



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Grado en Ingeniería en Tecnologías de las
telecomunicaciones

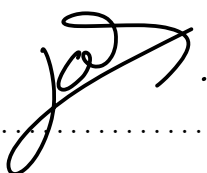
Plataforma para el aprendizaje de lenguajes de programación apoyada en técnicas de Machine Learning

Autor
Jaime Lloret Roqueta

Dirigido por
Atilano Fernández-Pacheco Sánchez-Migallón

Madrid
Julio 2024

Jaime Lloret Roqueta, declara bajo su responsabilidad, que el Proyecto con título **Plataforma de aprendizaje de lenguajes de programación apoyada en técnicas de Machine Learning** presentado en la ETS de Ingeniería (ICAI) de la Universidad Pontificia Comillas en el curso académico 2020/21 es de su autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.:
 Fecha: 4 / 07 / 2024

Autoriza la entrega:

EL DIRECTOR DEL PROYECTO

Atilano Fernández-Pacheco Sánchez-Migallón

71216314B ATILANO
RAMIRO
FERNÁNDEZ-
PACHECO

Digitally signed by
71216314B ATILANO RAMIRO
FERNÁNDEZ-PACHECO
Date: 2024.07.04 18:57:43
+02'00'



Fdo.: Fecha: 4 / 07 / 2024

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Jaime Lloret Roqueta **DECLARA** ser el titular de los derechos de propiedad intelectual de la obra: Plataforma para el aprendizaje de lenguajes de programación apoyada en técnicas de Machine Learning, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, los derechos de digitalización, de archivo, de reproducción, de distribución y de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- (a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- (b) Reproducir la en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- (c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- (d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- (e) Asignar por defecto a estos trabajos una licencia Creative Commons.

- (f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- (a) Que la Universidad identifique claramente su nombre como autor de la misma
- (b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- (c) Solicitar la retirada de la obra del repositorio por causa justificada.
- (d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

- (a) El autor se compromete a:
- (b) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- (c) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- (d) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- (e) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 4 de julio de 2024

ACEPTA

Fdo.: 

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

Grado en Ingeniería en Tecnologías de las
telecomunicaciones

Plataforma de aprendizaje de lenguajes de programación apoyada en técnicas de Machine Learning

Autor
Jaime Lloret Roqueta

Dirigido por
Atilano Fernández-Pacheco Sánchez-Migallón

Madrid
Julio 2024

Resumen

El proyecto desarrollado consiste en una plataforma educativa para enseñar lenguajes de programación de manera personalizada para todas las edades y niveles, de forma práctica y efectiva, llamada CodeLingo. Esta plataforma utiliza técnicas de machine learning e inteligencia artificial para personalizar las recomendaciones de cursos y mejorar la interacción con el usuario a través de su asistente virtual, Coddy. Los usuarios cuentan con una plataforma donde pueden practicar lecciones, participar en desafíos, leer documentación, ver tutoriales en video y compartir sus aprendizajes con amigos. La aplicación ha sido construida usando Flutter y Dart, y se ha diseñado para que sea lo más accesible posible.

La creciente demanda de habilidades de programación ha impulsado el desarrollo de plataformas educativas [19]. El objetivo de CodeLingo es proporcionar una herramienta que no solo enseñe a programar, sino que también motive a los estudiantes a aprender a pensar de forma creativa, superando desafíos a través de una experiencia práctica y colaborativa desde el primer momento. El avance constante de la tecnología de la información y la creciente demanda de profesionales cualificados en lenguajes de programación han creado la necesidad de soluciones educativas innovadoras y accesibles.

Palabras clave: Programación, Plataforma educativa, Machine Learning, Firebase, Inteligencia Artificial, Flutter, Dart

1. Estado del Arte

El proceso de aprendizaje de programación puede resultar intimidante para muchas personas, especialmente para los principiantes. La educación en programación enfrenta desafíos como la falta de accesibilidad para un público diverso, la percepción de complejidad de los lenguajes de programación y la falta de motivación para mantener un aprendizaje constante. En este contexto, el éxito de plataformas como Duolingo ha demostrado el potencial de las técnicas de gamificación y aprendizaje personalizado para mejorar la educación en línea. Estas técnicas son fundamentales para crear una transformación educativa en la sociedad digital actual.

2. Definición del Proyecto

Inspirado en modelos de aplicaciones EdTech del mercado, el proyecto CodeLingo analiza tecnologías para mejorar la retención de usuarios mediante recomendaciones de cursos basadas en el progreso y las preferencias de cada uno. La plataforma cuenta con un asistente virtual, Coddy, que puede resolver dudas y ofrecer explicaciones sobre ejercicios, y dispone de un sistema de retroalimentación del progreso con recompensas asociadas. Además, la aplicación incluye un chat para fomentar la interacción entre usuarios y se realiza un estudio económico para asegurar la viabilidad y sostenibilidad del proyecto a largo plazo. El objetivo principal de CodeLingo es proporcionar una herramienta educativa que no solo enseñe a programar, sino que también motive a los estudiantes a aprender a pensar de manera creativa, superando desafíos a través de una experiencia práctica y colaborativa desde el principio.

3. Desarrollo del Modelo

La arquitectura del sistema CodeLingo está diseñada para proporcionar una experiencia de usuario fluida y eficiente, integrando diversos servicios backend para manejar autenticación, almacenamiento de datos y recomendaciones personalizadas. La estructura del sistema se divide en varios componentes principales, cada uno con un rol específico en el funcionamiento de la plataforma. En el esquema de la figura 5.1 se puede comprobar que la arquitectura se basa en un enfoque de microservicios, lo que permite una gestión más eficiente de los recursos y facilita el mantenimiento y la actualización del sistema.

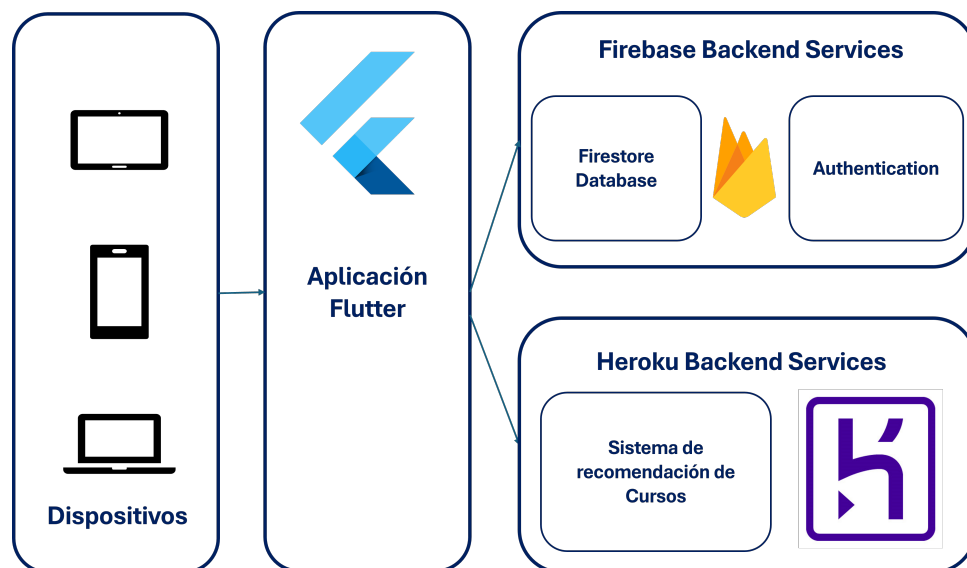


Figure 1: Arquitectura de CodeLingo

4. Desarrollo del Sistema

El desarrollo del frontend de CodeLingo ha sido planeado y ejecutado para ofrecer una experiencia de usuario intuitiva, atractiva y efectiva. Se ha prestado especial atención al diseño UI/UX, asegurando que cada componente de la interfaz no solo sea visualmente atractivo, sino también funcional y fácil de usar. Más adelante, en el capítulo 5 se detallan todas las pantallas de la aplicación. Una de las pantallas más importantes para el usuario es el lobby principal, el cual actúa como el centro de navegación de la aplicación, donde los usuarios pueden acceder a las diferentes funcionalidades y contenidos disponibles.

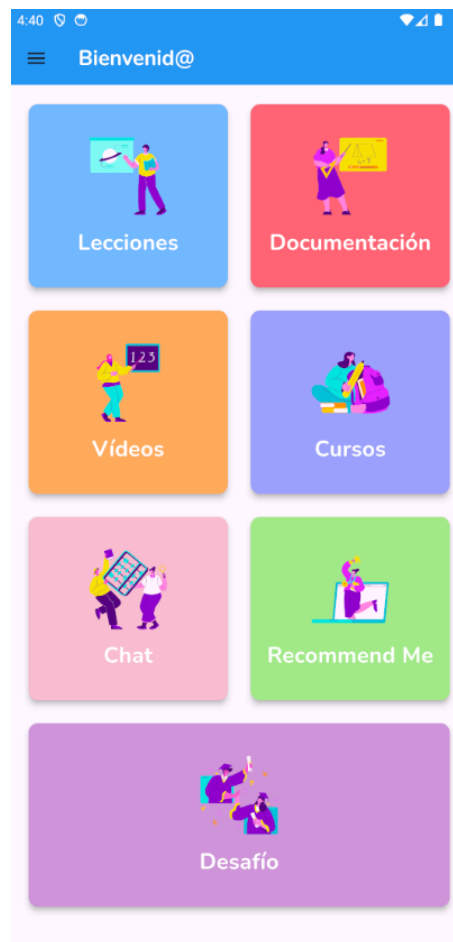


Figure 2: Pantalla de Lobby principal

Algunos ejemplos de otras pantallas de la aplicación que cuentan con funcionalidades variadas son las que se muestran en las figuras [5.16](#), [5.17](#), [5.18](#), correspondientes a la funcionalidad de Completar lecciones, en particular la [5.18](#) se muestra la funcionalidad de Cuddy y otras como [5.13](#) y [5.14](#) relativas al perfil del usuario.

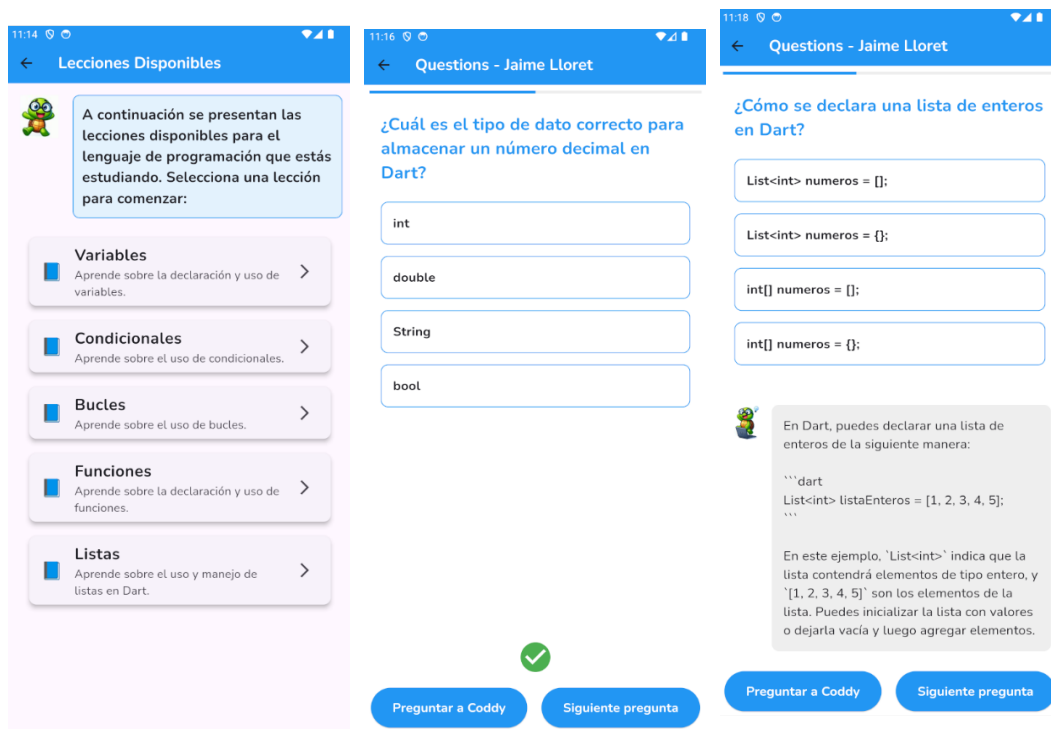


Figure 3: Pantalla de Lecciones

Figure 4: Pantalla de Ejercicios

Figure 5: Pantalla de Ejercicios con ayuda de Cuddy

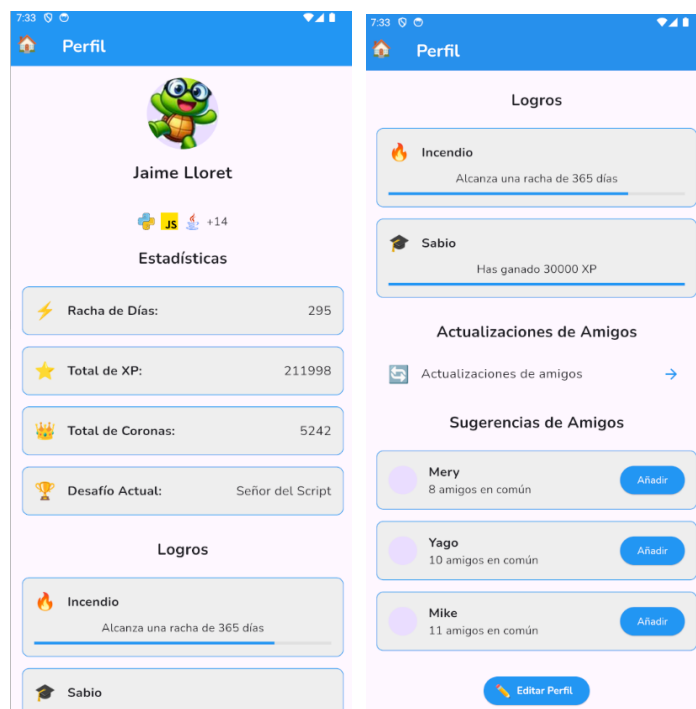


Figure 6: Pantalla 1 de Perfil del usuario

Figure 7: Pantalla 2 de Perfil del usuario

5. Sistema de Recomendación de Cursos

El sistema de recomendación de CodeLingo funciona mediante técnicas de machine learning, basándose en una serie de datos proporcionados por el usuario, como la experiencia previa, los objetivos de aprendizaje, los lenguajes de programación preferidos, los frameworks de interés, las áreas de interés, el formato de curso preferido y la duración deseada del curso. Estos datos se utilizan para generar un perfil detallado del usuario, como se muestra en la tabla 5.3

Preferencias de entrada
Experiencia previa
Objetivos de aprendizaje
Lenguajes de programación preferidos
Frameworks de interés
Áreas de interés
Formato de curso (e.g., video, lectura)
Duración del curso deseada

Table 1: Tabla de preferencias de entrada

Las descripciones de los cursos también se estructuran en un formato similar, como se muestra en la tabla 5.4

Campos de los cursos del dataset
Título del curso
URL del curso
Imagen del curso
Resumen del curso
Calificación y número de reseñas
Duración y cantidad de lecciones
Nivel de dificultad
Precio

Table 2: Tabla de campos de los cursos del dataset

Seguidamente, se muestra un ejemplo de interacción de un usuario con una serie de preferencias y el resultado de la recomendación de cursos:

```
{
  "experience": "Avanzado",
  "learning_objective": "Mejorar habilidades actuales",
  "languages": ["Python"],
  "frameworks": ["React"],
  "areas": ["Desarrollo web"],
  "course_format": "Proyectos basados en casos reales",
  "course_duration": "Más de 10 horas"
}
```

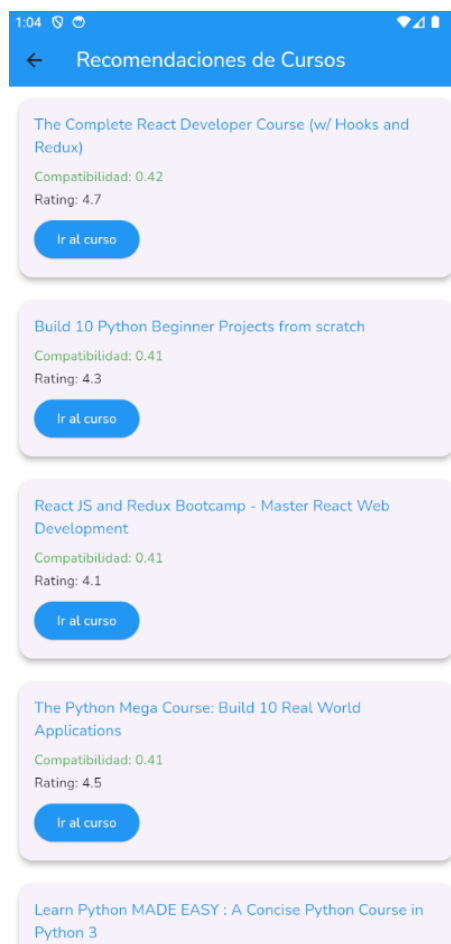


Figure 8: Ejemplo de recomendación de cursos en base a preferencias del usuario

Aplicamos la técnica TF-IDF (Term Frequency-Inverse Document Frequency) para convertir tanto las descripciones de los cursos como el perfil del usuario en representaciones vectoriales. La similitud entre el vector de preferencias del usuario y los vectores de los cursos se calcula utilizando la similitud de coseno. Esto nos permite identificar los cursos que son más similares a lo que el usuario busca. La similitud del coseno se calcula utilizando la siguiente fórmula:

$$\text{Similitud del Coseno} = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

6. Casos de Uso

En la figura [5.2](#) se muestra el diagrama de casos de uso de la aplicación CodeLingo. Este diagrama ilustra las diversas interacciones que los usuarios pueden tener con la plataforma, incluyendo la práctica de lecciones, la participación en desafíos, la lectura de documentación, la visualización de tutoriales en video y la interacción social con otros usuarios.

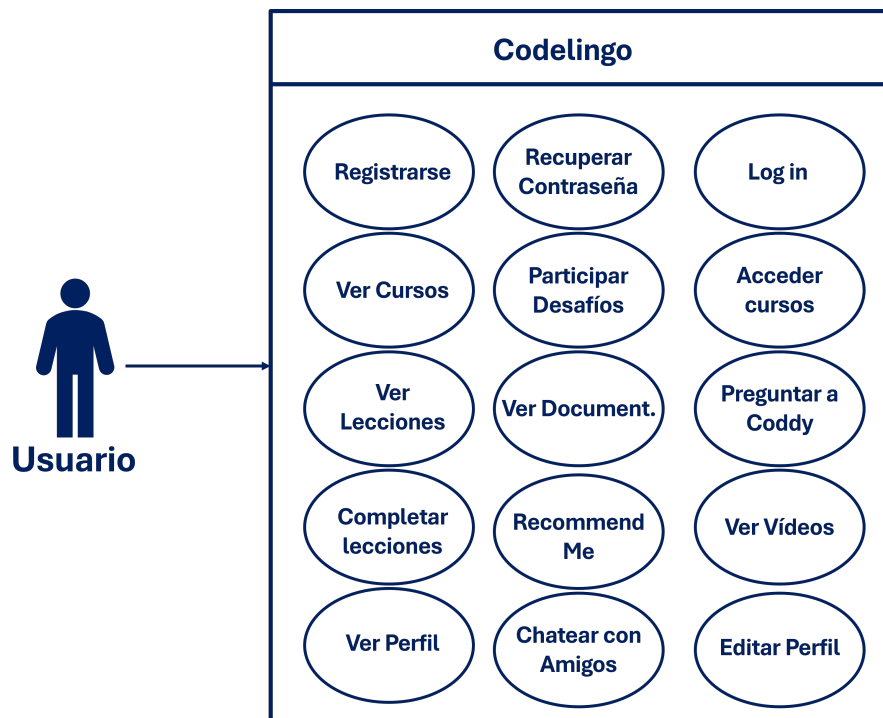


Figure 9: Diagrama de Casos de Uso de CodeLingo

7. Estudio Económico

Para asegurar la viabilidad y sostenibilidad del proyecto a largo plazo, se ha realizado un estudio económico detallado.

Atendiendo al modelo de las 3 C's, el modelo de negocio de Codelingo se fundamenta en tres pilares esenciales: creación, captura y configuración de valor. Estos pilares son representados de manera en el diagrama presentado en la Figura [6.1](#).

La propuesta de valor se dirige a un público objetivo específico, que incluye estudiantes y profesionales, entusiastas de la programación, instituciones educativas y empresas tecnológicas. Este enfoque segmentado permite que la plataforma se ajuste a las necesidades particulares de cada grupo, aumentando así su efectividad y atractivo.

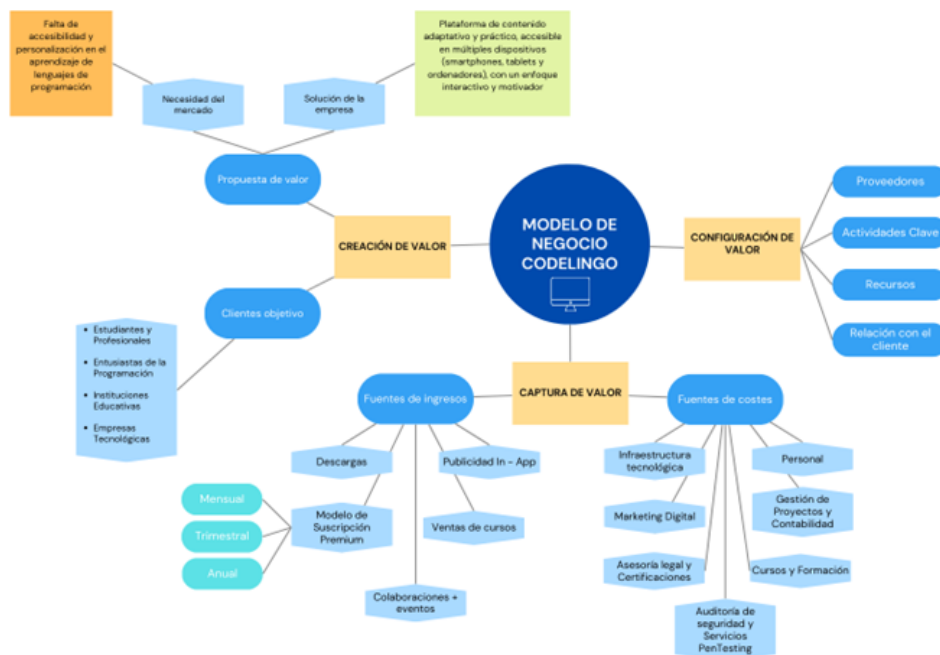


Figure 10: Modelo de Negocio de CodeLingo

Se ha realizado un business case del proyecto de 5 años de duración, para ello, se ha elaborado una hoja de parámetros de ingresos y de costes, con la que posteriormente se han calculado los principales flujos de caja entrantes y salientes de la aplicación en la forma de una cuenta de resultados. Este estudio se encuentra explicado en mayor profundidad en el capítulo 6 del trabajo. En paralelo, se han comparado las composiciones de las principales fuentes de ingresos y costes del proyecto en los años 1 y 5 para analizar como cambian los pesos de los distintos factores a lo largo de los años, tal y como se muestra en las figuras [6.4](#) y [6.6](#).

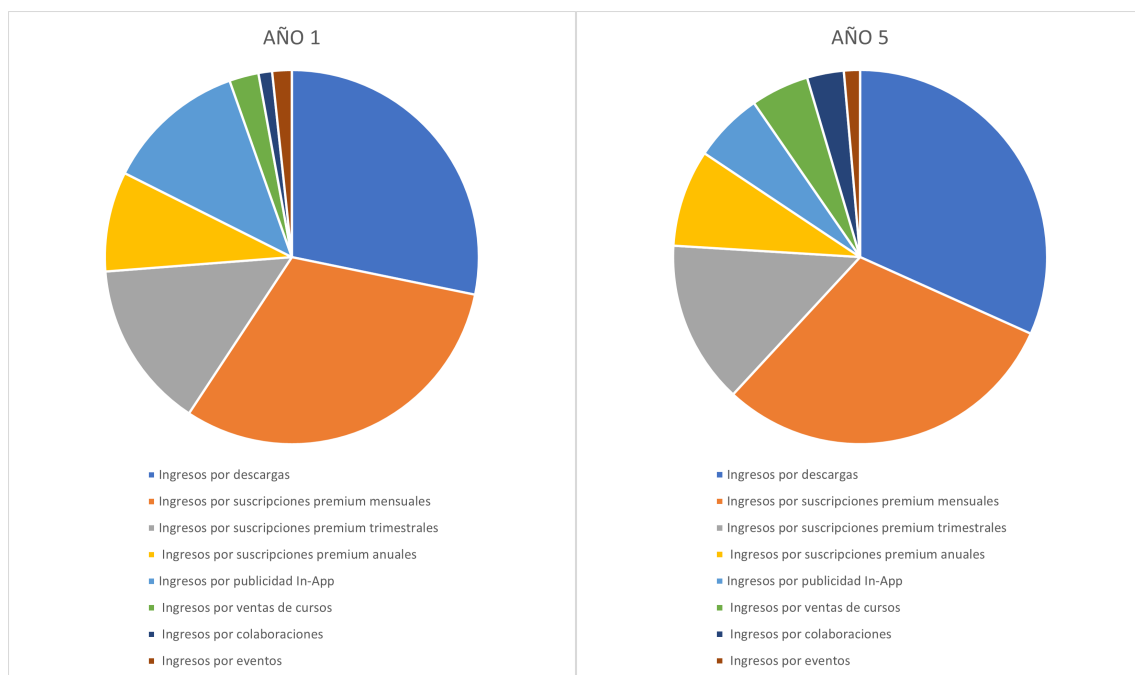


Figure 11: Comparativa de gráficos circulares de las fuentes de los ingresos del año 1 y 5

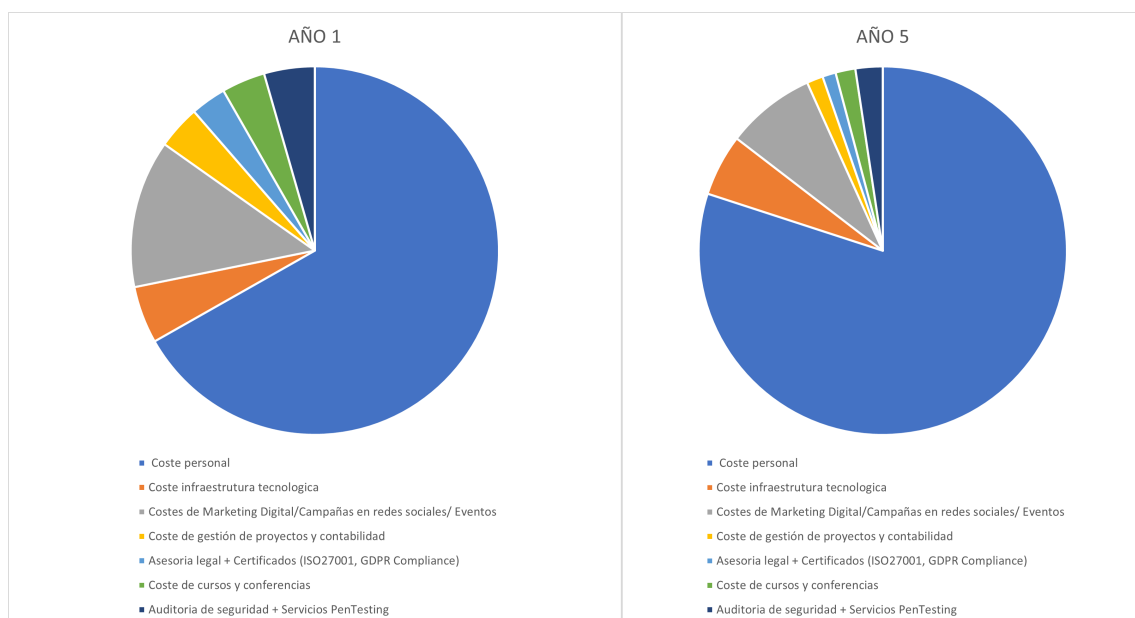


Figure 12: Comparativa de gráficos circulares de las fuentes de los costos del año 1 y 5

Finalmente, la figura 26 muestra La evolución de beneficios después de impuestos, una métrica muy atractiva para los inversores del proyecto. Se trata de una evolución muy positiva a lo largo de los años resultando en un crecimiento del 278,11 % del quinto año respecto al primero. Esto refleja que, a lo largo del transcurso del proyecto, el impacto de las principales fuentes de ingresos es mayor frente al impacto de los agentes de costes más significativos. Asimismo, otro indicador interesante para los inversores es el del VAN. Los flujos estimados para el proyecto dan lugar a un VAN de 398.556,44€, haciendo de Codelingo un proyecto viable y rentable en el largo plazo.

	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Beneficio antes de impuestos (euros)	55.446	72.480	101.470	131.927	209.650
Impuestos	13.862	18.120	25.367	32.982	52.413
Beneficio despues de impuestos (euros)	41.585	54.360	76.102	98.945	157.238

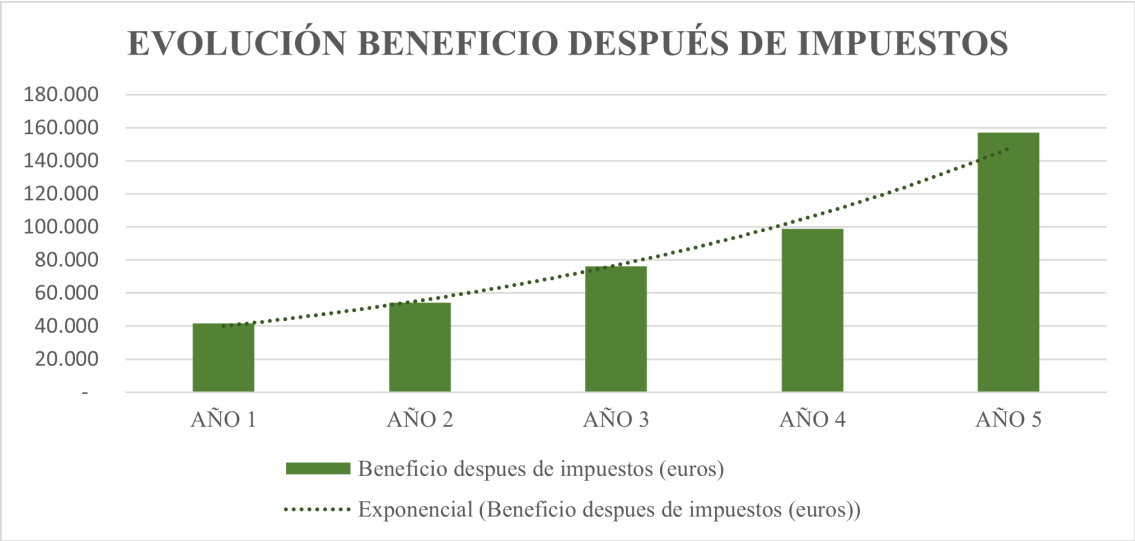


Figure 13: Evolución de los beneficios después de impuestos en 5 años

8. Conclusiones

El análisis del proyecto Codelingo revela aspectos importantes sobre la metodología y las tecnologías utilizadas.

La metodología aplicada mostró tanto aciertos como áreas de mejora. Un estudio inicial más exhaustivo de las tecnologías podría haber optimizado tiempos y mejorado la calidad del producto final. La organización del trabajo en sprints y el enfoque secuencial del desarrollo, comenzando por el frontend y terminando con el sistema de recomendación, fueron eficientes. La encapsulación de funcionalidades, probándolas por separado antes de integrarlas, aseguró que cada componente funcionara correctamente sin afectar la lógica general del proyecto.

El uso de Flutter, Dart, Firebase y Heroku fue crucial para alcanzar los objetivos. Aunque no se completó el despliegue de la aplicación, la configuración de entornos de producción y la integración de herramientas de análisis como Firebase Analytics habrían proporcionado una mejor visión del uso y rendimiento de la aplicación, mejorando la experiencia del usuario.

El modelo GPT-3.5-turbo, seleccionado por su rendimiento en procesamiento de lenguaje natural, destacó en varias áreas como atención al cliente, generación de contenido y análisis de datos. Además, el sistema de recomendación de cursos basado en el cálculo del coseno de similitud mostró resultados prometedores, ofreciendo una personalización efectiva de contenidos.

Para futuros desarrollos, se recomienda investigar y adoptar tecnologías de producción más robustas, implementar herramientas analíticas para obtener feedback detallado del usuario, mejorar la personalización de lecciones utilizando machine learning, introducir elementos gamificados como ligas y competencias entre usuarios, y organizar actividades temáticas y colaborativas para fomentar la participación. También es vital optimizar la gestión de datos y la respuesta de las APIs para asegurar la eficiencia de la aplicación bajo cargas elevadas.

Estas mejoras no solo aumentarán la funcionalidad y atractivo de Codelingo, sino que también asegurarán que la plataforma pueda escalar y adaptarse a las demandas de un entorno de producción profesional, ofreciendo una experiencia de aprendizaje más rica y eficaz.

Abstract

The developed project consists of an educational platform called CodeLingo designed to teach programming languages in a personalized, practical, and effective manner for all ages and levels. This platform uses machine learning and artificial intelligence techniques to personalize course recommendations and enhance user interaction through its virtual assistant, Coddy. Users have access to a platform where they can practice lessons, participate in challenges, read documentation, watch video tutorials, and share their learnings with friends. The application is built using Flutter and Dart, and it is designed to be as accessible as possible.

The growing demand for programming skills has driven the development of educational platforms [19]. The goal of CodeLingo is to provide a tool that not only teaches programming but also motivates students to learn to think creatively, overcoming challenges through a practical and collaborative experience from the start. The constant advancement of information technology and the increasing demand for qualified professionals in programming languages have created the need for innovative and accessible educational solutions.

Keywords: Programming, Educational Platform, Machine Learning, Firebase, Artificial Intelligence, Flutter, Dart

1. State of the Art

The process of learning programming can be intimidating for many people, especially beginners. Programming education faces challenges such as lack of accessibility for a diverse audience, the perception of complexity in programming languages, and the lack of motivation to maintain constant learning. In this context, the success of platforms like Duolingo has demonstrated the potential of gamification techniques and personalized learning to improve online education. These techniques are essential for creating an educational transformation in today's digital society.

2. Project Definition

Inspired by market EdTech application models, the CodeLingo project analyzes technologies to improve user retention through course recommendations based on each user's progress and preferences. The platform features a virtual assistant, Coddy, that can answer questions and offer explanations about exercises, and it includes a feedback system on progress with associated rewards. Additionally, the application includes a chat to encourage user interaction, and an economic study is conducted to ensure the long-term viability and sustainability of the project. The main objective of CodeLingo is to provide an educational tool that not only teaches programming but also motivates students to learn to think creatively, overcoming challenges through a practical and collaborative experience from the start.

3. Model Development

The architecture of the CodeLingo system is designed to provide a smooth and efficient user experience, integrating various backend services to handle authentication, data storage, and personalized recommendations. The system structure is divided into several main components, each with a specific role in the platform's functionality. As shown in the diagram in Figure 5.1 the architecture is based on a microservices approach, allowing for more efficient resource management and easier system maintenance and updates.

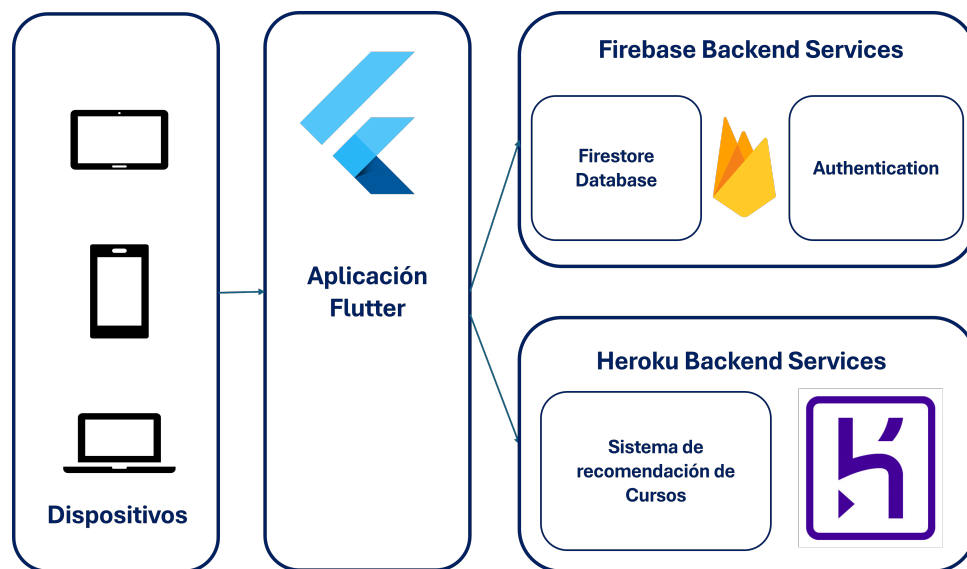


Figure 14: CodeLingo Architecture

4. System Development

The development of CodeLingo's frontend has been planned and executed to offer an intuitive, attractive, and effective user experience. Special attention has been paid to UI/UX design, ensuring that each interface component is not only visually appealing but also functional and easy to use. In Chapter 5, all application screens are detailed. One of the most important screens for the user is the main lobby, which acts as the navigation center of the application, where users can access the different available functionalities and content.

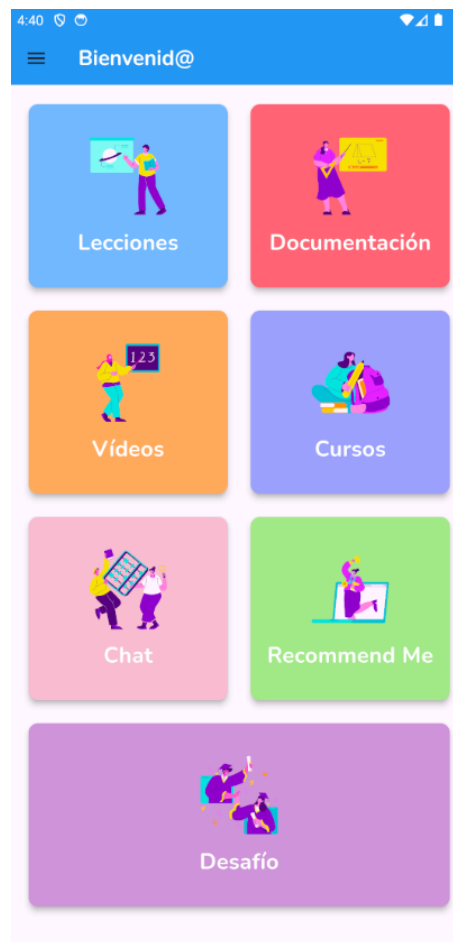


Figure 15: Main Lobby Screen

Examples of other screens with various functionalities are shown in Figures 5.16, 5.17, and 5.18, corresponding to the Complete Lessons functionality. In particular, Figure 5.18 shows Cuddy's functionality, and others like Figures 5.13 and 5.14 relate to the user profile.

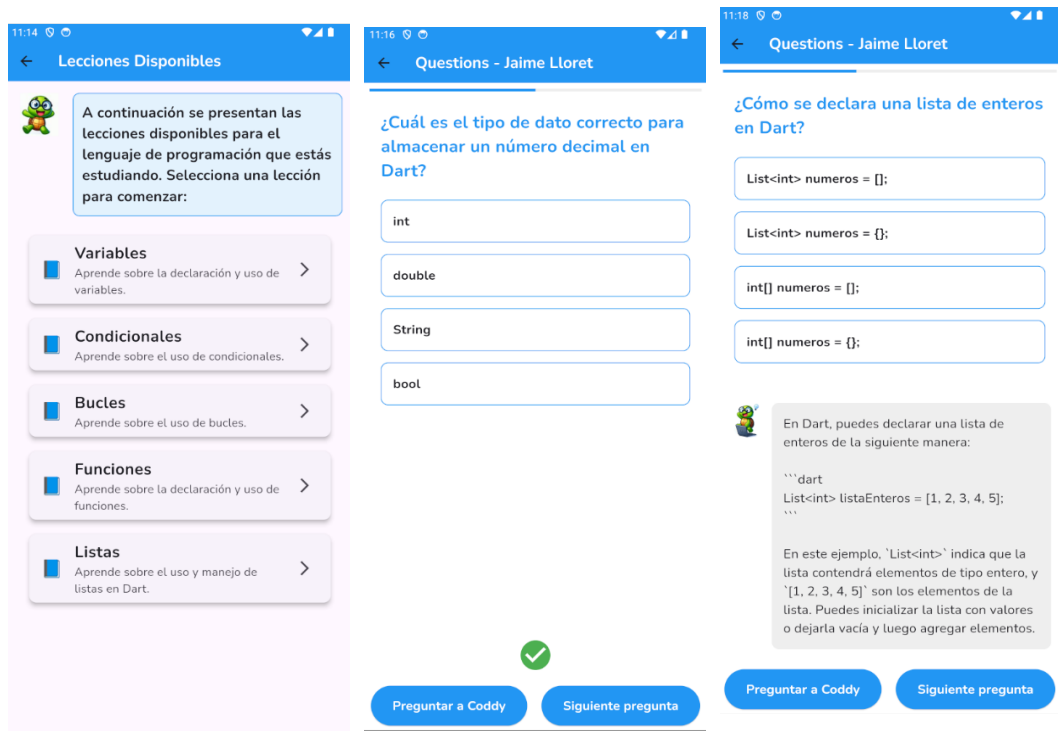


Figure 16: Lessons Screen

Figure 17: Exercises Screen

Figure 18: Exercises Screen with Cuddy's Help

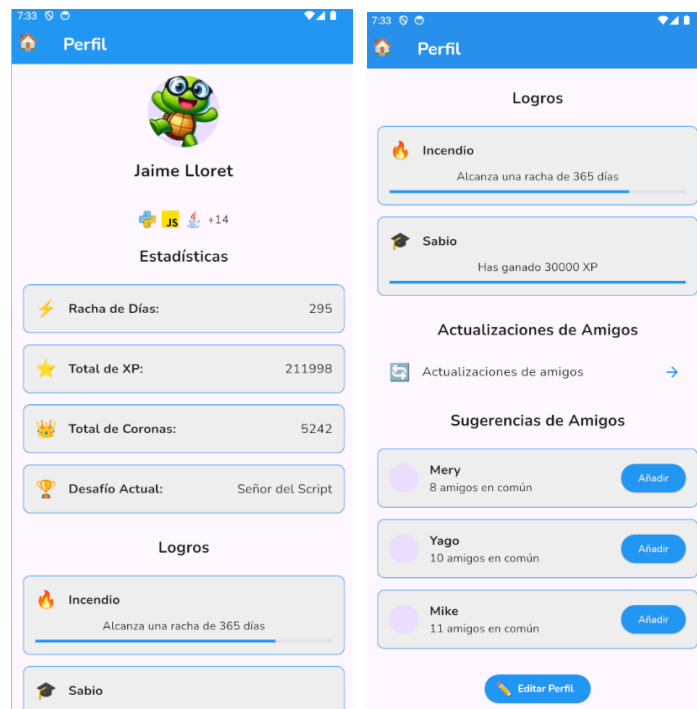


Figure 19: User Profile Screen 1

Figure 20: User Profile Screen 2

5. Course Recommendation System

The CodeLingo recommendation system works using machine learning techniques, based on a series of user-provided data such as prior experience, learning objectives, preferred programming languages, frameworks of interest, areas of interest, preferred course format, and desired course duration. This data is used to generate a detailed user profile, as shown in Table 5.3

Input Preferences
Previous experience
Learning objectives
Preferred programming languages
Frameworks of interest
Areas of interest
Course format (e.g., video, reading)
Desired course duration

Table 3: Input Preferences Table

Course descriptions are also structured in a similar format, as shown in Table 5.4

Course Dataset Fields
Course title
Course URL
Course image
Course summary
Rating and number of reviews
Duration and number of lessons
Difficulty level
Price

Table 4: Course Dataset Fields Table

Next, an example of user interaction with a set of preferences and the resulting course recommendations is shown:

```
{
  "experience": "Advanced",
  "learning_objective": "Improve current skills",
  "languages": ["Python"],
  "frameworks": ["React"],
  "areas": ["Web Development"],
  "course_format": "Project-based on real cases",
  "course_duration": "More than 10 hours"
}
```

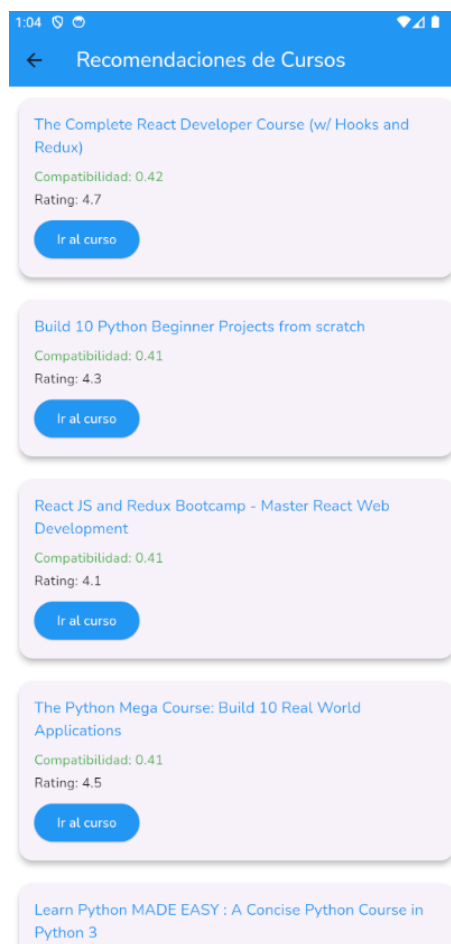


Figure 21: Example of Course Recommendation Based on User Preferences

We apply the TF-IDF (Term Frequency-Inverse Document Frequency) technique to convert both course descriptions and the user profile into vector representations. The similarity between the user's preference vector and the course vectors is calculated using cosine similarity. This allows us to identify the courses that are most similar to what the user is looking for. Cosine similarity is calculated using the following formula:

$$\text{Cosine Similarity} = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

6. Use Cases

Figure 5.2 shows the use case diagram of the CodeLingo application. This diagram illustrates the various interactions users can have with the platform, including practicing lessons, participating in challenges, reading documentation, watching video tutorials, and social interaction with other users.

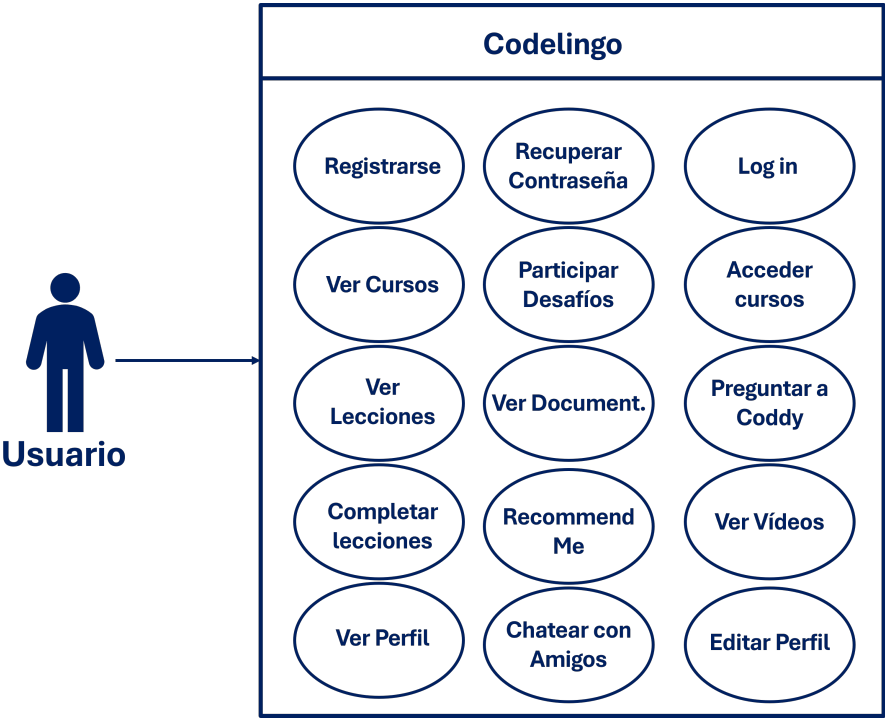


Figure 22: CodeLingo Use Case Diagram

7. Economic Study

To ensure the project's long-term viability and sustainability, a detailed economic study has been conducted.

Following the 3 C's model, CodeLingo's business model is based on three essential pillars: creation, capture, and value configuration. These pillars are represented in the diagram presented in Figure 6.1.

The value proposition targets a specific audience, including students and professionals, programming enthusiasts, educational institutions, and tech companies. This segmented approach allows the platform to meet the particular needs of each group, thereby increasing its effectiveness and appeal.

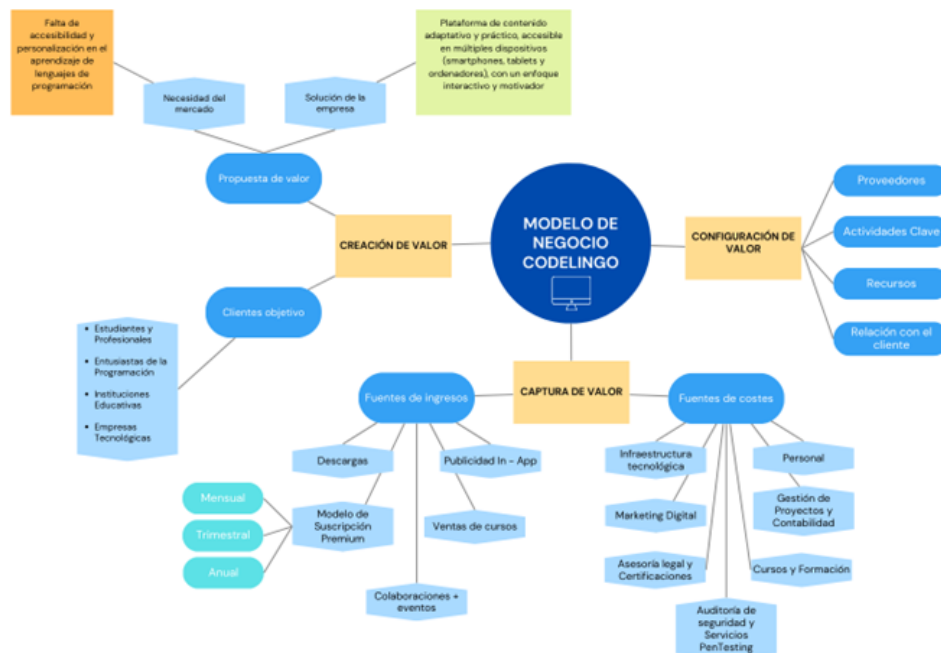


Figure 23: CodeLingo Business Model

A business case for the 5-year project duration has been prepared. For this, a sheet of income and cost parameters has been drawn up, with which the main cash inflows and outflows of the application have been subsequently calculated in the form of an income statement. This study is explained in greater depth in Chapter 6 of the work. In parallel, the compositions of the project's main sources of income and costs in years 1 and 5 have been compared to analyze how the weights of the different factors change over the years, as shown in Figures 6.4 and 6.6.

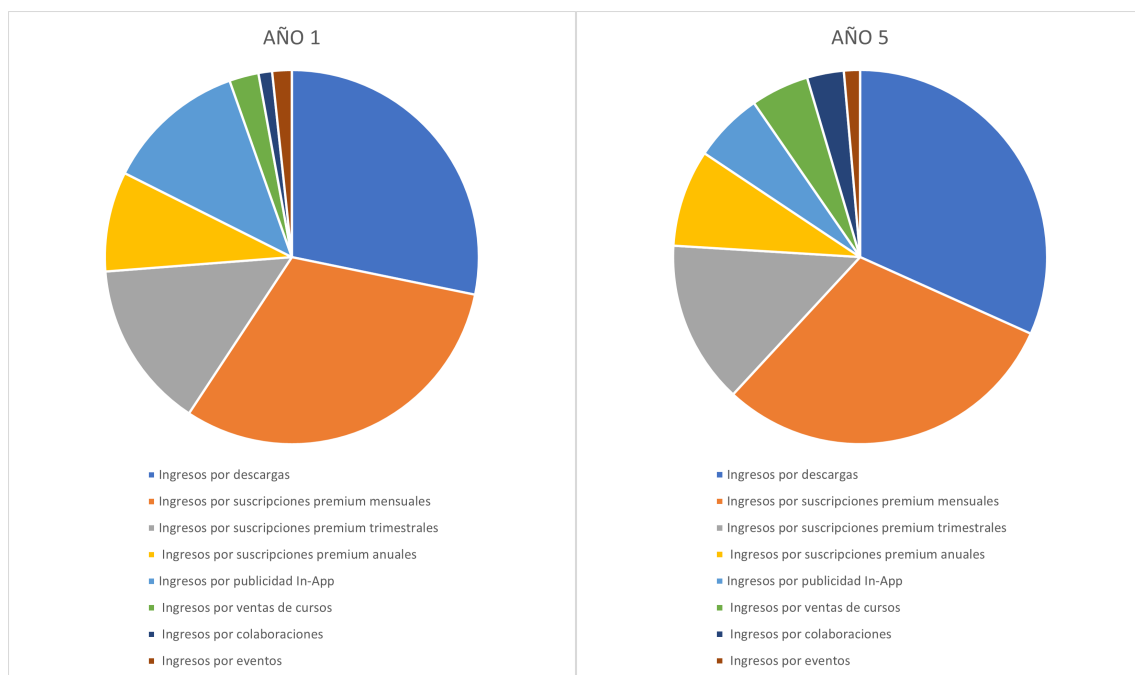


Figure 24: Comparison of Pie Charts of Income Sources for Years 1 and 5

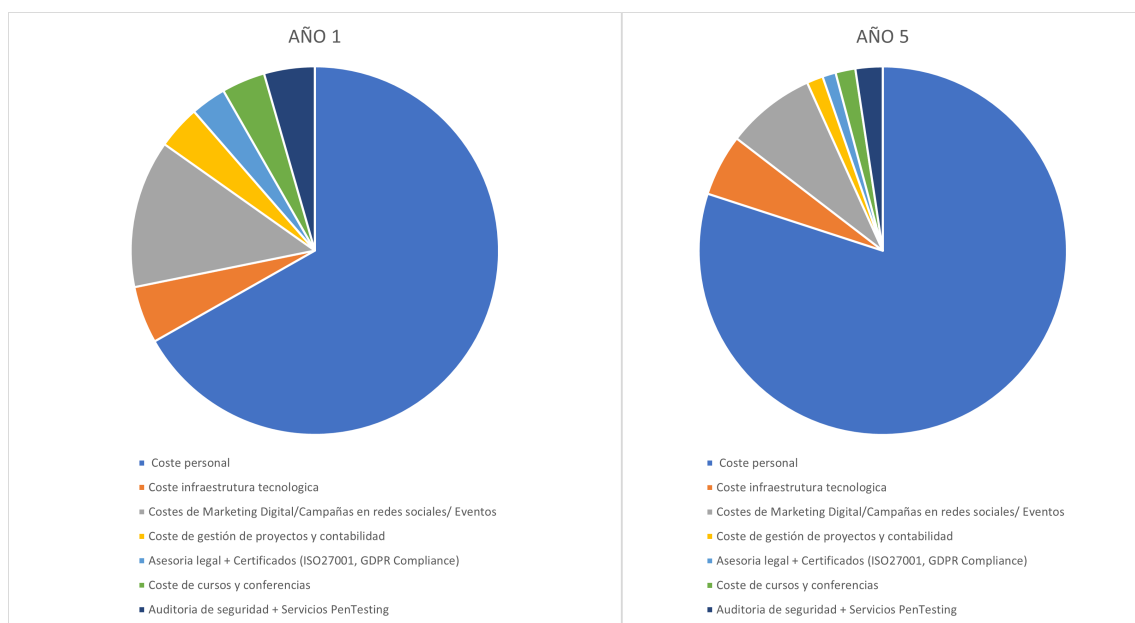


Figure 25: Comparison of Pie Charts of Cost Sources for Years 1 and 5

Finally, Figure 26 shows the evolution of after-tax profits, a very attractive metric for project investors. It shows a very positive evolution over the years, resulting in a 278.11% growth in the fifth year compared to the first. This reflects that, throughout the project's course, the impact of the main sources of income is greater compared to the impact of the most significant cost agents. Another interesting indicator for investors is the NPV. The estimated cash flows for the project result in an NPV of €398,556.44, making Codelingo a viable and profitable long-term project.

	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5
Profit before taxes (euros)	55,446	72,480	101,470	131,927	209,650
Taxes	13,862	18,120	25,367	32,982	52,413
Profit after taxes (euros)	41,585	54,360	76,102	98,945	157,238

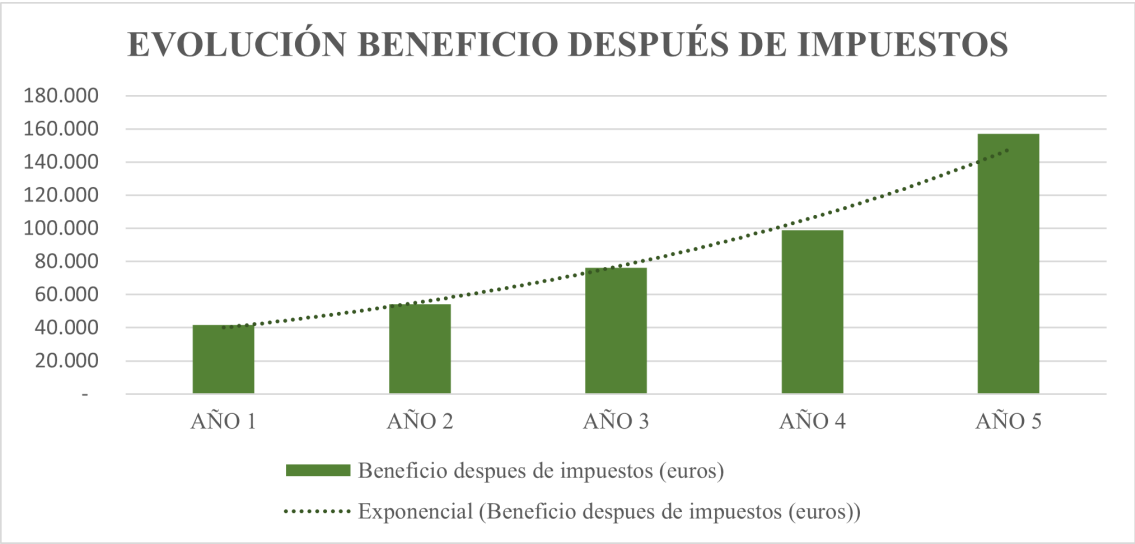


Figure 26: Evolution of After-Tax Profits Over 5 Years

8. Conclusions

The analysis of the Codelingo project reveals important aspects regarding the methodology and technologies used.

The applied methodology showed both successes and areas for improvement. A more thorough initial study of the technologies could have optimized time and improved the final product quality. The organization of work in sprints and the sequential development approach, starting with the frontend and ending with the recommendation system, were efficient. The encapsulation of functionalities, testing them separately before integration, ensured that each component worked correctly without affecting the overall project logic.

The use of Flutter, Dart, Firebase, and Heroku was crucial to achieving the objectives. Although the application deployment was not completed, configuring production environments and integrating analysis tools like Firebase Analytics would have provided a better understanding of the application's usage and performance, enhancing the user experience.

The GPT-3.5-turbo model, selected for its performance in natural language processing, excelled in several areas such as customer service, content generation, and data analysis. Additionally, the course recommendation system based on cosine similarity calculation showed promising results, offering effective content personalization.

For future developments, it is recommended to research and adopt more robust production technologies, implement analytical tools to obtain detailed user feedback, improve lesson personalization using machine learning, introduce gamified elements such as leagues and competitions among users, and organize thematic and collaborative activities to encourage participation. It is also vital to optimize data management and API responses to ensure the application's efficiency under high loads.

These improvements will not only enhance Codelingo's functionality and appeal but also ensure that the platform can scale and adapt to the demands of a professional production environment, offering a richer and more effective learning experience.

We shall never surrender

Agradecimientos

En primer lugar, quiero agradecer este trabajo a mis padres y a mi hermano, por sus ánimos diarios incansables y sus esfuerzos por brindarme la oportunidad de estudiar en esta universidad. Su motivación y apoyo constante han sido una fuente inagotable de inspiración y fuerza para mí.

A mis amigos, por estar siempre ahí, ofreciendo su ayuda, compañía y ánimos cuando más los necesitaba, y en especial a María Alsina, por ser mi mayor fan y esa persona que nunca se baja del barco. Sus consejos, recomendaciones, recordatorios y ,sobretudo, entusiasmo han sido un pilar fundamental en este viaje. "Los verdaderos amigos no se encuentran por casualidad, se hacen en el camino y se quedan para siempre."

Por último a mi director del proyecto, Atilano, por saber guiarme, motivarme, darme el feedback necesario y permitirme desarrollar el proyecto que quería realizar. Su conocimiento y dedicación han resultado fundamentales en todo momento.

Contents

1	Introducción	1
1.1	Enfoque del proyecto	2
1.2	Motivación	3
1.2.1	Relevancia de aprender a programar a una edad temprana	3
1.2.2	Transformación en la educación	3
1.2.3	Ampliación del Conocimiento	4
2	Descripción del Proyecto	5
2.1	Objetivos	5
2.2	Planificación	6
2.3	Metodología	9
3	Estado del Arte	11
3.1	Modelos de Aprendizaje en Línea	11
3.1.1	MOOCs	11
3.2	Herramientas y Plataformas Existentes para el Aprendizaje de Pro-	
	gramación	13
3.3	Tecnologías Emergentes en Educación y Programación	16
3.3.1	Inteligencia Artificial	16
3.3.2	Machine Learning	18
3.4	Análisis del Modelo de Duolingo	19
3.4.1	Gamificación en Duolingo	19
3.4.2	Personalización del Aprendizaje	20
4	Tecnología y Herramientas Utilizadas	22
4.1	Lenguajes de Programación	22
4.1.1	Flutter y Dart	22
4.1.2	Python	25
4.2	Diseño de Base de Datos en Firebase	26
4.3	Visual Studio Code	27
4.4	Android Studio	28
4.5	Github	30
4.5.1	Heroku	31

4.6 Paquetes del proyecto	32
5 Sistema Desarrollado	35
5.1 Diseño del Desarrollo	35
5.1.1 Esquema y Arquitectura del Sistema	35
5.1.2 Diagrama de Casos de Uso	38
5.1.3 Requisitos Funcionales	39
5.1.4 Requisitos No Funcionales	41
5.1.5 Componentes del Proyecto	43
5.2 Desarrollo del Backend	46
5.2.1 Authentication	46
5.2.2 Cloud Firestore Database	50
5.2.3 Implementación de Firebase en el Proyecto	53
5.3 Desarrollo del Frontend	54
5.3.1 Diseño UI/UX	55
5.4 Implementación de Inteligencia Artificial	68
5.5 Implementación de Machine Learning	72
5.5.1 Sistema de Recomendación Basado en Contenido	72
5.5.2 Beneficios del Sistema de Recomendación	78
5.5.3 Sincronización con Heroku	79
5.5.4 Datasets y Limpieza de Datos	79
6 Estudio Económico	81
6.1 Modelo de negocio	81
6.1.1 Propuesta de valor	83
6.2 Análisis DAFO	84
6.3 Estimación Económica	84
6.3.1 Precisión de Parámetros & Fuentes de ingresos y de costes	85
6.3.2 Análisis de resultados	90
7 Análisis de Resultados	95
7.1 Conclusiones sobre la metodología	95
7.2 Conclusiones sobre las Tecnologías	95
7.3 Análisis de los Resultados	96
7.4 Recomendaciones para futuros trabajos	97
Appendix	99
A Alineación del Proyecto con los ODS	99

B Manual de Instalación	102
B.1 Visual Studio Code	102
B.2 Flutter	103
B.3 Instalación de un Emulador para iOS	104
B.4 Verificación del Entorno de Desarrollo	105
B.5 Instalación de un Emulador de Android	105
B.6 Configuración del Desarrollo para Android	106
B.7 Prueba, Debug y Deploy de la Aplicación	107
B.7.1 Acceder a la Aplicación	107
B.7.2 Debug de la Aplicación	107
C ANEXO C	108
Bibliografía	114

List of Figures

1	Arquitectura de CodeLingo	iv
2	Pantalla de Lobby principal	v
3	Pantalla de Lecciones	vi
4	Pantalla de Ejercicios	vi
5	Pantalla de Ejercicios con ayuda de Cuddy	vi
6	Pantalla 1 de Perfil del usuario	vi
7	Pantalla 2 de Perfil del usuario	vi
8	Ejemplo de recomendación de cursos en base a preferencias del usuario	viii
9	Diagrama de Casos de Uso de CodeLingo	ix
10	Modelo de Negocio de CodeLingo	x
11	Comparativa de gráficos circulares de las fuentes de los ingresos del año 1 y 5	xi
12	Comparativa de gráficos circulares de las fuentes de los costos del año 1 y 5	xi
13	Evolución de los beneficios después de impuestos en 5 años	xii
14	CodeLingo Architecture	xvi
15	Main Lobby Screen	xvii
16	Lessons Screen	xviii
17	Exercises Screen	xviii
18	Exercises Screen with Cuddy's Help	xviii
19	User Profile Screen 1	xviii
20	User Profile Screen 2	xviii
21	Example of Course Recommendation Based on User Preferences	xx
22	CodeLingo Use Case Diagram	xxi
23	CodeLingo Business Model	xxii
24	Comparison of Pie Charts of Income Sources for Years 1 and 5	xxiii
25	Comparison of Pie Charts of Cost Sources for Years 1 and 5	xxiii
26	Evolution of After-Tax Profits Over 5 Years	xxiv
2.1	Esquema de Clean Code Architecture	7
2.2	Fases típicas de un proyecto de desarrollo de software	8
2.3	Etapas de la Metodología Agile	9

3.1	Ecosistema de MOOCs mostrando las conexiones entre diferentes plataformas y entidades.	13
3.2	Interfaz interactiva de Codecademy donde los estudiantes pueden escribir y probar código en tiempo real	14
3.3	Ejemplo de problema de codificación en LeetCode, con opciones para escribir y probar soluciones.	15
3.4	Interfaz de Codeforces durante una competición, mostrando la clasificación y los problemas.	16
3.5	Gasto esperado en Educación para el año 20230	17
3.6	Elementos de la gamificación en Duolingo	19
4.1	My BMW	23
4.2	Plataforma Dart	24
4.3	Bibliotecas utilizadas en el proyecto	25
4.4	Funcionalidades Firebase	27
4.5	Visualización de Codelingo en diferentes Emuladores Android	29
5.1	Arquitectura de Codelingo	35
5.2	Diagrama de Casos de Uso de CodeLingo	38
5.3	Proveedores de Authentication de Codelingo en la Consola de Firebase Authentication	47
5.4	Usuarios registrados en CodeLingo	48
5.5	Estructura de la base de datos de Cloud Firestore para la funcionalidad de chat en CodeLingo	50
5.6	Estructura de la base de datos de usuarios en Cloud Firestore	51
5.7	Índices configurados en Cloud Firestore	52
5.8	Mascota Duo de la app Duolingo	55
5.9	Coddy motivando a los usuarios	56
5.10	Pantalla de Home	57
5.11	Pantalla de Login	57
5.12	Pantalla de Registro	57
5.13	Pantalla 1 de Perfil del usuario	58
5.14	Pantalla 2 de Perfil del usuario	58
5.15	Pantalla de Lobby principal	59
5.16	Pantalla de Lecciones	60
5.17	Pantalla de Ejercicios	60
5.18	Pantalla de Ejercicios con ayuda de Coddy	60
5.19	Pantalla de Documentación	61
5.20	Pantalla de Vídeos	62
5.21	Pantalla de Cursos con buscador	63
5.22	Pantalla 1 del Chat perspectiva de Miguel	64
5.23	Pantalla 2 del Chat perspectiva de Jaime	64
5.24	Pantalla de descripción de Recommend Me	65

5.25 Pantalla de Nivel de experiencia	65
5.26 Pantalla de objetivo del aprendizaje	65
5.27 Pantalla de lenguaje de interés	66
5.28 Pantalla de framework de interés	66
5.29 Pantalla de área de interés	66
5.30 Pantalla de nivel del curso	67
5.31 Pantalla de formato de curso de interés	67
5.32 Pantalla de duración del curso	67
5.33 Pantalla de las categorías de los desafíos	68
5.34 Pantalla los niveles de los desafíos	68
5.35 Pantalla de ejercicios de desafío	68
5.36 Esquema de Arquitectura transformers ChatGPT	69
5.37 Captura de pantalla funcionalidad "Preguntar a Cuddy"	71
5.38 Contador de cursos por valoraciones	73
5.39 Ejemplo de similitud del coseno	74
5.40 Resultado 1 de Recomendación de cursos	76
5.41 Resultado 2 de Recomendación de cursos	77
5.42 Implementación del sistema en Heroku	79
5.43 Distribución de los cursos de programación por nivel de dificultad	80
6.1 Modelo de Negocio de Codelingo	81
6.2 Propuesta de Valor de Codelingo	83
6.3 Gráfico de barras de la evolución de ingresos del proyecto	91
6.4 Comparativa de gráficos circulares de las fuentes de los ingresos del año 1 y 5	91
6.5 Gráfico de barras de la evolución de costes del proyecto	92
6.6 Comparativa de gráficos circulares de las fuentes de los costes del año 1 y 5	93
6.7 Evolución de los beneficios después de impuestos en 5 años	94
A.1 Objetivos de desarrollo sostenible	100
B.1 Página de descarga de Visual Studio Code	103
B.2 Página de descarga de Extensión de Flutter	103
B.3 Página de descarga de Extensión de Dart	103
B.4 Página de descarga del SDK de Flutter	104
B.5 Verificación en la terminal de flutter doctor	105
B.6 Verificación en la terminal de flutter doctor	106
B.7 Página de descarga de Android Studio	106

List of Tables

1	Tabla de preferencias de entrada	vii
2	Tabla de campos de los cursos del dataset	vii
3	Input Preferences Table	xix
4	Course Dataset Fields Table	xix
3.1	Elementos de gamificación en Duolingo	20
5.1	Requisitos Funcionales de CodeLingo	40
5.2	Requisitos No Funcionales de CodeLingo	42
5.3	Tabla de preferencias de entrada	72
5.4	Tabla de campos de los cursos del dataset	73
6.1	Análisis DAFO	84
C.1	Ingresos proyectados de Codelingo	109
C.2	Costes de personal de Codelingo	110
C.3	Costes de infraestructura tecnológica de Codelingo	111
C.4	Costes de marketing, administración, legales, formación y ciberseguridad de Codelingo	112
C.5	Cuenta de resultados de Codelingo	113

Listings

5.1	Servicio de autenticación en Flutter	48
5.2	Reglas de seguridad de Firebase Firestore	51
5.3	Servicio de chat en Flutter	52
5.4	Dependencias de Firebase en <code>pubspec.yaml</code>	54
5.5	Inicialización de Firebase en el archivo principal	54
5.6	Función <code>_askCuddy</code>	70
5.7	Ejemplo de preprocesamiento de datos	80

Chapter 1

Introducción

"Programar es una necesidad desde la infancia en un mundo en el que la tecnología está cambiando la manera en que nos relacionamos e impregna todas las formas actuales de comunicación y creatividad. Su estudio implica desarrollar la capacidad analítica y el pensamiento crítico." [5]

A medida que la industria de la tecnología continúa expandiéndose, el conocimiento de lenguajes de programación se ha convertido en una habilidad esencial. Según un informe de las Naciones Unidas, la programación es una habilidad valiosa que puede abrir muchas puertas en el mundo laboral, y su demanda ha aumentado significativamente en la última década [19]. La demanda de profesionales altamente cualificados en lenguajes de programación no ha hecho más que aumentar en la última década, y se espera que siga siendo así en el futuro. Según datos recientes, se estima un aumento del 22% en los puestos de desarrollador de software para el año 2030 [18], siendo los sectores que demandan más programadores en España las empresas TIC y el sector bancario y financiero.

Sin embargo, aprender a programar no es exclusivo para quienes se dedican al desarrollo de software. Las competencias en programación son necesarias en muchos otros campos, como la ciencia de datos, la inteligencia artificial, la ingeniería, la economía, la biología, y más.

Según datos de Nuclio Digital School, tras el confinamiento de 2020, el interés por los cursos intensivos de programación, conocidos como bootcamps, ha crecido un 30%. Este aumento está relacionado con la creciente empleabilidad en el sector del desarrollo de software, impulsada por una acelerada transformación digital [2]. Lo que normalmente habría llevado 8 años en condiciones normales se logró en solo 8 meses debido a las circunstancias vividas en 2020.

La encuesta "2023 Developer Survey" de Stack Overflow, una de las más am-

CHAPTER 1. INTRODUCCIÓN

plias y detalladas en la comunidad de desarrolladores, ha proporcionado información valiosa sobre las tendencias y desafíos actuales en el aprendizaje de programación. Esta encuesta recoge datos de miles de desarrolladores de todo el mundo, ofreciendo una visión integral de las experiencias, preferencias y obstáculos que enfrentan tanto los principiantes como los expertos en el campo de la programación.

Uno de los datos más relevantes revelados por la encuesta es que el 80% de los encuestados aprendieron a programar utilizando recursos en línea, como videos, blogs y foros [11]. Este porcentaje ha aumentado significativamente desde la encuesta de 2022, reflejando una clara tendencia hacia el aprendizaje autodirigido y la preferencia por métodos de estudio más flexibles y accesibles. Además, los datos muestran que los jóvenes de 18 años o menos son los más propensos a utilizar estos recursos en línea, mientras que los encuestados de entre 25 y 34 años combinan el aprendizaje en línea con cursos tradicionales y certificaciones.

La programación no es solo una habilidad técnica, sino el núcleo de la innovación en la era digital y una herramienta para resolver problemas en un mundo cada vez más digitalizado. En este marco, aprender a programar es abrir una puerta hacia un futuro lleno de oportunidades y avances.

1.1 Enfoque del proyecto

El enfoque de este proyecto es la creación de una plataforma cuyo objetivo es enseñar programación de manera interactiva para que usuarios de todas las edades aprendan habilidades desde el principio sin necesidad de leer extensas documentaciones ni libros. Gracias al uso de Inteligencia Artificial y técnicas de Machine Learning, se logrará personalizar el aprendizaje y adecuarlo al nivel de cada usuario, motivándolos para alcanzar sus metas a la vez que entreteniéndolos para que la plataforma sea disfrutada como un juego. Para esto, se analizarán modelos de diferentes aplicaciones EdTech como el de Duolingo. Se estudiarán las mejores tecnologías para favorecer la retención de los usuarios mediante un sistema de recomendación de cursos y lecciones atendiendo a las métricas de progreso de cada uno de ellos. También se realizará un estudio económico (business case) una vez analizadas las distintas variables del producto y de los servicios ofrecidos para asegurar la viabilidad del proyecto.

El objetivo no es solo facilitar el acceso a la programación, sino también motivar a los estudiantes a través de una experiencia práctica y atractiva. La plataforma proporcionará ejercicios interactivos, proyectos y feedback inmediato, lo que permite a los usuarios aplicar sus conocimientos de manera efectiva desde un primer momento. Además, la comunidad de CodeLingo fomentará la colaboración y el intercambio de conocimientos, creando un entorno social que acompañe a los estudiantes durante las diversas etapas de su aprendizaje.

1.2 Motivación

El propósito detrás de la investigación y desarrollo de esta plataforma está impulsado por una serie de factores que convergen en la necesidad de abordar desafíos tanto educativos como tecnológicos en el campo de la programación. Esta sección responde a la pregunta: ¿Por qué desarrollar una plataforma de aprendizaje de lenguajes de programación?

1.2.1 Relevancia de aprender a programar a una edad temprana

Dada la mencionada relevancia de la programación en el entorno actual y sus estadísticas de crecimiento y demanda es necesario implementar aprender a programar desde una corta edad. Aprender a programar no solo implica adquirir habilidades técnicas, sino también desarrollar capacidades cruciales como la resolución de problemas, la estructuración del pensamiento y la creatividad. Introducir la programación temprano en la educación fomenta el desarrollo de un pensamiento lógico y estructurado, habilidades que son fundamentales en cualquier campo profesional.

Es por ello que en 2023 surgió el Programa Código Escuela 4.0, impulsado por el Ministerio de Educación y Formación Profesional, el cual está desempeñando un papel crucial al introducir conceptos de programación y pensamiento computacional en las aulas tanto de primaria como de ESO y Bachillerato. A través de asignaturas como informática, robótica, diseño e impresión 3D, realidad aumentada, virtual y mixta, placas programables e inteligencia artificial, su objetivo es preparar a las futuras generaciones para adaptarse a un mundo cada vez más digitalizado [3]. Este programa tiene como objetivos el acceso a lenguajes de programación adaptados, como Scratch, en el que se introducen conceptos de programación y pensamiento computacional en las aulas desde una temprana edad. Esto garantiza que las nuevas generaciones puedan estar preparadas para engrentar los desafíos tecnológicos del futuro.

1.2.2 Transformación en la educación

A día de hoy, la educación en programación presenta desafíos como la falta de accesibilidad para un público diverso, la complejidad percibida de los distintos lenguajes de programación y la falta de motivación para mantener un aprendizaje constante. El proceso de aprendizaje de programación puede ser intimidante para muchos, especialmente para aquellos que son nuevos en el campo. La mayoría de los recursos disponibles requieren una lectura extensa de teoría y una inversión significativa de tiempo antes de que los estudiantes puedan comenzar a aplicar sus conocimientos de manera práctica. Aprender a programar en un lenguaje específico implica adquirir

CHAPTER 1. INTRODUCCIÓN

conocimientos sobre sus estructuras y particularidades, así como familiarizarse con el entorno de desarrollo y las herramientas necesarias para su ejecución. Para aquellos que ya poseen ciertas habilidades, este proceso se simplifica al relacionar estas nuevas nociones con su base previa de conocimiento. La combinación del aprendizaje de programación y las herramientas correspondientes puede generar una curva de aprendizaje pronunciada.

Para suavizar esta curva, CodeLingo aborda esta limitación proporcionando una experiencia de aprendizaje altamente práctica, intuitiva y adaptativa, similar a la metodología utilizada por Duolingo y otras EdTechs. Asimismo, este tipo de plataformas BYOD (Bring Your Own Device) facilitan la accesibilidad en cualquier momento y desde cualquier lugar por medio de una app en tu dispositivo electrónico.

Esta plataforma se propone como una solución innovadora que plantea otra forma en la cual las personas abordan el aprendizaje de lenguajes de programación, ofreciendo una experiencia efectiva, práctica y atractiva para estudiantes de todas las edades y niveles de habilidad.

1.2.3 Ampliación del Conocimiento

Según el cofundador y famoso presidente ejecutivo de la marca Apple Steve Jobs: *“Todo el mundo debería aprender programación, porque te enseña a cómo pensar”*. Esta afirmación destaca la importancia de la programación no solo como una habilidad técnica, sino también como una herramienta fundamental para el desarrollo del pensamiento crítico y la resolución de problemas.

Estudios recientes han demostrado que aprender a programar tiene numerosos beneficios cognitivos. Un meta-análisis de la editorial de publicaciones científicas Frontiers encontró que aprender a programar tiene un efecto positivo y significativo en habilidades cognitivas como el pensamiento crítico, la resolución de problemas y las habilidades espaciales. En él se argumentaba que aprender a programar involucra procesos de pensamiento similares a otros de campos como el del modelado matemático [12]. Estos beneficios son particularmente evidentes en contextos educativos bien diseñados que fomentan un aprendizaje activo y centrado en el estudiante. En resumen, la programación enseña a los individuos a descomponer problemas complejos, diseñar soluciones eficientes, pensar de manera lógica y a utilizar el sentido común, habilidades que son transferibles a cualquier disciplina.

Chapter 2

Descripción del Proyecto

2.1 Objetivos

El objetivo principal de CodeLingo es desarrollar una plataforma educativa que revolucione la forma en que las personas aprenden lenguajes de programación, proporcionando una herramienta accesible, práctica y atractiva para estudiantes de todas las edades y niveles de habilidad. La plataforma busca:

- **Facilitar el acceso al aprendizaje de programación:** Ofrecer una interfaz amigable y un diseño intuitivo que permita a los usuarios comenzar a aprender programación sin necesidad de conocimientos previos.
- **Personalizar la experiencia de aprendizaje:** Utilizar técnicas de Machine Learning para adaptar el contenido y los ejercicios a las necesidades y el progreso individual de cada usuario.
- **Motivar a los estudiantes:** Emplear elementos de gamificación, como recompensas y niveles, para mantener a los usuarios comprometidos y motivados.
- **Fomentar la práctica continua:** Proporcionar ejercicios prácticos y proyectos desde las primeras etapas del aprendizaje, para que los estudiantes puedan aplicar inmediatamente lo que han aprendido.
- **Crear una comunidad de aprendizaje:** Facilitar la interacción entre los usuarios a través de foros, retos y proyectos colaborativos, fomentando un ambiente de aprendizaje social y colaborativo.

Al ofrecer una plataforma de aprendizaje de programación accesible y personalizada, se anticipa que CodeLingo ayudará a reducir las barreras de entrada al mundo de la programación, permitiendo que más personas adquieran habilidades tecnológicas críticas en un entorno flexible y atractivo. Esto, a su vez, puede contribuir a aumentar la empleabilidad de los usuarios al proporcionarles competencias demandadas en el mercado laboral actual.

CHAPTER 2. DESCRIPCIÓN DEL PROYECTO

Además, al fomentar una comunidad de aprendizaje colaborativa y basada en la práctica, CodeLingo puede impulsar la innovación y la creatividad entre los usuarios, preparándolos para enfrentar desafíos tecnológicos reales y promover su desarrollo profesional continuo.

2.2 Planificación

El desarrollo de CodeLingo se ha realizado siguiendo las normas y buenas prácticas establecidas en manuales, guías y paradigmas de programación reconocidos, asegurando que todo el proceso se lleve a cabo conforme a los estándares de un trabajo profesional de ingeniería de software. El proyecto abarca varias fases de desarrollo y despliegue, asegurando que la plataforma no solo cumpla con los objetivos establecidos, sino que también sea sostenible y escalable a largo plazo [8]. Las principales fases del proyecto incluyen:

- 1. Fase de Investigación y Planificación:** En esta etapa inicial, se llevó a cabo una investigación del mercado de juegos que enseñan a programar (SoloLearn, CodeCombat o Python Challenge) e identificar las necesidades de los usuarios. Se utilizó una combinación de encuestas con potenciales usuarios para identificar las funcionalidades más demandadas como la ya mencionada anteriormente de la plataforma Stack Overflow o la realizada por la plataforma FreeCodeCamp en 2021 a dieciocho mil individuos [6] de la cual se pudieron extraer datos tan valiosos como la mediana de tiempo que pasaban aprendiendo a programar, el tipo de recursos que usaban para aprender a programar, sus orígenes, motivaciones, trabajos, intereses profesionales y muchos más. De esta manera se definieron objetivos específicos y se establecieron los requerimientos funcionales y no funcionales de la plataforma CodeLingo.
- 2. Desarrollo del Prototipo:** Se desarrolló un prototipo funcional de la plataforma que incluye características esenciales como la interfaz de usuario, las pantallas de registro y login, y los primeros módulos de aprendizaje, como una primera toma de contacto con Flutter y Dart. Durante la creación de la interfaz, se analizaron las principales tendencias de diseño presentes en plataformas como Figma, Dribbble y Pinterest, las cuales cuentan con una gran cantidad de templates y ejemplos subidos por su comunidad. En esta fase inicial, se aplicaron principios y patrones de diseño modernos, tales como la Clean Architecture, popularizada por Robert C. Martin. Este enfoque asegura que el sistema esté estructurado en capas (UI, Data, Device y Domain), donde cada capa tiene responsabilidades específicas y se comunica únicamente con las capas inmediatamente contiguas [13]. Estas prácticas no solo hicieron el código más mantenible y escalable, sino que también facilitaron su extensión futura. La Clean Architecture, junto con los principios de Clean Code y el diseño modular, permitió que el prototipo de CodeLingo se construyera con un enfoque

CHAPTER 2. DESCRIPCIÓN DEL PROYECTO

robusto y sostenible. La aplicación de estas metodologías garantizó que el código fuera fácil de entender, modificar y mantener, sentando una base sólida para el desarrollo continuo y la evolución de la plataforma.

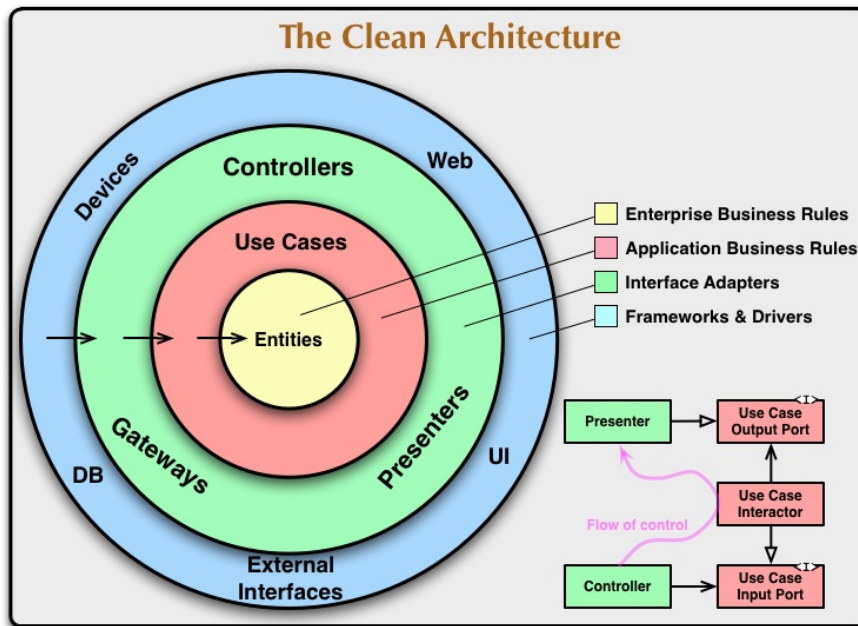


Figure 2.1: Esquema de Clean Code Architecture

3. **Implementación de Funcionalidades Avanzadas:** Basándose en el prototipo y en la información obtenida a partir de las encuestas mencionadas, se planificó la adición de funcionalidades avanzadas como el algoritmo de Machine Learning para la personalización del contenido y el sistema de recomendación de cursos. Se propusieron elementos de gamificación como un sistema de medallas, logros y recompensas, inspirados en el éxito de aplicaciones como Duolingo. También se consideraron herramientas de colaboración entre usuarios, como un foro de preguntas o un chat entre seguidores, similares a los que se encuentran en MOOCs como Moodle, Udemy o Coursera. Además, se planificó la integración de recursos gráficos y multimedia para enriquecer la experiencia de aprendizaje, utilizando bibliotecas como Flutter SVG para manejar gráficos vectoriales y asegurar que la app fuera responsiva a diferentes tamaños de ventana. Como trabajo futuro, se buscará la obtención de licencias para tecnologías que puedan ser utilizadas en producción, permitiendo así su despliegue en plataformas como Apple Store y Google Play.
4. **Pruebas y Optimización:** Se realizó una fase de pruebas con usuarios beta para identificar y corregir errores, mejorar la usabilidad y optimizar el rendimiento de la plataforma. Este proceso incluyó pruebas de integración para asegurar que los diferentes módulos funcionaran juntos sin problemas, y

CHAPTER 2. DESCRIPCIÓN DEL PROYECTO

pruebas de aceptación del usuario para validar que la plataforma cumpliera con los requisitos definidos.

- 5. Despliegue y Monitoreo:** Se planeó el lanzamiento oficial de CodeLingo en la PlayStore tras completar las pruebas y optimizaciones. Se propusieron mecanismos de monitoreo y análisis continuo para evaluar el desempeño de la plataforma utilizando Google Cloud Platform y Firebase, lo cual permitiría la realización de mejoras continuas basadas en los bugs reportados por los usuarios y el análisis de datos de uso. Para ello, se usarían herramientas de monitoreo y análisis como Google Analytics y Firebase Analytics, con el objetivo de recolectar y analizar datos sobre el uso de la plataforma, identificar áreas de mejora y facilitar la implementación de actualizaciones necesarias. Como trabajo futuro, se buscará la obtención de licencias para tecnologías que puedan ser utilizadas en producción, permitiendo así su despliegue en plataformas como Apple Store y Google Play.
- 6. Mantenimiento y Actualizaciones:** El proceso de desarrollo sigue un ciclo de vida iterativo, permitiendo la incorporación de mejoras continuas y adaptaciones según las necesidades emergentes de los usuarios. Así mismo, se establece un sistema de gestión de versiones utilizando Git y GitHub para mantener el control de cambios y facilitar la colaboración entre los desarrolladores.

En la Figura [2.2](#) se describen las fases típicas del desarrollo de software, lo cual es fundamental para entender la estructura del proyecto CodeLingo.

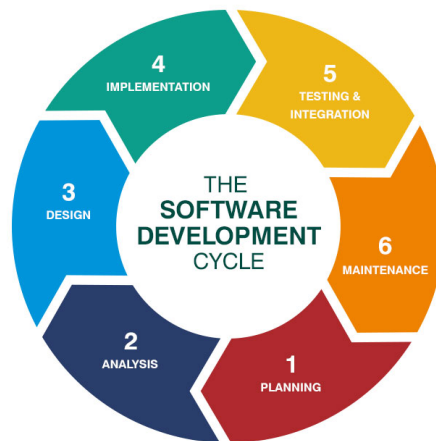


Figure 2.2: Fases típicas de un proyecto de desarrollo de software

2.3 Metodología

El desarrollo de CodeLingo se ha basado en una metodología Agile, ampliamente reconocida y utilizada en la industria del software actual. Este enfoque se caracteriza por su flexibilidad y capacidad de adaptación, convirtiéndolo en la opción ideal para un proyecto así.

Esta metodología se estructura en interacciones, divididas a su vez en sprints, que son periodos cortos de tiempo con objetivos específicos. Cada sprint se centra en la entrega de funcionalidades concretas, permitiendo avances visibles y tangibles del proyecto. Al finalizar cada sprint, se realiza una evaluación exhaustiva del trabajo realizado. Esto garantiza que el producto se encuentre en un estado utilizable en todo momento, lo que a su vez permite obtener retroalimentación de los usuarios y realizar ajustes o mejoras. Además la metodología ágil fomenta la colaboración constante entre miembros del equipo, promoviendo una toma de decisiones más ágil y efectiva. Estas ventajas combinadas contribuyen a la creación de software de calidad y al cumplimiento de los objetivos del proyecto de manera eficiente.

En la Figura 2.3 se describen las fases típicas en forma de ciclo para la aplicación de la metodología agile, desde el planteamiento hasta su lanzamiento posterior.

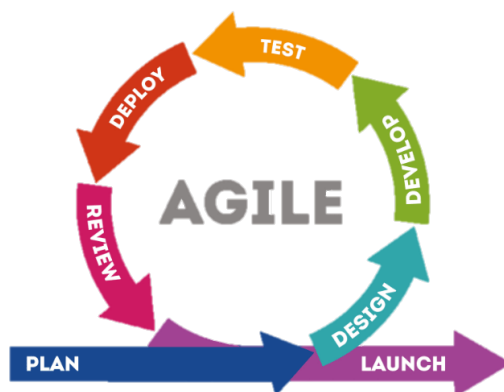


Figure 2.3: Etapas de la Metodología Agile

Chapter 3

Estado del Arte

Problema analizado: desarrollo y viabilidad de una plataforma de aprendizaje de lenguajes de programación respaldada por técnicas de Machine Learning, siguiendo el modelo exitoso de Duolingo para el aprendizaje de idiomas

3.1 Modelos de Aprendizaje en Línea

El aprendizaje en línea ha transformado la educación en la última década. Este cambio se ha visto impulsado por el desarrollo de tecnologías avanzadas y el acceso a internet de alta velocidad en prácticamente el mundo entero. Dentro de los modelos de aprendizaje en línea, los MOOCs (Massive Open Online Courses) y las plataformas personalizadas han emergido como las más prominentes, ofreciendo soluciones educativas accesibles y efectivas a nivel global.

3.1.1 MOOCs

Los MOOCs representan una innovación significativa en el ámbito de la educación, proporcionando acceso democratizado al conocimiento y promoviendo el aprendizaje colaborativo a gran escala. En el contexto del proyecto CodeLingo, la incorporación de funcionalidades inspiradas en los MOOCs puede enriquecer significativamente la experiencia de aprendizaje.

Los MOOCs se distinguen por su naturaleza masiva, abierta y en línea, permitiendo que un número ilimitado de estudiantes de diversos perfiles accedan a contenido educativo de alta calidad de forma gratuita y a través de internet. Esta accesibilidad ha sido un factor clave en su creciente popularidad, posicionándolos como una alternativa valiosa para la formación continua y el desarrollo profesional.

Uno de los principales atributos de los MOOCs es su enfoque en la colaboración y el trabajo en equipo. Las plataformas de MOOCs integran diversas herramientas

CHAPTER 3. ESTADO DEL ARTE

diseñadas para facilitar la interacción entre estudiantes, fomentando un ambiente de aprendizaje colaborativo y dinámico. Algunas de estas herramientas incluyen:

- **Foros de discusión:** Espacios virtuales donde los participantes pueden intercambiar ideas, plantear dudas, compartir recursos y colaborar en proyectos grupales. Estos foros actúan como un núcleo central de intercambio de conocimiento, permitiendo a los estudiantes aprender unos de otros y resolver problemas en conjunto.
- **Comentarios y anotaciones:** Funcionalidades que permiten a los estudiantes dejar comentarios en los materiales del curso. Esto no solo enriquece el contenido, sino que también promueve el debate y la reflexión crítica entre los participantes, facilitando una comprensión más profunda del material.
- **Grupos de trabajo:** Herramientas que permiten la creación de grupos virtuales donde los estudiantes pueden trabajar juntos en tareas específicas. Esta colaboración no solo fomenta el aprendizaje entre pares, sino que también desarrolla habilidades blandas como la comunicación y el trabajo en equipo.

La integración de estas funcionalidades en la plataforma CodeLingo tiene el potencial de transformar la experiencia de aprendizaje de los usuarios. Al facilitar la interacción y la colaboración entre los estudiantes, se pueden promover entornos de aprendizaje más ricos y dinámicos. Además, estas herramientas pueden ayudar a los usuarios a desarrollar habilidades blandas esenciales, como la resolución de problemas, la comunicación efectiva y la capacidad de trabajar en equipo. En términos de implementación, se puede considerar la utilización de foros de discusión para que los estudiantes de CodeLingo intercambien ideas sobre proyectos de programación, compartan soluciones de código y se apoyen mutuamente en el proceso de aprendizaje. Los comentarios y anotaciones en los materiales del curso pueden proporcionar un valor añadido al contenido, permitiendo a los usuarios contribuir con sus propios conocimientos y perspectivas. Por último, la creación de grupos de trabajo puede facilitar la colaboración en proyectos de programación más complejos, incentivando a los usuarios a aplicar sus conocimientos en un entorno práctico y colaborativo.

La Figura 3.1 ilustra el ecosistema de los MOOCs, destacando las conexiones entre diversas plataformas como Coursera, Khan Academy, Udacity y edX, y sus vínculos con universidades, organizaciones sin fines de lucro, capital de riesgo y otras entidades. Esta red de conexiones resalta la colaboración y el apoyo mutuo en el desarrollo y la difusión de cursos en línea, subrayando la importancia de la cooperación interdisciplinaria y multi-institucional en la educación moderna.

La adopción de estas prácticas, inspiradas en los MOOCs, en CodeLingo no solo mejorará la experiencia de aprendizaje, sino que también fortalecerá la comunidad de usuarios, creando un ecosistema de aprendizaje interactivo y apoyado en la colaboración.

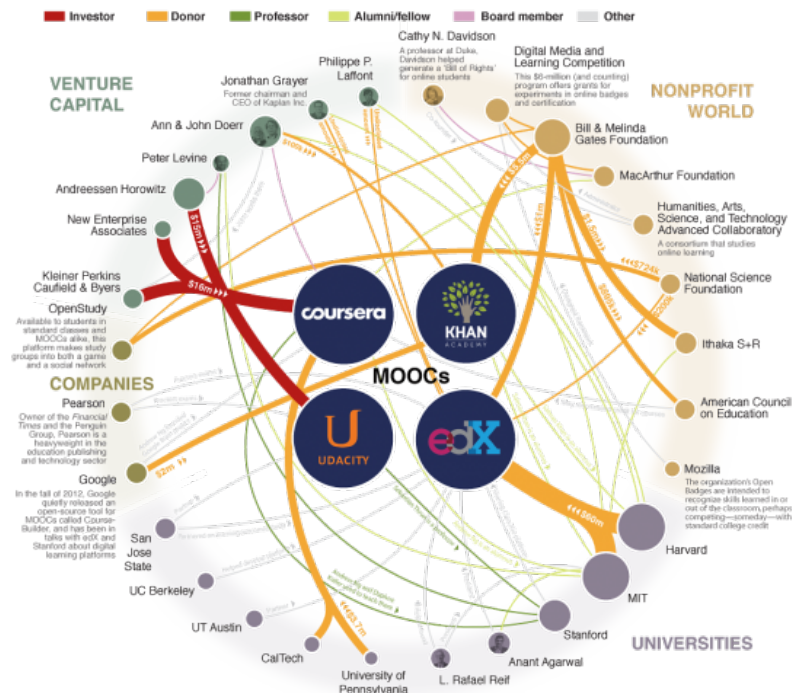


Figure 3.1: Ecosistema de MOOCs mostrando las conexiones entre diferentes plataformas y entidades.

3.2 Herramientas y Plataformas Existentes para el Aprendizaje de Programación

El aprendizaje de la programación se ha beneficiado enormemente del desarrollo de diversas herramientas y plataformas. Estas plataformas se pueden dividir en dos categorías principales: plataformas interactivas y cursos en línea. Cada una de estas categorías ofrece distintos enfoques y metodologías para aprender a programar, cada una con sus propias ventajas y desventajas.

Las plataformas educativas en línea han revolucionado la forma en que se enseña y se aprende programación. A continuación, se presenta una revisión de tres de las plataformas más populares: Codecademy, LeetCode y Codeforces, destacando sus características, ventajas y desventajas.

Codecademy

Codecademy, lanzada en 2011, es conocida por su enfoque interactivo en la enseñanza de la programación. La plataforma ofrece cursos en varios lenguajes de programación, incluyendo Python, JavaScript y Ruby. Su principal atractivo es la interactividad: los estudiantes pueden escribir y ejecutar código directamente en el navegador, recibiendo retroalimentación inmediata sobre sus errores y aciertos.

Codecademy es especialmente efectiva para principiantes debido a su presentación

CHAPTER 3. ESTADO DEL ARTE

clara y progresiva de los conceptos. Sin embargo, su profundidad en cursos avanzados es limitada y muchas características avanzadas están disponibles solo mediante una suscripción de pago, lo que puede ser una barrera para algunos estudiantes.

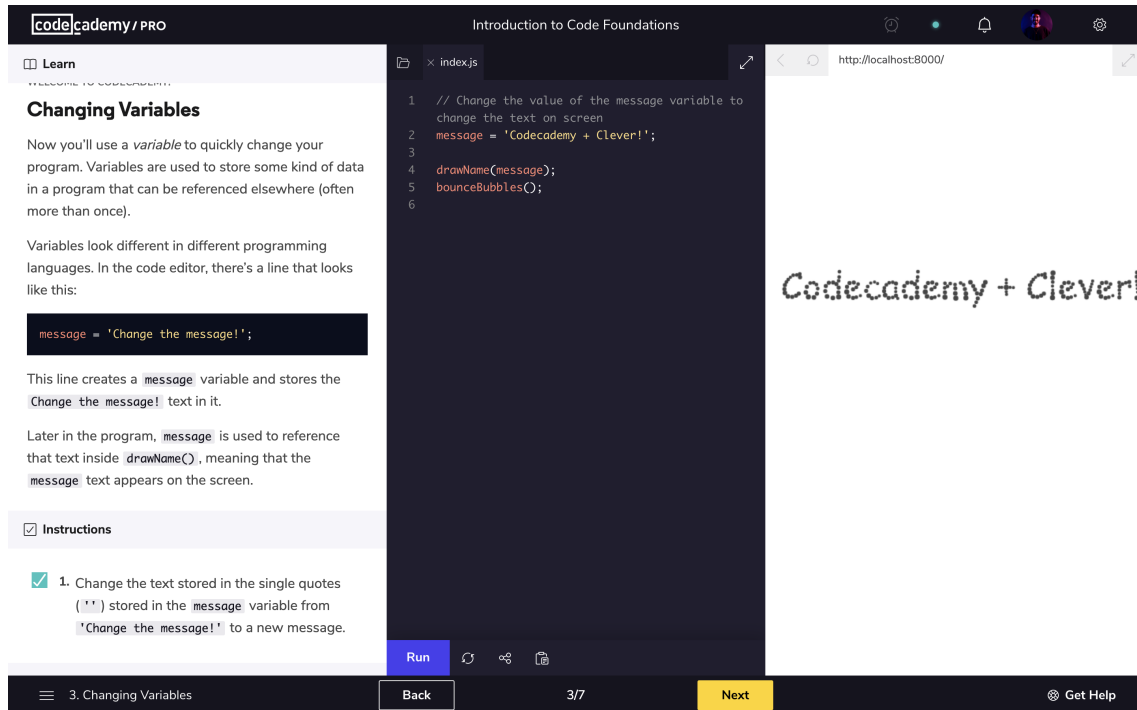


Figure 3.2: Interfaz interactiva de Codecademy donde los estudiantes pueden escribir y probar código en tiempo real

CHAPTER 3. ESTADO DEL ARTE

LeetCode

LeetCode es una plataforma ampliamente utilizada para mejorar habilidades en algoritmos y estructuras de datos, así como para prepararse para entrevistas técnicas en empresas tecnológicas. A diferencia de Codecademy, LeetCode se centra en proporcionar problemas de codificación que abarcan diversos niveles de dificultad y temas.

LeetCode es particularmente valioso para la preparación de entrevistas técnicas en grandes empresas como Google, Facebook y Amazon (FAANG). La plataforma ofrece un entorno para practicar y perfeccionar habilidades necesarias para estas entrevistas. Sin embargo, puede ser intimidante para principiantes debido a la falta de contenido estructurado y la complejidad de algunos problemas.

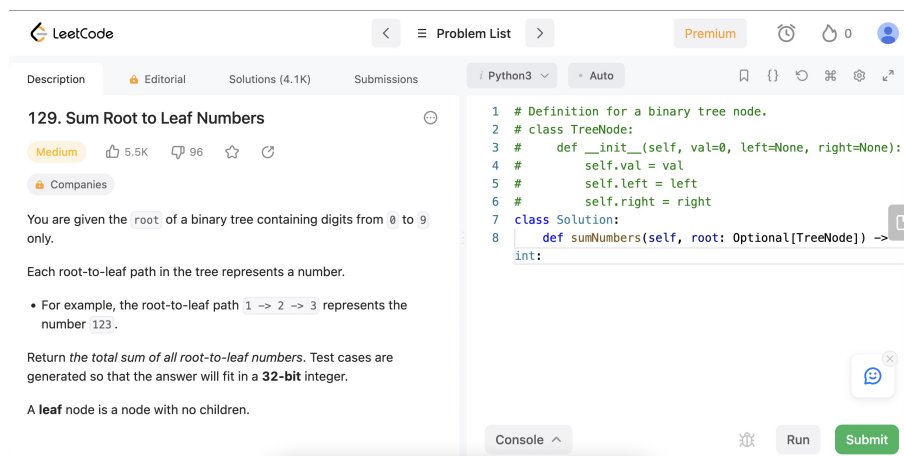


Figure 3.3: Ejemplo de problema de codificación en LeetCode, con opciones para escribir y probar soluciones.

Codeforces

Codeforces, fundada en 2010, es una plataforma popular en la comunidad de programación competitiva. La plataforma organiza competiciones regulares, conocidas como "contests", donde los programadores pueden poner a prueba sus habilidades en tiempo real.

Una característica destacada de Codeforces es su sistema de rating, que clasifica a los participantes según su desempeño en las competiciones. Esto no solo proporciona una medida de habilidad relativa, sino que también añade un elemento de gamificación que motiva a los usuarios a mejorar. Sin embargo, la plataforma puede ser abrumadora para los nuevos programadores debido a la dificultad de los problemas y la falta de un camino de aprendizaje estructurado.

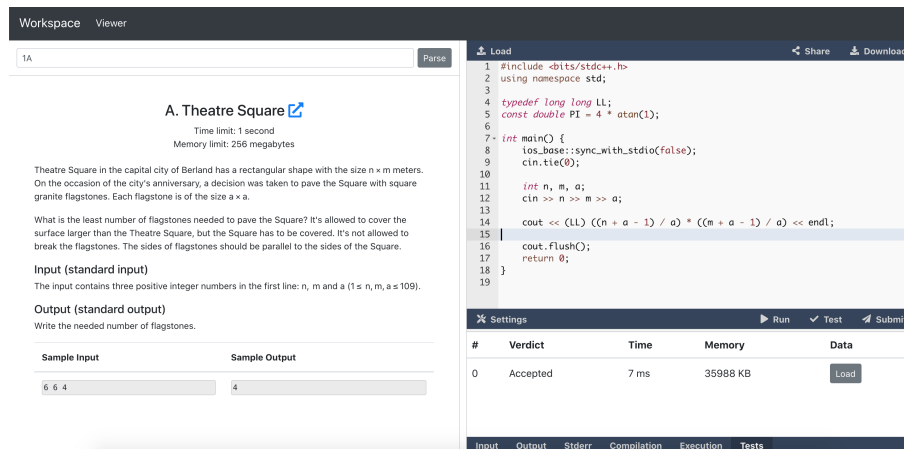


Figure 3.4: Interfaz de Codeforces durante una competición, mostrando la clasificación y los problemas.

Conclusión

Estas plataformas educativas en línea ofrecen distintas ventajas y se adaptan a diferentes necesidades y niveles de experiencia. Codecademy es ideal para principiantes que buscan una introducción interactiva a la programación, LeetCode es excelente para aquellos que desean mejorar sus habilidades algorítmicas y prepararse para entrevistas técnicas, y Codeforces es invaluable para los interesados en la programación competitiva. La elección de la plataforma adecuada depende de los objetivos específicos y el nivel de experiencia del usuario.

3.3 Tecnologías Emergentes en Educación y Programación

3.3.1 Inteligencia Artificial

La inteligencia artificial (IA) ha revolucionado el ámbito educativo al introducir innovaciones que han mejorado significativamente la experiencia de aprendizaje. Herramientas avanzadas, como los tutores virtuales y las plataformas personalizadas, hacen que la educación sea más accesible, interactiva y eficiente. Estas tecnologías permiten a los estudiantes aprender a su propio ritmo y recibir apoyo personalizado en cualquier momento, eliminando las barreras tradicionales del aula. Según un informe de HolonIQ, se espera que el mercado global de IA en la educación crezca a una tasa compuesta anual del 45.12%, alcanzando los 10 mil millones de dólares en 2030 [9].

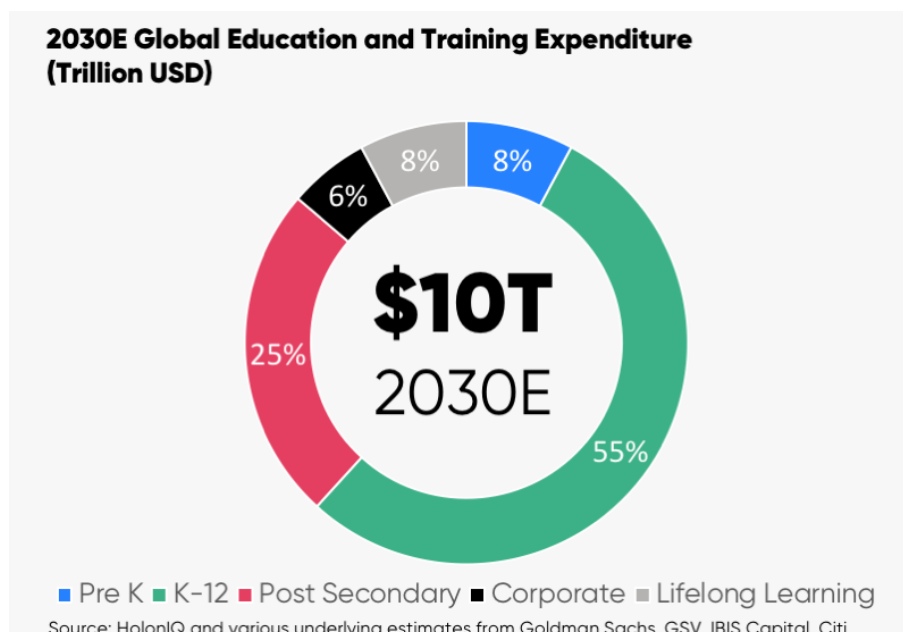


Figure 3.5: Gasto esperado en Educación para el año 2030

Entre las principales ventajas de la IA en la educación se encuentra el aprendizaje adaptativo, que ajusta el contenido y la dificultad de las lecciones según el progreso y las necesidades individuales de cada estudiante. Además, la retroalimentación inmediata permite a los alumnos conocer sus errores y corregirlos al instante, mejorando así el proceso de aprendizaje. La IA también identifica patrones de aprendizaje y áreas problemáticas comunes, proporcionando a los docentes datos valiosos para personalizar su enseñanza. Por último, la asistencia en tiempo real, a través de chatbots y tutores virtuales, ofrece apoyo continuo a los estudiantes, resolviendo sus dudas y motivándolos a continuar aprendiendo. un estudio realizado por la Universidad de Stanford encontró que el uso de tutores virtuales basados en IA puede aumentar la retención de conocimientos en un 30% [17].

Como casos de éxito pasados, se analiza el caso de Khan Academy, una institución educativa estadounidense, sin ánimo de lucro fundada en 2008, que dispone de ejercicios, vídeos y cursos en línea tanto de ciencias y humanidades como de matemáticas. Como muestra de su éxito, Khan Academy se encuentra disponible en más de 50 idiomas y tiene presencia en más de 190 países registrando más de 150 millones de usuarios en la actualidad. Durante marzo de 2023 la institución lanzó su nuevo instrumento de tutoría basado en la IA llamado Khamingo, que consiste en un tutor virtual inteligente a disposición del usuario. Los estudiantes que lo utilizaron destacaron especialmente la ventaja de poder hacer preguntas en cualquier momento. Tener dudas en algunas materias puede ser común y pedir ayuda puede ser incómodo. Los docentes descubrieron el gran número de preguntas que los alumnos evitaban hacer en clase y utilizaron esta información como oportunidad para abordar los conceptos de mayor dificultad en más detalle elaborando

planes de estudios más personalizados. Según un estudio reciente, los alumnos de escuelas tradicionales que utilizan este método durante 18 horas al año mejoran su nivel de conocimientos del 30% al 50%, en comparación con otros estudiantes [20].

3.3.2 Machine Learning

El Machine Learning (ML) está transformando el ámbito educativo, proporcionando herramientas avanzadas para personalizar y optimizar el aprendizaje de los estudiantes. Numerosos estudios y aplicaciones prácticas demuestran cómo el ML puede mejorar significativamente los resultados educativos y la experiencia de aprendizaje.

Un estudio realizado por la Universidad de Stanford demostró que los algoritmos de ML pueden predecir con mayor precisión el rendimiento de los estudiantes en comparación con métodos tradicionales. Utilizando algoritmos como Random Forests y Support Vector Machines, los investigadores lograron identificar patrones de comportamiento y rendimiento académico que permitieron desarrollar intervenciones más efectivas para mejorar el desempeño de los estudiantes [22].

Además, los algoritmos de ML se han utilizado para predecir la deserción escolar con alta precisión. En un estudio, se encontró que los algoritmos de ML superaron a las predicciones de los profesores en la identificación de estudiantes en riesgo de abandono escolar a medida que avanzaba el año académico. Este enfoque combinó datos académicos y comportamentales para ofrecer una predicción más completa y precisa, lo que permitió a los educadores intervenir de manera oportuna [?].

El uso de técnicas de ML, como el análisis de regresión y las redes neuronales, también ha sido efectivo en la personalización del aprendizaje. Por ejemplo, un estudio sobre plataformas de aprendizaje adaptativo mostró que los sistemas basados en ML podían ajustar el contenido educativo en tiempo real según las necesidades individuales de los estudiantes, mejorando así la retención de conocimientos y el compromiso del estudiante [15].

Por último, también se ha aplicado en el análisis de datos educativos a gran escala para identificar patrones de aprendizaje y áreas problemáticas. Un ejemplo notable es el uso de modelos de aprendizaje profundo para analizar el comportamiento y rendimiento de los estudiantes, lo que permite a los educadores adaptar sus métodos de enseñanza y ofrecer apoyo específico donde más se necesita [16].

Estos estudios y aplicaciones demuestran el enorme potencial del Machine Learning para mejorar la educación, ofreciendo herramientas poderosas para personalizar el aprendizaje, predecir resultados y desarrollar intervenciones efectivas que beneficien tanto a estudiantes como a educadores.

3.4 Análisis del Modelo de Duolingo

Duolingo, la plataforma web líder en el aprendizaje de idiomas, ha revolucionado la enseñanza de lenguas al adoptar un enfoque innovador y altamente práctico. El éxito de Duolingo se basa en su enfoque en la comprensión intuitiva, eliminando la necesidad de lecciones extensas de gramática para completar un curso en un idioma. Esta singularidad radica en su capacidad para ayudar a los usuarios a entender el significado de oraciones de manera natural, con la ayuda de imágenes, una pestaña de traducción y correcciones automáticas. Este enfoque ha demostrado ser enormemente eficaz para motivar a los estudiantes y hacer que el aprendizaje sea más accesible.

3.4.1 Gamificación en Duolingo

Duolingo utiliza la gamificación para transformar el aprendizaje de idiomas en algo atractivo y efectivo, incorporando puntos de experiencia (XP), ligas, insignias, rachas diarias, corazones y monedas virtuales (lingots/gemas).

Estos elementos motivan a los usuarios mediante recompensas intrínsecas y extrínsecas, fomentando el compromiso continuo y la práctica diaria, lo cual es crucial para la retención del idioma. Es la combinación de competencia, desafío y feedback positivo lo que consigue transformar el estudio en una experiencia lúdica y para muchos adictiva, contribuyendo al éxito de la plataforma.



Figure 3.6: Elementos de la gamificación en Duolingo

CHAPTER 3. ESTADO DEL ARTE

Table 3.1: Elementos de gamificación en Duolingo

Elemento	Descripción	Aportación al usuario
Puntos y Niveles	La ganancia de XP se logra al completar lecciones y permite al usuario avanzar de nivel.	Sensación de logro y reflejo del progreso educativo.
Ligas y Clasificaciones	Se realizan competiciones semanales a través de las cuáles se accede a una clasificación entre usuarios enlazados.	Motivación adicional a través de la competencia amistosa.
Insignias y Trofeos	Obtención de insignias al alcanzar hitos concretos.	Recompensas visibles que motivan y reconocen el progreso del usuario.
Rachas Diarias	Permite al usuario mantener una racha de estudio diario.	Fomenta la constancia y la práctica regular.
Llingots/Gemas	Monedas virtuales que se usan en la tienda	Se utilizan para recompensar a las personas por diversos logros y así comprar atuendos para la mascota.
Vidas	Intentos disponibles del usuario para realizar un nivel	Fomentar la precisión y evitar errores durante el aprendizaje.

3.4.2 Personalización del Aprendizaje

El modelo de Duolingo se centra en un enfoque de aprendizaje uno a uno (1:1), lo que permite a los estudiantes aprender idiomas de manera autónoma. La plataforma adapta constantemente la experiencia de aprendizaje para cada alumno, lo que se traduce en una enseñanza altamente personalizada. Esta adaptación se perfecciona continuamente a través de la recopilación y el análisis de datos de los usuarios, lo que optimiza aún más el proceso de enseñanza.

Una característica clave de Duolingo es su seguimiento de las estadísticas de todas las palabras enseñadas explica Burr Settles Director of Research en Duolingo en su artículo *"How we learn how you learn"* [14], lo que incluye la frecuencia con la que se presentan y cómo de bien las recuerdas. Utilizan estas estadísticas para predecir la probabilidad de que recuerdes una palabra en cualquier momento, lo que contribuye a una experiencia de aprendizaje más efectiva y eficiente.

Duolingo tiene una base de millones de estudiantes en todo el mundo que generan miles de millones de estadísticas sobre el aprendizaje de idiomas cada día. Esto les ha permitido combinar técnicas de Machine Learning y Big Data con la teoría

CHAPTER 3. ESTADO DEL ARTE

psicolingüística de las curvas de olvido, lo que ha resultado en la creación de un innovador modelo estadístico conocido como regresión de semivida (HLR) [4]. Este modelo se inspira en otros métodos utilizados en el ámbito del "Big Data", como la regresión logística, pero utiliza una función de probabilidad exponencial para estimar la semivida de una palabra en la memoria del estudiante extranjero.

Chapter 4

Tecnología y Herramientas Utilizadas

4.1 Lenguajes de Programación

4.1.1 Flutter y Dart

Para el desarrollo del proyecto CodeLingo, se ha optado por utilizar Flutter y Dart, dos tecnologías que han demostrado ser efectivas en el campo del desarrollo de aplicaciones multiplataforma. Esta elección se fundamenta en varios motivos que abarcan desde la eficiencia y la velocidad de desarrollo hasta el respaldo de la comunidad de desarrolladores y el apoyo de Google.

En primer lugar, Flutter, lanzado por Google en mayo de 2017, se trata de un SDK (Software Development Kit) de código abierto diseñado para desarrollar aplicaciones multiplataforma de alta calidad para iOS, Android, web y escritorio desde una única base de código. Gracias a esta característica, ha ganado una vasta adopción, ya que no solo reduce el tiempo y los costos de desarrollo, sino que también asegura una mayor coherencia y calidad en las aplicaciones producidas. Un ejemplo notable del éxito de Flutter es la aplicación de Google Ads, que ha sido reconstruida utilizando esta tecnología, proporcionando una experiencia de usuario fluida y consistente en todas las plataformas.

Además, muchas empresas hoy en día optan por Flutter para desarrollar sus aplicaciones, como es el caso de BMW. En octubre de 2019, un grupo de ingenieros de BMW decidió avanzar con Flutter, reconociendo que eliminar la dependencia de plataformas específicas permitiría una estructura organizativa más flexible y centrada en la innovación. En menos de un año, lanzaron la aplicación My BMW, que se ha establecido en 47 países como una interfaz universal para los usuarios, mejorando significativamente la experiencia del usuario. Este caso ilustra cómo Flutter permite a las empresas crear aplicaciones con un diseño UI/UX de alta calidad de manera eficiente [1].



Figure 4.1: My BMW

Dart, el lenguaje de programación subyacente a Flutter, también ha sido un factor clave en esta decisión. Lanzado en 2011 por Google, Dart se creó para mejorar la velocidad del desarrollo y la creación de scripts. Con su sintaxis clara y fácil de aprender, Dart facilita a los desarrolladores escribir código eficiente y mantenible. Desde su lanzamiento, Dart se ha convertido en el "ingrediente secreto" de Flutter, ayudando a los desarrolladores a diseñar el look and feel de muchas aplicaciones móviles y web.

La popularidad de Dart ha despegado desde entonces. En el "State of the Octoverse 2019" de GitHub, Dart fue nombrado el lenguaje de programación de más rápido crecimiento, con un incremento del 532% en su adopción entre la comunidad de desarrolladores [7]. Este crecimiento se debe en gran parte a la capacidad de Flutter y Dart para permitir a los desarrolladores crear aplicaciones más rápidas, útiles y elegantes, facilitando la colaboración en tiempo real entre diseñadores y desarrolladores.

La imagen adjunta 4.3 ilustra de manera clara las capacidades de Dart tanto como herramienta de desarrollo como de producción. En la fase de desarrollo, Dart ofrece una compilación incremental rápida y la funcionalidad de stateful "Hot Reload", permitiendo a los desarrolladores ver los cambios de código en tiempo real sin perder el estado de la aplicación. En producción, Dart se destaca por su rápida salida nativa y su tiempo de ejecución reducido, lo que resulta en aplicaciones más rápidas y eficientes. Además, Dart puede compilarse para múltiples arquitecturas y plataformas, incluyendo ARM32, ARM64, Intel x86_64 y JavaScript, lo que permite que las aplicaciones tengan un rendimiento cercano al nativo, aprovechando al máximo las capacidades de hardware de los dispositivos.

Para lograr la multiplataforma, Flutter permite escribir una sola vez el código y ejecutarlo en múltiples plataformas, eliminando la necesidad de mantener dos bases de código separadas para iOS y Android. Esto se logra mediante el uso de "widgets" de Flutter, que son componentes visuales altamente personalizables y reutilizables.

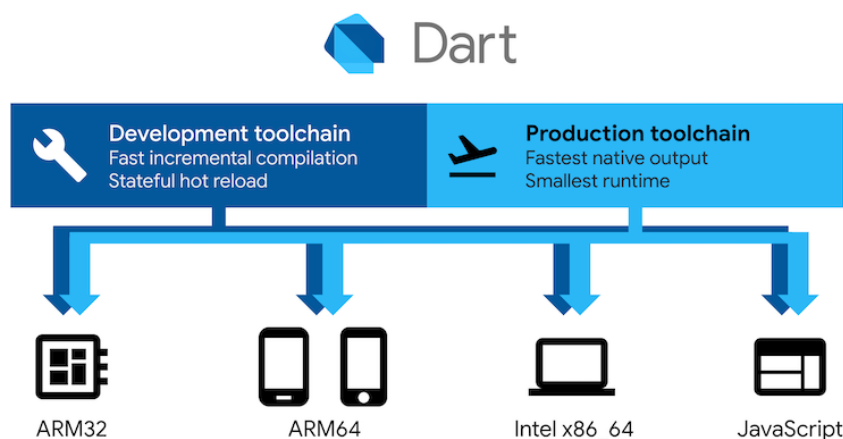


Figure 4.2: Plataforma Dart

Una de las principales ventajas del desarrollo multiplataforma es la eficiencia en el desarrollo. Con una única base de código, los desarrolladores pueden crear aplicaciones para ambas plataformas, reduciendo significativamente el tiempo y el esfuerzo requeridos. Mantener una sola base de código es mucho más sencillo y económico que hacerlo para dos bases de código separadas. Flutter permite crear una experiencia de usuario consistente en todas las plataformas, utilizando los mismos widgets y lógica de UI. Esto asegura que la interfaz y la funcionalidad sean uniformes, proporcionando una experiencia cohesiva a los usuarios. Desde el punto de vista económico, el desarrollo multiplataforma puede reducir considerablemente los costos, ya que se necesita un menor número de desarrolladores y el tiempo de desarrollo es más corto. Flutter cuenta con una creciente comunidad de desarrolladores y una amplia gama de paquetes y plugins que facilitan la integración de funcionalidades comunes como autenticación, bases de datos y redes.

Hoy en día, grandes empresas utilizan Flutter y Dart para crear sus propias aplicaciones. Este dúo ha tenido un éxito notable en China, con gigantes como Alibaba, Baidu, ByteDance y Tencent adoptando estas tecnologías. Actualmente, Google Play, la tienda de aplicaciones más popular de Android, cuenta con 125,000 aplicaciones construidas con Dart y Flutter, según datos de la propia Google.

El desarrollo nativo ofrece un rendimiento óptimo, ya que las aplicaciones están optimizadas para aprovechar al máximo las capacidades del hardware del dispositivo. Esto es crucial para aplicaciones de alto rendimiento como juegos o aplicaciones que manejan gráficos intensivos. Los desarrolladores nativos tienen acceso inmediato a las APIs más recientes y avanzadas del sistema operativo, lo que permite una integración más completa y funcionalidades más avanzadas. Las aplicaciones nativas pueden adaptarse mejor a las guías de diseño de cada plataforma, ofreciendo una experiencia más familiar y consistente para los usuarios de iOS y Android.

Sin embargo, también tiene sus desventajas. El desarrollo de aplicaciones nativas requiere escribir y mantener dos bases de código separadas, una para iOS y otra para

CHAPTER 4. TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

Android, lo que aumenta el tiempo de desarrollo y los costos. Las actualizaciones y el mantenimiento deben realizarse por separado para cada plataforma, lo que puede ser engorroso y propenso a errores. A menudo se necesitan equipos de desarrollo más grandes con habilidades especializadas en cada plataforma, lo que puede ser más costoso y difícil de gestionar.

En conclusión, al optar entre el desarrollo nativo y el multiplataforma, es esencial considerar las necesidades específicas del proyecto, incluyendo el rendimiento deseado, las funcionalidades del dispositivo y los recursos disponibles. El enfoque más adecuado dependerá de los objetivos y las restricciones del proyecto, y a menudo, una combinación de ambos enfoques puede ser la solución óptima. La elección de Flutter y Dart para el proyecto CodeLingo está respaldada por su eficiencia, versatilidad y el apoyo de una comunidad en crecimiento y de Google. Estas tecnologías no solo facilitan un desarrollo rápido y de alta calidad, sino que también aseguran que CodeLingo pueda ofrecer una experiencia de usuario excepcional y uniforme en todas las plataformas. Adoptar Flutter y Dart permite crear una herramienta educativa y accesible para todos los usuarios interesados en aprender lenguajes de programación.

4.1.2 Python

Para el desarrollo de Codelingo se ha optado también por utilizar Python para el procesamiento de datos y la implementación de servicios de backend.

Python proporciona una robusta colección de bibliotecas, destacándose pandas, que facilitan el tratamiento y la limpieza de los datos. Dicha capacidad ha sido esencial a la hora de preparar los datasets obtenidos de Kaggle, asegurando que los datos estuvieran en el formato adecuado y limpio para su uso en el modelo de recomendación. Las operaciones complejas como la eliminación de valores nulos, la normalización de datos y la transformación de estructuras de datos se han simplificado enormemente gracias a esta biblioteca.

También se ha hecho uso de Flask, un microframework de Python que permite la creación de aplicaciones web de manera rápida y con un diseño sencillo en la fase de desarrollo. Su flexibilidad ha permitido construir un servicio de recomendación de cursos que se integra con facilidad con el frontend de la aplicación.

En este proyecto se ha utilizado también scikit-learn para aplicar técnicas de procesamiento de texto y calcular similitudes entre descripciones de cursos y preferencias de usuarios.



Figure 4.3: Bibliotecas utilizadas en el proyecto

La comunidad de Python y la abundancia de recursos disponibles han sido cruciales durante el desarrollo del proyecto. La documentación y los tutoriales en línea han facilitado la implementación de soluciones eficientes. Además, la versatilidad de Python y su capacidad para manejar tareas de procesamiento de datos y desarrollo backend, gracias a su ecosistema rico en bibliotecas y su accesibilidad, han sido esenciales para la realización del proyecto.

4.2 Diseño de Base de Datos en Firebase

Firebase ha sido utilizado en este proyecto debido a su integración fluida y eficiente con Flutter, permitiendo el desarrollo de aplicaciones para iOS, Android y web de manera sencilla. La integración de Firebase se realizó mediante la CLI de Firebase, que facilita la configuración y el despliegue de servicios en la nube.

Firebase proporciona una amplia gama de servicios como bases de datos en tiempo real, autenticación de usuarios, almacenamiento en la nube y notificaciones push, entre otros como los que se detallan en la figura 4.4. Estos servicios permiten desarrollar aplicaciones robustas sin necesidad de gestionar infraestructura compleja. En este proyecto, Firebase se ha utilizado principalmente para la autenticación de usuarios y el almacenamiento de datos.

La configuración inicial de Firebase en el proyecto incluyó la creación de un proyecto en la consola de Firebase, seguido de la adición de aplicaciones para las plataformas iOS, Android y web. Se generaron y descargaron archivos de configuración específicos para cada plataforma, como ‘google-services.json’ para Android y ‘GoogleService-Info.plist’ para iOS, que luego se incluyeron en el proyecto Flutter.

La CLI de Firebase facilita la gestión de estos archivos y la implementación de servicios, permitiendo realizar operaciones como la inicialización del proyecto, la implementación de funciones en la nube y la configuración de la autenticación de usuarios con comandos sencillos. Esta herramienta ha sido fundamental para integrar rápidamente las capacidades de Firebase en la aplicación CodeLingo.

La facilidad de integración de Firebase con Flutter, junto con su amplia gama de servicios, ha permitido crear una aplicación robusta y escalable.



Figure 4.4: Funcionalidades Firebase

4.3 Visual Studio Code

En el desarrollo de CodeLingo, se ha optado por utilizar Visual Studio Code (VS Code) como el entorno de desarrollo principal. Esta elección se debe a varias razones que incluyen la eficiencia, la flexibilidad y la amplia gama de extensiones disponibles que mejoran significativamente la experiencia de desarrollo. A continuación, se detallan las razones específicas y las extensiones utilizadas:

Visual Studio Code es un editor de código fuente gratuito, liviano y altamente personalizable, desarrollado por Microsoft. A continuación, se enumeran algunas de las principales razones por las que se eligió VS Code para este proyecto:

- **Selección de dispositivos para la ejecución:** VS Code permite seleccionar y configurar fácilmente el tipo de dispositivo que se utilizará durante la ejecución del código, lo que es especialmente útil para probar aplicaciones en diferentes entornos.
- **Reconocimiento y popularidad:** VS Code es uno de los editores de código más reconocidos y utilizados a nivel mundial, lo que garantiza un amplio soporte de la comunidad y abundantes recursos de aprendizaje.

CHAPTER 4. TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

- **Personalización:** VS Code es altamente personalizable, permitiendo a los desarrolladores ajustar la interfaz y las funcionalidades según sus necesidades específicas.
- **Integración de múltiples lenguajes de programación:** VS Code admite una amplia variedad de lenguajes de programación y ofrece soporte específico para Dart y Flutter, esenciales para el desarrollo de CodeLingo.

Para maximizar la eficiencia y productividad en el desarrollo de CodeLingo, se han utilizado varias extensiones de VS Code que ofrecen funcionalidades específicas para el desarrollo con Flutter y Dart:

- **Awesome Flutter Snippets:** Esta extensión proporciona snippets de código útiles para Flutter, lo que permite a los desarrolladores escribir código más rápidamente y con menos errores.
- **Pubspec Assist:** Facilita la gestión del archivo `pubspec.yaml`, permitiendo agregar, actualizar y eliminar dependencias de manera sencilla.
- **Image Preview:** Permite previsualizar imágenes directamente en VS Code, lo que es especialmente útil cuando se trabaja con recursos gráficos en el desarrollo de la interfaz de usuario.
- **Error Lens:** Mejora la visibilidad de los errores y advertencias en el código al mostrarlos directamente en la línea correspondiente, lo que facilita la identificación y corrección de problemas.
- **Material Icon Theme:** Proporciona un conjunto de iconos atractivos y fácilmente reconocibles para diferentes tipos de archivos y carpetas, mejorando la organización y navegación del proyecto.

En resumen, Visual Studio Code se ha convertido en una herramienta esencial para el desarrollo de CodeLingo, no solo por su versatilidad y rendimiento, sino también por las poderosas extensiones que permiten una experiencia de desarrollo más fluida y eficiente.

4.4 Android Studio

Android Studio ha sido una herramienta crucial en el desarrollo de CodeLingo, principalmente debido a su capacidad para emular diferentes dispositivos móviles. Esta funcionalidad ha permitido probar y depurar la aplicación en múltiples entornos, asegurando su correcta ejecución en una amplia variedad de dispositivos.

En la Figura [4.5](#) se pueden observar los emuladores utilizados durante el desarrollo, incluyendo modelos como Pixel 3a y Note-10. Estos emuladores proporcionan

CHAPTER 4. TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

un entorno controlado para probar la aplicación en diferentes versiones de Android y dispositivos con diversas configuraciones de hardware.

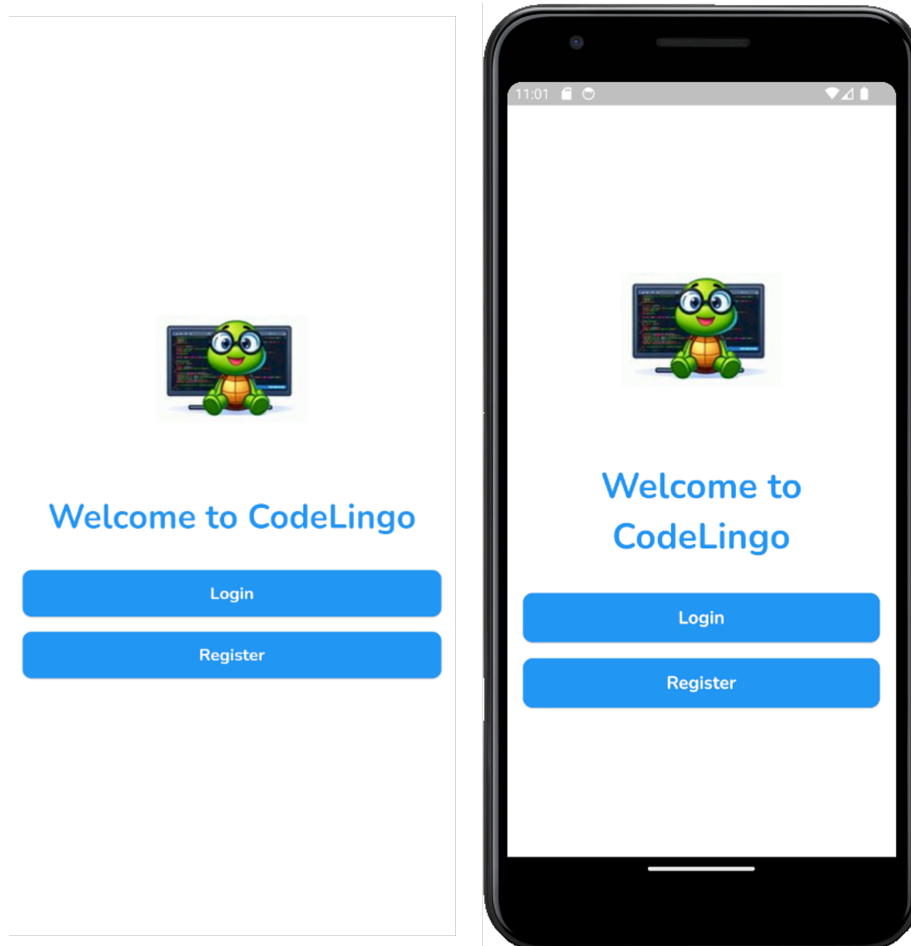


Figure 4.5: Visualización de CodeLingo en diferentes Emuladores Android

Android Studio ofrece varias ventajas que han sido esenciales para el desarrollo de CodeLingo:

- **Emulación de Dispositivos:** La capacidad de emular una amplia gama de dispositivos ha permitido probar la aplicación en condiciones variadas, identificando y resolviendo problemas específicos de cada tipo de dispositivo.
- **Integración con Herramientas de Desarrollo:** Android Studio se integra perfectamente con herramientas de desarrollo como Gradle, lo que facilita la gestión de dependencias y el proceso de construcción de la aplicación.
- **Depuración Avanzada:** Las herramientas de depuración de Android Studio han permitido un análisis detallado del comportamiento de la aplicación, facilitando la identificación y corrección de errores.

CHAPTER 4. TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

- **Perfilado de Rendimiento:** Las herramientas de perfilado han sido útiles para optimizar el rendimiento de la aplicación, asegurando una experiencia de usuario fluida y eficiente.

Las sensaciones al utilizar Android Studio han sido muy positivas. Su entorno integrado ofrece una experiencia de desarrollo completa y coherente, con todas las herramientas necesarias en un solo lugar. La emulación precisa de dispositivos ha sido particularmente útil para asegurar que CodeLingo funcione correctamente en una amplia variedad de escenarios.

En resumen, Android Studio ha sido una herramienta fundamental para el desarrollo de CodeLingo, proporcionando un entorno robusto y completo para la creación, prueba y optimización de la aplicación en diferentes dispositivos Android.

4.5 Github

GitHub ha sido una herramienta fundamental en el desarrollo de CodeLingo por diversas razones. A continuación, se detallan las principales utilidades y aplicaciones de GitHub en el proyecto:

GitHub es una plataforma de desarrollo colaborativo de software que permite alojar proyectos utilizando el sistema de control de versiones Git. Entre sus principales ventajas se encuentran:

- **Colaboración y Aprendizaje:** GitHub facilita la colaboración entre desarrolladores, permitiendo compartir y trabajar en proyectos de manera conjunta. Además, es una fuente valiosa de aprendizaje, ya que alberga una vasta cantidad de proyectos de código abierto que pueden ser estudiados y utilizados como referencia.
- **Repositorio Centralizado:** GitHub actúa como un repositorio centralizado donde se puede almacenar el código del proyecto, realizar seguimiento de cambios, y gestionar versiones de manera eficiente.
- **Acceso a Proyectos Externos:** La plataforma permite acceder a proyectos desarrollados por otros usuarios, lo que ha sido de gran ayuda para encontrar soluciones y ejemplos prácticos. Esto ha sido particularmente útil para el desarrollo de funcionalidades específicas en CodeLingo.

Para el desarrollo de CodeLingo, se utilizó GitHub no solo para almacenar y gestionar el código del proyecto, sino también para encontrar y utilizar proyectos de terceros que aportaron funcionalidades adicionales. Un ejemplo destacado es el uso de un repositorio que realiza web scraping de ejercicios de programación disponibles en internet. Este repositorio facilitó la recopilación de ejercicios que se integraron en la plataforma CodeLingo.

CHAPTER 4. TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

El repositorio en cuestión, disponible en: <https://github.com/falcon883/Sanfoundry-MCQ-Saver> realiza web scraping de una serie de ejercicios de programación de diferentes sitios web. Funciona mediante la ejecución de scripts que acceden a las páginas web, extraen los ejercicios y los formatean para ser utilizados en la plataforma.

El uso de este repositorio permitió no solo la ampliación del banco de ejercicios disponibles en CodeLingo, sino también la automatización del proceso de recopilación de datos, ahorrando tiempo y recursos valiosos durante el desarrollo del proyecto.

En resumen, GitHub ha sido una herramienta esencial para el desarrollo de CodeLingo, proporcionando un entorno colaborativo y centralizado para la gestión del código, facilitando el aprendizaje y la integración de proyectos externos que han enriquecido la funcionalidad de la plataforma.

4.5.1 Heroku

Heroku ha sido una herramienta clave en el despliegue y la gestión del proyecto CodeLingo. Se trata de una plataforma como servicio (PaaS) de la empresa Salesforce que permite a los desarrolladores construir, ejecutar y operar aplicaciones completamente en la nube. La elección de Heroku se ha basado en su facilidad de uso, su integración con Git y su capacidad para manejar aplicaciones web modernas con requisitos diversos.

Heroku ofrece un entorno gestionado que permite centrarse en el desarrollo del código en lugar de preocuparse por la infraestructura subyacente. Utiliza contenedores ligeros conocidos como dynos para ejecutar aplicaciones, lo que facilita escalar los servicios según la demanda. En el contexto de CodeLingo, Heroku ha sido utilizado para desplegar el backend basado en Python y Flask, que proporciona recomendaciones de cursos basadas en las preferencias del usuario.

El proceso de despliegue en Heroku comienza con la creación de una aplicación en la plataforma a través de la interfaz web o utilizando la línea de comandos de Heroku (Heroku CLI). La integración con Git permite que el despliegue de nuevas versiones de la aplicación sea tan sencillo como hacer un push a un repositorio remoto. Cada push desencadena automáticamente un proceso de construcción y despliegue en Heroku, asegurando que la aplicación esté siempre actualizada.

Heroku proporciona una amplia variedad de add-ons y servicios que pueden ser integrados fácilmente en las aplicaciones. Estos incluyen herramientas de monitoreo, servicios de almacenamiento en caché y mucho más. En el caso de CodeLingo, se ha utilizado principalmente para gestionar el despliegue del backend y asegurar que esté disponible y funcionando correctamente.

Además, Heroku ofrece capacidades de escalado automático, lo que significa que la aplicación puede manejar un aumento repentino en el tráfico sin necesidad de intervención manual. Esto es particularmente útil para aplicaciones que pueden

CHAPTER 4. TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

experimentar variaciones en la carga de trabajo, garantizando así un rendimiento constante y una alta disponibilidad.

El uso de Heroku ha proporcionado varias lecciones valiosas. En primer lugar, se ha aprendido a configurar y gestionar entornos de despliegue en la nube, una habilidad esencial en el desarrollo moderno de software. También ha permitido comprender la importancia de la integración continua y el despliegue continuo (CI/CD), ya que Heroku facilita estas prácticas mediante su integración con Git y otras herramientas de desarrollo.

Finalmente, Heroku ha demostrado ser una plataforma fiable y escalable, adecuada para aplicaciones de todos los tamaños. La experiencia adquirida en el uso de Heroku no solo ha mejorado la capacidad para desplegar y gestionar aplicaciones en la nube, sino que también ha proporcionado una base sólida para el desarrollo de aplicaciones web robustas y escalables en el futuro. La facilidad de uso y la potencia de Heroku hacen de esta plataforma una elección acertada para el despliegue de CodeLingo y para futuros proyectos de desarrollo de software.

4.6 Paquetes del proyecto

En el desarrollo de CodeLingo, se ha utilizado una variedad de packages de Flutter para agregar funcionalidades y mejorar la eficiencia del desarrollo. Estos packages son desarrollados y mantenidos por la comunidad de Flutter y están disponibles a través de pub.dev. A continuación, se describen los principales packages utilizados y sus funciones específicas:

Flutter es un framework de desarrollo de aplicaciones multiplataforma que permite crear aplicaciones nativas para iOS y Android con una única base de código. Los packages en Flutter son bibliotecas adicionales que extienden las capacidades del framework, proporcionando componentes preconstruidos y funcionalidades adicionales que pueden ser fácilmente integradas en un proyecto. Para importar estos packages, se debe especificar en el archivo `pubspec.yaml` y luego ejecutar el comando `flutter pub get` para descargar e integrar los packages en el proyecto.

- **cupertino_icons: ^1.0.6:** Proporciona los iconos del iOS, que son esenciales para crear interfaces de usuario con apariencia nativa en dispositivos Apple.
- **flutter_highlight: ^0.7.0:** Utilizado para resaltar sintaxis de código en diferentes lenguajes de programación, esencial para una aplicación de aprendizaje de programación.
- **google_fonts: ^6.2.1:** Permite utilizar fuentes de Google en la aplicación, mejorando la estética y la personalización de la tipografía.
- **carousel_slider: ^4.0.0:** Proporciona un componente de carrusel para

CHAPTER 4. TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

deslizar entre diferentes elementos, útil para mostrar tutoriales o características destacadas.

- **flutter_rating_bar: ^4.0.0**: Añade una barra de calificación interactiva, utilizada para permitir a los usuarios calificar lecciones o ejercicios.
- **charts_flutter: ^0.12.0** y **syncfusion_flutter_charts: ^21.2.4**: Ambos packages proporcionan herramientas para crear gráficos y visualizaciones de datos.
- **provider: ^6.1.2**: Una solución para la gestión del estado en Flutter, permitiendo una arquitectura más limpia y eficiente.
- **flutter_dotenv: ^5.1.0**: Facilita la gestión de variables de entorno, esenciales para manejar configuraciones sensibles como claves de API.
- **http: ^1.2.1**: Un package para realizar solicitudes HTTP, utilizado para interactuar con APIs y servicios web.
- **webview_flutter: ^4.0.0**: Permite integrar vistas web dentro de la aplicación, útil para mostrar contenido web sin salir de la aplicación.
- **youtube_player_flutter: ^8.0.0**: Proporciona un reproductor de YouTube integrado, esencial para mostrar videos educativos directamente en la aplicación.
- **fl_chart: ^0.68.0**: Otro package para crear gráficos y visualizaciones de datos.
- **csv: ^6.0.0** y **excel: ^2.0.7**: Utilizados para la manipulación y gestión de archivos CSV y Excel, respectivamente.
- **url_launcher: ^6.1.7**: Permite abrir URLs en el navegador, hacer llamadas telefónicas, enviar correos electrónicos, etc.
- **font_awesome_flutter: ^10.1.0**: Proporciona acceso a los iconos de FontAwesome, ampliando las opciones de iconografía en la aplicación.
- **flutter_staggered_grid_view: ^0.4.0**: Añade un layout de grid escalonado, útil para crear diseños de interfaz más dinámicos.
- **flutter_animate: ^3.0.0**: Proporciona herramientas para animaciones en Flutter, mejorando la interacción y la experiencia del usuario.
- **firebase_core: ^3.1.1**, **firebase_auth: ^5.1.1** y **cloud_firestore: ^5.0.2**: Packages esenciales para la integración con Firebase, utilizado para autenticación de usuarios y almacenamiento de datos en la nube.

CHAPTER 4. TECNOLOGÍA Y HERRAMIENTAS UTILIZADAS

- **async:** ^2.8.2: Proporciona utilidades adicionales para la programación asincrónica en Dart.

Estos packages han sido integrados en el proyecto para enriquecer la funcionalidad de CodeLingo, simplificar el desarrollo y mejorar la experiencia del usuario. La comunidad de Flutter continúa desarrollando y manteniendo estos packages, lo que permite a los desarrolladores aprovechar las últimas innovaciones y mejoras en sus proyectos.

Chapter 5

Sistema Desarrollado

5.1 Diseño del Desarrollo

5.1.1 Esquema y Arquitectura del Sistema

La arquitectura del sistema CodeLingo está diseñada para proporcionar una experiencia de usuario fluida y eficiente, integrando diversos servicios backend para manejar autenticación, almacenamiento de datos y recomendaciones personalizadas. La estructura del sistema se divide en varios componentes principales, cada uno con un rol específico en el funcionamiento de la plataforma. En el esquema de la figura 5.1 se puede comprobar que la arquitectura se basa en un enfoque de microservicios, lo que permite una gestión más eficiente de los recursos y facilita el mantenimiento y la actualización del sistema.

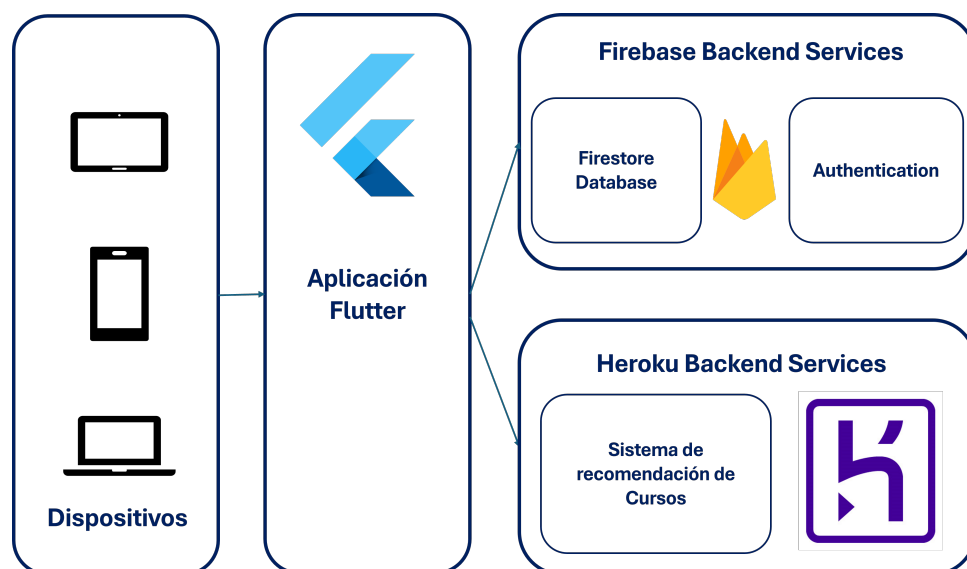


Figure 5.1: Arquitectura de Codelingo

CHAPTER 5. SISTEMA DESARROLLADO

Los componentes principales del sistema son:

- **Dispositivos (iOS/Android/Web):** Los usuarios acceden a CodeLingo a través de aplicaciones móviles y web. Estas aplicaciones están desarrolladas utilizando Flutter, un framework de desarrollo de aplicaciones multiplataforma de Google. Flutter permite compartir una única base de código para múltiples plataformas, lo que reduce significativamente el tiempo y el esfuerzo de desarrollo. La interfaz de usuario (UI) y la experiencia de usuario (UX) se diseñan para ser consistentes y receptivas en todos los dispositivos.
- **Aplicación Flutter:** La aplicación Flutter actúa como el cliente principal que interactúa con los servicios backend. Se encarga de varias funciones críticas:
 - **Autenticación de usuarios:** Utiliza Firebase Authentication para gestionar el registro e inicio de sesión de los usuarios. Los métodos de autenticación incluyen correo electrónico y contraseña, Google, Facebook y otros proveedores de identidad.
 - **Visualización de contenido:** Muestra preguntas, cursos, chats y otros contenidos educativos a los usuarios. La UI se adapta dinámicamente a los datos recibidos de los servicios backend.
 - **Comunicación con servicios backend:** Realiza llamadas API y operaciones de base de datos para interactuar con Firebase y Heroku. Maneja la lógica de negocio localmente antes de enviar o recibir datos de los servicios remotos.
- **Firebase Backend Services:** Firebase proporciona una infraestructura backend robusta y escalable. Incluye varios servicios críticos:
 - **Cloud Firestore Database:** Una base de datos NoSQL en tiempo real que almacena y sincroniza datos de la aplicación. Esta base de datos es altamente escalable y permite actualizaciones y lecturas en tiempo real, lo que es esencial para funciones como el chat y la colaboración en vivo. Se utiliza para almacenar información del usuario, preguntas, respuestas, datos de chat y otra información relevante.
 - **Authentication:** Un servicio que gestiona la autenticación de usuarios de manera segura. Proporciona varias opciones de autenticación, incluyendo autenticación por correo electrónico y contraseña, y autenticación mediante proveedores externos como Google y Facebook. Este servicio asegura que solo los usuarios autorizados puedan acceder a la aplicación y a sus datos.
- **Heroku Backend Services:** Heroku se utiliza para desplegar servicios adicionales que no están cubiertos por Firebase, específicamente:

CHAPTER 5. SISTEMA DESARROLLADO

- **Sistema de Recomendación de Cursos:** Este servicio se encarga de recibir las preferencias del usuario desde la aplicación Flutter, procesar la información y devolver recomendaciones de cursos personalizadas. Está implementado en Heroku debido a su flexibilidad para desplegar y escalar aplicaciones basadas en contenedores. Utiliza algoritmos de machine learning para analizar las preferencias y comportamientos del usuario y generar recomendaciones precisas.

El flujo de trabajo dentro de la arquitectura del sistema se describe a continuación:

1. **Autenticación de Usuario:** Los usuarios se registran o inician sesión en la aplicación Flutter. La autenticación se maneja mediante Firebase Authentication, que verifica las credenciales del usuario y proporciona tokens de acceso seguros. Este proceso asegura que solo los usuarios autenticados puedan acceder a los datos y funcionalidades de la aplicación.
2. **Acceso y Sincronización de Datos:** Una vez autenticados, la aplicación accede a la base de datos de Cloud Firestore para obtener y sincronizar datos en tiempo real. Esto incluye la información del perfil del usuario, preguntas, respuestas y datos de chat. La sincronización en tiempo real asegura que los datos estén siempre actualizados en todos los dispositivos conectados.
3. **Recomendación de Cursos:** Los usuarios pueden enviar sus preferencias de cursos a través de la aplicación Flutter. Estas preferencias son enviadas al servicio de recomendación en Heroku mediante una llamada API. Heroku procesa los datos utilizando algoritmos de machine learning y devuelve una lista de cursos recomendados, que luego se muestra en la aplicación.
4. **Interacción en Tiempo Real:** Las interacciones en tiempo real, como el chat entre usuarios, se gestionan mediante Cloud Firestore. Los mensajes de chat y otras actualizaciones se sincronizan instantáneamente en todos los dispositivos conectados, proporcionando una experiencia de usuario fluida y receptiva.

Se ha optado por esta distribución debido a la flexibilidad y escalabilidad que ofrece. La arquitectura de microservicios permite desarrollar, desplegar y escalar cada componente de manera independiente. Esto es crucial para manejar diferentes aspectos de la aplicación de manera eficiente y para facilitar el mantenimiento y la implementación de nuevas funcionalidades. La elección de Firebase y Heroku como proveedores de servicios backend se debe a su robustez, facilidad de integración y capacidad de escalar automáticamente según la demanda. Firebase proporciona una solución completa para la autenticación y el almacenamiento de datos en tiempo real,

CHAPTER 5. SISTEMA DESARROLLADO

mientras que Heroku permite desplegar servicios personalizados con alta flexibilidad y capacidad de integración.

En resumen, la arquitectura de CodeLingo está diseñada para ser robusta, escalable y segura, integrando múltiples servicios backend para proporcionar una experiencia de usuario rica y personalizada. Esta arquitectura permite a los desarrolladores centrarse en la creación de valor para el usuario final, mientras que los servicios gestionados se encargan de la infraestructura y el escalado automático.

5.1.2 Diagrama de Casos de Uso

A continuación se presenta el diagrama de casos de uso para el sistema CodeLingo, centrado en el actor usuario:

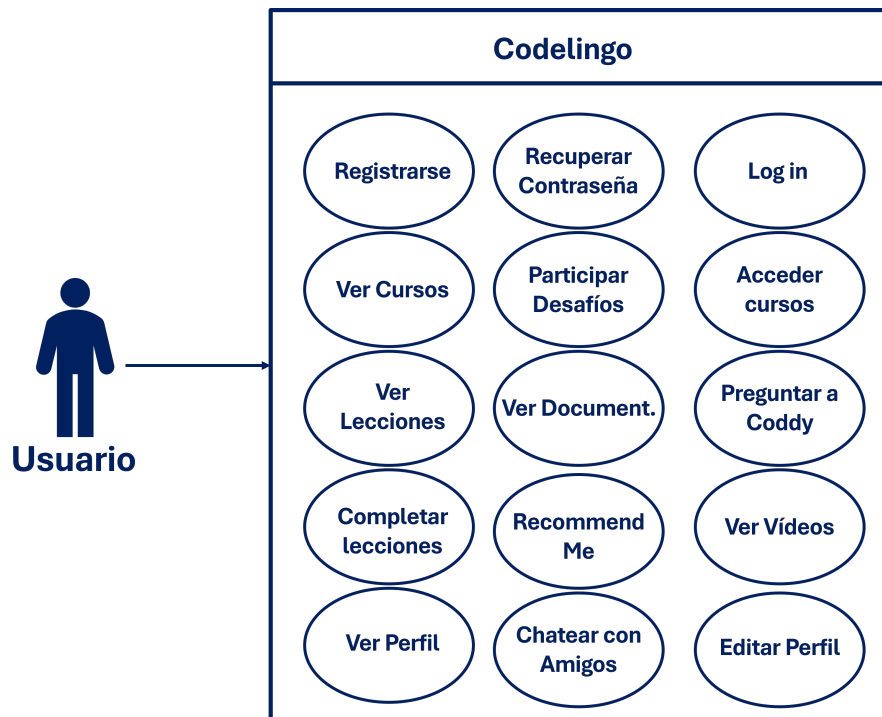


Figure 5.2: Diagrama de Casos de Uso de CodeLingo

- **Registrarse:** El usuario puede crear una nueva cuenta en la plataforma.
- **Recuperar Contraseña:** El usuario puede recuperar su contraseña en caso de olvido.
- **Log in:** El usuario puede iniciar sesión en una cuenta existente.
- **Ver Cursos:** El usuario puede ver una lista de cursos de programación disponibles.

CHAPTER 5. SISTEMA DESARROLLADO

- **Acceder cursos:** El usuario puede seleccionar y acceder a un curso específico.
- **Ver Lecciones:** El usuario puede ver las lecciones dentro de un curso seleccionado.
- **Ver Document.:** El usuario puede acceder a documentos relevantes relacionados con el curso.
- **Completar lecciones:** El usuario puede completar una lección, que incluye teoría, ejercicios prácticos y cuestionarios.
- **Participar Desafíos:** El usuario puede participar en desafíos de programación para poner a prueba sus habilidades.
- **Recommend Me:** El usuario recibe recomendaciones personalizadas basadas en su rendimiento.
- **Ver Vídeos:** El usuario puede ver videos educativos relacionados con el curso.
- **Ver Perfil:** El usuario puede ver su perfil personal y la información relacionada.
- **Editar Perfil:** El usuario puede editar la información de su perfil.
- **Chatear con Amigos:** El usuario puede interactuar con otros estudiantes a través del chat.
- **Preguntar a Coddy:** El usuario puede hacer preguntas a Coddy, un asistente virtual.

5.1.3 Requisitos Funcionales

Los requisitos funcionales son especificaciones detalladas que describen las capacidades y comportamientos necesarios que un sistema debe tener para cumplir sus objetivos y satisfacer las necesidades de los usuarios. En el contexto de una aplicación educativa como CodeLingo, estos requisitos aseguran que la plataforma proporcione una experiencia de usuario fluida, efectiva y eficiente. A continuación, se presentan los requisitos funcionales específicos de la aplicación CodeLingo.

Table 5.1: Requisitos Funcionales de CodeLingo

Requisito Funcional
Registro de Usuarios
Autenticación de Terceros
Visualización y Respuesta de Preguntas
Sincronización de Datos en Tiempo Real
Chat en Tiempo Real
Recomendación de Cursos Personalizados
Visualización de Videos Educativos
Calificación de Lecciones y Ejercicios
Procesamiento de Recomendaciones en Heroku

- **Registro de Usuarios:** La plataforma debe permitir a los usuarios registrarse utilizando su correo electrónico y una contraseña segura. Este requisito es fundamental para gestionar la identidad de los usuarios y proporcionarles acceso personalizado a la plataforma. La implementación incluye formularios de registro con validación de datos y almacenamiento seguro de la información del usuario en la base de datos.
- **Autenticación de Terceros:** CodeLingo debe ofrecer la opción de iniciar sesión a través de proveedores de autenticación de terceros como Google y Facebook. Este requisito facilita el proceso de inicio de sesión y mejora la experiencia del usuario al reducir el número de credenciales que deben recordar. La integración de OAuth 2.0 asegura que el proceso sea seguro y conforme a las mejores prácticas de autenticación.
- **Visualización y Respuesta de Preguntas:** La plataforma debe proporcionar una interfaz intuitiva que permita a los usuarios visualizar y responder preguntas de programación. Esta funcionalidad es crucial para fomentar la participación activa y el aprendizaje colaborativo. La interfaz debe incluir características como filtros de búsqueda, categorización de preguntas y un sistema de votación para destacar las mejores respuestas.
- **Sincronización de Datos en Tiempo Real:** Los datos del usuario, incluyendo su progreso y actividades, deben ser almacenados y sincronizados en tiempo real utilizando Cloud Firestore. Este requisito garantiza que los datos estén siempre actualizados y disponibles desde cualquier dispositivo, mejorando la continuidad y coherencia del aprendizaje.
- **Chat en Tiempo Real:** La aplicación debe permitir la comunicación en tiempo real entre los usuarios a través de un sistema de chat. Esta funcionalidad es esencial para facilitar la colaboración y el intercambio de conocimientos entre los usuarios. La implementación debe incluir características como

CHAPTER 5. SISTEMA DESARROLLADO

la gestión de contactos, notificaciones de mensajes nuevos y la posibilidad de crear grupos de chat.

- **Recomendación de Cursos Personalizados:** CodeLingo debe utilizar algoritmos de recomendación para sugerir cursos personalizados basados en las preferencias y el historial de aprendizaje del usuario. Esta funcionalidad aumenta la relevancia del contenido ofrecido y mejora la experiencia de aprendizaje. Los algoritmos deben considerar factores como el nivel de habilidad, intereses previos y el rendimiento en cursos anteriores.
- **Visualización de Videos Educativos:** La plataforma debe integrar la capacidad de ver videos educativos directamente desde YouTube. Los videos son un recurso educativo valioso que puede complementar los materiales de texto y ejercicios prácticos. La integración debe permitir la reproducción fluida de videos y ofrecer controles básicos como pausa, reproducción y ajuste de volumen.
- **Calificación de Lecciones y Ejercicios:** CodeLingo debe permitir a los usuarios calificar las lecciones y ejercicios completados. Esta funcionalidad proporciona feedback valioso a los creadores de contenido y ayuda a otros usuarios a identificar los recursos más útiles y efectivos. La implementación debe incluir un sistema de calificación intuitivo y fácil de usar, así como la posibilidad de dejar comentarios detallados.
- **Procesamiento de Recomendaciones en Heroku:** El sistema debe utilizar Heroku para procesar y devolver recomendaciones de cursos. Heroku, una plataforma en la nube, ofrece escalabilidad y rendimiento necesarios para manejar grandes volúmenes de datos y solicitudes en tiempo real. La integración con Heroku asegura que las recomendaciones sean generadas y entregadas de manera eficiente.

5.1.4 Requisitos No Funcionales

Los requisitos no funcionales son atributos que describen el funcionamiento del sistema desde una perspectiva de calidad. Estos requisitos aseguran que el sistema no solo cumple con las funcionalidades necesarias, sino que también ofrece un rendimiento, seguridad, disponibilidad y mantenibilidad adecuados. A continuación, se presentan los requisitos no funcionales específicos de la aplicación CodeLingo.

Table 5.2: Requisitos No Funcionales de CodeLingo

Requisito No Funcional
Escalabilidad
Seguridad
Disponibilidad
Rendimiento
Usabilidad
Mantenibilidad
Portabilidad
Compatibilidad
Soporte y Documentación

- **Escalabilidad:** El sistema debe ser capaz de manejar un número creciente de usuarios y volumen de datos sin degradar su rendimiento. Esto implica la capacidad de expandir los recursos del sistema de manera eficiente y adaptarse a la demanda variable, utilizando tecnologías de escalado automático en la nube, como las proporcionadas por Firebase y Heroku.
- **Seguridad:** El sistema debe garantizar la protección de los datos del usuario mediante el uso de autenticación segura y encriptación de datos. Es esencial implementar medidas como la autenticación multifactor (MFA), cifrado de datos en tránsito y en reposo, y auditorías de seguridad regulares para identificar y mitigar vulnerabilidades.
- **Disponibilidad:** El sistema debe estar disponible y accesible para los usuarios al menos el 99.5% del tiempo. Esto requiere la implementación de arquitecturas redundantes, planes de recuperación ante desastres y monitoreo continuo del sistema para detectar y resolver problemas de manera proactiva.
- **Rendimiento:** El sistema debe ofrecer tiempos de respuesta rápidos, con tiempos de carga de páginas inferiores a 2 segundos en condiciones normales. Esto se puede lograr mediante la optimización del código, el uso eficiente de cachés y la implementación de prácticas de desarrollo enfocadas en el rendimiento.
- **Usabilidad:** El sistema debe proporcionar una interfaz de usuario intuitiva y fácil de usar, que facilite la navegación y el acceso a las funcionalidades. Esto implica realizar pruebas de usabilidad con usuarios reales, diseñar interfaces consistentes y accesibles, y seguir las mejores prácticas de diseño de experiencia de usuario (UX).

CHAPTER 5. SISTEMA DESARROLLADO

- **Mantenibilidad:** El sistema debe ser fácil de mantener y actualizar, con una arquitectura modular que permita la incorporación de nuevas funcionalidades sin afectar las existentes. El uso de principios de arquitectura limpia y el seguimiento de las mejores prácticas de desarrollo de software son esenciales para asegurar la mantenibilidad.

- **Portabilidad:** El sistema debe ser compatible con múltiples plataformas (iOS, Android, Web), permitiendo su uso en diferentes dispositivos. Esto requiere el desarrollo con tecnologías multiplataforma como Flutter, asegurando que el sistema funcione de manera consistente en todas las plataformas soportadas.

- **Compatibilidad:** El sistema debe integrarse fácilmente con otros sistemas y servicios, como APIs de terceros y servicios de backend. Esto implica diseñar APIs bien definidas y utilizar estándares abiertos para asegurar la interoperabilidad con otras soluciones tecnológicas.

- **Soporte y Documentación:** El sistema debe estar bien documentado, proporcionando guías de usuario, manuales de instalación y mantenimiento, y soporte técnico. La documentación completa y accesible es crucial para ayudar a los usuarios y desarrolladores a entender y utilizar el sistema de manera efectiva.

5.1.5 Componentes del Proyecto

La estructura de carpetas del proyecto CodeLingo está diseñada para seguir principios de Clean Architecture, asegurando una organización clara y una separación adecuada de las responsabilidades. A continuación se presenta la estructura de carpetas utilizada:

CHAPTER 5. SISTEMA DESARROLLADO

Listado de la estructura de carpetas de Codelingo

```
codelingo/  
|-- lib/  
|   |-- core/  
|   |-- features/  
|   |   |-- auth/  
|   |       |-- data/  
|   |           |-- datasources/  
|   |           |-- models/  
|   |           |-- repositories/  
|   |       |-- domain/  
|   |           |-- entities/  
|   |           |-- provider/  
|   |           |-- repositories/  
|   |           |-- usecases/  
|   |       |-- presentation/  
|   |           |-- controllers/  
|   |           |-- pages/  
|   |           |-- widgets/  
|   |-- env.dart  
|   |-- firebase_options.dart  
|   |-- main.dart  
|-- assets/  
|   |-- data/  
|   |-- fonts/  
|   |-- icons/  
|   |-- images/  
|   |-- questions/  
|-- backend/  
|   |-- app.py  
|-- firebase.json  
|-- .env  
|-- pubspec.yaml  
|-- requirements.txt  
|-- runtime.txt  
|-- Procfile
```

- **lib/core/**: Contiene componentes centrales y utilitarios que son compartidos a lo largo de la aplicación. Estos pueden incluir configuraciones, temas y funciones comunes que no pertenecen a una funcionalidad específica.
- **lib/features/auth/**: Agrupa todo lo relacionado con la funcionalidad de autenticación, subdividida en tres capas principales según el patrón de diseño de Clean Architecture:
 - **data/**: Maneja las interacciones directas con los datos. Contiene las subcarpetas:
 - * **datasources/**: Fuentes de datos, como conexiones a APIs o bases de datos.
 - * Un ejemplo es `datasources/auth_remote_data_source.dart`, donde se gestionan las interacciones con el servicio de autenticación remoto.
 - * **models/**: Definiciones de los modelos de datos utilizados por la aplicación.
 - Un ejemplo es `models/user_model.dart`, que define la estructura de los datos de usuario.

CHAPTER 5. SISTEMA DESARROLLADO

- * **repositories/**: Implementaciones de los repositorios que se encargan de la gestión de datos.
 - Un ejemplo es `repositories/auth_repository_impl.dart`, que implementa las interfaces del repositorio de autenticación.
- **domain/**: Contiene la lógica de negocio pura, independiente de cualquier tecnología o plataforma específica. Subcarpetas:
 - * **entities/**: Definiciones de las entidades del dominio, que representan los objetos del negocio.
 - Un ejemplo es `entities/user.dart`.
 - * **provider/**: Proveedores de servicios y dependencias necesarias para la lógica de negocio.
 - * **repositories/**: Interfaces de los repositorios, que definen cómo interactuar con las fuentes de datos.
 - Un ejemplo es `repositories/auth_repository.dart`.
 - * **usecases/**: Casos de uso que encapsulan la lógica de negocio específica.
 - Por ejemplo, `usecases/login_user.dart` define la lógica para el caso de uso de inicio de sesión de usuarios.
- **presentation/**: Maneja la interfaz de usuario y la interacción con el usuario. Subcarpetas:
 - * **controllers/**: Controladores que manejan la lógica de presentación.
 - Un ejemplo es `controllers/login_controller.dart`.
 - * **pages/**: Páginas y vistas de la aplicación.
 - Un ejemplo es `pages/login_page.dart`.
 - * **widgets/**: Widgets reutilizables utilizados en las interfaces de usuario.
- **lib/env.dart**: Gestiona variables de entorno y configuraciones sensibles utilizando el paquete `dotenv`.
- **lib/firebase_options.dart**: Contiene configuraciones específicas de Firebase, necesarias para la inicialización de servicios.
- **lib/main.dart**: Punto de entrada de la aplicación, donde se inicializa y configura el entorno de ejecución.
- **assets/**: Contiene recursos gráficos y de datos utilizados en la aplicación.
 - **data/**: Datos de configuración y otros archivos de datos.
 - **fonts/**: Fuentes utilizadas en la aplicación.
 - **icons/**: Iconos utilizados en la aplicación.

CHAPTER 5. SISTEMA DESARROLLADO

- **images/**: Imágenes utilizadas en la aplicación.
- **questions/**: Contiene preguntas y configuraciones relacionadas.
- **backend/**: Contiene el código del backend de la aplicación.
 - Un ejemplo es `app.py`, que gestiona el sistema de recomendación de cursos y otros servicios backend.
- **firebase.json**: Contiene configuraciones específicas de Firebase para el despliegue y uso de servicios como Firestore y Authentication.
- **.env**: Archivo que contiene variables de entorno, utilizado para almacenar configuraciones sensibles como claves de API.
- **pubspec.yaml**: Archivo de configuración de Flutter que especifica las dependencias y otras configuraciones del proyecto.
- **requirements.txt**: Lista de dependencias necesarias para la ejecución del backend, especialmente relevante para entornos de Heroku.
- **runtime.txt**: Especifica la versión de Python (u otro lenguaje) a utilizar en el entorno de ejecución de Heroku.
- **Procfile**: Define los comandos que deben ejecutarse para iniciar la aplicación en Heroku.

La organización modular del sistema facilita la mantenibilidad, reusabilidad y escalabilidad del código, permitiendo una separación clara de las responsabilidades y mejorando la estructura general del proyecto.

5.2 Desarrollo del Backend

El desarrollo del backend de la plataforma CodeLingo se ha centrado en la utilización de los servicios de Firebase, específicamente Firebase Authentication y Cloud Firestore Database. Estos servicios proporcionan una solución robusta y escalable para la autenticación de usuarios y la gestión de datos en tiempo real, facilitando el desarrollo de aplicaciones modernas y eficientes.

5.2.1 Authentication

Firebase Authentication se ha empleado para gestionar el registro y la autenticación de los usuarios en la plataforma CodeLingo. Este servicio ofrece una variedad de métodos de autenticación, incluyendo correo electrónico y contraseña, autenticación

CHAPTER 5. SISTEMA DESARROLLADO

con Google, Facebook, y otros proveedores de identidad. La implementación se realizó con el objetivo de asegurar una experiencia de usuario sencilla y segura.

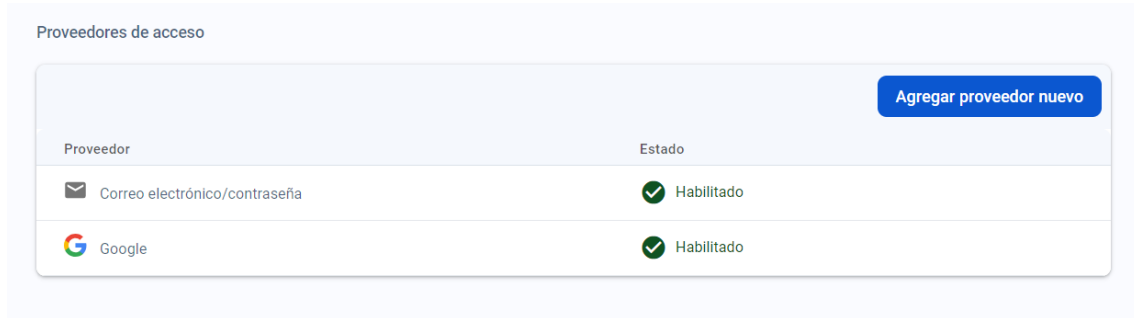


Figure 5.3: Proveedores de Authentication de Codelingo en la Consola de Firebase Authentication

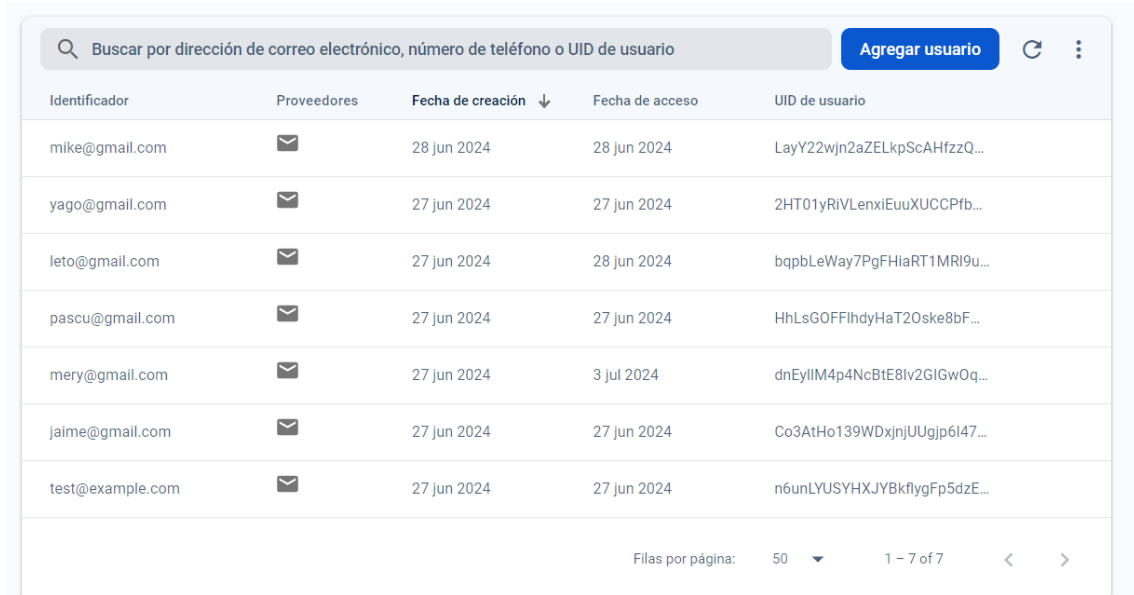
El proceso de autenticación se realiza a través de la integración del SDK de Firebase Authentication en la aplicación Flutter. El flujo básico de autenticación se compone de la siguiente forma:

1. **Registro de Usuario:** Los usuarios pueden registrarse utilizando su dirección de correo electrónico y una contraseña. Los datos de registro se envían a Firebase Authentication, que se encarga de crear un nuevo usuario en el sistema.
2. **Inicio de Sesión:** Los usuarios registrados pueden iniciar sesión proporcionando sus credenciales. Firebase Authentication verifica las credenciales y devuelve un token de autenticación.
3. **Autenticación con Proveedores Externos:** Se ha habilitado la autenticación con Google para ofrecer opciones adicionales a los usuarios. Este proceso se realiza mediante el uso de los SDKs correspondientes y su integración con Firebase Authentication.

El nivel de dificultad en la implementación de Firebase Authentication es moderado, debido a la necesidad de configurar correctamente los proveedores de autenticación y manejar los diferentes flujos de autenticación en la aplicación. A continuación se muestra una captura de pantalla de la consola de Firebase configurada para la autenticación de usuarios:

UNIVERSIDAD PONTIFICIA COMILLAS
Escuela Técnica Superior de Ingeniería (ICAI)
Grado en Ingeniería en Tecnologías de Telecomunicación

CHAPTER 5. SISTEMA DESARROLLADO



Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID de usuario
mike@gmail.com	✉	28 jun 2024	28 jun 2024	LayY22wjn2aZELkpScAHfzzQ...
yago@gmail.com	✉	27 jun 2024	27 jun 2024	2HT01yRiVLexiEuuXUCCPfb...
leto@gmail.com	✉	27 jun 2024	28 jun 2024	bqpbLeWay7PgFHiaRT1MRI9u...
pascu@gmail.com	✉	27 jun 2024	27 jun 2024	HhLsG0FFIhdyHaT20ske8bF...
mery@gmail.com	✉	27 jun 2024	3 jul 2024	dnEyllM4p4NcBtE8lv2GIGwOq...
jaime@gmail.com	✉	27 jun 2024	27 jun 2024	Co3AtHo139WDxjnjUUGjp6l47...
test@example.com	✉	27 jun 2024	27 jun 2024	n6unLYUSYHXJYBkflygFp5dzE...

Filas por página: 50 1 - 7 of 7

Figure 5.4: Usuarios registrados en CodeLingo

El siguiente código muestra la implementación de un servicio de autenticación utilizando Firebase Authentication:

```
1 import 'package:firebase_auth/firebase_auth.dart';
2
3 class AuthService {
4   final FirebaseAuth _firebaseAuth = FirebaseAuth.instance;
5
6   Future<User?> signInWithEmailAndPassword(String email, String
7     password) async {
8     try {
9       UserCredential result = await _firebaseAuth.
10         signInWithEmailAndPassword(
11           email: email, password: password);
12       return result.user;
13     } catch (e) {
14       print(e);
15       return null;
16     }
17   }
18
19   Future<User?> registerWithEmailAndPassword(String email, String
20     password) async {
21     try {
22       UserCredential result = await _firebaseAuth.
23         createUserWithEmailAndPassword(
24           email: email, password: password);
25       return result.user;
26     } catch (e) {
```

CHAPTER 5. SISTEMA DESARROLLADO

```
23     print(e);
24     return null;
25 }
26 }
27
28 Future<bool> isEmailAlreadyRegistered(String email) async {
29     try {
30         final signInMethods = await _firebaseAuth.
31             fetchSignInMethodsForEmail(email);
32         return signInMethods.isNotEmpty;
33     } catch (e) {
34         print(e);
35         return false;
36     }
37 }
38
39 Future<void> signOut() async {
40     await _firebaseAuth.signOut();
41 }
```

Listing 5.1: Servicio de autenticación en Flutter

El servicio de autenticación en Flutter implementado mediante Firebase Authentication proporciona una serie de métodos cruciales para la gestión de usuarios:

- **signInWithEmailAndPassword**: Este método permite a los usuarios iniciar sesión en la aplicación utilizando su correo electrónico y contraseña. Si las credenciales son correctas, se devuelve una instancia de **User**, representando al usuario autenticado.
- **registerWithEmailAndPassword**: Este método registra nuevos usuarios utilizando un correo electrónico y una contraseña. Si el registro es exitoso, se devuelve una instancia de **User**, representando al usuario recién creado.
- **isEmailAlreadyRegistered**: Este método verifica si una dirección de correo electrónico ya está registrada en el sistema. Esto es útil para evitar duplicados y mejorar la experiencia del usuario durante el registro.
- **signOut**: Este método cierra la sesión del usuario actual, asegurando que los datos de la sesión anterior no se mantengan activos.

Cada uno de estos métodos utiliza el paquete **firebase_auth** para interactuar con el servicio de autenticación de Firebase, garantizando una integración fluida y segura en la aplicación CodeLingo. La implementación de estos métodos permite una gestión eficiente y segura de los usuarios, asegurando que solo los usuarios autenticados puedan acceder a las funcionalidades de la aplicación. Esta integración no solo mejora la seguridad, sino que también proporciona una experiencia de usuario más conveniente al permitir múltiples opciones de autenticación.

CHAPTER 5. SISTEMA DESARROLLADO

5.2.2 Cloud Firestore Database

Cloud Firestore se ha utilizado como la base de datos en tiempo real de CodeLingo. Este servicio permite almacenar y sincronizar datos entre los usuarios y la base de datos de forma eficiente y segura. La estructura de la base de datos se ha diseñado para soportar las diferentes funcionalidades de la plataforma, tales como el almacenamiento de perfiles de usuario, progreso en los cursos, y datos de ejercicios.

La integración de Cloud Firestore en la aplicación Flutter se ha realizado mediante el SDK de Firebase para Flutter. Las operaciones básicas incluyen:

1. **Lectura de Datos:** Utilizando el método `get()`, se pueden leer documentos individuales o colecciones completas de la base de datos.
2. **Escritura de Datos:** Con el método `set()`, se pueden escribir datos en documentos específicos, permitiendo actualizaciones y creación de nuevos registros.
3. **Actualización de Datos:** El método `update()` permite modificar campos específicos dentro de un documento existente.
4. **Eliminación de Datos:** A través del método `delete()`, se pueden eliminar documentos o colecciones completas.

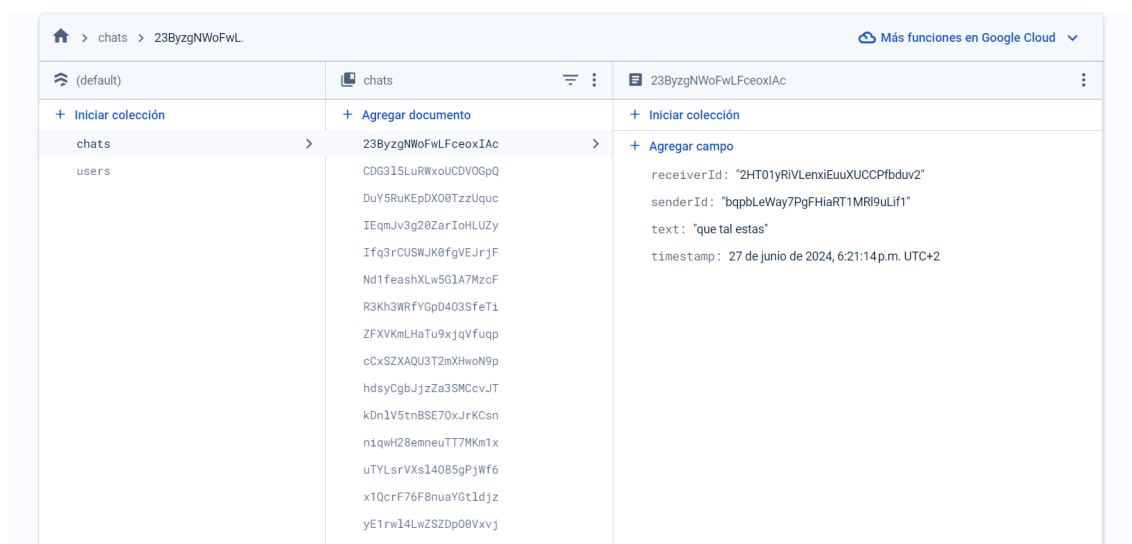


Figure 5.5: Estructura de la base de datos de Cloud Firestore para la funcionalidad de chat en CodeLingo

La complejidad de la implementación de Cloud Firestore radica en el diseño adecuado de la estructura de la base de datos y la gestión de las reglas de seguridad.

CHAPTER 5. SISTEMA DESARROLLADO

Se han definido reglas de seguridad en Firestore para controlar el acceso y las operaciones permitidas a los usuarios autenticados, garantizando así la integridad y privacidad de los datos.

En [5.6](#) se presenta una captura de pantalla de la consola de Firebase con la estructura de la base de datos y un ejemplo de las reglas de seguridad configuradas:

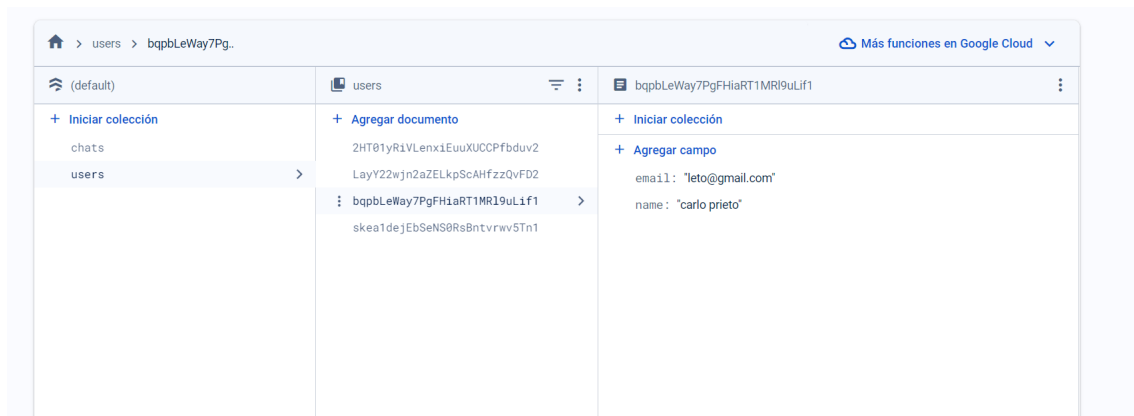


Figure 5.6: Estructura de la base de datos de usuarios en Cloud Firestore

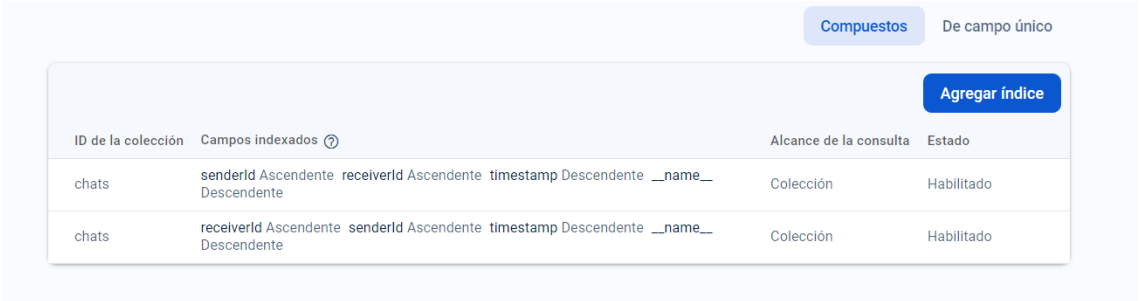
Las reglas de seguridad se han configurado de la siguiente manera para permitir el acceso solo a usuarios autenticados y controlar las operaciones permitidas:

```
1 service cloud.firestore {  
2   match /databases/{database}/documents {  
3     match /{document=**} {  
4       allow read, write: if request.auth != null;  
5     }  
6   }  
7 }
```

Listing 5.2: Reglas de seguridad de Firebase Firestore

En las capturas proporcionadas, se observa la configuración de índices en la base de datos para optimizar las consultas y mejorar el rendimiento. Por ejemplo, se han creado índices en la colección "chats" para los campos `senderId`, `receiverId`, y `timestamp`, tanto en orden ascendente como descendente:

CHAPTER 5. SISTEMA DESARROLLADO



The screenshot shows the Cloud Firestore console interface. At the top, there are tabs for 'Compuestos' (selected) and 'De campo único'. A blue button labeled 'Agregar índice' is in the top right. Below is a table with the following data:

ID de la colección	Campos indexados ⓘ	Alcance de la consulta	Estado
chats	senderId Ascendente receiverId Ascendente timestamp Descendente __name__ Descendente	Colección	Habilitado
chats	receiverId Ascendente senderId Ascendente timestamp Descendente __name__ Descendente	Colección	Habilitado

Figure 5.7: Índices configurados en Cloud Firestore

El siguiente código muestra la implementación de un servicio de chat utilizando Cloud Firestore:

```

1 import 'package:cloud_firestore/cloud_firestore.dart';
2
3 class ChatService {
4   final FirebaseFirestore _firestore = FirebaseFirestore.instance;
5
6   // Get instance of Firestore
7   FirebaseFirestore get firestoreInstance => _firestore;
8
9   // Send message
10  Future<void> sendMessage(String senderId, String receiverId,
11    String text) async {
12    await _firestore.collection('chats').add({
13      'text': text,
14      'senderId': senderId,
15      'receiverId': receiverId,
16      'timestamp': FieldValue.serverTimestamp(),
17    });
18  }
19
20  // Get messages sent by the user to the friend
21  Stream<QuerySnapshot> getSentMessages(String userId, String
22    friendId) {
23    return _firestore.collection('chats')
24      .where('senderId', isEqualTo: userId)
25      .where('receiverId', isEqualTo: friendId)
26      .orderBy('timestamp', descending: true)
27      .snapshots();
28  }
29
30  // Get messages received by the user from the friend
31  Stream<QuerySnapshot> getReceivedMessages(String userId, String
32    friendId) {
33    return _firestore.collection('chats')
34      .where('senderId', isEqualTo: friendId)
35      .where('receiverId', isEqualTo: userId)
36      .orderBy('timestamp', descending: true)

```

CHAPTER 5. SISTEMA DESARROLLADO

```
34         .snapshots();  
35     }  
36 }
```

Listing 5.3: Servicio de chat en Flutter

El servicio de chat implementado en Flutter mediante Cloud Firestore ofrece una solución robusta para la comunicación en tiempo real entre usuarios. Los métodos clave implementados en este servicio son los siguientes:

- **sendMessage:** Este método permite enviar un mensaje de texto desde un usuario remitente (**senderId**) a un usuario receptor (**receiverId**). El mensaje se almacena en la colección "chats" de Firestore junto con un sello de tiempo (**timestamp**) generado por el servidor, asegurando que los mensajes se ordenen cronológicamente.
- **getSentMessages:** Este método devuelve un flujo de datos (**Stream<QuerySnapshot>**) que contiene todos los mensajes enviados por un usuario específico (**userId**) a otro usuario específico (**friendId**). Los mensajes se ordenan en orden descendente de acuerdo a su sello de tiempo, proporcionando una vista en tiempo real de los mensajes enviados.
- **getReceivedMessages:** Similar al método anterior, este método devuelve un flujo de datos que contiene todos los mensajes recibidos por un usuario (**userId**) de otro usuario (**friendId**). Los mensajes también se ordenan en orden descendente por tiempo.

La implementación de estos métodos aprovecha las funcionalidades avanzadas de Firestore para gestionar la comunicación en tiempo real, garantizando que los mensajes se sincronicen instantáneamente entre los usuarios. Esta funcionalidad es crucial para aplicaciones de chat, donde la inmediatez y la fiabilidad son esenciales para una experiencia de usuario satisfactoria.

Las reglas de seguridad en Firestore se han configurado para restringir el acceso a los datos solo a usuarios autenticados, asegurando que la privacidad y la integridad de los datos se mantengan. La configuración de índices mejora el rendimiento de las consultas, permitiendo que las operaciones de lectura y escritura se realicen de manera eficiente incluso con grandes volúmenes de datos.

La integración de Cloud Firestore en CodeLingo proporciona una solución escalable y segura para la gestión de datos en tiempo real, facilitando funcionalidades avanzadas como el chat en tiempo real y la sincronización instantánea de datos entre los usuarios.

5.2.3 Implementación de Firebase en el Proyecto

Para integrar Firebase en el proyecto CodeLingo, se han seguido los siguientes pasos y se han incluido varios archivos de configuración necesarios:

CHAPTER 5. SISTEMA DESARROLLADO

1. **Configuración del Proyecto en Firebase Console:** Se creó un nuevo proyecto en Firebase Console y se configuraron los servicios de Authentication y Firestore Database.
2. **Integración del SDK de Firebase:** Se añadieron las dependencias necesarias en el archivo `pubspec.yaml` de Flutter:

```
1 dependencies:  
2   firebase_core: ^1.10.0  
3   firebase_auth: ^3.3.5  
4   cloud_firestore: ^3.1.5
```

Listing 5.4: Dependencias de Firebase en `pubspec.yaml`

3. **Configuración de Archivos de Inicialización:** Se añadió el archivo `google-services.json` en el directorio `android/app` y `GoogleService-Info.plist` en el directorio `ios/Runner`.
4. **Inicialización de Firebase en el Código:** En el archivo principal de la aplicación, se inicializó Firebase utilizando el método `Firebase.initializeApp()`:

```
1 void main() async {  
2   WidgetsFlutterBinding.ensureInitialized();  
3   await Firebase.initializeApp();  
4   runApp(MyApp());  
5 }
```

Listing 5.5: Inicialización de Firebase en el archivo principal

Estos pasos aseguran que la aplicación esté correctamente configurada para utilizar los servicios de Firebase, proporcionando una base sólida para la autenticación de usuarios y la gestión de datos en tiempo real.

5.3 Desarrollo del Frontend

El desarrollo del frontend de CodeLingo ha sido planeado y ejecutado para ofrecer una experiencia de usuario intuitiva, atractiva y efectiva. Se ha prestado especial atención al diseño UI/UX, asegurando que cada componente de la interfaz no solo sea visualmente atractivo, sino también funcional y fácil de usar. En esta sección, detallaremos los principios y decisiones de diseño que han guiado el desarrollo del frontend de CodeLingo, incluyendo el uso de la mascota Cuddy, la elección de la fuente Nunito, y la paleta de colores basada en azul y blanco.

5.3.1 Diseño UI/UX

El diseño de la interfaz de usuario (UI) y la experiencia de usuario (UX) son componentes críticos en el desarrollo de cualquier aplicación moderna. Un buen diseño UI/UX no solo hace que la aplicación sea agradable a la vista, sino que también facilita la navegación y el uso eficiente de sus funcionalidades. En CodeLingo, gracias al uso de las mejores prácticas y ejemplos exitosos en el sector, como Duolingo, se ha conseguido crear una experiencia de aprendizaje inmersiva y efectiva.

5.3.1.1 Uso de la Mascota Coddy

Coddy, la mascota de CodeLingo, ha sido diseñada utilizando IA de Microsoft Copilot. Al igual que Duo en [5.8](#), Coddy juega un papel crucial en la interfaz de la aplicación, proporcionando orientación, motivación y feedback a los usuarios. La presencia de Coddy hace que la experiencia de aprendizaje sea más amigable y menos intimidante, especialmente para los nuevos usuarios.

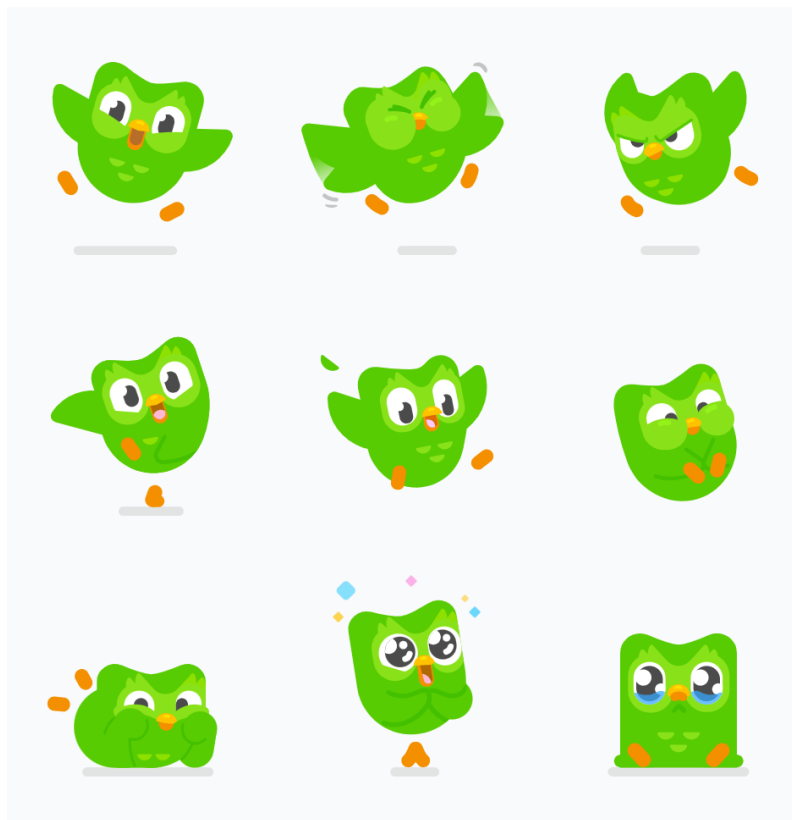


Figure 5.8: Mascota Duo de la app Duolingo

CHAPTER 5. SISTEMA DESARROLLADO

El prompt utilizado para generar a Coddy fue el siguiente:

Create a friendly and cute turtle character named Coddy. Coddy should be wearing glasses and interacting with a laptop or computer, showcasing that it is learning or programming. The turtle should have a happy and welcoming expression to make users feel comfortable and motivated.

5.3.1.2 Elección de la Fuente: Nunito

Para mantener una estética moderna y amigable, se ha elegido la fuente Nunito, que se asemeja mucho a la fuente utilizada por Duolingo. Nunito es una fuente sans-serif que ofrece una excelente legibilidad y un diseño limpio, contribuyendo a una experiencia de usuario más cómoda y placentera. Gracias las directrices de Duolingo en cuanto a tipografía y diseño, como se describe en su [guía de identidad]

5.3.1.3 Paleta de Colores: Azul y Blanco

La paleta de colores de CodeLingo se ha basado en dos colores principales: azul y blanco. Esta combinación no solo es visualmente agradable, sino que también transmite una sensación de profesionalismo y confianza. El uso del azul como color primario ayuda a crear una conexión emocional positiva, mientras que el blanco proporciona un fondo limpio y despejado que mejora la legibilidad y el enfoque en los contenidos importantes.

5.3.1.4 Importancia de Tener una Mascota como Coddy

La incorporación de una mascota como Coddy en la interfaz de CodeLingo tiene múltiples beneficios. Primero, actúa como un guía interactivo, ofreciendo instrucciones y sugerencias que ayudan a los usuarios a navegar y utilizar la aplicación de manera más efectiva. Segundo, Coddy aporta un elemento de gamificación al proceso de aprendizaje, haciendo que sea más divertido y motivador. Finalmente, la presencia de Coddy humaniza la aplicación, creando una conexión emocional con los usuarios que puede aumentar su compromiso y retención.



Figure 5.9: Coddy motivando a los usuarios

5.3.1.5 Log In / Register

Esta sección permite a los usuarios registrarse o iniciar sesión en la plataforma, proporcionando una interfaz sencilla y directa para comenzar a utilizar los servicios de CodeLingo.

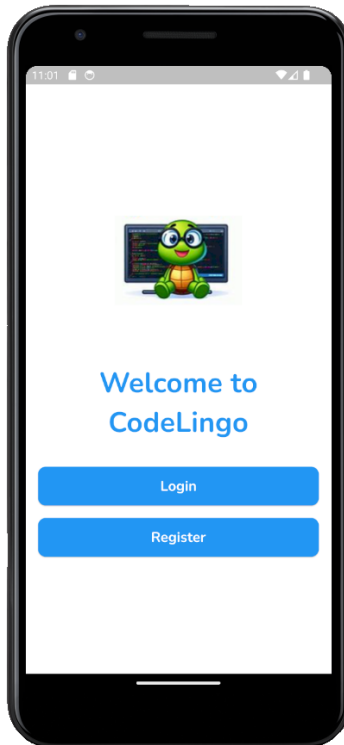


Figure 5.10: Pantalla de Home

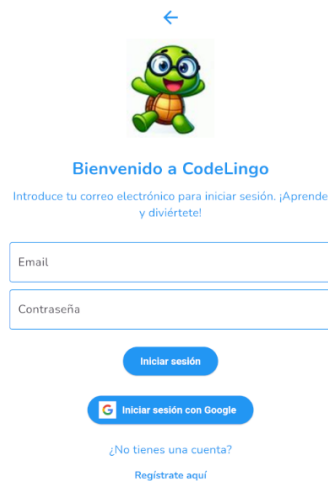


Figure 5.11: Pantalla de Login

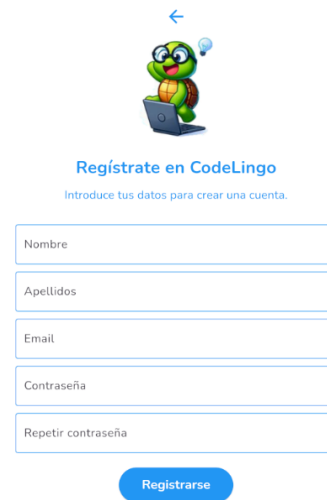


Figure 5.12: Pantalla de Registro

CHAPTER 5. SISTEMA DESARROLLADO

5.3.1.6 Perfil de Usuario

En esta sección, los usuarios pueden ver y editar su perfil, gestionar sus preferencias y visualizar estadísticas de su progreso de aprendizaje.



Figure 5.13: Pantalla 1 de Perfil del usuario

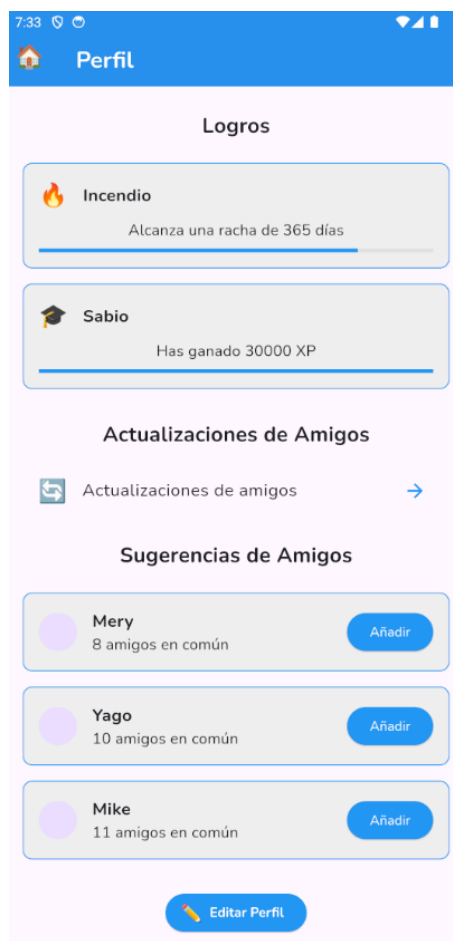


Figure 5.14: Pantalla 2 de Perfil del usuario

5.3.1.7 Lobby principal

El lobby principal actúa como el centro de navegación de la aplicación, donde los usuarios pueden acceder a las diferentes funcionalidades y contenidos disponibles.

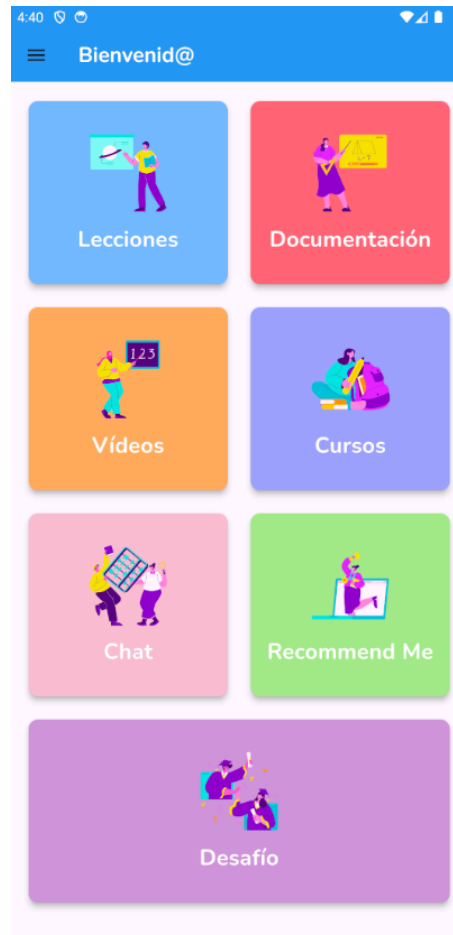


Figure 5.15: Pantalla de Lobby principal

CHAPTER 5. SISTEMA DESARROLLADO

5.3.1.8 Lecciones

La sección de lecciones permite a los usuarios acceder a diversos cursos y ejercicios interactivos para aprender nuevos conceptos y practicar sus habilidades de programación.

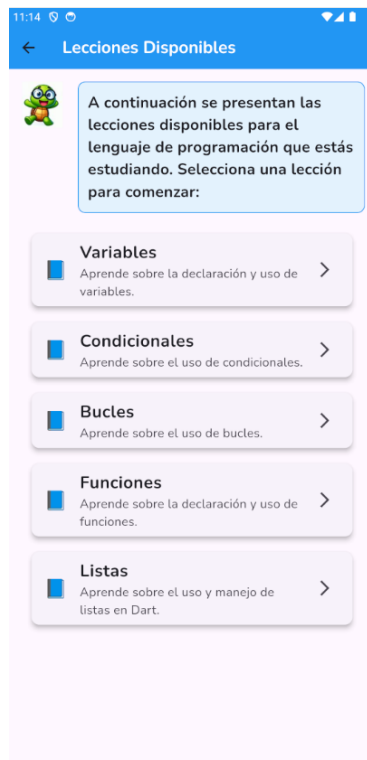


Figure 5.16: Pantalla de Lecciones

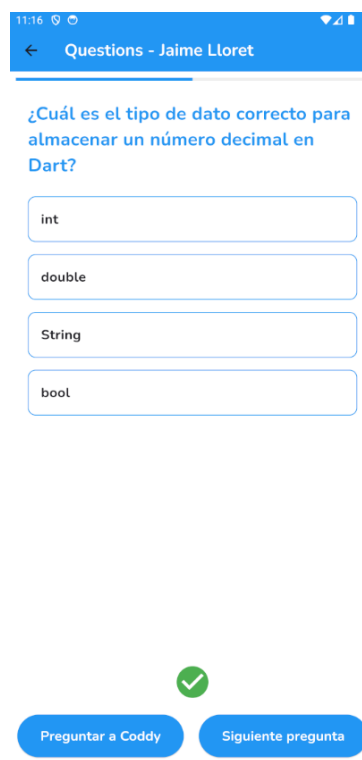


Figure 5.17: Pantalla de Ejercicios

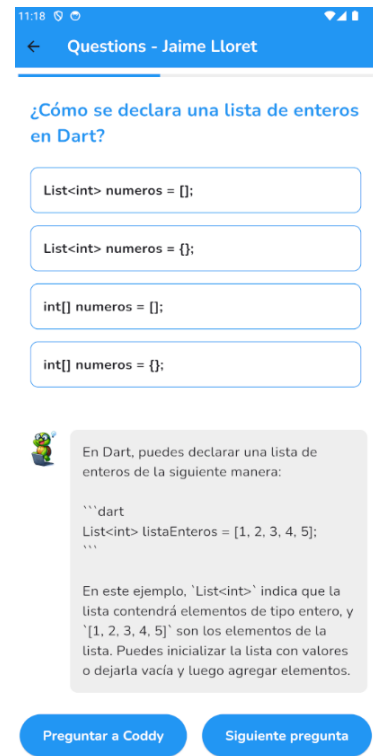


Figure 5.18: Pantalla de Ejercicios con ayuda de Cuddy

5.3.1.9 Documentación

La documentación es una herramienta esencial que proporciona a los usuarios acceso a información detallada y referencias que apoyan su proceso de aprendizaje.

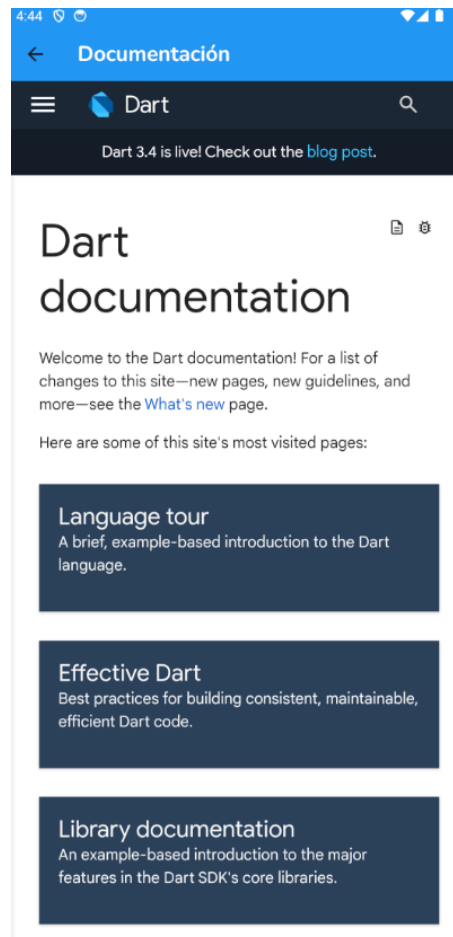


Figure 5.19: Pantalla de Documentación

5.3.1.10 Videos

La sección de videos ofrece a los usuarios tutoriales y contenido visual que complementa el aprendizaje teórico y práctico.

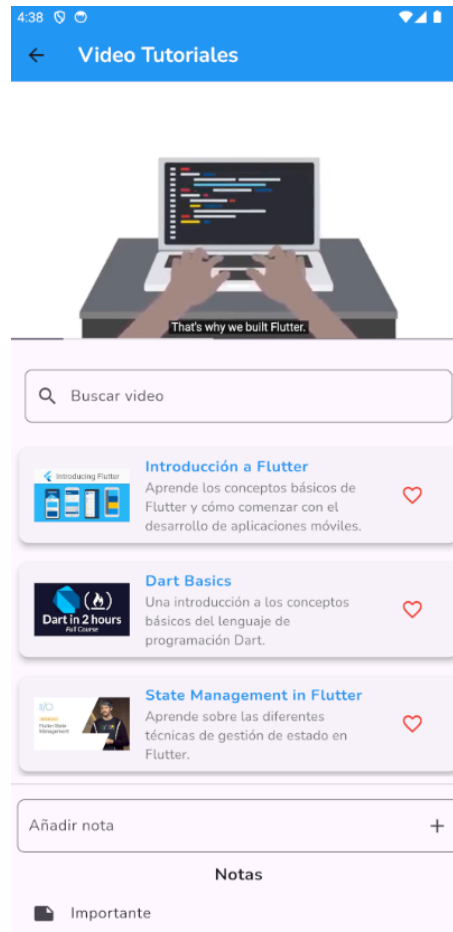


Figure 5.20: Pantalla de Vídeos

CHAPTER 5. SISTEMA DESARROLLADO

5.3.1.11 Cursos disponibles

En esta sección, los usuarios pueden explorar y suscribirse a diferentes cursos disponibles en la plataforma, ampliando así su conocimiento y habilidades.



Figure 5.21: Pantalla de Cursos con buscador

CHAPTER 5. SISTEMA DESARROLLADO

5.3.1.12 Chat

La funcionalidad de chat permite a los usuarios comunicarse entre sí, facilitando la colaboración y el intercambio de conocimientos.



Figure 5.22: Pantalla 1 del Chat
perspectiva de Miguel



Figure 5.23: Pantalla 2 del Chat
perspectiva de Jaime

CHAPTER 5. SISTEMA DESARROLLADO

5.3.1.13 Recommend Me

La función "Recommend Me" sugiere lecciones y cursos personalizados según el progreso y las preferencias del usuario, optimizando su experiencia de aprendizaje.

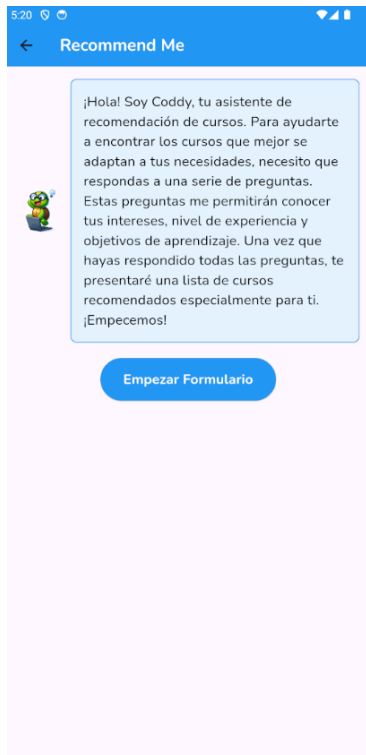


Figure 5.24: Pantalla de descripción de Recommend Me

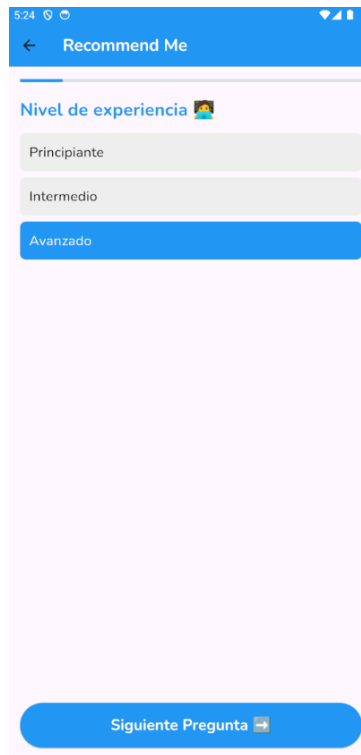


Figure 5.25: Pantalla de Nivel de experiencia

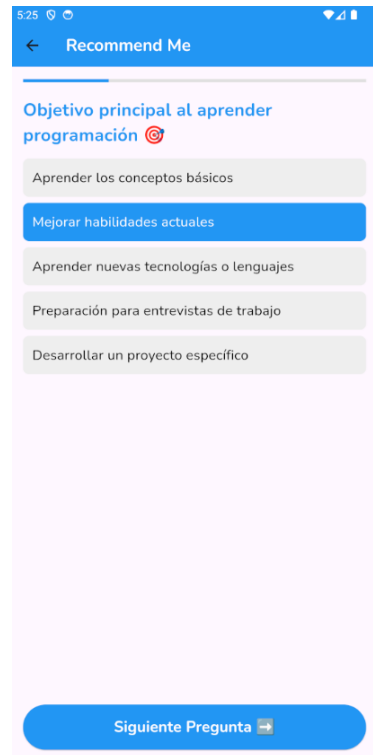


Figure 5.26: Pantalla de objetivo del aprendizaje

CHAPTER 5. SISTEMA DESARROLLADO

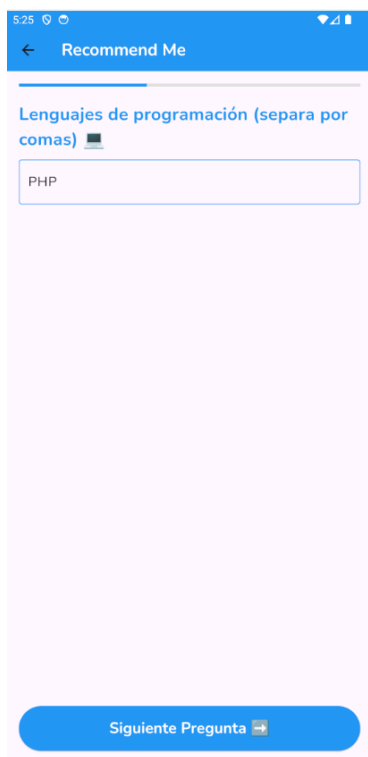


Figure 5.27: Pantalla de lenguaje de interés

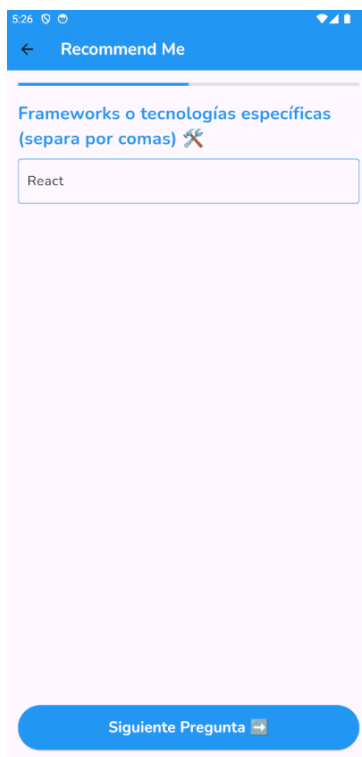


Figure 5.28: Pantalla de framework de interés

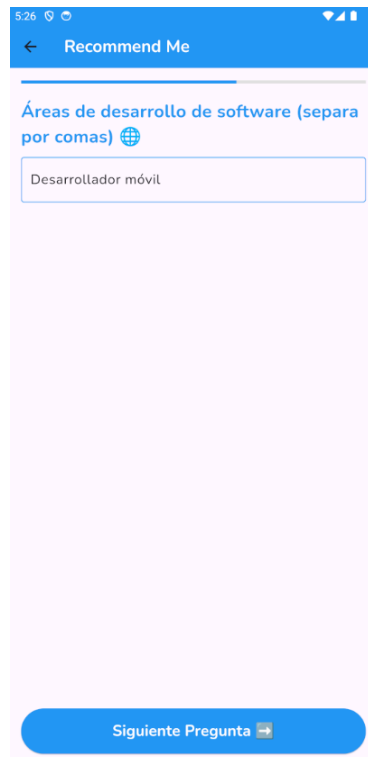


Figure 5.29: Pantalla de área de interés

CHAPTER 5. SISTEMA DESARROLLADO

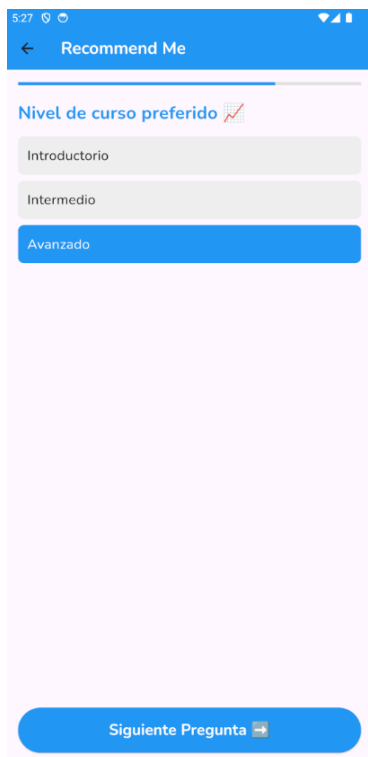


Figure 5.30: Pantalla de nivel del curso

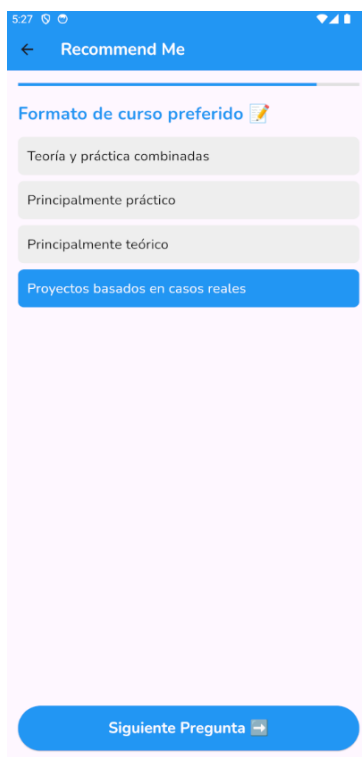


Figure 5.31: Pantalla de formato de curso de interés

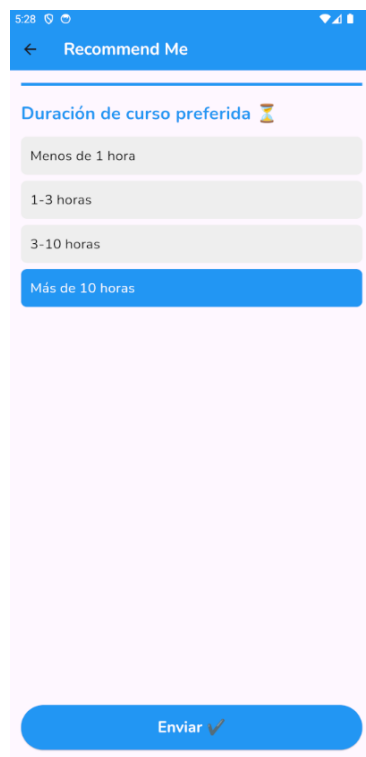


Figure 5.32: Pantalla de duración del curso

5.3.1.14 Desafíos

En este apartado se muestran las pantallas de desafíos los cuales proporcionan retos prácticos que motivan y refuerzan el aprendizaje continuo. En primer lugar se encuentra 5.33 que organiza los contenidos en áreas temáticas según su dificultad y en segundo lugar, 5.34 en la que se puede ver la estructura del progreso del usuario en el que se divide cada categoría en 5 niveles de dificultad:



Figure 5.33: Pantalla de las categorías de los desafíos

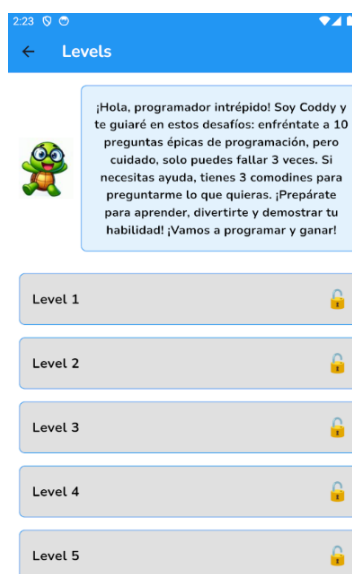


Figure 5.34: Pantalla los niveles de los desafíos

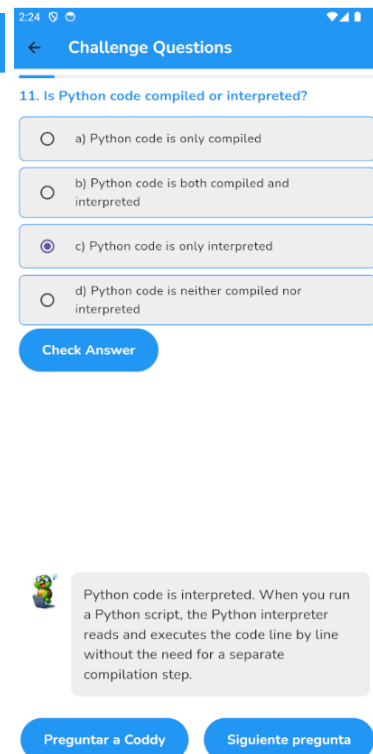


Figure 5.35: Pantalla de ejercicios de desafío

5.4 Implementación de Inteligencia Artificial

En este proyecto, se ha decidido integrar la API de ChatGPT desarrollada por OpenAI para ofrecer asistencia interactiva a los usuarios durante el uso de la app y de esta manera mejorar la experiencia del usuario al proporcionar respuestas inteligente y contextualmente relevantes. En este apartado se describe desde la implementación en el proyecto, los motivos detrás de la elección de ChatGPT sobre otras opciones disponibles y algunos aspectos técnicos.

ChatGPT utiliza la arquitectura de transformers, una innovación en el campo del

CHAPTER 5. SISTEMA DESARROLLADO

procesamiento del lenguaje natural (NLP). Los transformers permiten que el modelo preste atención a diferentes partes de la entrada a medida que genera la salida, procesando y generando texto de manera eficiente, haciendo que las respuestas sean coherentes y manejando secuencias de texto más largas. Esto ayuda a los usuarios a resolver sus dudas y avanzar en su aprendizaje de programación de una manera más efectiva pudiendo leer la explicación en todo momento.

La figura 5.37 muestra la arquitectura de un Transformer mostrando el flujo de datos desde el embedding de entrada hasta el embedding de salida, pasando por las capas de atención y feed forward, permitiendo generar texto de alta calidad y relevante para las consultas del usuario.

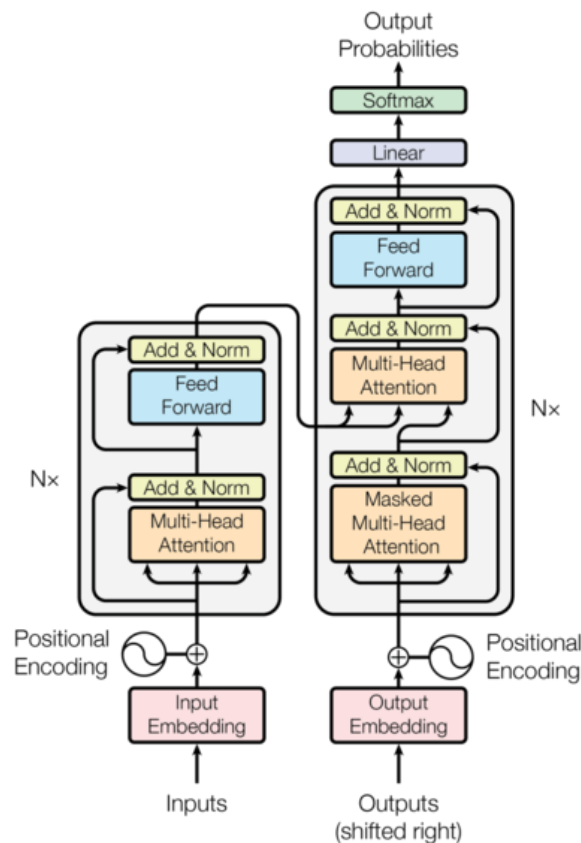


Figure 5.36: Esquema de Arquitectura transformers ChatGPT

La API de ChatGPT proporciona una interfaz sencilla para integrar las capacidades del modelo en aplicaciones de terceros. Mediante solicitudes HTTP, los desarrolladores pueden enviar consultas al modelo y recibir respuestas generadas dinámicamente. La API también permite ajustar parámetros como la longitud de la respuesta y la temperatura para controlar la creatividad de las respuestas.

La integración de la API de ChatGPT en la aplicación se llevó a cabo en varios pasos clave:

1. Obtención de la clave API: para utilizar la API, primero es necesario obtener

CHAPTER 5. SISTEMA DESARROLLADO

una clave de API desde el portal de desarrolladores de OpenAI, la cual se utiliza para autenticar las solicitudes a la API.

2. La configuración del entorno se realiza utilizando el paquete `flutter_dotenv` para gestionar las variables de entorno de manera segura. La clave de API se almacena en un archivo `.env` que no se incluye en el control de versiones para mantener la seguridad.

3. Las solicitudes a la API se realizan utilizando la biblioteca `http` en Flutter. A continuación, se muestra un fragmento de código que ilustra cómo se envía una consulta a la API y se procesa la respuesta.

```
1 Future<void> _askCuddy(String question) async {
2   final apiKey = dotenv.env['API_KEY'];
3   final response = await http.post(
4     Uri.parse('https://api.openai.com/v1/chat/completions'),
5     headers: {
6       'Content-Type': 'application/json',
7       'Authorization': 'Bearer $apiKey',
8     },
9     body: jsonEncode({
10       'model': 'gpt-3.5-turbo',
11       'messages': [
12         {'role': 'system', 'content': 'Eres un asistente de
          Inteligencia Artificial especializado en ayudar a los
          usuarios a resolver preguntas de programacion y
          proporcionar orientacion en sus proyectos de aprendizaje
          de lenguajes de programacion. Tu objetivo principal es
          ofrecer respuestas claras, precisas y utiles que mejoren
          la experiencia de aprendizaje del usuario.'},
13         {'role': 'user', 'content': question}
14       ],
15       'max_tokens': 50,
16       'temperature': 0.5,
17     }),
18   );
19
20   if (response.statusCode == 200) {
21     final responseData = jsonDecode(utf8.decode(response.bodyBytes)
22       );
23     setState(() {
24       _response = responseData['choices'][0]['message']['content'];
25     });
26   } else {
27     setState(() {
28       _response = 'Failed to get response: ${response.statusCode}';
29     });
30   }
31 }
```

Listing 5.6: Función `_askCuddy`

CHAPTER 5. SISTEMA DESARROLLADO

La respuesta de la API se maneja asegurando que el texto se procese correctamente utilizando UTF-8 para manejar adecuadamente los caracteres especiales y acentos en otros idiomas.

Se ha implementado un botón de ayuda en la interfaz de usuario que permite a los usuarios solicitar asistencia de ChatGPT. Al hacer clic en el botón, se envía el enunciado de la pregunta actual a la API y se muestra la respuesta generada. Para mejorar la experiencia del usuario, el texto de la respuesta de ayuda se resetea cuando el usuario avanza a la siguiente pregunta. Esto garantiza que cada pregunta se maneje de manera independiente. La respuesta de ChatGPT se presenta en la interfaz junto con una imagen de la mascota "Coddy". Esto crea una experiencia visual atractiva, simulando un bocadillo de cómic que contiene la respuesta generada.

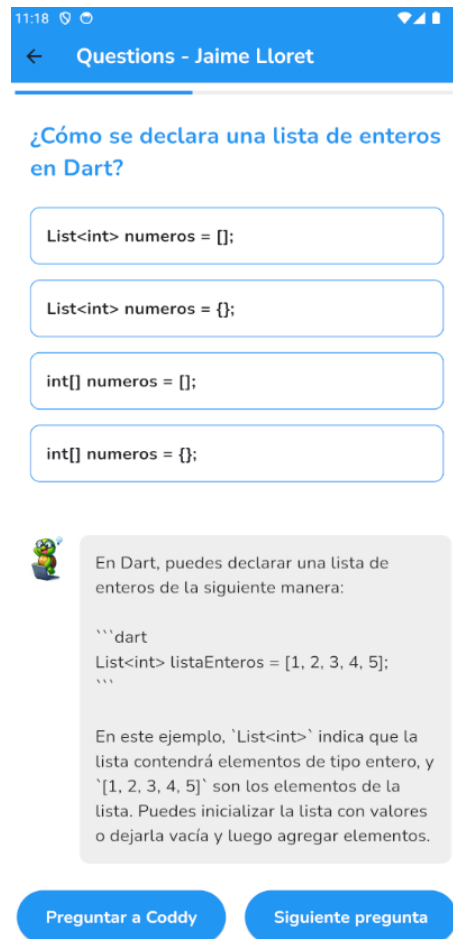


Figure 5.37: Captura de pantalla funcionalidad "Preguntar a Coddy"

La API de ChatGPT es fácil de integrar y está bien documentada, lo que facilita su implementación en aplicaciones de terceros y permite un alto grado de personalización en las respuestas, adaptándose a diversos casos de uso y necesidades específicas del proyecto [10].

CHAPTER 5. SISTEMA DESARROLLADO

En el contexto de los modelos de lenguaje, los tokens son las unidades mínimas de texto (como palabras o partes de palabras) que el modelo procesa. La longitud de una respuesta generada se mide en tokens. OpenAI cobra por el uso de la API en función de la cantidad de tokens utilizados. Es importante considerar el costo estimado para el proyecto, especialmente si se espera un alto volumen de consultas. Para optimizar el uso de tokens y reducir costos, se pueden implementar estrategias como limitar la longitud de las respuestas y estructurar las solicitudes de manera eficiente.

5.5 Implementación de Machine Learning

En la plataforma CodeLingo, se ha implementado un sistema de recomendación basado en contenido para sugerir cursos de programación a los usuarios. Este sistema utiliza técnicas de procesamiento de lenguaje natural (NLP) y aprendizaje automático para analizar las descripciones de los cursos y las preferencias del usuario, a fin de sugerir cursos que mejor se adapten a sus necesidades.

5.5.1 Sistema de Recomendación Basado en Contenido

Un sistema de recomendación es una herramienta que sugiere productos, servicios o información a los usuarios basándose en el análisis de datos. En CodeLingo, se utiliza un sistema de filtrado basado en contenido que analiza las descripciones de los cursos y las preferencias del usuario para realizar recomendaciones personalizadas. A continuación, se detallan los componentes y el flujo del sistema:

5.5.1.1 Entrada de Datos del Usuario

El sistema recibe las preferencias del usuario a través de un formulario o una solicitud API. Estas preferencias incluyen:

Preferencias de entrada
Experiencia previa
Objetivos de aprendizaje
Lenguajes de programación preferidos
Frameworks de interés
Áreas de interés
Formato de curso (e.g., video, lectura)
Duración del curso deseada

Table 5.3: Tabla de preferencias de entrada

CHAPTER 5. SISTEMA DESARROLLADO

5.5.1.2 Carga y Preprocesamiento de Datos

El sistema carga un dataset de cursos que contiene información detallada sobre cada curso, incluyendo:

Campos de los cursos del dataset
Título del curso
URL del curso
Imagen del curso
Resumen del curso
Calificación y número de reseñas
Duración y cantidad de lecciones
Nivel de dificultad
Precio

Table 5.4: Tabla de campos de los cursos del dataset

Antes de realizar cualquier análisis, los datos se preprocesan para asegurarnos de que no hay valores nulos y que todos los campos necesarios están completos. Además, se filtraron los cursos cuya valoración media fuera mayor o igual a 4.0 para asegurar que los cursos recomendados fueran de alta calidad.

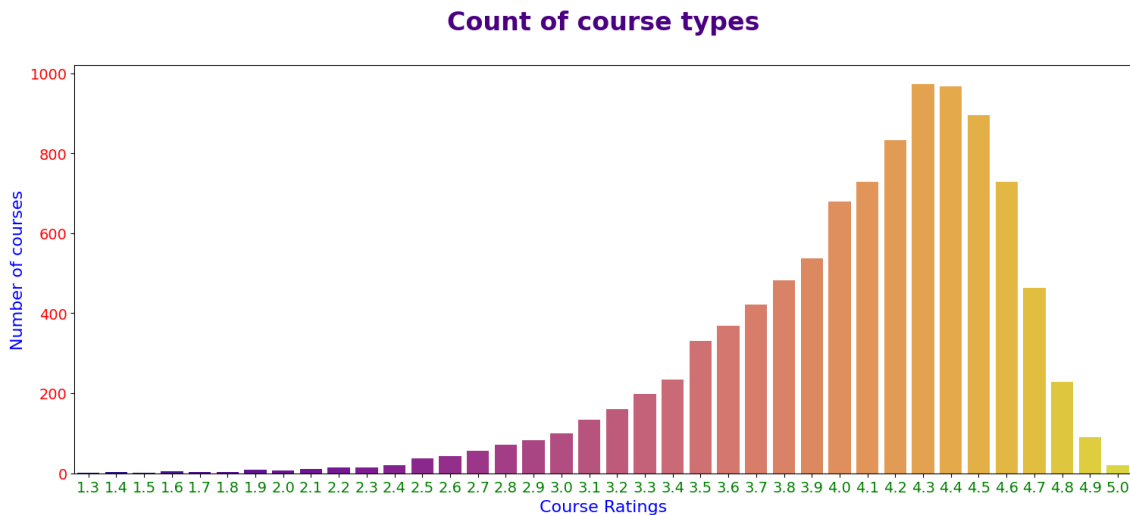


Figure 5.38: Contador de cursos por valoraciones

5.5.1.3 Vectorización del Texto

Utilizamos el vectorizador TF-IDF (Term Frequency-Inverse Document Frequency) para convertir las descripciones de los cursos y las preferencias del usuario en vectores

CHAPTER 5. SISTEMA DESARROLLADO

numéricos. Este proceso permite que el texto sea analizado y comparado mediante cálculos matemáticos.

5.5.1.4 Cálculo de Similitud

Aplicamos la medida de similitud de coseno para calcular la similitud entre el vector de preferencias del usuario y los vectores de los cursos disponibles. Esto nos permite identificar los cursos que son más similares a lo que el usuario busca. La similitud del coseno se calcula utilizando la siguiente fórmula:

$$\text{Similitud del Coseno} = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

Para entender mejor el funcionamiento del cálculo de la similitud del coseno, se presenta un ejemplo visual en el que se estudia la relación de Jaime con diferentes productos 5.39



Figure 5.39: Ejemplo de similitud del coseno

5.5.1.5 Generación de Recomendaciones

El sistema de recomendación de cursos de CodeLingo utiliza una técnica basada en la similitud de coseno para seleccionar los cursos que mejor se adaptan al perfil del usuario. Este perfil se construye a partir de una serie de preferencias que el usuario proporciona, tales como su nivel de experiencia, objetivos de aprendizaje, lenguajes de programación y frameworks preferidos, áreas de interés, formato de curso deseado y duración del curso. El sistema analiza estas preferencias y calcula la similitud entre el perfil del usuario y la descripción de los cursos disponibles. Los cursos con la mayor similitud son seleccionados y presentados al usuario ordenados de mayor a menor compatibilidad.

Cada recomendación incluye el título del curso, la URL para acceder al curso y una imagen representativa del curso. A continuación se presentan dos ejemplos de generación de recomendaciones y una comparación entre ellos:

Ejemplo 1 En este ejemplo, las preferencias del usuario son las siguientes:

```
{
  "experience": "Avanzado",
  "learning_objective": "Mejorar habilidades actuales",
  "languages": ["Python"],
  "frameworks": ["React"],
  "areas": ["Desarrollo web"],
  "course_format": "Proyectos basados en casos reales",
  "course_duration": "Más de 10 horas"
}
```

El sistema procesa estas preferencias y selecciona los cursos que tienen mayor similitud con el perfil del usuario. La Figura [5.40](#) muestra los resultados obtenidos.

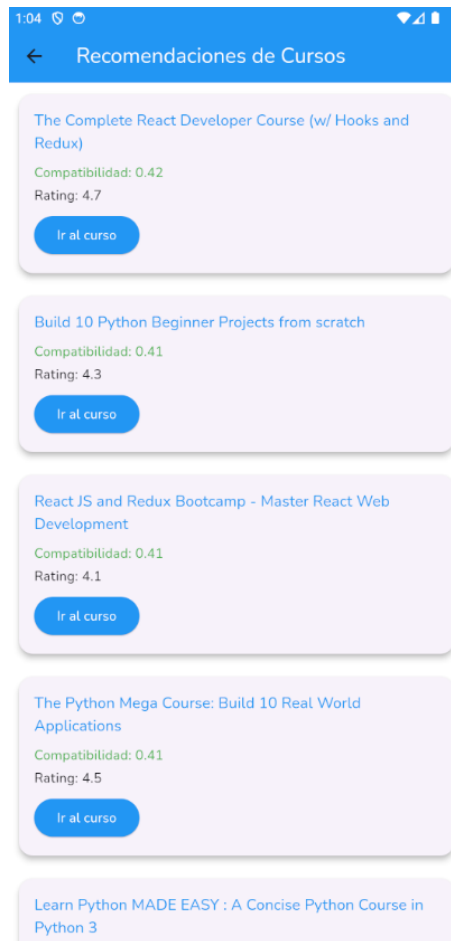


Figure 5.40: Resultado 1 de Recomendación de cursos

Ejemplo 2 En este segundo ejemplo, las preferencias del usuario son las siguientes:

```
{
  "experience": "Intermedio",
  "learning_objective": "Mejorar habilidades actuales",
  "languages": ["Java"],
  "frameworks": ["Spring"],
  "areas": ["Backend"],
  "course_format": "Teoría y práctica combinadas",
  "course_duration": "3-10 horas"
}
```

Nuevamente, el sistema procesa estas preferencias y selecciona los cursos que mejor se alinean con el perfil del usuario. La Figura 5.41 muestra los resultados obtenidos en este caso.

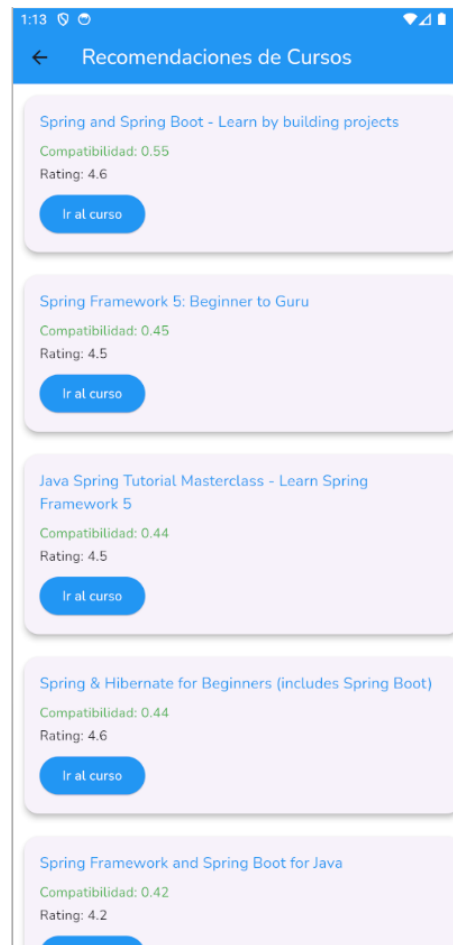


Figure 5.41: Resultado 2 de Recomendación de cursos

Comparación entre Ejemplos Comparando los dos ejemplos, podemos observar que el sistema de recomendación de CodeLingo adapta sus resultados en función de las diferentes preferencias de los usuarios:

- **Ejemplo 1:** Las preferencias del usuario están orientadas hacia un nivel avanzado en Python y React, con un enfoque en el desarrollo web a través de proyectos prácticos de larga duración. Como resultado, los cursos recomendados en la Figura 5.40 son específicos de estas tecnologías y formatos, destacando cursos avanzados y proyectos completos.
- **Ejemplo 2:** En este caso, las preferencias del usuario se centran en un nivel intermedio en Java y Spring, con un enfoque en el desarrollo backend y un formato de curso que combina teoría y práctica. Los cursos recomendados en la Figura 5.41 reflejan estas preferencias, presentando cursos de nivel intermedio en Spring y Java, con un enfoque teórico-práctico.

CHAPTER 5. SISTEMA DESARROLLADO

Esta comparación demuestra la capacidad del sistema para ajustar las recomendaciones según el perfil individual del usuario, ofreciendo cursos que se alinean estrechamente con sus objetivos de aprendizaje y preferencias tecnológicas.

5.5.1.6 Interfaz de Usuario

Las recomendaciones se muestran en una interfaz amigable, similar a la de plataformas de streaming como Netflix, donde el usuario puede ver los cursos recomendados de manera atractiva y organizada. Al hacer clic en un curso, el usuario es redirigido a la página del curso en cuestión.

5.5.2 Beneficios del Sistema de Recomendación

- **Personalización:** Cada usuario recibe recomendaciones únicas basadas en sus intereses y experiencia.
- **Eficiencia:** El proceso de recomendación es rápido y eficiente, permitiendo al usuario encontrar cursos relevantes sin necesidad de navegar manualmente por una lista extensa.
- **Escalabilidad:** El sistema puede manejar grandes volúmenes de datos y usuarios, siendo adecuado para plataformas de aprendizaje de gran escala.

CHAPTER 5. SISTEMA DESARROLLADO

5.5.3 Sincronización con Heroku

Para gestionar y desplegar el sistema de recomendación, se ha utilizado Heroku como plataforma de alojamiento. Heroku permite implementar aplicaciones web de manera rápida y eficiente, integrándose fácilmente con la infraestructura de Firebase.

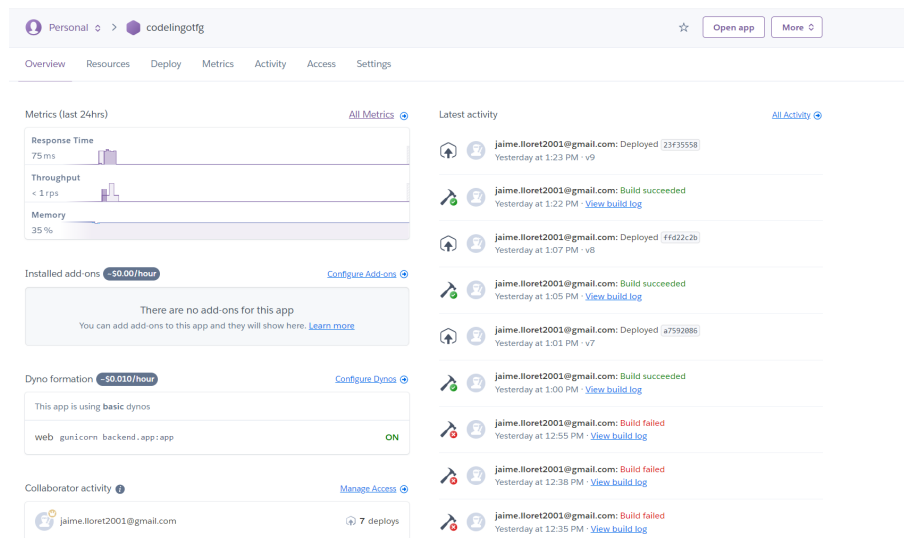


Figure 5.42: Implementación del sistema en Heroku

5.5.4 Datasets y Limpieza de Datos

El dataset utilizado para los cursos se obtuvo de Kaggle y se preprocesó para asegurar la calidad de los datos. El proceso de limpieza incluyó la eliminación de valores nulos, la normalización de los textos y la transformación de las descripciones en vectores TF-IDF. Solo se utilizaron cursos con una valoración media mayor o igual a 4.0 para garantizar la calidad de las recomendaciones. Además, durante el proceso de datos se generaron varios resultados, como la distribución de los cursos por nivel de dificultad.

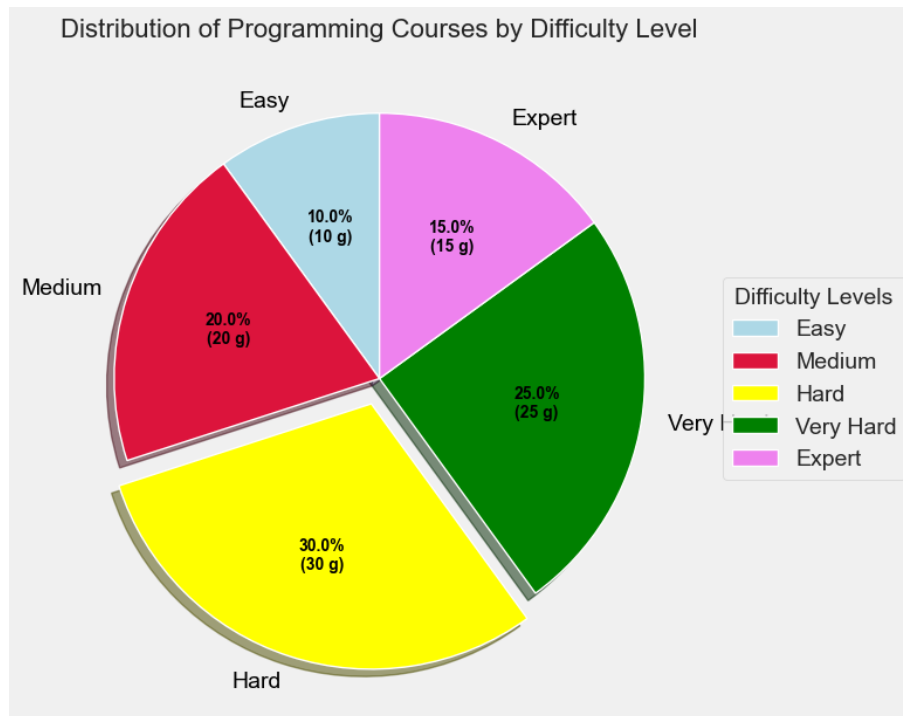


Figure 5.43: Distribución de los cursos de programación por nivel de dificultad

```
1 import pandas as pd
2 from sklearn.feature_extraction.text import TfidfVectorizer
3
4 # Cargar el dataset
5 df = pd.read_csv('udemy_tech.csv')
6
7 # Eliminar valores nulos
8 df.dropna(inplace=True)
9
10 # Filtrar cursos con valoraciones >= 4.0
11 df = df[df['rating'] >= 4.0]
12
13 # Vectorizar las descripciones
14 vectorizer = TfidfVectorizer(stop_words='english')
15 X = vectorizer.fit_transform(df['course_description'])
```

Listing 5.7: Ejemplo de preprocesamiento de datos

Este proceso garantiza que los datos estén listos para ser utilizados en el sistema de recomendación, proporcionando resultados precisos y relevantes para los usuarios de CodeLingo.

Chapter 6

Estudio Económico

6.1 Modelo de negocio

Atendiendo al modelo de las 3 C's, el modelo de negocio de Codelingo se fundamenta en tres pilares esenciales: creación, captura y configuración de valor. Estos pilares son representados de manera en el diagrama presentado en la Figura 6.1.

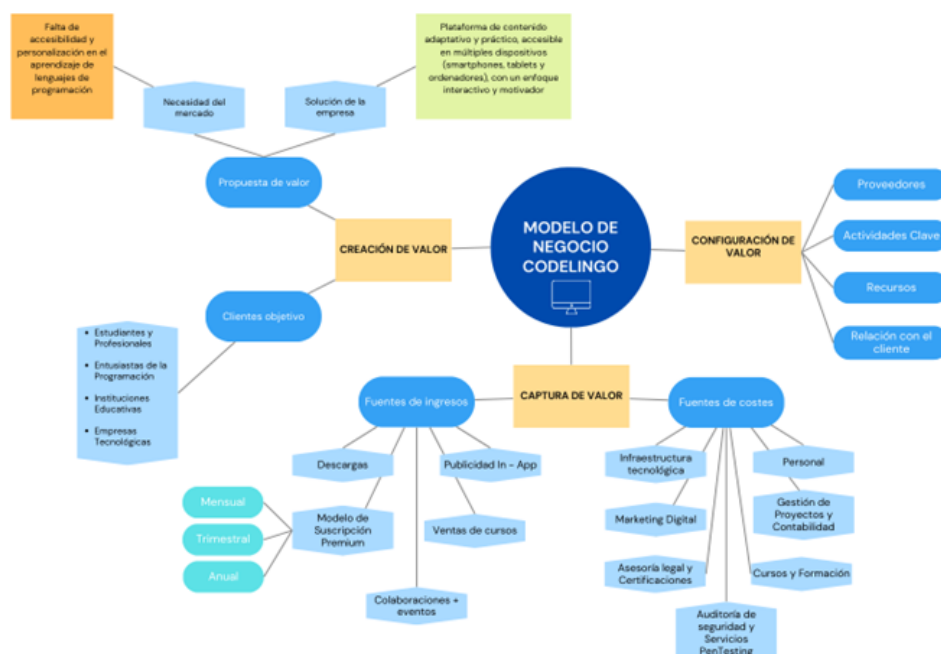


Figure 6.1: Modelo de Negocio de Codelingo

La creación de valor en Codelingo se enfoca en ofrecer una propuesta de valor atractiva y diferenciada. Esta propuesta responde a una necesidad del mercado: la

CHAPTER 6. ESTUDIO ECONÓMICO

falta de accesibilidad y personalización en el aprendizaje de lenguajes de programación. Codelingo aborda esta necesidad mediante una plataforma de contenido adaptativo y práctico, accesible desde múltiples dispositivos (smartphones, tablets y ordenadores), con un enfoque interactivo y motivador.

Además, la propuesta de valor se dirige a un público objetivo específico, que incluye estudiantes y profesionales, entusiastas de la programación, instituciones educativas y empresas tecnológicas. Este enfoque segmentado permite que la plataforma se ajuste a las necesidades particulares de cada grupo, aumentando así su efectividad y atractivo.

Un ejemplo de cómo Codelingo crea valor es a través de su interfaz intuitiva y atractiva, que facilita el uso y la navegación, mejorando la experiencia del usuario y fomentando el aprendizaje continuo.

La captura de valor en Codelingo se realiza mediante diversas fuentes de ingresos bien definidas, las cuales se explicarán en detalle en los siguientes apartados del capítulo. Estas incluyen ingresos por descargas, suscripciones premium (mensuales, trimestrales y anuales), publicidad in-app, ventas de cursos, colaboraciones y eventos.

La configuración de valor se centra en los recursos y actividades clave necesarios para entregar la propuesta de valor de manera eficiente y efectiva. En Codelingo, esto incluye una infraestructura tecnológica robusta, marketing digital, asesoría legal y certificaciones, y actividades de auditoría de seguridad y servicios de pentesting.

El personal juega un papel fundamental en esta configuración, con roles esenciales como desarrolladores frontend y backend, diseñadores UX/UI, especialistas en marketing y soporte técnico. Estos profesionales garantizan que la plataforma funcione sin problemas y continúe evolucionando para satisfacer las necesidades de los usuarios.

Las relaciones con los proveedores, actividades clave y la gestión de proyectos y contabilidad también son componentes vitales en esta configuración, asegurando que la operación de Codelingo sea eficiente y alineada con sus objetivos estratégicos. Las principales actividades de la App son:

- Desarrollo de software: crear y mantener la plataforma utilizando tecnologías modernas.
- Creación de contenidos: desarrollo de elecciones, ejercicios y tutoriales de alta calidad.
- Marketing y publicidad
- R & D: mejorar continuamente la plataforma mediante actualizaciones.
- Soporte al cliente: proporcionar soporte técnico y educativo a los usuarios.

CHAPTER 6. ESTUDIO ECONÓMICO

En cuanto a la relación con los clientes, Codelingo busca recoger opiniones, feedback y aportaciones de los usuarios con la intención de mejorar la experiencia de aprendizaje a largo plazo.

El análisis del modelo de negocio de Codelingo revela una estrategia bien articulada que integra la creación, captura y configuración de valor. La plataforma no solo responde a una necesidad significativa en el mercado, sino que también ofrece múltiples canales de ingresos y una estructura operativa sólida. Estos elementos combinados posicionan a Codelingo como una oferta atractiva tanto para usuarios como para inversores, con un potencial de crecimiento y sostenibilidad en el competitivo sector de la educación tecnológica.

6.1.1 Propuesta de valor

A continuación, se presenta una ilustración detallada de la propuesta de valor de Codelingo:

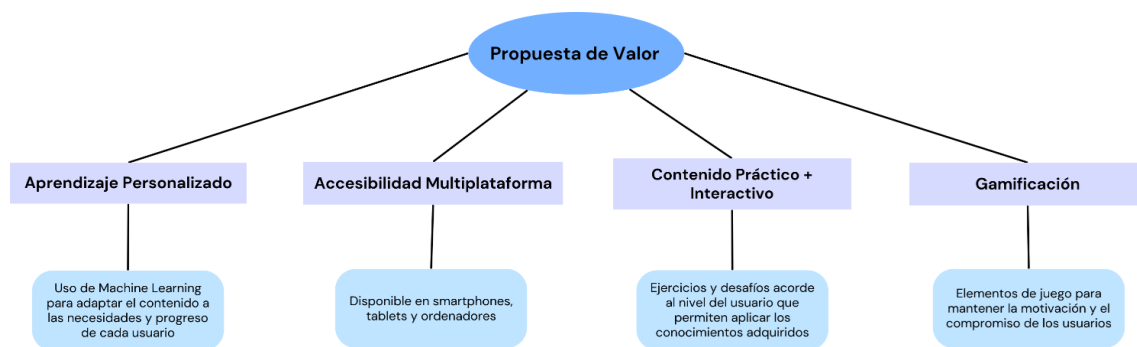


Figure 6.2: Propuesta de Valor de Codelingo

6.2 Análisis DAFO

En la tabla 6.1 se presenta una análisis DAFO de Codelingo:

FORTALEZAS	DEBILIDADES
• Enfoque personalizado • Interfaz intuitiva y atractiva • Enfoque en la práctica y la colaboración • Flexibilidad y accesibilidad (BYOD) • Base de Datos de Contenidos Inicial	• Recursos financieros y humanos • Funcionalidades avanzadas de AI • Reconocimiento de marca inicial limitado
OPORTUNIDADES	AMENAZAS
• Creciente demanda de programadores • Colaboraciones y alianzas con instituciones educativas y empresas tecnológicas • Alta innovación en contenidos • Expansión internacional	• Competencia de mercado • Adaptación tecnológica • Cambios en las políticas de las plataformas móviles como App-store o Google Play

Table 6.1: Análisis DAFO

6.3 Estimación Económica

En cuanto a la estimación económica, se ha elaborado un business case con su respectivas hoja de ingresos, hoja de costes y cuenta de resultados con el fin de estudiar la viabilidad económica del proyecto.

Se presentarán una serie de resultados asociados a un escenario objetivo o caso más probable, que más adelante será comparado con un escenario conservador (pesimista) y con otro algo más optimista.

CHAPTER 6. ESTUDIO ECONÓMICO

6.3.1 Precisión de Parámetros & Fuentes de ingresos y de costes

A continuación se muestra un conjunto de parámetros que se han empleado como herramientas de cálculo para estimar los ingresos y costes que conforman el business case elaborado:

PARAMETROS DE INGRESO	CUANTÍA
Precio de descarga	5,00 €
Numero medio de descargas al mes en año 1	1000
Porcentaje de incremento anual de descargas	35%
Precio por suscripción mensual	9,99 €
Precio por suscripción trimestral	27,99 €
Precio por suscripción anual	99,99 €
Porcentaje de la distribucion de suscriptores plan mensual	60%
Porcentaje de la distribucion de suscriptores plan trimestral	30%
Porcentaje de la distribucion de suscriptores plan anual	10%
Porcentaje de conversión a premium año 1 -2	5,1%
Porcentaje de conversión a premium año 3	6,8%
Porcentaje de conversión a premium año 4	6,8%
Porcentaje de conversión a premium año 5	10%
Retención promedio de la suscripción mensual (meses)	6
Retención promedio de la suscripción trimestral (trimestres)	2
Retención promedio de la suscripción anual (anual)	1
Usuarios activos mensuales en la app en año 1	3000
Porcentaje de incremento anual de usuarios activos	10%
Número de veces que entra un usuario a la app al mes	60
Ingreso por cada mil impresiones de anuncios	15,00 €
Probabilidad de aparición de anuncio por entrada a la app	80,00%
Precio promedio de venta de un curso o contenido adicional en año 1	4,99 €
Precio promedio de venta de un curso o contenido adicional en años 2 -5	19,99 €
Porcentaje de usuarios que compran cursos adicionales	3,0%
Precio de inscripción para eventos y talleres año 1	5,00 €
Precio de inscripción para eventos y talleres años 2-5	10,00 €
Número de eventos organizados al mes	2
Número promedio de participantes por evento	30
Porcentaje de incremento anual de participantes por evento	5%
Comisión media obtenida por colaboraciones con empresas	500,00 €
Número de acuerdos de colaboración realizados al año	5
Incremento de número de acuerdos por año	2n -2

CHAPTER 6. ESTUDIO ECONÓMICO

Entre las fuentes de ingresos de Codelingo, distinguimos las siguientes categorías:

- Ingresos por descargas
- Ingresos por suscripciones premium mensuales
- Ingresos por suscripciones premium trimestrales
- Ingresos por suscripciones premium anuales
- Ingresos por publicidad In - App
- Ingresos por ventas de cursos
- Ingresos por colaboraciones
- Ingresos por eventos

Es importante mencionar que todos los valores asignados a los parámetros en la tabla anterior son estimaciones derivadas de información disponible en línea sobre aplicaciones EdTech existentes y de la lógica del mercado.

Establecer un precio de descarga de 5€ se considera un valor adecuado, puesto que el precio de la mayoría de aplicaciones Edtech del mercado oscila entre los 4€ y los 10€ (excluyendo compras In - App). Se espera que el número de descargas al mes crezca anualmente en un 35% atendiendo al potencial del proyecto para ganar visibilidad en el mercado. A medida que el marketing digital y la experiencia compartida entre usuarios aumentan, se espera que así lo haga el índice de descargas.

Los usuarios tendrán la opción de suscribirse a un plan premium de la aplicación mediante tres alternativas diferentes (plan mensual, trimestral o anual) de acuerdo a sus preferencias. Este plan les posibilitará disfrutar de múltiples beneficios exclusivos como practicar errores cometidos en las lecciones en una sección especializada para ello, la eliminación de anuncios, mayor oferta de recompensas en el menú de desafíos en relación a los logros alcanzados, etc.

Otra notable fuente de ingresos una vez el proyecto avance será la publicidad In -App, en la cual se anotarán alrededor de 15€ de ingresos por cada mil impresiones de anuncios por cada vez que un usuario entre a la aplicación, que se estima ser de 60 veces al mes. La predicción de la probabilidad de que un anuncio aparezca en la app por cada entrada de usuario es de un 80%.

Por último, una parte de los ingresos provendrá de las comisiones obtenidas a raíz de la venta de cursos del sistema de recomendación de Codelingo (pertenecientes a plataformas ya mencionadas como Udemy y Coursera), además de las colaboraciones de la app con otras empresas e ingresos obtenidos de los eventos organizados. Cabe destacar que la empresa trazará colaboraciones con otras EdTechs del sector como FreeCode Camp y Codecademy, aumentando anualmente el número de

UNIVERSIDAD PONTIFICIA COMILLAS
Escuela Técnica Superior de Ingeniería (ICAI)
Grado en Ingeniería en Tecnologías de Telecomunicación

CHAPTER 6. ESTUDIO ECONÓMICO

alianzas acorde con su crecimiento y expansión. Asimismo, Codelingo tiene la intención de llevar a cabo eventos como Hackathons temáticos, contests como los de Codeforces o talleres intensivos / conferencias educativas.

PARAMETROS DE COSTE	CUANTÍA
Personal:	
Frontend Developer	45.000,00 €
Backend Developer	60.000,00 €
Full-stack Developer	60.000,00 €
Marketing specialist	35.000,00 €
Soporte técnico	30.000,00 €
UI/UX Designer	45.000,00 €
QA y Testers	- €
Product Manager	60.000,00 €
Infraestructura tecnológica: (AWS, Google Cloud, Azure...)	
Coste/hora de computación en la nube	0,10 €
Número inicial de instancias	1
Incremento del número de instancias al año	1
Número de horas en un mes	720
Coste/GB al mes de almacenamiento en la nube	0,023 €
Numero de GB de almacenamiento	500
Coste/GB al mes de transferencia de los datos en la nube	0,12 €
Número de GB de transferencia	200
Coste por hora Balanceador de carga AWS ELB	0,0225 €
Coste por GB de Datos procesados Balanceador de carga	0,008 €
Datos en GB procesados mensualmente año 4	1000
Datos en GB procesados mensualmente año 5	2000
Implementación de Redis	
Coste de la instancia por hora	0,02 €
Coste configuración y pruebas	1.000,00 €
Coste mantenimiento anual	500,00 €
Coste al mes Hosting	30,00 €
Coste al año Dominio	15,00 €

UNIVERSIDAD PONTIFICIA COMILLAS
Escuela Técnica Superior de Ingeniería (ICAI)
Grado en Ingeniería en Tecnologías de Telecomunicación

CHAPTER 6. ESTUDIO ECONÓMICO

Coste por hora de instancia base de datos	0,25 €
Coste por GB de almacenamiento de Base de Datos	0,10 €
Número de GB de almacenamiento en base de datos	500,00 €
Coste por llamada de la API del Asistente de Inteligencia Artificial	0,002 €
Número al mes de llamadas a la API	50000
Coste por cada mil mensajes de notificaciones Push	1,00 €
Estimación mensual de mensajes	5000
Coste por cada GB al mes de Almacenamiento de contenidos Multimedia	0,023 €
Coste por cada GB de transferencia de contenidos Multimedia	0,12 €
Estimación almacenamiento en GB mensual de contenidos Multimedia	100
Estimación de GB de transferencia mensual de contenidos Multimedia	50
Coste anual del certificado SSL	60,00 €
Coste mensual de protección DDoS	20,00 €
Coste mensual de gestión de monitoreo y alertas	50,00 €
Coste por GB de gestión de Logs	2,00 €
Marketing y publicidad:	
Coste mensual Marketing digital: Google Ads, Facebook Ads, etc.	1.000,00 €
Camapañas redes sociales	500,00 €
Coste mensual de Eventos y conferencias	200,00 €
Incremento anual de los costes de marketing	10,00%
Operaciones y Administración:	
Coste mensual de gestión de proyectos y contabilidad	500,00 €
Costes legales y de Compliance:	
coste mensual de asesoría legal	200,00 €
Coste anual Certificación (ISO 27001, GDPR Compliance)	2.500,00 €
Formación y desarrollo del personal:	
Coste mensual de Cursos y conferencias	500,00 €
Ciberseguridad:	
Coste de auditoría de seguridad	3.000,00 €
Número de auditorías de seguridad en año 1 y 2	1
Número de auditorías de seguridad en año 3, 4 y 5	2
Servicios de PenTesting	4.000,00 €
Impuesto de sociedades (% sobre ingresos)	25%
Tipo de interés (%)	2%

CHAPTER 6. ESTUDIO ECONÓMICO

Entre las fuentes de costes de Codelingo, distinguimos las siguientes secciones principales:

- Costes de personal
- Costes de infraestructura tecnológica
- Costes de marketing digital / campañas en redes sociales
- Costes de operaciones y administración
- Costes legales y de Compliance
- Costes de cursos y conferencias
- Costes de ciberseguridad

En cuanto al personal, la empresa empezará contando con un Full -stack developer y un UX/UI designer. A medida que crece el proyecto, a partir del año 2, se sumarán a la plantilla Frontend y Backend developers, encargados de soporte técnico, Product managers y Marketing Specialists.

Asimismo, existen una serie de costes asociados a la infraestructura tecnológica de la aplicación:

- Servicios en la nube
- Hosting y dominios
- Bases de datos
- Servicios de Asistente de Inteligencia Artificial
- Servicios de Notificaciones Push
- Almacenamiento de Contenidos Multimedia
- Certificados SSL y Seguridad
- Monitoreo y Logística

La infraestructura tecnológica de Codelingo está diseñada para escalar con el crecimiento del usuario, utilizando servicios en la nube que aseguran la disponibilidad y la seguridad de los datos. Cabe destacar que, a partir del cuarto año, se ha decidido implementar un balanceador de carga cuya funcionalidad te permite escalar conforme aumentan los usuarios y el tráfico, y facilita el balanceo automático y sencillo durante los picos de tráfico inesperados e instantáneos.

CHAPTER 6. ESTUDIO ECONÓMICO

Dada la importancia del marketing digital y el predominio de la publicidad en línea frente a otros recursos publicitarios en la actualidad, la empresa pretende invertir en servicios SEO a través de palabras clave y backlinks además de impulsar campañas en redes sociales como Facebook, Instagram o LinkedIn.

Añadido a lo anterior se encuentran los costes correspondientes a la formación y desarrollo del personal (desarrolladores de software) , costes de gestión de proyectos y contabilidad y costes asociados a asesorías legales y consecución de certificados (ISO27001, GDPR Compliance).

Por último, se llevará a cabo una auditoría de seguridad informática anual durante los dos primeros años del proyecto y dos de ellas a partir del tercer año, con intención de conocer el estado de la protección de los activos de Codelingo en todo momento. Asimismo, se destinará una cantidad fija a la contratación de servicios PenTesting para analizar las potenciales vulnerabilidades de la infraestructura tecnológica.

En el Anexo D se incluyen las tablas generadas con la herramienta de Microsoft Excel en las cuáles se detalla el cálculo de todas las fuentes de ingresos y costes comentados anteriormente. Adicionalmente, se adjunta la tabla de cuenta de resultados con los correspondientes márgenes anuales de la empresa.

6.3.2 Análisis de resultados

En el escenario base, la evolución de los ingresos a lo largo de la duración del proyecto tiene la siguiente forma:

CHAPTER 6. ESTUDIO ECONÓMICO

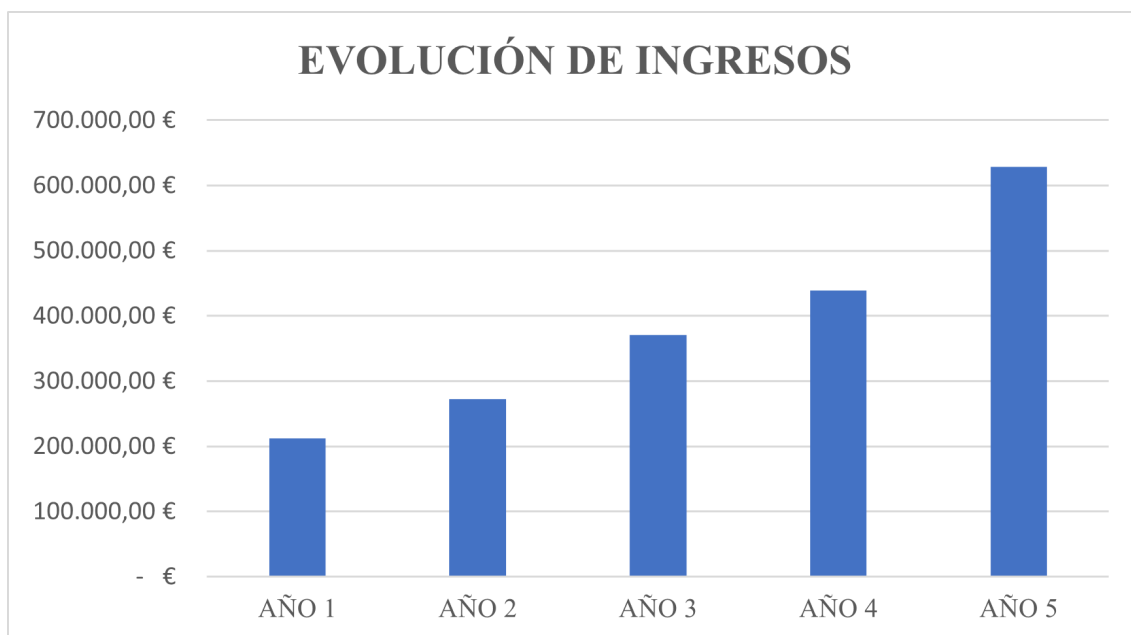


Figure 6.3: Gráfico de barras de la evolución de ingresos del proyecto

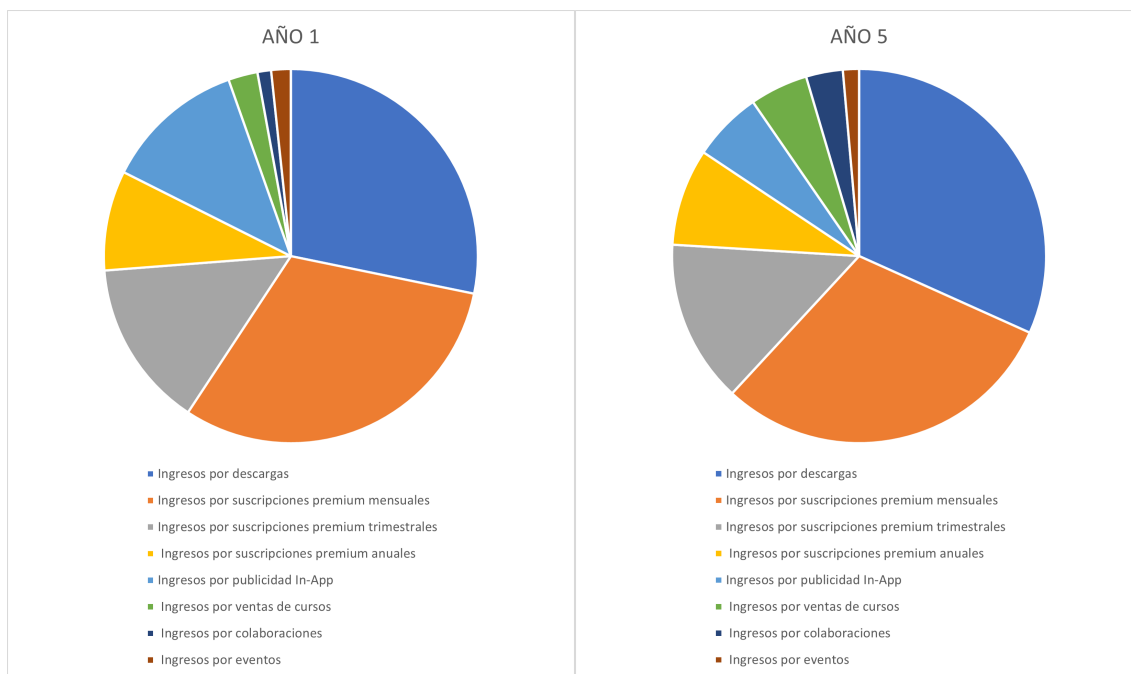


Figure 6.4: Comparativa de gráficos circulares de las fuentes de los ingresos del año 1 y 5

El importe de ingresos que obtiene la app de forma anual crece con una tendencia lineal, pasando de generar 212.631,05€ en el primer año a un total de 628.377,33€

CHAPTER 6. ESTUDIO ECONÓMICO

en el quinto año de operación, cuya evolución corresponde a un porcentaje de crecimiento del 195,5%. Este elevado incremento en las ganancias estimadas del proyecto se debe, principalmente, al incremento del 35% anual en el índice de descargas y en el aumento de conversión a premium por parte de los usuarios (la cifra estimada fluctúa de un 5,1% en los años 1 y 2 a un 6,8% en los años 3 y 4 y a un 10% en el año 5). Asimismo, se ha valorado que el porcentaje de aumento de los usuarios activos mensuales sea de un 10% al año, lo que aporta un significativo auge en las ganancias dados los pronósticos de venta de cursos y eventos dirigidos a este grupo de interés.

Por otro lado, cabe destacar que las principales fuentes de ingresos representan un porcentaje similar del total entre el primer y el quinto año. Uno de los cambios más significativos se encuentra en la publicidad In-App, ya que esta sección pasa de conformar un 12 % del total a un 6% del mismo, esto se debe al número de usuarios nuevos que comienzan a pagar por los servicios de la aplicación, los cuales suponen más ganancias que la publicidad.

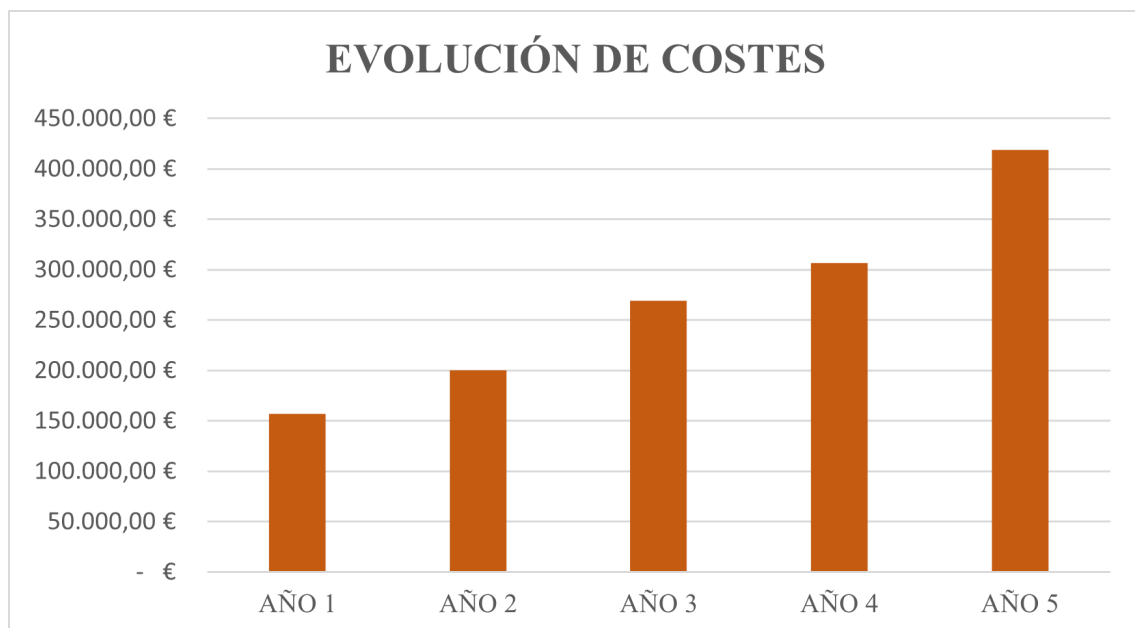


Figure 6.5: Gráfico de barras de la evolución de costes del proyecto

CHAPTER 6. ESTUDIO ECONÓMICO



Figure 6.6: Comparativa de gráficos circulares de las fuentes de los costes del año 1 y 5

En cuanto a los costes incurridos por la App, crecen de forma casi lineal, pasando de una cifra de 157.184,60€ en el primer año a un total de 418.727,21€ en el quinto año de operación, cuya evolución corresponde a un porcentaje de crecimiento del 166,4%. Este elevado incremento en los costes estimados del proyecto se debe, esencialmente, al incremento de la plantilla de trabajadores de la App como desarrolladores Frontend y Backend o Product Managers, que se eleva desde un 67% a un 80%. Por otro lado, el peso de la infraestructura tecnológica se mantiene constante ya que sigue el mismo patrón de crecimiento que el nivel de usuarios activos, mientras que el coste de Marketing Digital, Campañas en redes sociales y Eventos baja relativamente con relación al resto de costes. Esto se debe a que en el año 1 se da más importancia a la inversión en Marketing para dar la App a conocer, en cambio en el año 5 ya se cuenta con un gran número de usuarios activos y otras fuentes de costes cobran mayor importancia.

	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Beneficio antes de impuestos (euros)	55.446	72.480	101.470	131.927	209.650
Impuestos	13.862	18.120	25.367	32.982	52.413
Beneficio despues de impuestos (euros)	41.585	54.360	76.102	98.945	157.238

CHAPTER 6. ESTUDIO ECONÓMICO

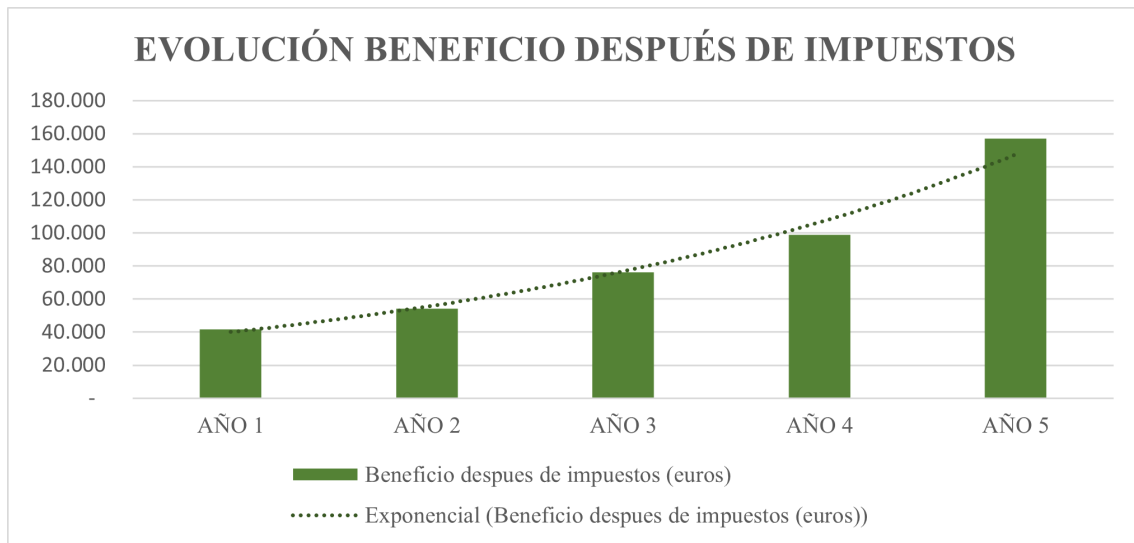


Figure 6.7: Evolución de los beneficios después de impuestos en 5 años

La evolución de beneficios después de impuestos podría ser una métrica muy atractiva para los inversores del proyecto. Se trata de una evolución muy positiva a lo largo de los años resultando en un crecimiento del 278,11 % del quinto año respecto al primero. Esto refleja que, a lo largo del transcurso del proyecto, el impacto de las principales fuentes de ingresos es mayor frente al impacto de los agentes de costes más significativos. Asimismo, otro indicador interesante para los inversores es el del VAN. Los flujos estimados para el proyecto dan lugar a un VAN de 398.556,44€, haciendo de Codelingo un proyecto viable y rentable en el largo plazo.

Chapter 7

Análisis de Resultados

7.1 Conclusiones sobre la metodología

La metodología empleada en el desarrollo del proyecto Codelingo ha presentado tanto aciertos como áreas de mejora. Uno de los aspectos críticos del planteamiento ha sido la dedicación inicial al estudio exhaustivo de las tecnologías y frameworks a utilizar. Un análisis más profundo de las tecnologías desde el comienzo podría haber permitido implementar inicialmente herramientas de desarrollo, y progresar hacia tecnologías de producción conforme avanzaba el proyecto. Este enfoque podría haber optimizado los tiempos y mejorado la calidad del producto final.

Sin embargo, la metodología también ha tendido aspectos positivos significativos. La organización del trabajo en sprints ha permitido una distribución efectiva de las tareas y ha facilitado el seguimiento del progreso. La estrategia de desarrollo, comenzando con el frontend, seguido de la integración con el backend, la implementación del servicio de chat GPT-3.5 Coddy, y finalmente el sistema de recomendación de cursos, ha demostrado ser eficiente. Este enfoque secuencial permitió abordar los desafíos técnicos de manera estructurada asegurando la cohesión del sistema. Para incorporar estas funcionalidades, se ha seguido, en todo momento, el principio de encapsulación. Primero, se ha desarrollado y probado cada funcionalidad por separado en proyectos más simples. Luego, se ha trasladado la lógica de cada una al proyecto principal. De esta manera, al integrar las nuevas funcionalidades en el proyecto grande, se asegura que cada una funcione correctamente de forma independiente y que su integración no afecte la lógica general del proyecto.

7.2 Conclusiones sobre las Tecnologías

El uso de tecnologías específicas como Flutter, Dart, Firebase o Heroku en el proyecto Codelingo ha sido fundamental para alcanzar los objetivos planteados. No obstante, una implementación completa del despliegue de la aplicación no se ha

CHAPTER 7. ANÁLISIS DE RESULTADOS

llevado a cabo. Este despliegue habría permitido obtener resultados más precisos de pruebas y feedback de los usuarios, además de facilitar la implementación de servicios adicionales de Firebase para análisis y estadísticas.

El despliegue total implicaría la configuración de entornos de producción, la gestión de bases de datos en tiempo real y la integración de herramientas de análisis como Firebase Analytics. Estos cambios habrían proporcionado una visión más clara del uso y rendimiento de la aplicación, permitiendo un análisis mejorado de las pruebas unitarias con el usuario y una mejora en su experiencia final.

7.3 Análisis de los Resultados

El modelo utilizado para el asistente de chat, GPT-3.5-turbo, se seleccionó por su alto rendimiento en tareas de procesamiento de lenguaje natural. Comparado con otros modelos disponibles en el mercado, GPT-3.5-turbo ofrece una combinación equilibrada de precisión y eficiencia. Estudios recientes han demostrado que este modelo es capaz de generar respuestas coherentes y contextualmente relevantes, como dice Oriol Zertuche, CEO de Cody AI[21], estos son algunos de sus casos de uso:

1. Atención al cliente para ofrecer respuestas precisas a las consultas de los clientes
2. Generación de contenidos para elaborar artículos e informes de alta calidad y bien fundamentados.
3. Análisis de datos para ayudar a interpretarlos y comprenderlos.
4. Herramientas educativas para crear material didáctico interactivo
5. Ayuda a la investigación para acelerar la recuperación de información
6. Ayuda a la programación para generar fragmentos de código y soluciones
7. Diagnósticos médicos para ayudar a los profesionales sanitarios
8. Revisión de documentos jurídicos para analizar eficazmente textos jurídicoslo que lo hace ideal para

La implementación del sistema de recomendación de cursos basado en el cálculo del coseno de similitud de los vectores de preferencias de los usuarios con las descripciones de los cursos también han mostrado resultados prometedores. Esta técnica, utilizada por grandes plataformas de recomendación como Amazon y Netflix, es efectiva en cuanto a la personalización de contenidos. Según el artículo de [JAY-WRKR Tech](#), el cálculo del coseno del ángulo entre el vector de perfil del usuario y el vector de elementos es una metodología sólida para determinar la similitud entre preferencias y productos.

7.4 Recomendaciones para futuros trabajos

De cara a mejorar y expandir CodeLingo para elaboraciones futuras, se sugieren varias recomendaciones:

Adopción de Tecnologías de Producción: Es esencial investigar y adoptar frameworks y tecnologías que estén diseñadas específicamente para entornos de producción. Por ejemplo, Django, conocido por su robustez y amplio soporte, podría reemplazar a Flask para ofrecer una mayor escalabilidad y manejo de carga. Además, el uso de servicios de contenedores como Docker puede facilitar la gestión de despliegues de manera más eficiente, asegurando un entorno más controlado y replicable.

Análisis de Resultados y Feedback del Usuario: Aunque no se logró un despliegue completo, se realizó un análisis preliminar sobre la interacción de los usuarios durante la fase de pruebas. Este análisis reveló un interés significativo en funcionalidades de personalización y métodos de aprendizaje interactivos. Para futuros desarrollos, es crucial implementar herramientas analíticas que proporcionen datos detallados sobre el uso y la satisfacción del usuario, permitiendo realizar ajustes y mejoras basadas en información concreta y precisa.

Mejoras en la Personalización de Lecciones: Siguiendo el ejemplo de aplicaciones como Duolingo, se pueden añadir más opciones de personalización en las lecciones. Utilizar técnicas avanzadas de machine learning puede ayudar a adaptar el contenido al nivel y las preferencias de cada usuario. Implementar Modelos de Lenguaje Grande (LLM) permitirá ofrecer una personalización más profunda y específica en las lecciones, mejorando así la experiencia de aprendizaje.

Ligas y Competencias entre Amigos: Crear ligas donde los usuarios puedan competir con amigos para ganar experiencia y motivarse mutuamente es una estrategia probada en otras plataformas educativas. Este enfoque gamificado fomenta el uso continuo y el compromiso del usuario, aumentando la retención y el disfrute del proceso de aprendizaje.

Desafíos y Creatividad: Los desafíos pueden llevarse a un nivel superior mediante la inclusión de actividades temáticas, colaborativas y en tiempo real. Organizar hackathons virtuales, competencias de codificación en tiempo real y proyectos colaborativos permite a los usuarios trabajar juntos en tareas específicas. Además, introducir recompensas y reconocimientos virtuales puede incentivar la participación y el logro de objetivos.

Optimización y Rendimiento: Es vital mejorar la optimización del rendimiento de la aplicación, especialmente en la gestión de datos y la respuesta de las APIs. Esto incluye optimizar las consultas a bases de datos, utilizar técnicas de caché y implementar balanceadores de carga para manejar mejor el tráfico. Estas medidas asegurarán que la aplicación funcione de manera eficiente, incluso bajo una carga elevada.

Estas recomendaciones están diseñadas para no solo mejorar la funcionalidad

CHAPTER 7. ANÁLISIS DE RESULTADOS

y el atractivo de CodeLingo, sino también para asegurar que la plataforma pueda escalar y adaptarse a las demandas de un entorno de producción profesional. Implementando estas mejoras, CodeLingo puede ofrecer una experiencia de aprendizaje más rica, personalizada y eficaz para todos sus usuarios.

Appendix A

Alineación del Proyecto con los ODS

En el año 2015, la comunidad internacional se unió en un esfuerzo sin precedentes: la Agenda 2030. Esta iniciativa, adoptada por todos los Estados Miembros de las Naciones Unidas, estableció 17 Objetivos de Desarrollo Sostenible (ODS) que buscan alcanzar un futuro más justo y equitativo para todos bajo el lema "*No dejar a nadie atrás*". Entre los principales propósitos de la Agenda 2030 se encuentran la erradicación de la pobreza, la protección del planeta y la reducción de las desigualdades. Cada uno de estos objetivos está compuesto por metas específicas que deben alcanzarse en los próximos diez años.

Para contribuir al cumplimiento de estos objetivos, surge este proyecto. En un mundo cada vez más digitalizado, donde la programación se ha convertido en una habilidad fundamental para el éxito personal y profesional, CodeLingo busca democratizar el acceso a la educación en programación, brindando a todos la oportunidad de adquirir las habilidades necesarias para desenvolverse en la era digital.

Se caracteriza por su enfoque inclusivo y accesible, diseñado para atender las necesidades de usuarios de todos los niveles de experiencia, desde principiantes hasta programadores experimentados. La plataforma ofrece una amplia gama de cursos, herramientas interactivas y recursos personalizados que se adaptan al ritmo de aprendizaje de cada usuario.

Al facilitar el acceso al aprendizaje de programación, CodeLingo contribuye a la igualdad de oportunidades en la era digital. La plataforma permite a personas de cualquier origen socioeconómico, género o ubicación geográfica adquirir las habilidades necesarias para participar activamente en la sociedad digital y acceder a las oportunidades que esta ofrece.

Analizando la descripción de cada uno de los diecisiete objetivos, se ha llegado a la conclusión que los dos que presentan mayor relación con el propósito del desarrollo

APPENDIX A. ALINEACIÓN DEL PROYECTO CON LOS ODS

del proyecto son:



Figure A.1: Objetivos de desarrollo sostenible

ODS4.Educación de calidad:

CodeLingo, se alinea con este objetivo al promover una educación inclusiva, equitativa y de calidad en este ámbito. La plataforma elimina las barreras de acceso mediante cursos gratuitos, una interfaz accesible y recursos en varios idiomas. Además, ofrece una experiencia de aprendizaje personalizada que se adapta al ritmo y necesidades de cada usuario, gracias al uso de Machine Learning. CodeLingo no solo brinda conocimientos, sino que también permite ponerlos en práctica a través de herramientas interactivas, desafíos y proyectos. De esta manera, CodeLingo se convierte en una herramienta fundamental para el desarrollo sostenible y la igualdad de oportunidades en la era digital, permitiendo que todos los individuos puedan adquirir las habilidades necesarias para desenvolverse en este mundo cada vez más tecnológico y alcanzar su máximo potencial.

ODS8. Trabajo decente y crecimiento económico:

Este objetivo busca promover el crecimiento económico sostenido, inclusivo y sostenible, el empleo pleno y productivo y el trabajo decente para todos. En este contexto, la plataforma democratiza el acceso a la educación en programación, habilitando a las personas para adquirir las competencias necesarias para acceder a nuevos empleos en diversos sectores, aumentar la productividad y la competitivi-

APPENDIX A. ALINEACIÓN DEL PROYECTO CON LOS ODS

dad de las empresas, fomentar la inclusión social y la reducción de las brechas económicas, impulsar el desarrollo de nuevas empresas y la creación de productos y servicios innovadores, y asegurar que los usuarios adquieran las habilidades más relevantes para acceder a empleos de calidad en el sector tecnológico. De esta manera, CodeLingo contribuye a un futuro más próspero y equitativo para todos, donde las personas puedan participar activamente en la economía digital y alcanzar su máximo potencial.

ODS10. Reducción de las desigualdades:

Fomenta el desarrollo de soluciones tecnológicas innovadoras que abordan los desafíos que enfrentan las comunidades más pobres, como la falta de acceso a servicios básicos, educación de calidad o atención médica. A través de la inclusión financiera y el acceso a servicios digitales, la plataforma facilita la gestión de recursos y la participación en la economía formal, contribuyendo a la reducción de la pobreza y al empoderamiento de las comunidades. CodeLingo también juega un papel crucial en el fortalecimiento de la gobernanza, promoviendo la transparencia, la rendición de cuentas y la participación ciudadana en la toma de decisiones. Mediante el uso de tecnologías digitales, la plataforma facilita la gestión de datos, la comunicación con la ciudadanía, la consulta pública y el debate, optimizando la eficiencia y la eficacia de la administración pública. Se posiciona como un aliado clave para la reducción de las desigualdades en la era digital, impulsando la inclusión social, el empoderamiento individual, el desarrollo sostenible y la construcción de un mundo más justo y equitativo para todos. La plataforma no solo brinda herramientas y oportunidades, sino que también promueve la colaboración, la innovación y la participación activa de las personas en la construcción de un futuro mejor.

En resumen, este proyecto se alinea con los ODS 4, 8 y 10 al abordar la educación inclusiva y de calidad, así como la reducción de las desigualdades, especialmente en el ámbito de la programación. Al asegurar que la plataforma sea accesible para todos los usuarios, se contribuye a una educación inclusiva, con las mismas posibilidades para todo el mundo garantizando el beneficio de la herramienta y participar plenamente en el proceso educativo

Appendix B

Manual de Instalación

En este Manual de Instalación, se va a explicar paso por paso, los elementos necesarios y la configuración del entorno de trabajo para la aplicación. A lo largo del proyecto ya se detallaron todas las tecnologías a utilizar. A continuación, se muestra cómo instalar los softwares más relevantes.

B.1 Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente ligero pero potente, disponible para Windows, macOS y Linux. Sigue estos pasos para instalarlo:

1. Visita el sitio web oficial de Visual Studio Code en <https://code.visualstudio.com/>.
2. Descarga el instalador correspondiente a tu sistema operativo.
3. Ejecuta el instalador y sigue las instrucciones en pantalla para completar la instalación.
4. Una vez instalado, abre VS Code y navega a la sección de **Extensiones** en la barra lateral izquierda.
5. Instala las siguientes extensiones recomendadas:
 - Dart y Flutter para el desarrollo en Flutter.
 - Python para el desarrollo del backend.

APPENDIX B. MANUAL DE INSTALACIÓN

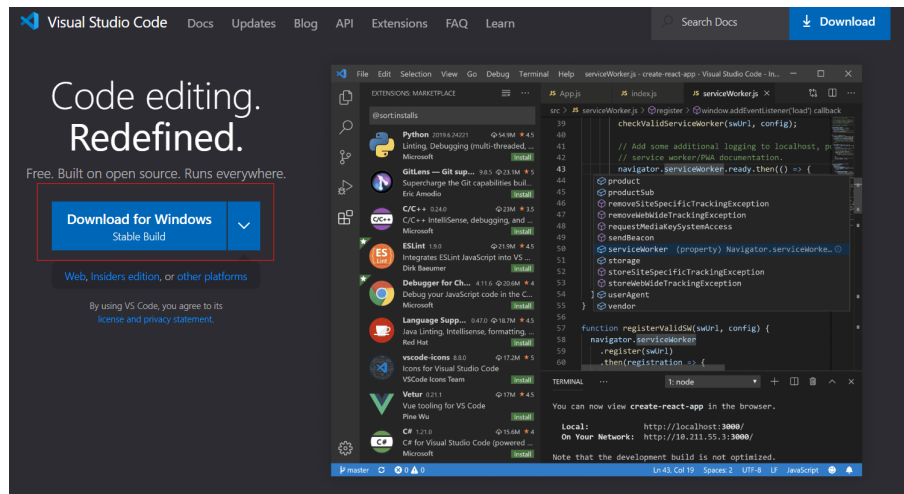


Figure B.1: Página de descarga de Visual Studio Code

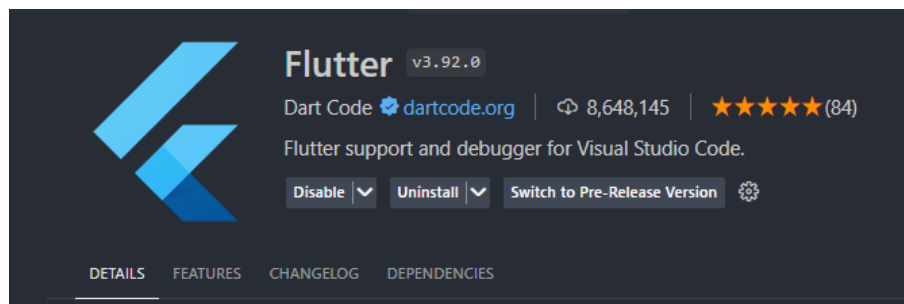


Figure B.2: Página de descarga de Extensión de Flutter

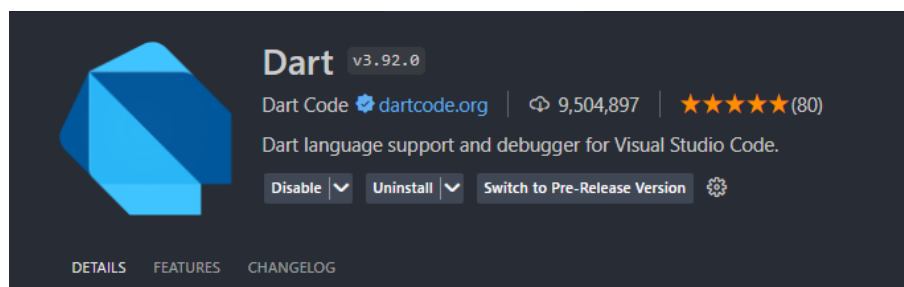


Figure B.3: Página de descarga de Extensión de Dart

B.2 Flutter

Flutter es un framework de código abierto para el desarrollo de aplicaciones nativas. Para instalar Flutter:

1. Visita el sitio web oficial de Flutter en <https://flutter.dev/>
2. Descarga el SDK de Flutter para tu sistema operativo.

APPENDIX B. MANUAL DE INSTALACIÓN

3. Extrae el archivo descargado a la ubicación deseada en tu sistema.
4. Añade Flutter a tu PATH del sistema:
 - En Windows, busca "Variables de entorno" en el menú de inicio y edita las variables de usuario para incluir la ruta al directorio `flutter_bin`.
 - En macOS y Linux, abre tu archivo `.bashrc` o `.zshrc` y añade la línea `export PATH="$PATH:$HOME/flutter/bin"`.
5. Abre una terminal y ejecuta `flutter doctor` para verificar la instalación.

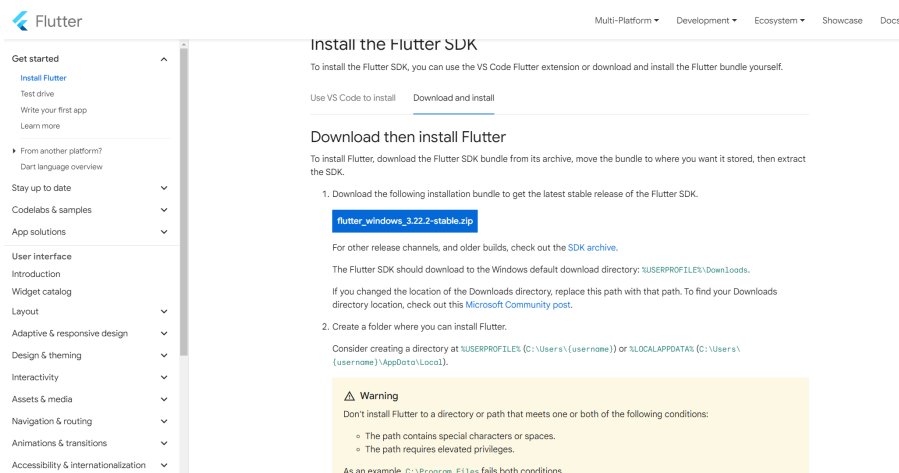


Figure B.4: Página de descarga del SDK de Flutter

B.3 Instalación de un Emulador para iOS

Para desarrollar y probar aplicaciones en iOS, necesitas un Mac con Xcode instalado. Sigue estos pasos:

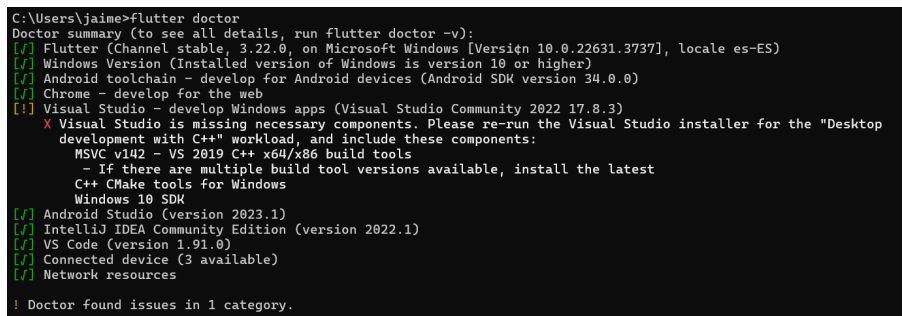
1. Abre la App Store en tu Mac y busca Xcode.
2. Descarga e instala Xcode.
3. Abre Xcode y navega a **Preferences > Locations** para asegurarte de que la línea de comandos de Xcode esté configurada.
4. Abre una terminal y ejecuta `sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer`.
5. Ejecuta `sudo xcodebuild -runFirstLaunch` para completar la configuración.

APPENDIX B. MANUAL DE INSTALACIÓN

B.4 Verificación del Entorno de Desarrollo

Para asegurarte de que todo está correctamente configurado, realiza las siguientes verificaciones:

1. Abre una terminal y ejecuta `flutter doctor`.
2. Revisa la salida para identificar cualquier problema. La salida debe indicar que todos los componentes necesarios están instalados y configurados.
3. Soluciona cualquier problema que aparezca en la salida de `flutter doctor`.



```
C:\Users\jaime>Flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.22.0, on Microsoft Windows [Version 10.0.22631.3737], locale es-ES)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.8.3)
X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++" workload, and include these components:
   - MSVC v142 - VS 2019 C++ x64/x86 build tools
   - If there are multiple build tool versions available, install the latest
   - C++ CMake tools for Windows
   - Windows 10 SDK
[✓] Android Studio (version 2023.1)
[✓] IntelliJ IDEA Community Edition (version 2022.1)
[✓] VS Code (version 1.91.0)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.
```

Figure B.5: Verificación en la terminal de flutter doctor

B.5 Instalación de un Emulador de Android

Para desarrollar y probar aplicaciones en Android, sigue estos pasos:

1. Descarga e instala Android Studio desde <https://developer.android.com/studio>
2. Abre Android Studio y sigue las instrucciones del asistente de configuración.
3. Instala el SDK de Android y las herramientas necesarias.
4. Navega a **AVD Manager** y crea un nuevo emulador de Android.
5. Selecciona un dispositivo y una versión de Android, luego sigue las instrucciones para configurar el emulador.
6. Inicia el emulador desde el **AVD Manager**.

APPENDIX B. MANUAL DE INSTALACIÓN

```
C:\Users\jaime>Flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.22.0, on Microsoft Windows [Versión 10.0.22631.3737], locale es-ES)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.8.3)
[X] Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop
development with C++" workload, and include these components:
    MSVC v142 - VS 2019 C++ x64/x86 build tools
    - If there are multiple build tool versions available, install the latest
    C++ CMake tools for Windows
    Windows 10 SDK
[✓] Android Studio (version 2023.1)
[✓] IntelliJ IDEA Community Edition (version 2022.1)
[✓] VS Code (version 1.91.0)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 1 category.
```

Figure B.6: Verificación en la terminal de flutter doctor

B.6 Configuración del Desarrollo para Android

Configura tu entorno de desarrollo para Android con los siguientes pasos:

1. Abre Android Studio.
2. Navega a **Preferences > Appearance & Behavior > System Settings > Android SDK**.
3. Asegúrate de que las casillas de las siguientes herramientas estén seleccionadas:
 - Android SDK Platform-Tools
 - Android SDK Build-Tools
 - Android SDK Tools
4. Aplica los cambios y cierra las preferencias.
5. Abre una terminal y ejecuta `flutter config -android-sdk [ruta_del_sdk_de_android]` para configurar Flutter con el SDK de Android.

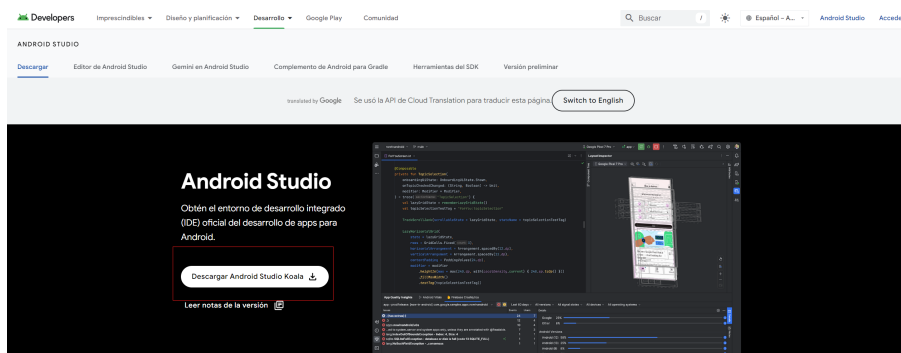


Figure B.7: Página de descarga de Android Studio

B.7 Prueba, Debug y Deploy de la Aplicación

B.7.1 Acceder a la Aplicación

Una vez instalada la aplicación en tu dispositivo, simplemente abre la aplicación desde el menú de aplicaciones para comenzar a utilizarla.

B.7.2 Debug de la Aplicación

Para realizar pruebas y depuración de la aplicación:

1. Abre VS Code y carga tu proyecto Flutter.
2. Conecta tu dispositivo o inicia un emulador.
3. Usa el comando **flutter run** en la terminal integrada de VS Code o utiliza el botón de **Run** y **Debug** en la barra de herramientas.
4. Coloca puntos de interrupción (breakpoints) en tu código para detener la ejecución y analizar el comportamiento de la aplicación.
5. Utiliza el panel de **Debug Console** para revisar los logs y realizar seguimiento de las variables durante la ejecución.

Appendix C

ANEXO C

	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Ingresos por descargas	60.000,00 €	81.000,00 €	109.350,00 €	147.622,50 €	199.290,38 €
Ingresos por suscripciones premium mensuales	66.029,90 €	72.632,89 €	106.528,25 €	117.181,07 €	189.557,61 €
Ingresos por suscripciones premium trimestrales	30.833,78 €	33.917,16 €	49.745,17 €	54.719,69 €	88.517,14 €
Ingresos por suscripciones premium anuales	18.358,16 €	20.193,98 €	29.617,84 €	32.579,62 €	52.702,33 €
Ingresos por publicidad In-App	25.920,00 €	28.512,00 €	31.363,20 €	34.499,52 €	37.949,47 €
Ingresos por ventas de cursos	5.389,20 €	23.748,12 €	26.122,93 €	28.735,23 €	31.608,75 €
Ingresos por colaboraciones	2.500,00 €	5.000,00 €	10.000,00 €	15.000,00 €	20.000,00 €
Ingresos por eventos	3.600,00 €	7.560,00 €	7.938,00 €	8.334,90 €	8.751,65 €
TOTAL INGRESOS	212.631,05 €	272.564,16 €	370.665,39 €	438.672,53 €	628.377,33 €

Table C.1: Ingresos proyectados de Codelingo

Se ha elaborado una hoja de ingresos que detalla los flujos de caja entrantes a lo largo de los 5 años analizados:

	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Coste de personal:					
Número de full-stack developers	1	1	1	1	1
Número de UX/UI Designers	1	1	1	1	1
Número de QA y Testers	0	0	0	0	0
Número de encargados Soporte Técnico	0	0	0	1	1
Número de Marketing Specialists	0	1	1	1	1
Número de Frontend developers	0	0	0	0	1
Número de Backend developers	0	0	1	1	1
Número de Product Managers	0	0	0	0	1
Total coste personal	105.000,00 €	140.000,00 €	200.000,00 €	230.000,00 €	335.000,00 €

Table C.2: Costes de personal de Codelingo

En contrapartida, se ha elaborado una hoja de costes que detalla los principales gastos de la empresa a lo largo de la vida del proyecto, cuyas principales secciones son:

	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Infraestructura tecnológica:					
Balanceador de carga (1-Sí / 0-No)	-	-	-	1	1
Servicios en la Nube	1.290,00 €	2.580,00 €	3.870,00 €	6.321,60 €	7.902,00 €
Hosting y dominios	375,00 €	375,00 €	375,00 €	375,00 €	375,00 €
Bases de Datos	2.760,00 €	3.360,00 €	3.960,00 €	4.560,00 €	5.160,00 €
Servicios de Asistente de Inteligencia Artificial	1.200,00 €	1.320,00 €	1.452,00 €	1.597,20 €	1.756,92 €
Servicios de Notificaciones Push	60,00 €	66,00 €	72,60 €	79,86 €	87,85 €
Almacenamiento de Contenidos Multimedia	99,60 €	199,20 €	298,80 €	398,40 €	498,00 €
Certificados SSL y Seguridad	300,00 €	300,00 €	300,00 €	300,00 €	300,00 €
Monitoreo y Logística	1.800,00 €	3.000,00 €	4.200,00 €	5.400,00 €	6.600,00 €
Total infraestructura tecnológica	7.884,60 €	11.200,20 €	14.528,40 €	19.032,06 €	22.679,77 €

Table C.3: Costes de infraestructura tecnológica de Codelingo

	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Marketing					
Costes de Marketing Digital/Campañas en redes sociales/Eventos	20.400,00 €	24.684,00 €	27.152,40 €	29.867,64 €	32.854,40 €
Operaciones y Administración					
Coste de gestión de proyectos y contabilidad	6.000,00 €	6.000,00 €	6.000,00 €	6.000,00 €	6.000,00 €
Costes legales y de Compliance					
Asesoría legal + Certificados (ISO27001, GDPR Compliance)	4.900,00 €	4.900,00 €	4.900,00 €	4.900,00 €	4.900,00 €
Formación y desarrollo del personal					
Coste de cursos y conferencias	6.000,00 €	6.300,00 €	6.615,00 €	6.945,75 €	7.293,04 €
Ciberseguridad					
Auditoría de seguridad + Servicios PenTesting	7.000,00 €	7.000,00 €	10.000,00 €	10.000,00 €	10.000,00 €
Total	157.184,60 €	200.084,20 €	269.195,80 €	306.745,45 €	418.727,21 €

Table C.4: Costes de marketing, administración, legales, formación y ciberseguridad de Codelingo

CUENTA DE RESULTADOS	1	2	3	4	5
	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Ingresos por descargas	60.000,00 €	81.000,00 €	109.350,00 €	147.622,50 €	199.290,38 €
Ingresos por suscripciones premium mensuales	66.029,90 €	72.632,89 €	106.528,25 €	117.181,07 €	189.557,61 €
Ingresos por suscripciones premium trimestrales	30.833,78 €	33.917,16 €	49.745,17 €	54.719,69 €	88.517,14 €
Ingresos por suscripciones premium anuales	18.358,16 €	20.193,98 €	29.617,84 €	32.579,62 €	52.702,33 €
Ingresos por publicidad In-App	25.920,00 €	28.512,00 €	31.363,20 €	34.499,52 €	37.949,47 €
Ingresos por ventas de cursos	5.389,20 €	23.748,12 €	26.122,93 €	28.735,23 €	31.608,75 €
Ingresos por colaboraciones	2.500,00 €	5.000,00 €	10.000,00 €	15.000,00 €	20.000,00 €
Ingresos por eventos	3.600,00 €	7.560,00 €	7.938,00 €	8.334,90 €	8.751,65 €
TOTAL INGRESOS	212.631,05 €	272.564,16 €	370.665,39 €	438.672,53 €	628.377,33 €
	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Coste personal	105.000,00 €	140.000,00 €	200.000,00 €	230.000,00 €	335.000,00 €
Coste infraestructura tecnologica	7.884,60 €	11.200,20 €	14.528,40 €	19.032,06 €	22.679,77 €
Costes de Marketing Digital/Campañas en redes sociales/ Eventos	20.400,00 €	24.684,00 €	27.152,40 €	29.867,64 €	32.854,40 €
Coste de gestión de proyectos y contabilidad	6.000,00 €	6.000,00 €	6.000,00 €	6.000,00 €	6.000,00 €
Asesoría legal + Certificados (ISO27001, GDPR Compliance)	4.900,00 €	4.900,00 €	4.900,00 €	4.900,00 €	4.900,00 €
Coste de cursos y conferencias	6.000,00 €	6.300,00 €	6.615,00 €	6.945,75 €	7.293,04 €
Auditoria de seguridad + Servicios PenTesting	7.000,00 €	7.000,00 €	10.000,00 €	10.000,00 €	10.000,00 €
TOTAL COSTES	157.184,60 €	200.084,20 €	269.195,80 €	306.745,45 €	418.727,21 €
	AÑO 1	AÑO 2	AÑO 3	AÑO 4	AÑO 5
Beneficio antes de impuestos (euros)	55.446	72.480	101.470	131.927	209.650
Impuestos	13.862	18.120	25.367	32.982	52.413
Beneficio despues de impuestos (euros)	41.585	54.360	76.102	98.945	157.238

Table C.5: Cuenta de resultados de Codelingo

Bibliography

- [1] Scaling customer-centric product development at bmw group with flutter. *flutter.dev*, 2020.
- [2] Santander Open Academy. ¿qué es un lenguaje de programación y cuáles son los más demandados en 2023? *Santander Open Academy Blog*, 2023.
- [3] Boletín Oficial del Estado. Programa código escuela 4.0, 2023.
- [4] Brendan Meeder Burr Settles. A trainable spaced repetition model for language learning. *Duolingo Blog*.
- [5] Code.org. La programación empieza a abrirse paso en los colegios. *El Mundo*, 2023.
- [6] Dilan Eren. The 2021 new coder survey – more than 18,000 people share how and why they are learning to code. *freeCodeCamp*, 2021.
- [7] Github. The state of octoverse 2019. *Octoverse*, 2019.
- [8] Denis Grankin. Agile software development lifecycle (sdlc): Phases and methodologies explained. *DDI development*, 2021.
- [9] HolonIQ. Five scenarios for the future of learning and talent. *HolonIQ*.
- [10] *Api Reference*.
- [11] Stack Overflow. Stack overflow developer survey 2023, 2023.
- [12] Fazilat Siddiq Ronny Scherer and Bárbara Sánchez-Scherer. Some evidence on the cognitive benefits of learning to code. *Frontiers in Psychology*, 2019.
- [13] Alfredo Bautista Santos. Clean architecture en flutter. *Medium*, 2020.
- [14] Burr Settles. How we learn how you learn. *Duolingo blog*, 2016.
- [15] SpringerLink. A review of machine learning methods used for educational data. *SpringerLink*.

- [16] SpringerOpen. Artificial intelligence in higher education: the state of the field. *SpringerOpen*.
- [17] Cynthia Pollard Cynthia Pollard Sara White Susanna Loeb, Sarah Novicof. The effects of virtual tutoring on young readers: Results from a randomized controlled trial. *National Student Support Accelerator*, 2021.
- [18] Consulting I. T. La demanda de programadores se duplicará en los próximos años. *ITReseller Tech Consulting*, 2023.
- [19] UNCTAD. Informe sobre tecnología e información. *Naciones Unidas*, 2021.
- [20] UNESCO. La escuela en la era de la inteligencia artificial. *Correo de la UNESCO*, 2023.
- [21] Oriol Zertuche. Openai presenta gpt-3.5 turbo-instruct: Un potente salto adelante. *MeetCodyAI*.
- [22] Xiaoming Zhai and Min Lu. Machine learning applications in educational studies. *Frontiers in Education*.