respuestas por parte de los participantes en el proceso de validación, es necesario realizar algunas operaciones matemáticas que permitan obtener el promedio total de SUS.

6.1.4.1. Resultado de validación

Se asigna un valor a cada una de las respuestas y, posteriormente, se divide las preguntas en dos tipos (par o impar), y se realiza la suma de todos los puntos totales. En caso de ser par, se resta 25 al total, y si es impar, se resta 5 a la puntuación obtenida. Posteriormente, se suman los dos resultados obtenidos de las preguntas impares y pares, y este será multiplicado por 2.5. La Operación 6.1 muestra el SUS total de un participante; este proceso es aplicado para cada uno de los 30 usuarios que participaron en la prueba de validación.

$$\text{SUS}_{\text{participante}} = (16 + 15) \times 2.5 = 77.5 \quad (6.1)$$

Finalmente, al obtener el SUS por cada participante, es necesario obtener el promedio total de SUS. La Operación 6.2 muestra de forma simplificada el proceso de obtención del promedio, dando como resultado **68.25**. Según la Gráfica 6.1, al tener una puntuación mayor a 68, se considera como un sistema **Aceptable**.

$$\bar{X} = \frac{\sum_{i=1}^{30} X_i}{30} = \frac{77.5 + 65 + 60 + \dots + 85}{30} = 68.25 \quad (6.2)$$

6.2. Validación de la efectividad de las sugerencias

Uno de los enfoques principales del sistema es ofrecer sugerencias que le permitan a los usuarios obtener recomendaciones viables y óptimas, con el fin de que estos puedan jugar partidas de LoL con la confianza de que la sugerencia hecha por parte del sistema es de alta calidad, al ser obtenida a partir de datos de competiciones de alto nivel.

6.2.1. Validación cualitativa

Para verificar que las recomendaciones cuentan con la mejor viabilidad para ser usadas en un entorno real con jugadores de diversos niveles de habilidad y experiencia, se sometió a un grupo de 10 jugadores de League of Legends con diferentes habilidades y tiempo en el juego a probar el sistema con el objetivo de utilizar alguna de las sugerencias ofrecidas para, posteriormente, contestar un cuestionario de 5 preguntas que permitan obtener información sobre su experiencia con las recomendaciones ofrecidas por el sistema y si les fue funcional dentro de una partida.

6.2.1.1. Cuestionario

El cuestionario aplicado consta de 5 preguntas con escala de Likert, en la que cada respuesta tiene un valor asignado del uno al cinco, con lo que se busca obtener información sobre la experiencia usando una de las recomendaciones en una partida de League of Legends.

Pregunta 1. ¿Qué tanto cree que el campeón recomendado se adaptó a su estilo de juego?

- 1 - Nada adaptado
- 2 - Poco adaptado
- 3 - Neutral
- 4 - Bien adaptado
- 5 - Totalmente adaptado

Pregunta 2. ¿Qué tan satisfecho quedó con el rendimiento del campeón recomendado durante la partida?

- 1 - Nada satisfecho
- 2 - Poco satisfecho
- 3 - Neutral
- 4 - Satisfecho
- 5 - Muy satisfecho

Pregunta 3. ¿Considera que el sistema de recomendación le ayudó a mejorar su desempeño en la partida?

- 1 - Para nada
- 2 - Poco
- 3 - Neutral
- 4 - Sí
- 5 - Mucho

Pregunta 4. ¿Qué tan adecuados le parecieron los ítems y runas sugeridos por el sistema?

- 1 - Nada adecuados
- 2 - Poco adecuados
- 3 - Neutral
- 4 - Adecuados
- 5 - Muy adecuados

Pregunta 5. ¿Le gustaría seguir utilizando este sistema de recomendación en futuras partidas?

- 1 - Para nada
- 2 - Poco
- 3 - Neutral
- 4 - Sí
- 5 - Mucho

A partir de estas 5 preguntas se busca conocer si las sugerencias se adaptan a su estilo de juego y cuál fue su experiencia o desempeño utilizando el personaje, ítems y runas sugeridos por el sistema. Se busca obtener un promedio final de aprobación superior al 80 %, lo que indicaría que las sugerencias hechas por el sistema son viables para ser utilizadas en una partida de League of Legends.

6.2.1.2. Descripción de la muestra

Los participantes seleccionados para participar en esta prueba son jugadores habituales o que han jugado el juego de forma recurrente con distintas características, como estilo de juego o personajes preferidos.

La Gráfica 6.17 muestra que hay una distribución variada en cuanto al nivel de experiencia de los jugadores, siendo una mayoría del 40 % principiantes, mientras que el resto tiene un poco más de experiencia en el juego.

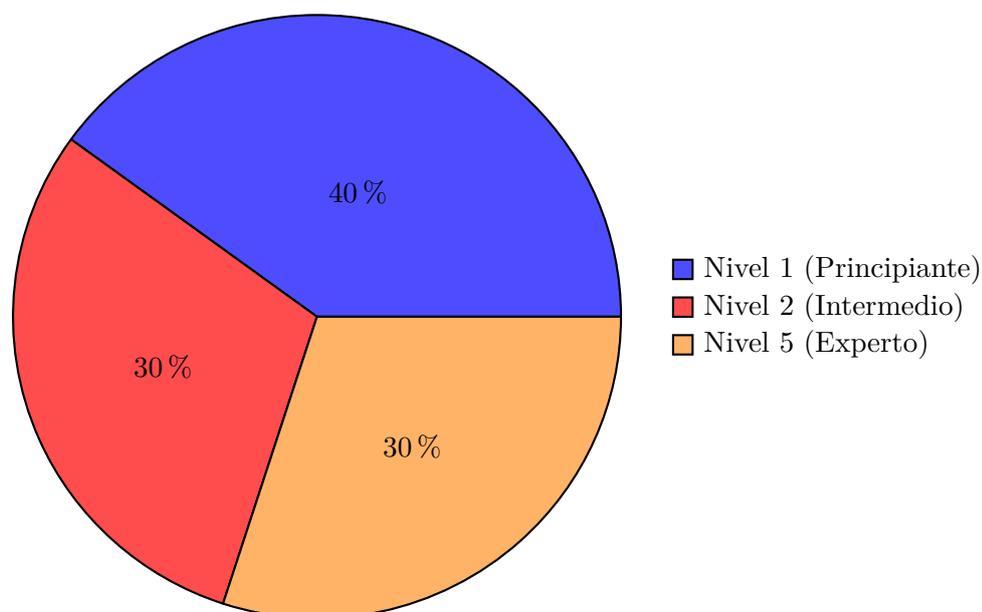


Figura 6.17: Distribución del Nivel de Experiencia

6. RESULTADOS Y VALIDACIÓN

Por otra parte, se les hizo un cuestionamiento acerca de su estilo de juego. La Gráfica 6.18 describe que la mayoría tiene un estilo más agresivo, siendo un 40 % de los participantes, mientras que el resto juega de forma equilibrada o más defensiva.

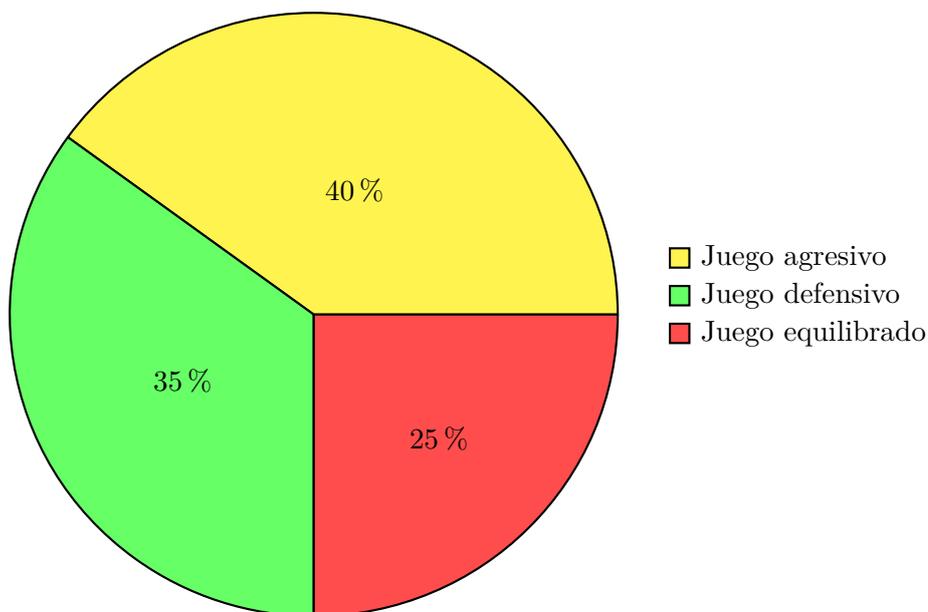


Figura 6.18: Distribución del Estilo de Juego

Estas preguntas de segmentación permiten conocer mejor las características respecto al juego de los participantes, dando oportunidad de conocer cuál es la percepción sobre las sugerencias del sistema de recomendación, con una muestra poblacional variada y con diversos rasgos dentro del juego.

6.2.1.3. Aplicación del cuestionario

Una vez que el grupo de encuestados realizó pruebas del sistema, jugando partidas de League of Legends con los personajes, runas e ítems proporcionados por el sistema, se les aplicó un cuestionario de 5 preguntas que busca conocer cuál fue su experiencia utilizando el sistema de recomendación y si las sugerencias proporcionadas les fueron de ayuda.

La primera pregunta busca saber si los participantes percibieron que las recomendaciones hechas por el sistema se adaptaron a su estilo de juego. La Gráfica 6.19 muestra que un 30 % se mostró neutral, mientras que con una percepción positiva de 70 %, los participantes respondieron que sintieron que la recomendación se adaptó a su estilo de juego, bien o totalmente.

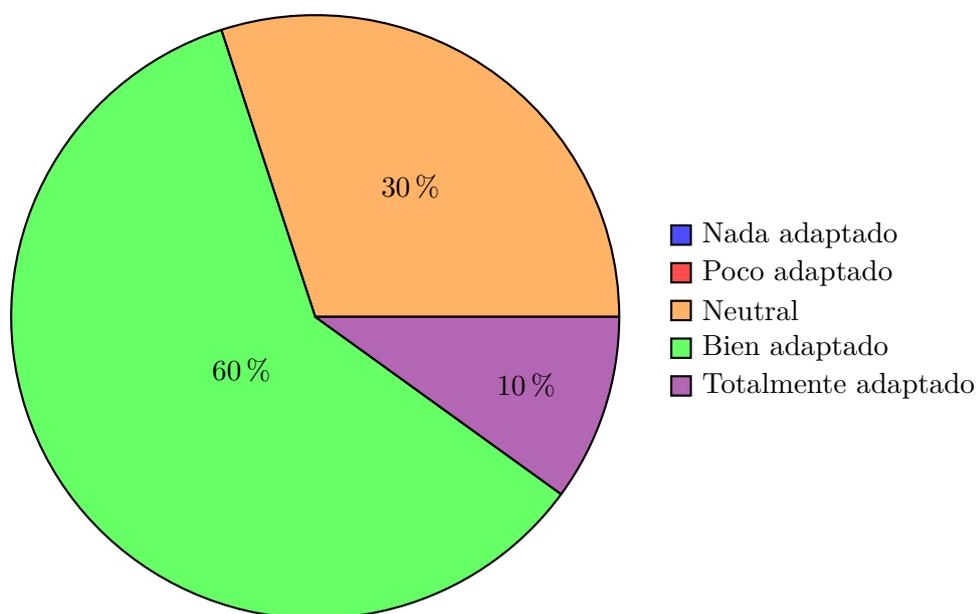


Figura 6.19: ¿Qué tanto cree que el campeón recomendado se adaptó a su estilo de juego?

Posteriormente, se buscó conocer cuál fue su nivel de satisfacción con el rendimiento de la recomendación del personaje durante una partida. La Gráfica 6.20 muestra que un 30% de los participantes se mostró neutral ante el cuestionamiento, mientras que un 60% respondió que se sintieron satisfechos con el desempeño de la recomendación, y el resto de los participantes se mostró muy satisfecho ante el rendimiento de la sugerencia hecha por el sistema.

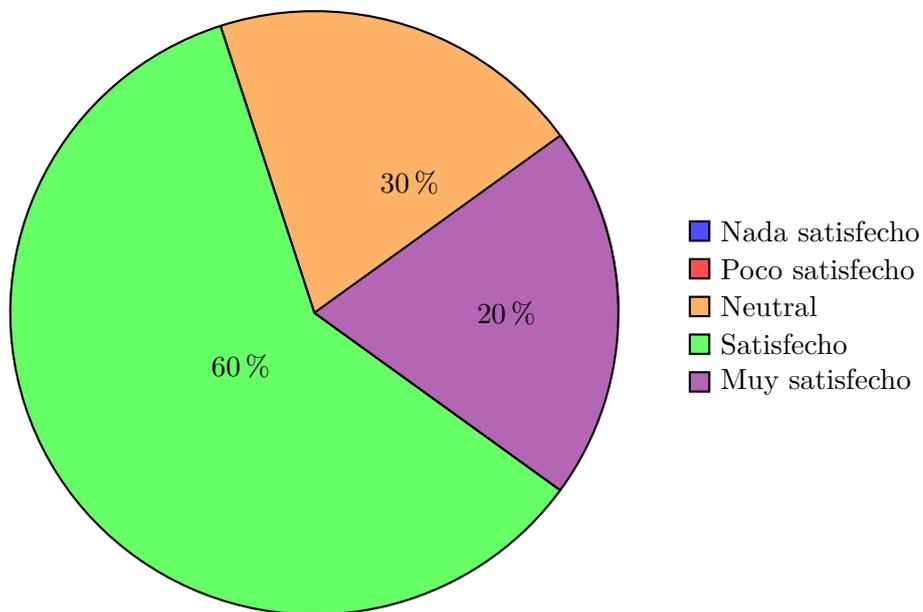


Figura 6.20: ¿Qué tan satisfecho quedó con el rendimiento del campeón recomendado durante la partida?

En la pregunta 3, los participantes respondieron considerando si la recomendación les ayudó a mejorar su desempeño dentro de una partida. La Gráfica 6.21 muestra que todos los participantes consideran que les ayudó a mejorar su desempeño de buena forma, lo que indica que las recomendaciones hechas por el sistema permiten mejorar el desempeño de los jugadores, en mayor o menor medida.

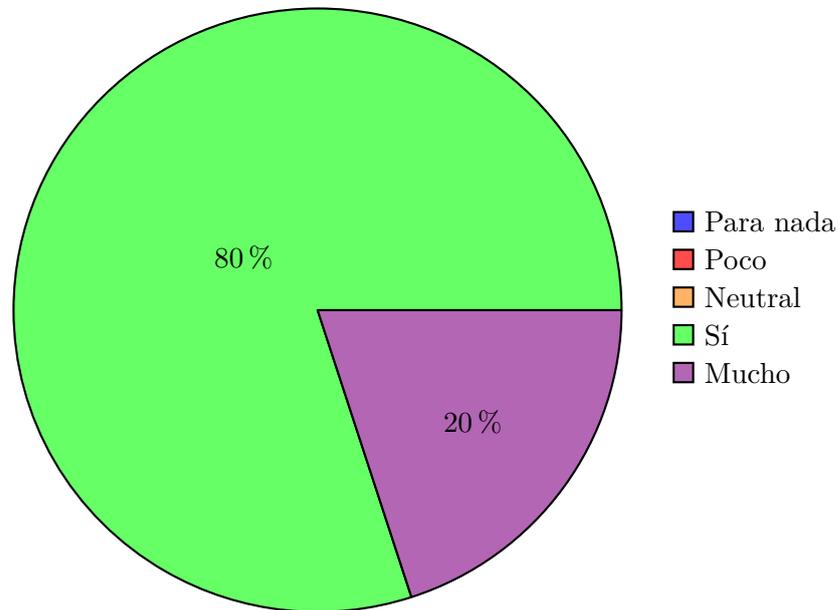


Figura 6.21: ¿Considera que el sistema de recomendación le ayudó a mejorar su desempeño en la partida?

El cuestionamiento 4 es acerca de la percepción de los jugadores sobre los ítems y runas sugeridos por el sistema experto. La Gráfica 6.22 muestra que el 100 % de los participantes respondieron de manera positiva que los ítems y runas sugeridas fueron adecuadas, demostrando que estas sugerencias tienen una buena efectividad en partidas.

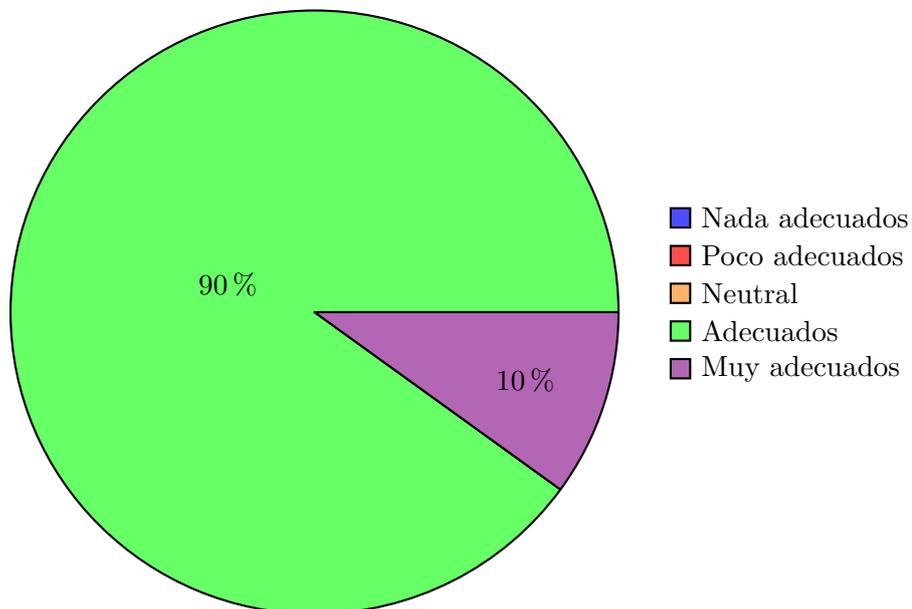


Figura 6.22: ¿Qué tan adecuados le parecieron los ítems y runas sugeridos por el sistema?

Por último, la pregunta 5 es acerca de la preferencia de este sistema para usarlo en un futuro. La Gráfica 6.23 muestra una aprobación favorable para seguir usando el sistema de recomendación de campeones de League of Legends, al contar con un 80% de los participantes que les gustaría usar el sistema en un futuro, mientras que los jugadores restantes se mostraron neutrales ante el cuestionamiento.

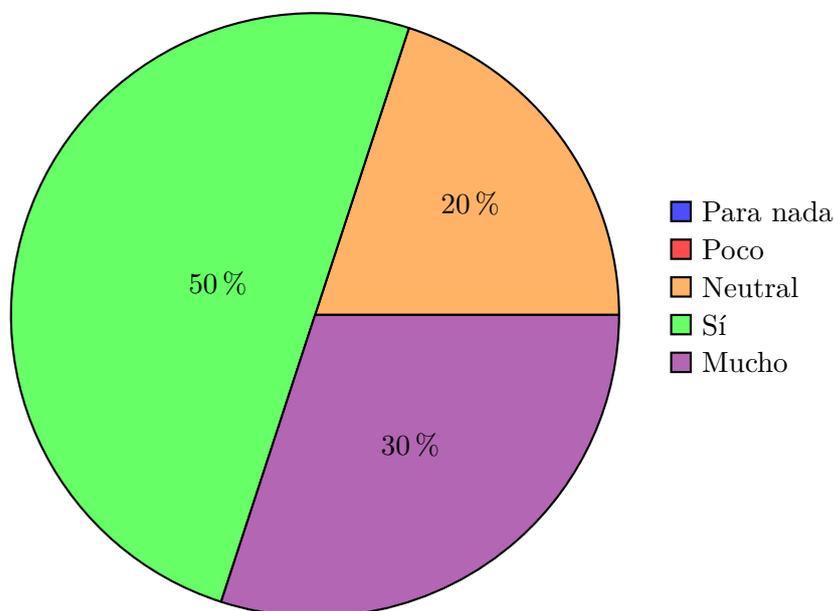


Figura 6.23: ¿Le gustaría seguir usando el sistema en el futuro?

A partir de las respuestas de este cuestionario se pueden identificar opiniones positivas acerca de las recomendaciones de personajes, runas e ítems ofrecidas por el sistema de recomendación, lo que, de primera mano, indicaría que tiene una aprobación que puede alcanzar los estándares deseados.

6.2.1.4. Resultados de la validación

Una vez realizada la prueba con jugadores de diferentes niveles de experiencia y estilos de juego de League of Legends, y tras contestar una serie de preguntas acerca de su experiencia con el sistema de recomendación, el método para identificar el nivel de aprobación de las recomendaciones hechas por el sistema experto se basa en obtener el promedio de cada respuesta por cada una de las preguntas y, al final, obtener un promedio general que permita conocer el índice de validación de las sugerencias. A partir de esto, la Tabla 6.1 describe los promedios obtenidos por cada pregunta, que a primeras impresiones demuestra valores positivos.

6. RESULTADOS Y VALIDACIÓN

| Pregunta | Promedio |
|----------|----------|
| P1 | 3.8 |
| P2 | 4.0 |
| P3 | 4.1 |
| P4 | 4.1 |
| P5 | 4.1 |

Tabla 6.1: Promedio por pregunta en escala de 1 a 5

Finalmente, a partir de los promedios obtenidos de cada pregunta, se obtuvo un promedio general que permitirá demostrar el porcentaje de validación de las sugerencias de personajes, runas e ítems del sistema experto. La Ecuación 6.3 muestra que el promedio de validación general es de 80.4%, lo que demuestra que tiene una validación por arriba de lo propuesto, indicando que las recomendaciones hechas por el sistema son viables y óptimas para su uso.

$$\bar{X} = \frac{3.8 + 4.0 + 4.1 + 4.1 + 4.1}{5} = \frac{20.1}{5} = 4.02 \Rightarrow \text{Porcentaje general} = \frac{4.02}{5} \times 100 = 80.4\% \quad (6.3)$$

A partir de esta validación, se pudo verificar que las sugerencias ofrecidas por el sistema experto tienen una aprobación positiva por parte de un grupo de jugadores con diversas características dentro del juego, obteniendo un 80.4% de validación, dejando en claro que las recomendaciones son efectivas y permiten a los jugadores obtener información que les ayude a mejorar dentro de sus partidas de League of Legends.

6.2.2. Validación cuantitativa

La calidad de las reglas puede ser medida a través de las métricas utilizadas en la ejecución del algoritmo Apriori, permitiendo identificar la calidad de las reglas desde el momento de su generación. Las métricas utilizadas para medir la calidad de las reglas son las siguientes:

- Soporte mínimo.
- Confianza mínima.

Estas métricas permiten filtrar u obtener diversas reglas del conjunto de datos analizado, según los valores asignados a cada métrica, lo que puede significar que dependiendo de la configuración se pueden obtener más o menos reglas de asociación con

niveles de efectividad diversos.

Soporte mínimo

El soporte mínimo permite filtrar las reglas que no aparecen con suficiente frecuencia, lo que permite reducir el número de reglas raras o irrelevantes. El valor utilizado para esta métrica es el que viene por defecto en la configuración del algoritmo, siendo 0.1, lo cual significa que una regla debe contar con una frecuencia de aparición de al menos un 10% de las partidas analizadas para ser considerada válida. Elegir un valor más bajo podría incluir reglas que disminuyan el nivel de efectividad de las recomendaciones, mientras que uno muy alto podría excluir relaciones relevantes que se encuentren en este umbral.

Confianza mínima

La confianza mínima es un umbral que permite aceptar aquellas reglas que tienen una probabilidad alta de cumplirse, por lo que si una regla tiene una confianza menor al mínimo establecido, se considera débil y es descartada. Para este caso, el valor de confianza mínima es de 0.85, lo que indica que solo se aceptarán aquellas reglas que tengan una probabilidad de cumplirse superior al 85%. Esto permite generar una confianza aceptable sobre las reglas, al tener una alta probabilidad de cumplirse.

Al aplicarse esta configuración, se pudieron obtener 175 reglas de asociación, las cuales representan contar con una frecuencia de aparición de al menos un 10% de las partidas analizadas y que cuentan con un valor de probabilidad de cumplirse de al menos 85%. Otro factor que permite identificar que esta configuración es la más adecuada es que, al aplicarse variaciones en los valores de las métricas, el número de reglas de asociación disminuía drásticamente o aumentaba en gran cantidad, lo que indicaba que más reglas estaban siendo aceptadas pero con valores que no representan los niveles de efectividad deseados.

6.3. Discusión

A partir de las pruebas de validación aplicadas, se puede observar que el sistema de recomendación de campeones de LoL, en el aspecto de usabilidad, destaca en lo siguiente:

- Fácil de usar.
- Interfaz intuitiva.
- Navegación fluida.
- Funcionalidades bien integradas.

6. RESULTADOS Y VALIDACIÓN

Cada una de estas fortalezas permite identificar que los usuarios no tendrán dificultades para hacer uso del sistema, al contar con un diseño que permite una buena usabilidad.

Por otra parte, se identificó como un área de mejora la percepción de los usuarios en la veracidad de las recomendaciones del sistema, ya que los usuarios tienen dudas respecto a la confiabilidad de las sugerencias hechas por el sistema. Por lo que es necesario mejorar diversos procesos para avanzar en este aspecto:

- Optimización y mejora de los algoritmos de análisis de datos.
- Hacer más robusta la base de conocimientos.
- Actualizaciones constantes.
- Manejo de información con respecto a cómo se obtienen y presentan las recomendaciones.

Al evaluar el sistema mediante la escala SUS (System Usability Scale), se obtuvo un promedio total de 68.25, lo que indica que el sistema se encuentra en el límite entre lo “Marginal” y “Aceptable”, dando a entender que, si bien es aceptable, es necesario aplicar diversas mejoras que permitan aumentar ese promedio de validación. Con futuras actualizaciones y evaluaciones constantes, el objetivo siempre será superar estas métricas, buscando el beneficio de los usuarios y que se sientan seguros al usar este sistema al tomar decisiones informadas que les permitan tener una mejor experiencia de juego.

Con respecto a las validaciones sobre el nivel de efectividad de las recomendaciones hechas por el sistema experto, se demuestra que son altamente efectivas, al contar con la aprobación por parte de jugadores de League of Legends con diversos rasgos de juego, como el nivel de experiencia. Siendo superior al 80 % de efectividad, demuestra que los jugadores pueden tener confianza en las sugerencias hechas y que pueden ser usadas en una partida de forma óptima, permitiendo generar diversos beneficios como una percepción en la mejora del rendimiento. Por último, la validación cuantitativa permitió evaluar las métricas utilizadas en el algoritmo Apriori para obtener reglas de asociación que sean altamente efectivas. Esto permitió identificar que cada una de las reglas obtenidas cuenta con una frecuencia de aparición de al menos el 10 % de los datos de las partidas analizadas; además, estas cuentan con una probabilidad de cumplirse superior al 85 %, lo que indica que las sugerencias hechas por el sistema experto son altamente viables y con una probabilidad alta de ser efectivas en una partida de League of Legends.

Conclusiones

La presente investigación, que abarca desde los fundamentos teóricos de los sistemas expertos hasta su aplicación práctica en el ámbito de los videojuegos, aplica la integración de la minería de datos y los sistemas expertos que permitan la obtención de conocimiento con el fin de realizar toma de decisiones informadas.

En el marco teórico, la investigación abarcó los sistemas expertos abordando sus características y componentes más importantes. Además de conocer las diferentes técnicas de minería de datos y los modelos predictivos y descriptivos, junto con las reglas de asociación y algoritmos como Apriori, FP-Growth y ECLAT, esta información fue importante para la posterior etapa de implementación del sistema y poder procesar los datos del juego. Además, se mencionaron las bases del caso de estudio, en este caso, el videojuego League of Legends (LoL), lo que permitió conocer mejor las características de este juego, así como los elementos que lo componen y mecánicas principales del juego.

El marco metodológico integra las principales características de algunas metodologías de desarrollo tales como ágiles tradicionales, siendo la metodología en cascada la elegida para su aplicación en este proyecto. Además de metodologías enfocadas en el análisis de datos como la metodología Knowledge Discovery in Databases (KDD), la cual se integró junto a la metodología en cascada para el desarrollo y análisis de datos.

El marco tecnológico permitió identificar las herramientas de desarrollo utilizadas para la implementación del sistema. Con herramientas como React para el frontend, Django para el backend, y PostgreSQL, permitiendo el desarrollo final de un sistema de recomendación en un ambiente web. En el caso del análisis y visualización de los datos se hizo uso de librerías especializadas para análisis de datos de Python, como Pandas, NumPy, y el software de visualización Power BI, las cuales permitieron obtener información relevante y conocimiento que permitió la creación de un sistema experto, basado en reglas de asociación, mediante el algoritmo Apriori.

El proceso de la implementación y aplicación de las metodologías elegidas para el desarrollo del sistema y análisis de datos implicó una serie de procesos que permitieron la aplicación de técnicas de minería de datos a un conjunto de datos. Mediante la apli-

cación del algoritmo Apriori y clustering a través del algoritmo K-means se obtuvieron 175 reglas de asociación, las cuales permitieron el desarrollo de la base de conocimientos y de inferencias. Posteriormente, se desarrolló un sistema web que permite obtener las características más importantes de un jugador de League of Legends, entre las que se encuentran el rol, estilo de juego y experiencia.

Como resultados obtenidos, el sistema experto montado en un ambiente web permite al usuario obtener sugerencias adaptadas a su estilo de juego, mediante un sencillo formulario e interfaz que permite mostrar la información necesaria para que el jugador pueda realizar una elección informada de un campeón.

A partir de la prueba de validación SUS, aplicada a 30 personas, se obtuvo un puntaje de 68.25, lo que según sus métricas clasifica el sistema como Aceptable. Esto refleja que es fácil de usar, con una interfaz intuitiva, navegación fluida y funciones bien integradas. Sin embargo, se detectó como área de mejora la confianza de los usuarios en la veracidad de las recomendaciones, por lo que es necesario optimizar los algoritmos de análisis de datos, ampliar y fortalecer la base de conocimientos, mantener actualizaciones constantes y explicar mejor cómo se obtienen y presentan dichas recomendaciones.

Por otro lado, las recomendaciones del sistema fueron bien recibidas: un 80% de aprobación por parte de 10 personas que las probaron en distintas partidas y luego respondieron un cuestionario sobre su experiencia en League of Legends. Además, las reglas obtenidas con el algoritmo Apriori mostraron una probabilidad de cumplirse de al menos un 80%, lo que las hace efectivas y útiles en diferentes situaciones de juego.

Finalmente, este proyecto sienta las bases para aplicar técnicas de minería de datos en distintos campos, en este caso los videojuegos, abriendo la puerta a futuras mejoras y desarrollos tanto en este ámbito como en otros.

Trabajo a futuro

El presente proyecto ha establecido una buena base para el desarrollo de sistemas de recomendación en el ámbito de los videojuegos, en este caso para League of Legends (LoL), sin embargo, existen diversas áreas de oportunidad en las que se podría mejorar:

- Mejora de los algoritmos de minería de datos: Probar diferentes alternativas a los algoritmos usados actualmente, como el uso de diferentes algoritmos como FP-Growth o ECLAT, o la implementación de técnicas de aprendizaje automático como algoritmos de aprendizaje profundo, que permitan mejorar el proceso de análisis de datos en el ámbito de tiempo, rendimiento y funcionalidad.
- Ampliación de la base de conocimientos: El sistema actual se basa en 175 reglas de asociación, por lo que aumentar la cantidad de datos analizados permitiría extraer

un mayor número de reglas y patrones, lo que podría conducir a recomendaciones más precisas y personalizadas.

- Actualizaciones constantes del sistema: Implementar mejoras o nuevas funcionalidades que le permitan al usuario mejorar su experiencia de usuario.
- Integración de otros videojuegos: Los diversos procesos desarrollados en este proyecto se pueden adaptar y aplicarse a otros juegos competitivos, permitiendo explorar otros videojuegos y su posible relación con la minería de datos.

La implementación de estas mejoras permitiría mejorar las debilidades con las que cuenta el sistema y también implementar funcionalidades que le permitan al usuario mejorar su experiencia en el uso de este sistema. Además, la posibilidad de explorar otros videojuegos y formas de aplicar diversas técnicas de minería de datos permitiría obtener conocimiento que permita a los usuarios mejorar.

A.1. Manual de Usuario

Este manual de usuario tiene como objetivo guiar a los usuarios en el uso del sistema de sugerencias de campeones (Personajes) del videojuego League of Legends (LoL), conocido como "PRO Build". Diseñado para proporcionar instrucciones detalladas y claras sobre las funcionalidades del sistema, usabilidad y cómo aprovechar al máximo el sistema de recomendación de campeones. Dirigido a:

- Jugadores de LoL: Jugadores con poca o nula experiencia en el videojuego que deseen conocer el funcionamiento del sistema de forma detallada.

A.1.1. Alcance del sistema

La aplicación web de sugerencias de campeones del videojuego LoL es un sistema que permite obtener, a partir de un cuestionario breve sobre las características más importantes de los jugadores, como estilo de juego o nivel de experiencia, recomendaciones personalizadas de campeones, runas y objetos más adecuados según las características de cada jugador. El sistema está basado en la aplicación de diversos algoritmos de Minería de Datos a un conjunto de datos obtenido a partir de una competencia profesional del videojuego, permitiendo obtener recomendaciones precisas y basadas en jugadores de alto nivel de experiencia.

Funcionalidades Principales

El sistema cuenta con dos diferentes funcionalidades que permiten al usuario obtener recomendaciones personalizadas.

- Sugerencias Personalizadas de Personajes: Recomendación de personajes a partir del rol, estilo de juego y nivel de habilidad.

- Sugerencias personalizadas de Objetos y Runas: Recomendación de Objetos y Runas adaptadas a cada personaje en específico y a las características de cada jugador.

Estas funcionalidades están diseñadas para ser accesibles tanto para jugadores casuales como para aquellos con conocimientos más avanzados, que requieran de alguna información extra que les pueda ser de utilidad.

A.1.2. Audiencia

El sistema de sugerencias de personajes de League of Legends está dirigido a una amplia variedad de usuarios, desde jugadores casuales hasta aquellos más competitivos. La audiencia principal incluye:

- Jugadores nuevos (principiantes): Usuarios que están comenzando a jugar League of Legends y necesitan ayuda para elegir campeones que se adapten a su nivel de experiencia.
- Jugadores ocasionales (intermedios): Público que ya tiene conocimiento básico del juego pero quiere conocer nuevos campeones o aquellos que buscan mejorar su desempeño con recomendaciones personalizadas.
- Jugadores avanzados (expertos): Jugadores con conocimientos más avanzados que buscan mejorar su selección de campeones para partidas competitivas o que deseen comparar campeones u objetos y analizar su posible rendimiento.

A.1.3. Requisitos del sistema

Al ser una aplicación web, permite que la gran mayoría de los usuarios puedan acceder a través de un navegador de internet mediante la URL de acceso al sistema, lo que permite que sea accesible desde cualquier dispositivo que cuente con acceso a internet y un navegador compatible.

Navegadores compatibles

Se recomienda utilizar navegadores actualizados y compatibles con las tecnologías modernas. A continuación, se detallan los navegadores soportados:

- Google Chrome
- Mozilla Firefox
- Safari
- Navegadores basados en Chromium (Edge, Opera, Brave)

Para una experiencia adecuada y sin interrupciones se recomienda:

- Navegador actualizado a la última versión para evitar problemas de compatibilidad o rendimiento.
- JavaScript habilitado: La aplicación requiere que JavaScript esté activado en el navegador para funcionar correctamente.

En caso de existir problemas al acceder a la aplicación, verificar si se está utilizando uno de los navegadores compatibles y que cumpla con los requisitos mencionados.

Acceso al sistema

El método para acceder al sistema es mediante la URL de acceso [A.1](#), utilizando un navegador de internet. Esta URL permite acceder a la página principal del sistema en donde se mostrará toda la información necesaria para hacer uso del sistema de recomendaciones.

Enlace A.1: URL de Acceso a PRO Build

```
1 https://pro-build.netlify.app/
```

A.1.4. Módulos Principales

El sistema de sugerencias “PRO Build” cuenta con 4 diferentes módulos, que permiten desde mostrar información y detalles sobre el sistema, hasta áreas específicas para obtener información sobre un campeón.

A.1.4.1. Landing Page

La Landing Page es la página principal del sistema en donde se muestran y describen las características más importantes del sistema, además de que es el elemento que ayuda a redireccionar a otros módulos del sistema. Las secciones de esta página son:

Header

Sección del sistema que permite acceder a los diferentes módulos del sistema mediante botones que indican la función de cada módulo. La Figura [A.1](#) muestra los siguientes elementos del header:

- Nombre del Sistema "Professional Rift Oracle".
- Botón “Inicio”, direcciona a la Landing Page del sistema.
- Botón "Pro Build", permite acceder al proceso de recomendación de campeones.



Figura A.2: Hero Section



Figura A.1: Header

Hero Section

Esta sección tiene como objetivo atraer al usuario con elementos visuales que sean atractivos y textos descriptivos de las funciones del sistema. La Figura A.2 muestra:

- Imagen de uno de los personajes más reconocidos del videojuego.
- Título y descripción de la función principal del sistema.

¿Cómo Funciona?

Sección enfocada en mostrar cuál es el funcionamiento del sistema mediante descripciones y elementos visuales como videos o imágenes que describan el funcionamiento del sistema y cuál es el resultado final al hacer uso del sistema. La Figura A.3 muestra una descripción de cuál es el objetivo del sistema y cuál es la problemática que busca resolver, además de un gif que muestra cuál es el funcionamiento y proceso para hacer uso del sistema y obtener una recomendación personalizada. Por otra parte, la Figura A.4 describe cuál es el funcionamiento del sistema, mediante una descripción que muestra el origen de los datos, funcionamiento y resultados al usar este sistema, representado por una imagen que permite observar el formato de las recomendaciones y la información que muestran.

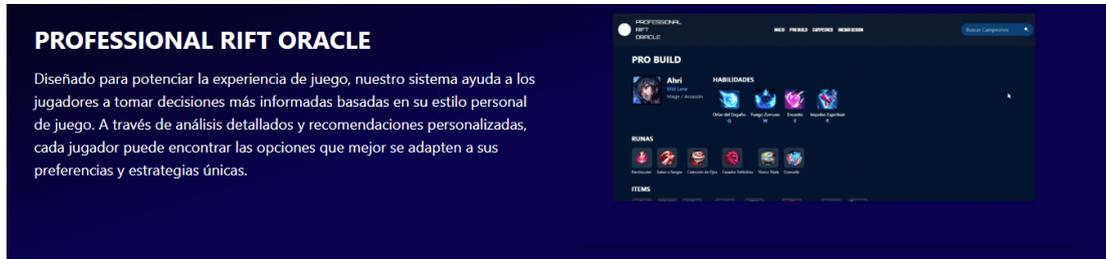


Figura A.3: Objetivo del Sistema

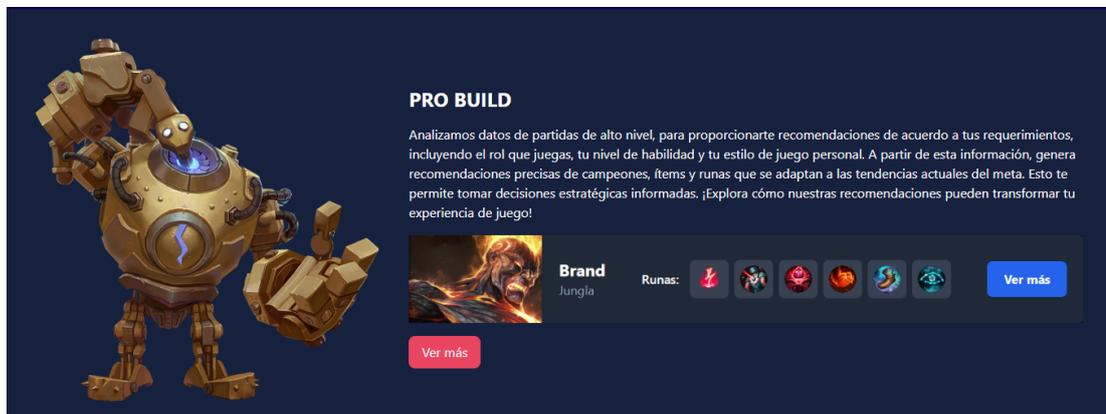


Figura A.4: Funcionamiento del sistema

A.1.4.2. Cuestionario de características de jugador

Sección del sistema que permite iniciar con el proceso de recomendación de personajes mediante un cuestionario de 3 preguntas que permite obtener detalles sobre las características de los jugadores. Se puede acceder mediante el botón de "Pro Build.^{en} el Header o mediante el Enlace [A.2](#).

Enlace A.2: URL de Acceso al cuestionario de características de jugador

1 <https://pro-build.netlify.app/pro-build>

Rol

Esta pregunta permite obtener información sobre la posición en el mapa o rol de cada jugador en el juego, con las siguientes 5 opciones:

- TOP LANE
- JUNGLA

- MID LANE
- AD CARRY
- SUPPORT

La Figura A.5 describe los elementos del cuestionario para la sección del Rol, en donde se muestra lo siguiente:

- Nombre de la sección y una descripción de las opciones para elegir.
- 5 opciones con los diferentes roles del juego, con el icono y nombre de cada rol y una descripción que ayuda a identificar mejor la posición en el mapa.
- Botones de navegación que permiten ir hacia la Landing Page o a la siguiente pregunta.

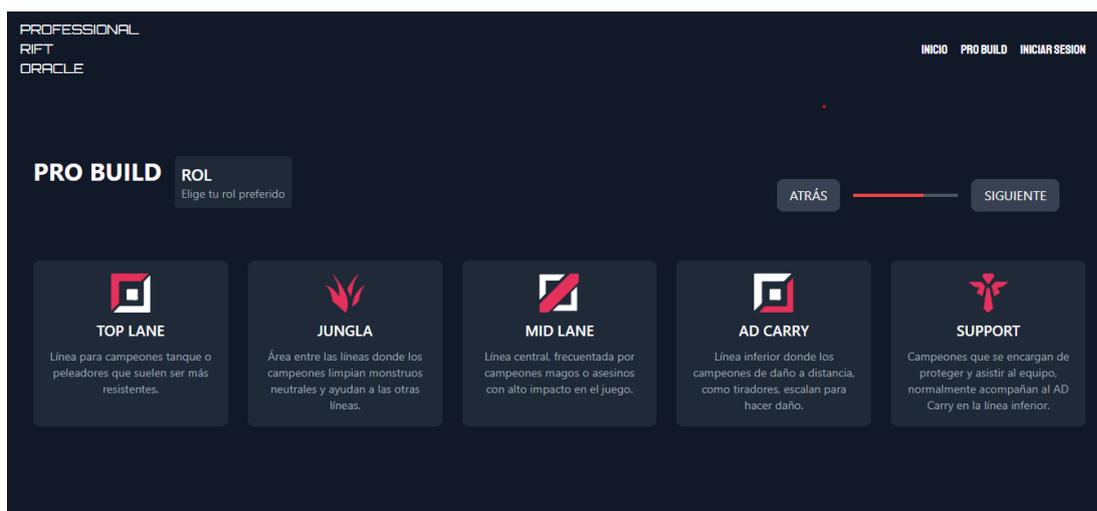


Figura A.5: Pregunta para elegir Rol

Estilo de juego

Permite conocer cuál es el estilo de juego de cada jugador, con las siguientes 3 opciones:

- Agresivo: Ideal para quienes buscan atacar constantemente.
- Balanceado: Enfocado en la supervivencia y control del juego.
- Defensivo: Jugadores que controlan el mapa y toman decisiones estratégicas.

La Figura A.6 muestra las 3 opciones disponibles para esta pregunta, además de un video que permite demostrar de forma visual el estilo de juego correspondiente a cada opción.

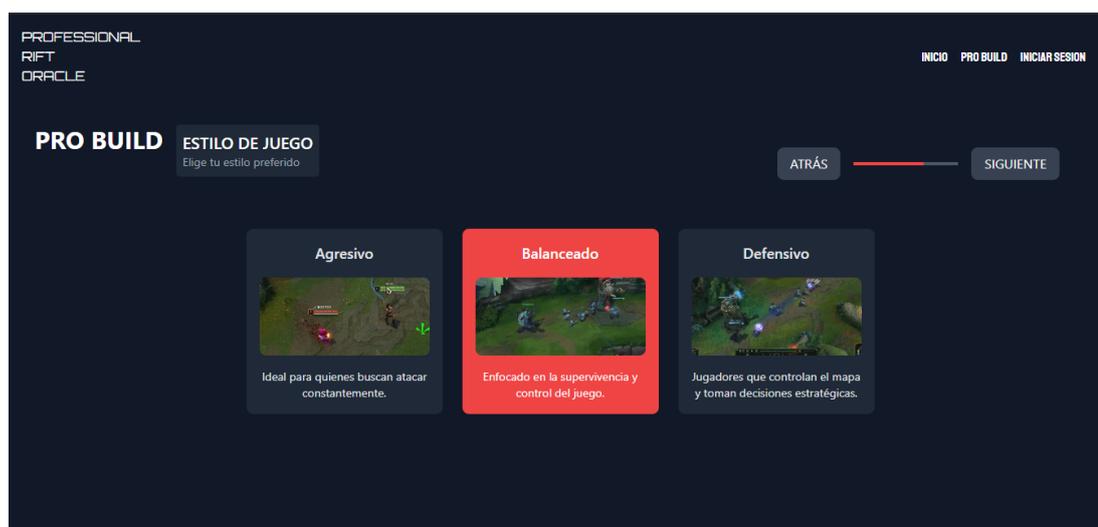


Figura A.6: Pregunta para elegir Estilo de juego

Nivel de experiencia

El usuario puede elegir su nivel de experiencia en el juego mediante las siguientes 3 opciones:

- Principiante: Ideal para aquellos nuevos en el juego.
- Intermedio: Jugadores que tienen experiencia básica.
- Experto: Jugadores muy experimentados y estratégicos.

La Figura A.7 muestra las 3 opciones disponibles para el nivel de experiencia, además de un botón que permite terminar con el cuestionario de características de jugador y pasar a la sección de recomendaciones.



Figura A.7: Pregunta para seleccionar nivel de experiencia

A.1.4.3. Sección de recomendaciones

Una vez concluido el proceso de recabado de las características de cada jugador mediante un cuestionario, con la información obtenida el sistema mostrará 5 recomendaciones de acuerdo con la información que el usuario seleccionó. La Figura A.8 exhibe cada uno de los elementos de la sección de recomendaciones.

- Información: Muestra la información recabada en el cuestionario de características del jugador.
- Información básica: Información general de los campeones sugeridos, en donde se muestra la imagen del personaje, nombre y rol de cada uno.
- Runas: Sugerencia de objetos adaptados a cada personaje y características de cada jugador.
- Ver más: Botón que permite tener más información de una sugerencia en específico.

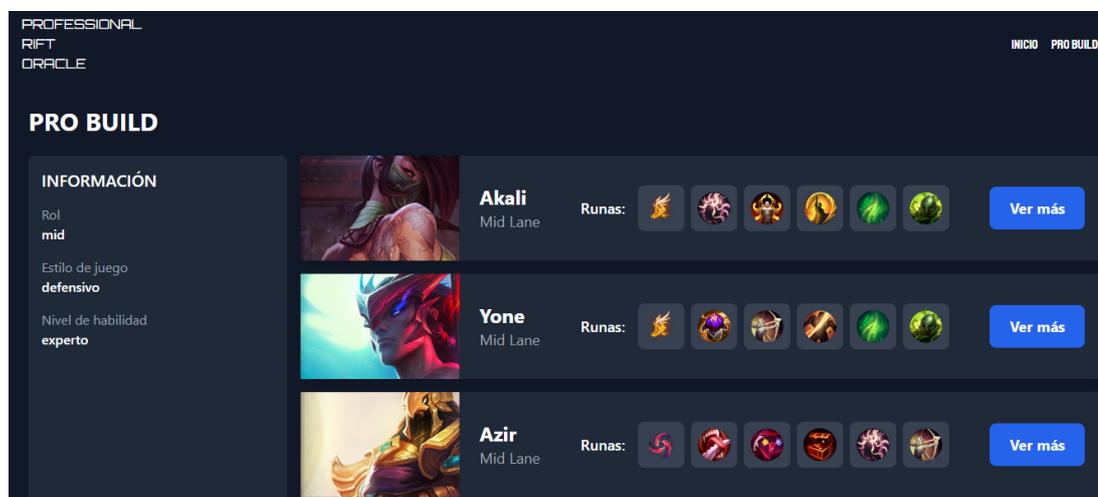


Figura A.8: Sección de Recomendaciones

A.1.4.4. Detalles sobre la recomendación

Permite al usuario tener más información sobre un personaje en específico, donde encontrará lo siguiente:

- Información general del campeón: Nombre, rol y tipo de personaje.
- Habilidades: Habilidades predefinidas de cada personaje dentro del juego.
- Runas: Objetos y habilidades que el jugador puede elegir para su personaje antes de entrar a una partida de LoL.
- Ítems: Objetos y habilidades que el jugador puede comprar para su personaje dentro de una partida.

La Figura A.9 muestra los distintos elementos de esta sección, en donde destacan las runas e ítems como elementos principales al ser parte de las sugerencias personalizadas del sistema de recomendación, adaptadas a cada personaje y características de cada jugador.

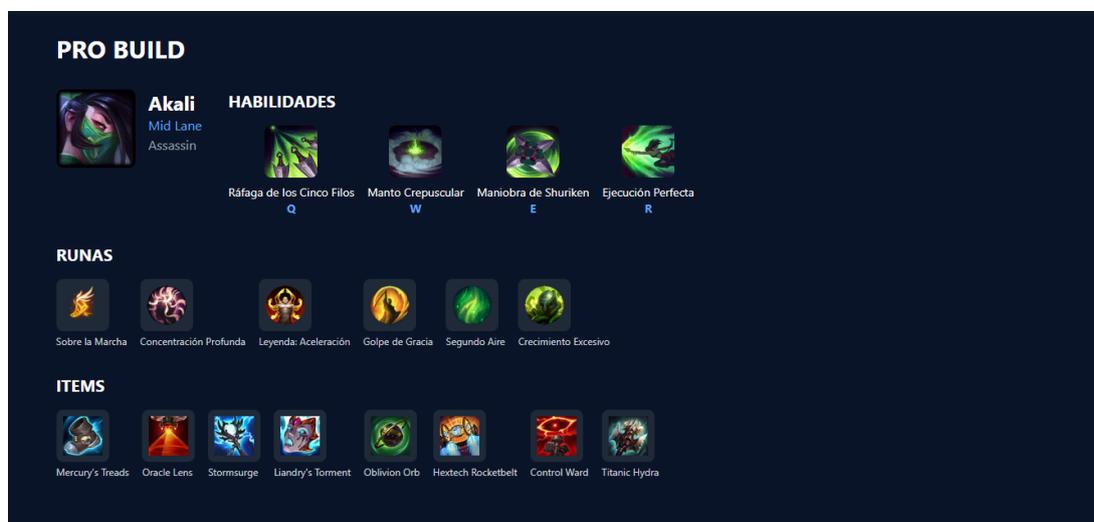


Figura A.9: Detalles sobre un campeón

Cada uno de estos módulos conforma el sistema de recomendaciones “Pro Build”, los cuales permiten seguir un proceso lineal en el que el usuario primero responde un cuestionario para después obtener diferentes sugerencias que le permitan optar por alguna de estas y obtener información más detallada de la recomendación en específico.

A.1.5. Solución de problemas

Al ser un sistema en constante cambio, es susceptible a diferentes errores, ya sean externos como problemas de navegador o de la API de “Riot Games”, o problemas internos como errores en las sugerencias.

A.1.5.1. Errores comunes

A continuación, se enlistan los posibles errores y soluciones a las problemáticas más comunes que se pueden presentar al momento de hacer uso del sistema:

La aplicación no carga correctamente

- Actualiza tu navegador a la última versión.
- Limpia la caché y las cookies del navegador.
- Intenta acceder desde otro navegador compatible.

Las sugerencias de personajes no se muestran

- Verifica que hayas seleccionado correctamente tus preferencias (rol, estilo de juego, etc.).
- Asegúrate de tener una conexión a internet estable.
- Si el problema persiste, recarga la página o cierra y vuelve a abrir la aplicación.

La aplicación es lenta o se congela

- Cierra otras pestañas o aplicaciones que consuman muchos recursos.
- Reinicia tu navegador o dispositivo.
- Si el problema persiste, contacta al soporte técnico.

A.1.5.2. Preguntas frecuentes (FAQ)

¿Cómo se generan las sugerencias de personajes?

Las sugerencias se basan en tus preferencias (rol, estilo de juego, nivel de dificultad) y en datos de partidas de jugadores profesionales.

¿La aplicación está actualizada con los últimos parches de LoL?

La aplicación se actualiza periódicamente con el fin de mostrar la información más reciente de cada actualización.

¿Puedo usar la aplicación en mi dispositivo móvil?

Sí, la aplicación es compatible con dispositivos móviles a través del navegador. Sin embargo, para una mejor experiencia, se recomienda usarla en una computadora.

¿Es gratuita la aplicación?

Sí, la aplicación es completamente gratuita para todos los usuarios.

A.1.5.3. Contacto de soporte

En caso de encontrar un error o problema que esté afectando la experiencia de usuario, el método de contacto es a través de correo electrónico.

Correo electrónico

Envía un correo a **spectrxtt@gmail.com** con una descripción detallada del problema. Incluye capturas de pantalla si es posible.

Una vez resuelto el problema, se te notificará que el error ha sido corregido. Con tus reportes, el sistema podrá mejorar para todos y ofrecer un mejor servicio.

Referencias

- [1] ActiveState. How to install scikit-learn, 2021. URL <https://www.activestate.com/resources/quick-reads/how-to-install-scikit-learn/>. 65
- [2] Alf. Manual de matplotlib en python, 2020. URL <https://aprendeconalf.es/docencia/python/manual/matplotlib/>. Accessed: 2024-10-23. 59
- [3] Alf. Manual de numpy en python, 2022. URL <https://aprendeconalf.es/docencia/python/manual/numpy/>. 58
- [4] Amazon Web Services. ¿qué es la minería de datos? La minería de datos, explicada, n.d. URL <https://aws.amazon.com/es/what-is/data-mining/>. Recuperado de Amazon Web Services. 15
- [5] AS. League of legends para novatos: conceptos básicos, 2017. URL https://esports.as.com/league-of-legends/ver-partido-League-of-Legends_0_1081691831.html. 22
- [6] Asana. ¿qué es kanban?, 2024. URL <https://asana.com/es/resources/what-is-kanban>. Última consulta: 28 de agosto de 2024. 40, 42
- [7] Atlassian. Git bash: definición, comandos y primeros pasos, 2024. URL <https://www.atlassian.com/es/git/tutorials/git-bash>. Último acceso: 13 de octubre de 2024. 49
- [8] Sebastián Badaro, Leonardo Javier Ibañez, and Martín Jorge Agüero. Sistemas expertos: Fundamentos, metodologías y aplicaciones. *Universidad de Palermo, Facultad de Ingeniería*, 10 2013. Fecha de recepción: julio 2013; Fecha de aceptación: octubre 2013. 11
- [9] Eugenia Bahit. *Scrum y eXtreme Programming para Programadores*. No hay publisher, Buenos Aires, Argentina, 2012. ISBN 1205011563509. URL <https://safecreative.org/work/1205011563509>. Distributed under Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License. 38, 40

REFERENCIAS

- [10] Bimático. ¿para qué se utiliza power bi, qué es capaz de hacer y dónde descargarlo? <https://www.bimatico.com/es/bi-news/para-que-se-utiliza-power-bi-que-es-capaz-de-hacer-y-donde-descargarlo>, 2019. 63
- [11] Red Bull. ¿cuáles son las posiciones en league of legends?, 2023. URL <https://www.redbull.com/es-es/posicion-league-of-legends>. 25
- [12] Red Bull. La historia de League of Legends: del mod de Warcraft III al fenómeno mundial, n.d. URL <https://www.redbull.com/mx-es/historia-league-of-legends>. 22
- [13] Cris Busquets. Medir la usabilidad con el sistema de escalas de usabilidad (SUS), n.d. URL <https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>. 116
- [14] Enrique Castillo, José Manuel Gutiérrez, and Ali S. Hadi. *Sistemas Expertos y Modelos de Redes Probabilísticas*. Universidad de Cantabria, Santander, España, 1996. 11, 12
- [15] CEI. ¿qué es figma?, n.d. URL <https://cei.es/que-es-figma/>. 48
- [16] Administrador CEUPE. ¿qué es un sistema informático? componentes, características y ejemplos, 2024. URL <https://www.ceupe.com/blog/sistema-informatico.html>. Consultado el 22 de julio de 2024. 7, 8
- [17] Ciencia de Datos. Reglas de asociación y algoritmo Apriori con R, n.d. URL https://www.cienciadedatos.net/documentos/46_reglas_de_asociacion_y_algoritmo_apriori. Recuperado de [cienciadedatos.net](https://www.cienciadedatos.net). 20
- [18] Mozilla Contributors. Introducción a express/node, n.d.. URL https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction. Accessed: 2024-10-13. 49
- [19] Mozilla Contributors. Primeros pasos en react, n.d.. URL https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started. Accessed: 2024-10-13. 49
- [20] Edgar Corona Organiche, Abraham Jorge Jiménez Alfaro, and Griselda Cortés Barrera. Principales metodologías en el desarrollo de proyectos de minería de datos. In *Actas de la Conferencia Nombre de la Conferencia*, n.d. 44, 45, 46
- [21] DAIL. Sistemas expertos: Usos comerciales, s.f. URL <https://www.dail.es/sistemas-expertos-usos-comerciales/>. 14
- [22] DataCamp. Association rule mining in python: Complete guide, 2021. URL <https://www.datacamp.com/es/tutorial/association-rule-mining-python>. 64

-
- [23] DataCamp. Tutorial de pandas en python: La guía definitiva para principiantes. <https://www.datacamp.com/es/tutorial/pandas>, 2024. 56
- [24] DataScientest. Pandas: La biblioteca de python dedicada a la data science. <https://datascientest.com/es/pandas-python>, 2022. 55
- [25] DataScientest. Numpy: la biblioteca python, 2024. URL <https://datascientest.com/es/numpy-la-biblioteca-python>. 58
- [26] Universidad de Valladolid. Introducción a matplotlib, n.d. URL https://www2.eii.uva.es/fund_inf/python/notebooks/Bibliotecas/01_Introduccion_a_Matplotlib/Introduccion_a_Matplotlib.html. 60
- [27] El Baúl del Programador. Aprendizaje no supervisado: Reglas de asociación, 2023. URL <https://elbauldelprogramador.com/aprendizaje-nosupervisado-reglas/#algoritmo-eclata-idsec-3-3-2-namesec-3-3-2a>. 21
- [28] Matplotlib Developers. Instalación de matplotlib, n.d.. URL <https://matplotlib.org/stable/install/index.html>. Accessed: 2024-10-23. 60
- [29] NumPy Developers. Instalar numpy, n.d.. URL <https://numpy.org/es/install/>. 59
- [30] DiagramasUML. ¿qué es el kdd o proceso de descubrimiento de conocimiento? <https://diagramasuml.com/que-es-el-kdd-o-proceso-de-descubrimiento-de-conocimiento/>, 2019. Recuperado el 31 de agosto de 2024. 43
- [31] Digital Talent Agency. Modelo waterfall o en cascada, 2018. Tema 1: Metodologías de Gestión de Proyectos. 36, 37
- [32] Douran Biraki. Figma: Diseño web, 2024. URL <https://www.hackaboss.com/blog/figma-diseno-web>. Accedido el 18 de septiembre de 2024. 48
- [33] Dremio. Conexión a dremio desde microsoft power bi desktop. <https://docs.dremio.com/cloud/sonar/client-apps/microsoft-power-bi/working-from-datasets/connecting-to-dremio-cloud-from-pbi-desktop/>, 2024. 64
- [34] Dremio. Getting started with docker. <https://docs.dremio.com/current/get-started/docker-quickstart/>, n.d.. 63
- [35] Dremio. Dremio documentation: What is dremio? <https://docs.dremio.com/24.3.x/>, n.d.. 61
- [36] Dheeru Dua. apyori: A simple implementation of apriori algorithm, 2017. URL <https://pypi.org/project/apyori/>. 64
-

REFERENCIAS

- [37] EBAC. ¿qué es figma?, 2023. URL <https://ebac.mx/blog/que-es-figma>. 48
- [38] EloKing. ¿qué significa moba?, n.d. URL <https://es.eloking.com/glossary/general/moba>. Recuperado el 14 de agosto de 2024. 22
- [39] Carlos Espino Timón, Xavier Martínez Fontes, and Atanasi Daradoumis Haralabus. Análisis predictivo: técnicas y modelos utilizados y aplicaciones del mismo - herramientas *Open Source* que permiten su uso. Trabajo de Fin de Grado, Grado en Ingeniería Informática, Business Intelligence, enero 2017. 15, 16
- [40] Marcos Antonio Espinoza Mina. Crisp-dm: Conocimiento y comunicación de una metodología para minería de datos. In *Retos y perspectivas de las tecnologías de información*. Universidad Tecnológica ECOTEC, Universidad Agraria del Ecuador, n.d. Capítulo III. 45
- [41] Fandom. Campeones, n.d.. URL <https://leagueoflegendsoficial.fandom.com/es/wiki/Campeones>. Accedido: 14 de agosto de 2024. 24
- [42] Fandom. Runas (histórico) - wiki league of legends, n.d.. URL [https://leagueoflegends.fandom.com/es/wiki/Runas_\(Historico\)](https://leagueoflegends.fandom.com/es/wiki/Runas_(Historico)). Última consulta: 21 de agosto, 2024. 26
- [43] Esteban Canle Fernández. ¿qué es django?, 2002. URL <https://www.tokioschool.com/noticias/que-es-django/>. 50
- [44] Esteban Canle Fernández. Python request: Qué es y cómo funciona. <https://www.tokioschool.com/noticias/python-request/>, 2022. 54
- [45] Andy Fernández Castañete. Implementación de un algoritmo apriori-like para el minado de reglas de asociación en las declaraciones aduanales de mercancías en cuba. Trabajo de diploma, Universidad de las Ciencias Informáticas, La Habana, Cuba, Junio 2012. Facultad 3. 20
- [46] Luis Carlos Molina Félix and Ramon Sangüesa i Solé. *Reglas de asociación*. Universitat Oberta de Catalunya, n.d. URL <http://www.uoc.edu>. 18, 19
- [47] Barcelona Geeks. MI | algoritmo eclat, 2022. URL <https://barcelonageeks.com/ml-algoritmo-eclat/>. Accessed: 2024-08-14. 21
- [48] GoDaddy. Sistema operativo: Qué es, fundamentos y características, 2024. URL <https://www.godaddy.com/resources/es/digitalizacion/sistema-operativo-que-es>. Consultado el 22 de julio de 2024. 8
- [49] Alberto González Torralbo. Diseño de una herramienta de análisis de rendimiento de jugadores para equipos de alto nivel de esports. Trabajo fin de grado, Escuela Técnica Superior de Ingenieros Informáticos, Madrid, 7 2019. Director: Paula Fernández Arias, PhD. 74, 77

-
- [50] Prometheus GS. Análisis de datos: Descriptivo, predictivo y prescriptivo, 2019. URL <https://prometeusgs.com/analisis-de-datos-diferencias/>. 17
- [51] Matias Gutiérrez. Técnicas de minería de datos, 2013. URL <https://mineriadatos1.blogspot.com/2013/06/tecnicas-mineria-de-datos.html>. 17
- [52] Héctor Herreros Orcajo. Sistema de recomendación de league of legends mediante agrupación y redes de neuronas. Trabajo fin de grado, Universidad Carlos III de Madrid, Leganés, España, septiembre 2020. 71, 77
- [53] Luis Amador Hidalgo. *Inteligencia Artificial y Sistemas Expertos*. Universidad de Córdoba, Facultad de Ciencias Económicas y Empresariales (ETEA), Córdoba, n.d. 9, 10, 11
- [54] OK Hosting. ¿cuáles son los tipos de software y su clasificación?, 2024. URL <https://okhosting.com/blog/tipos-de-software-su-clasificacion/>. Consultado el 22 de julio de 2024. 8
- [55] Hosting Plus. Metodología de cascada: Fases y desarrollo, 2021. URL <https://www.hostingplus.mx/blog/metodologia-de-cascada-fases-y-desarrollo/>. 36
- [56] HubSpot. ¿qué es pandas en python y cómo utilizarlo? <https://blog.hubspot.es/website/que-es-pandas-python>, 2023. 56
- [57] IBM. Clustering models, 2021. URL <https://www.ibm.com/docs/es/spss-modeler/saas?topic=nodes-clustering-models>. Último acceso: 31 de julio de 2024. 17
- [58] Inesdi Digital Business School. Ejemplos cotidianos de ia: Cómo la inteligencia artificial está presente en nuestro día a día, 2023. URL <https://www.inesdi.com/blog/ejemplos-cotidianos-de-ia/>. Consultado el 11 de septiembre de 2024. 14
- [59] Acceso a la Información y Protección de Datos Personales Instituto Nacional de Transparencia. Aspectos técnicos – marco internacional de competencias de protección de datos personales para estudiantes, 2021. URL https://micrositios.inai.org.mx/marcocompetencias/?page_id=372. Consultado el 22 de julio de 2024. 7, 8
- [60] IONOS. Modelo en espiral: Desarrollo de software en etapas iterativas, 2023. URL <https://www.ionos.mx/startupguide/productividad/modelo-en-espiral/>. 37, 38
- [61] IT Sitio. Conozca de qué se trata moba, n.d. URL <https://www.itsitio.com/gaming/conozca-de-que-se-trata-moba/>. 22

- [62] J2Logo. Python requests: Peticiones http en python. <https://j2logo.com/python/python-requests-peticiones-http/>, n.d. 55
- [63] Kinsta. ¿qué es postgresql? una introducción completa, 2023. URL <https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/>. 52
- [64] League of Legends Wiki. Súbditos - wiki de league of legends, 2014. URL <https://leagueoflegends.fandom.com/es/wiki/subditos>. 27
- [65] League of Legends Wiki. League of Legends, n.d. URL https://leagueoflegendsoficial.fandom.com/es/wiki/League_of_Legends. 22
- [66] Diana Sofía Lora Ariza. Data mining techniques in video games. Master's thesis, Universidad Complutense de Madrid, Madrid, 9 2015. 72, 77
- [67] Danny Lévano Rodríguez. *Metodologías de desarrollo de software: Guía completa para la adopción de buenas prácticas de desarrollo de software*. No hay publisher, n.d. Mg. Danny Lévano Rodríguez. 39, 40
- [68] Esteban Gabriel Maida and Julián Pacienza. Metodologías de desarrollo de software. Tesis de licenciatura, Pontificia Universidad Católica Argentina Santa María de los Buenos Aires, Facultad de Química e Ingeniería "Fray Rogelio Bacon", diciembre 2015. Cátedra Seminario de Sistemas. 35
- [69] Beatriz Beltrán Martínez. Minería de datos, n.d. MC, fecha no disponible. 16, 17
- [70] Alberto Mateos Rama. En busca de patrones de comportamiento entre jugadores de league of legends. Trabajo fin de grado, Universidad Autónoma de Madrid, Madrid, España, julio 2019. 69, 77
- [71] MEDAC. Historia de League of Legends: Evolución del juego, n.d. URL <https://medac.es/blogs/masteres-online/historia-league-of-legends>. 22
- [72] Microsoft. Conversión entre dataframes de pyspark y pandas: Azure databricks. <https://learn.microsoft.com/es-es/azure/databricks/pandas/pyspark-pandas-conversion>, 2024. 57
- [73] Microsoft. ¿qué es power bi? <https://learn.microsoft.com/es-es/power-bi/fundamentals/power-bi-overview>, 2024. 63
- [74] Microsoft Corporation. Conceptos de minería de datos, 2023. URL <https://learn.microsoft.com/es-es/analysis-services/data-mining/data-mining-concepts?view=asallproducts-allversions>. Último acceso: 30 de julio de 2024. 15
- [75] Ariel Monteserin. Reglas de asociación. Presentación, n.d. Inteligencia Artificial. 18

-
- [76] EBAC México. Qué es react y para qué sirve: ventajas y desventajas, casos de uso, características, quién lo utiliza, 2023. URL <https://ebac.mx/blog/que-es-react>. Accessed: 2024-10-13. 49
- [77] Andrés Navarro Cadavid, Juan Daniel Fernández Martínez, and Jonathan Morales Vélez. Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA*, 11(2):30–39, 2013. ISSN 1692-8261. URL <http://www.redalyc.org/articulo.oa?id=496250736004>. 39
- [78] Héctor Oscar Nigro, Daniel Xodo, Gabriel Corti, and Damián Terren. Kdd (knowledge discovery in databases): Un proceso centrado en el usuario. *INCA/INTIA - Departamento de Computación y Sistemas, Facultad de Ciencias Exactas - UNICEN*, n.d. 42
- [79] League of Legends Wiki. Jungla - wiki de league of legends, n.d.. URL <https://leagueoflegends.fandom.com/es/wiki/Jungla>. 28
- [80] League of Legends Wiki. Oro - wiki de league of legends, n.d.. URL <https://leagueoflegends.fandom.com/es/wiki/Oro>. 27
- [81] League of Legends Wiki. Objetos - wiki de league of legends, n.d.. URL <https://leagueoflegendsoficial.fandom.com/es/wiki/Objetos/LoL>. 28
- [82] Oracle. ¿qué es un almacén de datos? <https://www.oracle.com/mx/database/what-is-a-data-warehouse/>, n.d. 61
- [83] Cristina Ortega. Estadística descriptiva: Qué es, objetivo, tipos y ejemplos, 2024. URL <https://www.questionpro.com/blog/es/estadistica-descriptiva/>. QuestionPro, February 16. 17
- [84] Wendy Julieth Molina Palacios. Modelo en espiral, n.d. Documento académico sobre metodologías de desarrollo de software. 37, 38
- [85] Joel Pinho Lucas. *Métodos de clasificación basados en asociación aplicados a sistemas de recomendación*. Tesis doctoral, Universidad de Salamanca, Salamanca, España, Octubre 2010. Directora: Dra. D^a. María N. Moreno García. 18, 19, 20
- [86] PyPro. Modelos de agrupamiento, n.d. URL <https://www.pypro.mx/app/curso/machine-learning-con-python/modelos-de-agrupamiento>. Último acceso: 31 de julio de 2024. 17
- [87] Renato Pérez-Gómez. Generación de reglas de asociación para productos de retail utilizando el algoritmo FP-Growth paralelo. *Universidad de Lima*, 2019. renato08lasalle13@gmail.com. 20
- [88] Tomás León Quintanar. Sistemas expertos y sus aplicaciones. Monografía para obtener el título de Licenciado en Computación, Instituto de Ciencias Básicas e Ingeniería, Pachuca de Soto, HGO., 2007. Asesor: Lic. en Comp. Luis Islas Hernández. 8, 9, 10, 11

REFERENCIAS

- [89] Riot Games. League of legends - descarga del juego, 2021. URL <https://www.leagueoflegends.com/es-es/download/>. 29
- [90] Riot Games. Cómo jugar, n.d. URL <https://www.leagueoflegends.com/es-es/how-to-play/>. 21, 22
- [91] Enrique Rodríguez. Sistemas expertos en acción: Ejemplos basados en inteligencia artificial. <https://canalinnova.com/sistemas-expertos-en-accion-ejemplos-basados-en-inteligencia-artificial/>, n.d. Consultado el 25 de julio de 2024. 14
- [92] SAS Institute Inc. Minería de datos, 2023. URL https://www.sas.com/es_mx/insights/analytics/data-mining.html. 15
- [93] Layla Scheli. Apache arrow: Leer dataframe con memoria cero. *Medium*, 2021. 57
- [94] Tokio School. ¿qué es scikit-learn?, 2023. URL <https://www.tokioschool.com/noticias/que-es-scikit-learn/>. 65
- [95] Master Data Scientist. Scikit-learn for data science, n.d. URL <https://www.master-data-scientist.com/scikit-learn-data-science/>. 64
- [96] Stefano Serafinelli. Qué es la escala SUS y cómo usarla para medir la usabilidad, March 2024. URL <https://www.teacuplab.com/es/blog/que-es-la-escala-sus-y-como-usarla-para-medir-la-usabilidad/>. 115
- [97] Amazon Web Services. Contenedores de docker | ¿qué es docker?, n.d.. URL <https://aws.amazon.com/es/docker/>. Accedido: 22 de octubre de 2024. 62
- [98] Amazon Web Services. ¿qué es django? - explicación del software django, n.d.. URL <https://aws.amazon.com/es/what-is/django/>. 50
- [99] Pure Storage. ¿qué es postgresql?, n.d. URL <https://www.purestorage.com/es/knowledge/what-is-postgre-sql.html>. 52
- [100] S. R. Timarán-Pereira, I. Hernández-Ateaga, S. J. Caicedo-Zambrano, A. Hidalgo-Troya, and J. C. Alvarado-Pérez. El proceso de descubrimiento de conocimiento en bases de datos. In *Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional*, pages 68–86. Ediciones Universidad Cooperativa de Colombia, Bogotá, 2016. doi: 10.16925/9789587600490. URL <http://dx.doi.org/10.16925/9789587600490>. 42, 43, 44
- [101] Toolify. Descubre las ventajas del data lakehouse con dremio. <https://www.toolify.ai/es/ai-news-es/descubre-las-ventajas-del-data-lakehouse-con-dremio-2163821>, n.d. 62

- [102] Miguel Torres Gómez. esports, un enfoque basado en inteligencia artificial. Trabajo fin de grado, Universidad de Málaga, Málaga, España, junio 2021. 67, 77
- [103] Universidad Nacional Autónoma de México. Sistemas expertos, n.d. URL https://virtual.cuautitlan.unam.mx/intar/?page_id=200. 8
- [104] Melissa Álvarez Martell, Mariana Marcelino-Aranda, Alberto Macías Alcívar, and José Carlos Novoa Sandoval. Metodología tradicional vs ágil para la gestión de proyectos de software. *Boletín No. 83*, marzo 2021. 35