

---

**AccIA: Accesibilidad Cognitiva en la Era de las  
IAs**

**AccIA: Cognitive Accessibility in the Age of AIs**

---



**Trabajo de Fin de Grado  
Curso 2024–2025**

**Autora  
Aya Yahya Amar**

**Directoras  
Virginia Francisco Gilmartín  
Raquel Hervás Ballesteros**

**Grado en Ingeniería Informática  
Facultad de Informática  
Universidad Complutense de Madrid**



AccIA: Accesibilidad Cognitiva en la Era  
de las IAs

AccIA: Cognitive Accessibility in the Age  
of AIs

Trabajo de Fin de Grado en Ingeniería Informática

**Autora**  
Aya Yahya Amar

**Directoras**  
Virginia Francisco Gilmartín  
Raquel Hervás Ballesteros

**Convocatoria:** *Mayo/Junio 2025*  
**Calificación:** *9,6*

**Grado en Ingeniería Informática**  
**Facultad de Informática**  
**Universidad Complutense de Madrid**

**14 de junio de 2025**



# Dedicatoria

*A mi familia y a mis amigos, por estar a mi lado en cada etapa de la carrera y hacerla más amena.*



# Agradecimientos

Quiero expresar mi más sincero agradecimiento a mis tutoras, Virginia y Raquel, por su guía experta, paciencia y compromiso durante cada etapa de este proyecto. Su apoyo ha sido fundamental para alcanzar este logro. También, dar las gracias a Eloísa, por su ayuda y contribución en partes de la investigación de este trabajo.



# Resumen

A medida que las aplicaciones de Inteligencia Artificial Generativa (IAG) avanzan rápidamente, nos encontramos con que muchas de estas herramientas no están diseñadas con un enfoque inclusivo para personas con discapacidad cognitiva. Las interfaces de estas herramientas suelen dar por hecho conocimientos previos de uso de chats, utilizan lenguaje formal y carecen de adaptaciones específicas. Aunque existen soluciones tecnológicas que intentan abordar este problema, como asistentes de texto a voz, estas herramientas a menudo no son lo suficientemente flexibles para ajustarse a las necesidades específicas de estos usuarios al agrupar todas las discapacidades bajo un mismo enfoque o priorizando otras necesidades. Esto crea una barrera significativa para la inclusión digital, impidiendo que un sector importante de la población aproveche plenamente las ventajas de las tecnologías de IA generativa.

Para intentar cerrar esta brecha, se ha desarrollado *OlivIA*, una aplicación web que proporciona una interfaz accesible y adaptada a las necesidades de personas con discapacidad cognitiva. El proyecto comenzó con una fase de análisis de requisitos, en la que se identificaron las necesidades específicas de este colectivo, sirviendo como base para crear el diseño de una interfaz accesible que simplifica la interacción y reduce la sobrecarga cognitiva. Para adaptar la experiencia a cada usuario, *OlivIA* incluye un cuestionario inicial que recopila información sobre preferencias y dificultades, almacenándola en un perfil centralizado que se utiliza para ajustar dinámicamente las respuestas generadas por los modelos de lenguaje. Además, una de las características más destacadas de la aplicación es su capacidad para facilitar tanto la creación de preguntas como la interacción con las respuestas generadas. Para ello, incorpora plantillas predefinidas que permiten al usuario formular preguntas de forma rápida y sencilla, rellenando solo las palabras clave necesarias. Asimismo, cuenta con botones de ayuda específicos que facilitan la interacción con las respuestas generadas, ofreciendo opciones como resúmenes, ejemplos y simplificaciones. Esto no solo reduce la carga cognitiva, sino que también promueve una comunicación más clara y efectiva, mejorando significativamente la experiencia del usuario.

A nivel tecnológico, *OlivIA* adopta una arquitectura *serverless*, eliminando la necesidad de servidores backend y mejorando la escalabilidad y velocidad de respuesta. Utiliza la API de *Groq* para acceder a modelos avanzados como *Meta Llama 4*, conocidos por su precisión y rapidez, y se complementa con *OpenRouter* para conectar con otros modelos como *Mistral* y *Deepseek*, garantizando flexibilidad y

redundancia.

En el futuro, se espera seguir optimizando *OlivIA*, integrando nuevos modelos de lenguaje y ampliando las opciones de personalización para ofrecer una experiencia más precisa e inclusiva.

## Palabras clave

Accesibilidad Cognitiva, Inteligencia Artificial Generativa, Interfaces Inclusivas, Personalización de Respuestas, Grandes Modelos de Lenguaje (LLMs), Aplicaciones Web, Arquitectura Serverless, Diseño Accesible.

# Abstract

As Generative Artificial Intelligence (GAI) applications rapidly evolve, many of these tools are not designed with an inclusive approach for individuals with cognitive disabilities. The interfaces of such tools often assume prior knowledge of chat usage, use formal language, and lack specific adaptations. While there are technological solutions that attempt to address this issue, such as text-to-speech assistants, these tools are often not flexible enough to meet the specific needs of these users, as they tend to group all disabilities under a single approach or prioritize other needs. This creates a significant barrier to digital inclusion, preventing a considerable portion of the population from fully benefiting from generative AI technologies.

To help close this gap, *OlivIA* has been developed: a web application that provides an accessible interface tailored to the needs of people with cognitive disabilities. The project began with a requirements analysis phase, identifying the specific needs of this group to serve as a foundation for designing an accessible interface that simplifies interaction and reduces cognitive load. To personalize the experience for each user, *OlivIA* includes an initial questionnaire that gathers information about user preferences and difficulties. This information is stored in a centralized profile and used to dynamically adjust the responses generated by language models. One of the most notable features of the application is its ability to support both question creation and interaction with generated answers. To do this, it incorporates predefined templates that allow users to formulate questions quickly and easily by filling in only the necessary keywords. It also includes specific help buttons to assist with understanding responses, offering options such as summaries, examples, and simplified versions. This not only reduces cognitive load but also promotes clearer and more effective communication, significantly improving the user experience.

From a technological standpoint, *OlivIA* adopts a serverless architecture, removing the need for backend servers and improving scalability and response speed. It uses the *Groq* API to access advanced models like *Meta Llama 4*, known for their accuracy and speed, and is complemented by *OpenRouter* to connect with other models such as *Mistral* and *Deepseek*, ensuring flexibility and redundancy.

In the future, *OlivIA* is expected to continue evolving by integrating new language models and expanding personalization options to offer an even more accurate and inclusive experience.

## Keywords

Cognitive Accessibility, Generative Artificial Intelligence, Inclusive Interfaces, Response Personalization, Large Language Models (LLMs), Web Applications, Serverless Architecture, Accessible Design.

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Plan de Trabajo . . . . .	3
1.4. Estructura de la memoria . . . . .	4
<b>Introduction</b>	<b>7</b>
1.5. Motivation . . . . .	7
1.6. Objectives . . . . .	8
1.7. Work Plan . . . . .	9
1.8. Document Structure . . . . .	10
<b>2. Estado del Arte</b>	<b>13</b>
2.1. Discapacidad cognitiva . . . . .	13
2.2. Accesibilidad en interfaces web . . . . .	15
2.2.1. Estándares y Requisitos de Accesibilidad Web . . . . .	16
2.2.2. Herramientas para medir la accesibilidad en aplicaciones web . . . . .	23
2.3. Inteligencia artificial generativa . . . . .	39
2.3.1. <i>ChatGPT</i> . . . . .	41
2.3.2. <i>ChatSonic</i> . . . . .	43
2.3.3. <i>LuzIA</i> . . . . .	49
2.3.4. <i>Perplexity AI</i> . . . . .	52
2.3.5. <i>Claude</i> . . . . .	55
2.3.6. <i>YouChat</i> . . . . .	57
2.3.7. <i>Gemini (Google)</i> . . . . .	62
2.3.8. Conclusiones . . . . .	63
<b>3. Captura de requisitos</b>	<b>67</b>
3.1. Análisis de la accesibilidad en <i>ChatGPT</i> . . . . .	67
3.1.1. Análisis automático . . . . .	68
3.1.2. Análisis manual . . . . .	71
3.2. Probando ChatGPT con usuarios con discapacidad cognitiva . . . . .	72

3.2.1.	Diseño . . . . .	73
3.2.2.	Resultados . . . . .	76
3.2.3.	Cuestiones importantes a destacar de la actividad . . . . .	77
3.3.	Requisitos . . . . .	79
<b>4.</b>	<b>Diseño</b>	<b>85</b>
4.1.	Bocetos . . . . .	85
4.1.1.	Cuestionario . . . . .	86
4.1.2.	Pantalla principal / Generador de preguntas . . . . .	91
4.1.3.	Generador de respuestas . . . . .	93
4.2.	Diseño final . . . . .	94
4.2.1.	Cuestionario . . . . .	95
4.2.2.	Pantalla principal / Generador de preguntas . . . . .	96
4.2.3.	Historial . . . . .	100
4.2.4.	Configuración . . . . .	100
4.2.5.	Generador de respuesta . . . . .	102
4.3.	Conclusión . . . . .	104
<b>5.</b>	<b>Implementación</b>	<b>109</b>
5.1.	Tecnologías y arquitectura de la aplicación . . . . .	109
5.1.1.	Cliente . . . . .	110
5.1.2.	API . . . . .	111
5.2.	Gestión del estado del usuario durante la navegación . . . . .	111
5.3.	Componentes de la aplicación . . . . .	117
5.3.1.	Personalización . . . . .	117
5.3.2.	Sistema . . . . .	121
5.3.3.	Prompts . . . . .	127
<b>6.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>133</b>
6.1.	Conclusiones . . . . .	133
6.2.	Trabajo Futuro . . . . .	135
	<b>Conclusions and Future Work</b>	<b>139</b>
6.3.	Conclusions . . . . .	139
6.4.	Future Work . . . . .	141
	<b>Bibliografía</b>	<b>143</b>
	<b>A. Cuestionario pre-test para estudiantes</b>	<b>145</b>
	<b>B. Cuestionario post-test para estudiantes</b>	<b>147</b>
	<b>C. Cuestionario pre-test para cada estudiante a rellenar por la tutora ACCEDE</b>	<b>149</b>

D. Cuestionario post-test para cada estudiante a rellenar por las tutoras que se le supervisaron durante la actividad 151



# Índice de figuras

2.1. Análisis realizado con WAVE de la web de ChatGPT. . . . .	25
2.2. Interfaz principal del Pa11y Dashboard con múltiples análisis. . . . .	27
2.3. Informe detallado de accesibilidad para ChatGPT en Pa11y Dashboard. 28	
2.4. Interfaz de Axe DevTools mostrando una vista general de los problemas encontrados en ChatGPT . . . . .	29
2.5. Axe DevTools mostrando uno de los problemas en el análisis de ChatGPT . . . . .	29
2.6. La Interfaz de SortSite localizando fallos de accesibilidad en ChatGPT 31	
2.7. SortSite analizando diferentes parámetros y proponiendo posibles mejoras . . . . .	31
2.8. DYNO MAPPER analizando Gemini . . . . .	32
2.9. Interfaz de DYNO Accessibility Testing . . . . .	33
2.10. DYNO MAPPER en la opción de inventario . . . . .	34
2.11. Interfaz de DYNO Accessibility Audit . . . . .	34
2.12. Interfaz de DYNO Accessibility . . . . .	36
2.13. Interfaz de AChecker junto con las pautas de accesibilidad que evalúa. 38	
2.14. Análisis de accesibilidad de ChatGPT con AChecker. . . . .	39
2.15. Interfaz web de ChatGPT al abrir la aplicación . . . . .	44
2.16. Búsqueda y Autocompletado de texto en ChatGPT en su versión web 44	
2.17. Historial de búsqueda en ChatGPT en su versión web . . . . .	45
2.18. Chat personalizado de ChatGPT en su versión web . . . . .	45
2.19. Chat en el que se muestran los botones para evaluar la respuesta de ChatGPT en su versión web . . . . .	46
2.20. Chat en el que se muestran como los usuarios tienen la opción de reenviar o volver a ejecutar una consulta en ChatGPT en su versión web . . . . .	46
2.21. Chat en el que se muestran dos opciones de respuesta en ChatGPT en su versión web . . . . .	47
2.22. Interfaz móvil de ChatGPT al abrir la aplicación. . . . .	47
2.23. Chat en el que se le pide que de una respuesta de otra forma . . . . .	48
2.24. Búsqueda en ChatSonic . . . . .	49
2.25. Interfaz web de ChatSonic . . . . .	50

2.26. Búsqueda en ChatSonic . . . . .	50
2.27. Historial de chats de ChatSonic . . . . .	51
2.28. Interfaz y búsqueda de LuzIA . . . . .	53
2.29. Búsqueda en Perplexity AI web . . . . .	54
2.30. Interfaz de Perplexity AI web . . . . .	55
2.31. Interfaz de Perplexity AI móvil . . . . .	56
2.32. Interfaz de <i>Claude</i> con la opción de elegir cómo te responde . . . . .	58
2.33. Opción de crear estilos de respuesta en <i>Claude</i> . . . . .	58
2.34. Diferentes tipos de fuentes en <i>Claude</i> . . . . .	59
2.35. Comparación de Claude con diferentes estilos de fuente . . . . .	60
2.36. Interfaz móvil de <i>Claude</i> con funcionalidad de consultas por voz . . . . .	61
2.37. Interfaz principal junto con los botones predeterminados de YouChat . . . . .	61
2.38. Ejemplo de uso de YouChat . . . . .	62
2.39. Interfaz de Gemini . . . . .	63
2.40. Ejemplo de Gemini . . . . .	64
2.41. Ejemplo de Gemini analizando una gráfica . . . . .	64
4.1. Página de inicio del cuestionario . . . . .	86
4.2. Pantalla de elección del Camino . . . . .	87
4.3. Pantalla de retos en el Viaje . . . . .	88
4.4. Pantalla de elección de Herramientas . . . . .	89
4.5. Pantalla de ayuda del Compañero Virtual . . . . .	89
4.6. Pantalla de Confirmación de Ayuda del Compañero Virtual . . . . .	90
4.7. Pantalla de Ritmo de la Aventura . . . . .	90
4.8. Pantalla de Confirmación Final . . . . .	91
4.9. Pantalla de Generación de Preguntas principal . . . . .	92
4.10. Pantalla de Generación de Preguntas tras seleccionar una opción . . . . .	93
4.11. Pantalla de Generación de Respuestas . . . . .	94
4.12. Pantalla de Confirmación Final . . . . .	95
4.13. Página de inicio de cuestionario final . . . . .	96
4.14. Elección de discapacidad en cuestionario final . . . . .	97
4.15. Elección de dificultades en cuestionario final . . . . .	98
4.16. Elección de herramientas en cuestionario final . . . . .	99
4.17. Pantalla final del cuestionario . . . . .	99
4.18. Pantalla principal de versión final . . . . .	100
4.19. Historial abierto sin registros . . . . .	101
4.20. Historial abierto con registros . . . . .	101
4.21. Configuración del cuestionario inicial . . . . .	103
4.22. Generando respuesta de usuario . . . . .	104
4.23. Respuesta generada para el usuario . . . . .	105
4.24. Botón de “Responder en lenguaje más sencillo” . . . . .	106
4.25. Guardado en el historial . . . . .	107
5.1. Diagrama de cliente y API . . . . .	110

5.2.	useState de summary . . . . .	112
5.3.	Página 1 del Cuestionario - Nombre . . . . .	112
5.4.	Cuestionario: (a) Página 2 - Discapacidad, (b) Página 3 - Retos. . . . .	114
5.5.	Página 4 del Cuestionario - Herramientas . . . . .	115
5.6.	Página 5 del Cuestionario - Resumen . . . . .	115
5.7.	Diagrama de flujo de los componentes de <i>OlivIA</i> . . . . .	116
5.8.	Interfaz Principal . . . . .	119
5.9.	Interfaz de panel de configuración . . . . .	120
5.10.	Efecto de guardado de configuración . . . . .	121
5.11.	Texto de respuesta de IA generativa . . . . .	123
5.12.	Ejemplo de Conversación generada por plantilla “Formular desde cero”	124
5.13.	Botones Interactivos con la respuesta generada por la IA . . . . .	125
5.14.	Botón de reformular pregunta . . . . .	125
5.15.	Historial de <i>OlivIA</i> con chats generados . . . . .	126



# Índice de tablas

2.1. Comparativa de herramientas de accesibilidad . . . . .	37
2.2. Comparativa de aplicaciones IA generativa . . . . .	66



# Capítulo 1

## Introducción

*“El poder de la web está en su universalidad. El acceso para todos, independientemente de la discapacidad, es un aspecto esencial.”*

— Tim Berners-Lee

El primer capítulo de este documento presentará una introducción al Trabajo de Fin de Grado dividido en diferentes secciones. En primer lugar, la Sección 1.1, explicará la motivación que ha llevado a la realización y selección del tema del trabajo. A continuación, en la Sección 1.2 se presentarán los objetivos que se pretenden alcanzar. Posteriormente, en la Sección 1.3, se describe el plan de trabajo seguido para la consecución de los objetivos planteados y su cronograma. Por último, la estructura de la memoria se detallará en la Sección 1.4.

### 1.1. Motivación

La evolución de los grandes modelos de lenguaje (LLMs, por sus siglas en inglés), impulsados por redes neuronales profundas, ha generado asistentes de conversación que imitan el comportamiento humano, como *chatbots*. Empresas importantes, como *OpenAI*, *Microsoft* y *Google* entre a otras, están desarrollando *chatbots* basados en estas nuevas tecnologías de IA que son capaces de generar texto e imágenes a partir de un texto que define el objetivo del usuario. Dicho resultado parece relativamente humano y el uso de *chatbots* ha ido aumentando en popularidad y tanto el público general como profesional utilizan esta tecnología para realizar una gran gama de tareas.

No obstante, este progreso no ha beneficiado igualmente a las personas con discapacidad cognitiva. Este grupo enfrenta desafíos significativos al intentar acceder a estas nuevas tecnologías. Las interfaces de IA generativa actuales suelen ser poco intuitivas y carecen de las adaptaciones necesarias para garantizar la accesibilidad cognitiva. Menús complejos, iconos confusos y opciones de personalización limitadas son algunos de los obstáculos que encuentran estos usuarios. Además, el lenguaje complejo y abstracto utilizado por estas herramientas dificulta la comprensión e interacción efectiva para estas personas.

Las personas con discapacidades cognitivas se enfrentan a varios desafíos a la hora de usar este tipo de tecnologías:

- Dificultad para interactuar con las interfaces de usuario. Las interfaces de IA generativa pueden ser poco intuitivas o carecer de adaptaciones que faciliten su uso para personas con discapacidad cognitiva. Esto incluye menús complejos, símbolos poco claros o una gama limitada de opciones de personalización.
- Dificultad para generar el mensaje más adecuado para lograr el resultado que buscan. Las personas con discapacidades cognitivas pueden tener dificultades para generar mensajes claros y concisos. La capacidad para articular preguntas o instrucciones de manera eficaz puede verse afectada por limitaciones en el procesamiento cognitivo, la expresión verbal o la organización del pensamiento.
- Dificultad para entender el lenguaje utilizado. Las IAs generativas tienden a usar un lenguaje demasiado complejo y abstracto en las respuestas generadas que, sin adaptaciones para las personas con discapacidad cognitiva, puede resultar confuso y difícil de manejar.
- Dificultad para personalizar los resultados. Las personas con discapacidad cognitiva suelen necesitar ajustes específicos en la presentación de la información o las interacciones cuando utilizan aplicaciones digitales. La falta de opciones de personalización puede obstaculizar el uso eficaz de las herramientas de IA generativa.

Por todas estas razones, surge la necesidad de adaptar estas tecnologías para que sean accesibles para los usuarios con algún tipo de discapacidad cognitiva.

## 1.2. Objetivos

La finalidad de este Trabajo de Fin de Grado es desarrollar una aplicación con una interfaz accesible y personalizable diseñada con la intención de servir como un puente entre las personas con discapacidad cognitiva y las IAs generativas. Este proyecto se centrará en abordar las diversas necesidades de las personas con discapacidad cognitiva asegurando que el avance de las tecnologías como las IAs generativas se desarrolle en paralelo con la inclusión de estos colectivos.

Para ello, este trabajo se propone alcanzar varios objetivos específicos, entre los que destacan:

- Revisar la literatura actual sobre interfaces de IA generativa para identificar las tendencias y mejores prácticas en accesibilidad cognitiva.
- Estudiar las características necesarias para que las interfaces sean accesibles para personas con discapacidad cognitiva, considerando aspectos como la simplicidad, claridad y personalización.

- Realizar estudios de campo que permitan observar directamente los desafíos que enfrentan personas con discapacidad cognitiva al utilizar las aplicaciones de IA generativa actuales.
- Extraer los requisitos funcionales y no funcionales de la aplicación a desarrollar, basados en la investigación previa y los estudios de campo realizados.
- Diseñar e implementar una interfaz de usuario que permita a personas con discapacidad cognitiva interactuar con la aplicación de manera efectiva, utilizando principios de diseño centrado en el usuario.

Además, como parte del TFG, también se plantean los siguientes objetivos académicos:

- Aplicar los conocimientos adquiridos en el grado a un proyecto real, integrando teoría y práctica en el diseño de tecnología accesible.
- Aprender y experimentar con herramientas de IA generativa, adquiriendo un conocimiento profundo de su funcionamiento, limitaciones y potencial para adaptar su uso en contextos de accesibilidad cognitiva.
- Adquirir nuevos conocimientos y habilidades en el área de diseño de interfaces, accesibilidad cognitiva y desarrollo de aplicaciones, que contribuyan al crecimiento personal y profesional del equipo.
- Aprender a adaptar soluciones tecnológicas para personas con discapacidades cognitivas comprendiendo los principios fundamentales que hacen que una interfaz sea intuitiva y fácil de usar.
- Reflexionar sobre la responsabilidad del ingeniero en la creación de soluciones accesibles, cultivando una perspectiva profesional orientada hacia la equidad y el acceso universal a la tecnología.

## 1.3. Plan de Trabajo

A continuación se describe el plan de trabajo seguido para la consecución de los objetivos propuestos en este proyecto:

1. Investigación y Análisis del Estado del Arte (Septiembre - Noviembre 2024)
  - Revisión de literatura sobre accesibilidad cognitiva, interfaces inclusivas e inteligencia artificial generativa.
  - Identificación y análisis de herramientas de evaluación de accesibilidad y de diferentes aplicaciones de inteligencia generativa.
  - Evaluación de la accesibilidad de *ChatGPT* mediante un estudio con usuarios reales.
2. Captura de Requisitos (Diciembre 2024 - Febrero 2025)

- Análisis de los datos obtenidos en el estudio de campo con usuarios reales, así como de los resultados de la evaluación de accesibilidad de la interfaz de *ChatGPT*, con el objetivo de identificar las principales barreras de accesibilidad cognitiva y extraer los requisitos funcionales y no funcionales necesarios para el desarrollo de la aplicación.
- Investigación, selección y experimentación con diferentes APIs de modelos de inteligencia artificial generativa, evaluando su integración y rendimiento para su posterior implementación en la aplicación web.

### 3. Diseño de la Aplicación (Febrero 2025)

- Elaboración de bocetos y prototipos iniciales de la interfaz accesible.
- Priorización de requisitos funcionales.
- Diseño de funcionalidades de personalización, simplificación del lenguaje y adaptación de las respuestas.
- Validación de los bocetos iniciales y desarrollo del diseño final.

### 4. Implementación (Marzo - Mayo 2025)

- Desarrollo completo de la aplicación web en su versión funcional.
- Implementación del cuestionario inicial, sistema de personalización de respuestas, botones de ayuda y sistema de prompts personalizados.
- Implementación de la arquitectura modular de componentes, sistema de historial de conversación, panel de configuración e integración con los modelos de lenguaje mediante API.
- Publicación de la página web.
- Documentación técnica del desarrollo.

### 5. Conclusiones y Trabajo Futuro (Abril - Mayo 2025)

- Revisión global del trabajo desarrollado, identificando aspectos pendientes, reflexionando sobre los aprendizajes obtenidos durante el proyecto y proponiendo líneas de mejora y ampliación de cara a futuras implementaciones.
- Traducción al inglés de los capítulos 1, 4 y del resumen.
- Revisión completa de la memoria y preparación de la defensa del TFG.

## 1.4. Estructura de la memoria

La memoria se divide en los siguientes capítulos:

- Capítulo 1: Se presenta la introducción, donde se trata la motivación del tema del TFG, los objetivos a cumplir y la estructura tanto en español como en inglés.

- 
- Capítulo 2: Se presenta el estado del arte, donde trata la definición de las interfaces con accesibilidad cognitiva en el contexto de las aplicaciones de IA generativa, junto con las herramientas que evalúan la accesibilidad de las soluciones ya desarrolladas.
  - Capítulo 3: Se detalla la captura de requisitos, el análisis inicial de *ChatGPT* y la prueba con usuarios con discapacidad cognitiva.
  - Capítulo 4: Se presenta el diseño de la aplicación web, comenzando por los bocetos iniciales utilizados para explorar distintas alternativas visuales y funcionales, y continuando con el diseño final implementado, que incorpora mejoras centradas en la usabilidad, accesibilidad y experiencia de usuario.
  - Capítulo 5: Se presenta la implementación de la aplicación web a partir del diseño final, detallando cómo se han llevado a la práctica la lista de requisitos funcionales y no funcionales y cómo se ha integrado cada componente de la interfaz.
  - Capítulo 6: Se presentan las conclusiones y el trabajo futuro a implementar tanto en español como en inglés.



# Introduction

*“The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.”*

— Tim Berners-Lee

The first chapter of this document presents an introduction to the Final Degree Project, divided into different sections. First, Section 1.5 will explain the motivation behind the selection and execution of the project. Next, Section 1.6 will present the objectives to be achieved. Subsequently, Section 1.7 describes the work plan followed to achieve the proposed objectives, along with its timeline. Finally, the structure of the document will be detailed in Section 1.8.

## 1.5. Motivation

The evolution of large language models (LLMs) driven by deep neural networks has led to the development of conversational assistants, such as chatbots, that mimic human behavior. Major companies like *OpenAI*, *Microsoft*, and *Google*, among others, are creating chatbots based on these new AI technologies, capable of generating text and images based on a user-defined goal. The results generated often appear relatively human, and the use of chatbots has been increasing in popularity, with both the general public and professionals using this technology for a wide range of tasks.

However, this progress has not equally benefited people with cognitive disabilities. This group faces significant challenges when trying to access these new technologies. Current generative AI interfaces tend to be less intuitive and lack the necessary adaptations to ensure cognitive accessibility. Complex menus, unclear icons, and limited customization options are some of the obstacles faced by these users. Additionally, the complex and abstract language used by these tools can make understanding and effective interaction difficult for people with cognitive disabilities.

People with cognitive disabilities face several challenges when using this type of technology:

- Difficulty interacting with user interfaces: Generative AI interfaces can be unintuitive or lack the necessary adaptations to facilitate use for people with cognitive disabilities. This includes complex menus, unclear symbols, or a limited range of customization options.

- Difficulty generating the most appropriate message to achieve the desired outcome: People with cognitive disabilities may struggle to create clear and concise messages. Their ability to articulate questions or instructions effectively can be affected by limitations in cognitive processing, verbal expression, or thought organization.
- Difficulty understanding the language used: Generative AIs tend to use language that is too complex and abstract in their responses, which, without proper adaptations, can be confusing and challenging for people with cognitive disabilities.
- Difficulty personalizing the outputs: People with cognitive disabilities often require specific adjustments in the presentation of information or interactions when using digital applications. The lack of personalization options can hinder the effective use of generative AI tools.

For all these reasons, there is a need to adapt these technologies to be accessible to users with some form of cognitive disability.

## 1.6. Objectives

The purpose of this Final Degree Project is to develop an application with an accessible and customizable interface designed to serve as a bridge between people with cognitive disabilities and generative AIs. This project aims to address the diverse needs of people with cognitive disabilities, ensuring that the advancement of technologies like generative AIs is developed in parallel with the inclusion of these groups.

To achieve this, the project sets out several specific objectives, including:

- Reviewing the current literature on generative AI interfaces to identify trends and best practices in cognitive accessibility.
- Studying the necessary features for interfaces to be accessible to people with cognitive disabilities, considering aspects such as simplicity, clarity, and personalization.
- Conducting field studies to directly observe the challenges faced by people with cognitive disabilities when using current generative AI applications.
- Extracting the functional and non-functional requirements for the application to be developed, based on prior research and field studies.
- Designing and implementing a user interface that allows people with cognitive disabilities to interact effectively with the application, using principles of user-centered design.

Additionally, as part of the Final Degree Project, the following academic objectives are also proposed:

- Applying the knowledge acquired during the degree to a real project, integrating theory and practice in the design of accessible technology.
- Learning and experimenting with generative AI tools, gaining a deep understanding of their functioning, limitations, and potential for adaptation in cognitive accessibility contexts.
- Acquiring new skills in interface design, cognitive accessibility, and application development that contribute to the personal and professional growth of the team.
- Learning to adapt technological solutions for people with cognitive disabilities by understanding the fundamental principles that make an interface intuitive and easy to use.
- Reflecting on the engineer's responsibility in creating accessible solutions, cultivating a professional perspective oriented toward equity and universal access to technology.

## 1.7. Work Plan

The following describes the work plan followed to achieve the objectives proposed in this project:

1. Research and Analysis of the State of the Art (September - November 2024)
  - Literature review on cognitive accessibility, inclusive interfaces, and generative artificial intelligence.
  - Identification and analysis of accessibility evaluation tools and various generative AI applications.
  - Accessibility evaluation of *ChatGPT* through a study with real users.
2. Requirements Gathering (December 2024 - February 2025)
  - Analysis of the data obtained from the field study with real users, as well as from the accessibility evaluation of the *ChatGPT* interface, in order to identify the main cognitive accessibility barriers and extract the functional and non-functional requirements necessary for the development of the application.
  - Research, selection, and experimentation with different generative AI model APIs, evaluating their integration and performance for subsequent implementation in the web application.
3. Application Design (February 2025)
  - Development of initial mockups and prototypes for the accessible interface.

- Prioritization of functional requirements.
- Design of personalization features, language simplification, and response adaptation.
- Validation of initial mockups and development of the final design.

#### 4. Implementation (March - May 2025)

- Full development of the functional version of the web application.
- Implementation of the initial questionnaire, response personalization system, help buttons, and prompt customization system.
- Implementation of the modular architecture of components, conversation history system, configuration panel, and integration with language models via API.
- Deployment of the web application.
- Technical documentation of the development process.

#### 5. Conclusions and Future Work (April - May 2025)

- Global review of the work carried out, identifying pending aspects, reflecting on the knowledge acquired during the project, and proposing improvement lines and extensions for future implementations.
- Translation into English of Chapters 1 and 4, as well as the summary.
- Full review of the report and preparation for the final TFG defense.

## 1.8. Document Structure

The document is divided into the following chapters:

- Chapter 1: Presents the introduction, covering the motivation for the project, the objectives to be achieved, and the structure of the document in both Spanish and English.
- Chapter 2: Presents the state of the art, defining cognitively accessible interfaces in the context of generative AI applications, along with the tools used to assess the accessibility of already developed solutions.
- Chapter 3: Details the requirements gathering, initial analysis of *ChatGPT*, and user testing with people with cognitive disabilities.
- Chapter 4: Presents the design of the web application, starting with initial sketches used to explore different visual and functional alternatives, followed by the final design, which incorporates improvements focused on usability, accessibility, and user experience.

- Chapter 5: Presents the implementation of the web application based on the final design, detailing how the list of functional and non-functional requirements has been implemented and how each interface component has been integrated.
- Chapter 6: Presents the conclusions and future work to be implemented in both Spanish and English.



# Capítulo 2

## Estado del Arte

Este capítulo presentará conceptos clave relacionadas con la accesibilidad cognitiva y la inteligencia artificial generativa, proporcionando un contexto esencial para este Trabajo de Fin de Grado. Además, se explorarán herramientas de evaluación de accesibilidad y aplicaciones similares a este proyecto, identificando tendencias y buenas prácticas que guiarán el desarrollo de una solución inclusiva.

### 2.1. Discapacidad cognitiva

Aproximadamente una de cada siete personas sufre algún tipo de discapacidad ya sea permanente, temporal u ocasional que afecta con su manera de interactuar con nuestro entorno a lo largo de su vida e incluso en su día a día<sup>1</sup>.

La discapacidad cognitiva se conoce como la condición en la cual una persona tiene dificultades significativas en el funcionamiento intelectual general. Estas dificultades se manifiestan durante el período de desarrollo y pueden afectar la capacidad de una persona para aprender y comprender nueva información, así como para utilizar estrategias para resolver problemas en la vida diaria (American Psychiatric Association, 2013).

Además de las discapacidades cognitivas generales que afectan el funcionamiento intelectual, existen otras condiciones neurodesarrolladas que también influyen en la manera en que una persona procesa la información, interactúa con el entorno y resuelve problemas. Entre estas condiciones se encuentran la hiperactividad (TDAH), la dislexia o el trastornos del espectro autista (TEA). Cada una de estas condiciones tiene un impacto único en el aprendizaje y el procesamiento de la información:

- El TDAH<sup>2</sup> es una condición común que afecta la capacidad de centrarse, difícil poder quedarse quieto y pensar antes de actuar (impulsividad). Es uno de los trastornos mentales más comunes entre niños y adolescentes. Además, el TDAH a menudo coexiste con dificultades de aprendizaje.

---

<sup>1</sup><https://cuatroochenta.com/el-superpoder-invisible-del-diseno-accesible/>

<sup>2</sup><https://www.understood.org/es-mx/articles/what-is-adhd>

- La dislexia<sup>3</sup> es un trastorno específico del aprendizaje que consiste en la dificultad en la lectura debido a inconvenientes para identificar los sonidos del habla y aprender a relacionarlos con las letras y las palabras (decodificación). La mayoría de los niños con dislexia pueden tener un buen resultado en la escuela con enfoques y métodos de enseñanza adaptados para facilitar su aprendizaje. Las personas con dislexia, tanto niños como adultos, pueden tener dificultad para descifrar texto o seguir narrativas extensas, lo que puede afectar el uso de tecnologías que dependen del lenguaje escrito.
- El TEA<sup>4</sup> es un trastorno del desarrollo que afecta el desarrollo cerebral, afectando la interacción social y la comunicación, y presenta patrones de conducta repetitivos y restringidos. Esto puede dificultar la interacción con interfaces que carecen de claridad visual o que requieren un alto grado de interpretación.

Las personas con discapacidad cognitiva enfrentan una serie de desafíos específicos que dificultan su capacidad para aprender, comunicarse y desenvolverse en entornos digitales y físicos. Estos obstáculos varían según la naturaleza de la discapacidad y el contexto, pero todos ellos impactan en la calidad de vida y en la participación social y educativa. A continuación, se describen algunos de los principales desafíos a los que se enfrentan este grupo de personas:

- Dificultades de lectura y/o habla<sup>5</sup>: Estas dificultades pueden limitar su capacidad para comunicarse de manera efectiva o para interactuar con materiales escritos, especialmente si no están adaptados a sus necesidades.
- Dificultades de comprensión<sup>6</sup>: Las personas con discapacidades cognitivas pueden experimentar dificultades para procesar y entender información. Esto puede incluir problemas para interpretar conceptos abstractos, seguir instrucciones complejas, identificar relaciones entre ideas o comprender lenguaje figurado como metáforas y analogías. Estas dificultades afectan su capacidad para captar el significado de mensajes o tareas cuando no están expresados de forma clara, concreta y directa.
- Problemas de memoria y atención<sup>7</sup>: La falta de memoria a corto plazo y las dificultades para concentrarse dificultan la retención de información y la continuidad en tareas, especialmente en personas con trastorno por déficit de atención e hiperactividad (TDAH).
- Distracciones<sup>8</sup>: Los entornos ruidosos o caóticos pueden ser particularmente desafiantes para aquellos con TDAH o TEA, lo que resalta la necesidad de un entorno estructurado para maximizar el rendimiento.

---

<sup>3</sup><https://www.mayoclinic.org/es/diseases-conditions/dyslexia/symptoms-causes/syc-20353552>.

<sup>4</sup><https://www.mayoclinic.org/es/diseases-conditions/autism-spectrum-disorder/symptoms-causes/syc-20352928>

<sup>5</sup><https://www.mayoclinic.org/es/diseases-conditions/dyslexia/symptoms-causes/syc-20353552>

<sup>6</sup><https://cedec.intef.es/discapacidad-intelectual-y-accesibilidad-cognitiva>

<sup>7</sup><https://cdincbarcelona.com/>

<sup>8</sup><https://www.theminiadhcoach.es/post/el-tdah-y-las-distracciones>

- Dificultades en la toma de decisiones<sup>9</sup>: Algunas personas con discapacidad cognitiva pueden experimentar ansiedad o inseguridad al tomar decisiones, especialmente cuando se enfrentan a múltiples opciones o incertidumbres, lo cual es común en personas con trastornos del espectro autista (TEA) por la disfunción ejecutiva.
- Problemas con la planificación y organización: Esto incluye problemas para establecer prioridades, dividir tareas complejas en pasos más pequeños, gestionar el tiempo de manera efectiva o anticipar las consecuencias de sus acciones. Estas dificultades pueden afectar su capacidad para completar proyectos o realizar actividades que requieran una secuencia lógica de pasos.
- Sobrecarga sensorial<sup>10</sup>: Aquellos con TEA pueden sentirse abrumados en entornos con demasiados estímulos, lo que impacta su bienestar y concentración. La exposición a un entorno ruidoso o visualmente sobrecargado puede ser especialmente desafiante para estas personas.
- Situaciones de pánico<sup>11</sup>: Los individuos con TEA pueden experimentar ansiedad intensa en situaciones sociales o ante estímulos inesperados, afectando su bienestar emocional y su participación en actividades sociales.

## 2.2. Accesibilidad en interfaces web

La accesibilidad<sup>12</sup>, en el contexto de sistemas interactivos, es hacer que las interfaces sean perceptibles, operables y comprensibles para personas con distintos tipos de discapacidades.

Una interfaz accesible hace que las personas, independientemente de sus capacidades, puedan percibir, operar y comprender su funcionamiento de manera efectiva. Esto implica crear diseños que sean fáciles de percibir, utilizar y entender. La accesibilidad no solo beneficia a personas con discapacidades permanentes, sino también a quienes enfrentan limitaciones temporales o situaciones contextuales que dificultan el uso de un producto. Por ejemplo, utilizar una aplicación de mensajería en un entorno ruidoso, como un concierto, puede complicar la recepción de notificaciones auditivas, lo que subraya la importancia de opciones accesibles, como señales visuales o vibraciones.

Implementar interfaces con diseños accesibles ayuda a prevenir la discriminación hacia el 16 % de la población mundial<sup>13</sup> que vive con algún tipo de discapacidad. Dentro de este grupo, las discapacidades cognitivas afectan entre el 1 % y el 3 % de la población global<sup>14</sup>. Diseñar interfaces accesibles no solo promueve la inclusión, sino

<sup>9</sup><https://www.elsonidodelahierbaalcrecer.com/2023/07/autismo-y-toma-de-decisiones>

<sup>10</sup><https://tea.som360.org/es/blog/estimulos-sensoriales-personas-tea>

<sup>11</sup><https://www.autismoburgos.es/la-ansiedad-en-las-personas-con-tea/>

<sup>12</sup><http://uiaccess.com/justask/es/overview.html>

<sup>13</sup><https://www.who.int/es/news-room/fact-sheets/detail/disability-and-health>

<sup>14</sup><https://www.nataliamorales.com.co/trastornos-neurologicos-en-ninos/discapacidad-cognitiva-ninos>

que también aporta beneficios significativos en una amplia variedad de sectores<sup>15</sup>:

- **Beneficio social:** La accesibilidad permite que las personas con discapacidad sean más independientes, facilitando su participación en actividades diarias como leer noticias o realizar compras en línea.
- **Beneficio tecnológico:** las webs accesibles son priorizadas por motores de búsqueda, como *Google*, gracias a características como el texto alternativo, estructuras de encabezados adecuadas y enlaces descriptivos por lo que mejora la indexación y el posicionamiento en las páginas de resultados, aumentando así las visitas y las posibilidades de conversión.
- **Beneficio económico:** Incorporar accesibilidad ayuda a captar nuevos clientes y mercados, aumentando así el potencial de crecimiento y las oportunidades comerciales.

Para garantizar que las interfaces accesibles sean efectivas y cumplan con los estándares establecidos, es importante considerar dos aspectos clave, que se verán en las siguientes subsecciones: las pautas internacionales que definen los principios de accesibilidad y las herramientas que permiten medir y evaluar el cumplimiento de estos estándares.

### 2.2.1. Estándares y Requisitos de Accesibilidad Web

En el desarrollo de tecnologías digitales accesibles, existen diversos estándares y normativas internacionales que proporcionan guías fundamentales para garantizar la inclusión de todas las personas, independientemente de sus capacidades. Entre los principales marcos se encuentran las Pautas de Accesibilidad para el Contenido Web (WCAG), las pautas para accesibilidad cognitiva del W3C (COGA), el conjunto de especificaciones ARIA (Accessible Rich Internet Applications), la Sección 508 de la Ley de Rehabilitación de los Estados Unidos, la normativa BITV 1.0 alemana o la Stanca Act italiana. Estas herramientas no solo establecen criterios técnicos, sino que también promueven buenas prácticas para el diseño inclusivo. A continuación, se explorará cada uno de estos marcos en detalle.

#### 2.2.1.1. Pautas de Accesibilidad para el Contenido Web (WCAG)

Para crear interfaces accesibles y ofrecer una experiencia inclusiva a todos los usuarios, es esencial seguir las Pautas de Accesibilidad para el Contenido Web, o *WCAG* (por sus siglas en inglés). Estas pautas fueron desarrolladas por el World Wide Web Consortium (*W3C*), a través de su Iniciativa de Accesibilidad en la Web (*WAI*), como un estándar global para mejorar la accesibilidad del contenido en línea permitiendo que personas con discapacidades visuales, auditivas, motrices o cognitivas, puedan interactuar de manera autónoma con el contenido web<sup>16</sup>.

<sup>15</sup><https://accessibilias.es/2021/11/a-quien-beneficia-la-accesibilidad/>

<sup>16</sup><https://www.w3.org/WAI/standards-guidelines/wcag/>

Las WCAG están organizadas en cuatro principios clave: perceptible, operable, comprensible y robusto (o POUR, por sus siglas en inglés)<sup>17</sup>. Esto significa que los sitios web deben ser visibles y entendibles para todos los usuarios, operables mediante diversas tecnologías de asistencia, fáciles de comprender y compatibles con diferentes plataformas y dispositivos, incluidos los móviles.

Las WCAG incluyen tres niveles de conformidad<sup>18</sup> que miden el grado de accesibilidad de un sitio web: A (nivel básico), AA (nivel intermedio) y AAA (nivel avanzado). Cada nivel incluye los criterios del nivel anterior más criterios adicionales. Es decir, el nivel AA abarca todos los criterios del nivel A y agrega otros nuevos, mientras que el nivel AAA incluye los criterios de los niveles A y AA, además de otros más avanzados. El nivel A se centra en garantizar la accesibilidad básica, cubriendo aspectos esenciales que impiden que algunos usuarios accedan al contenido, como problemas relacionados con el uso del teclado o la percepción visual mínima. El nivel AA aborda barreras comunes para personas con discapacidades visuales, auditivas, físicas, del habla, cognitivas, del lenguaje, de aprendizaje y neurológicas, mejorando la usabilidad general. El nivel AAA, por su parte, busca alcanzar el máximo grado de accesibilidad, cubriendo requisitos avanzados para adaptarse a las necesidades específicas de diversas discapacidades.

Al aplicar las WCAG, los diseñadores y desarrolladores pueden asegurar que sus sitios web y aplicaciones sean accesibles para una audiencia más amplia. Esto no solo beneficia a los usuarios con discapacidades, sino que también mejora la experiencia de navegación para muchos otros usuarios.

A continuación se presentan los criterios contenidos en cada principio que están orientados a garantizar la accesibilidad para personas con discapacidad cognitiva<sup>19</sup>:

1. Perceptibilidad: La información de la web debe presentarse de manera que sea accesible tanto visual como auditivamente para todas las personas.
  - WCAG 1.1.1 (Nivel A) - Contenido no textual: Todo contenido no textual (imágenes, íconos, gráficos, tablas, mapas de imagen, etc.) debe tener una alternativa textual que describa su significado para facilitar la comprensión a usuarios con dificultades de memoria y procesamiento. Las imágenes informativas deben tener un texto alternativo claro, mientras que las decorativas deben llevar “alt=” para ser ignoradas por los lectores de pantalla. Además, gráficos y tablas requieren un resumen o descripción completa que complemente el texto alternativo breve. Para la seguridad, es preferible evitar CAPTCHAs tradicionales. Si se necesita un sistema de verificación, se pueden usar contraseñas de un solo uso (OTP). Si es indispensable un CAPTCHA, se recomienda uno invisible o, en su defecto, un CAPTCHA con alternativas visuales y auditivas.
  - WCAG 1.3.1 (Nivel A) - Información y relaciones: La información y la estructura del contenido deben ser presentadas de manera que puedan ser percibidas y comprendidas por todos los usuarios. Utilizar etiquetas

<sup>17</sup><https://accedeme.com/blog/los-principios-pour-de-las-wcag/>

<sup>18</sup><https://accesibilidadweb.dlsi.ua.es/?menu=que-significan-niveles-a-aa-aaa-accesibilidad-web>

<sup>19</sup><https://guia-wcag.com/es/>

de campo (*field labels*) descriptivas que permanezcan visibles y que expliquen la función de un campo de entrada es esencial, ya que permite que los usuarios comprendan el propósito de cada campo y no dependan únicamente de su memoria.

2. Operabilidad: La interfaz web debe ser manejable por cualquier usuario.

- WCAG 2.3.1 (Nivel A) - Tres parpadeos o menos: El contenido que parpadea o destella debe limitarse a un máximo de tres parpadeos por segundo. El contenido visual que parpadea o cambia rápidamente puede ser extremadamente distractor y puede dificultar que algunos usuarios se concentren en la información importante de la página.
- WCAG 2.2.2 (Nivel A) - Pausa, para, esconde: Se deben reducir las distracciones causadas por contenido que se mueve, parpadea o cambia automáticamente, y dar a los usuarios la posibilidad de tomar el control sobre este contenido.

3. Comprensibilidad: El contenido web y su uso deben ser fáciles de entender

- WCAG 2.4.3 (Nivel A) - Orden de enfoque: Cuando los usuarios navegan secuencialmente a través del contenido, deben poder encontrar la información en un orden que sea coherente con el significado y pueda ser operado con el teclado. Esto reduce la confusión al permitirle al usuario formarse un modelo mental consistente del contenido. Pueden existir diferentes órdenes que reflejen las relaciones lógicas en el contenido. Por ejemplo, al navegar por los componentes de una tabla, el orden de enfoque podría moverse fila por fila o columna por columna, según las relaciones lógicas del contenido. Esto permite que el usuario comprenda mejor la estructura de la tabla y su contenido, ya que el orden de navegación refleja cómo están organizados los datos. Cualquier orden puede satisfacer este criterio de conformidad.
- WCAG 3.3.3 (Nivel AA) - Error sugerido: Cuando un error en la entrada de datos ocurra, los usuarios deben recibir sugerencias claras para corregirlo, reduciendo la frustración y el abandono de formularios.
- WCAG 3.3.6 (Nivel AA) - Prevención de errores: Se deben ofrecer opciones que permitan a los usuarios revisar y corregir la información antes de enviarla, así como la posibilidad de deshacer acciones. De este modo, los usuarios tienen la oportunidad de identificar y corregir errores antes de que se realicen, lo que es fundamental para prevenir consecuencias no deseadas.
- WCAG 2.4.6 (Nivel AA) - Encabezados y etiquetas: Los encabezados y etiquetas en las páginas web deben ser claros y descriptivos para facilitar la comprensión y navegación. Encabezados claros permiten a los usuarios identificar rápidamente el contenido y su organización y reduce la necesidad de pulsaciones.

- WCAG 3.3.5 (Nivel AAA) - Ayuda: Se debe proporcionar elementos visuales que refuercen el contenido textual y brindar ayuda contextual cuando sea posible, como incluir iconos que representen acciones de manera clara y ofrecer mensajes de ayuda. La ayuda contextual solo necesita proporcionarse cuando la etiqueta no es suficiente para describir todas las funcionalidades, y su existencia debe ser evidente para el usuario, quien debe poder acceder a ella en cualquier momento. El texto de ayuda puede ser proporcionado por el autor del contenido o por el agente de usuario, basándose en información determinada y específica de la tecnología.
- WCAG 3.1.5 (Nivel AAA) - Lenguaje claro y simple: Se debe usar un lenguaje sencillo, evitar términos especializados, salvo que sean necesarios y aumentar el uso de definiciones o ejemplos. La claridad del contenido puede evaluarse mediante fórmulas de legibilidad, lo que permite adaptar los textos a una audiencia más amplia, independientemente del nivel educativo.

4. Robustez: El diseño debe funcionar con diversas tecnologías actuales y futuras.

- WCAG 4.1.2 (Nivel A) - Nombre, función, valor: Todos los componentes de la interfaz de usuario (como formularios, enlaces y elementos generados por *scripts*) deben tener un nombre y función claros que puedan ser identificados por el software. Además, debe ser posible especificar y consultar los valores y estados de estos elementos a través de ayudas técnicas. Además, es crucial que los controles sean estándar, y si no lo son, se deben proporcionar ayudas técnicas que expliquen cómo funcionan para facilitar la interacción.

### 2.2.1.2. Cognitive Accessibility (COGA)

Las pautas para accesibilidad cognitiva del W3C (COGA, por sus siglas en inglés) se centra en garantizar que las personas con discapacidades cognitivas, de aprendizaje o neurológicas puedan interactuar eficazmente con entornos digitales. Estas pautas, desarrolladas por el W3C como un complemento de las WCAG, abordan barreras que dificultan la comprensión, navegación e interacción en plataformas digitales<sup>20</sup>.

Según Plena Inclusión Madrid (2020), los principios COGA junto con las pautas específicas asociadas a cada uno son los siguientes:

1. Proveer contenido comprensible y claro: Este principio se enfoca en facilitar la comprensión del contenido para personas con discapacidades cognitivas.
  - Reducir la complejidad del contenido y utilizar lenguaje claro y directo: Eliminar tecnicismos innecesarios y usar lenguaje claro, directo y fácil de entender. Por ejemplo, en lugar de escribir “procesamiento de datos asíncrono”, utilizar “procesamiento de datos en segundo plano”

---

<sup>20</sup><https://www.w3.org/WAI/cognitive/>

- Simplificar el lenguaje eliminando tecnicismos innecesarios y proporcionando definiciones accesibles.
  - Usar diseño visual claro: Diseñar contenidos organizados con listas, encabezados claros y viñetas para dividir la información en fragmentos manejables.
  - Proveer retroalimentación inmediata y comprensible cuando ocurra un error: Cuando un usuario comete un error, como no llenar un campo obligatorio, proporcionar un mensaje comprensible y concreto, como “Por favor, completa este campo”.
2. Diseño predecible y consistente: Este principio se centra en garantizar que las interfaces sean coherentes y predecibles para evitar confusión.
- Facilitar la navegación y el uso de interfaces consistentes para que los usuarios puedan predecir interacciones futuras: Colocar botones, menús y enlaces en las mismas posiciones en cada página.
  - Usar patrones de diseño familiares que los usuarios puedan reconocer fácilmente: Usar iconos, colores y flujos que los usuarios ya reconozcan (por ejemplo, un icono de lupa para búsquedas).
  - Minimización de distracciones: Evitar animaciones o transiciones inesperadas que puedan distraer a los usuarios.
3. Reducir la carga cognitiva: Este principio busca simplificar tareas para que los usuarios no se sientan sobrecargados.
- Disminuir la cantidad de información o pasos necesarios para completar tareas: Para formularios largos, dividir las tareas en secciones manejables (por ejemplo, “Información personal”, “Dirección”, “Método de pago”).
  - Evitar sobrecargar al usuario con demasiadas opciones: Mostrar solo las opciones más importantes para evitar confusión o parálisis de decisión.
  - Ofrecer sugerencias automáticas y validaciones en formularios para minimizar errores: Proveer sugerencias o correcciones automáticas en tiempo real, como autocompletar un campo o resaltar errores.
4. Proveer ayuda y soporte contextual: Este principio se centra en diseñar sistemas que guíen a los usuarios durante su uso.
- Ofrecer asistencia integrada para guiar al usuario en el uso de la aplicación.
  - Tutoriales interactivos: Incluir explicaciones iniciales al usar la aplicación por primera vez.
  - Instrucciones paso a paso: En tareas complejas, mostrar instrucciones claras y breves para cada paso.
  - Incorporar iconos o gráficos intuitivos para mejorar la comprensión del contenido: Los iconos deben ser reconocibles y representar claramente su propósito (por ejemplo, un sobre para indicar “mensajes”).

5. Apoyar la adaptación y personalización: Este principio permite que los usuarios ajusten la interfaz según sus necesidades.
  - Permitir a los usuarios personalizar la interfaz según sus necesidades: Permitir ajustes como tamaño de fuente, colores, contraste, estilos o la presentación de elementos visuales.
  - Evitar cambios inesperados: Asegurar que la configuración o funcionalidad no cambien sin el consentimiento del usuario.
  - Permitir al usuario simplificación del contenido: Ofrecer la opción de ocultar funciones avanzadas y usar una versión más básica de la aplicación.
  - Facilidad para deshacer cambios: Proveer un botón para revertir configuraciones o volver al estado inicial fácilmente.
6. Mejorar la accesibilidad del tiempo y ritmo: Este principio se centra en facilitar el uso de la interfaz para quienes necesitan más tiempo para completar tareas.
  - Flexibilidad en los límites de tiempo: Evitar que el sistema expulse al usuario por inactividad o permitir que desactive límites de tiempo
  - Ofrecer control de multimedia: Ofrecer controles claros para pausar, detener o ajustar la velocidad de videos o animaciones.
  - Proporcionar modos alternativos de interacción para tareas que requieran rapidez o precisión: para tareas basadas en tiempo, como cuestionarios con tiempo extendido.
7. Proveer retroalimentación y validación accesibles: Este principio se enfoca en que la retroalimentación sea clara, comprensible y útil.
  - Usar mensajes de error amigables que expliquen claramente el problema y cómo resolverlo: Explicar qué error ocurrió y cómo solucionarlo (por ejemplo, “La contraseña debe tener al menos 8 caracteres”).
  - Proveer confirmaciones visuales y auditivas para las acciones completadas con éxito: Usar alertas visuales (como un *check* verde) y auditivas (un sonido suave) para confirmar acciones exitosas.
8. Diseñar para todos: Este principio busca crear productos inclusivos desde el inicio, garantizando que funcionen bien para diversas capacidades.
  - Compatibilidad universal: Asegurar que las aplicaciones sean compatibles con herramientas de asistencia, como lectores de pantalla o dispositivos de entrada alternativos.
  - Diseño inclusivo: Considerar las necesidades de usuarios con discapacidades cognitivas desde la fase de diseño.
  - Pruebas de usabilidad: Realizar evaluaciones directas con personas con discapacidades cognitivas para identificar barreras y oportunidades de mejora.

### 2.2.1.3. Accessible Rich Internet Applications (ARIA)

ARIA<sup>21</sup> es un conjunto de especificaciones desarrollado por el W3C que tiene como objetivo mejorar la accesibilidad de las aplicaciones web dinámicas. Proporciona atributos que pueden ser añadidos al código HTML para describir mejor el propósito, el estado o el comportamiento de los elementos de una página web, especialmente aquellos creados con JavaScript u otras tecnologías que no son nativamente accesibles. Entre sus características clave se encuentran:

- Roles: Definen la función de un elemento en la interfaz, como “botón”, “menú” o “diálogo”.
- Propiedades: Proporcionan información adicional sobre el estado de los elementos, como “expandido” o “deshabilitado”.
- Eventos accesibles: Permiten comunicar cambios dinámicos en el contenido, asegurando que las tecnologías de asistencia, como lectores de pantalla, puedan interpretarlos correctamente.

ARIA es fundamental para asegurar que los sitios web modernos, que a menudo dependen de interacciones dinámicas, sean accesibles para usuarios con discapacidades visuales o motoras.

### 2.2.1.4. Sección 508

La Sección 508<sup>22</sup> es una norma legal de accesibilidad digital derivada de la Ley de Rehabilitación de los Estados Unidos que obliga a las agencias federales a garantizar que su tecnología electrónica e informática sea accesible para personas con discapacidades. Desde 2017, los estándares de la Sección 508 están alineados con las WCAG 2.0, nivel AA, lo que garantiza consistencia en los criterios de accesibilidad. Aunque es una ley de los Estados Unidos, su adopción ha influido en otras legislaciones y estándares internacionales relacionados con la accesibilidad digital.

La Sección 508 es particularmente relevante para proyectos que buscan colaboración o cumplimiento con entidades gubernamentales en los Estados Unidos y sirve como un ejemplo de cómo la accesibilidad puede ser incorporada a nivel normativo.

### 2.2.1.5. BITV 1.0

La BITV 1.0 (Barrierefreie Informationstechnik-Verordnung)<sup>23</sup> es una normativa alemana que regula la accesibilidad de los sitios web y aplicaciones digitales, especialmente en el ámbito público. Fue adoptada en 2002 como parte de la Ley Federal de Igualdad de Derechos para Personas con Discapacidades (BGG, por sus siglas en alemán). Su objetivo principal es garantizar que las tecnologías de la información sean accesibles para todas las personas, incluidas aquellas con discapacidades.

<sup>21</sup><https://www.w3.org/WAI/ARIA/>

<sup>22</sup><https://www.section508.gov/manage/laws-and-policies/>

<sup>23</sup><https://www.accessi.org/bitv>

La BITV 1.0 está basada en los principios de accesibilidad definidos en las WCAG 1.0 y clasifica el cumplimiento en diferentes niveles. El Nivel 2 de BITV 1.0 equivale aproximadamente al nivel AA de las WCAG 1.0. Desde su implementación, se han introducido actualizaciones y modificaciones para alinearse con las WCAG 2.0 y WCAG 2.1, lo que refleja el compromiso de Alemania con la accesibilidad digital y la integración de personas con discapacidades en el entorno digital.

#### 2.2.1.6. Stanca Act

La Stanca Act (Ley 4/2004, conocida oficialmente como “Disposiciones para favorecer el acceso de los sujetos discapacitados a los instrumentos informáticos”)<sup>24</sup> es una legislación italiana que regula la accesibilidad de los sitios web y servicios digitales en el país. Fue promulgada en 2004 y establece obligaciones específicas para las entidades públicas y las empresas privadas que ofrecen servicios financiados por el gobierno. La ley se basa en los principios de las WCAG y busca garantizar la igualdad de acceso para las personas con discapacidades. Al igual que la Sección 508, la Stanca Act está diseñada para asegurar que la tecnología digital sea accesible para todos los usuarios. Incluye estándares relacionados con la navegación, la percepción visual y auditiva, la comprensibilidad del contenido y la facilidad de interacción.

En 2018, la Stanca Act fue actualizada para alinearse con las WCAG 2.0, reforzando su compatibilidad con los estándares internacionales. La normativa también fomenta el uso de buenas prácticas en el diseño accesible y establece procedimientos de evaluación para verificar el cumplimiento, lo que ayuda a las organizaciones a identificar y corregir deficiencias en la accesibilidad de sus sitios web y servicios.

### 2.2.2. Herramientas para medir la accesibilidad en aplicaciones web

Las herramientas para medir la accesibilidad de aplicaciones web son recursos esenciales para desarrolladores y diseñadores, ya que les permiten evaluar si sus aplicaciones cumplen con estándares de accesibilidad, como las Pautas de Accesibilidad al Contenido en la Web (WCAG). Estas herramientas son fundamentales para identificar, analizar y corregir problemas que puedan dificultar el acceso a las aplicaciones para personas con diversas discapacidades, incluidas las cognitivas.

A continuación, se presentan algunas de las herramientas más efectivas para evaluar la accesibilidad en aplicaciones web<sup>25</sup>.

#### 2.2.2.1. WAVE (Web Accessibility Evaluation Tool)

*WAVE*<sup>26</sup> es una herramienta en línea que permite evaluar la accesibilidad de páginas y aplicaciones web, generando informes visuales que identifican problemas de

<sup>24</sup><https://www.tpgi.com/understanding-the-stanca-act-italys-digital-accessibility-law/>

<sup>25</sup>El análisis de las herramientas se realizó en noviembre de 2024, por lo que algunas de sus características han podido evolucionar desde entonces

<sup>26</sup><https://wave.webaim.org/help>

conformidad con las pautas WCAG. Además, *WAVE* considera también el estándar ARIA.

*WAVE* ofrece extensiones para navegadores (*Chrome*, *Firefox* y *Edge*), lo que permite realizar análisis directamente desde el navegador y acceder a páginas protegidas por contraseñas o con contenido dinámico. La herramienta realiza un análisis detallado para identificar problemas específicos de accesibilidad, utilizando un sistema de iconos visuales superpuestos que destacan los elementos clave en la interfaz. Un ejemplo de este análisis puede observarse en la Figura 2.1, donde se evalúa la accesibilidad de la interfaz de *ChatGPT*. Los resultados del análisis se representan mediante diferentes tipos de iconos:

- Errores (iconos rojos): Señalan problemas críticos, como la ausencia de texto alternativo o etiquetas incorrectas en formularios.
- Advertencias (iconos amarillos): Indican elementos que podrían mejorarse para cumplir con las pautas de accesibilidad.
- Configuraciones correctas (iconos verdes): Resaltan componentes bien implementados, como enlaces accesibles o encabezados semánticamente estructurados.
- Aspectos semánticos (iconos morados y azules): Destacan encabezados, regiones, roles ARIA y otros elementos que facilitan la navegación para usuarios que emplean tecnología asistencial.

El análisis realizado por *WAVE* incluye un panel de resumen, que se puede ver en el lado izquierdo de la Figura 2.1, donde se presentan indicadores cuantitativos sobre:

- Errores críticos: Problemas que afectan directamente la accesibilidad.
- Advertencias: Elementos que requieren revisión para mejorar la experiencia del usuario.
- Problemas de contraste: Problemas relacionados con la legibilidad del texto frente al fondo.
- Características estructurales: Elementos correctamente configurados que contribuyen a la accesibilidad, como encabezados bien definidos.

Adicionalmente, *WAVE* incluye una lista de verificación de accesibilidad cognitiva<sup>27</sup>, que ofrece pautas para abordar necesidades específicas de usuarios con este tipo de discapacidades. Algunos de los puntos de revisión más destacados son:

- Consistencia: Garantizar que la navegación y el diseño sean uniformes en todo el sitio para facilitar la orientación del usuario.

---

<sup>27</sup><https://wave.webaim.org/cognitive>

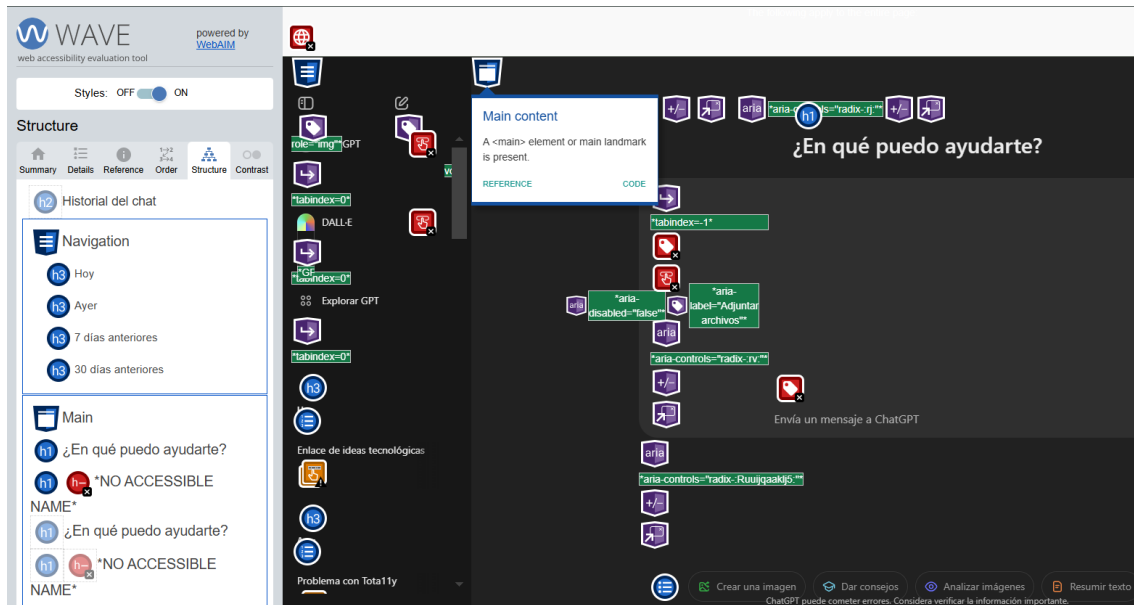


Figura 2.1: Análisis realizado con WAVE de la web de ChatGPT.

- Transformabilidad: Permitir el aumento del tamaño del texto hasta un 200-300 % sin pérdida de funcionalidad y asegurar un contraste adecuado entre el texto y el fondo.
- Multi-modalidad: Proporcionar contenido en diferentes formatos (texto, audio, video) con subtítulos, transcripciones y uso de imágenes que refuercen el mensaje.
- Concentración y estructura: Utilizar espacio en blanco, encabezados claros y listas organizadas, evitando distracciones como animaciones, sonidos de fondo o ventanas emergentes.
- Legibilidad y lenguaje: Usar lenguaje claro, simple y directo, evitando tecnicismos innecesarios, y emplear fuentes legibles (*sans-serif*) y párrafos breves.
- Orientación y errores: Incluir herramientas como migas de pan, ofrecer mensajes de error claros y permitir que el usuario controle los cambios de contenido. También se recomienda proporcionar múltiples métodos de búsqueda y navegación.
- Compatibilidad con tecnología asistencial: Incluir texto alternativo en imágenes, etiquetas descriptivas en formularios, estructura lógica en los encabezados y navegación accesible por teclado.

#### 2.2.2.2. Pa11y Dashboard

*Pa11y*<sup>28</sup> es una herramienta de código abierto para la evaluación de la accesibilidad web que se ejecuta en un entorno de consola, lo que significa que los usuarios

<sup>28</sup><https://github.com/pa11y/pa11y-dashboard>

deben tener conocimientos básicos de la línea de comandos para utilizarla. Su propósito principal es facilitar el análisis automático de sitios web, verificando que cumplan con las WCAG. Para ello, realiza auditorías de accesibilidad en páginas web y genera informes detallados sobre los problemas encontrados. Estos informes proporcionan información clara sobre las áreas que requieren atención, permitiendo a los desarrolladores y diseñadores abordar estos problemas de manera efectiva.

Aunque *Pa11y* proporciona varias ventajas, su uso a través de la línea de comandos puede resultar complejo para usuarios con poca experiencia técnica dificultando el proceso de configuración y análisis. Para superar esta barrera, surge *Pa11y Dashboard*, una interfaz web de código abierto basada en el generador de informes de accesibilidad *Pa11y*. Diseñada para simplificar la gestión y monitorización de la accesibilidad de sitios web de manera eficiente y cómoda. Se caracteriza por lo siguiente:

- Permite analizar simultáneamente múltiples páginas y realizar un seguimiento histórico de los resultados.
- Es altamente personalizable, permitiendo ajustar parámetros como:
  - El nivel de conformidad de las WCAG que se desean evaluar.
  - Configuración de tiempos y periodicidad de los análisis automáticos.
  - Integración con otros entornos o sistemas mediante *APIs*.
- *Pa11y Dashboard* se puede integrar con herramientas de Integración Continua (CI) y Despliegue Continuo (CD), para permitir que los análisis de accesibilidad se realicen de forma automatizada dentro de los flujos de trabajo de desarrollo de software. Así, garantiza que las auditorías de accesibilidad se ejecuten con cada cambio en el código, evitando que los problemas se acumulen y afecten el producto final.
- Permite una visualización centralizada de los resultados, como se muestra en la Figura 2.2, donde cada página analizada se identifica con indicadores claros de conformidad:
  - Errores críticos: Resaltados en rojo, representan problemas graves que afectan directamente la accesibilidad.
  - Advertencias: En naranja, señalan elementos que requieren revisión pero no constituyen incumplimientos estrictos de las WCAG.
  - Notificaciones: En azul, proporcionan información adicional o elementos relevantes a monitorizar.
- Permite profundizar en los resultados de cada página. En la Figura 2.3, se observa un ejemplo de informe detallado generado para ChatGPT. Este informe incluye:
  - Identificación precisa de las reglas WCAG incumplidas (por ejemplo, principios como Perceptible y Robusto).

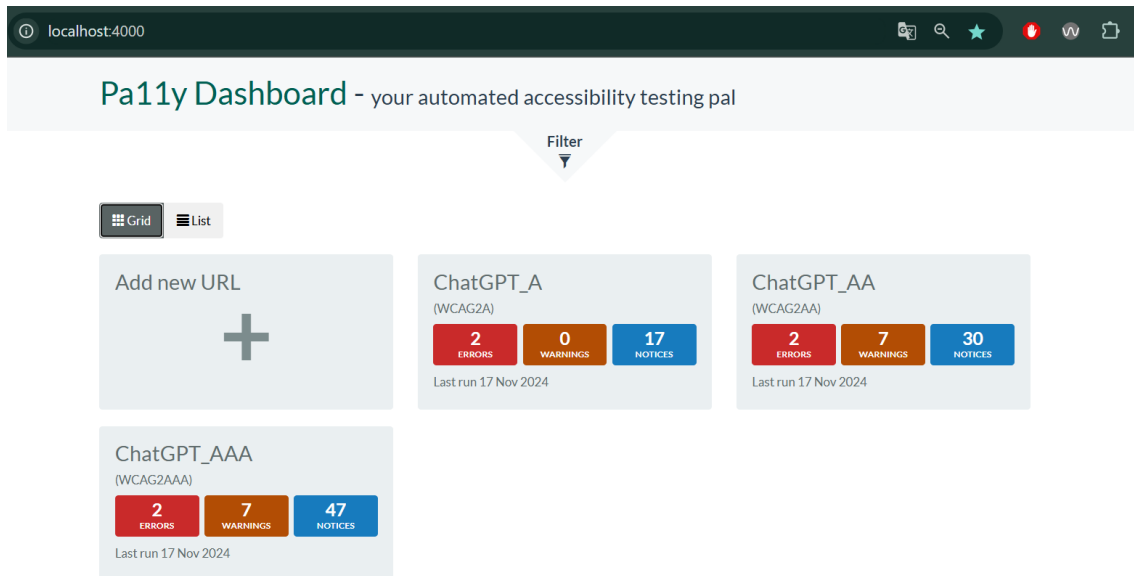


Figura 2.2: Interfaz principal del Pa11y Dashboard con múltiples análisis.

- Mensajes explicativos y contextuales sobre cada problema detectado.
- Opciones para descargar los resultados en formatos *JSON* o *CSV*, facilitando su integración en flujos de trabajo personalizados.
- Vista directa en el navegador para identificar visualmente dónde se encuentran los problemas en la página.

### 2.2.2.3. Axe DevTools

*Axe DevTools*<sup>29</sup> es una herramienta de pago, con una versión de prueba gratuita de 14 días, para evaluar la accesibilidad de las aplicaciones web. Esta herramienta es ampliamente utilizada por desarrolladores debido a su precisión en la identificación de problemas y su compatibilidad con las pautas WCAG. Esta disponible como una extensión para navegadores (*Chrome*, *Firefox*), donde se integra como apartado en la sección de investigación del navegador y como una biblioteca integrable en proyectos de desarrollo. Además, permite realizar auditorías de accesibilidad de forma automática y manual, verifica el uso adecuado de atributos ARIA y examina que los elementos interactivos y las estructuras de encabezado estén correctamente implementados, ayudando a garantizar que el sitio sea navegable para personas que dependen de tecnología asistiva.

En la Figura 2.4 se observa la versión de extensión para navegador donde se detectaron 8 problemas en total, divididos en 3 críticos, 3 serios, 1 moderado y 1 menor realizando el análisis de *ChatGPT*. Dicho análisis se realizó bajo las pautas WCAG 2.1 AA con mejores prácticas activadas, enfocándose únicamente en problemas detectados automáticamente. En la Figura 2.5, se presenta un ejemplo de los

<sup>29</sup><https://www.deque.com/axe/>

Figura 2.3: Informe detallado de accesibilidad para ChatGPT en Pa11y Dashboard.

problemas detectados automáticamente mencionados anteriormente, donde la herramienta destaca un uso incorrecto de atributos ARIA en un elemento *form*. Este tipo de análisis, además de mostrar de que manera fue detectado, indica su impacto, que pautas no cumple y cuando fue encontrado.

*Axe Deque* ofrece una API robusta y un conjunto de herramientas avanzadas, diseñadas para integrarse con marcos de pruebas automatizadas como *Selenium* y *Cypress*. Estas herramientas permiten identificar, diagnosticar y abordar problemas de accesibilidad directamente en el flujo de trabajo de desarrollo, promoviendo la creación de aplicaciones inclusivas desde las etapas iniciales del proceso.

#### 2.2.2.4. SortSite

*SortSite*<sup>30,31</sup> es una herramienta avanzada diseñada para auditar la accesibilidad, usabilidad, cumplimiento de estándares y privacidad en páginas web. Creada por *PowerMapper*, destaca por su capacidad de identificar de manera eficiente problemas relacionados con las WCAG y la Sección 508. Está disponible tanto como aplicación de escritorio, como en versión web, lo que la hace versátil para distintos entornos de trabajo. Su funcionalidad y accesibilidad la hacen una herramienta útil para profesionales involucrados en el desarrollo de sitios web que buscan cumplir con los estándares internacionales de accesibilidad e inclusión. La plataforma permite realizar análisis completos de sitios web generando informes detallados que pueden personalizarse según las necesidades específicas del proyecto. Estos informes son importantes para identificar y corregir barreras de accesibilidad, errores en el diseño

<sup>30</sup><https://www.powermapper.com/products/sortsite/>

<sup>31</sup><https://rapidforms.co/blog/22-web-accessibility-testing-tools-for-forms-2024/>

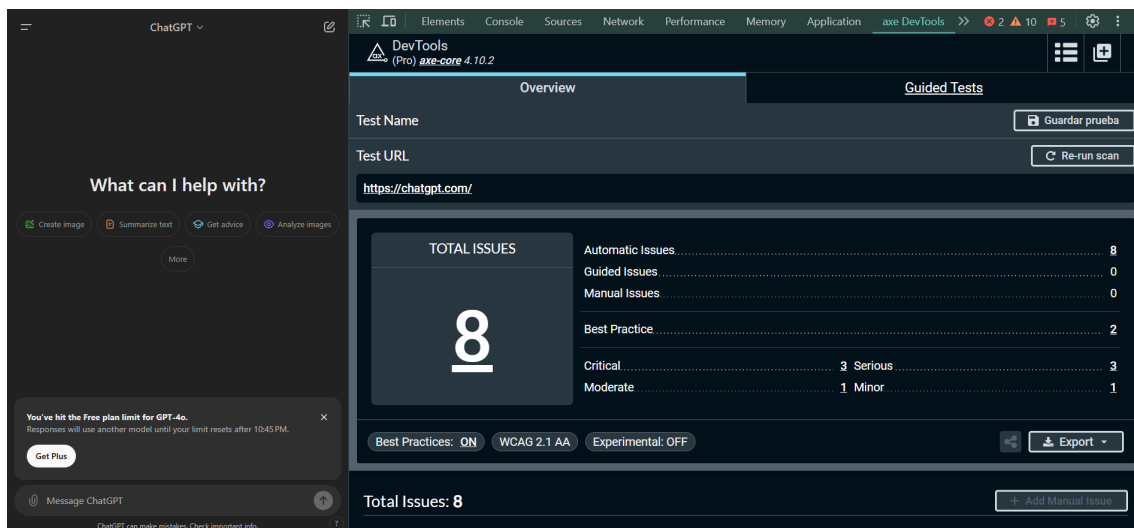


Figura 2.4: Interfaz de Axe DevTools mostrando una vista general de los problemas encontrados en ChatGPT

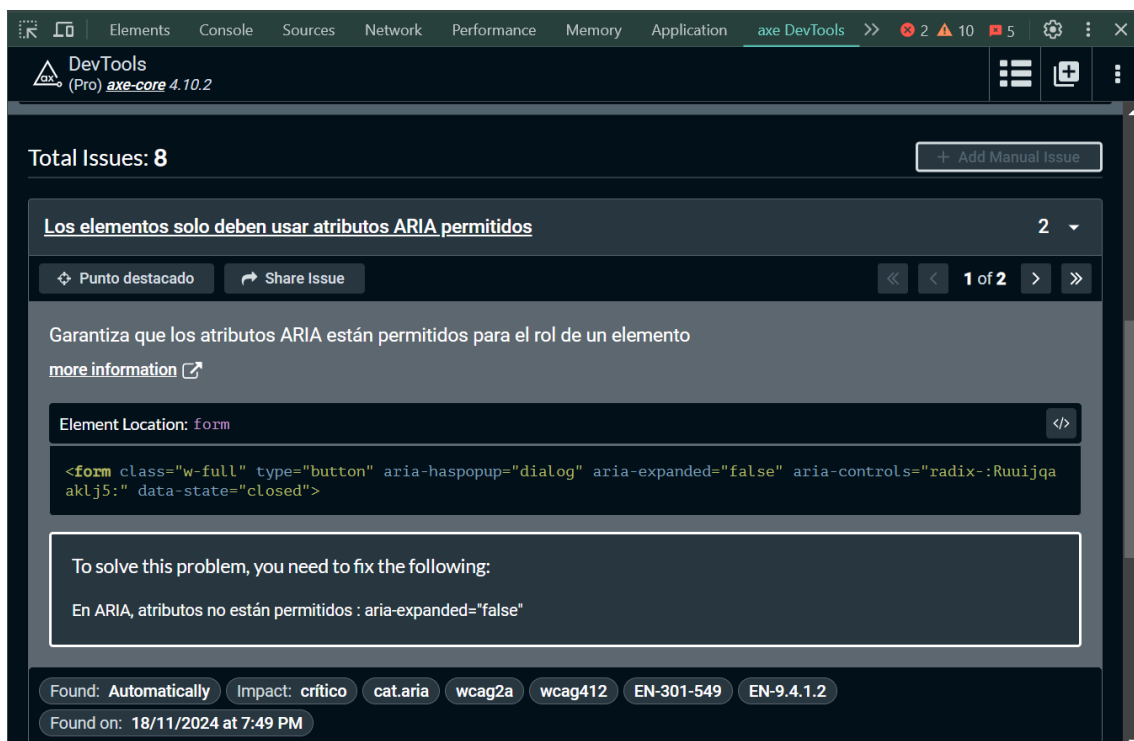


Figura 2.5: Axe DevTools mostrando uno de los problemas en el análisis de ChatGPT

y problemas relacionados con la accesibilidad. Como se muestra en la Figura 2.6, tras analizar *ChatGPT*, la herramienta evalúa diversos parámetros y proporciona recomendaciones específicas para implementar mejoras. Este proceso se detalla en el informe, tal como se ilustra en la Figura 2.7:

- Errores (circulo rojo): Señalan problemas y da opciones para solventarlos.
- Advertencias (circulo naranja): Señala aspectos que podrían optimizarse para cumplir con las directrices de accesibilidad.
- Configuraciones correctas (circulo azul): Destacan componentes correctamente implementados, como enlaces accesibles o encabezados con una estructura semántica adecuada.

Asimismo, *SortSite* ofrece opciones para personalizar las características que se desean analizar:

- Idiomas: Analiza también si hay faltas de ortografía pero actualmente solo deja que sea en inglés, francés o que no se analice.
- Accesibilidad: Permite seleccionar que WCAG se quiere aplicar o si se quiere comprobar la Sección 508.
- Compatibilidad: Mide la compatibilidad con los distintos buscadores.
- Estándares: Analiza el cumplimiento de estándares web fundamentales para garantizar un sitio funcional, accesible y compatible. Evalúa el código HTML, verificando su semántica, validez y obsolescencia, y el uso eficiente de CSS y JavaScript, asegurando prácticas modernas, compatibilidad entre navegadores y optimización para rendimiento. También mide la adecuación a criterios de SEO técnico, como etiquetas meta, enlaces internos y *Core Web Vitals*, además de analizar accesibilidad (etiquetas ARIA, texto alternativo) y estándares de seguridad, como HTTPS y encabezados clave.
- Usabilidad: Mide qué tan fácil, eficiente y agradable es para los usuarios navegar por ella y realizar tareas específicas.

El uso de la herramienta *ShortSite* destaca por su simplicidad y eficiencia al permitir a los usuarios gestionar y optimizar enlaces de manera rápida y accesible. Su interfaz sencilla facilita la creación de enlaces cortos, mejorando la experiencia de usuario al reducir tiempos y mejorar la organización de la información. En general, su facilidad de uso y su enfoque en la accesibilidad hacen de *ShortSite* una herramienta valiosa para aquellos que buscan una solución rápida y eficaz para mejorar la accesibilidad.

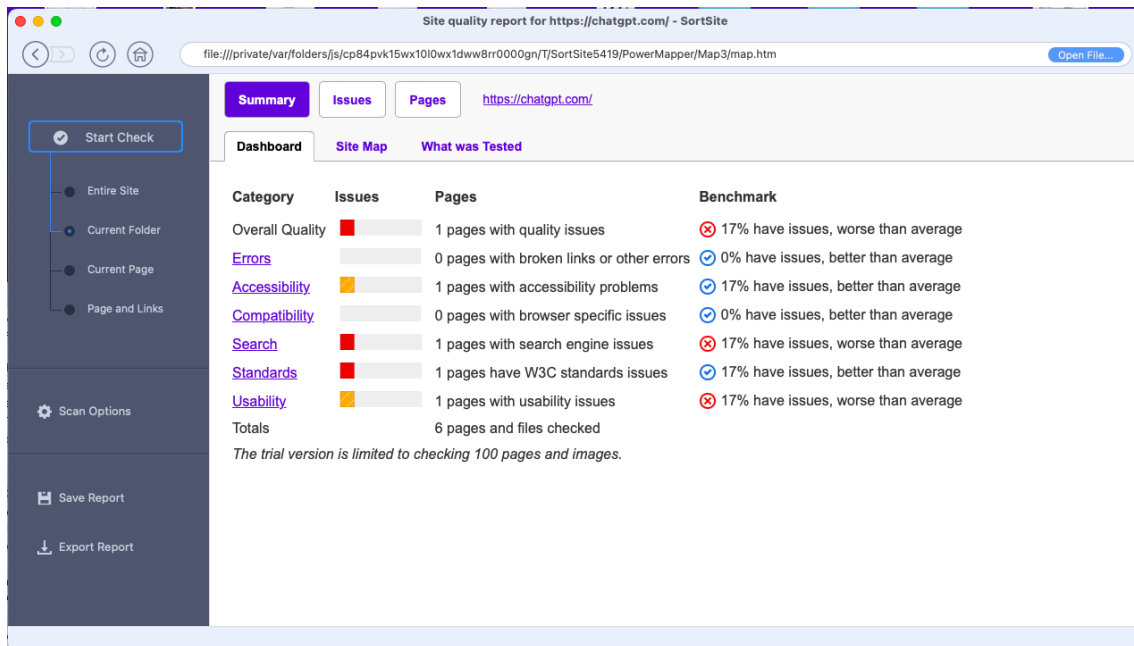


Figura 2.6: La Interfaz de SortSite localizando fallos de accesibilidad en ChatGPT

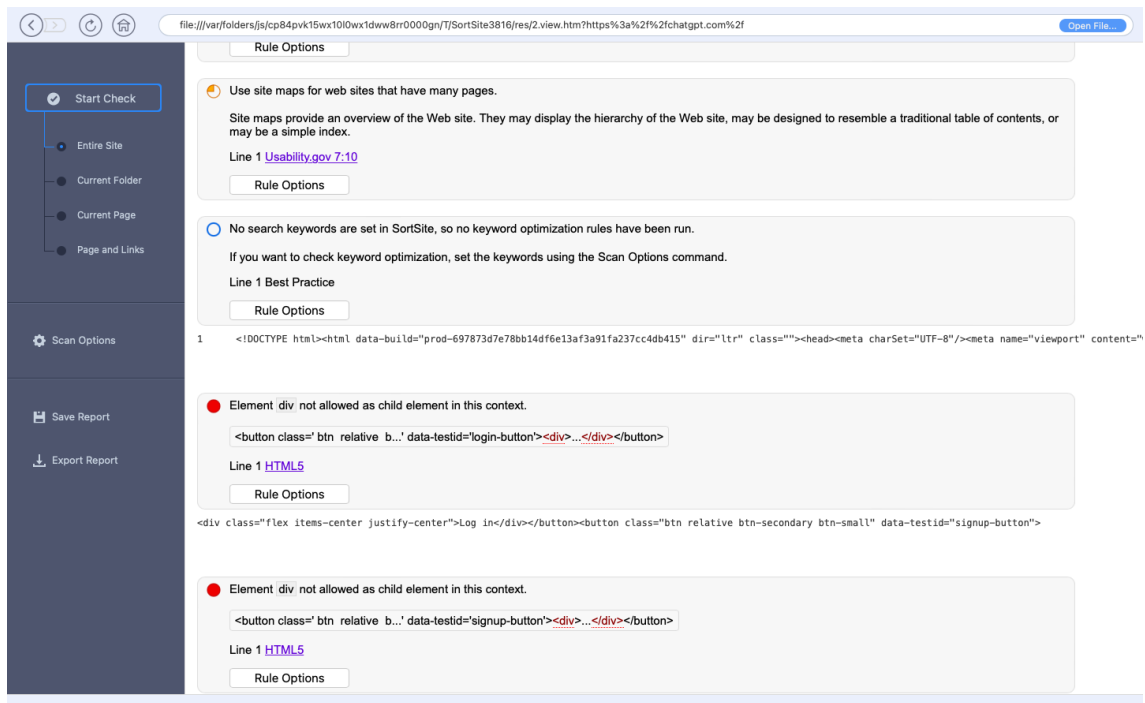


Figura 2.7: SortSite analizando diferentes parámetros y proponiendo posibles mejoras

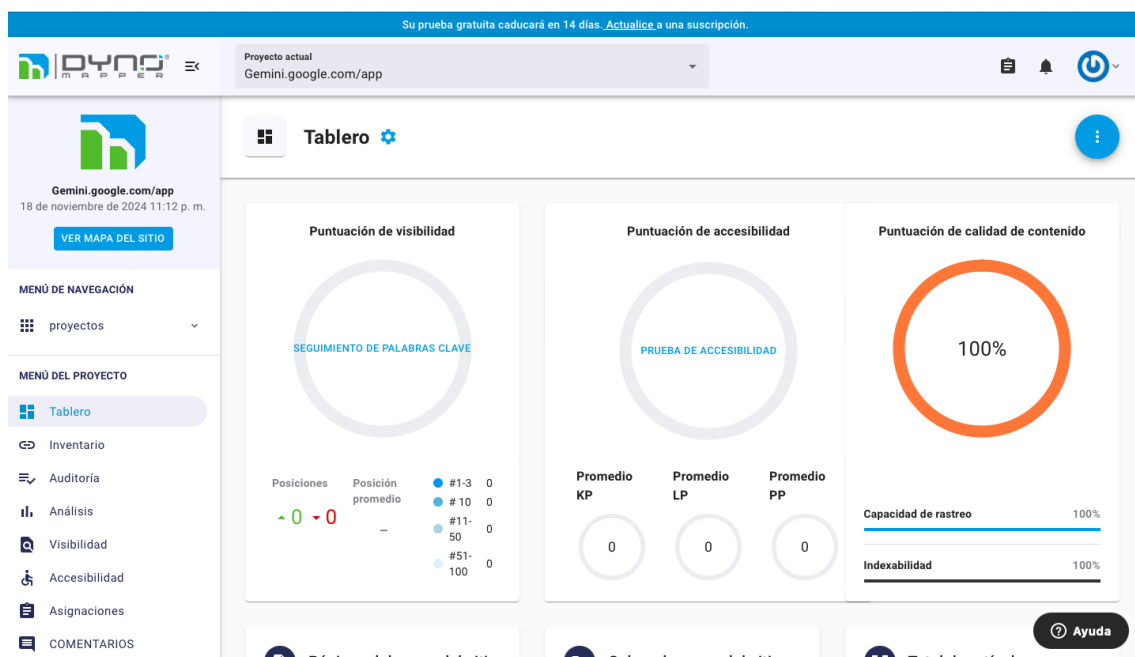


Figura 2.8: DYNO MAPPER analizando Gemini

### 2.2.2.5. DYNO Mapper

*DYNO Mapper*<sup>32,33</sup>, desarrollada por *Indigo Design Company*, es una herramienta web avanzada para la auditoría y gestión de contenido en páginas web, que integra funciones de accesibilidad, como la detección de problemas (como texto alternativo ausente), contrastes de color inadecuados y enlaces rotos, SEO y creación de mapas de sitio. Está desarrollada principalmente para optimizar la experiencia del usuario y el rendimiento del sitio y permite identificar y resolver problemas de accesibilidad en base a las pautas WCAG y la Sección 508. Su interfaz ofrece un panel centralizado donde los usuarios pueden acceder a proyectos activos y ver métricas clave como accesibilidad, visibilidad y calidad, como se observa en la Figura 2.8, donde se usa la aplicación de IA generativa *Gemini*, como ejemplo para el análisis. Además, cuenta con un generador de mapas visuales que ofrece una representación clara, tras hacer el análisis, del estado de su accesibilidad ayudando a identificar áreas problemáticas y planificar mejoras de manera visual y comprensible, como se aprecia en la Figura 2.9. *DYNO Mapper* es compatible con dispositivos móviles y, aunque es principalmente una herramienta de pago, ofrece una versión gratuita que permite realizar hasta 3 análisis de páginas web. En el menú derecho de la Figura 2.8, donde la herramienta hace un análisis de una página web, se pueden observar los siguientes elementos:

- Panel: El panel principal de *DYNO Mapper* proporciona una visión general del estado del sitio web, mostrando métricas clave relacionadas con contenido, accesibilidad, visibilidad y analítica. Este apartado facilita un acceso rápido a proyectos recientes, mapas del sitio y datos esenciales, permitiendo a los

<sup>32</sup><https://dynamapper.com>

<sup>33</sup><https://www.browserstack.com/guide/wcag-ada-testing-tools>

**DYNO MAPPER**  
Accessibility Testing

https://gemini.google.com/aj

7 KNOWN PROBLEMS    25 LIKELY PROBLEMS    179 POTENTIAL PROBLEMS

**1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background.**

**Success Criteria 1.4.10 Reflow (AA)**

`<meta>` The page does not allow scaling.

```
<meta name="viewport" content="initial-scale=1,minimum-scale=1,width=device-width,interactive-widjet=resize-content">
```

**Repair:** Do not disable user scaling from meta tag

---

**Priority 1**

**1.1 Provide a text equivalent for every non-text element.**

`<script>` `script` must have a `noscript` section.

```
<script type="text/javascript" async="" src="https://www.googletagmanager.com/gtag/js?id=G-T0C69Z071S&l=dataLayer&cx=c&gtm=45He4be0v9113054662za200" nonce=""></script>
```

**Repair:** Add a `<noscript>` section immediately following the `<script>` that provides the same functionality as the `<script>`.

Figura 2.9: Interfaz de DYNO Accessibility Testing mostrando los problemas principales



usuarios evaluar de un vistazo el desempeño general del sitio. Además, se puede personalizar para resaltar la información más relevante según las necesidades del proyecto.

- **Inventario:** Como se muestra en la Figura 2.10, el apartado de inventario crea un listado completo y estructurado de todas las páginas, archivos y recursos del sitio web. Incluye detalles importantes como títulos, descripciones y enlaces, ayudando a visualizar la estructura del contenido. También identifica problemas técnicos, como enlaces rotos o duplicidad de contenido, ofreciendo una base sólida para comprender el estado actual del sitio y planificar mejoras.
- **Auditoría:** La sección de auditoría realiza un análisis exhaustivo del contenido del sitio web para evaluar su calidad, relevancia y optimización como se puede ver en la Figura 2.11, detecta problemas técnicos como enlaces rotos, contenido duplicado y errores en las redirecciones. También examina etiquetas de metadatos y encabezados para garantizar que estén alineados con las mejores prácticas de SEO, ofreciendo recomendaciones concretas para mejorar el desempeño del sitio.
- **Analítica:** *DYNO Mapper* integra datos de Google Analytics para ofrecer métricas sobre tráfico, comportamiento de los usuarios y conversiones. Este apartado permite identificar las páginas más visitadas, analizar patrones de uso y medir el desempeño del sitio en función de objetivos específicos. La información proporcionada ayuda a tomar decisiones informadas para optimizar la experiencia del usuario y el impacto del contenido.
- **Visibilidad:** En el apartado de visibilidad, *DYNO Mapper* analiza el rendimiento del sitio en motores de búsqueda, evaluando factores como palabras clave, clasificaciones y estado de indexación. Este módulo identifica oportunidades para mejorar el posicionamiento orgánico del sitio y permite monitorear su presencia en línea, asegurando un impacto positivo en el tráfico y en la estrategia SEO.
- **Accesibilidad:** La sección de accesibilidad realiza pruebas para verificar que el sitio cumpla con las pautas de accesibilidad, WCAG tal y como se muestra en la Figura 2.9. También genera reportes detallados con recomendaciones para mejorar la inclusión y cumplir con la Sección 508 tal y como se observa en la Figura 2.12.
- **Asignaciones:** Esta funcionalidad permite asignar tareas específicas relacionadas con proyectos, como la actualización de contenido o la solución de problemas detectados. Facilita la colaboración entre equipos al definir responsabilidades, plazos y objetivos claros. También permite rastrear el progreso de las tareas asignadas, asegurando que se cumplan dentro del plazo establecido y mejorando la gestión del proyecto.
- **Comentarios:** El apartado de comentarios proporciona una herramienta para recopilar y gestionar observaciones sobre contenido, páginas o proyectos específicos. Es ideal para centralizar la retroalimentación de equipos y clientes,

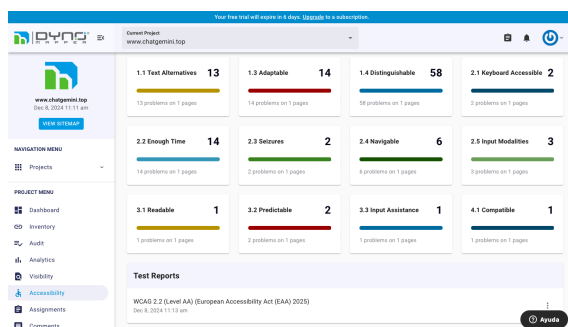


Figura 2.12: Interfaz de DYN0 Accessibility mostrando los problemas principales

permitiendo agregar notas directamente en los mapas del sitio o en elementos concretos. Este sistema mejora la comunicación y ayuda a documentar las necesidades de cambio de forma eficiente.

Estos apartados hacen de *DYNO Mapper* una solución integral para gestionar y optimizar sitios web, combinando análisis técnico, estrategias de mejora y herramientas de colaboración.

#### 2.2.2.6. AChecker

*AChecker*<sup>34</sup> es una herramienta gratuita y automatizada de verificación de accesibilidad que evalúa páginas *HTML* para garantizar que sean accesibles para todas las personas, incluidas aquellas con discapacidades, mediante el uso de tecnologías de asistencia. *AChecker* destaca por permitir revisar los resultados y ajustar la evaluación según el contexto y la interpretación humana, algo que no suelen hacer otras herramientas automatizadas. Además, en su interfaz principal, permite elegir entre diversos estándares de accesibilidad para evaluar la conformidad. Los estándares que ofrece son los siguientes:

- **WCAG:** En *AChecker*, las WCAG se aplican como base para evaluar si el contenido web cumple con los principios fundamentales de accesibilidad. Por ejemplo, utiliza diferentes niveles de conformidad para analizar una página HTML, PDFs, dispositivos móviles y computadoras
- **BITV 1.0:** *AChecker* permite a los usuarios seleccionar la norma BITV 1.0 para evaluar la accesibilidad según los estándares alemanes. Por ejemplo, puede identificar si un sitio web utiliza combinaciones de colores que sean perceptibles para personas con daltonismo o si cumple con los requisitos de contraste mínimo establecidos por esta normativa.
- **Sección 508:** La Sección 508 se aplica en *AChecker* para verificar que los sitios web y documentos electrónicos cumplan con los estándares de accesibilidad exigidos por las agencias gubernamentales de los Estados Unidos. Por ejemplo, la herramienta puede detectar enlaces que no tienen descripciones claras o

<sup>34</sup><https://github.com/cg-a11y/AChecker>

Herramientas	WAVE	Pa1ly Dashboard	Axe DevTools By Deque	SortSite	DYNO Mapper	AChecker
Tipo	Extensión de navegador	Aplicación web, CI/CD	Extensión de navegador + API	Herramienta de escritorio	Herramienta web	Herramienta web
Usabilidad	Auditoría manual y visual	Auditorías automatizadas	Auditorías mixtas	Auditoría general (SEO, Acc.)	Análisis de contenido	Análisis de contenido
Pautas Aplicadas	WCAG 2.1, ARIA	WCAG 2.1	WCAG 2.1, ARIA, Sección 508	WCAG 2.1, Sección 508	WCAG 2.1	WCAG 2.1, Sección 508
Análisis	Mixto	Automático	Mixto	Mixto	Automático	Automático
Compatibilidad	Chrome, Firefox	CI/CD (Jenkins, GitHub Actions)	Chrome, Edge + API	Windows, macOS	Basado en navegador	Basado en navegador
Enfoque	Problemas en el DOM	Automatización de pruebas	Recomendaciones prácticas	Accesibilidad y SEO	Organización de contenido	Accesibilidad, velocidad y SEO

Tabla 2.1: Comparativa de herramientas de accesibilidad

formularios sin etiquetas asociadas a sus campos de entrada, que son errores comunes que afectan a usuarios con discapacidades visuales o cognitivas.

- **Stanca Act (Ley 4/2004):** Al seleccionar la opción de evaluación basada en la Stanca Act, *AChecker* analiza sitios web italianos para garantizar que cumplan con las normativas de accesibilidad locales. Por ejemplo, la herramienta puede identificar si los formularios de contacto tienen instrucciones claras y visibles para todos los usuarios, o si los elementos interactivos, como botones y enlaces, son accesibles mediante el teclado, un requisito clave de esta normativa.

Estos estándares permiten realizar evaluaciones detalladas de accesibilidad en función de diferentes normativas internacionales y contextos legales, como se muestra en la Figura 2.13.

Como se observa en la Figura 2.13, *AChecker* también permite medir el SEO y la velocidad junto con la accesibilidad de la página web introducida. En la Figura 2.14, se observa el análisis realizado para *ChatGPT*. En este análisis, se aprecia que la puntuación de accesibilidad de la aplicación web es de 88, lo que indica un buen nivel de accesibilidad, aunque con oportunidades de mejora. Para ello, la herramienta proporciona un listado de fallos, y en la sección visible de la Figura 2.14, destaca un error relacionado con los atributos ARIA, donde algunos valores `aria-*` no coinciden con los roles asignados. El mensaje de error aclara que cada rol ARIA soporta un subconjunto específico de atributos `aria-*`. Además, se incluye un ejemplo concreto de código *HTML* que muestra un botón con varios atributos `aria-*` incorrectamente asignados, junto con un enlace para aprender a hacer coincidir correctamente los atributos ARIA con sus roles.

### 2.2.2.7. Conclusiones




En la Tabla 2.1 se presenta una comparativa detallada de las herramientas analizadas en esta sección, donde se analizan aspectos clave como el tipo de herramienta, el enfoque de usabilidad, las pautas de accesibilidad aplicadas, el tipo de análisis realizado (manual, automático o mixto), la compatibilidad con distintos entornos y navegadores, y el enfoque funcional.

## Check web page accessibility

This tool calls [AChecker](#) or [Lighthouse](#) for web and [Tingun](#) for PDF accessibility checking by webservice.

Web Page URL  
<https://chatgpt.com/>

Guideline:

<input type="radio"/> WCAG 2.1 AA 	<input type="radio"/> WCAG 1.0 AA <small>HTML</small>
<input checked="" type="radio"/> WCAG 2.1 AA 	<input type="radio"/> WCAG 1.0 AAA <small>HTML</small>
<input type="radio"/> WCAG 2.0 AA 	<input type="radio"/> WCAG 1.0 A <small>HTML</small>
<input type="radio"/> WCAG 2.0 AA <small>HTML</small>	<input type="radio"/> BITV 1.0 (Level 2) <small>HTML</small>
<input type="radio"/> WCAG 2.0 AAA <small>HTML</small>	<input type="radio"/> Section 508 <small>HTML</small>
<input type="radio"/> WCAG 2.0 A <small>HTML</small>	<input type="radio"/> Stanca Act <small>HTML</small>

Include:

- Best Practices
- SEO
- Speed

Figura 2.13: Interfaz de AChecker junto con las pautas de accesibilidad que evalúa.

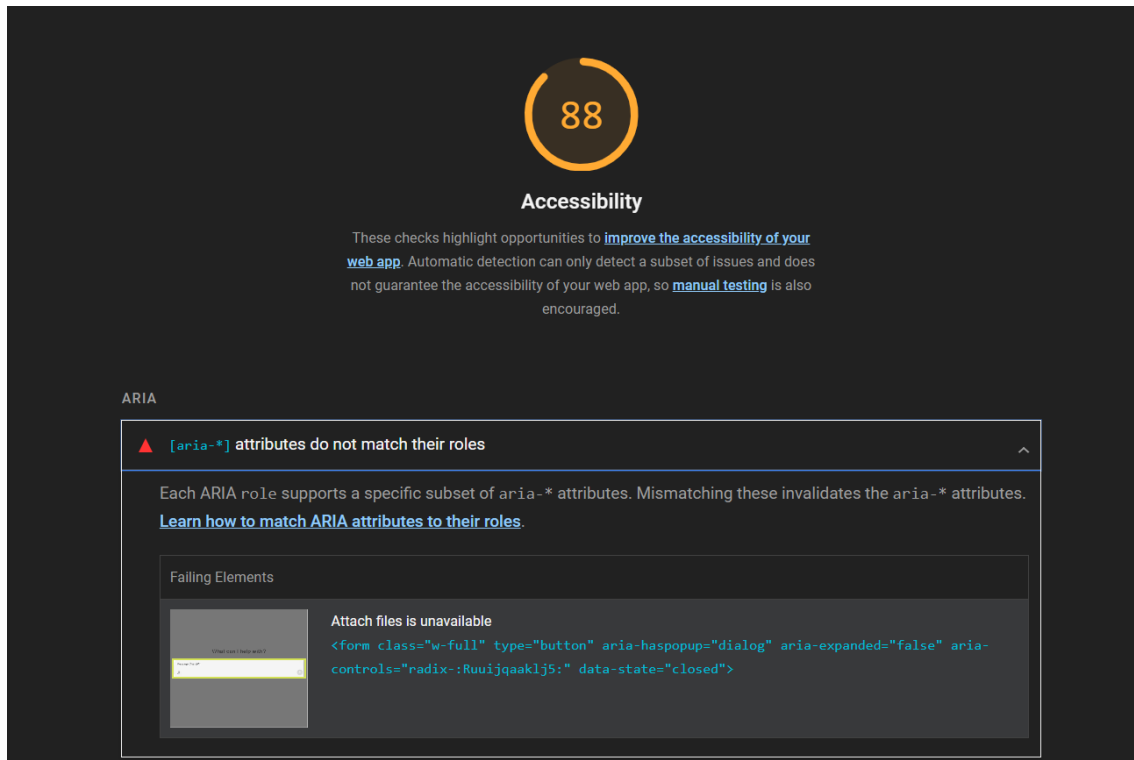


Figura 2.14: Análisis de accesibilidad de ChatGPT con AChecker.

Tras el análisis realizado, se concluye que las herramientas *AChecker*, *WAVE* y *Pa11y Dashboard*, además de ser gratuitas, son especialmente útiles y fáciles de utilizar para identificar problemas de accesibilidad y proponer soluciones. En contraste, herramientas como *Axe DevTools*, *SortSite* y *DYNO Mapper* ofrecen funcionalidades más avanzadas, pero sus modelos de pago las hacen menos accesibles para proyectos con presupuestos más ajustados. Por tanto, mientras que las opciones gratuitas son más que suficientes para necesidades básicas, las opciones de pago resultan más adecuadas para proyectos complejos.

## 2.3. Inteligencia artificial generativa

Según la RAE<sup>35</sup>, la inteligencia artificial es la “*disciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones comparables a las que realiza la mente humana, como el aprendizaje o el razonamiento lógico*”. Es decir, su objetivo es crear y diseñar sistemas capaces de reconocer patrones, tomar decisiones y resolver problemas de manera autónoma aprendiendo de datos existentes.

Por otro lado, la inteligencia artificial generativa, es un subtipo de inteligencia artificial que no solo hereda estas capacidades, sino que se distingue por su habilidad de crear contenido nuevo e innovador como conversaciones, historias, imágenes, vídeos y música basándose en los datos con los que ha sido entrenada. Lo que define a una IA generativa es su capacidad para crear nuevo contenido a partir de datos

<sup>35</sup><https://dle.rae.es/inteligencia>

existentes. Es “generativa” porque, en lugar de solo reconocer patrones o tomar decisiones, produce algo nuevo y original que no existía previamente. Es esta capacidad creativa lo que la distingue de otros tipos de inteligencia artificial.

La IA generativa representa el siguiente paso en la evolución de la inteligencia artificial. Para ello, se entrena analizando grandes volúmenes de datos diversos, lo que le permite aprender diferentes materias como lenguajes humanos y de programación, arte, química o biología, entre otros. A partir de este aprendizaje, puede reutilizar los datos de entrenamiento para resolver nuevos problemas. Esto es posible gracias a modelos profundos, como las redes neuronales y las arquitecturas *Transformers*. Cuanto más diversos y extensos sean los datos, más riqueza de patrones podrá aprender el modelo.

La inteligencia artificial generativa está impulsando el tercer paradigma de interacción entre humanos y computadoras, descrito por Nielsen (2023) como “especificación de resultados basada en intenciones”. Este paradigma supone un cambio significativo frente a los dos anteriores (procesamiento por lotes y las interfaces basadas en comandos). El primer paradigma, el procesamiento por lotes, requería que los usuarios definieran por completo las tareas a ejecutar mediante lotes de instrucciones que se procesaban de forma diferida, sin retroalimentación ni posibilidad de corrección inmediata, lo que resultaba lento y poco intuitivo. El segundo paradigma, la interacción basada en comandos, permitió turnos entre el usuario y la máquina, ofreciendo la posibilidad de ajustar los comandos después de recibir resultados. Este enfoque evolucionó a través de sistemas como las líneas de comandos, los terminales textuales y las interfaces gráficas de usuario (GUI), que dominaron la interacción durante más de 60 años gracias a su capacidad de mostrar el estado del sistema de forma visual y accesible. El tercer paradigma, representado por las interfaces de IA generativa, permite a los usuarios especificar el resultado deseado sin detallar los pasos necesarios para alcanzarlo. Aunque esta aproximación promete interacciones más naturales y accesibles, Nielsen advierte que estas interfaces están aún en una etapa muy temprana de desarrollo, con problemas de usabilidad importantes, como la necesidad de “ingenieros de prompts” para obtener buenos resultados y la dificultad para entender cómo se generan las respuestas. A pesar de su potencial transformador, queda mucho camino por recorrer para hacer las interfaces de las IA generativas actuales intuitivas y eficientes. A medida que evolucionen, se espera que las interfaces futuras integren elementos de este paradigma con características de las GUI tradicionales, logrando un enfoque híbrido más intuitivo y eficiente.

Para que las aplicaciones que utilizan inteligencia artificial generativa sean más fáciles de usar por personas con discapacidad cognitiva, es necesario que incluyan ciertas características que hagan su uso más sencillo y comprensible. Según Zhu y Luo (2023) y Rosner y Trewin (2019) algunas mejoras importantes a implementar son:

- **Experiencia personalizada:** La IA debería adaptarse a las necesidades de cada usuario, ofreciendo funciones más simples o complejas según lo que cada uno pueda manejar. Esto incluye hacer las tareas más fáciles o sugerir pasos basados en lo que el usuario ya ha hecho.
- **Interfaz simple y fácil de entender:** La IA debería tener una pantalla clara, con

pocas opciones y botones fáciles de identificar para que el usuario no se sienta abrumado. Usar iconos y palabras sencillas puede ayudar a evitar confusiones y que la persona se sienta abrumada.

- Instrucciones claras y respuestas rápidas: La IA debe explicar de forma fácil qué está haciendo y qué necesita que el usuario haga. También debería dar respuestas rápidas para que la persona sepa qué ocurre en cada paso.
- Diferentes formas de interactuar: Permitir que la IA se controle con la voz, el texto o gestos, puede ayudar a que las personas con diferentes capacidades puedan usarla sin problema. Las interfaces de voz, por ejemplo, pueden ser muy útiles si alguien tiene problemas para escribir.
- Ayudas visuales y auditivas: Usar colores, imágenes, y sonidos claros puede guiar mejor al usuario y ayudar a quienes tienen dificultades con la lectura o la comprensión. También, añadir videos explicativos.
- Gestión de errores sencilla: Si ocurre un error, la IA debe explicarlo de manera simple, sin usar palabras técnicas difíciles, y sugerir cómo arreglarlo sin hacer sentir al usuario que cometió un gran error.
- Recordar lo que el usuario necesita: La IA debería ser capaz de recordar lo que el usuario prefiere y ofrecer recordatorios o explicaciones adicionales para que no se pierda o se confunda en el proceso.

A continuación, se presenta una breve descripción de algunas de las aplicaciones de inteligencia artificial generativa de texto más utilizadas<sup>36</sup>.

### 2.3.1. *ChatGPT*

Basada en el modelo de lenguaje GPT-4 de OpenAI, *ChatGPT*<sup>37,38</sup> es una herramienta de inteligencia artificial generativa que permite responder preguntas, generar contenido, realizar tareas de programación y resumir información de manera clara y eficiente. En España, *ChatGPT* está disponible tanto en un plan gratuito con funciones básicas como en versiones de pago que ofrecen capacidades más ampliadas. Aunque ambas versiones comparten características similares, la principal diferencia radica en el acceso a estas funcionalidades. En la versión gratuita, los usuarios pueden realizar un número limitado de peticiones utilizando el modelo de lenguaje de la versión de pago (GPT-4o), que incluye la opción de cargar archivos e imágenes y crear imágenes con *DALL-E*. Una vez que se alcanza ese límite, el modelo de lenguaje cambia automáticamente a la versión gratuita (GPT-3.5), que solo permite la generación de texto. La versión de pago optimiza la experiencia al ofrecer mayor velocidad, prioridad en horas pico y un número superior de peticiones diarias.

<sup>36</sup>El análisis de las herramientas se realizó en noviembre de 2024, por lo que algunas de sus características han podido evolucionar desde entonces

<sup>37</sup><https://openai.com/gpt4>

<sup>38</sup><https://www.techtarget.com/whatis/definition/ChatGPT>

Además, incluye acceso mejorado a funciones avanzadas, como la edición de imágenes con *DALL-E (inpainting)*, el manejo de contextos más amplios, y la capacidad de cargar archivos e imágenes, junto con acceso temprano a herramientas nuevas, garantizando una experiencia más completa y eficiente. Como se puede ver en la Figura 2.15, la interfaz es limpia y sencilla, sin muchas distracciones. Los usuarios pueden acceder fácilmente al cuadro de texto donde se introduce las preguntas o comandos. Además tiene tareas preconfigurada que incluyen botones interactivos para creación de imágenes, generación de resúmenes, o incluso la programación de *scripts*. Estos botones optimizan la interacción, permitiendo a los usuarios realizar tareas complejas de forma más rápida y sencilla.

*ChatGPT* puede recordar información compartida en conversaciones anteriores, como preferencias o detalles importantes, para ofrecer una experiencia más personalizada. Es importante aclarar que *ChatGPT* no realiza búsquedas automáticas en internet; solo utiliza la información que tiene integrada en su modelo o las herramientas activas que permiten consultar la web.

*ChatGPT* está disponible como aplicación web y como aplicación móvil para Android e iOS e incluye una API que permite integrarlo en aplicaciones de terceros. Para ambas versiones, *ChatGPT* ofrece las siguientes funcionalidades:

- Carga de documentos e imágenes: Se pueden cargar, de manera limitada en la versión gratuita, archivos e imágenes para hacerle consultas sobre estos e incluso que haga cambios sobre los propios archivos y los genere con los cambios solicitados para poder ser descargados de nuevo por el usuario.
- Autocompletar la consulta: Como se puede ver en la Figura 2.16, a medida que el usuario empieza a escribir su pregunta o comando en el cuadro de texto, la herramienta sugiere opciones basadas en el contenido parcial introducido. Estas sugerencias son generadas en tiempo real y están diseñadas para ofrecer alternativas que podrían coincidir con la intención del usuario.
- Historial de conversaciones: Permite gestionar un historial de conversaciones, como se puede ver en la parte izquierda de la Figura 2.17, lo que facilita la consulta de interacciones pasadas y mejora la eficiencia y la experiencia de usuario. Esta función es útil para usuarios que deseen retomar un tema previamente tratado o acceder a información específica sin necesidad de repetir la consulta.
- *Chats* personalizados: Permiten configurar y organizar conversaciones según proyectos o áreas temáticas específicas, como se muestra en la Figura 2.18. Además, es posible ajustar el tono y estilo de las respuestas, optimizar el tiempo evitando repeticiones y mantener la privacidad al separar la información. Esto los hace ideales para gestionar múltiples proyectos de forma eficiente y adaptada a cada necesidad.
- Evaluación de respuestas: Los usuarios pueden evaluar la calidad de las respuestas indicando si fueron correctas, como se puede ver en la Figura 2.19, lo que puede ayudar a mejorar la precisión del modelo en futuras interacciones.

- Ajuste de respuestas: Los usuarios tienen la opción de reenviar o volver a ejecutar una consulta para obtener variaciones o respuestas más precisas, como se puede ver en la Figura 2.20. Esta retroalimentación continua contribuye a una experiencia más personalizada y eficiente.

La aplicación web presenta las siguientes características que no posee la versión móvil:

- Modificación de sección de respuesta: Los usuarios pueden modificar secciones de la respuesta que proporciona *ChatGPT*. Esto permite a los usuarios personalizar y ajustar las respuestas proporcionadas, enfocándose en áreas específicas para hacer consultas más concretas. Esto incluye la edición directa de partes de la respuesta, como solicitar más detalles, eliminar información irrelevante o reformular secciones para mayor claridad o precisión. Esta interacción dinámica facilita un proceso colaborativo, donde las respuestas se adaptan progresivamente a las necesidades del usuario, permitiendo obtener información más útil y específica sin reiniciar la conversación como se puede ver en la Figura 2.23.
- Múltiples respuestas: Se ofrecen múltiples respuestas a una misma consulta, como se muestra en la Figura 2.21, permitiendo al usuario elegir la opción más adecuada según sus necesidades.

En cuanto a la aplicación móvil, la característica que la diferencia de la versión web, es la funcionalidad de voz, representada por el icono del micrófono y las ondas en la barra de texto en la Figura 2.22, permitiendo realizar consultas por voz pudiendo elegir hasta el tono de voz que genera la respuesta. Además, al abrir la aplicación móvil, aparece el teclado, como se puede ver en la Figura 2.22, facilitando la escritura de la consulta.

### 2.3.2. *ChatSonic*

*ChatSonic*<sup>39,40</sup> es una aplicación de inteligencia artificial generativa desarrollada por *Writesonic*, diseñada para crear contenido tanto textual como visual mediante comandos de texto y voz. Utiliza procesamiento de lenguaje natural combinado con búsqueda en tiempo real. En la Figura 2.24 se puede observar como es una búsqueda en *ChatSonic*, como se puede apreciar la interfaz es simple y permite interactuar mediante texto o voz, brindando flexibilidad a los usuarios y además permiten realizar búsquedas en tiempo real en internet. Esto significa que puede proporcionar información más actualizada para responder preguntas sobre temas recientes o en constante cambio, como eventos actuales, noticias o lanzamientos de productos. Su integración con *Google* facilita el acceso a datos actualizados y proporciona resultados más relevantes para los usuarios. Además, está disponible como aplicación web y ofrece una API que permite integrarse en sistemas externos y tiene memoria de búsquedas antiguas. La API de *ChatSonic* es compatible con varios lenguajes de

<sup>39</sup><https://writesonic.com/chat?ref=offerone>

<sup>40</sup><https://aitools.fyi/blog/chatsonic-unleashed-the-dawn-of-next-gen-ai-conversations>

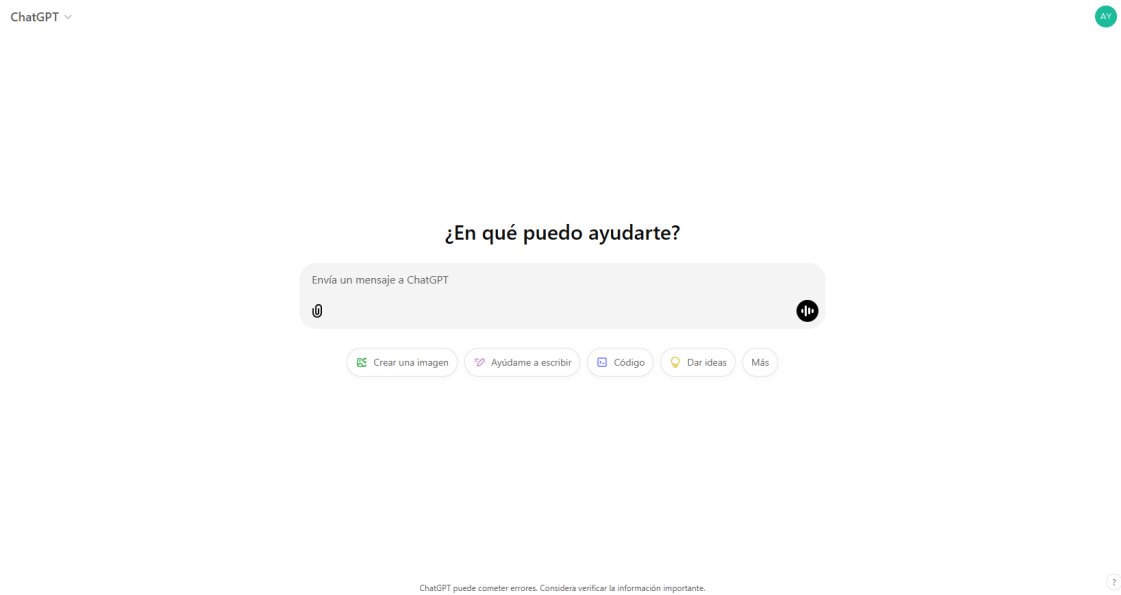


Figura 2.15: Interfaz web de ChatGPT al abrir la aplicación

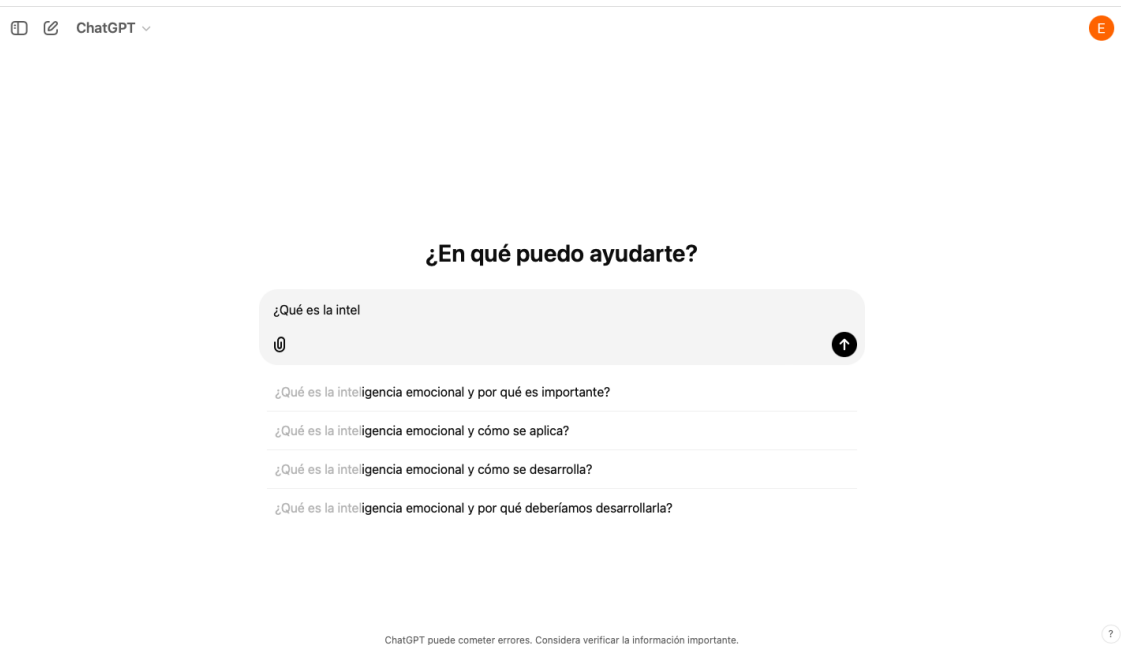


Figura 2.16: Búsqueda y Autocompletado de texto en ChatGPT en su versión web

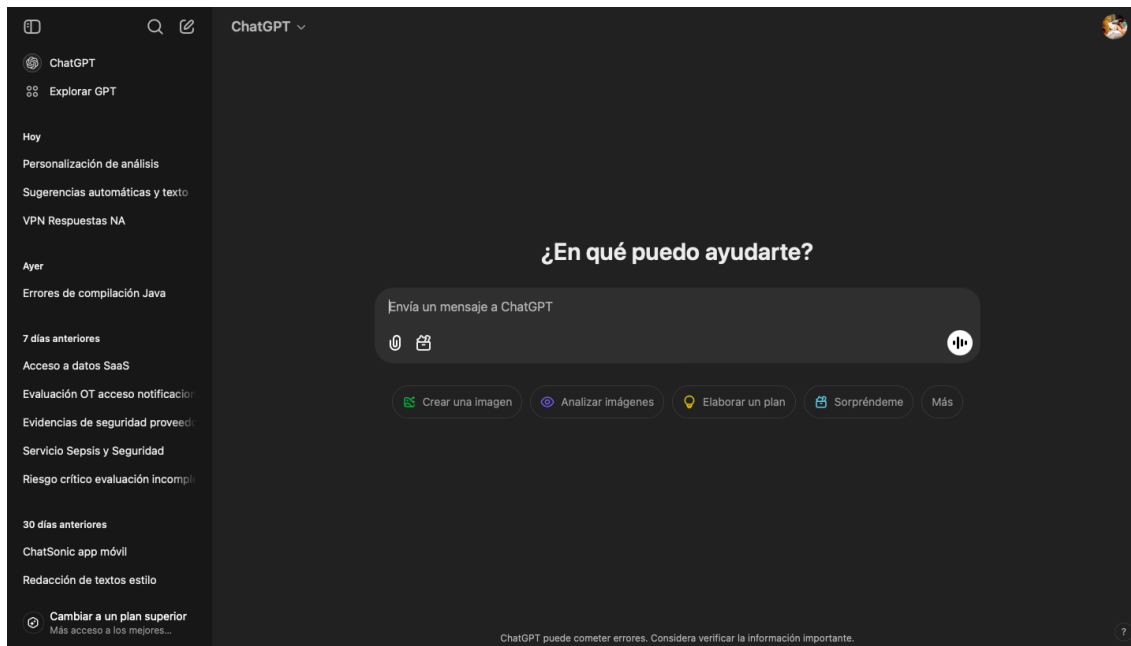


Figura 2.17: Historial de búsqueda en ChatGPT en su versión web

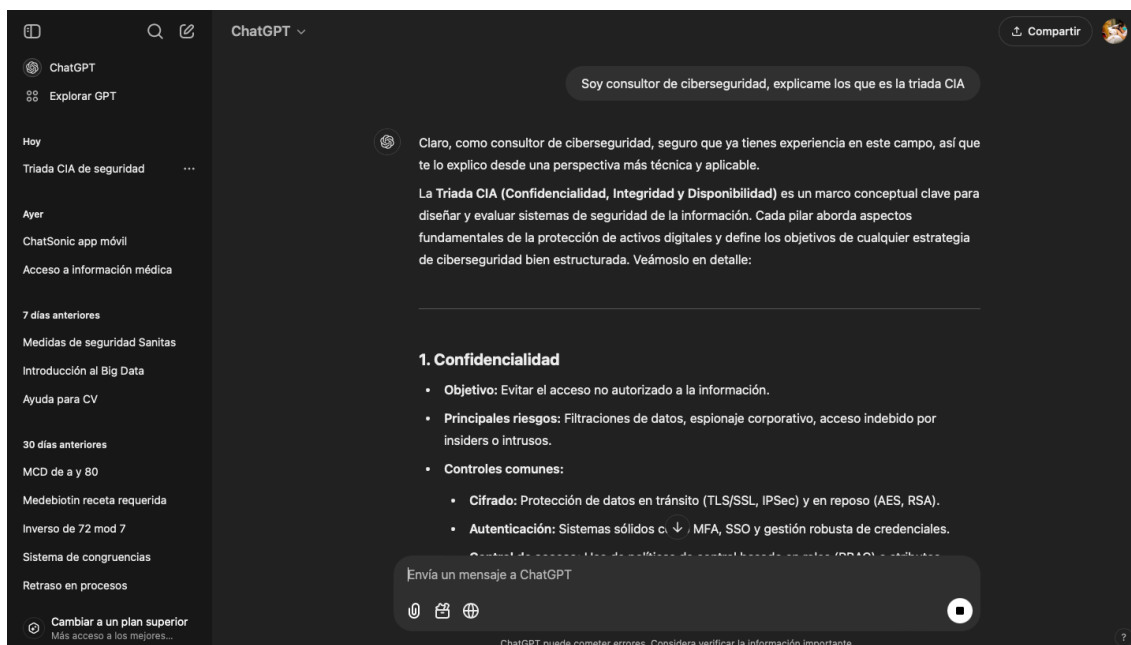


Figura 2.18: Chat personalizado de ChatGPT en su versión web

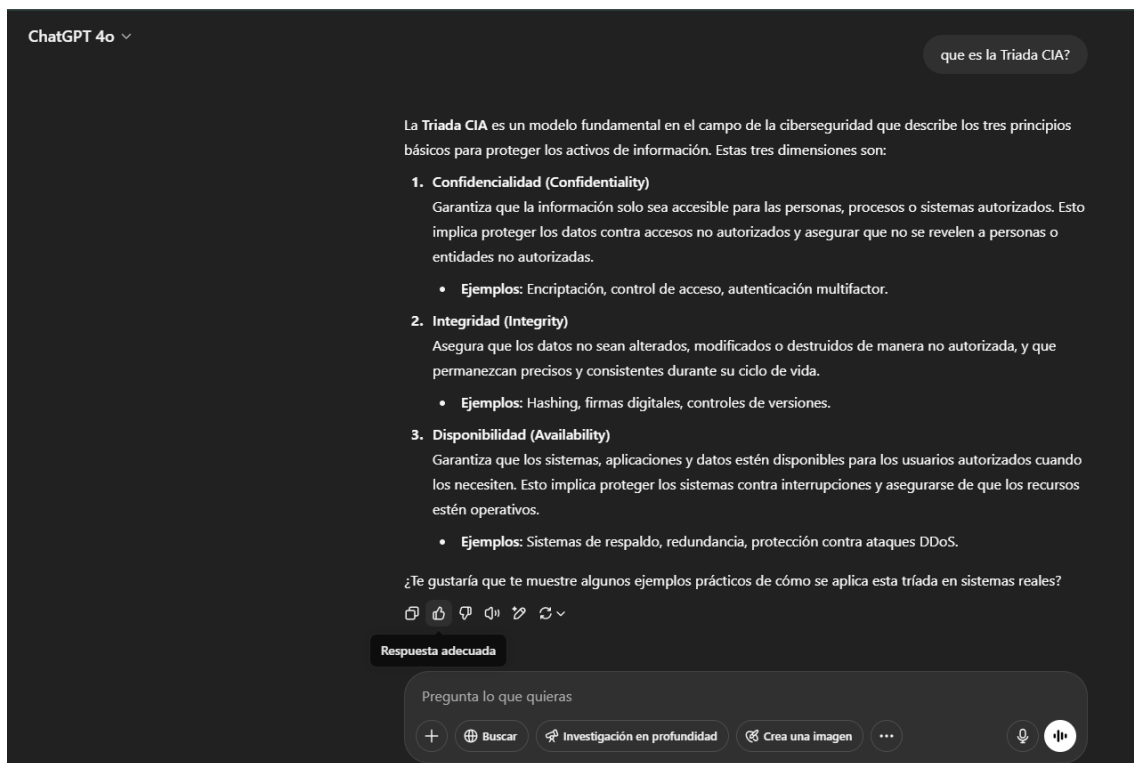


Figura 2.19: Chat en el que se muestran los botones para evaluar la respuesta de ChatGPT en su versión web

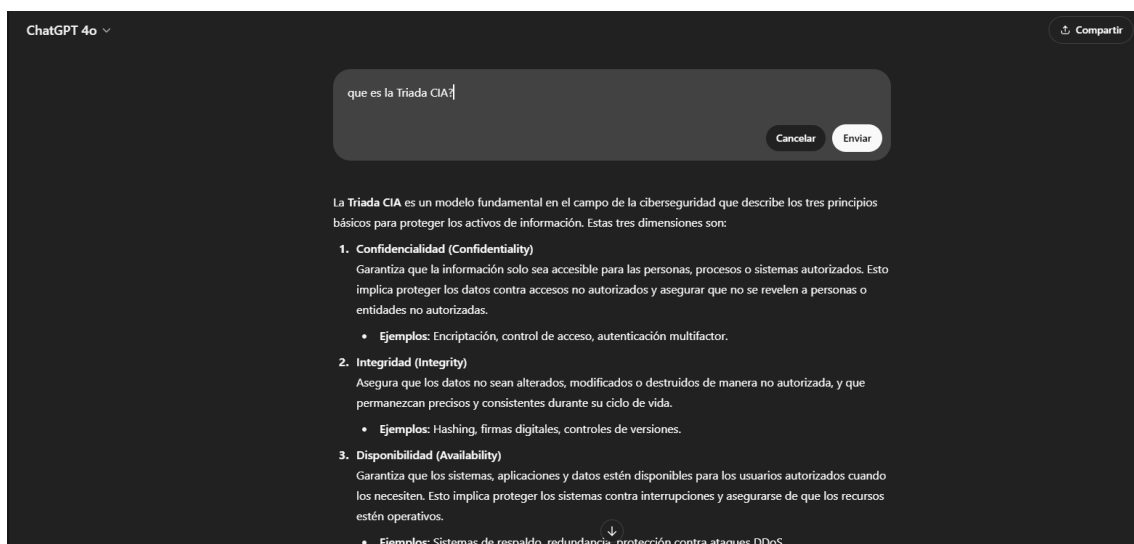


Figura 2.20: Chat en el que se muestran como los usuarios tienen la opción de reenviar o volver a ejecutar una consulta en ChatGPT en su versión web

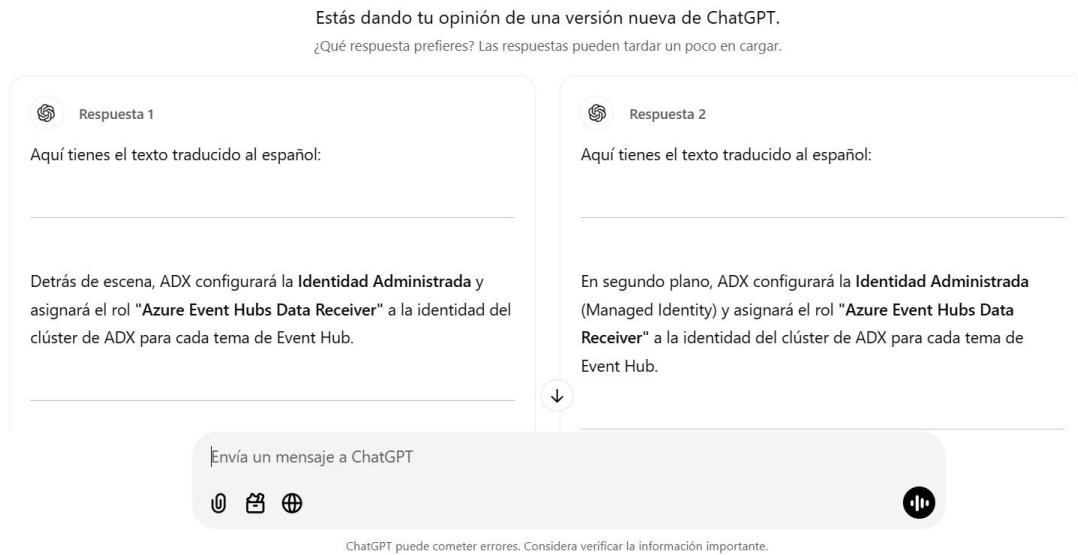


Figura 2.21: Chat en el que se muestran dos opciones de respuesta en ChatGPT en su versión web

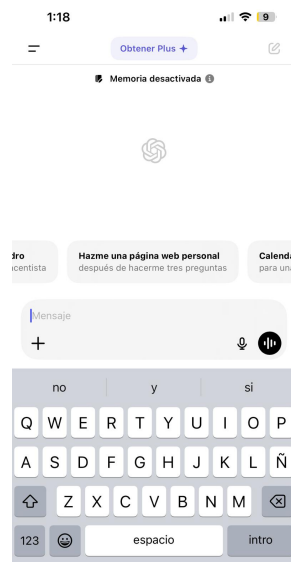


Figura 2.22: Interfaz móvil de ChatGPT al abrir la aplicación.

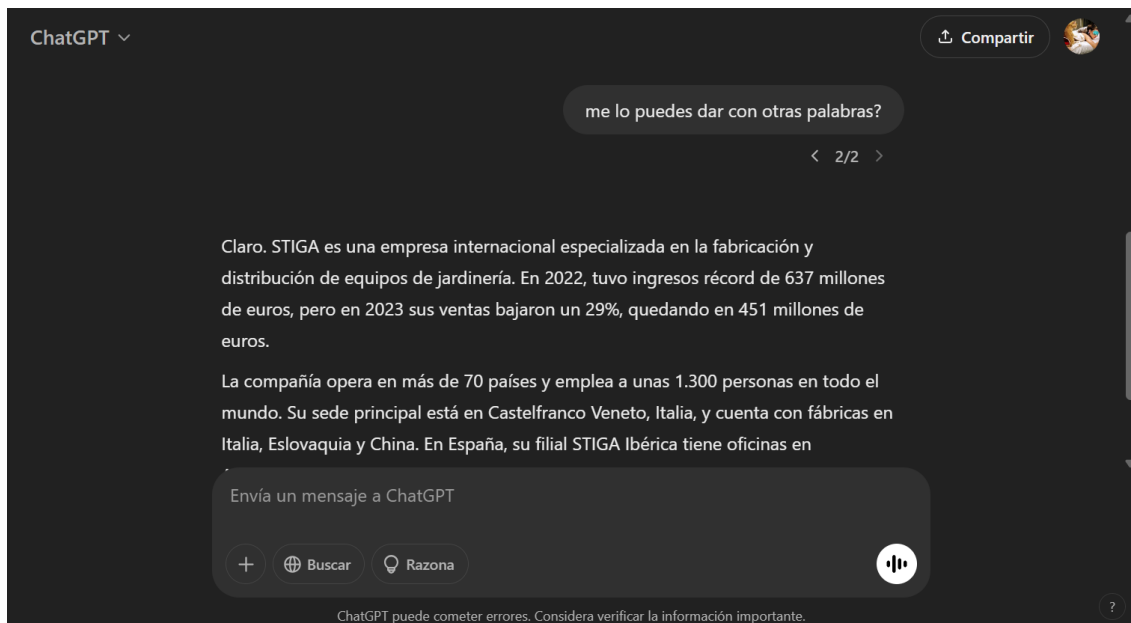


Figura 2.23: Chat en el que se le pide que de una respuesta de otra forma

programación, incluyendo Python y JavaScript, y permite realizar solicitudes HTTP directas a través de peticiones POST. Además, es flexible en la personalización de respuestas mediante “prompt” o instrucciones de contexto, permitiendo ajustar el tono, estilo y formato. También ofrece funciones avanzadas como búsqueda en tiempo real y memoria contextual para mejorar la precisión y relevancia de las respuestas. Los límites de uso de la aplicación varían según el plan de suscripción elegido, con opciones tanto gratuitas como de pago.

Como se observa en la Figura 2.25, los usuarios pueden utilizar un cuadro de texto para escribir sus preguntas o comandos y, además, cuentan con botones que ofrecen sugerencias automáticas. *ChatSonic* admite múltiples idiomas para la generación de texto, lo que permite a los usuarios comunicarse y crear contenido en diversos idiomas. Sin embargo, su interfaz está diseñada principalmente para ser utilizada en inglés. Como se puede apreciar en la Figura 2.26 *ChatSonic* permite cierta personalización de su interfaz, enfocada principalmente en aspectos funcionales como la selección de modos de escritura para adaptar el tono o estilo del contenido (formal, casual, creativo, etc.), la posibilidad de alternar entre texto y comandos de voz para una interacción más flexible, y ajustes de búsqueda en tiempo real gracias a su integración con *Google*, lo que asegura resultados más relevantes. Sin embargo, las opciones para modificar completamente el diseño visual de la interfaz, como colores o disposición de elementos, son limitadas y dependen del nivel de acceso proporcionado por el plan de suscripción elegido. También ofrece la opción de generar respuestas múltiples a una misma consulta, permitiendo a los usuarios elegir la respuesta que mejor se adapte a sus necesidades. Esta funcionalidad es útil cuando se busca obtener diversas perspectivas o soluciones para una pregunta o tarea específica, brindando mayor flexibilidad en la interacción.

*ChatSonic* ofrece funcionalidades como el historial de chats, que permite revisar conversaciones anteriores, como se puede observar en la Figura 2.27, y la opción

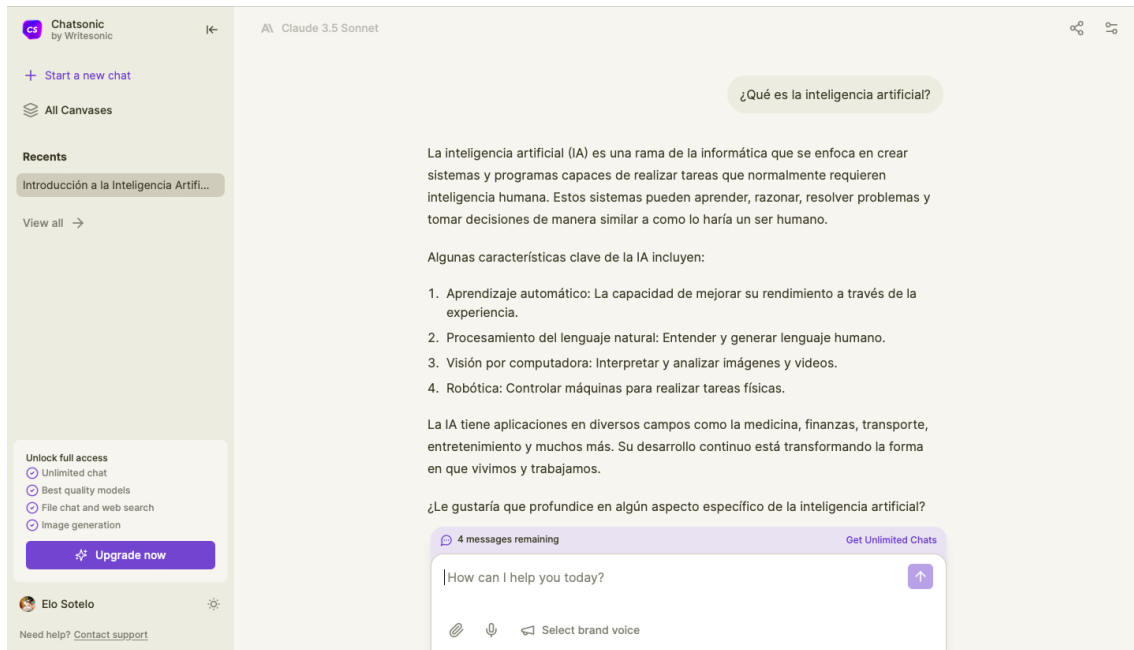


Figura 2.24: Búsqueda en ChatSonic

de subir archivos, ideal para interactuar con documentos directamente en el chat. Además, permite cierta personalización en las interacciones, como ajustar el tono de las respuestas o configurar preferencias para adaptar la experiencia. Sin embargo, no cuenta con características avanzadas como la edición directa de las respuestas generadas ni opciones para modificar dinámicamente secciones específicas del contenido durante la conversación.

### 2.3.3. *LuzIA*

*LuzIA*<sup>41</sup> es una aplicación de inteligencia artificial generativa diseñada para facilitar la creación de contenido textual. Su principal objetivo es proporcionar una experiencia sencilla y accesible para generar textos, mejorar redacciones, crear ideas, resumir información, escribir artículos, entre otras tareas relacionadas con la producción de contenido. Además, la aplicación admite la carga de documentos, lo que permite que *LuzIA* los analice y genere contenido basado en ellos, tales como resúmenes o expansión de ideas. La interacción principal se realiza mediante texto, aunque también es posible interactuar mediante voz, como se observa en la Figura 2.28, permitiendo a los usuarios escribir sus consultas y recibir respuestas automáticas generadas por la IA, como muestra la Figura 2.28.

*LuzIA* está orientada a ofrecer interacción conversacional en múltiples plataformas, pero, al igual que *ChatGPT*, su uso a través de la API permite adaptarse a una variedad de escenarios de desarrollo.

*LuzIA* está disponible únicamente como aplicación móvil, tanto para Android como para iOS, y no cuenta con una API pública. Es principalmente de pago, pero ofrece una versión gratuita con limitaciones en cuanto al número de interacciones y

<sup>41</sup><https://luzia.com>

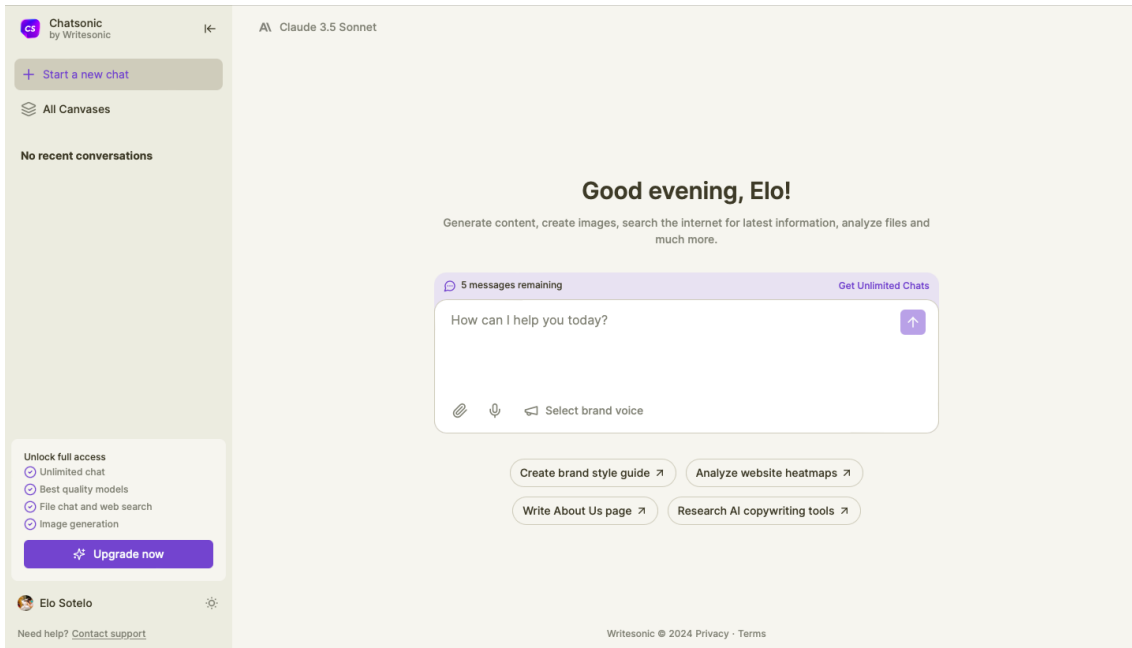


Figura 2.25: Interfaz web de ChatSonic

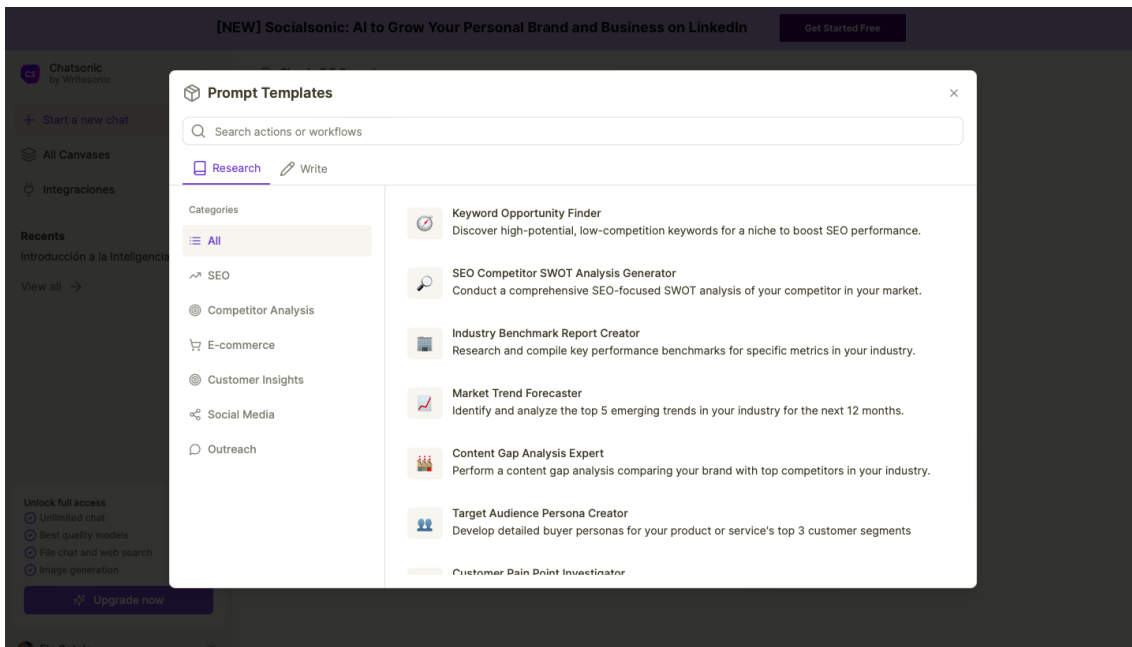


Figura 2.26: Búsqueda en ChatSonic



Figura 2.27: Historial de chats de ChatSonic

la cantidad de texto generado. Para acceder a funcionalidades adicionales, como un mayor número de interacciones, generación de texto más extensa, personalización de respuestas o acceso prioritario en momentos de alta demanda, los usuarios pueden optar por un plan de pago que incluye estas características avanzadas.

*LuzIA* ofrece, en su pantalla de inicio (ver Figura 2.28), la opción de elegir entre diversas personalidades, como Luzia, Profesora o Amiga, entre otras. Esta característica permite que la herramienta adapte sus respuestas según la personalidad seleccionada, brindando una experiencia más personalizada y acorde con la consulta realizada. En la pantalla de inicio, los usuarios también pueden acceder a un apartado de herramientas con las siguientes opciones:

- Matemáticas: Explica cómo resuelve cualquier problema matemático, detallando el proceso paso a paso.
- Lector de documentos: Permite adjuntar un documento para hacer preguntas relacionadas con su contenido, y obtiene resúmenes o resolverá cualquier duda.
- Creación de imágenes: Genera imágenes basadas en las características especificadas por el usuario.
- Reconocimiento de imágenes: Permite adjuntar una imagen para que la analice, la describa, la traduzca o resuelva dudas relacionadas con ella.

*LuzIA* permite acceder al historial de chats, lo que facilita retomar conversaciones anteriores, y ofrece la opción de subir archivos para interactuar con documentos directamente pulsando el icono “+” que se observa a la izquierda de la Figura 2.28. Sin embargo, no incluye funcionalidades como la personalización avanzada de las respuestas o la edición dinámica de secciones específicas del contenido durante la interacción. Aunque puede recordar información compartida previamente en una conversación, su capacidad de memoria está limitada al contexto actual del chat y no retiene datos a largo plazo. Además, *LuzIA* puede proporcionar información en tiempo real, como recomendaciones de restaurantes cercanos o el clima actual, lo que indica que tiene la capacidad de buscar información en internet para responder a las consultas de los usuarios.

### 2.3.4. *Perplexity AI*

*Perplexity AI*<sup>42,43</sup> es una aplicación de inteligencia artificial que utiliza un modelo de lenguaje avanzado que combina generación de texto con búsqueda profunda en internet, proporcionando respuestas rápidas y precisas. Para ello, permite a los usuarios interactuar mediante texto, escribiendo preguntas o solicitudes que la aplicación responde automáticamente, como se puede observar en la Figura 2.29, y ofrece una API para desarrolladores, facilitando su integración en otros sistemas o aplicaciones. La API de *Perplexity AI* es compatible con múltiples lenguajes de programación y permite realizar solicitudes HTTP directas. Los desarrolladores pueden autenticarse

---

<sup>42</sup><https://www.perplexity.ai>

<sup>43</sup><https://www.techtarget.com/searchenterpriseai/definition/Perplexity-AI>

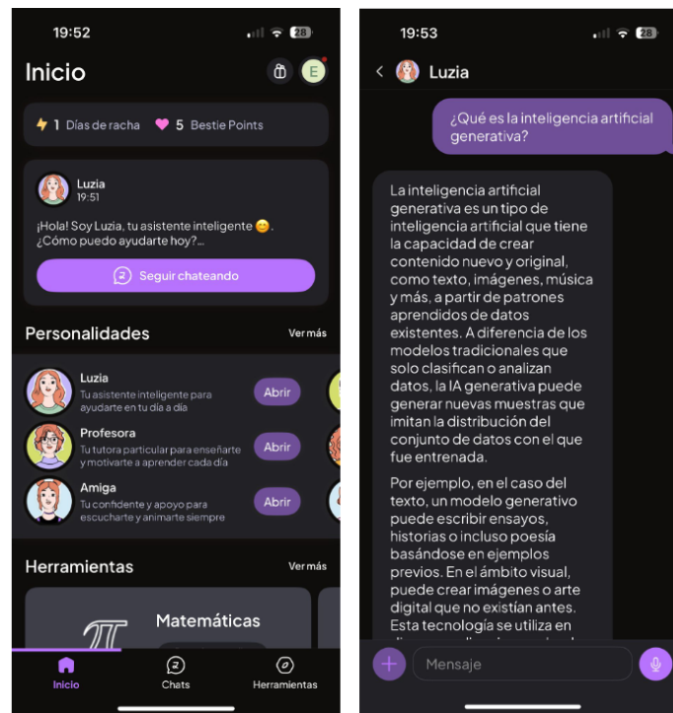


Figura 2.28: Interfaz y búsqueda de LuzIA

generando una clave de API desde la página de configuración de su cuenta y enviándola como un token de portador en el encabezado de autorización de cada solicitud. La API es compatible con clientes de OpenAI, lo que facilita su integración en aplicaciones existentes. Además, permite la personalización de las respuestas mediante “prompts” o instrucciones de contexto, ajustando el tono, estilo y formato según las necesidades del usuario.

*Perplexity AI* está disponible tanto como aplicación web como para móvil y la interfaz es limpia y sencilla, enfocada en ofrecer una experiencia eficiente sin distracciones. Los usuarios pueden introducir consultas a través de un cuadro de texto y obtener respuestas claras y concisas. Además, *Perplexity AI* cuenta con un diseño muy directo y minimalista, ideal para quienes prefieren una interfaz sin elementos innecesarios como se muestra en la Figura 2.30 para la versión web y en la Figura 2.31 para la versión móvil. *Perplexity AI* tiene la capacidad de generar resúmenes claros de grandes volúmenes de texto, lo que es útil para los usuarios que necesitan obtener información de manera rápida sin leer extensos artículos. *Perplexity AI* es una herramienta diseñada para ofrecer respuestas fundamentadas en datos provenientes de fuentes verificadas. Esto significa que, cuando respondes a una consulta, el modelo no solo genera información basada en conocimiento general, sino que también incluye referencias claras y rastreables a las fuentes de donde extrae los datos. Esto es especialmente útil para garantizar la precisión y la fiabilidad de las respuestas, ya que el usuario puede verificar la información presentada.

Además, la plataforma está optimizada para manejar consultas complejas. Esto implica que el usuario puede realizar preguntas más elaboradas, que requieran relacionar datos o profundizar en temas específicos, y obtendrás resultados detallados y



Figura 2.29: Búsqueda en Perplexity AI web

organizados. Este enfoque facilita la investigación rápida, ya que no necesitas buscar en múltiples sitios; *Perplexity AI* concentra y sintetiza la información relevante en una sola respuesta.

La versión móvil de *Perplexity AI* incluye la funcionalidad de interacción por voz, lo que permite a los usuarios realizar consultas habladas de manera rápida y sencilla. Esta característica está diseñada para ofrecer mayor comodidad y accesibilidad, especialmente cuando escribir no es práctico, facilitando así una experiencia más fluida e intuitiva para obtener información detallada directamente a través de comandos de voz.

*Perplexity AI* no tiene una memoria persistente que guarde automáticamente lo que el usuario introduce entre sesiones. Su funcionamiento se basa en procesar las consultas del usuario en tiempo real sin almacenar un historial permanente de interacciones. Esto significa que la información proporcionada por el usuario durante una sesión no se retiene para futuras consultas o personalización a menos que se implemente explícitamente una funcionalidad de cuenta o historial. Además, no permite subir archivos ni editar dinámicamente secciones específicas de las respuestas, limitando la personalización avanzada en las interacciones. Sin embargo, tiene una API, llamada *Sonar*, que permite integrar su inteligencia artificial generativa en aplicaciones personalizadas siendo compatible con clientes de *OpenAI*, lo que facilita su integración en herramientas existentes utilizando lenguajes de programación como Python.



Figura 2.30: Interfaz de Perplexity AI web

### 2.3.5. *Claude*

*Claude*<sup>44</sup> es una aplicación de inteligencia artificial que utiliza un modelo de lenguaje creado por *Anthropic* para responder preguntas y ayudar en la creación de textos y generación de contenido. Además, *Claude* permite la carga de un archivo o imagen y ofrece preguntas predeterminadas. También está disponible tanto como aplicación web como aplicación móvil y cuenta con una API integrable para facilitar su uso en otras plataformas.

A diferencia de otras aplicaciones de IA, *Anthropic* implementó en *Claude* un marco denominado “IA Constitucional” basado en una “constitución” de directrices éticas que guían el comportamiento del modelo, asegurando que sus respuestas estén alineadas con valores humanos fundamentales y evitando la generación de contenido perjudicial o sesgado<sup>45</sup>.

Además de ofrecer una interfaz simple, *Claude* permite al usuario personalizar el estilo de las respuestas según sus preferencias, como se muestra en la Figura 2.32. El usuario tiene la opción de seleccionar entre estilos de escritura predefinidos o incluso crear uno propio, ya sea analizando un archivo o escribiendo un párrafo para que *Claude* lo analice y ajuste el estilo en consecuencia, como se ilustra en la Figura 2.33.

Otra funcionalidad destacable que tiene *Claude* es que permite el cambio del estilo de la fuente del texto para facilitar su uso para las personas con dislexia, como se aprecia en la Figura 2.34. Se observa un ejemplo de *Claude*, en la Figura 2.35, donde se aplica la fuente para personas disléxicas comparado con como suele ser el estilo de fuente predeterminado visto en la Figura 2.35. *Claude* también permite

<sup>44</sup><https://www.anthropic.com/claude>

<sup>45</sup><https://bitscloud.com/claude-ai-todo-lo-que-necesitas-saber/>



Figura 2.31: Interfaz de Perplexity AI móvil

almacenar y recordar el historial de chats, esto permite a los usuarios mantener una “memoria” de conversaciones previas, lo que ayuda a generar respuestas más coherentes y personalizadas basadas en interacciones pasadas. Además, los usuarios podrían configurar chats personalizados, adaptando el comportamiento del modelo a sus necesidades específicas, lo que proporciona una experiencia más a medida.

*Claude* no solo se limita a la generación de texto a partir de las entradas proporcionadas por los usuarios, sino que también tiene la capacidad de realizar búsquedas en internet. Esto le permite obtener información en tiempo real, mejorando la calidad de sus respuestas al acceder a fuentes externas de manera dinámica. Además de textos, *Claude* permite la carga de archivos, incluyendo documentos en diversos formatos, lo que facilita el análisis y la personalización de las respuestas basadas en el contenido de esos archivos. También puede manejar imágenes, lo que amplía aún más su utilidad en tareas que involucran contenido visual, como la interpretación de gráficos o imágenes complejas.

La única funcionalidad diferente en la versión móvil de la versión web de *Claude* es la opción de realizar consultas por voz, representada por el icono del micrófono en la barra de texto, como se observa en la Figura 2.36.

*Claude* ofrece una API integrable que facilita la incorporación de sus capacidades de inteligencia artificial en diversas plataformas. Esta API es compatible con varios lenguajes de programación, incluyendo Python y TypeScript, y permite hacer solicitudes HTTP directas. Además, es flexible, permitiendo la personalización de las respuestas mediante el uso de “prompts” o instrucciones de contexto, lo que permite ajustar el tono, estilo o formato de las respuestas. Esto es útil en aplicaciones como *chatbots*, generación de contenido y herramientas interactivas.

También, *Claude* posee una capacidad avanzada para mantener el contexto de las conversaciones gracias a su amplia ventana de contexto de hasta 100,000 *tokens*, lo que le permite procesar y recordar información dentro de una sesión de chat de manera efectiva.

### 2.3.6. *YouChat*

*YouChat*<sup>46</sup> es la interfaz de chat de *You.com*, un motor de búsqueda que integra diferentes modelos de inteligencia artificial, como *Creative*, *Reacher*, *Gemini* o *GPT-4*, para ofrecer una experiencia de usuario más rica y adaptada a diversas necesidades. Cada modelo tiene características y capacidades únicas que permiten abordar distintos tipos de tareas y consultas de manera más efectivas. La interfaz de *YouChat*, como se observa en la Figura 2.37, presenta un diseño minimalista y oscuro que prioriza la facilidad de uso. Además, tal y como se observa en dicha Figura, se puede seleccionar entre los modelos mencionados anteriormente, lo que ofrece flexibilidad en la manera de interactuar con la inteligencia artificial. La pantalla principal ofrece accesos directos a las principales funcionalidades organizadas por categorías como Marketing, Ventas, Ingeniería, Producto, Análisis de Datos y Finanzas lo que permite que los usuarios seleccionen rápidamente el enfoque que necesitan según el contexto. Los botones destacados en la interfaz incluyen:

---

<sup>46</sup><https://you.com/youchat>

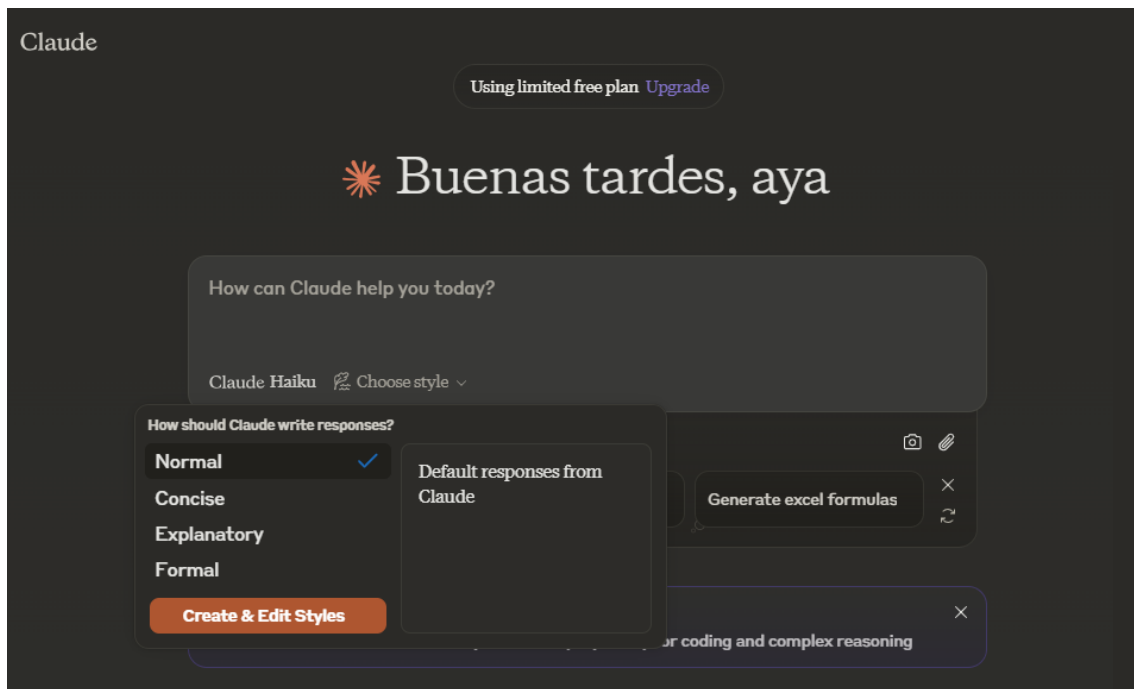


Figura 2.32: Interfaz de *Claude* con la opción de elegir cómo te responde

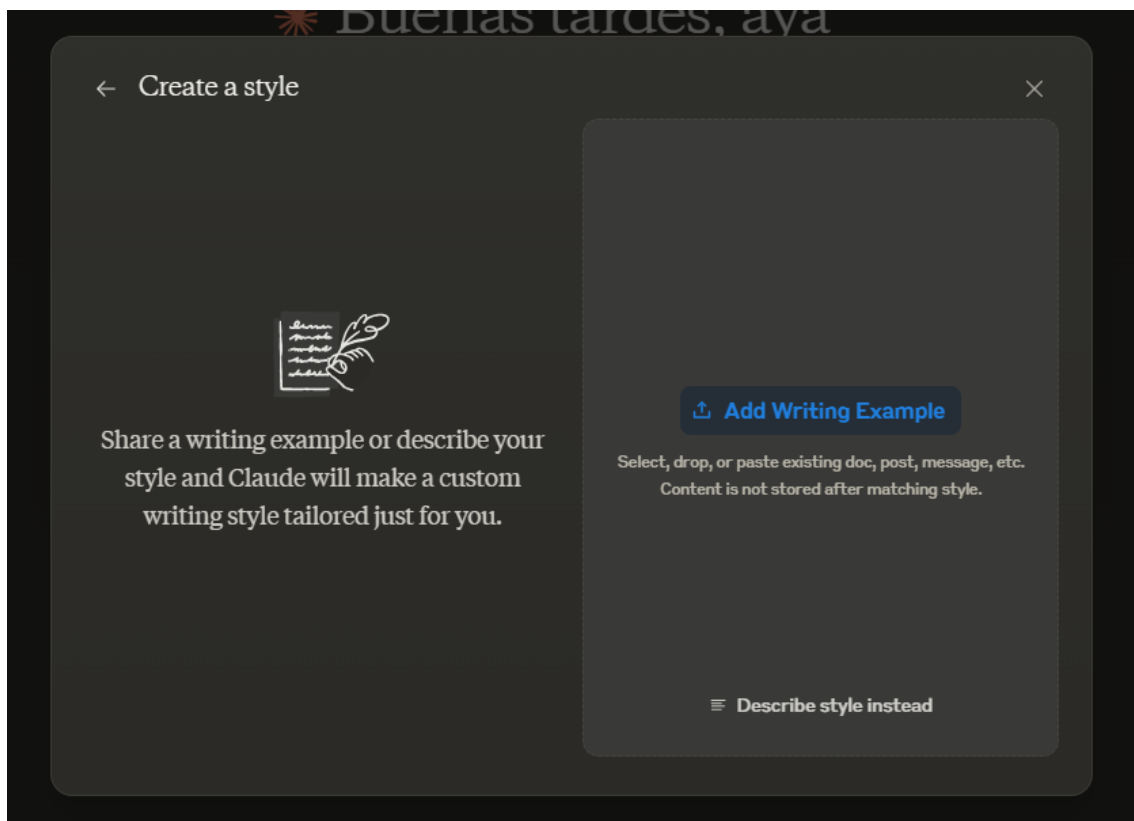


Figura 2.33: Opción de crear estilos de respuesta en *Claude*

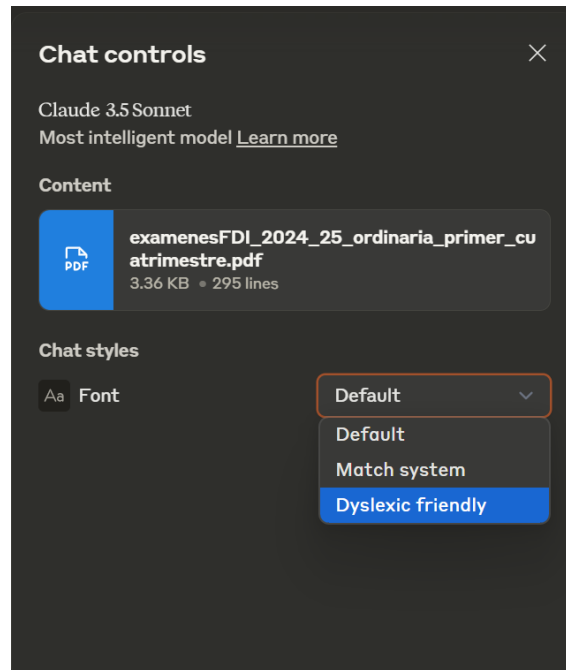


Figura 2.34: Diferentes tipos de fuentes en *Claude*

- Research: Para consultas investigativas profundas.
- Creative: Enfocado en tareas de creación visual o textual.
- Genius: Asistente para resolver problemas complejos paso a paso.
- Build Your Own: Herramienta para crear un agente personalizado.

Se observa en la Figura 2.38 que la sección de consultas textuales y el tamaño de texto permiten a los usuarios redactar preguntas largas o detalladas sin sentirse limitados. Este diseño facilita la redacción de consultas extensas. Además, *YouChat* organiza los resultados de manera clara y estructurada, incluyendo enlaces y recomendaciones adicionales. Por ejemplo, ante la consulta “¿Qué hora es en España?”, no solo proporciona la respuesta, sino que también explica los cambios de horario y las zonas horarias de forma educativa, como se muestra en la Figura 2.38. En cuanto al almacenamiento de conversaciones, la información disponible actualmente no especifica si *YouChat* guarda el historial de chats entre sesiones de forma persistente.

*YouChat* está disponible tanto en la versión web como en una aplicación móvil. La principal diferencia de la versión móvil es que aprovecha las funciones específicas del dispositivo para mejorar la experiencia del usuario. Por ejemplo, gracias al GPS, puede ofrecer recomendaciones locales y datos geográficos precisos basados en la ubicación del usuario. Además, la integración con la cámara permite subir imágenes, escanear documentos o leer códigos QR, lo que amplía las formas de interactuar con la inteligencia artificial de manera más práctica y versátil. Por último, la aplicación móvil incluye notificaciones push, que facilitan la comunicación en tiempo real a través de recordatorios, actualizaciones y alertas personalizadas.

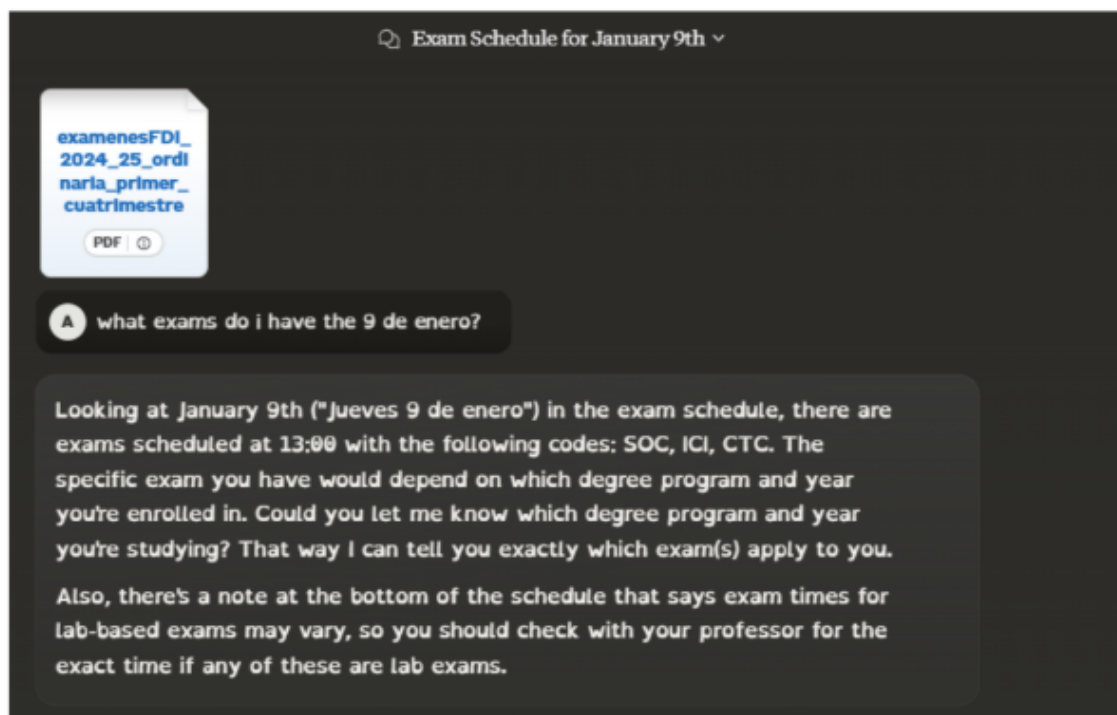
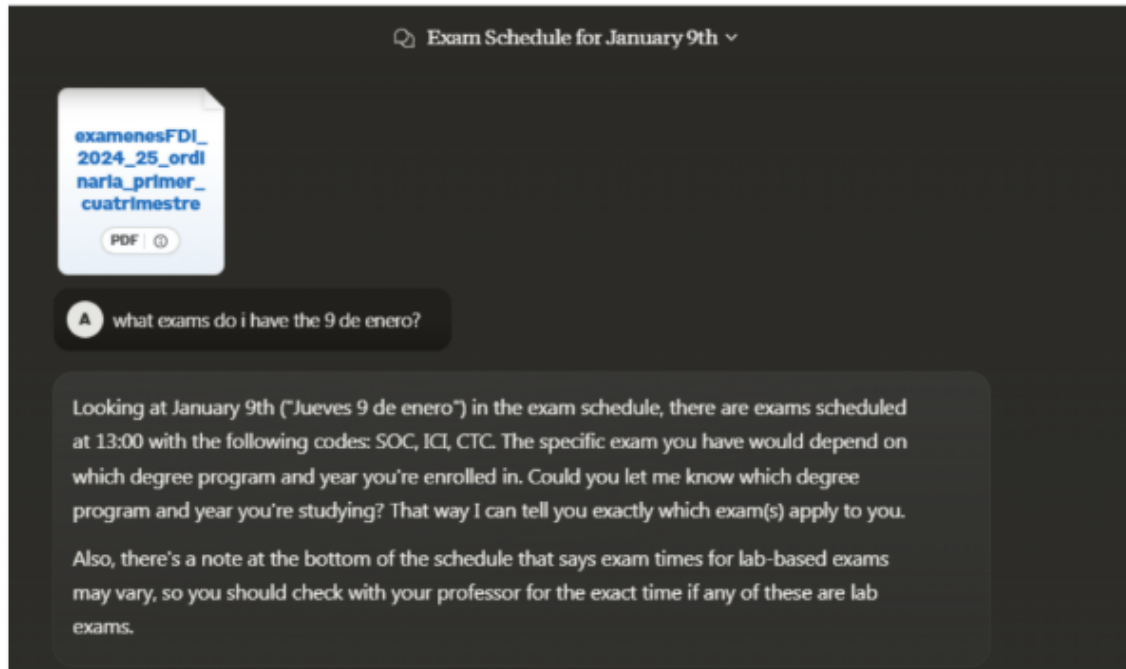


Figura 2.35: Comparación de Claude con diferentes estilos de fuente

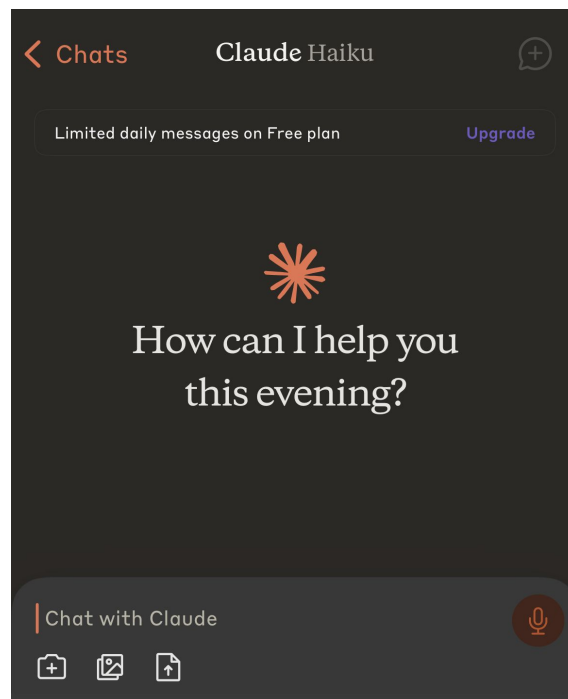


Figura 2.36: Interfaz móvil de *Claude* con funcionalidad de consultas por voz

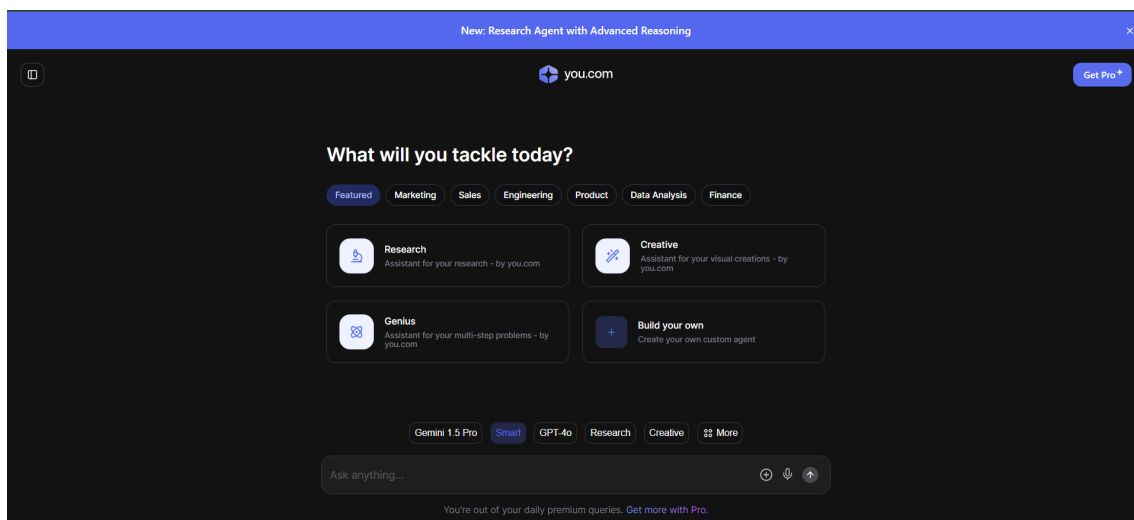


Figura 2.37: Interfaz principal junto con los botones predeterminados de YouChat

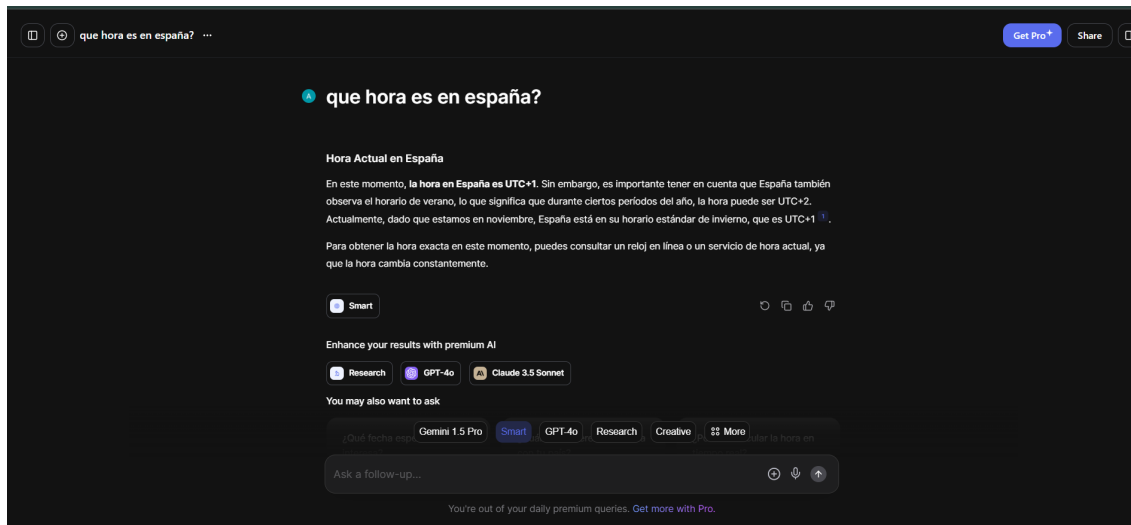


Figura 2.38: Ejemplo de uso de YouChat

### 2.3.7. Gemini (Google)

*Gemini*<sup>47</sup> es una aplicación que utiliza un modelo de inteligencia artificial desarrollado por *Google DeepMind*. Su principal objetivo es proporcionar respuestas precisas, relevantes y enriquecidas con datos obtenidos directamente de los servicios de *Google*, como el propio navegador, *Google Workspace*, *Google Maps* y *YouTube*, lo que lo hace ideal para usuarios que usan dichas plataformas.

*Gemini* posee una interfaz muy sencilla que incluye un campo de texto donde se pueden escribir preguntas o indicaciones detalladas, junto a iconos correspondientes para subir una imagen o grabar una nota de voz, como se ve en la Figura 2.39. La interacción con *Gemini* es muy versátil ya que los usuarios pueden comunicarse a través de texto, y tienen la posibilidad de grabar audios o subir imágenes, para que la IA los analice y brinde respuestas personalizadas. Además, presenta la opción de poder oír la respuesta que proporciona *Gemini* pulsando el icono que se encuentra encima del texto generado, a la derecha, como se observa en el ejemplo de uso en la Figura 2.40.

También cabe a destacar que *Gemini* está disponible tanto en aplicaciones móvil como web y es gratuita, aunque ofrece un plan de pago más avanzado con funcionalidades exclusivas diseñadas para aprovechar el *hardware* y las características específicas del dispositivo. Entre estas funcionalidades exclusivas se encuentra el análisis directo de imágenes y contenido local, que ofrece la posibilidad de capturar fotos en tiempo real o seleccionar archivos directamente desde la galería. Además, la aplicación móvil ofrece respuestas personalizadas basadas en la ubicación del usuario mediante la integración con el GPS del dispositivo. También incluye un soporte para uso *offline* básico, que permite descargar respuestas en formato de audio y realizar análisis limitados sin conexión a internet. Por último, dispone de notificaciones inteligentes, que envían alertas personalizadas a través del sistema de notificaciones

<sup>47</sup><https://www.xataka.com/basics/google-gemini-que-como-funciona-diferencias-gpt-cuando-podras-usar-este-modelo-inteligencia-artificial>

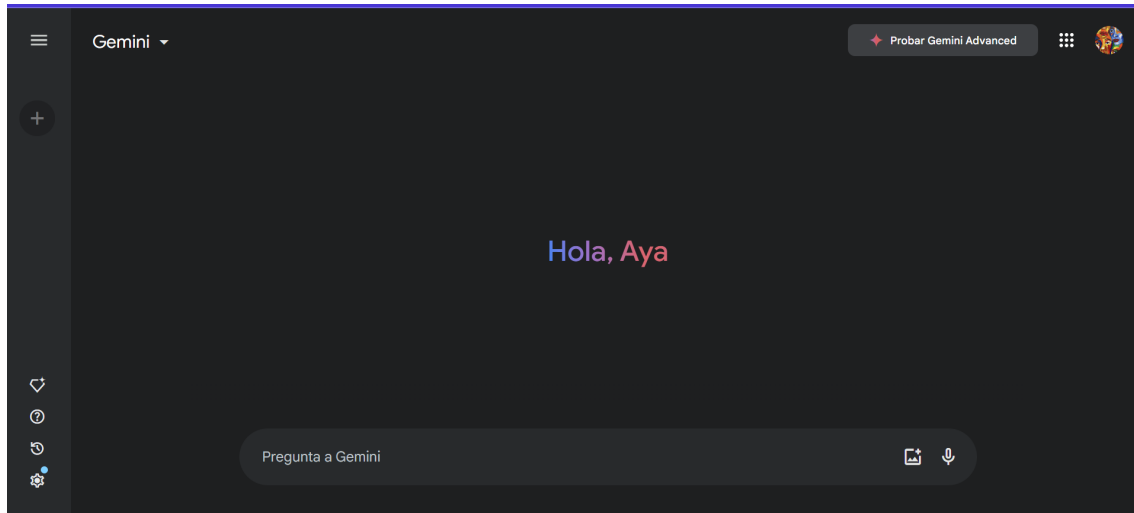


Figura 2.39: Interfaz de Gemini

del dispositivo móvil.

Por otro lado, como se puede observar en la Figura 2.41, la aplicación web de *Gemini* está optimizada para realizar análisis más complejos gracias a su capacidad de aprovechar los recursos en la nube. También ofrece una experiencia visual mejorada para tareas que requieren la visualización de gráficos avanzados, tablas de gran tamaño o mapas interactivos y permite una mayor integración con herramientas empresariales, como sistemas *Customer Relationship Management* (CRM), *Enterprise Resource Planning* (ERP) u otras aplicaciones específicas. También destaca por su capacidad para manejar múltiples sesiones simultáneamente, lo que la hace ideal para un entorno profesional o empresarial.

*Gemini* no almacena un historial de chats de forma persistente entre sesiones, lo que significa que no retiene las conversaciones de un usuario para futuras interacciones. Sin embargo, si almacena de forma predeterminada las conversaciones de los usuarios en sus cuentas de *Google*. Esta actividad se guarda para mejorar y desarrollar la IA, y puede ser revisada por moderadores humanos. Por defecto, las conversaciones se conservan durante 18 meses, pero los usuarios tienen la opción de gestionar o eliminar este historial accediendo a “Mi Actividad de *Google*”. Además, es posible desactivar la actividad en las aplicaciones de *Gemini* para que las futuras conversaciones no se almacenen. Sin embargo, incluso con esta opción desactivada, las conversaciones se retienen hasta 72 horas para permitir que *Google* proporcione el servicio y procese comentarios.

### 2.3.8. Conclusiones

En la Tabla 2.2 se muestra un resumen de todo lo visto en esta sección. En términos de interfaz, las herramientas comparadas comparten ciertos elementos comunes, como un cuadro de texto donde el usuario puede introducir su consulta. No obstante, algunas incorporan funciones adicionales, como la posibilidad de subir archivos, interactuar mediante consulta por voz, o incluso escuchar las respuestas mediante sistemas de *Text-to-Speech* (TTS), como es el caso de *ChatSonic*, *LuzIA* y *Gemini*.

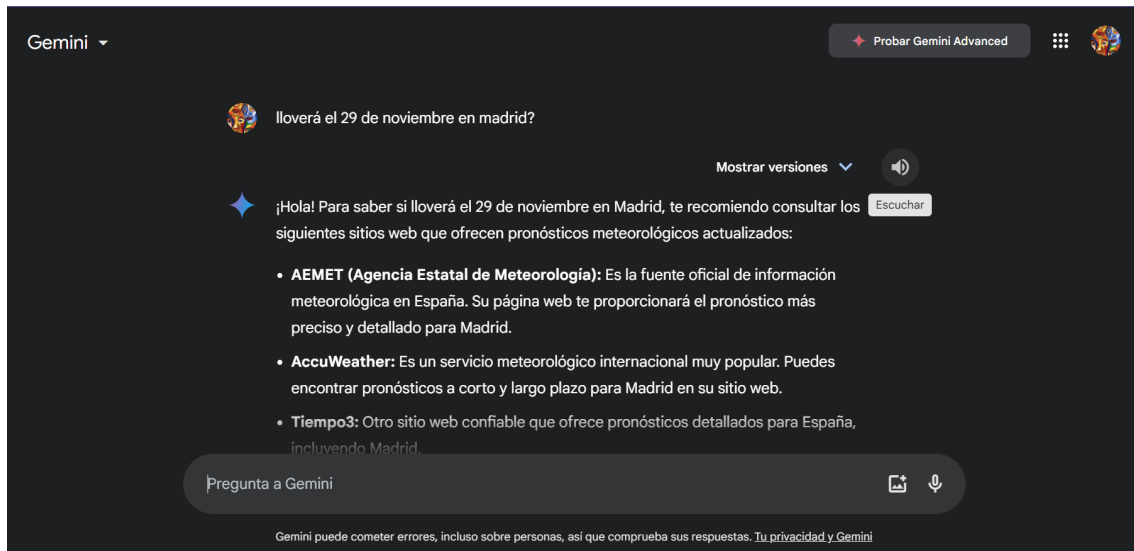


Figura 2.40: Ejemplo de Gemini

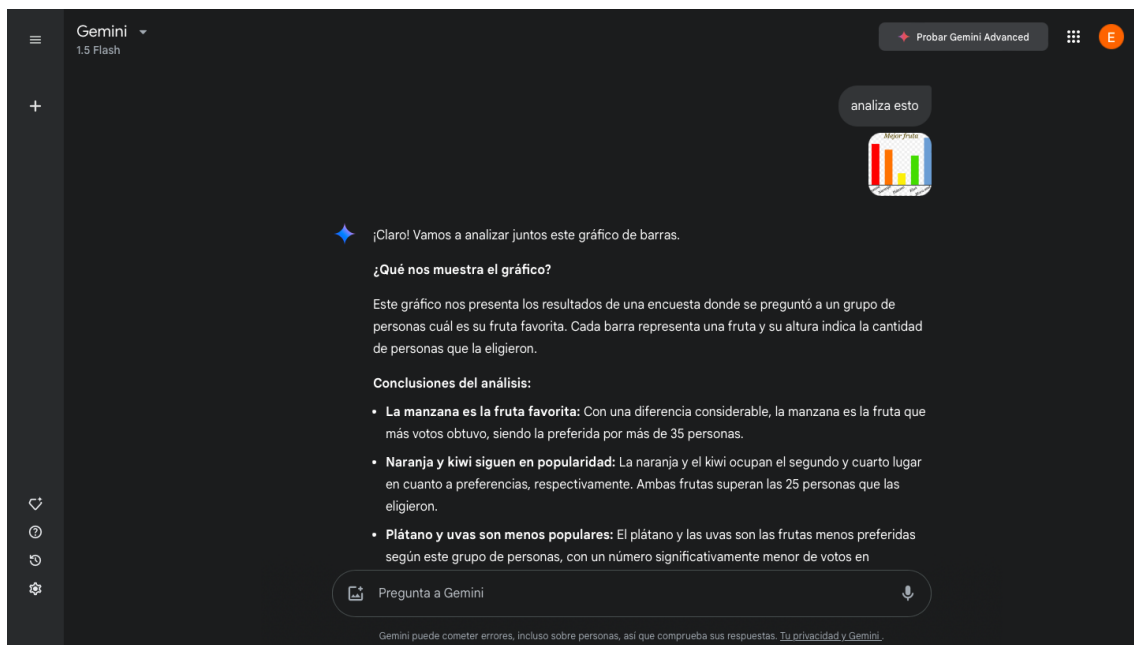


Figura 2.41: Ejemplo de Gemini analizando una gráfica

Estas funcionalidades no solo mejoran la accesibilidad, sino que también ofrecen una experiencia más interactiva y adaptada a las necesidades de los usuarios.

En cuanto a la presentación de las respuestas, todas las herramientas las muestran en formato de texto, aunque con algunas variaciones en el estilo y la interacción, como el uso de enlaces, negritas o recursos adicionales en las respuestas. Herramientas como *Perplexity AI*, *ChatSonic* y *Gemini* se destacan por incluir referencias o enlaces a fuentes externas en sus respuestas, lo que las hace especialmente útiles para quienes necesitan verificar la información proporcionada.

A pesar de estas similitudes, existen diferencias importantes entre las herramientas. *ChatGPT*, *ChatSonic* y *Gemini* se destacan por su flexibilidad al permitir no solo consultas en texto, sino también por voz y mediante la carga de archivos. Esta versatilidad les otorga una ventaja significativa, ya que permiten a los usuarios elegir la forma de interacción que les resulte más cómoda. En cambio, *YouChat* y *Perplexity AI* se limitan a consultas en formato de texto, lo que puede ser una limitación para aquellos usuarios que prefieren o necesitan otras formas de interacción, lo que les confiere una funcionalidad más simple y restringida.

En cuanto a las capacidades adicionales, *ChatSonic* y *Gemini* sobresalen al ofrecer funcionalidades como la generación de imágenes directamente desde las consultas. Asimismo, herramientas como *ChatGPT*, *ChatSonic*, *Claude* y *Gemini* son capaces de generar tablas o resúmenes estructurados, lo que incrementa significativamente su utilidad en tareas relacionadas con la organización y análisis de información.

En cuanto a la accesibilidad, *Claude* destaca por la inclusión de una fuente especialmente diseñada para personas con dislexia, lo que representa un avance importante hacia la inclusión digital. Sin embargo, la mayoría de las herramientas todavía carecen de opciones avanzadas para usuarios con discapacidades, lo que indica que hay margen de mejora en cuanto a la accesibilidad general de estas plataformas.

Por último, en términos de personalización y memoria contextual, *Gemini* sobresale gracias a su integración con los servicios de *Google*, lo que le permite ofrecer recomendaciones personalizadas basadas en el historial del usuario, proporcionando una experiencia más adaptada a las necesidades de cada persona. Además, herramientas como *ChatSonic* y *Perplexity AI* permiten personalizar el tono o el enfoque de las respuestas, mientras que *Gemini* destaca por ofrecer memoria tanto durante como entre sesiones, lo que mejora la continuidad de las interacciones con el usuario.

En resumen, todas las herramientas presentadas son soluciones competentes en el campo de la IA Generativa, pero la elección de la más adecuada dependerá de las necesidades del usuario. Quienes busquen interacciones más complejas y personalizadas podrían optar por *Gemini* o *ChatGPT*, mientras que *Perplexity AI* o *YouChat* serán más apropiadas para usuarios interesados en consultas más simples.

Aplicación	ChatGPT	ChatSonic	LuzIA	Perplexity AI	Claude	YouChat	Gemini
Modelo	GPT-3.5 / GPT-4	GPT-4	GPT-3 / GPT-4 (derivado)	Modelo interno derivado de GPT	Claude 3.5	Gemini 1.5 Pro / GPT-4o	Gemini (Google)
Introducción Consultas	Texto, voz, fichero	Texto, voz, fichero	Texto, voz	Texto	Texto, fichero	Texto, voz, fichero	Texto, voz, fichero
Funcionalidad de Accesibilidad	No tiene. No tiene integrado un sistema de TTS	Adapta el tono o estilo del contenido. Incluye una función de TTS para leer respuestas generadas	Incluye una función de TTS para leer respuestas generadas	No tiene integrado un sistema de TTS	Fuente para disléxicos, Selección del estilo de respuesta. No tiene integrado un sistema de TTS	No tiene integrado un sistema de TTS	Tiene TTS y utiliza Google TTS para convertir las respuestas a voz.
Edición de Consultas	Sí	Sí	No	No	Sí	No	No
Plataformas	Web, móvil, API	Web, móvil, API	WhatsApp, Telegram	Web, móvil	Web, API	Web	Web, móvil, API
Precio	Gratuito / Pago	Pago, prueba gratuita	Gratuito	Gratuito / Pago	Gratuito / Pago	Gratuito	Gratuito
Memoria Contextual	Durante sesión	Durante sesión	Durante interacción	Durante sesión	Durante sesión	Durante sesión	Durante y entre sesiones
Personalización	Moderada: Custom GPTs permiten ajustes	Básica: Perfiles específicos	Muy limitada	Limitada	Limitada	Limitada	Alta: Integración con servicios Google
Historial de Chats	Sí	Sí	No	Sí	Sí	Sí	No
Funciones Predeterminadas	Crear resúmenes, tablas e imágenes en la opción Dall-E	Crear resúmenes, tablas e imágenes	Crear imágenes, admite carga de documentos	Crear resúmenes	Crear resúmenes y tablas	Crear resúmenes	Crear resúmenes, tablas e imágenes
Referencia a Fuentes	No, se lo tienes que pedir	Sí	No	Sí	No, se lo tienes que pedir	No	Sí

Tabla 2.2: Comparativa de aplicaciones IA generativa

## Captura de requisitos

La captura de requisitos del Trabajo de Fin de Grado se llevó a cabo en tres fases, cada una enfocada en un aspecto clave del proceso:

1. Análisis de accesibilidad de ChatGPT: En primer lugar, se realizó un análisis de accesibilidad de *ChatGPT*, una de las plataformas más utilizadas en el campo de la IA generativa. Este análisis se llevó a cabo tanto de forma automática, utilizando las herramientas *WAVE* y *Pa11y Dashboard*, como de forma manual, siguiendo las pautas *COGA*. El objetivo fue identificar las principales deficiencias de la plataforma en términos de accesibilidad cognitiva. Este análisis se presenta en la Sección 3.1.
2. Observación directa con usuarios: Luego, se realizó una observación directa de usuarios con discapacidad cognitiva para comprender mejor sus necesidades y las dificultades específicas que enfrentan al interactuar con *ChatGPT*. Esta fase fue crucial para ajustar los requisitos del prototipo y garantizar que las soluciones propuestas sean realmente efectivas para este grupo de usuarios. Los detalles de esta fase se presentan en la Sección 3.2.
3. Definición de requisitos: Finalmente, se definieron los requisitos del prototipo accesible, basándose en los resultados del análisis de accesibilidad y las observaciones directas con los usuarios. Esta fase se centró en identificar las necesidades específicas de los usuarios con discapacidad cognitiva y traducirlas en requisitos funcionales y no funcionales claros que guiaran el diseño del sistema. Esta fase se describe en la Sección 3.3.

### 3.1. Análisis de la accesibilidad en *ChatGPT*

La accesibilidad es un principio clave en el diseño de cualquier tecnología. Sin embargo, a lo largo de la historia, este aspecto no siempre ha sido una prioridad en su desarrollo, lo que ha llevado a la exclusión de usuarios con discapacidades, particularmente aquellos que enfrentan discapacidades cognitivas.

Dada la creciente adopción de aplicaciones de IA generativa en diversos ámbitos, resulta imprescindible garantizar que estas sean inclusivas y accesibles. Evaluar la

accesibilidad de las plataformas existentes, como *ChatGPT*, es un paso necesario para asegurar que estas tecnologías sean verdaderamente inclusivas, permitiendo a los desarrolladores identificar barreras específicas que puedan dificultar su uso para ciertos grupos de usuarios. Este tipo de análisis no solo visibiliza las limitaciones actuales de estas herramientas, sino que también proporciona información clave para orientar el desarrollo hacia soluciones más equitativas y adaptadas a las necesidades de todos los usuarios, incluyendo aquellos con discapacidades cognitivas.

En este contexto, se ha decidido centrar el análisis en *ChatGPT* no solo debido a su popularidad y amplia aplicación en diferentes sectores, sino también porque representa un referente dentro de las herramientas de IA generativa, gracias a su capacidad de procesar lenguaje natural de forma avanzada. Analizar su accesibilidad nos permitirá identificar de manera clara las barreras que deben superarse para que herramientas similares sean verdaderamente inclusivas. Para ello, se realizó primero una evaluación automática con las herramientas *Pa11y Dashboard* y *WAVE*. En esta primera evaluación la mayoría de los errores identificados estaban relacionados principalmente con la discapacidad visual. Debido a esta limitación, se consideró necesario complementar el análisis con una evaluación manual basada en las pautas COGA, dado que estas se centran específicamente en la accesibilidad para personas con discapacidad cognitiva.

### 3.1.1. Análisis automático

Para evaluar de forma automática la accesibilidad de *ChatGPT*, se utilizaron dos herramientas: *WAVE* y *Pa11y Dashboard*. Ambas herramientas, tal y como se ha mostrado en la Sección 2.2.2, son gratuitas y fáciles de usar. Además de analizar los estándares WCAG 2.1, *WAVE* también analiza si se cumplen las pautas ARIA y ofrece análisis visuales en el navegador, destacando problemas de accesibilidad con iconos claros, y es útil para la corrección inmediata. Por otro lado, *Pa11y Dashboard* permite analizar múltiples páginas a la vez, realizar seguimientos históricos y personalizar las evaluaciones según los requisitos de WCAG 2.1, lo que resulta especialmente útil para proyectos más complejos o de gran escala. A diferencia de *WAVE*, que se centra en el análisis página por página y ofrece resultados inmediatos para correcciones rápidas, *Pa11y Dashboard* ofrece una perspectiva más completa y organizada, lo que permite detectar problemas de accesibilidad comunes en todo un sitio web y realizar un seguimiento del avance en las mejoras realizadas con el tiempo.

El análisis de *ChatGPT* con *WAVE*, que se realizó con fecha 11 de diciembre de 2024, sacó a la luz los siguientes errores:

- Falta de etiquetas de formulario. Se ha identificado que algunos campos de formulario no cuentan con una etiqueta asociada, lo que dificulta que los usuarios, especialmente aquellos que utilizan lectores de pantalla, comprendan el propósito de estos campos. Es fundamental añadir etiquetas adecuadas a todos los campos de los formularios para garantizar que los usuarios puedan interactuar correctamente con ellos.
- El idioma principal de la página no se indica o es inválido. Este error impide

que los lectores de pantalla puedan pronunciar el contenido correctamente. El atributo `lang` debe ser declarado correctamente para que el sistema de lectura sepa en qué idioma se encuentra el contenido, lo que mejoraría significativamente la accesibilidad para usuarios de tecnologías de asistencia.

- Se han detectado 30 botones que no tienen texto visible ni atributos ARIA que describan su función. Esto impide que los usuarios de lectores de pantalla comprendan el propósito de los botones y, por lo tanto, dificulta la interacción con la página. Es necesario agregar texto descriptivo o atributos ARIA para garantizar que los botones sean funcionales para todos los usuarios.
- Contraste insuficiente entre el texto y el fondo, lo que dificulta la legibilidad del contenido, especialmente para personas con baja visión. Según las pautas WCAG 2.1, el contraste mínimo recomendado es de 4.5:1 para texto normal y de 3:1 para texto grande. Algunos elementos de la interfaz de *ChatGPT*, sobre todo en su versión con tema claro, no alcanzan estos valores mínimos, presentando relaciones de contraste que en ciertos casos se sitúan entre 1.02:1 y 1.86:1.

Además de los errores anteriores, *WAVE* ha identificado algunos aspectos adicionales que se podrían mejorar:

- Texto de título redundante: En 28 casos, se detectó que el texto del atributo `title` coincide con el texto visible o con el texto alternativo. Esto puede generar confusión para los usuarios que utilizan tecnologías de asistencia, ya que el texto alternativo debería ofrecer información adicional en lugar de repetir lo que ya está visible en la interfaz.
- Problemas en elementos multimedia: Se encontraron varios elementos de medios, como videos y audios, que carecen de características accesibles fundamentales. Algunos de estos elementos no cuentan con controles, subtítulos, transcripciones o descripciones de audio, lo que dificulta el acceso al contenido para personas con discapacidades auditivas o visuales. Incorporar estos recursos es esencial para mejorar la experiencia de estos usuarios.

En resumen, el análisis de accesibilidad de *ChatGPT* realizado con *WAVE* evidencia que la plataforma no cumple con los niveles A, AA ni AAA de las WCAG 2.1. Los errores identificados, como la ausencia de etiquetas en los campos de formulario, la falta del atributo de idioma principal (`lang`), botones sin texto visible ni atributos ARIA, y el contraste insuficiente entre el texto y el fondo, violan requisitos clave de los niveles A y AA. Además, problemas adicionales como textos redundantes en los atributos `title` y la ausencia de características accesibles en elementos multimedia (como controles, subtítulos y transcripciones) limitan aún más la experiencia de usuarios con discapacidades visuales y auditivas, dificultando el cumplimiento de los estándares AAA.

El análisis realizado con *Pa11y Dashboard*, con fecha 11 de diciembre de 2024, sobre la accesibilidad de *ChatGPT* ha identificado las siguientes problemas que afectan la experiencia de uso para personas con discapacidad:

- Ausencia del atributo de idioma principal en la página: *ChatGPT* no incluye el atributo `lang` para especificar el idioma del documento. Esto dificulta que los lectores de pantalla pronuncien correctamente el contenido.
- Botones sin etiquetas accesibles: Varios botones en la interfaz de *ChatGPT* carecen de etiquetas accesibles o atributos ARIA, impidiendo que los lectores de pantalla informen a los usuarios sobre su propósito.
- Falta de alternativas textuales para contenido multimedia: Algunos elementos multimedia o gráficos en *ChatGPT* no cuentan con descripciones textuales, lo que limita su accesibilidad para usuarios con discapacidades visuales. Un ejemplo es la ausencia de descripciones textuales (atributo `alt`) en los iconos utilizados en la interfaz, como el icono de “copiar texto” o el botón de “enviar”.
- Elementos interactivos que provocan cambios de contexto inesperados: Algunos botones o campos de entrada en *ChatGPT* desencadenan cambios de contexto al recibir el foco, lo que puede confundir a los usuarios. Por ejemplo, al escribir en el cuadro de entrada y presionar la tecla **Enter**, se envía el mensaje y cambia el contexto de la conversación, lo que puede ser confuso para usuarios que no esperan un cambio inmediato.
- Contenido en movimiento sin controles para pausarlos: Elementos que parpadean, se desplazan o animan en *ChatGPT* por más de 5 segundos no ofrecen controles para detener o pausar la animación. Esto puede ser molesto o incluso peligroso para usuarios con epilepsia fotosensible. Por ejemplo, las animaciones de carga (el icono de puntos que se mueven mientras se genera una respuesta) no incluyen controles para pausar o detener el movimiento.
- Uso exclusivo de colores para transmitir información: La información codificada únicamente mediante colores en *ChatGPT* no está respaldada por texto o pistas visuales adicionales, lo que dificulta la comprensión para usuarios con daltonismo. Por ejemplo, los mensajes de error en *ChatGPT*, como cuando se desconecta la conexión a Internet, están destacados únicamente con un color rojo sin un texto descriptivo adicional.
- Título de la página no descriptivo: El elemento `<title>` en *ChatGPT* no describe adecuadamente el contenido de la página, lo que afecta la navegación y comprensión del propósito del sitio.
- Orden de contenido no lógico al deshabilitar estilos: El contenido de *ChatGPT* pierde su estructura lógica cuando los estilos CSS están deshabilitados, afectando a usuarios que dependen de lectores de pantalla. Por ejemplo, el cuadro de entrada de texto para enviar mensajes aparece visualmente desordenado, colocado antes de los mensajes generados en la conversación, lo que rompe el flujo lógico del contenido.
- Accesibilidad limitada en gestos y atajos de teclado: Los gestos complejos y atajos de teclado en *ChatGPT* no ofrecen alternativas más simples ni permiten su desactivación, lo que afecta a usuarios con movilidad reducida.

En resumen, el análisis realizado con *Pa11y Dashboard* demuestra que *ChatGPT* no cumple con los niveles A, AA ni AAA de las WCAG 2.1. Los problemas identificados, como la ausencia del atributo de idioma principal (`lang`), botones sin etiquetas accesibles, contraste insuficiente, falta de alternativas textuales para elementos multimedia, y contenido en movimiento sin controles, violan requisitos fundamentales de los niveles A y AA. Asimismo, la falta de soporte para gestos simplificados, el uso exclusivo de colores para transmitir información, y la estructura desordenada del contenido sin estilos CSS muestran que tampoco se alcanza el nivel AAA.

### 3.1.2. Análisis manual

Tras realizar el análisis automático de accesibilidad de *ChatGPT* con herramientas especializadas, se procedió a realizar una evaluación manual desde la perspectiva de los principios de COGA. En esta evaluación, se examinó cada principio y sus pautas específicas determinando su grado de cumplimiento:

1. Proveer contenido comprensible y claro: La interfaz de *ChatGPT* no cuenta con herramientas que faciliten la comprensión del contenido para personas con discapacidad cognitiva. No ofrece opciones de resúmenes automáticos, palabras clave resaltadas o explicaciones visuales complementarias que ayuden a estructurar mejor la información presentada. Tampoco incluye herramientas como mapas conceptuales, gráficos explicativos o esquemas que ayuden a organizar la información para mejorar la comprensión.
2. Diseño predecible y consistente: Aunque la interfaz de *ChatGPT* es minimalista, carece de elementos visuales que refuercen la consistencia y la predictibilidad. No hay una estructura de navegación clara con menús accesibles o botones de retroceso. Además, la falta de ayudas visuales como iconos o pistas contextuales hace que la navegación no sea intuitiva para los usuarios con dificultades cognitivas.
3. Reducir la carga cognitiva: La interfaz no segmenta adecuadamente la información, lo que puede sobrecargar cognitivamente al usuario. No existen divisiones visuales claras entre diferentes partes de una conversación, títulos o encabezados que organicen el contenido. La información se presenta como un bloque de texto continuo sin mecanismos que la estructuren en fragmentos más manejables, lo que dificulta su procesamiento.
4. Proveer ayuda y soporte contextual: La interfaz de *ChatGPT* no incluye tutoriales o guías iniciales para orientar al usuario en su uso. No hay indicaciones visibles sobre cómo formular preguntas efectivas ni opciones de asistencia integrada dentro de la interfaz. Además, la ausencia de ayudas emergentes o mensajes contextuales dificulta que los nuevos usuarios comprendan rápidamente cómo interactuar con la herramienta.
5. Apoyar la adaptación y personalización: La interfaz no permite ajustes visuales que favorezcan la accesibilidad, como el cambio de tamaño de fuente, el ajuste del contraste o la selección de esquemas de color accesibles. Tampoco ofrece

opciones para simplificar la interfaz o reducir elementos visuales que podrían ser distractores para algunos usuarios con dificultades cognitivas.

6. Mejorar la accesibilidad del tiempo y ritmo: La interfaz no proporciona controles para gestionar el ritmo de lectura del contenido. No existe una opción para pausar, ralentizar o reanudar la lectura del texto, lo que puede resultar problemático para usuarios que requieren más tiempo para procesar la información. Tampoco hay un modo de lectura asistida que permita recibir el contenido de forma más pausada o en fragmentos controlados.
7. Proveer retroalimentación y validación accesibles: La interfaz no incorpora indicadores visuales o auditivos que confirmen las acciones realizadas por el usuario. No hay iconos, cambios de color o sonidos que indiquen que una solicitud ha sido procesada correctamente. Esto puede dificultar la experiencia de usuarios con discapacidad cognitiva que necesitan confirmaciones explícitas de sus interacciones.
8. Diseñar para todos: Aunque *ChatGPT* es accesible desde distintos dispositivos, no se han implementado modos específicos para personas con discapacidad cognitiva. No hay una versión simplificada de la interfaz ni compatibilidad con tecnologías de asistencia avanzadas. Además, no existen pruebas documentadas con usuarios con necesidades cognitivas específicas, lo que limita su adecuación como una herramienta verdaderamente inclusiva.

El análisis de *ChatGPT* basado en los principios de COGA revela que, aunque su diseño sea simple y su lenguaje suele ser claro, presenta importantes limitaciones en cuanto a la accesibilidad cognitiva. Entre las principales deficiencias se encuentran la ausencia de guías o tutoriales que orienten a los usuarios, no ofrecer mecanismos para dividir explicaciones complejas en pasos más manejables y no brindar retroalimentación proactiva que confirme si los usuarios han comprendido las respuestas generadas.

## 3.2. Probando ChatGPT con usuarios con discapacidad cognitiva

El 19 de noviembre del 2024 se llevó a cabo, en la Facultad de Informática de la Universidad Complutense, una observación de usuarios con discapacidad cognitiva como parte del proyecto de investigación *HumanAI*. Esta actividad tuvo como objetivo principal evaluar cómo los usuarios con discapacidad cognitiva interactúan con plataformas de inteligencia artificial generativa, específicamente *ChatGPT*, para identificar barreras y oportunidades de mejora en términos de accesibilidad y usabilidad. Los estudiantes con discapacidad cognitiva que participaron son del título propio de la UCM llamado ACCEDE<sup>1</sup>. El programa ACCEDE es un título propio de la Universidad Complutense de Madrid dirigido a personas con discapacidad

---

<sup>1</sup><https://ucm.es/programa-accede>

intelectual y del desarrollo con el fin de formarles para ser “Técnico Auxiliar en Evaluación de Entornos Inclusivos”. Su finalidad es favorecer la inclusión socio-laboral de jóvenes con discapacidad intelectual a través de un programa de formación para el empleo y la inclusión universitaria en el entorno de la Universidad Complutense de Madrid. Para facilitar el seguimiento personalizado, los estudiantes se agruparon en pequeños subconjuntos, cada uno a cargo de una tutora distinta: la tutora académica del título ACCEDE, las directoras de este TFG, una compañera y yo. Cabe destacar que esta actividad fue posible gracias a la organización y colaboración de las directoras de TFG, quienes también son profesoras en el título ACCEDE. Como parte de su clase sobre inteligencias artificiales generativas (IAGs), nos dieron la oportunidad de participar en esta sesión, apoyando y resolviendo las posibles dudas al subconjunto de estudiantes que se nos asignaron durante la parte práctica de las actividades propuestas y realizando un análisis detallado en las evaluaciones posteriores para comprender mejor las necesidades y dificultades de los usuarios.

En esta sección se describirá el diseño de la actividad de observación directa con usuarios con discapacidad cognitiva, explicando la metodología utilizada para evaluar la interacción de estos usuarios con la plataforma. A continuación, se presentarán los resultados obtenidos, destacando las principales dificultades encontradas y patrones de uso identificados. Finalmente, se llevará a cabo un análisis de estos hallazgos, proporcionando recomendaciones para mejorar la accesibilidad y usabilidad del sistema.

### 3.2.1. Diseño

Antes de la actividad, se realizó un cuestionario a los estudiantes que se puede leer en el Apéndice A. Este cuestionario tenía como objetivo conocer el perfil del participante y su relación previa con la tecnología y la inteligencia artificial y fue diseñado para ser cumplimentado con el apoyo de su profesora, en caso de duda. El cuestionario incluía preguntas para conocer sus hábitos y actitudes frente al aprendizaje, como si prefieren pedir ayuda o resolver problemas de forma independiente, y cómo reaccionan cuando no entienden una explicación. También exploraba su relación con la tecnología, incluyendo la frecuencia con la que usan dispositivos como computadoras, teléfonos o tabletas, su confianza al utilizar nuevas aplicaciones y su capacidad para buscar y evaluar información en Internet. Además, se les preguntaba sobre su conocimiento previo de la inteligencia artificial, incluyendo si habían escuchado hablar de ella, cómo la describirían con sus propias palabras y si conocían herramientas como *ChatGPT*. Finalmente, se incluían preguntas para identificar sus expectativas y posibles preocupaciones sobre la actividad, como lo que esperaban aprender y si consideraban que la tecnología podría ayudarles en sus estudios.

También se le realizó un cuestionario a la tutora de los estudiantes en el título ACCEDE antes de la actividad, que se puede leer en el Apéndice C. Este cuestionario tenía como propósito obtener información individualizada sobre cada estudiante, incluyendo datos sobre su tipo y nivel de discapacidad, áreas específicas de dificultad cognitiva, grado de autonomía en el uso de herramientas tecnológicas y cualquier otra circunstancia relevante para su proceso de aprendizaje.

En conjunto, estos cuestionarios, diseñados por las directoras del TFG, tenían

como finalidad proporcionar un contexto integral sobre los participantes, capturando tanto la perspectiva de los propios estudiantes como la de la tutora del programa. Esta información fue fundamental para entender como enfrentar y adaptar la actividad a las necesidades particulares de cada participante al interactuar con tecnologías avanzadas como *ChatGPT*.

La sesión presencial con los usuarios con discapacidad cognitiva tenía como objetivo principal acercar la inteligencia artificial generativa a personas con discapacidad cognitiva, y observar como la utilizan, identificando posibles dificultades. Para ello, las directoras del TFG prepararon y expusieron una presentación donde se abordaban los siguientes puntos:

- *¿Qué es una IA Generativa?:* En este apartado se explicó el concepto de inteligencia artificial generativa, destacando su capacidad para crear contenido nuevo a partir de grandes volúmenes de datos previamente analizados. Se abordó cómo estas IA pueden generar texto, imágenes, audio y otros tipos de contenido, así como su funcionamiento basado en modelos de aprendizaje profundo, como los modelos *Transformers*.
- *¿Para Qué Podemos Usar ChatGPT?:* Aquí se explicaron algunas distintas utilidades de *ChatGPT*, entre las que se aportaron las siguientes que se consideraron las más útiles para facilitar la realización de tareas diarias y profesionales, como las siguientes:
  - Escribir textos, cartas, correos electrónicos.
  - Buscar información de algún tema.
  - Traducir texto.
  - Hacer resúmenes.
- *ChatGPT:* En este punto se explicaba la interfaz, sus características y cómo puede ayudarnos. Se detalló la disposición de los elementos dentro de la plataforma, incluyendo la caja de texto para introducir consultas, los botones de interacción y las opciones de configuración. También se analizaron las principales funcionalidades de *ChatGPT*, como la capacidad de mantener conversaciones contextuales, generar respuestas detalladas, adaptar el tono del texto según las necesidades del usuario o recordar conversaciones anteriores.
- *ChatGPT Puede Cometer Errores:* Se señalaron algunas limitaciones del sistema, como la posibilidad de respuestas incorrectas o desactualizadas. Aunque *ChatGPT* es una herramienta avanzada de generación de texto, no está exenta de cometer errores. Debido a que su funcionamiento se basa en modelos probabilísticos de predicción de palabras, puede generar respuestas incorrectas, incompletas o sesgadas. Se dieron diferentes ejemplos de como puede equivocarse *ChatGPT* hasta llegar a la conclusión de que “*ChatGPT* es como un ayudante que te da ideas, pero tú decides si la respuesta es buena”.
- *Consejos para Usar ChatGPT con Seguridad:* En este punto se dieron diferentes recomendaciones para el uso seguro de *ChatGPT*.

- Lee la respuesta y piensa si tiene sentido.
  - Si la respuesta es sobre algo importante (como datos para un trabajo o proyecto), confírmala en una página web, un libro o pregunta a un profesor.
  - Cuanto más clara sea la pregunta, mejor será la respuesta.
  - Si una respuesta es complicada, se puede pedir que lo explique con ejemplos o de forma más simple.
- ¿Cuáles Son los Límites de *ChatGPT*?: Aunque *ChatGPT* es una herramienta poderosa, tiene limitaciones importantes que es necesario comprender para un uso adecuado, en este punto se explicaron las que se consideraron más importantes:
- No es experto en temas específicos.
  - No puede dar opiniones personales ni tener sentimientos reales.
  - Su información puede estar desactualizada.

Después de la explicación, los estudiantes realizaron tres actividades guiadas. Una compañera, las directoras de este TFG y yo participamos en la actividad dando apoyo a los alumnos en la realización de estas actividades y tomando notas sobre las dificultades que encontraban:

- Actividad 1: Escudo de la UCM. Los participantes debían descubrir con la ayuda de *ChatGPT* por qué el cisne es el animal representado en el escudo de la Universidad Complutense de Madrid.
- Actividad 2: Clasificación de gastos. Los participantes debían usar *ChatGPT* para aprender la diferencia entre gastos por necesidad y gastos por deseo, aplicando lo aprendido para clasificar una serie de gastos proporcionados.
- Actividad 3: Preparación de una entrevista. Los participantes debían imaginar que iban a entrevistar a la Vicedecana de Ordenación Académica de la Facultad de Educación y usar *ChatGPT* para crear preguntas adecuadas y relevantes.

Al terminar las actividades, los participantes completaron un cuestionario que se puede ver en el apéndice B. Este cuestionario evaluaba la experiencia con *ChatGPT*, preguntando si les gustó, si lo encontraron útil, las dificultades que enfrentaron y si se sintieron autónomos usando la herramienta. Además, la tutora del programa, las directoras del TFG, una compañera y yo tuvimos que completar un cuestionario (apéndice D), para el grupo reducido de estudiantes asignados a cada una, para evaluar el desempeño de estos, observando su autonomía, frustración y capacidad de comprensión. Ambos cuestionarios también fueron diseñados por las directoras del TFG.

### 3.2.2. Resultados

En la actividad, que se realizó el 19 de noviembre de 2024 en la Facultad de Informática de la UCM, participaron 16 estudiantes. Las 5 observadoras que les acompañaron tomaron notas durante todo el proceso, con el fin de seguir de cerca las dinámicas y recopilar información detallada sobre la experiencia vivida por los estudiantes.

Los 16 estudiantes participantes tenían diferentes tipos y niveles de discapacidad intelectual (DI)<sup>2</sup>:

- El 73 % de los estudiantes tenían DI Ligera.
- El 13 % de los estudiantes tenían DI Límite.
- El 7 % de los estudiantes tenían DI Moderada.
- El 7 % de los estudiante tenían discapacidad visual binocular y DI Ligera.

En cuanto a su autonomía tecnológica:

- El 66 % de los estudiantes fueron clasificados como “Muy autónomos”.
- El 11 % como “Moderadamente autónomos”.
- El 6 % como “Poco autónomo”.
- El 11 % necesitan simplificación de tareas.
- El 6 % requiere motivación constante.

Del cuestionario pre-test rellenado por los estudiantes, se obtuvieron los siguientes resultados:

- El 63 % de los estudiantes indicaron que su decisión de pedir ayuda “depende de la situación”.
- El 19 % prefieren hacer las cosas por si solos.
- El 19 % indicaron que piden ayuda cuando la necesitan.
- 94 % afirmaron que a veces hacen preguntas si algo no les queda claro.
- 25 % creían que *ChatGPT* podría ser útil para estudiar.
- 19 % lo veían útil para realizar trabajos.
- 19 % lo consideraban útil para buscar información específica.

---

<sup>2</sup>La clasificación de la discapacidad intelectual es moderada, límite y ligera, de mayor a menor nivel cognitivo.

Del cuestionario pre-test, los resultados variaron según el nivel de discapacidad. Por ejemplo, los estudiantes con DI Límite mostraron más dificultades para formular preguntas adecuadas y verificar la información proporcionada por *ChatGPT*. En contraste, los estudiantes con DI Ligera mostraron mejor comprensión de respuestas y mayor autonomía.

En cuanto al cuestionario post-test realizado por las tutoras para cada estudiante del que estaban al cargo se observó que:

- El 67 % de los estudiantes con DI Límite necesitaron mucha ayuda para estructurar preguntas.
- El 33 % de los estudiantes con DI Límite lograron entender bien las respuestas.
- Solo el 18 % de los estudiantes con DI Ligera verificaron la información.
- El 73 % de los estudiantes con DI Ligera entendieron bien las respuestas.
- El 27 % de los estudiantes mostraron frustración en algún momento.

### 3.2.3. Cuestiones importantes a destacar de la actividad

Los resultados muestran que la mayoría de los estudiantes valoraron positivamente *ChatGPT*, considerándolo útil para sus estudios. Sin embargo, se identificaron los siguientes desafíos clave:

- Dificultades en la formulación de preguntas, especialmente en estudiantes con DI Límite.
- Problemas en la comprensión de respuestas extensas o complejas.
- Baja tasa de verificación de información (solo un 18 % verificó respuestas).
- Diferencias en autonomía: algunos estudiantes usaron *ChatGPT* con independencia, mientras que otros necesitaron apoyo constante.

Desde la perspectiva de las tutoras, muchos estudiantes mostraron dificultades para estructurar preguntas y verificar información, lo que sugiere la necesidad de adaptar *ChatGPT* para mejorar su accesibilidad. La frustración observada en algunos participantes estuvo relacionada con problemas técnicos o dificultades cognitivas.

Para mejorar la experiencia de estos estudiantes, se recomiendan las siguientes adaptaciones:

1. Implementar funciones de voz y respuestas más concisas, e incluir la posibilidad de entrada y salida de voz para facilitar la interacción con personas que tienen dificultades para escribir o leer. Además, proporcionar respuestas más breves y estructuradas, con la opción de expandir la información si el usuario lo necesita. Esto ayudaría a evitar la sobrecarga cognitiva y mejorar la comprensión del contenido.

2. Incorporar mecanismos para ayudar en la formulación de preguntas, y diseñar sugerencias automáticas de preguntas basadas en la consulta del usuario, proporcionando ejemplos o preguntas guiadas. Esto facilitaría el uso de *ChatGPT* para aquellas personas que tienen dificultades en la estructuración de frases o que no saben exactamente cómo formular sus dudas.
3. Diseñar estrategias para fomentar la verificación de información, incluir citas o referencias a fuentes confiables cuando sea posible, además de ofrecer un sistema de verificación en el que el usuario pueda recibir consejos sobre cómo validar la información proporcionada. También sería útil contar con un botón que ofrezca explicaciones alternativas o preguntas sugeridas para que el usuario contraste diferentes perspectivas.
4. Resolver problemas técnicos como mensajes persistentes, optimizar la interfaz para evitar la acumulación excesiva de texto en pantalla, permitiendo borrar o archivar respuestas fácilmente. También se podría implementar un historial simplificado en el que las respuestas anteriores se minimicen o se agrupen, evitando la saturación visual.
5. Reducir la cantidad de información que muestra la respuesta dada por la IA haciendo que aparezca poco a poco o incluso en *bullets*, incorporar un sistema de visualización progresiva en el que la información se presente por partes, en vez de mostrar largos bloques de texto de inmediato. También se podría ofrecer un modo de “lectura simplificada” en el que las respuestas sean más esquemáticas y fáciles de seguir, con opción de expandir detalles si el usuario lo necesita.
6. Diseñar una interfaz sencilla y que llame la atención, utilizar colores contrastantes y elementos visuales que guíen al usuario de manera intuitiva, evitando sobrecarga de texto o diseños confusos. Además, se podrían implementar botones grandes y accesibles, junto con ilustraciones o iconos que refuercen el significado de las opciones disponibles, facilitando la navegación y el uso del sistema.
7. Personalización del nivel de complejidad del lenguaje, permitir que el usuario seleccione el nivel de dificultad del texto para que las respuestas sean más comprensibles según sus necesidades.
8. Incluir explicaciones visuales, incorporar iconos, gráficos o imágenes que refuercen la información textual, facilitando la comprensión del contenido.
9. Ofrecer resúmenes automáticos, generar resúmenes breves y estructurados antes de mostrar respuestas largas para facilitar la comprensión de la información esencial.
10. Modo de asistencia guiada, implementar un sistema que ayude a los usuarios a interactuar paso a paso con la IA mediante indicaciones claras y opciones predefinidas.

11. Mayor control sobre el tono y la estructura de las respuestas, permitir a los usuarios elegir entre respuestas más cortas, directas o detalladas, según su preferencia y capacidad de procesamiento.
12. Sistema de retroalimentación simplificado, incluir opciones fáciles para que los usuarios puedan indicar si una respuesta les resultó útil o si necesitan una explicación diferente.

En conclusión, aunque *ChatGPT* puede ser una herramienta valiosa para personas con discapacidad cognitiva, es necesario mejorar su accesibilidad y la forma en que presenta y verifica la información. Con las adaptaciones adecuadas, podría convertirse en un recurso inclusivo y efectivo para el aprendizaje de este grupo de estudiantes.

### 3.3. Requisitos

Tras analizar diversas IAs generativas, evaluar la accesibilidad de *ChatGPT* conforme a las pautas WCAG y las recomendaciones del grupo COGA, y realizar pruebas prácticas con usuarios con discapacidades cognitivas, se ha elaborado una lista de requisitos clave para garantizar que la IA generativa que se desarrollará sea accesible, inclusiva y adaptada a las necesidades de los usuarios con discapacidades cognitivas. Estos requisitos se dividen en dos categorías: requisitos no funcionales, que establecen cómo debe operar el sistema para asegurar una experiencia de uso adecuada y accesible, y requisitos funcionales, que describen las funciones específicas que debe cumplir el sistema.

A continuación, se detallan los requisitos no funcionales:

1. Interfaz sencilla y personalizable:
  - La interfaz debe ser simple, fácil de navegar y con un diseño limpio que evite distracciones.
  - Los botones deben estar claramente etiquetados y utilizar iconos familiares para acciones como “Ayuda” o “Volver al inicio”.
  - Se deben ofrecer opciones de personalización visual, como ajuste de tamaño de fuente, selección de colores, configuración de contraste y modificación de espaciado, permitiendo que cada usuario adapte la interfaz a sus necesidades.
2. Lenguaje simplificado y contenido comprensible:
  - Las respuestas deben ser claras, concisas y evitar tecnicismos innecesarios.
  - Se deben ofrecer definiciones contextuales de términos técnicos al hacer clic sobre ellos, con ejemplos prácticos o sinónimos.
  - El sistema debe incorporar un mecanismo de simplificación automática para adaptar las respuestas al nivel de comprensión del usuario.

- Para facilitar la comprensión de respuestas extensas, se debe incluir la opción de generar resúmenes al inicio o presentar respuestas breves que el usuario pueda ampliar según lo necesite.
3. Respuestas segmentadas y progresivas:
    - La información debe presentarse en bloques pequeños o pasos claros y secuenciales para reducir la sobrecarga cognitiva.
    - Se debe permitir al usuario avanzar por la respuesta de manera progresiva y acceder a más detalles solo si lo solicita.
    - Es recomendable incluir herramientas visuales, como diagramas, esquemas o listas, para facilitar la comprensión de conceptos complejos.
  4. Modalidades múltiples de comunicación:
    - Las respuestas deben estar disponibles en diferentes formatos: auditivos, visuales (imágenes o gráficos), textuales y pictogramas.
    - El usuario debe poder elegir el formato que mejor se adapte a sus necesidades o preferencias.
  5. Control del tiempo y ritmo de interacción:
    - El sistema debe permitir pausar, reanudar o ralentizar la generación de respuestas para que el usuario pueda procesarlas a su propio ritmo.
    - Las respuestas largas deben mostrarse en bloques progresivos, activados solo cuando el usuario esté listo para continuar.
    - El sistema debe incluir indicadores visuales para señalar el progreso de la respuesta generada.
  6. Soporte constante y adaptabilidad:
    - El sistema debe incorporar una opción de ayuda en tiempo real que ofrezca sugerencias, ejemplos claros y preguntas de seguimiento.
  7. Compatibilidad con tecnologías de apoyo:
    - El sistema debe garantizar la compatibilidad con lectores de pantalla, navegación por comandos de voz y otros dispositivos de asistencia, siguiendo estándares de accesibilidad como ARIA y las pautas WCAG.
  8. Fomento de la autonomía y confianza del usuario:
    - El sistema debe proporcionar retroalimentación positiva y empoderar al usuario mediante explicaciones claras y guiadas sobre cómo realizar tareas específicas.
    - Se debe incorporar retroalimentación proactiva preguntando si la respuesta fue entendida o si se necesita reformularla de otra manera.
  9. Diseño para todos:

- Se debe ofrecer un modo simplificado que reduzca funciones avanzadas, simplifique la interfaz y presente la información en formatos fáciles de procesar, garantizando una experiencia accesible para todos los usuarios.

Posteriormente, se han definido una serie de requisitos funcionales que han sido organizados en cuatro secciones clave:

1. Tutorial Interactivo (Explicación Inicial): Para facilitar el primer contacto con la herramienta, se incluirá un tutorial interactivo accesible y adaptativo.

- Explicación guiada paso a paso:
  - Uso de un asistente virtual animado (avatar, icono o voz amigable) para guiar la experiencia.
  - Permitir que el usuario haga su primera pregunta de prueba con opciones sugeridas para generar confianza.
- Opciones de navegación adaptativa:
  - Si el tutorial ya se ha realizado, no se mostrará al inicio, pero se podrá acceder desde la configuración.
  - Opción para omitir el tutorial si el usuario lo desea.
  - Opción para repetir pasos específicos según las necesidades del usuario.
- Explicaciones en múltiples formatos:
  - Texto en párrafos cortos y con resúmenes en *bullets*.
  - Opción de lectura en voz alta para usuarios con dificultades de lectura.
  - Uso de videos cortos o *GIFs* explicativos para facilitar la comprensión.
- Modo de prueba controlado:
  - Tras la explicación, se permitirá que el usuario pruebe la IA con un ejemplo guiado.
  - Ofrecer retroalimentación positiva para generar confianza en el usuario.

2. Cuestionario de Personalización Inicial: Este cuestionario adaptará la interfaz y las respuestas a las preferencias y necesidades individuales del usuario.

- Datos sobre el usuario:
  - Nombre.
  - Tipo de discapacidad (TEA, dislexia, TDAH, déficit de atención, dificultades de memoria o aprendizaje, etc.).
  - Dificultades en la comprensión de información (textos largos, recordar información compleja, mantener la atención por mucho tiempo, comprender palabras técnicas o difíciles, organizar ideas, seguir instrucciones paso a paso, etc.).

- Preferencias de personalización visual y de interacción:
    - Tamaño del texto.
    - Opciones de color y contraste.
    - Fuente optimizada para legibilidad.
    - Espaciado entre líneas y párrafos.
  - Preferencias de comunicación:
    - Formato preferido para recibir respuestas (frases cortas, imágenes, ejemplos prácticos, división en pasos pequeños, lectura en voz alta, etc.).
    - Cómo gestionar términos difíciles (uso de sinónimos, explicaciones al hacer clic, etc.).
    - Preferencia por recibir resúmenes en respuestas largas.
  - Interacción con la IA:
    - Asistencia en la formulación de preguntas.
    - Confirmación de la comprensión antes de continuar.
    - Sugerencias proactivas en caso de dudas.
  - Ritmo y tiempo:
    - Opciones para pausar, reanudar o ralentizar la presentación de respuestas.
    - Ajuste de la velocidad de lectura en voz alta.
3. Generación de Texto (Asistencia en la Formulación de Preguntas): Facilitar la formulación de preguntas claras y precisas mediante diferentes herramientas.
- Sugerencias automáticas:
    - Mostrar ejemplos de preguntas comunes según el contexto y las preferencias del usuario.
  - Botón de preguntas con opciones: Opciones rápidas como “Dame un ejemplo de...”, “Explícame con un ejemplo...”, “Resume en pocas palabras...”, “¿Qué significa [palabra]?”, “Dame sinónimos de [palabra]” y “¿Cómo se hace [acción] paso a paso?”.
  - Autocompletado de consultas: Sugerencias automáticas mientras el usuario escribe
  - Plantillas de preguntas: Formularios guiados para estructurar preguntas fácilmente.
  - Entrada por voz y sugerencias adaptativas:
    - Opción de dictar preguntas mediante reconocimiento de voz.
    - Reformulación automática de preguntas confusas con sugerencias.
  - Modo “Asistente de Preguntas” (Guía Paso a Paso):
    - Proceso guiado donde el usuario responde preguntas simples para estructurar su consulta.

4. Generación de Respuestas (Presentación de la Información): Garantizar que las respuestas sean claras, accesibles y adaptables a las preferencias del usuario.
  - Presentación progresiva de la respuesta:
    - Mostrar la respuesta en bloques pequeños y permitir que el usuario avance a su ritmo.
    - Opción de “Mostrar más detalles” si el usuario lo desea.
    - Mostrar primero en *bullets* y ofrecer un botón de “Mostrar como texto”.
  - Control sobre velocidad y formato:
    - Posibilidad de pausar y reanudar la generación de respuestas.
    - Botón de “leer en voz alta” con velocidad ajustable.
    - Opción para cambiar entre texto, audio, o imágenes según preferencia.
  - Opciones para modificar la respuesta generada:
    - Botón de “Resumir respuesta”.
    - Botón de “Explicar con más ejemplos”.
    - Botón de “Reformular en lenguaje más simple”.
  - Mejoras de comprensión:
    - Posibles palabras difíciles subrayadas que, al pulsar sobre ellas, muestran sinónimos o explicaciones o incorporar un botón de “Seleccionar palabras que no entiendas” para proporcionar sinónimos.
    - Pregunta automática al usuario:
      - “¿Esta respuesta fue útil?”
      - “¿Necesitas que lo explique de otra forma?”
      - “¿Quieres saber de donde salen las respuestas?”
    - Mostrar diagramas o imágenes cuando sea necesario.
  - Uso de imágenes e iconos para representar consultas:
    - Uso de pictogramas o *emojis* según el contexto, por ejemplo, para consultas sobre plantas: (icono de semilla) semillas, (icono de sol) sol, (icono de agua) agua.



# Capítulo 4

## Diseño

En el análisis de *ChatGPT* realizado en el capítulo anterior, se identificaron varios problemas en la interfaz y el sistema actual. Para abordar estas deficiencias, se definieron requisitos funcionales específicos que sirvieron como base para el diseño de la aplicación, priorizando aquellos que facilitan la accesibilidad para personas con discapacidad cognitiva y mejoran la experiencia de interacción con modelos de lenguaje. Estos requisitos se agrupan en tres grandes categorías: Tutorial, Cuestionario de Personalización Inicial, Generación de Texto y Generación de Respuestas. En este trabajo, se ha priorizado subsecciones de las tres últimas, enfocándonos en las funciones más críticas para la experiencia del usuario. Esto incluye:

- Cuestionario de Personalización Inicial: Adaptación del sistema a las necesidades individuales de cada usuario, incluyendo identificación de dificultades cognitivas, preferencias de comunicación y control del ritmo de interacción.
- Generación de Texto: Facilitar la formulación de preguntas a través de botones predefinidos, plantillas y autocompletado, reduciendo la carga cognitiva y guiando al usuario en la construcción de consultas.
- Generación de Respuestas: Presentación progresiva del contenido, opciones para interactuar con las respuestas generadas y preguntas para asegurar que el usuario ha comprendido la respuesta generada.

Por último, se construyó el diseño final de la aplicación, integrando estos componentes para ofrecer una experiencia fluida y adaptada a las necesidades de los usuarios. A continuación, se presenta el diseño inicial de la aplicación en la Sección 4.1, que incluye los bocetos desarrollados para cada componente principal, como el cuestionario, la pantalla principal y el generador de respuestas. Posteriormente, en la Sección 4.2, se detalla el diseño final, describiendo los cambios realizados para refinar estas ideas y adaptarlas mejor a las necesidades de los usuarios.

### 4.1. Bocetos

En esta sección se explicarán los bocetos iniciales desarrollados como parte del proceso de diseño de la interfaz. Estos bocetos tuvieron como objetivo explorar di-



Figura 4.1: Página de inicio del cuestionario

ferentes alternativas de diseño y disposición de elementos para garantizar una experiencia de usuario más fluida y accesible. A través de estas representaciones visuales preliminares, se buscó obtener una visión clara de cómo organizar la información y las funcionalidades de manera eficiente, considerando las necesidades específicas del usuario final. La retroalimentación recibida sobre estos primeros bocetos fue clave para la posterior evolución del diseño, permitiendo ajustar la interfaz antes de su implementación final.

#### 4.1.1. Cuestionario

El diseño del cuestionario (correspondiente al requisito funcional 2) está estructurado de manera que cada pantalla del proceso se presenta como un paso dentro de una “aventura”, utilizando un mapa visual que permite al usuario seguir su progreso, como se observa en la parte de abajo de la Figura 4.1. A medida que el usuario responde a cada pregunta, el mapa va avanzando, lo que brinda una representación clara del avance en el cuestionario.

En la primera pantalla, como se muestra en la Figura 4.1, el objetivo es permitir al usuario ingresar su nombre para personalizar su experiencia. El mapa al fondo indica el comienzo de la aventura y la progresión. El cuadro de texto es simple y destacada por un borde amarillo, lo que atrae la atención del usuario, acompañado por un botón de “siguiente” en color rojo para facilitar la acción. El diseño limpio y sin distracciones se asegura de que el usuario se concentre en completar este primer paso sin complicaciones, brindando una introducción amigable y accesible.

En la pantalla mostrada en la Figura 4.2, la interfaz mantiene la temática de

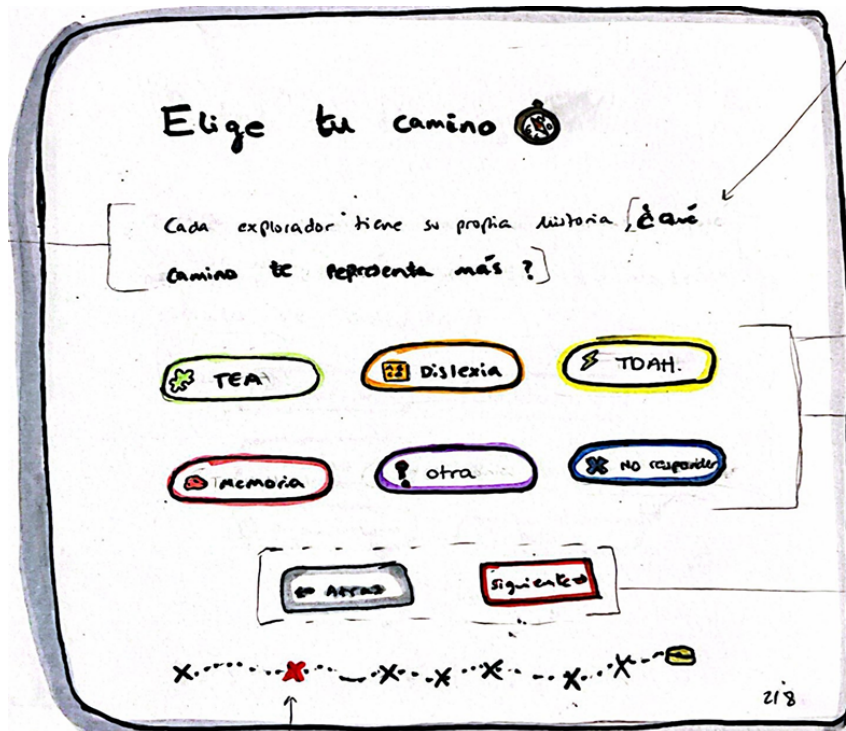


Figura 4.2: Pantalla de elección del Camino

aventura, presentando cada opción como una elección para un “explorador”. En esta etapa, el usuario debe seleccionar el camino que mejor representa su perfil, entendiendo por “camino” el tipo de discapacidad cognitiva que tiene. La pregunta principal aparece con una tipografía más grande que el resto del texto para captar mejor la atención del usuario. Las distintas opciones se presentan como botones diferenciados por colores, lo que facilita su identificación y selección. Además, al seleccionar uno de ellos, el botón cambiaría de color (en lugar de solo resaltar el borde), proporcionando una respuesta visual inmediata que refuerza la elección del usuario. Vemos en la parte inferior de la pantalla que se muestra por qué parte del mapa va el usuario para que pueda hacerse una idea de cuanto queda del cuestionario y cuanto más avanza se colorea la siguiente marca.

En la pantalla de “Retos en el viaje”, que se puede ver en la Figura 4.3, se solicita al usuario que indique sus principales dificultades al enfrentarse a diferentes tipos de información como la dificultad para entender textos largos o palabras difíciles. Cada opción se presenta a través de botones grandes y claros que permiten al usuario seleccionar fácilmente la dificultad que experimenta. Los colores siguen el mismo patrón de los botones anteriores, ayudando a identificar rápidamente las opciones.

En la siguiente pantalla “Elección de herramientas”, Figura 4.4, el usuario elige las herramientas que cree que le ayudarán a superar los retos mencionados anteriormente, como “lectura en voz”, “ejemplos” o “esquemas”. Los botones son grandes y fáciles de seleccionar, y la presencia de una mochila como icono complementa la temática de aventura, haciendo que el usuario sienta que está reuniendo las herramientas necesarias para su “viaje”. Los colores vibrantes en los botones refuerzan las opciones seleccionadas. La idea de esta pantalla es que sea un *drag and drop* y que

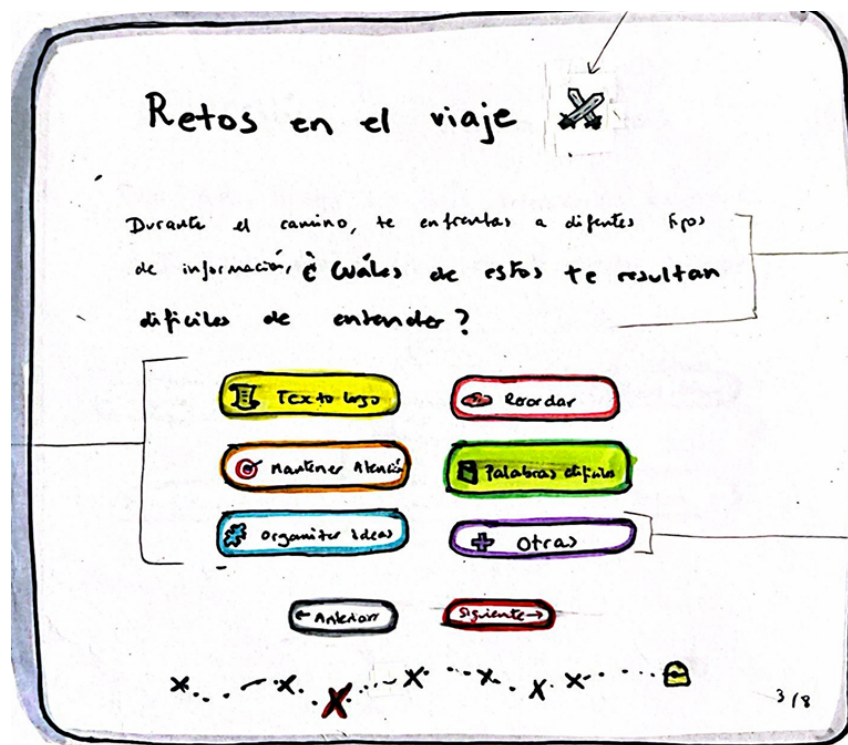


Figura 4.3: Pantalla de retos en el Viaje

el usuario arrastre a la mochilas aquellas herramientas que quiera utilizar para que sea más entretenido.

El usuario puede elegir cómo quiere que su compañero virtual le ayude durante el cuestionario, seleccionando opciones como “sólo ver ejemplos” o “explicar detalles”, como se muestra en la Figura 4.5, “Ayuda de compañero virtual”. El diseño mantiene el enfoque de aventura, con un robot simpático que actúa como guía durante el proceso. Los botones son grandes y fáciles de interactuar, y el color naranja del robot y los botones crean una atmósfera positiva. Esto no solo mejora la accesibilidad, sino que también hace la experiencia más agradable y personalizada.

En la pantalla mostrada en la Figura 4.6, “Confirmación de ayuda del compañero virtual”, el usuario confirma sus preferencias sobre cómo desea recibir ayuda del compañero virtual, eligiendo con un *toggle* entre “sí” o “no” para cada tipo de asistencia. Los botones de respuesta son fáciles de identificar y están codificados por colores (verde para “sí” y rojo para “no”), lo que facilita la toma de decisiones.

En la pantalla “Ritmo de la aventura” (Figura 4.7), el usuario dispone de un *toggle* para activar una opción que le permite pausar el texto generado, con el objetivo de procesar la información con mayor calma y evitar una experiencia abrumadora. Además, se ofrecen ajustes de velocidad para el habla de la lectura de las respuestas del cuestionario, permitiendo elegir entre las opciones “lento”, “normal” o “rápido”, como se muestra en la Figura 4.7. Los botones están diseñados con colores contrastantes y acompañados de iconos, lo que mejora su comprensión y facilita la selección. Esta opción de control de ritmo aporta un nivel adicional de personalización, permitiendo que cada usuario adapte la experiencia a sus propias necesidades.

La última pantalla “Has completado tu preparación para la aventura”, (Figura

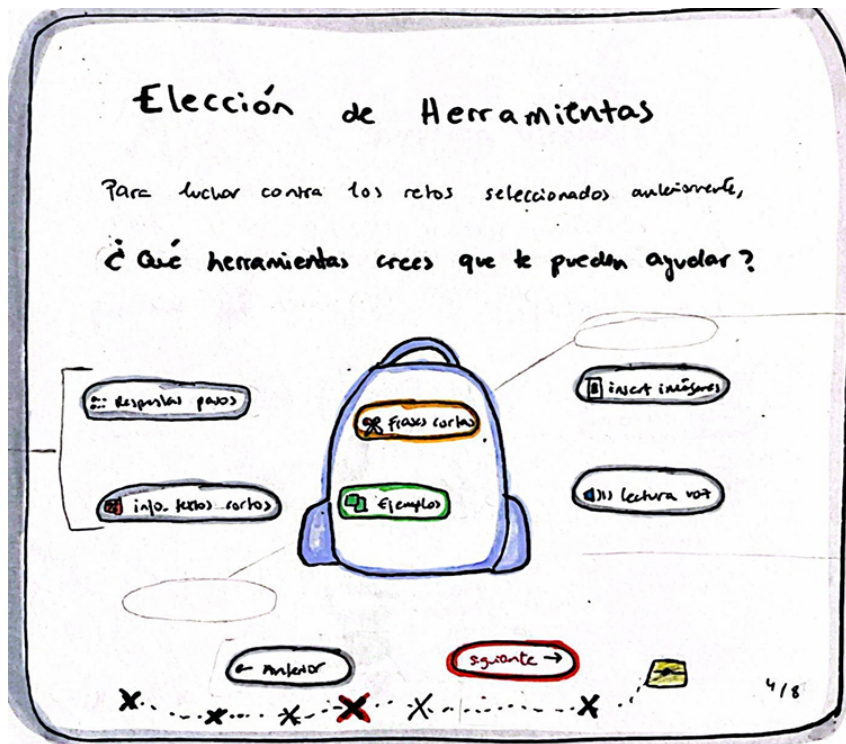


Figura 4.4: Pantalla de elección de Herramientas

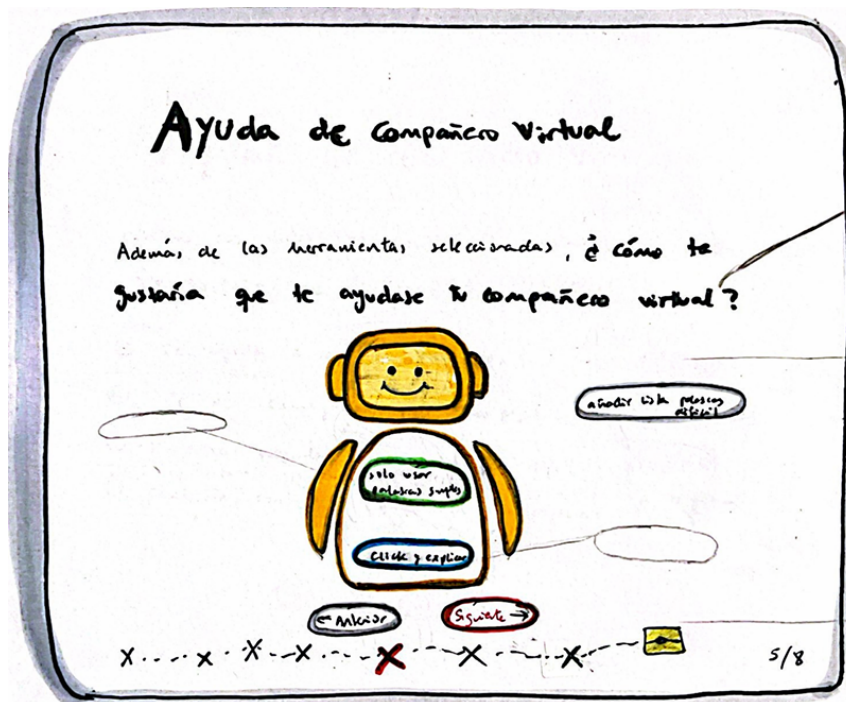


Figura 4.5: Pantalla de ayuda del Compañero Virtual





Figura 4.8: Pantalla de Confirmación Final

4.8) permite al usuario confirmar que está listo para comenzar la aventura, tras haber completado el proceso de personalización. El botón “Sí, estoy listo” es prominente y de color verde, lo que invita a la acción. El robot, que sonríe, agrega un toque amigable a la experiencia. La opción de volver a modificar alguna de las respuestas anteriores está disponible, lo que proporciona al usuario control sobre su experiencia. Esta pantalla cierra el proceso de personalización de manera positiva y motivadora, preparándolos para comenzar la aventura.

#### 4.1.2. Pantalla principal / Generador de preguntas

La pantalla principal, también conocida como la pantalla de generador de preguntas (correspondiente al requisito funcional 3), tiene como propósito facilitar la creación de preguntas personalizadas mientras mantiene la interacción simple y fluida, como se puede observar en la Figura 4.9. El objetivo principal de este sistema es ofrecer una estructura flexible que permita a los usuarios crear preguntas variadas. Las preguntas predeterminadas están organizadas en botones de colores que corresponden a diferentes tipos de preguntas, facilitando la selección por parte del usuario:

- El botón “Dame un ejemplo de ...” (amarillo) permite solicitar al usuario un ejemplo concreto de un concepto, ayudando a mejorar su comprensión y capacidad de aplicación práctica.
- El botón “Explícame con un ejemplo ...” (azul) va un paso más allá, permite solicitar al usuario una explicación más detallada con un ejemplo, lo que

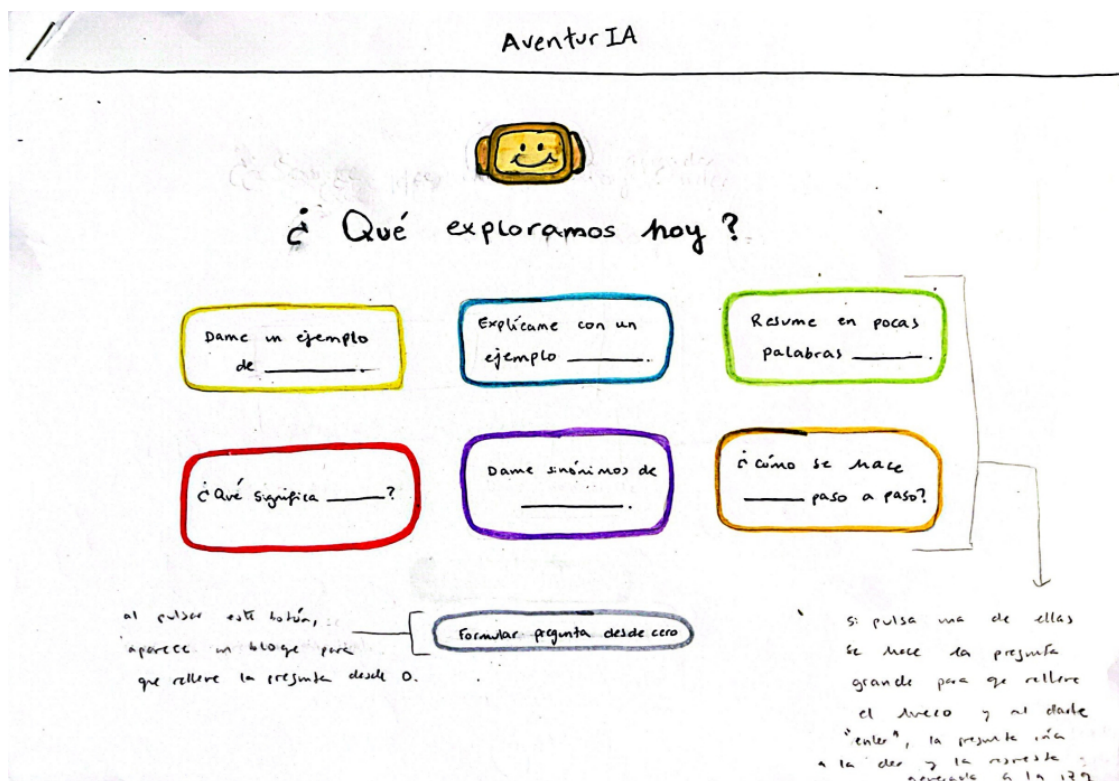


Figura 4.9: Pantalla de Generación de Preguntas principal

fomenta una comprensión más profunda del tema.

- El botón “Resume en pocas palabras ...” (verde) permite al usuario solicitar información de manera clara y precisa.
- Los botones “¿Qué significa ...?” (rojo) y “Dame sinónimos de ...” (morado) están orientadas para permitir que el usuario pueda preguntar sobre definiciones y vocabulario, respectivamente.
- El botón “¿Cómo se hace ... paso a paso?” (naranja) permite al usuario solicitar un procedimiento de un tema en concreto de manera estructurada.

También cuenta con un botón sin plantilla llamado “Formular pregunta desde cero” (gris). Es una opción clave para los usuarios que desean crear una pregunta completamente personalizada. Al seleccionar este botón, el usuario puede introducir cualquier tipo de pregunta que no esté cubierta por las opciones predeterminadas. Esta opción es ideal para situaciones en las que las preguntas predefinidas no son suficientes, dando al usuario el control total sobre la creación del contenido.

Cuando el usuario pulsa una de las preguntas, esta se agranda para destacarse en la pantalla, como se ve en la Figura 4.10, lo que ayuda a centrar la atención en la opción seleccionada. Al resaltar la pregunta, se facilita la lectura y la interacción, ya que el usuario no se siente abrumado por múltiples opciones simultáneas. Este enfoque también ayuda a guiar al usuario de manera intuitiva, asegurando que se concentre únicamente en la pregunta que está a punto de responder.

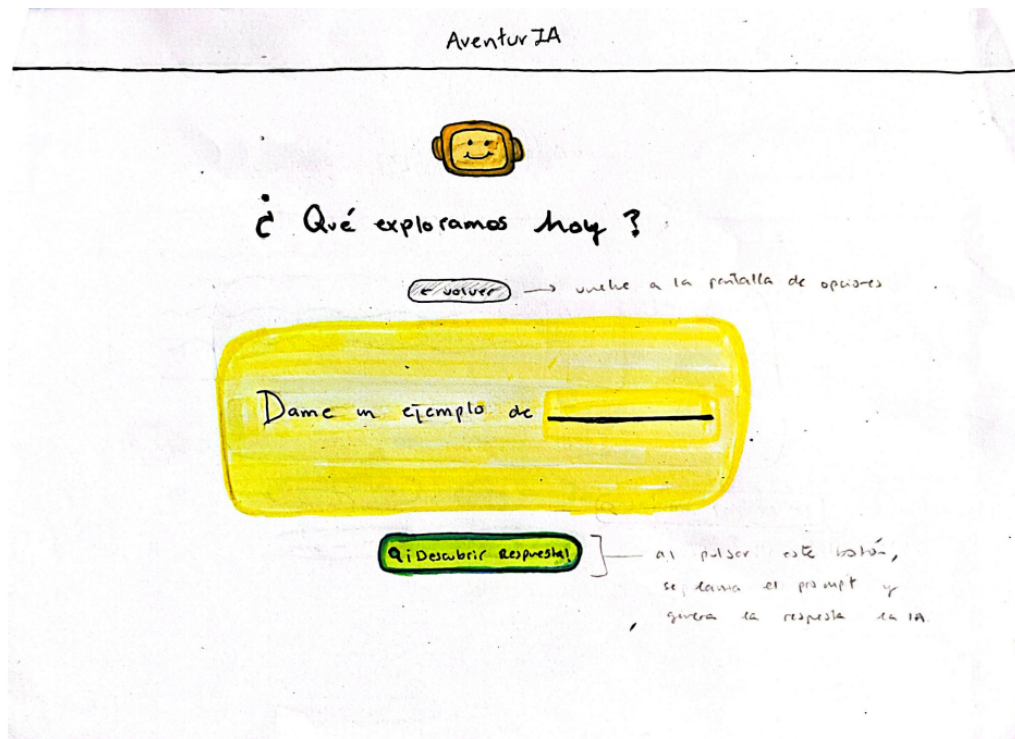


Figura 4.10: Pantalla de Generación de Preguntas tras seleccionar una opción

### 4.1.3. Generador de respuestas

A continuación se muestran las pantallas del proceso de generación de respuestas (correspondiente al requisito funcional 4). En la parte derecha se visualiza la consulta realizada por el usuario, y justo debajo, en la parte izquierda, se muestra la respuesta generada por la inteligencia artificial, como se aprecia en la Figura 4.11. Dicha respuesta se adapta dinámicamente en función de las opciones seleccionadas en el cuestionario, aplicando las preferencias de personalización establecidas por el usuario. Debajo de la respuesta, el compañero digital plantea la pregunta “¿Quieres que te ayude a entenderlo mejor?” y se presentan cuatro botones con diferentes opciones de interacción:

- “Responder en un lenguaje más sencillo” (rojo): destinado a usuarios que puedan tener dificultades para comprender la respuesta original. Al pulsarlo, incluye funciones complementarias como:
  - “Seleccionar palabras difíciles” (naranja): permite al usuario pulsar sobre términos complejos para ver sinónimos.
  - “Reformular todo el texto” (amarillo): reescribe la respuesta completa en un lenguaje más simple.
  - “Volver atrás” (gris): devuelve al generador de preguntas.
- “Explica con un ejemplo” (azul): proporciona al usuario un ejemplo relacionado con la respuesta para facilitar su comprensión.

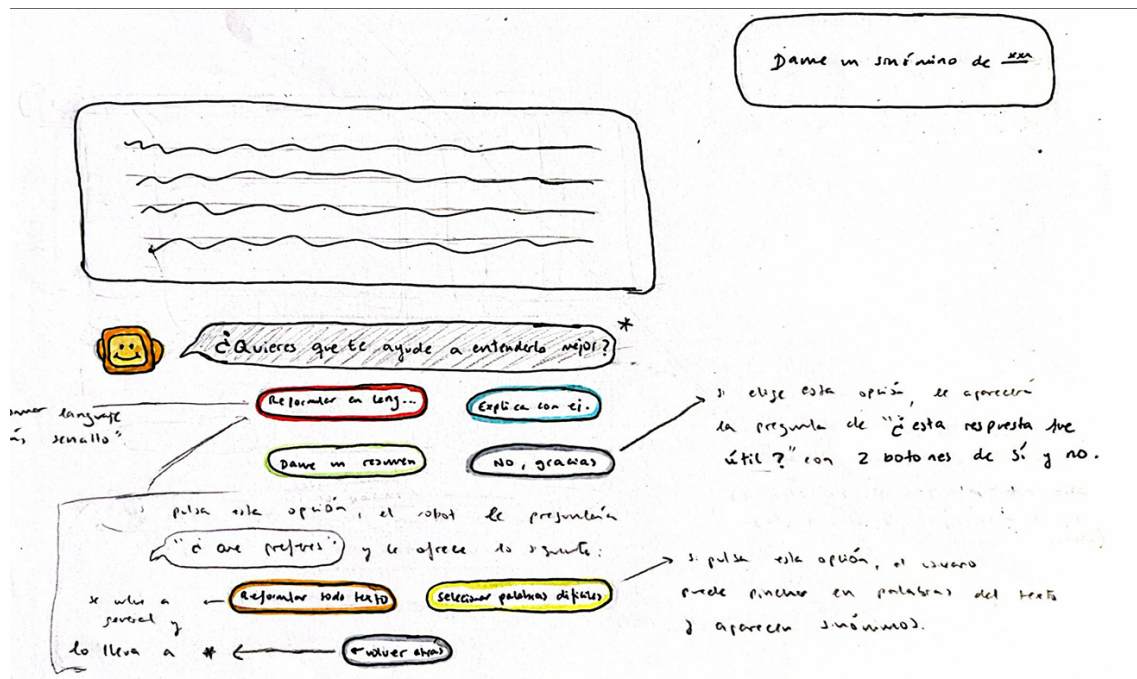


Figura 4.11: Pantalla de Generación de Respuestas

- “Dame un resumen” (verde): muestra una versión resumida de la respuesta generada, permitiendo al usuario captar la idea principal de forma rápida.
- “No, gracias” (gris): al seleccionar esta opción, se despliega la pregunta “¿Esta respuesta fue útil?” junto con dos botones: “Sí” y “No”, como se muestra en la Figura 4.12. Al pulsar “¡Sí, todo claro!”, de color verde, el compañero digital propone una nueva interacción con la frase “Las mejores aventuras empiezan con una pregunta, ¿quieres hacer otra?” y muestra de nuevo las plantillas de preguntas de la pantalla principal, permitiendo al usuario continuar con la experiencia de forma natural y sin interrupciones la conversación. En el caso de que pulse “No, tengo dudas” (naranja), se volverían a mostrar las opciones vistas en la Figura 4.11.

## 4.2. Diseño final

Tras la fase inicial de diseño basada en bocetos, antes de desarrollar la primera versión de la aplicación, realicé una sesión de *brainstorming* junto con las directoras de TFG para discutir y refinar estos bocetos, lo que nos permitió identificar mejoras clave y ajustar el enfoque del diseño. Esta sesión fue fundamental para definir el diseño final, asegurando que la aplicación fuera accesible y fácil de usar para los usuarios con discapacidad cognitiva. Se puede acceder a la aplicación a través del enlace <https://nilgroup.github.io/OlivIA-TFG-24-25/>.

En los siguientes apartados se describen en detalle los cambios realizados en cada parte de la interfaz, junto con las figuras correspondientes que ilustran el resultado final.

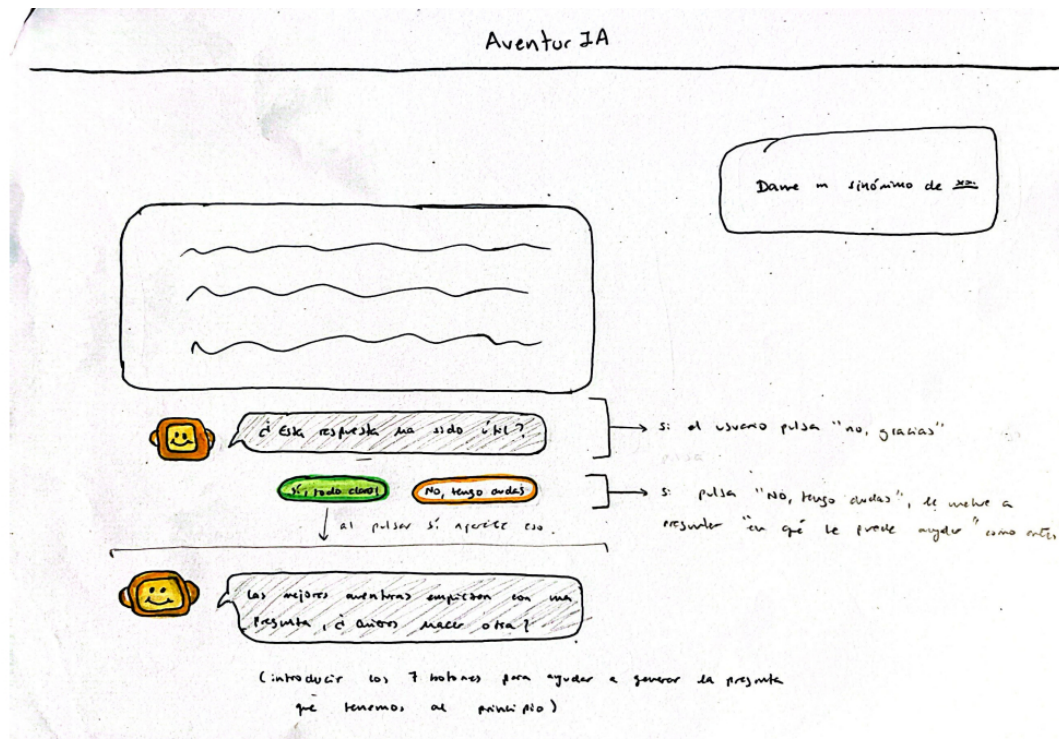


Figura 4.12: Pantalla de Confirmación Final

### 4.2.1. Cuestionario

En la versión final de la interfaz, el cuestionario inicial ha sido rediseñado para presentar la experiencia desde una perspectiva más clara y coherente con el funcionamiento real de la aplicación. A diferencia de los bocetos, donde se proponía una narrativa centrada en una aventura o viaje, como se observa en la Figura 4.1, en esta versión se ha optado por orientar el cuestionario hacia la creación de un “compañero digital” (Figura 4.13). Este cambio tiene como objetivo evitar confusiones y asegurar que el usuario entienda desde el principio la utilidad de las opciones que está configurando.

Otro ajuste importante ha sido la sustitución de mecanismos como la elección de discapacidad y dificultad mediante botones (Figuras 4.2 y 4.3), así como el uso de *drag and drop* (Figuras 4.4 y 4.5), por *toggles* (Figuras 4.14 y 4.15). Esta decisión se tomó con el fin de simplificar la interacción y mejorar la accesibilidad, permitiendo que los usuarios realicen elecciones de forma más clara y directa. Además, se optó por agrupar la pantalla de “Elección de herramientas” (Figura 4.4) con las de “Ayuda del compañero virtual” (Figuras 4.5 y 4.6), al ver que mantener ambas resultaba repetitivo, creando una versión final que se observa en en la Figura 4.16. En ésta, se observa que se ha incluido un pequeño ejemplo visual en cada opción del apartado con la finalidad de ayudar al usuario entender mejor cómo se aplicará cada herramienta seleccionada. Asimismo, se optó por eliminar la pantalla de “Ritmo de la aventura” (Figura 4.7), ya que se identificó que estaba más orientada a un perfil de usuario con discapacidad visual, un público distinto al que se enfoca esta primera versión de la aplicación.



Figura 4.13: Página de inicio de cuestionario final

Finalmente, a diferencia de los bocetos (Figura 4.8), se ha añadido a la pantalla final del cuestionario un resumen de todas las preferencias elegidas por el usuario (Figura 4.17). Esta funcionalidad permite revisar fácilmente las configuraciones sin necesidad de recordar cada paso anterior, y ofrece la posibilidad de modificar cualquier opción antes de continuar.

#### 4.2.2. Pantalla principal / Generador de preguntas

En la versión final de la aplicación, la pantalla principal ha sido modificada para mejorar la fluidez de la interacción y adaptarse al nuevo enfoque temático centrado en el compañero digital. A diferencia de los bocetos iniciales, donde al pulsar una plantilla de pregunta se abría una pantalla adicional (Figura 4.9 y Figura 4.10), ahora las plantillas se muestran directamente como bloques de texto que cambian dinámicamente según la opción seleccionada como se observa en la Figura 4.18. Esta modificación permite mantener la vista en una única pantalla, reduciendo la complejidad y facilitando la comprensión del flujo.




También se ha añadido un nuevo detalle que refuerza la cercanía y personalización de la experiencia: el nombre del usuario aparece visible en la pregunta inicial en la interfaz principal (Figura 4.18), generando una sensación de familiaridad durante la interacción.

Como parte de las funcionalidades añadidas, se han incorporado dos elementos importantes como un botón de acceso al historial, ubicado en la parte izquierda, y un botón de configuración en la parte superior derecha. Estas opciones no estaban presentes en los bocetos iniciales y permiten al usuario gestionar mejor su experiencia, ya sea revisando conversaciones anteriores o ajustando las preferencias del

**★ Paso 1: Cada persona es única**

Para que OlivIA pueda ayudarte mejor, quiere conocerte un poquito más.

**¿Con qué te sientes más identificado?**

 TEA	<input type="checkbox"/> No
 Dislexia	<input type="checkbox"/> No
 TDAH	<input checked="" type="checkbox"/> Sí
 Memoria	<input checked="" type="checkbox"/> Sí
 No responder	<input type="checkbox"/> No
 Otra opción	<input type="checkbox"/> No

[← Anterior](#) [Siguiete →](#)

- 📍 - 🏆

Figura 4.14: Elección de discapacidad en cuestionario final

**☀ Paso 2: ¡Haz que tu compañera digital sea tu mejor guía!**

**¿Qué te cuesta más entender?**

Elige todas las que veas necesarias:

- Textos largos Sí
- Palabras difíciles No
- Organizar ideas Sí
- Mantener la atención No
- Recordar No
- Otra opción No

[← Anterior](#) [Siguiete →](#)

Figura 4.15: Elección de dificultades en cuestionario final

### Paso 3: Dale una personalidad a tu compañera virtual

Teniendo en cuenta lo elegido antes

#### ¿Cómo quieres que te ayude tu compañero digital?

Elige todas las opciones que veas necesarias:

 Usar ejemplos  No

**Ejemplo:**  
Un planeta es como una pelota gigante que da vueltas alrededor de una luz muy fuerte, como el Sol. Por ejemplo, la Tierra es un planeta que gira alrededor del Sol.

 Respuestas en bullets  Si

**Ejemplo:**

-  Cuerpo celeste.
-  Órbita alrededor de una estrella.
-  Suficiente masa para ser esférico.

 Texto Corto  No

**Ejemplo:**  
Un planeta es un cuerpo celeste que gira alrededor de una estrella y tiene forma esférica.

 Frases cortas  No

**Ejemplo:**  
Es un cuerpo celeste. Orbita alrededor de una estrella. Tiene masa suficiente para volverse esférico.

← Anterior
Siguiente →

--- 🤖

Figura 4.16: Elección de herramientas en cuestionario final

### **iOlivIA está lista para ayudarte!**

Gracias por contarme cómo puedo ayudarte mejor. OlivIA ya está preparada para explicarte lo que necesites, cuando lo necesites.

**Resumen de tu nueva compañera virtual:**

-  Tu nombre: Aya
-  Te identificas con: TDAH Memoria
-  Te cuesta: Textos Largos Organizar Ideas
-  Te ayudará usando: No se seleccionó ninguna herramienta



**¿Quieres conocer a tu nueva compañera?**

 No,  
quiero  
cambiar algo

 Sí, estoy  
listo

--- 🤖

Figura 4.17: Pantalla final del cuestionario

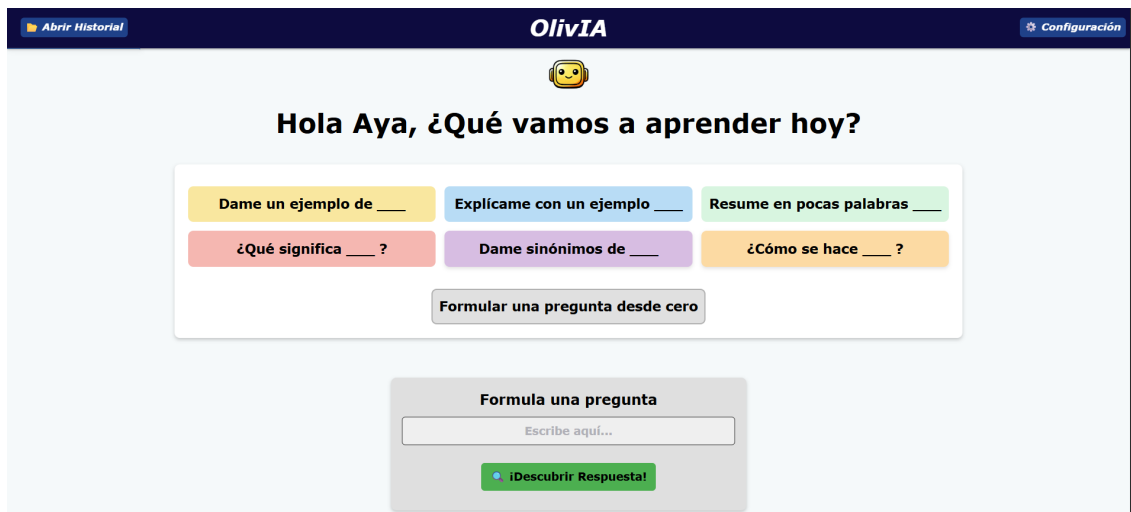


Figura 4.18: Pantalla principal de versión final

sistema.

### 4.2.3. Historial

Ni en el diseño inicial ni en los requisitos se contempló la inclusión de un historial, aunque más adelante se reconoció que esta funcionalidad aportaba valor y mejoraba la experiencia general. Esta herramienta permite al usuario consultar interacciones previas de forma clara y estructurada, manteniendo la coherencia visual con el resto de la interfaz. Por defecto, el historial se muestra colapsado bajo el botón “Abrir historial”, ubicado en la parte superior izquierda de la pantalla (Figura 4.18). Al pulsarlo, el texto cambia a “Cerrar historial”, y se despliega un panel lateral con los registros disponibles. En caso de que no exista ningún historial previo, se muestra un mensaje informativo indicando que no hay interacciones guardadas aún como se observa en la Figura 4.19. Cuando sí hay registros, estos aparecen organizados en bloques rectangulares que incluyen el título, que describe brevemente el tema tratado, junto con la fecha y hora en la que se generó la conversación (Figura 4.20). Cada entrada incluye un botón que permite abrir ese historial en pantalla completa, facilitando la lectura del contenido anterior sin perder contexto. Una vez revisado, el usuario puede cerrarlo fácilmente para volver a la interfaz principal.

Este sistema ofrece una forma intuitiva y accesible de revisar respuestas anteriores sin necesidad de repetir preguntas, aportando valor a la experiencia de uso y reforzando la sensación de control.

### 4.2.4. Configuración

Aunque no se contempló en los bocetos iniciales ni en los requisitos, durante el desarrollo se identificó la necesidad de incluir una sección de configuración para que los usuarios pudieran ajustar los datos del cuestionario inicial en cualquier momento. Esto permite personalizar la experiencia de uso y adaptar la aplicación a las necesidades cambiantes de cada usuario, mejorando tanto la accesibilidad como la



Figura 4.19: Historial abierto sin registros

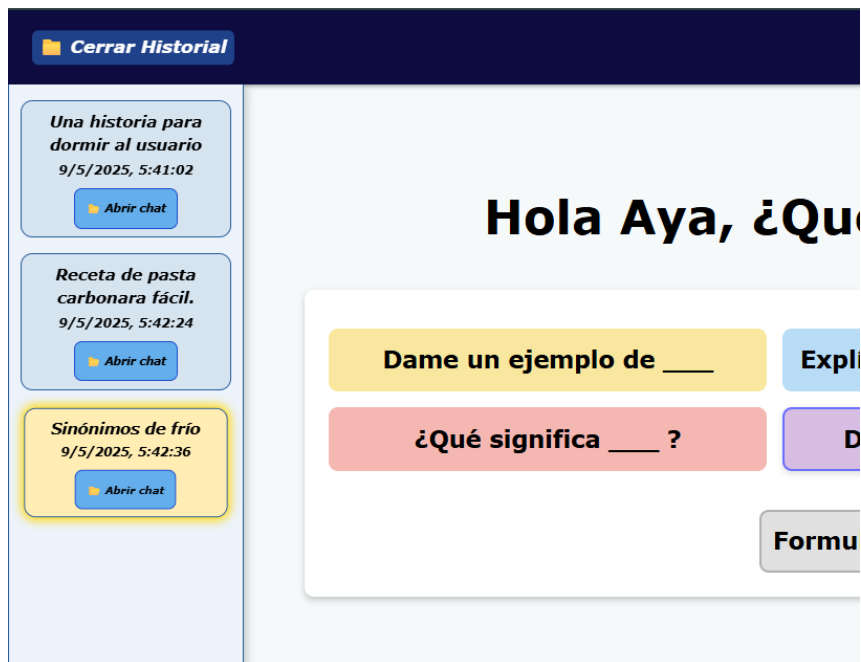


Figura 4.20: Historial abierto con registros

usabilidad del sistema.

Como se muestra en la Figura 4.21, esta sección está organizada en cuatro bloques principales que reflejan las categorías del cuestionario inicial:

- Nombre: Permite al usuario modificar su nombre para personalizar aún más la experiencia.
- Discapacidad: Incluye opciones como TEA, dislexia, TDAH, dificultades de memoria y una opción para indicar que el usuario prefiere no responder. Además, se ofrece la posibilidad de especificar una discapacidad personalizada mediante el campo “Otra”.
- Retos: Permite seleccionar las dificultades que el usuario enfrenta al procesar información, como textos largos, palabras difíciles, organizar ideas, mantener la atención y problemas de memoria.
- Herramientas: Ofrece opciones de apoyo adicionales, como el uso de ejemplos, respuestas en *bullets*, textos cortos y frases cortas, que pueden activarse o desactivarse según las preferencias del usuario.

Cada opción se presenta con un *toggle* que permite activarla o desactivarla, utilizando un diseño claro para indicar su estado (verde para activado y rojo para desactivado). En la parte inferior del panel se incluyen dos botones: “Descartar cambios” (en gris) para revertir cualquier modificación realizada, y “Guardar cambios” (en verde) para confirmar los ajustes. Esta sección es fundamental para garantizar que la aplicación siga siendo relevante y accesible a lo largo del tiempo, proporcionando una experiencia verdaderamente personalizada.

#### 4.2.5. Generador de respuesta

La versión final del generador de respuestas mantiene en gran medida la estructura planteada en los bocetos iniciales. Aunque visualmente se conserva el mismo enfoque, se han añadido opciones nuevas como la posibilidad de escuchar la respuesta generada y una gestión más clara de lo que ocurre al finalizar una conversación.

En primer lugar, cuando el usuario envía una pregunta, aparece un mensaje de “cargando” que indica que el sistema está generando la respuesta como se observa en la Figura 4.22. Esta animación ayuda a mantener al usuario informado y mejora la percepción de respuesta inmediata del sistema. Una vez generada la respuesta, se muestra por pantalla con la posibilidad de escucharla en voz alta pulsando el botón ubicado abajo-derecha de la respuesta (Figura 4.23), una funcionalidad nueva, que se encontraba en los requisitos, se incluyó pensando en la accesibilidad y la comodidad del usuario, especialmente cuando el contenido es extenso o se prefiere una experiencia auditiva. También se muestra una nueva funcionalidad de “ver más”, vista en los requisitos, que aparece automáticamente abajo-izquierda cuando una respuesta generada es larga (Figura 4.23), permitiendo al usuario expandir y contraer el contenido para mejorar la navegación y evitar sobrecarga cognitiva. También se han ajustado las opciones de interacción disponibles tras recibir la respuesta. Ya no

## Configuración del cuestionario

### Nombre

### Discapacidad

-  TEA
-  Dislexia
-  TDAH
-  Memoria
-  Prefiero no responder
-  Otra

### Retos

-  Textos largos
-  Palabras difíciles
-  Organizar ideas
-  Mantener la atención
-  Memoria
-  Otra

### Herramientas

-  Usar ejemplos
-  Respuestas en bullets
-  Texto corto
-  Frases cortas

Figura 4.21: Configuración del cuestionario inicial



Figura 4.22: Generando respuesta de usuario

existe un botón explícito de “volver”, sino que al pulsar el mismo botón otra vez, “Responder en lenguaje más sencillo”, se vuelve directamente a la vista de botones principal (de la Figura 4.24 a la Figura 4.23). Esto simplifica el flujo y evita pasos innecesarios. Además, se ha modificado el comportamiento tras pulsar la opción “Lo tengo todo claro”. En lugar de volver a mostrar todas las preguntas, como se planteaba en los bocetos, la conversación finalizada se guarda automáticamente en el historial. Para reforzar esta acción, el historial se abre de forma automática y el nuevo registro se ilumina brevemente mediante una animación, indicando al usuario que ha sido correctamente guardado (Figura 4.25). Además, la interfaz vuelve a la pantalla principal, así evitando que el usuario se sienta abrumado y facilitando una experiencia más ordenada y controlada.


### 4.3. Conclusión


A través de los bocetos se pudo experimentar y visualizar distintos enfoques, pero fue en el desarrollo del diseño final donde se tomaron decisiones fundamentales que mejoraron la experiencia de uso. La incorporación de nuevas funcionalidades, los ajustes visuales y la simplificación de la interacción no solo respondieron a necesidades detectadas, sino que también permitieron construir una interfaz accesible y funcional. Este diseño final no solo resuelve los problemas planteados en el análisis inicial, sino que también sienta una base sólida para el crecimiento y mejora futura de la aplicación.

¡Hola! Te voy a dar la receta de lasaña de manera fácil.

**Ingredientes:**  
8-10 platos de pasta (lasaña)  
1 salsa de tomate casera (o comprada)  
500g de carne molida (opcional)  
1 cebolla picada  
2 dientes de ajo picados  
1 taza de queso ricotta  
1 taza de queso mozzarella rallado  
1/2 taza de queso parmesano rallado  
Sal y pimienta al gusto

**Paso a paso:**  
Precalienta el horno a 180°C.  
Cocina la pasta según las instrucciones del paquete. Escúrrela y déjala aparte.  
Si usas carne, cocínala en una sartén con un poco de aceite. Agrega la cebolla y el ajo picados. Cocina hasta que estén suaves.  
Agrega la salsa de tomate a la sartén con la carne. Revuelve bien.  
En un tazón, mezcla el queso ricotta con un huevo y un poco de sal.  
En un recipiente para hornear, coloca una capa de salsa de tomate en el fondo.  
Coloca 4 platos de pasta encima de la salsa.  
Agrega una capa de queso ricotta.  
Agrega una capa de queso mozzarella rallado.  
Repite los pasos 7...

[Ver más](#) 

 ¿Quieres que te ayude a entenderlo mejor?

[Explicame con un ejemplo](#) [Dame un resumen](#)

[Responder en lenguaje más sencillo](#) [No, gracias](#)

¿Prefieres formular la pregunta desde cero?

Escribe aquí tu pregunta...

[¡Descubrir Respuesta!](#)

Figura 4.23: Respuesta generada para el usuario

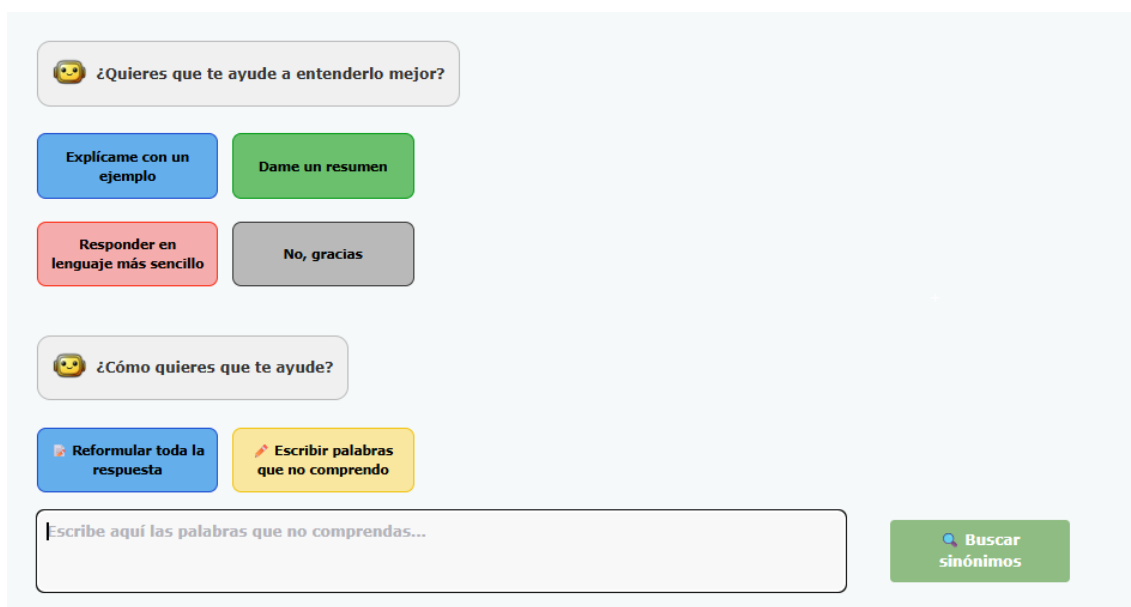


Figura 4.24: Botón de “Responder en lenguaje más sencillo”

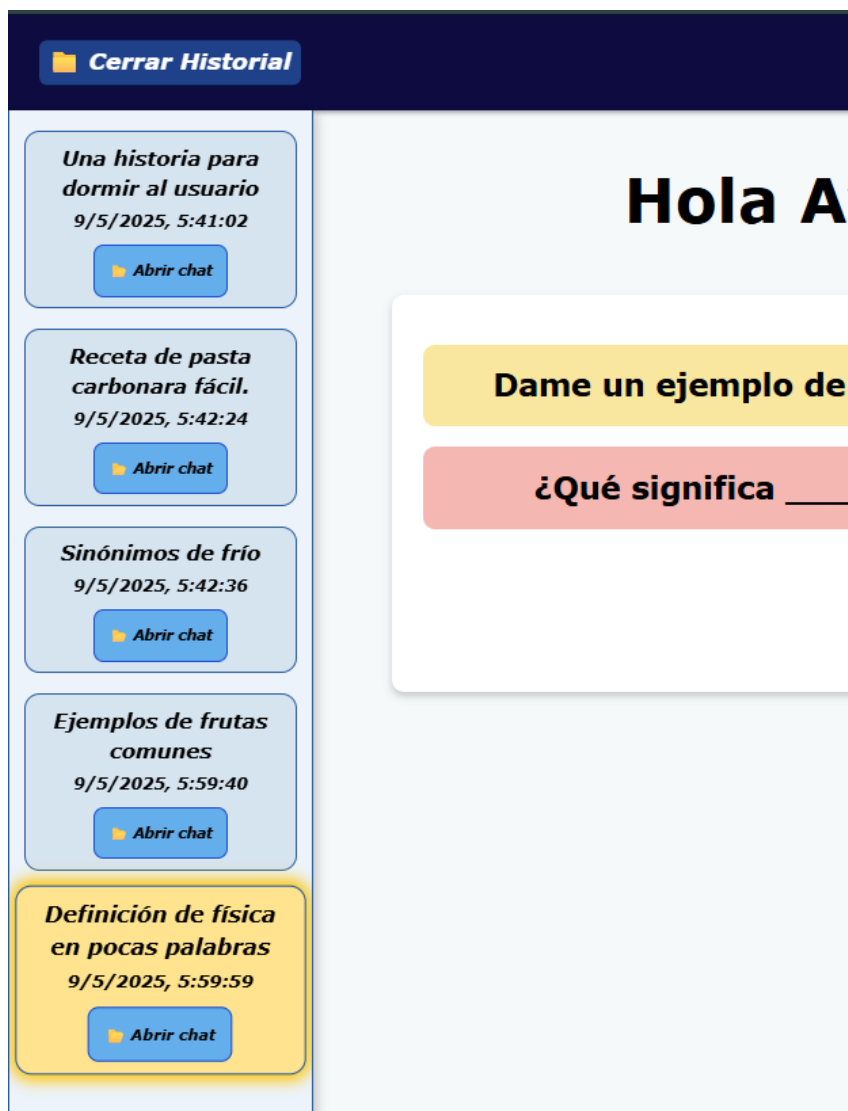


Figura 4.25: Guardado en el historial



# Capítulo 5

## Implementación

*OlivIA* es una aplicación web implementada con *React* cuyo código se encuentra en el repositorio del grupo NIL en GitHub<sup>1</sup>. En las siguientes secciones se explicará en detalle la arquitectura y cómo se han implementado cada una de las funcionalidades de la aplicación.

### 5.1. Tecnologías y arquitectura de la aplicación

Desde el punto de vista arquitectónico, *OlivIA* adopta un enfoque basado en cliente, siguiendo el modelo *serverless*, donde toda la lógica de funcionamiento se ejecuta en el navegador del usuario. Esto incluye la gestión del contenido, la interacción con el usuario y el renderizado dinámico de la interfaz, sin necesidad de contar con un servidor *backend* propio. Para procesar las operaciones de generación de preguntas, la aplicación genera una solicitud HTTP del tipo POST hacia un servicio externo de inteligencia artificial, el cual devuelve una respuesta generada por un modelo de lenguaje. En concreto, *OlivIA* se conecta principalmente a *Groq*<sup>2</sup>, una infraestructura cloud para modelos como *Meta Llama 4*, por su rapidez y eficiencia en la generación de respuestas. Como mecanismo de respaldo, utiliza *OpenRouter*<sup>3</sup>, una plataforma intermediaria que permite acceder a diversos modelos como *Mistral* o *Deepseek* desde una única interfaz, asegurando continuidad en caso de fallos o interrupciones en el servicio principal. De cara al futuro, se planea ampliar el rango de modelos utilizados, con el objetivo de asignar diferentes modelos a distintos tipos de interacciones, optimizando las respuestas según el contexto y la complejidad de cada consulta generativa.

Como se observa en la Figura 5.1, *OlivIA* se compone de una única aplicación cliente que se comunica directamente con servicios de IA externos via API. A continuación, se describen las tecnologías empleadas en ambos extremos, así como su justificación e integración.

---

<sup>1</sup><https://github.com/NILGroup/OlivIA-TFG-24-25>

<sup>2</sup><https://console.groq.com/home>

<sup>3</sup><https://openrouter.ai/>

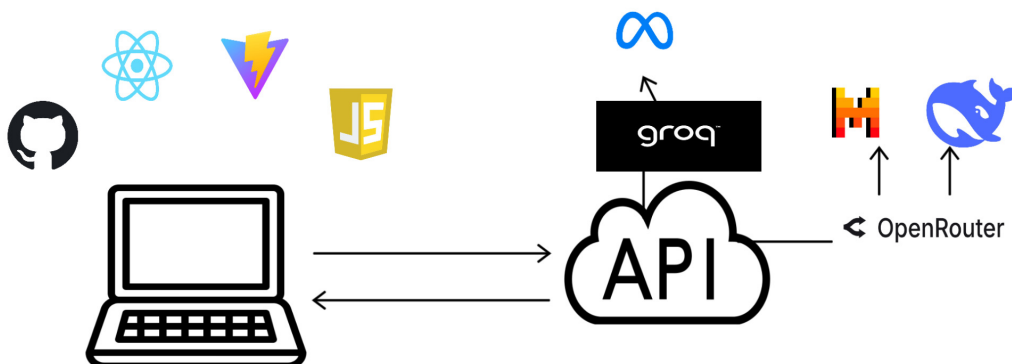


Figura 5.1: Diagrama de cliente y API

### 5.1.1. Cliente

En el lado del cliente, *OlivIA* se ha desarrollado utilizando *React*, una biblioteca de JavaScript ampliamente utilizada para construir interfaces de usuario dinámicas y modulares. La elección de *React* se debe tanto a su amplia adopción en el ecosistema web como a su modelo basado en componentes funcionales, que permite gestionar de forma eficiente el estado de la aplicación mediante hooks como *useState*. Este enfoque facilita el desarrollo y mantenimiento del proyecto, ya que permite estructurar la interfaz en bloques independientes, claros y reutilizables.

Para definir la estructura visual, se ha empleado JSX (JavaScript XML), una extensión de sintaxis que permite escribir código con apariencia similar a HTML directamente dentro de archivos JavaScript. Esta aproximación, aunque no estándar en el lenguaje, es una práctica común en el ecosistema React, ya que permite desarrollar interfaces complejas de forma más intuitiva, expresiva y legible.

Toda la lógica de interacción y manipulación de eventos en la aplicación se ha desarrollado íntegramente en JavaScript, lo que permite gestionar el flujo de datos, las respuestas del usuario y los cambios de estado de manera eficiente. La elección de JavaScript responde a su compatibilidad nativa con los navegadores y a su flexibilidad como lenguaje de programación para aplicaciones web modernas.

Para optimizar el flujo de trabajo durante el desarrollo, se ha utilizado Vite, una herramienta altamente optimizada para proyectos *frontend*. Vite ha sido seleccionado no solo por su rapidez en la transformación del código JSX a JavaScript estándar, sino también por su sistema de recarga en caliente (*hot reload*) que actualiza automáticamente la aplicación en el navegador al guardar el código, sin necesidad de recargar manualmente la página. Esta funcionalidad resulta especialmente importante en aplicaciones *serverless*, como *OlivIA*, al ser fundamental mantener activa la sesión de usuario para no tener pérdida de información. Asimismo, en el momento de generar la versión final para producción, Vite se encarga de empaquetar y optimizar el código, reduciendo su tamaño y mejorando significativamente el rendimiento de la aplicación.

Finalmente, para el despliegue de *OlivIA*, se ha optado por utilizar *GitHub Pages*, un servicio de alojamiento estático que permite servir aplicaciones directamente desde el navegador, sin necesidad de un servidor *backend* propio. La elección de *GitHub Pages* no solo se alinea con el enfoque *serverless* de *OlivIA*, sino que también resulta especialmente conveniente dado que todo el código fuente del proyecto ya se encuentra gestionado en un repositorio de *GitHub*. Esta integración facilita enormemente el proceso de publicación y mantenimiento de versiones de la aplicación, ofreciendo una solución gratuita, fiable y de rápida actualización para entornos de producción.

### 5.1.2. API

La comunicación de *OlivIA* con servicios de inteligencia artificial se ha diseñado de forma modular y flexible, lo que permite adaptarse fácilmente a distintos proveedores y modelos de lenguaje. *OlivIA* se ha integrado con dos plataformas distintas de servicios de inteligencia artificial. Por un lado, *Groq*, un proveedor de infraestructura *cloud* que permite acceder a modelos avanzados como *Meta Llama 4 Scout 17B Instruct*. Este proveedor fue elegido por su rapidez de respuesta y facilidad de uso. Por otro lado, se ha integrado *OpenRouter*, una plataforma que actúa como agregador de múltiples APIs de IA, destacando por ofrecer una amplia variedad de modelos, muchos de ellos de uso gratuito. La coexistencia de ambos servicios en el sistema permitirá seleccionar de manera dinámica el proveedor más adecuado según la tarea a realizar o las necesidades del usuario.

Durante las fases iniciales del desarrollo, se utilizó la herramienta *Apidog*<sup>4</sup> para probar y depurar las peticiones *HTTP* de forma independiente, antes de integrarlas en el flujo completo de la aplicación. Esto permitió verificar de forma aislada la correcta comunicación con los servicios externos y asegurar la validez de las respuestas obtenidas.

## 5.2. Gestión del estado del usuario durante la navegación

Para personalizar la experiencia de uso en *OlivIA*, la aplicación necesita conocer ciertas características del usuario. Esta información se recoge mediante un cuestionario inicial y se almacena en una estructura de datos centralizada llamada *summary*, que actúa como perfil de configuración de la sesión. A través de este mecanismo, es posible adaptar dinámicamente las respuestas de la IA en función de las preferencias, dificultades y necesidades expresadas por el propio usuario. El componente encargado de realizar esta recogida es *Questionario.jsx*, que implementa un formulario progresivo dividido en varias páginas. A medida que el usuario avanza en el cuestionario rellenando con sus datos, se van actualizando las propiedades del objeto *summary*, gestionado mediante el hook *useState*, como se observa en la Figura 5.2. Esta estructura contiene la siguiente información:

- Nombre: Nombre del usuario, recogida según la Figura 5.3.

---

<sup>4</sup><https://apidog.com/>

```
const [summary, setSummary] = useState({
  nombre: "",           // Para el caso 1
  discapacidad: [],    // Para el caso 2
  retos: [],           // Para el caso 3
  herramientas: [],    // Para el caso 4
});
```

Figura 5.2: useState de summary



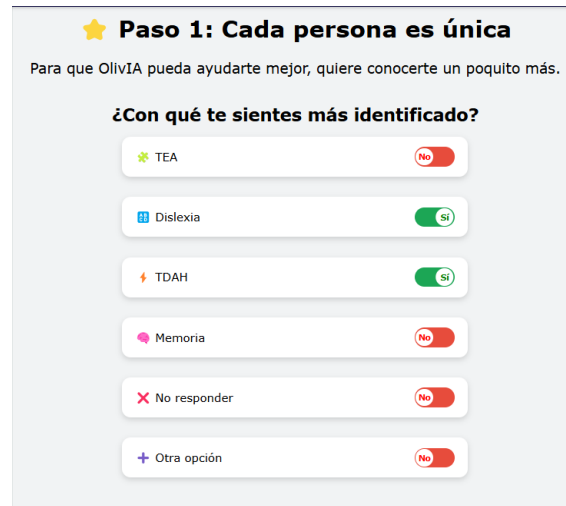
The image shows a digital questionnaire interface. At the top center is a yellow robot head icon. Below it is a bold heading: "¡Vamos a crear a tu Compañero Digital de Inteligencia Artificial!". Underneath is a welcome message: "¡Bienvenido! Hoy vas a diseñar a OlivIA, tu compañera digital única, hecho a tu medida. Ella te ayudará a aprender, resolver dudas y acompañarte en tu día a día." The main question is "¿Cómo te llamas?". Below this is a text input field containing the name "Aya". At the bottom of the input field is a red button with the text "Siguiente →". At the very bottom center, there is a small icon of a location pin and a person.

Figura 5.3: Página 1 del Cuestionario - Nombre

- Discapacidad: Array de condiciones con las que el usuario se identifica (Figura 5.4(a)). Las opciones disponibles son:
  - TEA.
  - Dislexia.
  - TDAH.
  - Memoria.
  - Prefiero no responder.
  - Otra opción personalizada, introducida mediante un campo de texto.
- Retos: Array de dificultades relacionadas con el aprendizaje o la comprensión (características que se evitarán a la hora de generar la respuesta) (Figura 5.4(b)). Las opciones mostradas en el cuestionario son:
  - Textos largos.
  - Palabras difíciles.
  - Organizar ideas.
  - Mantener la atención.
  - Recordar.
  - Otra opción personalizada (introducida manualmente).
- Herramientas: Array de ayudas seleccionadas para adaptar las respuestas (Figura 5.5). Las opciones disponibles son:
  - Ejemplo: Usar ejemplos explicativos.
  - Bullet: Estructurar la información en forma de lista.
  - Textos cortos: Respuestas con textos breves.
  - Frases Cortas: Explicaciones en frases simples y concisas.

Cada vez que el usuario modifica una respuesta, los cambios se aplican directamente sobre el estado *summary*, garantizando que el perfil esté sincronizado con las elecciones del usuario. Además, como se menciona anteriormente, el sistema permite registrar entradas personalizadas en los campos “Otra opción”, lo cual proporciona mayor flexibilidad.

Una vez completado el cuestionario, aparece una última pantalla con un resumen de las opciones seleccionadas (Figura 5.6). Este perfil se transfiere al componente principal de la aplicación mediante una función llamada *onComplete(summary)*, que es ejecutada desde *Questionario.jsx* al pulsar el botón verde de “Sí, estoy listo” localizado en la última pestaña. Esta función ha sido previamente definida en *App.jsx*, el componente que actúa como controlador principal de la aplicación. Se encarga de sobrescribir y actualizar el estado global *summary* con la información proporcionada por el cuestionario y de activar un cambio de vista mediante la variable booleana *completed* que hace pasar de la pantalla del cuestionario a la interfaz principal.



**★ Paso 1: Cada persona es única**

Para que OLIVIA pueda ayudarte mejor, quiere conocerte un poquito más.

**¿Con qué te sientes más identificado?**

- ✱ TEA  No
- 📖 Dislexia  Si
- ⚡ TDAH  Si
- 🧠 Memoria  No
- ✖ No responder  No
- + Otra opción  No

(a) Página 2 del Cuestionario - Discapacidad.



**☀ Paso 2: ¡Haz que tu compañera digital sea tu mejor guía!**

**¿Qué te cuesta más entender?**

Elige todas las que veas necesarias:

- 📄 Textos largos  Si
- ✱ Palabras difíciles  No
- 🗂 Organizar ideas  No
- 👁 Mantener la atención  Si
- 🧠 Recordar  No
- + Otra opción  No

(b) Página 3 del Cuestionario - Retos.

Figura 5.4: Cuestionario: (a) Página 2 - Discapacidad, (b) Página 3 - Retos.

También, al pasarle *summary* a *InterfazPrincipal.jsx*, permite que el usuario modifique su perfil desde el panel de configuración, implementado en *ConfigPanel.jsx*. Este componente muestra una copia editable del estado *summary* llamado *tempSummary*, gestionado de forma local. Los cambios del usuario se aplican sobre esta copia intermedia, y al pulsar “Guardar cambios” se actualiza el perfil original. Esta separación asegura que los cambios no se apliquen hasta que el usuario los confirme, permitiendo una experiencia de edición más segura y reversible.



### Paso 3: Dale una personalidad a tu compañera virtual

Teniendo en cuenta lo elegido antes

#### ¿Cómo quieres que te ayude tu compañero digital?

Elige todas las opciones que veas necesarias:

✍ Usar ejemplos

**Ejemplo:**  
Un planeta es como una pelota gigante que da vueltas alrededor de una luz muy fuerte, como el Sol. Por ejemplo, la Tierra es un planeta que gira alrededor del Sol.

📄 Respuestas en bullets

**Ejemplo:**  

- 🔸 Cuerpo celeste.
- 🔸 Órbita alrededor de una estrella.
- Suficiente masa para ser esférico.

📄 Texto Corto

**Ejemplo:**  
Un planeta es un cuerpo celeste que gira alrededor de una estrella y tiene forma esférica.

🗨 Frases cortas

**Ejemplo:**  
Es un cuerpo celeste. Orbita alrededor de una estrella. Tiene masa suficiente para volverse esférico.

Figura 5.5: Página 4 del Cuestionario - Herramientas

## ¡OlivIA está lista para ayudarte! 🚀

Gracias por contarme cómo puedo ayudarte mejor. OlivIA ya está preparada para explicarte lo que necesites, cuando lo necesites.

**Resumen de tu nueva compañera virtual:**

- 👤 Tu nombre: Aya
- ★ Te identificas con: TDAH Dislexia
- 📌 Te cuesta: Textos Largos Mantener Atencion
- 👉 Te ayudará usando: Respuestas en bullets



### ¿Quieres conocer a tu nueva compañera?

No, quiero cambiar algo

Sí, estoy listo

Figura 5.6: Página 5 del Cuestionario - Resumen

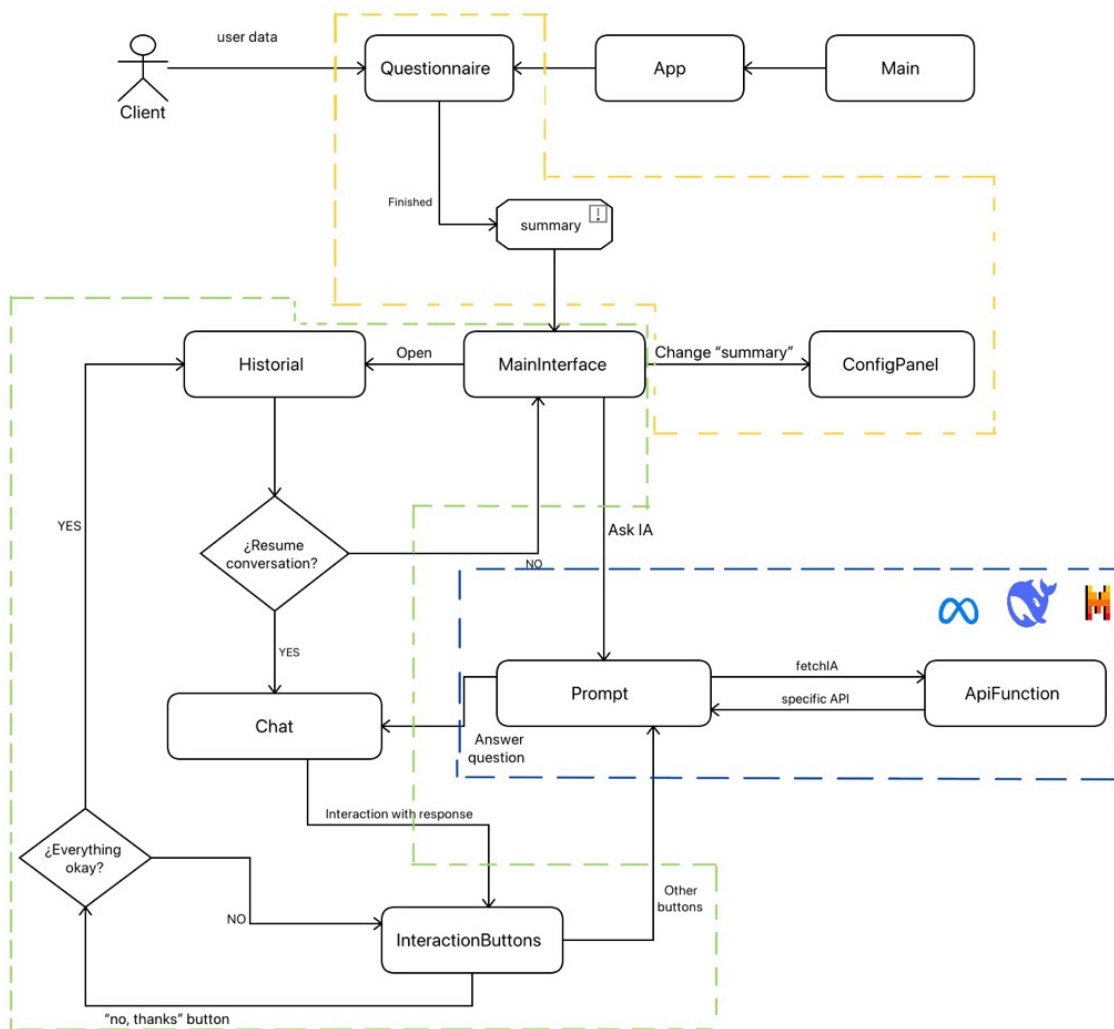


Figura 5.7: Diagrama de flujo de los componentes de *OlivIA*

## 5.3. Componentes de la aplicación

Los componentes de *OlivIA* se basan en una estructura modular, donde cada componente tiene una responsabilidad específica dentro del flujo funcional de la aplicación. Además, la aplicación se compone de un archivo *App.css*, que define el estilo de la aplicación web. En la Figura 5.7 se representa el diagrama de flujo de interacción entre los distintos módulos que la componen. Estos módulos se pueden agrupar en tres grandes secciones funcionales diferenciados por colores:

- Personalización (amarillo): Encargado de recoger y modificar el perfil del usuario. Incluye el módulo *Questionario.jsx*, responsable de recopilar la información del usuario mediante un formulario dividido en cinco pasos, el objeto *summary* con el perfil personalizado y el módulo *ConfigPanel.jsx*, que permite editar y actualizar dinámicamente el perfil de usuario mediante una copia temporal de *summary* antes de aplicar los cambios.
- Sistema (verde): Controla el flujo general de la aplicación, la navegación entre pantallas y la interacción conversacional. Incluye el módulo *Main.jsx*, cuya responsabilidad principal es indicar al navegador en qué parte del documento HTML debe mostrarse la interfaz de *OlivIA*, *App.jsx*, componente principal que dirige el flujo principal entre el cuestionario inicial y la interfaz conversacional, *InterfazPrincipal.jsx*, módulo que engloba la interfaz principal de la aplicación y gestiona el flujo completo del sistema, *Chat.jsx*, que gestiona el flujo conversacional del usuario, mostrando mensajes propios y las respuestas de la IA generativa, *ChatHistory.jsx*, que gestiona el historial de conversaciones anteriores, permitiendo guardar, visualizar o retomar sesiones pasadas y *BotonesInteraccion.jsx*, componente que agrupa los botones auxiliares que permiten interactuar con las respuestas dadas por la IA generativa.
- Prompts (azul): Se encarga de la construcción de los mensajes enviados a la IA y la comunicación con los modelos de lenguaje externos. Está compuesto por *Prompts.jsx*, que contiene la lógica de construcción y envío de prompts y *apiFunctions.jsx*, que realiza las llamadas HTTP a las APIs de proveedores externos.

A continuación, se describen en más detalle los módulos principales que forman cada uno de estas secciones.

### 5.3.1. Personalización

La sección de personalización abarca los módulos responsables de la construcción y modificación del perfil del usuario. Este perfil, modelado en el objeto *summary*, constituye el núcleo sobre el cual se adapta el comportamiento de la interfaz conversacional. La información recogida en esta fase condiciona la generación de prompts, la presentación de resultados y el uso de herramientas de ayuda en el resto de la aplicación. Aunque la estructura de *summary* ya ha sido analizada en la sección 5.2, en este apartado se detallan las decisiones arquitectónicas y de implementación que permiten su creación y edición.

### 5.3.1.1. Cuestionario Inicial

El módulo *Questionario.jsx* se encarga de la recogida de datos mediante un formulario progresivo dividido en cinco etapas (los detalles ya se han comentado en la Sección 5.2). Su estructura se basa en un enfoque de navegación secuencial controlado por una variable de estado *page*, gestionada con *useState*. Este valor determina qué vista se debe renderizar en cada momento a través de un *switch* central encapsulado en la función *renderPage()*. Esta estrategia permite una segmentación clara de cada pantalla, facilitando el mantenimiento y la incorporación de nuevas etapas en el futuro.

Toda la información recopilada durante el cuestionario se almacena en el objeto *summary*, que contiene todos los datos introducidos (nombre, discapacidades, retos, herramientas, etc.). A medida que el usuario va respondiendo a cada pregunta, este objeto debe ir actualizándose. Sin embargo, debido a las restricciones de React respecto a la mutabilidad del estado, esta actualización no se realiza directamente. En lugar de eso, este proceso se lleva a cabo usando una técnica de la actualización parcial del estado llamada el operador de propagación (...), que permite modificar solo una parte específica del objeto *summary* sin perder el resto de la información ya almacenada. Esta actualización se realiza mediante funciones específicas como *handleNameChange*, *togglediscapacidad*, *toggleReto* y *toggleTool*. Cada una de estas funciones encapsula la lógica necesaria para actualizar de forma segura y específica los distintos campos de *summary*. En todas ellas, se sigue un mismo patrón, se pasa una función al actualizador de estado *setSummary*, que recibe el valor actual (*prevSummary*) y devuelve una nueva versión del objeto. Dentro de esta función, se utiliza el operador de propagación (*...prevSummary*) para copiar el contenido previo y, a continuación, se sobrescribe únicamente el campo que se desea modificar. Por otro lado, para capturar opciones personalizadas introducidas por el usuario en los apartados “Otra opción”, se utiliza un sistema paralelo basado en variables auxiliares como *otraSeleccionada* y el objeto *otraData*. Este último almacena de forma separada el contenido textual introducido por el usuario y controla si dicho contenido ha sido guardado en *summary*, permitiendo una experiencia de edición controlada y reutilizable en diferentes apartados del formulario.

La navegación entre pantallas se realiza a través de las funciones *nextPage()* y *prevPage()*, que modifican el valor de *page*. Estas funciones están conectadas a los botones “Siguiente” y “Anterior” situados al pie de cada pantalla. Finalmente, en la última página se presenta un resumen del perfil generado, mediante la función *generateSummary()*, que muestra de forma visual toda la información recogida (Figura 5.6). Desde esta pantalla, el usuario puede optar por regresar a la primera página para modificar alguna entrada mediante *setPage(1)*, o confirmar su perfil actual pulsando “Sí, estoy listo”, lo que dispara la función *onComplete(summary)*. Esta acción transfiere el objeto *summary* al componente superior *App.jsx*, donde quedará almacenado como estado global para su uso posterior en la interfaz conversacional.

### 5.3.1.2. Panel de Configuración

El módulo *ConfigPanel.jsx* permite al usuario editar dinámicamente los datos introducidos previamente en el cuestionario inicial. Esta funcionalidad resulta esencial

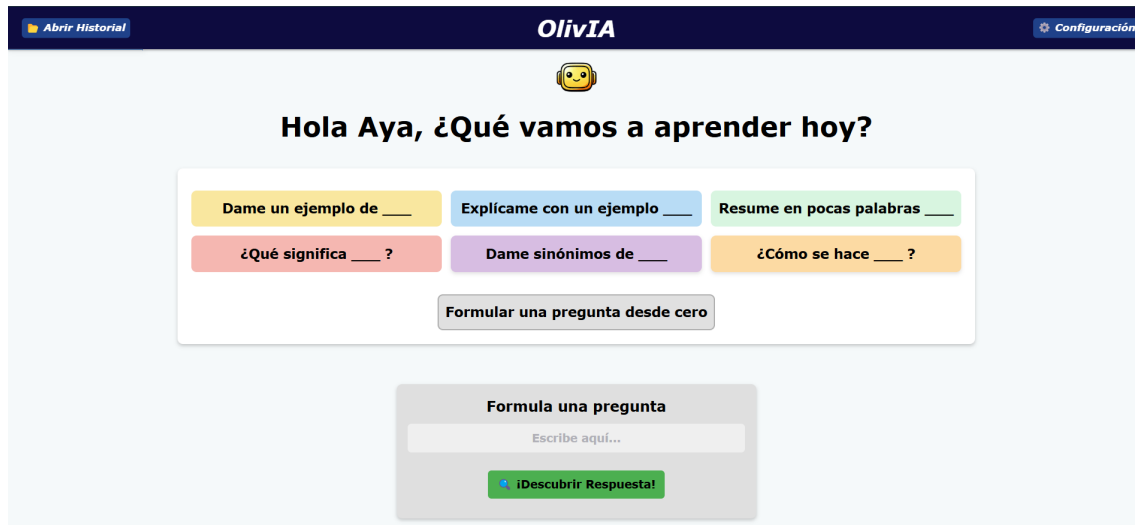


Figura 5.8: Interfaz Principal

para ofrecer una experiencia personalizable y flexible durante toda la sesión, permitiendo corregir errores o adaptar preferencias sin tener que reiniciar el proceso. Para acceder a dicho panel de configuración, la interfaz principal (*InterfazPrincipal.jsx*) incluye un botón en la esquina superior derecha (Figura 5.8, arriba a la derecha) que permite desplegar y ocultar el panel en cualquier momento, facilitando así la edición del perfil sin interrumpir el flujo de uso.

Desde un punto de vista técnico, este componente opera a partir de una copia intermedia del objeto *summary*, denominada *tempSummary*, que es gestionada como estado local editable. Esta estrategia evita que los cambios del usuario se apliquen directamente sobre el perfil activo hasta que se confirmen explícitamente, lo que ofrece un control de edición más seguro y reversible.

La interfaz del panel se genera dinámicamente recorriendo las claves relevantes de *summary* mediante *Object.entries(summary)*, filtrando por los campos editables *nombre*, *discapacidad*, *retos* y *herramientas*. Para cada uno de ellos, se muestra un bloque independiente con los controles adecuados (Figura 5.9). En el caso de *nombre*, se trata de un input de texto. Para los demás campos, se enseña conmutadores tipo *toggle*, representados como interruptores personalizados, que permiten seleccionar o deseleccionar cada opción. Las opciones disponibles para cada categoría se definen mediante una función auxiliar llamada *getOptionsForField()*. Además, otras funciones como *getLabelForOption()* y *getEmojiForOption()* permiten enriquecer visualmente la presentación del panel, mostrando una etiqueta comprensible y un emoji identificativo para cada elemento en vez del *id* que se utiliza para seleccionar la opción internamente.







La lógica de interacción se centraliza en la función *toggleOption()*, que es llamada cada vez que el usuario activa o desactiva una casilla. Esta función evalúa si la opción ya está presente en el array correspondiente dentro de *tempSummary*, y la añade o elimina según corresponda. En el caso especial de seleccionar “Otra opción”, se activa un campo de entrada adicional (controlado por el estado *otraOpciones*) que permite al usuario introducir un valor personalizado. Este valor se guarda y representa en

# OlivIA







## Configuración del cuestionario

### Nombre

### Discapacidad

 TEA	<input type="checkbox"/>
 Dislexia	<input checked="" type="checkbox"/>
 TDAH	<input checked="" type="checkbox"/>
 Memoria	<input type="checkbox"/>
 Prefero no responder	<input type="checkbox"/>
 Otra	<input type="checkbox"/>

### Retos

 Textos largos	<input checked="" type="checkbox"/>
 Palabras difíciles	<input type="checkbox"/>
 Organizar ideas	<input type="checkbox"/>
 Mantener la atención	<input checked="" type="checkbox"/>
 Memoria	<input type="checkbox"/>
 Otra	<input type="checkbox"/>

### Herramientas





 Usar ejemplos	<input type="checkbox"/>
 Respuestas en bullets	<input checked="" type="checkbox"/>
 Texto corto	<input type="checkbox"/>
 Frases cortas	<input type="checkbox"/>

Figura 5.9: Interfaz de panel de configuración

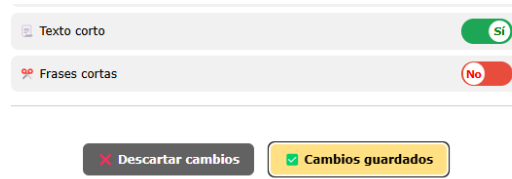


Figura 5.10: Efecto de guardado de configuración

*tempSummary* como una cadena con el prefijo “Otra - ”, tal y como se hace en el cuestionario.

En la parte inferior del panel se presentan dos botones, “Descartar cambios” y “Guardar cambios”. El primero restaura *tempSummary* a su valor original usando una copia directa del objeto *summary*, mientras que el segundo aplica los cambios, sobrescribiendo las propiedades relevantes del perfil activo. Para ofrecer retroalimentación visual al usuario de que se han guardado sus cambios, se activa temporalmente una clase CSS especial (*saved-effect*) que genera una animación de confirmación tras guardar los cambios y cambia de “Guardar cambios” a “Cambios guardados”, como se observa en la Figura 5.10.

### 5.3.2. Sistema

El núcleo funcional del sistema se encuentra en el módulo *InterfazPrincipal.jsx*, el cual centraliza y coordina el resto de los módulos secundarios responsables de la conversación, la configuración, el historial y la generación de prompts. Su diseño parte de una arquitectura orientada a estados, donde cada acción del usuario o evento interno modifica uno o varios estados gestionados mediante *useState*, y cuya actualización desencadena actualizaciones parciales que mantienen sincronizada la interfaz sin necesidad de recargar la página completa.

Al iniciar el componente, se definen múltiples estados que permiten controlar el flujo completo de la aplicación. Entre los principales se encuentran:

- *chatFlow*, que almacena el historial de mensajes enviados y recibidos durante la sesión.
- *prompt* y *selectedOption*, que controlan la entrada del usuario y la plantilla seleccionada para la generación de pregunta.
- *showChat*, que determina si mostrar la interfaz de entrada o el flujo conversacional.
- *showConfig*, *showHistory* y *activeChat*, que habilitan o deshabilitan la visualización del panel de configuración, el historial y sesiones pasadas, respectivamente.

La lógica principal de generación de prompts está encapsulada en el hook personalizado *usePromptFunctions*, importado desde *Prompts.jsx*. Este hook recibe el

perfil *summary* junto con *setters* clave del estado general, y devuelve funciones reutilizables como *sendPrompt*, *requestExample*, *requestSummary* o *generateTitleFromChat*. Esto permite mantener aislada la lógica de interacción con la API y facilita el mantenimiento del código.

Desde este punto, *InterfazPrincipal.jsx* es capaz de utilizar el contenido de *summary* para personalizar dinámicamente las interacciones con el modelo de lenguaje en función del perfil recogido durante el cuestionario. Esta personalización se aplica a través de la función *buildPrompt* (implementada en *Prompts.jsx*), en el bloque *personalInfo*, que tiene como objetivo generar una entrada que guíe al modelo de IA a responder conforme a las necesidades del usuario. Este bloque *personalInfo* se construye dinámicamente utilizando las propiedades de *summary*, y actúa como un conjunto de instrucciones adicionales que se adjuntan a la pregunta original formulada por el usuario.

El flujo de conversación comienza cuando el usuario selecciona una plantilla de pregunta (estado *selectedOption*) o escribe una consulta personalizada, como se observa en la Figura 5.8. Al pulsar el botón de “Descubrir respuesta”, se invoca *sendPrompt*, que internamente genera el prompt personalizado teniendo en cuenta el estado *summary* y lo envía al modelo de lenguaje mediante *apiFunctions.jsx*. Mientras se espera la respuesta, se actualiza el estado *loading* y se muestra un mensaje temporal (“Cargando...”) para que el usuario sepa que está en funcionamiento. La aplicación de *summary* en la respuesta se ve reflejado en la Figura 5.11, donde el usuario le pide una respuesta más larga y la IA generativa le responde destacando su discapacidad y procede a dar la respuesta en bullets.

### 5.3.2.1. Flujo del chat

Una vez recibida la respuesta, esta se añade al estado *chatFlow*, activando el flujo conversacional. En este punto entra en acción el componente *Chat.jsx*, cuya responsabilidad es mostrar todos los mensajes generados durante la sesión, tanto del usuario como de la IA (Figura 5.12) y gestionar las características adicionales de las respuestas generadas por parte la IA generativa. Cada mensaje generado por parte de la IA generativa se muestra utilizando *ReactMarkdown*, permitiendo así representarse en texto enriquecido (como negritas, listas o títulos). Si la respuesta es extensa (al llegar a los 1000 caracteres), se recorta y añade un botón “Ver más” (Figura 5.11) que, al ser pulsado, pasa a poner “Ver menos” y expande la totalidad del contenido gracias a la lógica gestionada por el estado *expandedResponses* y la función *toggleExpanded()*. Además, el componente incorpora un botón de reproducción por voz basado en la API *SpeechSynthesis*, que permite al usuario escuchar la respuesta generada (Figura 5.11). Este sistema controla el estado de lectura con variables como *speechState* y *activeSpeechId*, permitiendo reproducir, pausar o reanudar la lectura según corresponda y prefiera el usuario.

### 5.3.2.2. Botones para interactuar con la respuesta de la IA generativa

Tras cada respuesta de la IA, el usuario puede acceder a una serie de botones auxiliares agrupados en el módulo *BotonesInteraccion.jsx*. Estos botones son los mismos para todos los usuarios y permiten reformular o enriquecer la respuesta con

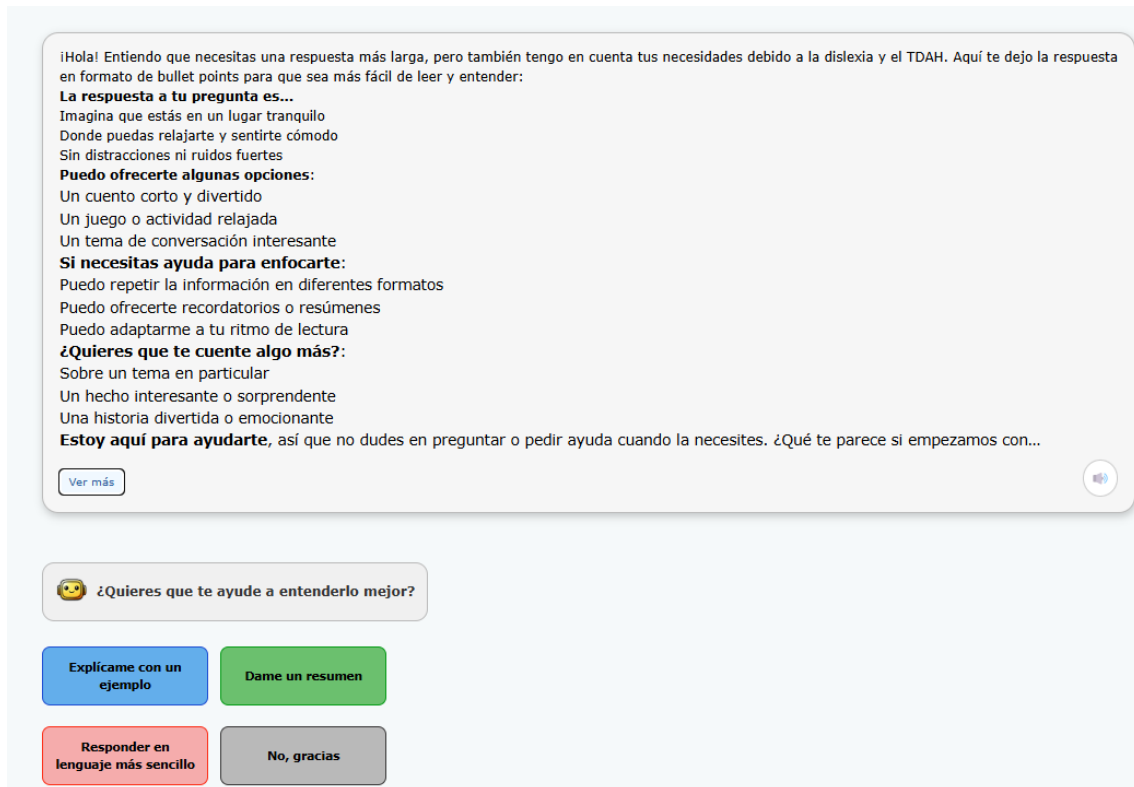


Figura 5.11: Texto de respuesta de IA generativa

opciones como “Dame un ejemplo”, “Resumir”, “Buscar sinónimos” o “Reformular en lenguaje más sencillo” (Figura 5.13). Al pulsar este último, se despliegan dos alternativas adicionales: una que permite reformular toda la respuesta generada de forma más simple, y otra que muestra un campo de texto para introducir palabras que no se comprenden, generando sinónimos y una definición breve para ellas (Figura 5.14). Además, el componente permite que el usuario escriba libremente su propia pregunta de seguimiento, utilizando `sendCustomPrompt()`, que construye un nuevo prompt personalizado.

Cuando el usuario pulsa el botón “No, gracias”, aparece una última interacción que pregunta “¿Te ha quedado todo claro?”, lo cual sirve como punto de cierre para la conversación. Si el usuario responde afirmativamente, se invoca la función `saveChatToHistory()`, encargada de almacenar la conversación actual en el historial. Esta función primero verifica si existe ya una conversación activa mediante el estado `activeChat`. En caso afirmativo, actualiza esa misma entrada del historial sobrescribiendo el contenido anterior con los últimos mensajes almacenados en `chatFlow`. Si no hay una conversación activa, se genera una nueva entrada en el historial con los siguientes elementos:

- Un título generado automáticamente a partir del contenido del chat (mediante la función `generateTitleFromChat()`).
- Una copia del flujo conversacional actual, contenido en `chatFlow`.
- La fecha y hora de guardado, generada dinámicamente mediante `new Da-`

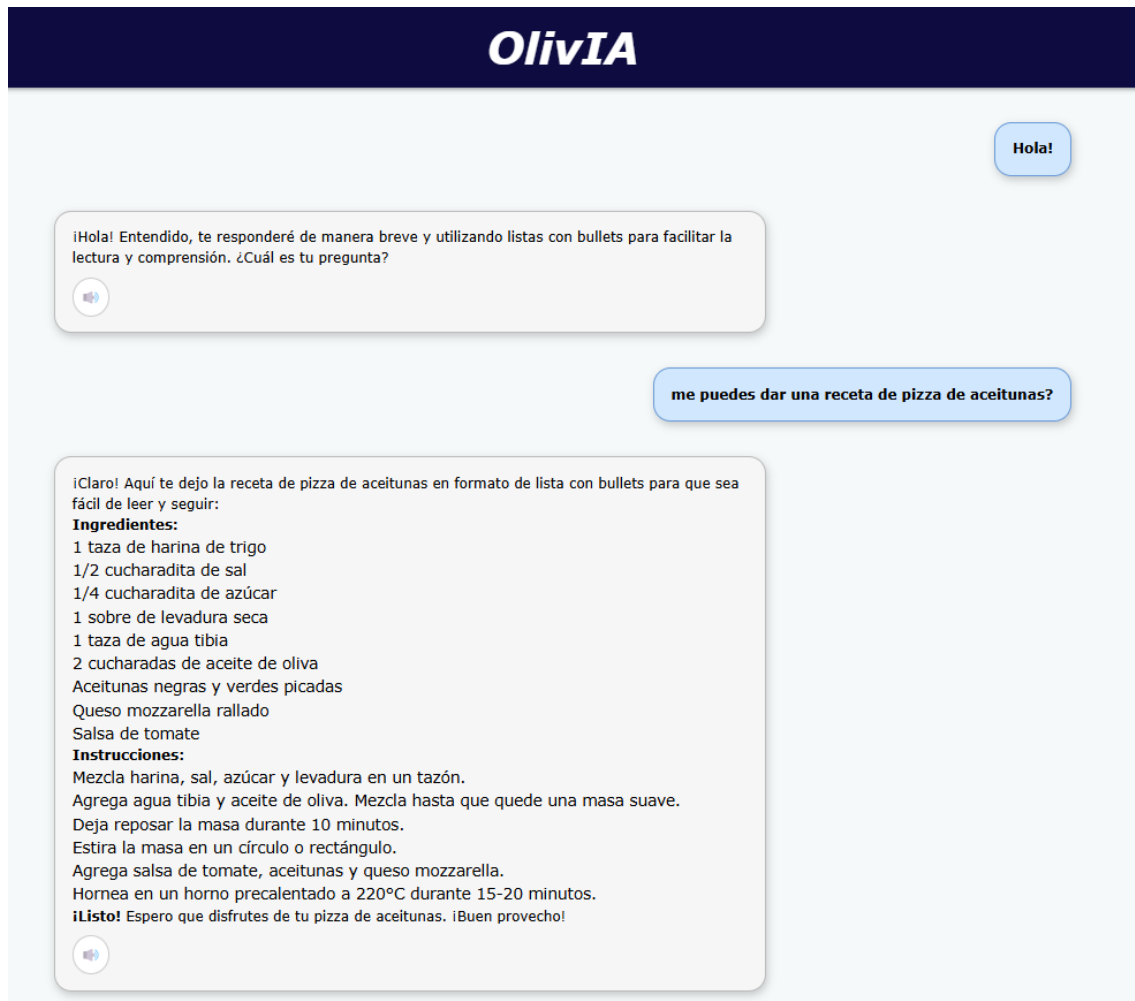


Figura 5.12: Ejemplo de Conversación generada por plantilla “Formular desde cero”

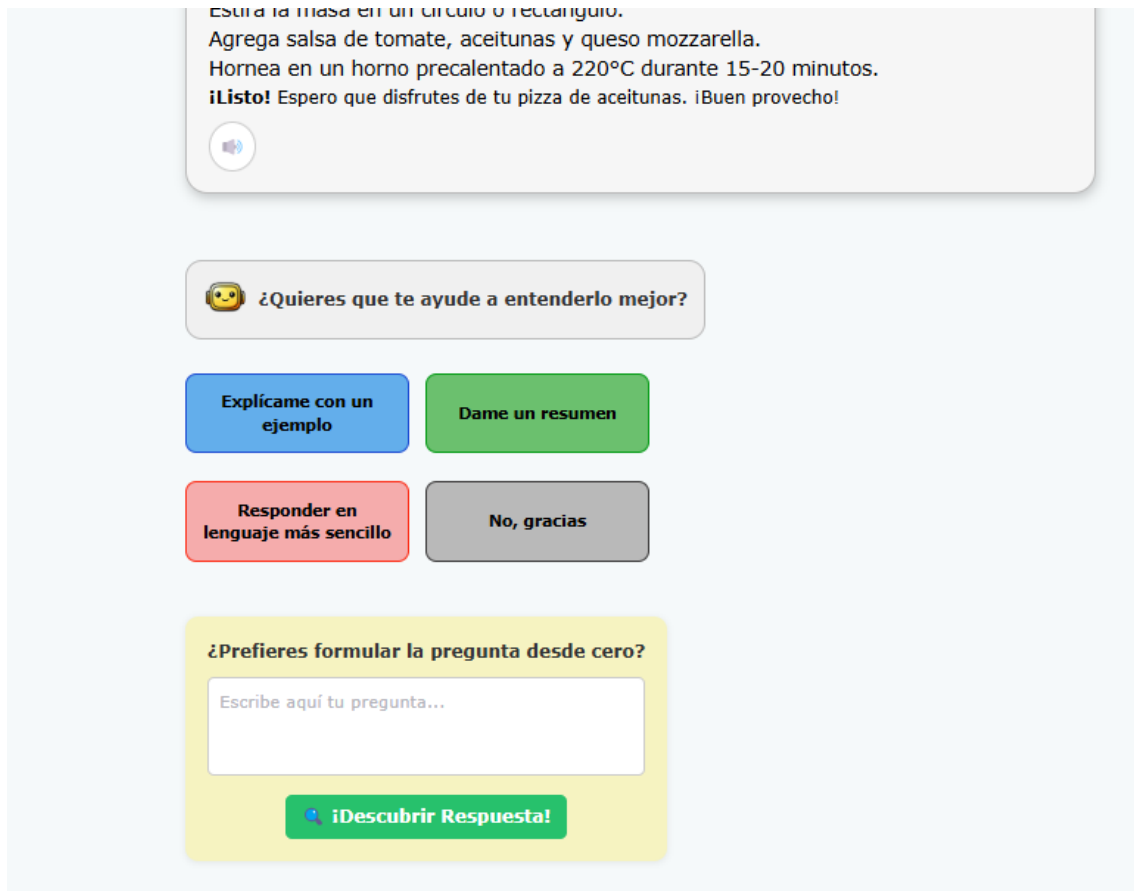


Figura 5.13: Botones Interactivos con la respuesta generada por la IA

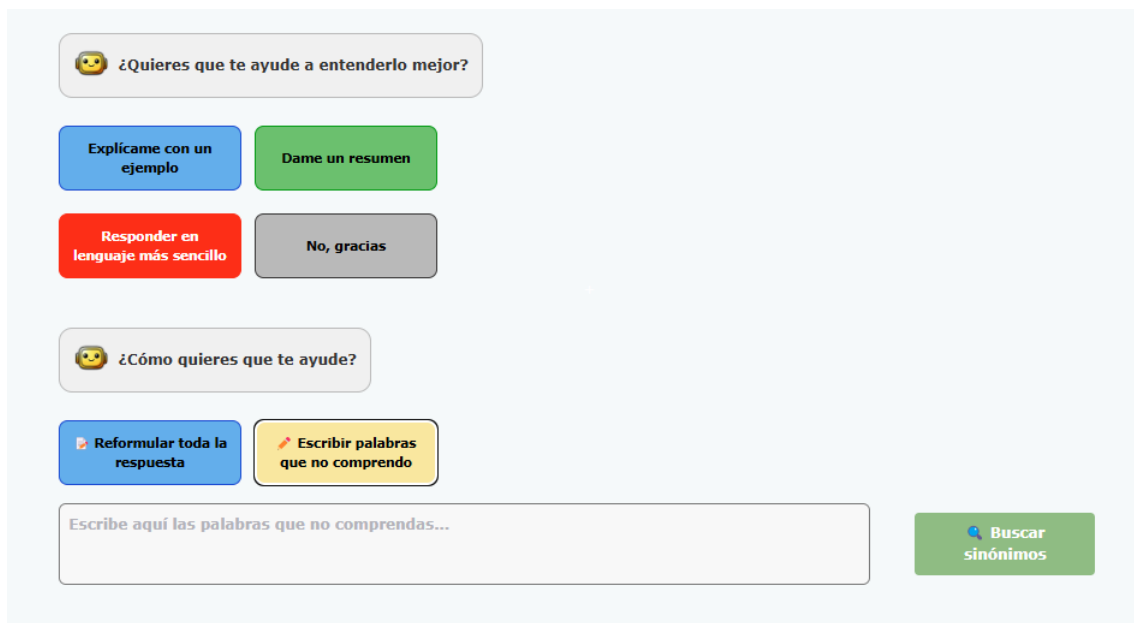


Figura 5.14: Botón de reformular pregunta



Figura 5.15: Historial de *OlivIA* con chats generados

`te().toLocaleString()`.

- Una marca que indica que se trata de una conversación reciente (flag *isNew*).

Una vez almacenado, se limpian los estados asociados a la sesión activa mediante una secuencia controlada: se resetea el flujo (`setChatFlow(//)`), se vacía la pregunta actual, se ocultan los botones de ayuda y se cierra la vista de conversación (`setShowChat(false)`) volviendo a la interfaz principal. Finalmente, se fuerza la visualización de la pestaña del historial (`setShowHistory(true)`), asegurando al usuario que ha sido guardado su chat y permitiendo acceder inmediatamente a dicha conversación. Este procedimiento garantiza la persistencia de los datos, evita duplicados y mantiene la aplicación en un estado listo para comenzar una nueva sesión sin interferencias de la anterior.

### 5.3.2.3. Historial

Paralelamente, la aplicación permite acceder al historial de conversaciones mediante un botón que despliega el componente `ChatHistory.jsx` (Figura 5.8, arriba a la izquierda). Este módulo presenta una barra lateral que muestra una lista de todas las sesiones previas guardadas. Cada sesión incluye un título (generado automáticamente por `generateTitleFromChat()`) y una marca temporal (Figura 5.15). Al pulsar sobre una entrada, se carga su contenido en `chatFlow` mediante la función `handleOpenChat()`, permitiendo al usuario reanudar la conversación desde donde la dejó. Asimismo, se puede cerrar la sesión activa y guardar los cambios mediante `handleCloseChat()`, que actualiza el historial sin perder el progreso. Este sistema asegura

que cada conversación sea persistente y reutilizable, ofreciendo así una experiencia continua y estructurada.

Este diseño modular y orientado a estados permite a *InterfazPrincipal.jsx* actuar como controlador central de la experiencia conversacional, distribuyendo el estado necesario a cada módulo y gestionando la interacción con el usuario de forma fluida, accesible y coherente con el perfil recogido inicialmente.

### 5.3.3. Prompts

El sector de *Prompts* agrupa los módulos responsables de construir los mensajes que se envían a los modelos de lenguaje y de gestionar su comunicación con servicios externos como *Groq* y *OpenAI*. Está formado por dos archivos principales: *Prompts.jsx*, que implementa el hook personalizado *usePromptFunctions*, y *apiFunctions.jsx*. Toda la lógica relacionada con la conexión a APIs externas se encuentra centralizada en el archivo *apiFunctions.jsx*, que define funciones genéricas como *fetchIA*, encargadas de realizar solicitudes *HTTP* mediante el método *POST*. A partir de esta función base, se han implementado funciones específicas como *fetchFromGroq*, que encapsulan detalles particulares del proveedor (como la URL, el modelo a utilizar o las cabeceras de autenticación) y se los pasan como parámetros a *fetchIA*. De este modo, la lógica de conexión permanece desacoplada y reutilizable, lo que facilita tanto el mantenimiento como la futura incorporación de nuevos servicios de inferencia. Estas funciones son utilizadas desde el módulo *Prompts.jsx*, donde se invocan de forma parametrizada según el tipo de petición que se desee realizar (resumen de la respuesta, reformulación, explicación con un ejemplo, etc.). Este enfoque modular permite alternar entre distintos modelos de IA generativa sin necesidad de modificar la estructura principal del código. Gracias a esta organización, añadir nuevas APIs o modelos en el futuro resulta sencillo, manteniendo la escalabilidad y limpieza del proyecto.

Las credenciales de acceso a las APIs se gestionan de manera segura mediante variables de entorno almacenadas en un archivo *.env*, siguiendo buenas prácticas de seguridad para evitar la exposición de datos sensibles en el repositorio público.

#### 5.3.3.1. Diseño modular y reutilización de lógica

Una de las decisiones clave fue centralizar toda la lógica de generación de prompts en el hook *usePromptFunctions*. Esto permite que las funciones de ayuda como “Resumir”, “Dar un ejemplo”, “Reformular” o “Buscar sinónimos” compartan una misma infraestructura interna, reutilizando la función principal *sendCustomPrompt()*. Esta función acepta como argumentos el contenido a enviar, un contexto textual que lo acompaña, un texto opcional para mostrar al usuario (*displayPrompt*) y una función *fetch* que determina qué API se usará. Este diseño favorece la extensibilidad y evita duplicación de código, ya que cualquier nueva funcionalidad basada en un prompt puede integrarse fácilmente.

### 5.3.3.2. Estructura del prompt

Cada mensaje enviado a la IA se construye con una estructura dividida en tres niveles:

- Texto base del usuario (*customPrompt*): Es la entrada principal generada por el usuario.
- *buildPrompt*: Es la encargada de formar el mensaje que se le manda a la IA por *apiPrompt*. Este proceso se compone de la unión de dos partes:
  - *displayPrompt*: Es la unión entre el texto base escrito por el usuario y, opcionalmente, el contexto necesario que pueda necesitar la IA para responder de manera coherente (el contexto es necesario cuando se parte de una plantilla o se interacciona con la respuesta a partir de un botón).
  - *personalInfo*: Es el bloque de información que se añade automáticamente al mensaje interno enviado al modelo para personalizar las respuestas según las necesidades del usuario. Este bloque se genera a partir del objeto *summary* e incluye tres partes:
    - *Discapacidad*: Lista las condiciones seleccionadas por el usuario, como “Déficit de atención” o “Dislexia”.
    - *Retos*: Incluye los elementos que el usuario ha marcado como difíciles de comprender, expresados como restricciones usando la frase “NO uses”, para que el modelo evite estructuras que podrían resultar confusas.
    - *Herramientas*: Enumera las herramientas seleccionadas para facilitar la comprensión, como “Frasas cortas” o “Ejemplos”, usando la expresión “Quiero que me generes la respuesta usando” para guiar al modelo hacia un estilo de respuesta preferido.
- *apiPrompt*: encargado que mandar el prompt final, creado por *buildPrompt*, desde *sendCustomPrompt* a a la IA correspondiente.

Gracias a esta arquitectura modular y parametrizable, *OlivIA* puede adaptarse fácilmente a distintas necesidades conversacionales y técnicas, manteniendo una separación clara entre la lógica del contenido, la personalización del perfil y la comunicación con modelos externos.

### 5.3.3.3. Funciones disponibles

El hook *usePromptFunctions* expone las siguientes funciones públicas para interactuar con la IA:

- *sendPrompt(prompt, selectedOption)*: envía el mensaje inicial generado desde una plantilla o introducido manualmente. Inserta automáticamente el mensaje de carga y gestiona la respuesta.

- *sendCustomPrompt(prompt, context, displayOverride, fetchFunction)*: función genérica que permite reutilizar la infraestructura de generación de prompts con cualquier tipo de contexto o proveedor de IA. Es la base sobre la que se construyen todas las funcionalidades auxiliares.
- *requestSummary()*: resume la última respuesta generada por la IA.
- *requestExample()*: genera un ejemplo relacionado con el contenido anterior.
- *requestSimplifiedResponse()*: reformula la última respuesta utilizando un lenguaje más accesible.
- *requestSynonyms(words)*: devuelve sinónimos y definiciones sencillas para las palabras introducidas por el usuario.
- *generateTitleFromChat()*: sintetiza un título corto representativo de la conversación actual, utilizado para nombrar chats en el historial.

#### 5.3.3.4. Comunicación con APIs externas

Las llamadas a modelos de lenguaje se realizan a través de funciones de conexión definidas en *apiFunctions.jsx*. Actualmente, se utiliza *fetchFromGroq()*, que conecta con el modelo *llama-4-scout-17b-16e-instruct* alojado por Groq. Internamente, esta función delega en *fetchIA()*, una función genérica parametrizable que permite reutilizar la misma estructura de petición para diferentes proveedores. Sus parámetros permiten definir:

- *url*: dirección del *endpoint* de la API.
- *model*: identificador del modelo a utilizar.
- *apiKey*: clave del proveedor almacenada en variables de entorno.
- *messages*: mensaje que se manda a la API o mensaje de error que devuelve dependiendo del proveedor.
- *headers* y *temperature*: configuraciones adicionales opcionales.

Este diseño permite añadir nuevos proveedores (por ejemplo, OpenAI) de forma inmediata simplemente declarando una nueva función (como *fetchFromOpenAI*) y pasándola como argumento a *sendCustomPrompt()*. De este modo, se puede personalizar la elección del modelo según el contexto, sin modificar el flujo general de la aplicación.

#### 5.3.3.5. Ejemplo de flujo completo

Cuando el usuario pulsa “Explica palabras que no entiendo” y escribe, por ejemplo, “Felicidad”, se ejecuta el siguiente flujo:

1. Llamada a la función *requestSynonyms*: Se llama a la función *requestSynonyms* con el texto introducido por el usuario (“Felicidad”). Esta función verifica que el texto no esté vacío y, si es válido, construye un *prompt* personalizado para buscar sinónimos usando:

```
requestSynonyms(“Felicidad”);
```

2. Generación del Prompt Personalizado: Dentro de *requestSynonyms*, se llama a *sendCustomPrompt* para construir el prompt personalizado con los siguientes parámetros:

- Texto principal (*customPrompt*): Texto introducido por el usuario, como “Felicidad”.
- Contexto (*context*): Texto adicional que se añade para dar más contexto a la IA y evitar que solo use el texto principal. Por ejemplo: “Dame un sinónimo y una definición corta y muy sencilla de”.
- Mensaje visible (*displayOverride*): Lo que se muestra en la interfaz para el usuario, en lugar del texto completo del prompt. En este caso: “Dame sinónimos de Felicidad”.
- Función de envío (*fetchFunction*): elige la API que se utiliza para enviar el prompt al modelo. Para este ejemplo, se usa *fetchFromGroq*.

Se mandaría como: *sendCustomPrompt*(“Felicidad”, “Dame un sinónimo y una definición corta y muy sencilla de”, “Dame sinónimos de Felicidad”, *fetchFromGroq*);

3. Construcción del *apiPrompt*: Dentro de *sendCustomPrompt*, se llama a *buildPrompt*, que se encarga de construir el mensaje que se enviará a la IA incluyendo la personalización seleccionada en el cuestionario:

```
const apiPrompt = buildPrompt(“Dame un sinónimo y una definición corta y muy sencilla de Felicidad”);
```

Dentro de *buildPrompt*, se combinan dos elementos principales:

- Texto que recibe por parámetro (*displayPrompt*): “Dame un sinónimo y una definición corta y muy sencilla de Felicidad” que se genera de unir el contexto creado por la función y el texto principal introducido por el usuario.
- Instrucciones personalizadas (*const personalInfo*) basadas en el perfil del usuario (*summary*):

Pero ten en cuenta estas instrucciones al responder:

- Mi Discapacidad: TDAH, Dislexia
- NO uses: Frases largas
- Quiero que me generes la respuesta usando: Bullets

El resultado final es un objeto con dos propiedades:

- *displayPrompt*: “Dame un sinónimo y una definición corta y muy sencilla de Felicidad”.
  - *apiPrompt*: Hola, respóndeme a esta pregunta: *displayPrompt*, *personalInfo*, es decir, “Hola, respóndeme a esta pregunta: Dame un sinónimo y una definición corta y muy sencilla de Felicidad Pero ten en cuenta estas instrucciones al responder: - Mi Discapacidad: TDAH, Dislexia - NO uses: Frases largas - Quiero que me generes la respuesta usando: Bullets.
4. Envío del Prompt al Modelo: *sendCustomPrompt* utiliza *fetchFunction* para enviar este *apiPrompt* al modelo de lenguaje seleccionado:

```
const response = await fetchFunction([ role: “user”, content: apiPrompt ]);
```

5. Actualización del Historial del Chat: Finalmente, el mensaje del usuario y la respuesta de la IA se añaden al flujo del chat, actualizando la interfaz:

```
setChatFlow((prev) =>[...prev, type: “user”, content: “Dame sinónimos de Felicidad” , type: “ai”, content: response ]);
```

6. Resultado en la Interfaz: El usuario verá su mensaje mandado como “Dame sinónimos de Felicidad” y, por otra parte, la respuesta generada por la IA, por ejemplo:
- “- Sinónimos: Alegría, dicha, satisfacción.
  - Definición: Son palabras que significan sentimientos positivos y agradables.”



## Conclusiones y Trabajo Futuro

En la Sección 6.1 se explican las conclusiones a las que se ha llegado tras realizar el proyecto y en la Sección 6.2 se describen las posibles mejoras que se podrían llevar a cabo en la aplicación.

### 6.1. Conclusiones

En este Trabajo de Fin de Grado, se planteó como objetivo principal el desarrollo de una aplicación web diseñada para facilitar el uso de tecnologías de inteligencia artificial generativa a personas con discapacidades cognitivas. Para cumplir con este objetivo, se llevó a cabo un análisis de requisitos, que incluyó una prueba práctica con usuarios, la evaluación de una aplicación de inteligencia artificial generativa existente con dos herramientas que miden la accesibilidad y la revisión de estándares de accesibilidad como WCAG y COGA. Este análisis permitió identificar las necesidades específicas del colectivo objetivo, sirviendo como base para definir los requisitos clave de la aplicación. Del conjunto completo, se detallan a continuación los requisitos a los que se ha dado prioridad en el presente trabajo para garantizar que la aplicación sea útil, accesible y funcional desde su primera versión:

- Requisitos dentro de “Cuestionario de personalización inicial”: Se ha desarrollado un sistema de configuración inicial que permite a los usuarios especificar sus preferencias y necesidades. Este cuestionario recopila información sobre dificultades cognitivas, preferencias de comunicación (como textos cortos, uso de ejemplos y frases simplificadas) y cuestiones a evitar (como textos largos, palabras difíciles y usar memoria). Toda esta información se almacena en un perfil centralizado (*summary*) que se utiliza para personalizar dinámicamente las respuestas generadas por los modelos de lenguaje, mejorando significativamente la experiencia del usuario al adaptar las respuestas a sus necesidades específicas.
- Requisitos dentro de “Generación de texto”: Se han incorporado botones predefinidos que llevan al uso de plantillas para facilitar la formulación de preguntas, incluyendo opciones como “Resumir...”, “Dame un ejemplo de...”, “Dame un de

sinónimo...”, “¿Cómo se hace...?” y “¿Qué significa...?”. Estas herramientas ayudan a reducir la carga cognitiva durante la creación de consultas, guiando al usuario y minimizando la necesidad de recordar estructuras complejas de preguntas.

- Requisitos dentro de “Generación de respuestas”: Se han incluido opciones para interactuar con las respuestas generadas, con botones que permiten dar simplificaciones adicionales, ejemplos y resúmenes y la opción de escribir la propia consulta.
- Interfaz sencilla y personalizable: La interfaz de *OlivIA* es sencilla, fácil de navegar y libre de distracciones, utilizando botones claramente etiquetados de colores primarios y con contraste y usando un tipo de fuente accesible.
- Lenguaje simplificado y contenido comprensible: Las respuestas son claras y concisas, evitando tecnicismos innecesarios. Además, se incluyen opciones para ver las definiciones contextuales para términos técnicos y para reformular respuestas complejas, haciendo que la información sea más comprensible para los usuarios con discapacidades cognitivas.
- Respuestas segmentadas y progresivas: La aplicación presenta las respuestas de manera progresiva si son muy largas mediante un botón de “Ver más”, permitiendo al usuario controlar cómo se muestra la información y evitando sobrecarga de información.
- Fomento de la autonomía y confianza del usuario: La aplicación guía al usuario a través de explicaciones claras, permitiendo el uso de plantillas o botones para ayudar a responder a la pregunta generada y preguntando si la respuesta fue comprendida pero dándole margen al usuario de poder partir desde cero si lo prefiere empoderando al usuario para que controle su experiencia.
- Diseño para todos: *OlivIA* presenta una interfaz simplificada y la información en formatos fáciles de procesar, garantizando una experiencia accesible para todos los usuarios.

Además, al comienzo del TFG, en la Sección 1.2, se plantearon varios objetivos académicos que también se han alcanzado:

- Aplicar los conocimientos adquiridos durante el grado: En este proyecto se han aplicado conceptos fundamentales aprendidos en varias asignaturas, incluyendo diseño de interfaces, ingeniería de software y programación de aplicaciones web. Esto ha permitido integrar teoría y práctica en un contexto real, desarrollando una solución tecnológica con un enfoque claro en accesibilidad.
- Aprender y experimentar con herramientas de IA generativa: A lo largo del proyecto, se ha trabajado en la integración de modelos de lenguaje generativo mediante el uso de APIs, aprendiendo a crear y ajustar prompts para personalizar las respuestas y adaptar la interacción según las necesidades cognitivas de los usuarios. Esto incluyó explorar cómo construir prompts efectivos, manejar

contextos para mejorar la precisión de las respuestas y optimizar el flujo de comunicación entre la interfaz y los modelos de IA.

- Adquirir habilidades en diseño de interfaces y desarrollo de aplicaciones accesibles: Se han adquirido nuevas habilidades en diseño centrado en el usuario, desarrollo de interfaces y accesibilidad cognitiva, contribuyendo al crecimiento personal y profesional en el ámbito del desarrollo de software inclusivo.
- Adaptar soluciones tecnológicas para personas con discapacidades cognitivas: El diseño de *OlivIA* se ha centrado en hacer la tecnología accesible para personas con discapacidades cognitivas, implementando estrategias como el uso de lenguaje claro, simplificación de contenido y personalización dinámica.
- Reflexionar sobre la responsabilidad del ingeniero en la creación de soluciones accesibles: Durante el desarrollo de *OlivIA*, se ha mantenido un enfoque constante en la responsabilidad ética del ingeniero en la creación de tecnologías inclusivas. Esto incluye no solo hacer que la tecnología sea accesible, sino también diseñar interfaces que respeten la autonomía de las personas con discapacidades cognitivas, asegurando que puedan interactuar de manera independiente y efectiva. Además, se ha considerado la importancia de crear soluciones tecnológicas que promuevan la inclusión digital y reduzcan las barreras de acceso a la información.

## 6.2. Trabajo Futuro

Después de desarrollar el proyecto y cumplir con los objetivos y la mayoría de los requisitos, es inevitable que queden algunas tareas pendientes para posible trabajo futuro. Desde el punto de vista de los requisitos, algunas funciones avanzadas que podrían mejorar aún más la accesibilidad y flexibilidad del sistema se han dejado para futuras versiones con el objetivo de mantener un enfoque claro en las necesidades más críticas del usuario en esta etapa inicial. Específicamente, estos requisitos son:

- Requisitos dentro de “Tutorial interactivo (explicación inicial)”: Aunque para esta primera versión, la interfaz sea sencilla, no se ha implementado un tutorial que el usuario pueda consultar en caso de duda o que pueda guiarle sobre como funciona la aplicación web para facilitar la interacción inicial. También faltan modos de prueba controlados para que los usuarios puedan practicar sin presión, así como explicaciones en formatos multimedia como texto corto, audio, videos o GIFs para mejorar la comprensión.
- Modalidades múltiples de comunicación: El sistema aún no incluye soporte para respuestas en formatos alternativos como pictogramas, contenido auditivo o gráficos, lo que podría ampliar significativamente su accesibilidad para usuarios con diferentes preferencias y necesidades.
- Control del tiempo y ritmo de interacción: Aunque se ha incorporado un mecanismo que facilita la gestión del ritmo de respuesta mediante un botón de

“Ver más”, el sistema no tiene actualmente implementado controles avanzados para pausar, reanudar o ajustar la velocidad de la generación de las respuestas en tiempo real.

- **Compatibilidad con tecnologías de apoyo:** Aunque el sistema está diseñado para ser compatible con lectores de pantalla, la integración completa con tecnologías de asistencia, como comandos de voz, navegación por teclado y dispositivos de entrada alternativos, aún no se ha completado, limitando su accesibilidad en entornos de uso más variados.
- **Modo “Asistente de Preguntas” (guía paso a paso):** Aunque se han añadido guías básicas para estructurar consultas, aún falta un sistema más robusto que permita a los usuarios navegar de manera asistida a través de procesos complejos de consulta.
- **Fomento de la autonomía y confianza del usuario:** Para futuras versiones, se podría considerar incluir en la aplicación retroalimentación positiva de forma activa.

Además, durante el proceso de desarrollo de la aplicación web, surgieron nuevas ideas para futuras mejoras que podrían tenerse en cuenta en próximas versiones:

- **Realizar pruebas de usabilidad con usuarios reales** para validar que las adaptaciones implementadas realmente facilitan la interacción para personas con discapacidad cognitiva. Esta es una de las mejoras más importantes, ya que permitiría identificar obstáculos específicos y nuevas necesidades que solo pueden ser detectadas a través de la observación directa y el análisis de experiencias reales. Esto no solo optimizaría la interfaz para hacerla más accesible, sino que también revelaría oportunidades para añadir funcionalidades que mejoren significativamente la experiencia del usuario.
- **Integrar más modelos de lenguaje** para ofrecer respuestas más precisas y adaptadas a diferentes perfiles de usuario, mejorando tanto la calidad como la precisión de las respuestas generadas.
- **Entrenar y probar diferentes modelos de lenguaje** para identificar cuáles responden mejor a distintos tipos de consultas, con el objetivo de optimizar la precisión y adaptabilidad de las respuestas. Esto podría incluir el uso de modelos especializados para tareas específicas, como generar resúmenes, proporcionar ejemplos o simplificar explicaciones, utilizando diferentes APIs según la naturaleza de la pregunta.
- **Implementar un sistema de retroalimentación proactiva** para evaluar continuamente si las respuestas son claras, útiles y adecuadas para cada usuario, promoviendo una experiencia más interactiva y personalizada.
- **Ampliar las opciones de personalización** para mejorar la accesibilidad, incluyendo configuraciones para ajustar el tamaño de letra, el contraste y otros aspectos visuales, permitiendo a los usuarios adaptar la interfaz a sus preferencias y necesidades específicas.

- Evaluar y mejorar continuamente la accesibilidad de la aplicación utilizando herramientas avanzadas como *Pa11y* y otras mencionadas en la Sección 2.2.2. Aunque durante el desarrollo se utilizó *WAVE* para medir la accesibilidad en tiempo real, el uso de herramientas adicionales permitirá identificar áreas de mejora más detalladas y asegurar que la aplicación cumpla con los estándares más rigurosos.
- Optimizar la eficiencia del código para reducir los tiempos de carga y mejorar el rendimiento en dispositivos móviles, garantizando una experiencia de uso fluida y rápida.

Con estas mejoras, se espera que *OlivIA* continúe evolucionando como una herramienta innovadora para personas con discapacidades cognitivas, promoviendo una interacción más inclusiva, precisa y efectiva con tecnologías de inteligencia artificial generativa.



# Conclusions and Future Work

In Section 6.3, the conclusions drawn after completing the project are explained, and in Section 6.4, the potential improvements and future directions for the application are described.

## 6.3. Conclusions

The main objective of this Final Degree Project was to develop a web application designed to facilitate the use of generative artificial intelligence technologies for people with cognitive disabilities. To achieve this goal, a requirements analysis was carried out, including practical user testing, the evaluation of an existing generative AI application using two accessibility tools, and a review of accessibility standards such as WCAG and COGA. This analysis helped identify the specific needs of the target audience, which served as the foundation for defining the application's key requirements. Among the full set of requirements, the following were prioritized to ensure that the application would be useful, accessible, and functional in its first version:

- Requirements within the “Initial Personalization Questionnaire”: An initial setup system was developed that allows users to specify their preferences and needs. This questionnaire collects information on cognitive difficulties, communication preferences (such as short texts, examples, and simplified phrases), and things to avoid (such as long texts, difficult words, and memory usage). All of this data is stored in a centralized profile (*summary*) and used to dynamically personalize the responses generated by the language models, significantly enhancing the user experience by tailoring outputs to their specific needs.
- Requirements within “Text Generation”: Predefined buttons using templates have been included to help users formulate questions more easily. These include options such as “Summarize...”, “Give me an example of...”, “Give me a synonym of...”, “How do you...?” and “What does ... mean?”. These features reduce cognitive load during question formulation by guiding users and minimizing the need to recall complex sentence structures.
- Requirements within “Response Generation”: Several interaction options were included, such as buttons for further simplifications, examples, summaries,

and the possibility of writing custom queries.

- Simple and customizable interface: The interface of *OlivIA* is simple, easy to navigate, and free of distractions, using clearly labeled buttons in high-contrast primary colors and an accessible font.
- Simplified language and understandable content: The responses are clear and concise, avoiding unnecessary jargon. Contextual definitions for technical terms and reformulations for complex responses are also included, making content more understandable to users with cognitive disabilities.
- Segmented and progressive responses: When responses are long, they are displayed progressively using a “Show more” button, allowing users to control how much information is displayed at once and preventing information overload.
- Promoting autonomy and user confidence: The application guides users through clear explanations and helpful templates or buttons to support question generation. It also checks for understanding while allowing users to start from scratch, thus empowering them to control their experience.
- Design for all: *OlivIA* presents a simplified interface and provides information in easy-to-process formats, ensuring an accessible experience for all users.

Additionally, several academic objectives outlined at the start of the project in Section 1.6 were achieved:

- Applying knowledge acquired during the degree: The project applied fundamental concepts learned in various subjects, including interface design, software engineering, and web development. This allowed for the integration of theory and practice in a real-world setting, resulting in a technology solution with a strong focus on accessibility.
- Learning and experimenting with generative AI tools: The project involved integrating language models through APIs, learning to craft and refine prompts to personalize responses and adapt interactions to users’ cognitive needs. This included exploring how to build effective prompts, manage context to improve accuracy, and optimize the communication flow between the interface and the AI models.
- Gaining skills in interface design and accessible development: New skills were acquired in user-centered design, interface development, and cognitive accessibility, contributing to both personal and professional development in inclusive software creation.
- Adapting technological solutions for people with cognitive disabilities: The design of *OlivIA* focused on making technology accessible for people with cognitive impairments, implementing strategies such as plain language, content simplification, and dynamic personalization.

- Reflecting on the engineer’s responsibility in creating accessible solutions: A continuous ethical focus was maintained throughout the project on the role of engineers in developing inclusive technologies. This involves not only ensuring technical accessibility but also designing interfaces that respect the autonomy of users with cognitive disabilities, allowing them to interact independently and effectively. The importance of digital inclusion and reducing access barriers to information was also emphasized.

## 6.4. Future Work

After completing the project and fulfilling the main objectives and requirements, several tasks remain as opportunities for future development. From the perspective of requirements, some advanced features that could significantly improve the system’s accessibility and flexibility were intentionally left for future versions, in order to focus on the most critical user needs during the initial phase. These include:

- Requirements within “Interactive Tutorial (Initial Explanation)”: While the current interface is simple, an interactive tutorial for helping users understand the web application has not yet been implemented. Also missing are guided testing modes for low-pressure exploration and multimedia explanations via short texts, audio, videos, or GIFs.
- Multiple communication modalities: The system currently lacks support for alternative formats such as pictograms, audio responses, or graphical content, which could greatly enhance accessibility for users with different needs and preferences.
- Time and pacing control: Although a “Show more” button allows users to manage response length, there are no advanced controls yet for pausing, resuming, or adjusting the speed of answer delivery in real time.
- Compatibility with assistive technologies: While designed to be compatible with screen readers, full integration with assistive technologies such as voice commands, keyboard navigation, and alternative input devices has yet to be completed, limiting its use in more diverse contexts.
- “Question Assistant” mode (step-by-step guide): Although basic templates have been included, a more advanced system is still needed to guide users step-by-step through the process of asking complex queries.
- Promoting autonomy and confidence: Future versions could include active positive feedback mechanisms to reinforce user confidence.

Additional improvement ideas that emerged during the development process include:

- Conduct usability testing with real users to validate whether the adaptations implemented truly facilitate interaction for people with cognitive disabilities.

This would help identify specific barriers and new needs that can only be revealed through direct user observation and feedback.

- Integrate additional language models to offer more accurate, tailored responses for different user profiles, improving the quality and relevance of the generated content.
- Train and test different models to determine which are best suited to various types of queries. This may involve using specialized models for summarization, example generation, or simplification, and switching between APIs based on query type.
- Implement a proactive feedback system to continuously assess whether responses are clear, useful, and appropriate for each user, fostering a more interactive and personalized experience.
- Expand personalization options to improve accessibility, such as font size, contrast, and visual display settings, allowing users to tailor the interface to their individual needs.
- Continuously evaluate and enhance the application's accessibility using advanced tools such as *Pa11y*, alongside those already used (e.g., *WAVE*), to identify detailed areas for improvement and ensure compliance with the strictest standards.
- Optimize code efficiency to reduce loading times and improve mobile device performance, ensuring a smooth and responsive user experience.

With these enhancements, *OlivIA* is expected to continue evolving as an innovative tool for individuals with cognitive disabilities, promoting more inclusive, accurate, and effective interaction with generative AI technologies.

# Bibliografía

- AMERICAN PSYCHIATRIC ASSOCIATION. *Diagnostic and Statistical Manual of Mental Disorders (DSM-5)*. American Psychiatric Publishing, Arlington, VA, 2013. Citado como APA.
- NIELSEN, J. AI: First New UI Paradigm in 60 Years. *NN group*, 2023.
- PLENA INCLUSIÓN MADRID. *Guía de pautas de accesibilidad cognitiva web*, 2020. Último acceso: 15 de febrero de 2025.
- ROSNER, D. K. y TREWIN, S. The promise of empathy: Design, disability, and knowing the “other”. En *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, páginas 1–13. ACM, New York, NY, USA, 2019.
- ZHU, Q. y LUO, J. Toward artificial empathy for human-centered design: A framework. 2023.



## Cuestionario pre-test para estudiantes

1. ¿Te gusta pedir ayuda o prefieres hacer las cosas solo?
2. ¿Sueles hacer preguntas si algo no te queda claro?
3. ¿Qué haces cuando no entiendes la respuesta a lo que has preguntado?
4. ¿Con qué frecuencia usas tecnología (computadora, teléfono, tablet)?
5. ¿Te sientes cómodo usando nuevas aplicaciones o programas?
6. ¿Qué tecnología usas más?
7. ¿Te resulta fácil buscar información en internet?
8. ¿Te fías siempre de la información que encuentras en internet?
9. ¿Qué pasos sigues cuando quieres saber si algo que lees es verdad?
10. ¿A quién pedirías ayuda si necesitas saber si algo de internet es verdad?
11. ¿Has escuchado hablar de la Inteligencia Artificial?
12. ¿Podrías describirla con tus palabras?
13. ¿Conoces alguna herramienta de IA? ¿Cuáles?
14. ¿Las has usado alguna vez? ¿Para qué las usaste?
15. ¿Conoces ChatGPT? ¿Qué es y para qué sirve ChatGPT?
16. ¿Qué esperas aprender con esta actividad?
17. ¿Hay algo específico que te preocupe o te parezca difícil sobre esta actividad?
18. ¿Crees que la tecnología puede ayudarte a hacer las tareas de la universidad?



# Apéndice **B**

## Cuestionario post-test para estudiantes

1. ¿Te gustó usar ChatGPT?
  - Me gustó mucho
  - Me gustó
  - No me gustó demasiado
  - No me gustó nada
2. ¿Te pareció útil ChatGPT para las actividades que hicimos?
  - Sí, fue muy útil
  - Fue algo útil
  - No fue muy útil
  - No fue nada útil
3. ¿Qué fue lo que más te gustó de ChatGPT?
4. ¿Qué fue lo que menos te gustó de ChatGPT?
5. ¿Tuviste alguna dificultad al hacer preguntas a ChatGPT?
  - Sí
  - No
    - Si la respuesta es sí, ¿qué dificultades tuviste?
6. ¿Pudiste comprobar si la información era correcta?
  - Sí
  - No
    - ¿Cómo trataste de comprobar si la información era correcta?
7. ¿Entendiste bien las respuestas que te dio ChatGPT?
  - Sí, siempre

- A veces
  - No, casi nunca
  - No, nunca
8. ¿Pudiste usar la información que te dio ChatGPT para hacer las tareas?
- Sí, siempre
  - Sí, en casi todas
  - Solo en algunas
  - No, en ninguna
9. ¿Hubo algo que no entendieras de las respuestas?
- Sí
  - No
    - ¿Cómo lo resolviste?
10. ¿Sientes que podrías usar ChatGPT para tareas escolares en el futuro sin ayuda?
- Sí
  - Sí pero con algo de ayuda
  - No, necesitaría ayuda todo el rato
11. ¿Crees que ChatGPT puede darte respuestas útiles?
- Sí
  - A veces
  - No
12. ¿Te gustaría seguir usando ChatGPT para aprender?
- Sí
  - No
  - No estoy seguro
13. ¿Te sientes más confiado usando tecnología después de esta actividad?
- Sí
  - No
  - No estoy seguro
14. ¿Crees que ChatGPT necesita mejoras para ser más fácil de entender?
- Sí
  - No
    - ¿Qué mejoras propondrías?

## Cuestionario pre-test para cada estudiante a rellenar por la tutora ACCEDE

1. Nombre del estudiante:
2. Apellido del estudiante:
3. Edad del estudiante:
4. Sexo del estudiante:
5. Tipo de discapacidad:
  - Auditiva
  - Visual
  - Motora
  - Cognitiva
6. Nivel de discapacidad:
7. Principales áreas de dificultad cognitiva:
  - Comprensión lectora
  - Memoria a corto plazo
  - Memoria a largo plazo
  - Atención:
    - Atención sostenida
    - Atención selectiva
    - Atención alternante
  - Expresión oral
  - Expresión escrita
  - Razonamiento lógico

- Funciones ejecutivas:
    - Inhibición
    - Flexibilidad Cognitiva
    - Memoria de trabajo
  - Otras (Por favor, añade aquí cualquier otra área de dificultad que considere relevante):
8. Nivel de Autonomía con Tecnología:
- Muy autónomo
  - Necesita poca ayuda
  - Necesita ayuda moderada
  - Necesita ayuda constante
9. Indica cualquier otra cuestión que creas que debemos tener en cuenta para este estudiante:

## Cuestionario post-test para cada estudiante a rellenar por las tutoras que se le supervisaron durante la actividad

1. ¿El estudiante necesitó mucha ayuda para formular preguntas adecuadas?
  - Sí, mucha ayuda
  - Ayuda moderada
  - Poca ayuda
  - Ninguna ayuda
2. ¿El estudiante entendió las respuestas de ChatGPT?
  - Sí
  - Con dificultad
  - No
3. ¿El estudiante pudo seguir los pasos para verificar la información?
  - Sí, lo hizo solo
  - Necesitó ayuda
  - No lo logró
4. ¿El estudiante mostró frustración o desmotivación en algún momento?
  - Sí, constantemente
  - Sí, en alguna ocasión
  - Sí, en un momento muy puntual
  - No
5. ¿El estudiante parecía disfrutar durante la actividad?

- Sí
  - Moderadamente
  - No
6. ¿El estudiante parecía aprender durante la actividad?
- Sí
  - Moderadamente
  - No
7. ¿Pudiste observar mejoras en la autonomía del estudiante al usar ChatGPT durante la actividad?
- Sí
  - Un poco
  - No
8. ¿Qué tipo de dificultades tuvo el estudiante?

