



**RehabilitIA: AI-Enhanced Speech-Language
Therapy Through a Web and Mobile Platform
for Personalized Aphasia Rehabilitation in
Spanish-Speaking Patients**

Santiago Navarrete Varela
Andrea Lucia Galindo Cera

August 31, 2025

This thesis is submitted in partial fulfillment of the requirements for the
undergraduate degree in *Systems and Computing Engineering*.

Advisor:
Prof. Ruben Francisco Manrique
Universidad de los Andes, Bogotá, Colombia

Department of Systems and Computing Engineering
Universidad de los Andes
Bogotá, Colombia

1 Introduction

1.1 Context

Aphasia is an acquired language disorder that affects the ability to speak, understand, read, and write, most commonly due to focal brain damage. It represents a major challenge not only for patients but also for their families and healthcare providers, since communication is central to daily life and social participation [1, 2]. Speech and language therapy is the primary intervention to support recovery [3], yet the availability of structured, accessible, and language-specific resources remains limited, especially in non-English-speaking contexts [4].

At the same time, rapid progress in Artificial Intelligence (AI) and Natural Language Processing (NLP) has led to the development of Large Language Models (LLMs), which are capable of generating and understanding human language with remarkable fluency. These models are increasingly explored in healthcare applications, offering new possibilities for personalization, scalability, and interactivity in therapy [5]. In parallel, the concept of agents, intelligent entities capable of perceiving their environment and influencing it through actions [6], has opened the door to building applications that can adapt dynamically to user needs and guide multi-step tasks.

In the field of speech-language therapy, these technological advances create an opportunity to design novel systems that combine clinical approaches with AI-driven support [7]. Evidence-based therapeutic methods such as *Verb Network Strengthening Treatment (VNest)* and *Spaced Retrieval (SR)* have shown promising results in improving communication and memory functions in people with aphasia [8][9]. However, most existing digital tools and AI resources are developed in English [4], leaving a gap for Spanish-speaking populations who face limited access to tailored digital support.

This context motivates the present work: the design and development of a web and mobile system that integrates LLMs and agentic workflows to support the creation and delivery of therapy exercises for Spanish-speaking individuals with aphasia. By situating the project at the intersection of clinical practice and AI innovation, the goal is to contribute both to the field of speech-language therapy and to the broader effort of extending the benefits of LLM technology beyond English-centered applications.

1.2 Problem Statement

Despite the recognized effectiveness of evidence-based interventions such as *Verb Network Strengthening Treatment (VNest)* and *Spaced Retrieval (SR)* [8, 9], their implementation in practice is still limited by several critical barriers. Traditional therapy delivery depends heavily on direct interaction with a speech-language pathologist [10], which often restricts the intensity and frequency of treatment sessions. For many patients, especially in low-resource or rural areas, this results in inconsistent therapy, slower recovery, and greater emotional and social strain. [10]

Existing digital solutions have the potential to expand access to therapy beyond the clinical setting, but they fall short in two fundamental ways. First, most available applications and resources are developed in English, leaving Spanish-speaking populations underserved [11]. This linguistic and cultural gap significantly reduces the availability of appropriate digital tools for millions of potential users. Second, many of the tools that do exist are static, providing only pre-defined exercises with limited adaptability. They often lack mechanisms for personalization, ongoing monitoring, or integration with therapist feedback, which are essential for maintaining motivation and ensuring clinical effectiveness.

This situation leaves Spanish-speaking individuals with aphasia at a clear disadvantage: they face reduced access to continuous therapy, limited opportunities for practicing evidence-based methods in their own language, and few platforms that allow therapists to oversee and adapt treatment plans [11]. Without solutions that address these gaps, therapy outcomes are compromised, and the potential of recent technological advances in artificial intelligence remains underutilized in this domain.

1.3 Justification

Aphasia is a prevalent and debilitating condition, often resulting from a cerebrovascular accident (CVA). Globally, 25–40% of stroke survivors develop aphasia, facing difficulties in speaking, understanding, reading, and writing [12]. In Colombia, stroke is the second leading cause of death and the main cause of adult disability, with 16,946 deaths reported in 2023 and many survivors experiencing long-term impairments [13]. Extrapolating from these figures, several thousand Colombians are likely affected by aphasia, highlighting a substantial population in need of speech-language therapy.

Speech-language therapy is the most effective intervention for improving communication in people with aphasia, yet access is often limited due to geographic, economic, and clinician availability barriers. Existing digital tools are largely developed in English, leaving Spanish-speaking patients with few culturally and linguistically appropriate resources [4, 11].

In this context, a web and mobile system integrating Large Language Models (LLMs) and agentic workflows is justified. Such a platform can provide adaptive, personalized, and evidence-based therapy directly on mobile devices, enabling patients to practice independently while allowing therapists to monitor progress and adjust interventions. By addressing the gap in Spanish-language resources, this project meets a critical public health need and leverages AI to enhance rehabilitation outcomes and quality of life for individuals with aphasia.

1.4 Objectives

1.4.1 General Objectives

Design and implement a web and mobile platform that, through the use of Large Language Models (LLMs) and agent-based workflows, supports the creation, personalization, and monitoring of speech-language therapies for Spanish-speaking individuals with aphasia, integrating evidence-based therapeutic methods and adapting exercises to each patient's needs and progress.

1.4.2 Specific Objectives

1. Develop a mobile application for patients that provides access to interactive therapy exercises based on configurable templates of Verb Network Strengthening Treatment (VNest) and Spaced Retrieval (SR), ensuring accessibility and adaptability to different performance levels.
2. Implement a web platform for therapists that offers tools for personalization, supervision, and monitoring of patient progress, incorporating adaptive content generation powered by Large Language Models (LLMs).
3. Integrate agent-based workflows to automate key therapeutic tasks, such as generating verb-role networks, scheduling spaced retrieval intervals, and systematically logging user performance.
4. Collaborate with speech-language therapy experts and individuals with aphasia to gather qualitative feedback on the mobile application and web platform.

1.4.3 Scope

This project covers the design, development, and deployment of a web and mobile system aimed at supporting Spanish-speaking individuals with aphasia, their caregivers, and optionally, speech-language therapists. The platform enables patients to independently interact with the system via a mobile application, while caregivers can input essential patient information to help tailor the experience. Therapists may use the web interface to monitor progress and provide guidance, but their involvement is not mandatory for the patient to use the system. The scope also includes collecting user feedback from patients and caregivers to evaluate usability, accessibility, and overall user experience. The system serves as a supportive tool for rehabilitation management, without providing medical diagnoses or treatments.

2 Related work

2.1 Referential framework

In this reference framework, we outline the key concepts essential for a proper understanding of the subject.

2.1.1 Aphasia

Aphasia is an acquired language disorder resulting from brain injury, typically due to a stroke in the left hemisphere, which impairs spontaneous speech, comprehension, reading, and writing. It is characterized by a partial or complete loss of language function, affecting both receptive and expressive modalities. Memory and other cognitive abilities are generally spared. [2] The classification of aphasia includes:

- **Fluent Aphasia:** Characterized by normal speech rhythm and fluency but with impaired semantic content. A common example is Wernicke's aphasia, where verbal comprehension is significantly compromised. [2]
- **Non-fluent Aphasia:** Characterized by slow, effortful speech production with generally preserved comprehension. Broca's aphasia is a representative example, where the individual can understand language but struggles to express themselves verbally. [2]

Other subtypes include global aphasia, which involves a severe loss of all language functions, and anomic aphasia, where individuals primarily struggle with naming objects. The diagnosis of aphasia is based on detailed clinical assessments, including neuropsychological tests and neuroimaging, to identify the localization and extent of brain lesions. [2]

2.1.2 Speech and Language Therapy

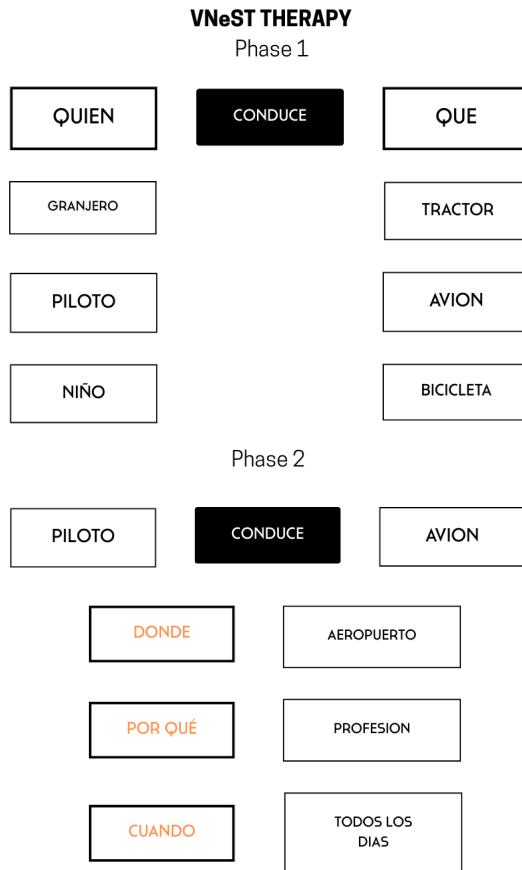
Speech-language therapy is a clinical intervention provided by certified speech-language pathologists aimed at assessing, diagnosing, and treating communication and swallowing disorders. In the context of aphasia rehabilitation, it encompasses restorative, functional, and socially oriented approaches designed to improve language comprehension, expression, and overall communicative effectiveness. This therapy addresses the impairment-based, activity-limiting, and participation-restricting consequences of aphasia, and is considered the predominant and evidence-supported treatment modality for individuals recovering from stroke-induced language deficits. [3]

2.1.3 Verb Network Strengthening Treatment (VNeST)

Verb Network Strengthening Treatment (VNeST) is a language rehabilitation approach designed for individuals with aphasia, particularly those with anomic or non-fluent variants. Unlike traditional noun-focused interventions, VNeST

places verbs at the center of therapy, under the assumption that verbs activate wider semantic networks and are essential for sentence construction. By strengthening the connections between verbs and their thematic roles (agents, objects, and adjuncts such as time, place, or reason), VNeST promotes more functional and generalizable communication. [14] [8]

The therapy typically unfolds through structured phases, which are illustrated in Figure 1. In the first phase, patients select a central verb and generate agent-object combinations (e.g., “the pilot drives the plane” or “the farmer drives the tractor”). The second phase expands these sentences by adding contextual details such as where, why, or when (e.g., “the pilot drives the plane at the airport every day”). In the third phase, patients are challenged with judgments of semantic plausibility, deciding whether newly generated sentences are meaningful or not. Finally, the fourth phase reinforces verb retrieval by explicitly asking the patient to identify the target verb practiced during the session.



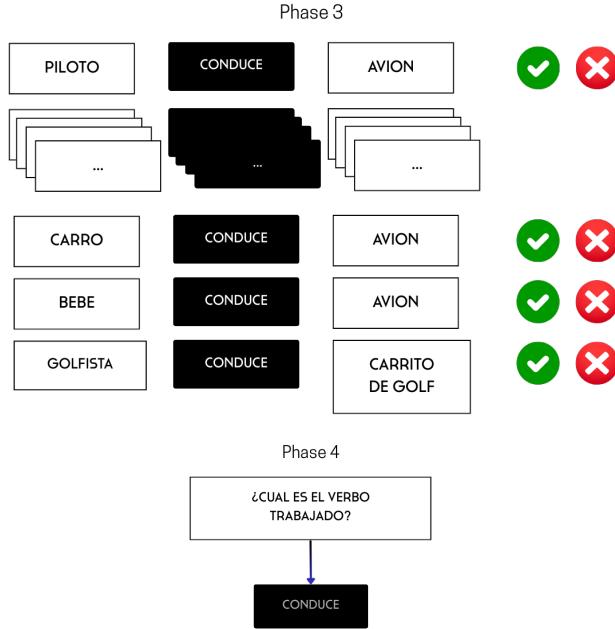


Figure 1: Phases of Verb Network Strengthening Treatment (VNeST).

This progressive structure encourages the patient to move from simple verb–argument generation to more complex and contextually rich sentence production. The method has demonstrated positive effects on both production and comprehension of language, with improvements extending beyond the trained items into spontaneous speech. Furthermore, technological tools such as web or mobile applications can facilitate the implementation of VNeST, supporting practice at home and enabling therapists or caregivers to adapt exercises to individual needs. [14] [8]

2.1.4 Spaced Retrieval (SRT)

Spaced Retrieval Training (SRT) is a memory-based intervention designed to enhance long-term retention of specific information in individuals with cognitive impairment or aphasia. Unlike therapies focused on language comprehension or production, SRT leverages implicit learning by repeatedly asking patients to recall targeted information over progressively longer intervals (e.g. 30 seconds, 1 minute, 2 minutes). If the response is correct, the interval increases; if incorrect, the information is reintroduced, and the interval is shortened. [9]

This technique is particularly effective in helping patients remember essential details such as names, instructions, locations, or communication strategies. SRT has shown success in improving functional memory and has been implemented in both clinical settings and digital tools. [9]

2.1.5 Artificial Intelligence and Natural Language Processing

Artificial intelligence (AI) refers to the development of computer systems capable of performing tasks that typically require human intelligence, such as reasoning, learning, and decision making. In healthcare, AI enables the analysis of complex medical data, supports clinical decision making, and improves patient care through automation and intelligent systems. [7]

Natural Language Processing (NLP) is a subfield of AI focused on the interaction between computers and human language. It involves the ability of machines to understand, interpret, generate and respond to human language in a meaningful way. NLP technologies, particularly large language models (LLMs), have shown potential in healthcare by enabling applications such as clinical documentation, patient communication, medical education, and diagnostic support. These models process a large amount of medical text and generate contextually relevant outputs, which contributes to a more efficient and personalized healthcare delivery. [5]

2.1.6 Large Language Models (LLMs)

Large Language Models (LLMs) are advanced artificial intelligence systems designed to understand, generate, and interact with human language on a scale. LLMs are trained on high-volume textual datasets, enabling them to capture complex linguistic patterns, contextual relationships, and domain-specific knowledge. In healthcare, LLMs demonstrate important capabilities in processing clinical documentation, supporting diagnosis, improving patient communication, and accelerating medical research. The training methods can be of different types: self-supervised learning methods (without human labeling or annotation) or semi-supervised learning methods (with a small amount of human-labeled data supported by many unlabeled data). [5]

2.1.7 AI Agents

AI agents refer to intelligent entities capable of perceiving their environment, processing information, and acting to achieve specific goals. They can take physical forms (e.g., robots, autonomous vehicles) or virtual forms (e.g., chatbots, voice assistants, intelligent software). Their main purpose is to support humans in decision-making, automate repetitive or complex tasks, and adapt to changing conditions in real time. [15]

In practice, AI agents are used in everyday applications such as recommendation systems (suggesting movies, music, or products), personal assistants (organizing schedules, responding to queries), and customer service (chatbots that provide immediate answers). In industry, they play a key role in fields like healthcare (assisting diagnosis, monitoring patients), finance (detecting fraud, advising investments), manufacturing (optimizing production lines), and education (personalized learning tools). [15] [6]

2.2 State of the Art

2.2.1 Aphasia Therapy

Speech and language therapy (SLT) remains the primary evidence-based intervention for aphasia rehabilitation. Clinical trials and systematic reviews have consistently demonstrated that therapy is most effective when delivered intensively and for extended periods, with recommendations of at least four hours per week of practice. Intensive SLT has been associated with measurable improvements in functional communication and language recovery even during the chronic stages of the disorder.[16]

Among the most widely studied approaches are Verb Network Strengthening Treatment (VNeST) and Spaced Retrieval (SR). VNeST is designed to strengthen semantic networks by training patients to generate agent–verb–patient structures (e.g., “The teacher reads the book”), thereby promoting lexical retrieval and facilitating generalization to untrained words and sentences. Multiple studies have shown that VNeST improves not only verb retrieval but also broader sentence production and discourse-level communication [17]. Spaced Retrieval, originally developed within memory rehabilitation, applies the principle of expanding intervals between recall attempts to consolidate learning. In aphasia therapy, SR has been adapted to support word retrieval and phrase production, showing positive outcomes particularly for individuals with anomia.[18, 19, 20]

Despite their demonstrated efficacy, traditional therapies face significant limitations. They require highly trained professionals, are resource-intensive, and often rely on frequent in-person sessions that may be inaccessible to many patients due to geographical, economic, or systemic barriers.[21] These challenges are even more pronounced for Spanish-speaking populations. A recent scoping review of treatments for Spanish speakers with aphasia found that, although some adaptations of established therapies exist, the overall number of studies is limited and evidence-based approaches remain scarce, especially in Latin America.[11] This gap underscores the need for complementary approaches, including technology-enhanced solutions, to extend therapy opportunities and maintain continuity beyond the clinical setting.

2.2.2 Technology in Aphasia Therapy

The increasing demand for intensive and sustained aphasia rehabilitation has motivated the exploration of digital technologies as a means to extend the reach of speech-language therapy. Given the shortage of trained professionals and the practical limitations of frequent in-person sessions, technology-based tools provide opportunities to increase the frequency, accessibility, and continuity of therapeutic interventions beyond the clinical setting.[21]

Several commercial solutions have been developed to support people with aphasia, although their functionality remains limited. Lingraphica is primarily offered as a dedicated communication device rather than a mobile application, while Tactus Therapy provides a series of mobile apps with structured exercises

targeting naming, comprehension, and repetition [22, 23]. Despite their clinical relevance, these tools are essentially static, relying on pre-programmed content with little opportunity for personalization or adaptive progression. Moreover, most are designed for English-speaking users, leaving a notable gap in resources for speakers of other languages, such as Spanish [22]. In addition, the financial cost of these products poses a significant barrier to accessibility, particularly in low-resource settings [22]. Finally, these platforms are not explicitly based on specific evidence-based therapy protocols, such as Verb Network Strengthening Treatment (VNeST) or Spaced Retrieval (SR), but instead focus on more general memory or language practice activities.

Research prototypes and reviews have highlighted the potential of mobile technologies to extend aphasia therapy beyond traditional clinical settings. For example, SCARA, a system developed in Portugal, enables speech-language pathologists to configure exercises, assign tasks, and monitor patient progress through a centralized dashboard, demonstrating the feasibility of computer-based therapy management [24]. Similarly, an integrative review by Brandenburg et al. (2013) examined the accessibility and potential uses of smartphones and tablets for people with aphasia, emphasizing their capacity to deliver cost-effective, context-sensitive, and portable support [22]. Mobile devices offer unique opportunities to increase therapy intensity and facilitate participation in daily life activities. However, the review also identified substantial barriers, including usability challenges related to language, vision, and motor impairments, as well as design shortcomings such as complex navigation, small buttons, or text-heavy interfaces. While these findings demonstrate that mobile technology can enhance rehabilitation and social participation, current solutions remain fragmented, often lacking systematic clinical validation and rarely grounded in specific evidence-based therapy protocols.

2.2.3 AI in Aphasia Therapy

Artificial intelligence (AI) has increasingly been investigated as a tool to support clinical practice in aphasia. Recent systematic reviews have highlighted that most applications of AI in this domain have concentrated on assessment and evaluation rather than on the direct provision of therapy. For instance, Adikari et al. (2023) reviewed the use of machine learning in aphasia rehabilitation and found that the majority of studies focused on automatic assessment of language impairments, classification of aphasia subtypes, or prediction of therapy outcomes, with relatively few addressing treatment delivery [25]. Similarly, Azevedo et al. (2023) emphasized that current AI applications are primarily designed to complement diagnostic processes and to provide decision support for clinicians, rather than to create or adapt therapeutic interventions [26].

Within these assessment-oriented applications, a number of approaches have been explored. Machine learning models have been used to automatically classify aphasia types based on speech and language features, to predict patients' recovery trajectories, and to estimate responsiveness to different therapy intensities [25]. Other projects have explored AI-driven writing support, such as word

prediction and spelling correction, to facilitate communication for people with aphasia [26]. Beyond these auxiliary uses, Constant Therapy (CT) represents a notable example of a clinically deployed digital therapeutic that integrates an AI-based recommendation engine. CT delivers more than 85 types of therapy tasks across multiple language and cognitive domains, and its adaptive algorithms continuously adjust difficulty and content according to individual performance, thereby sustaining patient engagement and maximizing therapeutic outcomes. Clinical studies and large-scale retrospective analyses have provided promising evidence of CT's efficacy, particularly in enabling intensive, self-managed practice for individuals with poststroke aphasia [27].

2.2.4 Automatic Speech Recognition for Aphasia

A prominent line of research within AI for aphasia has focused on Automatic Speech Recognition (ASR). The motivation behind these efforts is to provide real-time feedback on spoken language tasks, particularly naming, which traditionally requires direct supervision from a speech-language pathologist. By automating this process, ASR systems aim to increase therapy intensity and allow patients to engage in more independent practice.

One notable example is NUVA (Naming Utterance Verifier for Aphasia), which employs deep learning models to evaluate patients' naming attempts. NUVA demonstrated the feasibility of automatically detecting correct and incorrect responses in naming tasks, thereby offering a scalable way to monitor progress and reduce the workload of clinicians [28]. Other projects have explored fine-tuned speech recognition models adapted to atypical speech patterns in aphasia, reporting improvements in transcription accuracy when compared to off-the-shelf ASR systems.

Despite these promising results, ASR systems face several limitations when applied to aphasic speech. High rates of disfluencies, phonemic paraphasias, and incomplete utterances often challenge recognition accuracy [29]. Furthermore, most ASR models have been trained primarily on typical speech, making them less robust when confronted with the variability inherent in pathological speech [29, 30]. These issues may lead to biased or inconsistent feedback, potentially frustrating patients or misrepresenting their actual progress [28].

2.2.5 LLMs and Generative AI for Aphasia

The most recent developments in aphasia therapy have explored the use of large language models (LLMs) and generative artificial intelligence (AI) to move beyond recognition and assessment toward content generation and adaptive support. Unlike traditional digital tools, which rely on static exercises, LLMs offer the ability to dynamically create linguistically rich and contextually relevant therapeutic materials, opening new possibilities for personalized rehabilitation.

A pilot study of Aphasia-GPT demonstrated how generative AI can be applied to augmentative and alternative communication (AAC) by transforming agrammatic utterances into well-formed sentences and generating supportive

conversational scaffolds [31]. This approach showed promise in facilitating communication and reducing frustration for people with non-fluent aphasia. Similarly, Chatphasia integrated an LLM-driven cue generation system into a mobile therapy platform, providing hierarchical semantic and phonemic cues tailored to individual patient needs [29]. These studies illustrate the potential of LLMs to deliver personalized support and to automate components of evidence-based techniques such as cueing hierarchies.

Despite these encouraging advances, the application of LLMs in aphasia therapy remains in its infancy. Existing systems have been tested only in small-scale pilot studies, often limited to English-speaking populations. Moreover, they primarily focus on augmentative support or word retrieval tasks, rather than implementing structured, evidence-based therapeutic protocols such as Verb Network Strengthening Treatment (VNeST) or Spaced Retrieval (SR). As a result, current generative approaches remain experimental and lack integration into comprehensive therapy frameworks.

2.2.6 Limitations

Although the integration of technology and artificial intelligence into aphasia rehabilitation has shown promising results, several limitations remain evident in the current state of the art. Most existing systems are restricted in scope, focusing on narrow aspects of therapy such as word retrieval or simple comprehension tasks, while neglecting structured, evidence-based treatment protocols like Verb Network Strengthening Treatment (VNeST) or Spaced Retrieval (SR). As a result, their therapeutic relevance and generalization are limited.

Commercial tools, while accessible, are often static and offer minimal personalization, with most resources designed primarily for English-speaking populations [22, 23]. This creates a gap for speakers of other languages, including Spanish, where culturally and linguistically adapted digital therapies remain scarce. Furthermore, the financial cost of many commercial solutions creates additional barriers to access, particularly in low-resource settings.

Research prototypes have introduced more advanced functionalities, such as web-based dashboards and AI-driven feedback, yet these systems are typically tested only in small pilot studies [29, 31]. Their evaluations often lack longitudinal evidence of effectiveness, making it difficult to determine their long-term clinical impact. In addition, technical challenges persist: automatic speech recognition (ASR) systems still struggle with disfluencies, phonemic paraphasias, and incomplete utterances characteristic of aphasic speech [30], while generative AI systems have so far been limited to exploratory tasks rather than full-scale therapy delivery.

Taken together, the literature indicates that current technological solutions for aphasia therapy are fragmented, either overly static or experimental, and not yet integrated into robust clinical frameworks. These limitations highlight the need for platforms that can work with evidence-based methods, adapt dynamically to patients' progress, and expand access to underrepresented languages and populations.

2.2.7 State of the Art Conclusion

Aphasia rehabilitation has gradually evolved from traditional speech-language therapy to the incorporation of digital tools and, more recently, artificial intelligence. Evidence-based approaches such as Verb Network Strengthening Treatment (VNeST) and Spaced Retrieval (SR) have demonstrated strong clinical efficacy, but their implementation depends on intensive, therapist-led sessions that are not always accessible to patients. Commercial applications and research prototypes have expanded therapy beyond the clinic, yet most remain static, costly, and focused on English-speaking populations, offering limited personalization and little alignment with structured therapeutic protocols.

Artificial intelligence has opened new possibilities in aphasia care, with applications in automatic classification, prediction of outcomes, and speech recognition for real-time feedback. While these systems represent important progress, they continue to face technical challenges with aphasic speech and have limited clinical validation. Early explorations of large language models (LLMs) and generative AI point to the potential for adaptive cueing and dynamic content creation, but current implementations are experimental and narrow in scope. These gaps underscore the need for platforms that integrate established therapeutic methods with AI-driven adaptability, are accessible across languages, and deliver clinically relevant interventions. This thesis responds to that need by proposing a web and mobile system that leverages LLMs and intelligent agents to automatically generate Spanish-language therapy exercises grounded in VNeST and SR.

3 System Design and Methodology

3.1 Overview

This section presents the overall design, architecture, and implementation methodology of the RehabilitIA platform. The system comprises two interconnected applications: (i) a web interface for therapists to manage patients, configure and assign exercises, and monitor progress, and (ii) a mobile application for patients to access and complete therapy tasks. Both clients interact with a shared backend service that encapsulates business logic and coordinates an AI agent responsible for exercise generation and adaptive assistance. Data persistence and synchronization are handled through a NoSQL schema in Firebase Firestore, enabling real-time updates across both applications.

On the client side, the mobile and web applications follow a component-based architecture supported by centralized authentication and secure data access mechanisms. On the server side, the backend exposes authenticated endpoints and centralizes privileged operations, ensuring proper access control and consistent behavior across the system. Overall, the design prioritizes accessibility for people with aphasia, traceability and auditability of clinical actions, and extensibility to accommodate future therapy protocols, analytics modules, and AI model upgrades.

3.2 System Architecture

The overall architecture of the platform follows a multi-tier and service-oriented design that integrates a web client, a mobile client, a backend API service, and cloud-based data storage and authentication components. Figure 2 illustrates the interaction between these elements.

The system is composed of four core layers:

- **Client layer:** Consists of a web application developed in *React* for therapists and an Android/iOS mobile application developed in *Flutter* for patients. Both applications provide user interfaces adapted to their respective roles.
- **Backend layer:** Implemented in *FastAPI* (Python), this service hosts the logic related to the AI agent, including the generation and personalization of therapy exercises. It is deployed on a dedicated Ubuntu 24 server at Universidad de los Andes.
- **AI Agent layer:** The backend integrates an AI agent that uses the *Azure OpenAI GPT-4.1* model through REST endpoints. The agent automates four core processes: (1) generating Verb Network Strengthening Treatment (VNeST) exercises, (2) generating Spaced Retrieval (SR) exercises, (3) personalizing exercises according to each patient's profile, and (4) structuring profile information from patient dialogue into standardized JSON formats.
- **Data layer:** Both clients and the backend connect to a shared *Firebase Firestore* NoSQL database for real-time synchronization and data persistence. User authentication is handled via *Firebase Authentication*, and Cloud Functions are used for administrative tasks such as approving or rejecting therapist registration requests.

Communication between components follows a hybrid approach:

- The clients (web and mobile) access Firestore directly via the *Firebase SDK* for standard data operations (e.g., retrieving assigned exercises, updating progress).
- For AI-related tasks, both clients send requests to the backend through *FastAPI REST endpoints*, which handle communication with the AI agent and return generated or personalized content.
- The backend may also perform database reads/writes directly using the *Firebase Admin SDK*, particularly when data consistency or server-side validation is required.

3.3 System Architecture

The overall architecture of the platform follows a multi-tier and service-oriented design that integrates a web client, a mobile client, a backend API service, and cloud-based data storage and authentication components. Figure 2 illustrates the interaction between these elements.

The system is composed of four core layers:

- **Client layer:** Consists of a web application developed in *React* for therapists and an Android/iOS mobile application developed in *Flutter* for patients. Both applications provide user interfaces adapted to their respective roles.
- **Backend layer:** Implemented in *FastAPI* (Python), this service hosts the logic related to the AI agent, including the generation and personalization of therapy exercises. It is deployed on a dedicated Ubuntu 24 server at Universidad de los Andes (IP: 172.24.100.18, 4 CPU, 12 GB RAM, 45 GB storage).
- **AI Agent layer:** The backend integrates an AI agent that uses the *Azure OpenAI GPT-4.1* model through REST endpoints. The agent automates four core processes: (1) generating Verb Network Strengthening Treatment (VNeST) exercises, (2) generating Spaced Retrieval (SR) exercises, (3) personalizing exercises according to each patient's profile, and (4) structuring profile information from patient dialogue into standardized JSON formats.
- **Data layer:** Both clients and the backend connect to a shared *Firebase Firestore* NoSQL database for real-time synchronization and data persistence. User authentication is handled via *Firebase Authentication*, and Cloud Functions are used for administrative tasks such as approving or rejecting therapist registration requests.

Communication between components follows a hybrid approach:

- The clients (web and mobile) access Firestore directly via the *Firebase SDK* for standard data operations (e.g., retrieving assigned exercises, updating progress).
- For AI-related tasks, both clients send requests to the backend through *FastAPI REST endpoints*, which handle communication with the AI agent and return generated or personalized content.
- The backend may also perform database reads/writes directly using the *Firebase Admin SDK*, particularly when data consistency or server-side validation is required.

The client applications follow a straightforward architectural separation in which the user interface is handled locally on each device, while all data operations and privileged actions rely on Firebase services and the backend API. Both

the web and mobile clients authenticate users through Firebase Authentication and access Firestore directly for real-time data retrieval and updates. The web platform validates that authenticated users correspond to verified therapist accounts stored in the `terapeutas` collection, while the mobile application grants access only to users registered as `pacientes`. Administrative features, such as approving or rejecting therapist registration requests, are performed through dedicated Cloud Functions to ensure that sensitive operations occur exclusively on the server side.

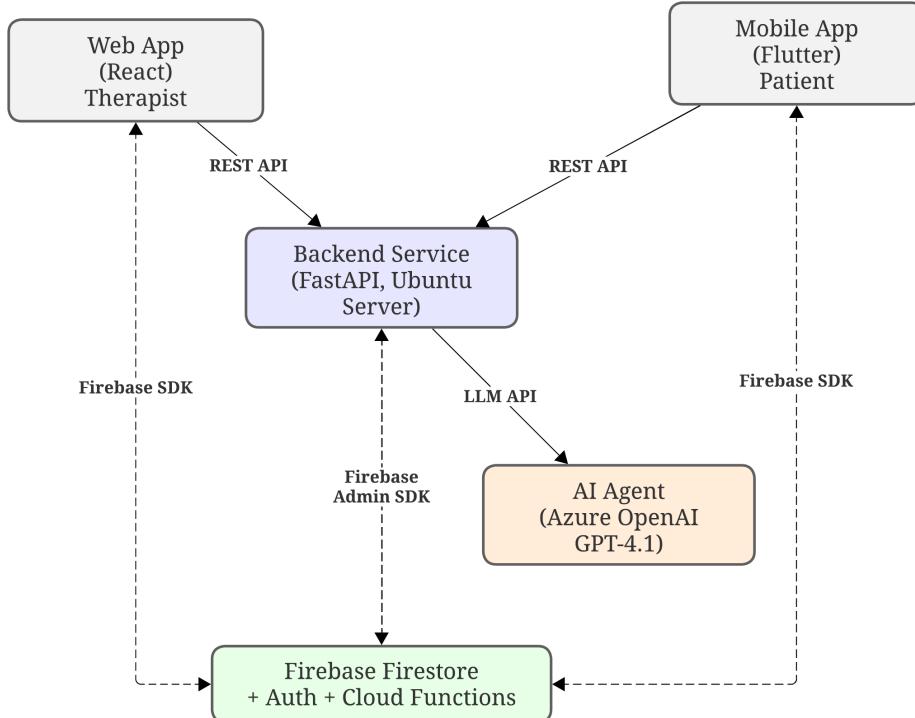


Figure 2: System architecture illustrating the interaction between the web and mobile applications, the backend service with the AI agent, and Firebase services (Firestore, Authentication, and Cloud Functions).

3.4 Design and User Experience (UX/UI)

3.4.1 UX/UI

Color Selection and Visual Identity. The color palette was selected to balance accessibility, emotional comfort, and visual structure. A soft, warm background tone (e.g., a light cream) minimizes visual fatigue, while a saturated accent color (such as orange) highlights actionable elements and areas requiring user attention. This contrast aligns with accessibility guidelines and

supports visual discrimination for individuals who experience slower or effortful processing.

Neutral grays and subtle shadows are used to differentiate interactive components such as cards or answer tiles. The overall aesthetic avoids dense text blocks, overly decorative icons, or unnecessary visual complexity, prioritizing clarity and therapeutic focus.

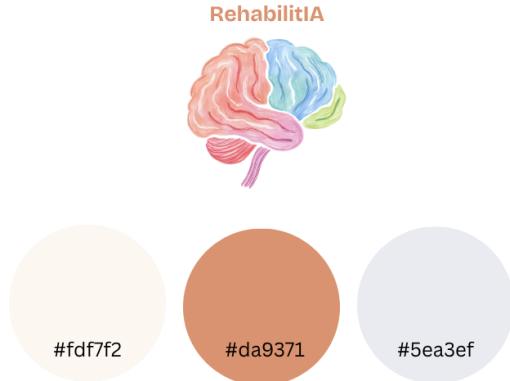


Figure 3: Color palette and logo.

Design Principles. The design of the mobile application is guided by several principles grounded in accessibility literature and the specific needs of individuals with aphasia:

- **Minimal Cognitive Load.** Interfaces display only the necessary information for the active task. Each screen contains a single question or exercise step, avoiding multi-element layouts that compete for attention.
- **Large, High-Contrast Buttons.** Interactive elements exceed 48 dp touch areas to accommodate reduced motor precision. Generous spacing prevents accidental selections.
- **Linear, Predictable Navigation.** The app employs simple, one-directional flows. Patients always move forward through a structured sequence (e.g., VNeST phases, SR intervals), reducing the risk of confusion or getting lost.
- **Reduced Need for Writing.** Because typing is cognitively and motorically demanding, most tasks rely on multiple-choice interactions or selection-based prompts. When writing is required, the system provides an optional microphone input.
- **Immediate Feedback.** To reinforce learning and compensate for short-term memory limitations, the app provides instant correctness validation and concise feedback messages directly below the selected response.

- **Accessible and Multimodal Interaction.** Visual cues such as icons, card layouts, and color-coded elements guide interaction without relying exclusively on text comprehension. Voice input supports users with limited motor precision or writing difficulties.

3.4.2 Mobile Application Design

The design of the mobile patient application follows a user-centered and accessibility-focused approach, shaped by the cognitive, linguistic, and motor limitations commonly observed in individuals with aphasia. Rather than replicating the structure of conventional mobile applications, the interface emphasizes clarity, simplicity, and predictability, reducing cognitive load and enabling patients to navigate exercises independently.

The therapeutic nature of the app requires immediate and unambiguous feedback. Because memory and attentional difficulties may cause patients to forget the reasoning behind their choices, responses are validated instantly, and corrective cues appear directly within the task flow. This ensures that learning opportunities occur in context, supporting the mechanisms of both Verb Network Strengthening Treatment (VNeST) and Spaced Retrieval (SR).

Overall, the interface relies on visual consistency, large touch targets, and high-contrast interactive elements. Tasks are designed to require recognition rather than recall, offering multiple-choice selections, simplified prompts, and optional voice input to reduce the need for typing. These design choices collectively aim to support autonomy, reduce frustration, and maximize therapeutic engagement.

Overview of Main Screens. The following elements characterize the patient experience across the major screens of the app:

Onboarding and Registration. The onboarding process consists of a sequence of full-screen steps, each containing a single prompt. This minimizes cognitive load and prevents users from encountering multiple fields simultaneously. Clear headings, large interactive controls, and a prominent “Next” button ensure that progression feels natural and non-intimidating.

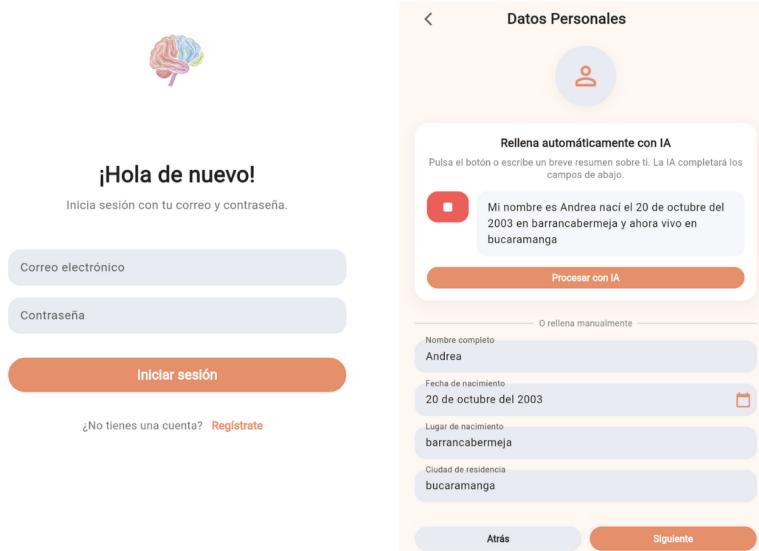


Figure 4: Overview of the main patient application screens.

Therapy Dashboard. The dashboard presents the two available therapeutic modules—VNeST and Spaced Retrieval—as large, clearly labeled cards. Distinct icons and color accents facilitate recognition and support users with impaired reading comprehension. No nested menus are used, reducing navigational friction.



Figure 5: Overview of the therapy dashboard.

VNeST Exercise Screens. Each phase of VNeST (verb selection, agent/object generation, contextual expansion, plausibility judgment) is presented as an isolated task. Interactive choices appear as large tiles or buttons. Feedback appears immediately after each selection, keeping the cognitive distance between action and feedback minimal.

Spaced Retrieval Screens. SR tasks display a short recall prompt accompanied by either multiple-choice options or an optional text field with microphone input. A color-coded timer conveys pacing without inducing stress. Correct or incorrect responses are shown instantly, and subsequent intervals begin automatically to avoid confusion.

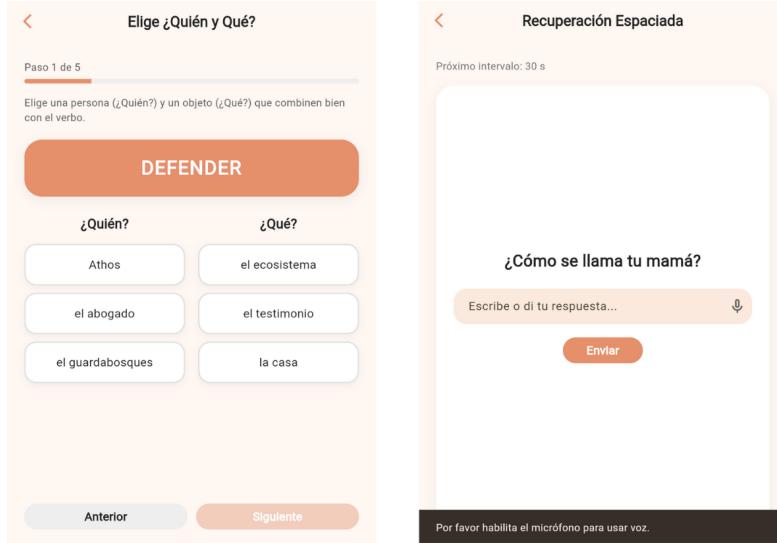


Figure 6: Overview of the VNeST and SR screens.

3.4.3 Web Application Design

The web application is designed primarily for speech-language therapists and administrative staff, providing a structured interface for creating, assigning, and reviewing therapy exercises, as well as managing therapist registration requests. Unlike the mobile patient application, which emphasizes simplicity and accessibility, the web interface prioritizes efficiency, clarity, and professional workflows. Each view is organized to minimize friction and reduce the time required for therapists to supervise patients, review progress, and generate personalized VNeST or Spaced Retrieval (SR) exercises.

The platform maintains a visual identity consistent with the mobile application, featuring warm backgrounds, high-contrast accents, and clean typography. However, the layout makes use of larger screens, presenting dashboards, tables, and forms with clear visual hierarchy and abundant spacing. This ensures fast navigation while preserving the therapeutic tone and brand identity of Rehabilitia.

The application includes dedicated flows for login, registration, patient supervision, exercise creation, and administrative approval of new therapists, each described below.

Login and Therapist Registration. Therapists access the platform through a unified login page featuring large input fields, a centered layout, and high-contrast buttons. The design communicates trust and professionalism while remaining visually consistent with the therapeutic brand.

The registration page collects essential information (name, profession, experience, and motivation to join Rehabilitia) through a clean, single-column

form. Subtle shadows and soft backgrounds help reduce visual clutter. Once a therapist submits a request, it appears in the administrator dashboard for approval.

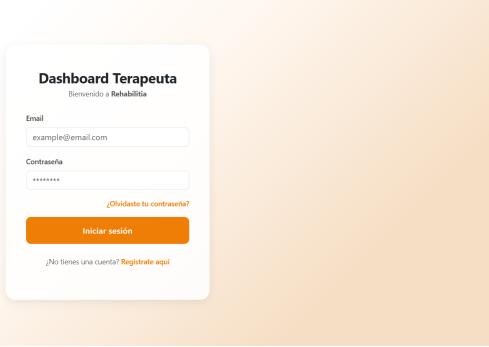


Figure 7: Login page for therapists.



Figure 8: Registration page for therapists.

Therapist Dashboard. Upon logging in, therapists are greeted with a dashboard summarizing key information: number of associated patients and exercises pending review. Each metric appears inside a card with rounded corners, soft gradients, and an illustrative icon. The goal is to provide immediate situational awareness without requiring multiple clicks.

The dashboard prioritizes clarity and quick access to high-priority items, helping therapists efficiently manage caseloads.

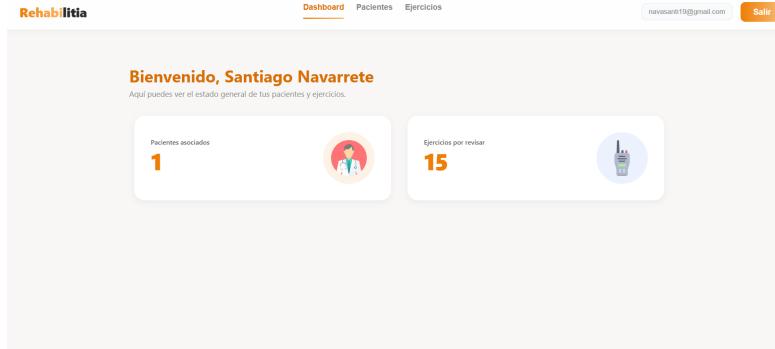


Figure 9: Therapist dashboard showing patient and exercise counters.

Patient List and Patient Detail. The patient module displays a searchable and filterable list of users assigned to the therapist. Upon selecting a patient, the interface highlights their active and completed exercises, grouped under the VNeST and Spaced Retrieval modalities.

Filters (status, verb, context) are positioned at the top in a horizontal bar, using dropdowns and input fields with clear separation and large touch targets for ergonomic use. The resulting table presents each assigned exercise with its status, personalization level, dates, and action buttons.

Pacientes			
NOMBRE	EMAIL	EJERCICIOS ASIGNADOS	ACCION
Andrea Galindo	andrealgalindoc@gmail.com	15	Ver detalles

Figure 10: Patient list view with filtering options.

The screenshot shows the 'Ejercicios' (Exercises) section of the Rehabilitia application. At the top, there are navigation links: Dashboard, Pacientes, Ejercicios, and a user email (navesant19@gmail.com). Below the header is a search bar with filters for 'Estado' (Todos), 'Verbo' (Buscar verbo), 'Contexto' (Buscar contexto), and a 'Limpiar' button. A large orange button labeled 'Nuevo Ejercicio' is visible. The main area displays a table of exercises assigned to patient EBA7695. The columns are: ID, CONTEXTO, VERBO, PERSONALIZADO, NIVEL, ESTADO, FECHA ASIGNADO, FECHA REALIZADO, and ACCIÓN. The data rows are:

ID	CONTEXTO	VERBO	PERSONALIZADO	NIVEL	ESTADO	FECHA ASIGNADO	FECHA REALIZADO	ACCIÓN
EBA7695	Deportes	defender	Sí	difícil	COMPLETADO	16/11/2025	16/11/2025	<button>Ver</button>
E66C51D	Viajes	visitar	Sí	medio	PENDIENTE	5/11/2025	—	<button>Ver</button>
E286C3F	Servicios de transporte	entregar	No	medio	PENDIENTE	5/11/2025	—	<button>Ver</button>

Figure 11: Patient detail view with exercise filtering options.

Exercise Management and Review. From the “Ejercicios” section, therapists can oversee all exercises created in the system, regardless of patient assignment. The module includes advanced filtering options (visibility, personalization, verb, context, patient ID), enabling precise control during review workflows.

Tables are intentionally spacious, with bold headers and tag-based visual indicators (e.g., “Pendiente”, “Por revisar”, “Completado”). Action buttons (“Revisar”, “Ver”) follow the brand’s orange accent color.

The screenshot shows the 'Gestión de Ejercicios' (Exercise Management) page. At the top, there are navigation links: Dashboard, Pacientes, Ejercicios, and a user email (navesant19@gmail.com). Below the header is a search bar with filters for 'Visibilidad' (Todos), 'Estado' (Todos), 'Personalizado' (Todos), 'Verbo' (Buscar verbo), 'Contexto' (Buscar contexto), 'ID Paciente' (Buscar ID paciente), and a 'Limpiar' button. An orange button labeled 'Nuevo ejercicio' is visible. The main area displays a table of exercises. The columns are: ID, CONTEXTO, VERBO, PERSONALIZADO, ASIGNADO A, REVISADO, and ACCIÓN. The data rows are:

ID	CONTEXTO	VERBO	PERSONALIZADO	ASIGNADO A	REVISADO	ACCIÓN
EBA7695	Deportes	defender	Sí	andrealgalindoc@gmail.com	<button>POR REVISAR</button>	<button>Revisar</button> <button>Ver</button>
E7D5395	Ir a un Restaurante	reservar	No	—	<button>POR REVISAR</button>	<button>Revisar</button> <button>Ver</button>
E66C51D	Viajes	visitar	Sí	andrealgalindoc@gmail.com	<button>POR REVISAR</button>	<button>Revisar</button> <button>Ver</button>
E31B41E	Viajes	visitar	No	—	<button>POR REVISAR</button>	<button>Revisar</button> <button>Ver</button>

Figure 12: Exercise management view with filters and review actions.

AI-Assisted Exercise Creation. The “Nuevo Ejercicio” page lets therapists generate customized VNeST exercises using the AI engine. The page presents a compact, centered form where therapists select context, difficulty level, and visibility settings.

A clean, high-contrast “Generar ejercicio” button finalizes the configuration. The design emphasizes clarity and minimizes the cognitive load required to configure new content.

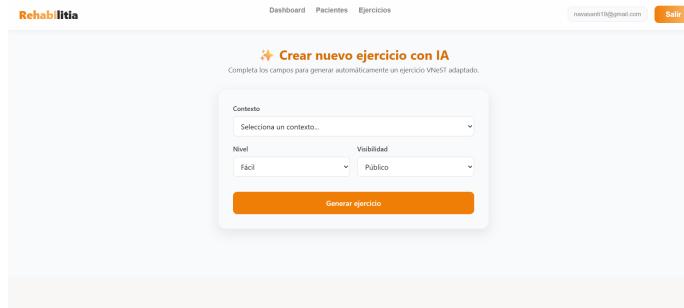


Figure 13: Page for generating a new AI-assisted VNeST exercise.

Administrator Dashboard for Therapist Requests. The administrator interface extends the same styling as the therapist view but includes tools for verifying therapist registration requests. Pending applications appear as cards containing the therapist’s submitted information and action buttons to approve or reject.

Completed and approved requests are listed separately to maintain organizational clarity. This structure allows the administrator to efficiently manage onboarding workflows without navigating deep menus.

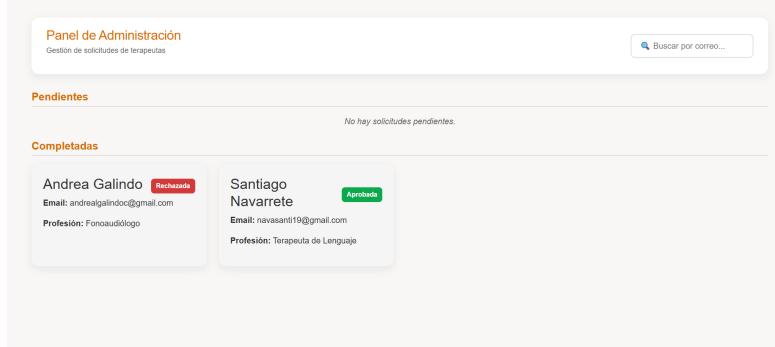


Figure 14: Administrator dashboard for processing therapist registration requests.

3.5 Functional Overview

The Rehabilitia platform is composed of two complementary applications—a patient-facing mobile application and a therapist-facing web application—that together implement the full therapeutic workflow of the system. The mobile application focuses on accessibility, simplicity, and autonomous exercise delivery for individuals with aphasia, while the web platform enables therapists and administrators to supervise patients, generate and review exercises, assign therapy tasks, and manage professional onboarding.

Both applications operate over the same Firestore database and collectively support a continuous loop of information: therapists configure or personalize exercises; patients complete them on the mobile app; the system records progress and dynamically adapts content; and therapists review results or generate new tasks as needed. Because the full interaction flow and detailed screen sequences of both applications are extensive, all user interface diagrams are organized into dedicated annexes. The mobile application flows are presented in Annex A, while the complete therapist and administrator web flows are provided in Annex B.

The remainder of this section summarizes the core functionality of each platform, emphasizing how they jointly support personalized, evidence-based interventions grounded in VNeST and Spaced Retrieval (SR) while ensuring usability for both patients and clinicians.

3.5.1 Mobile Application Functionality

The mobile application serves as the primary therapeutic interface for patients with aphasia. Its design prioritizes clarity, predictability, low cognitive load, and multimodal input support, enabling users to engage in therapy independently despite linguistic or motor limitations. The main components include registration, VNeST therapy delivery, Spaced Retrieval (SR) exercises, and the generation of personalized therapy tasks. Full visual flows and screen references are provided in Annex A.

Registration. The registration functionality guides new patients through a simplified onboarding flow designed to minimize cognitive and motor demands. Upon launching the application for the first time, users progress through a series of single-step screens that collect essential personal and autobiographical information. Each screen presents only one question or input type at a time, avoiding multi-field forms and reducing cognitive load.

Patients can provide information either through structured fields (e.g., selecting from predefined options) or through optional voice input when text entry may be challenging. Consistent with the accessibility principles described earlier, the interface uses large controls, high-contrast buttons, and linear navigation to ensure clarity at every stage. The data collected during registration—including personal details, daily routines, family structure, and meaningful objects—is later used to automatically generate personalized therapy content, particularly for Spaced Retrieval exercises. The full screen flow and example interface elements for the registration process are provided in Annex A.

VNeST Therapy. The VNeST functionality implements the complete Verb Network Strengthening Treatment protocol within an accessible mobile interface. Because this therapeutic workflow involves multiple sequential phases, its full navigation diagram and associated screens are provided in Annex A.2. Functionally, the module can be summarized in four main steps:

- **Context and verb selection.** When the patient enters the VNeST module, the system first displays a list of semantic contexts (e.g., *sports*, *transportation*, *household activities*), as shown in Annex A.2.1. After selecting a context, the patient sees the verbs associated with it, including any personalized or therapist-assigned verbs. Choosing a verb initiates the structured VNeST sequence.
- **Phase 1: Agent–Object generation.** The patient must select a semantically appropriate agent (subject) and object for the chosen verb. The interface presents several candidates, allowing progression only once a valid pair is selected. Incorrect choices trigger immediate explanatory feedback, and the evolving verb–agent–object sentence is displayed at the bottom of the screen to maintain context (Annex A.2.2).
- **Phase 2: Contextual expansion.** The base sentence is expanded by adding contextual elements such as *where*, *when*, and *why*. For each slot, the patient receives four multiple-choice options, with a single correct answer. Immediate feedback clarifies why each option is correct or incorrect, and correct selections are appended to the evolving sentence (Annex A.2.3).
- **Phase 3 and summary.** The third phase presents ten sentences—some plausible, some implausible—for the patient to classify (e.g., swiping right for plausible and left for implausible). Misclassifications receive semantic feedback, and a closing summary screen displays the final contextualized sentence built throughout the session, reinforcing integration and closure (Annex A.2.4 and Annex A.2.5).

Spaced Retrieval (SR) Therapy. The Spaced Retrieval module implements an adaptive recall-based memory protocol in which patients repeatedly retrieve a target piece of autobiographical information over progressively increasing time intervals. The general flow of the SR exercise and the corresponding screens are presented in Annex A.3. The core functional behavior is:

- **Progressive interval structure.** The exercise begins at the shortest interval and requires the patient to successfully recall the target information a total of five times. Each recall trial is displayed individually (Annex A.3). Patients may respond either by typing or using the built-in microphone, accommodating users with motor or writing difficulties.
- **Adaptive timing and feedback.** Correct responses cause the system to double the interval before the next recall attempt (e.g., 30s → 60s, 60s → 120s). Incorrect responses trigger immediate feedback reminding the patient of the correct information and reset the interval to the last value for which the patient had previously answered correctly. The timing and interval progression logic are summarized in Annex A.3.

- **Completion criterion.** The exercise concludes once the patient completes five consecutive correct intervals. A summary screen acknowledges completion and reinforces the successfully retrieved information, maintaining a simple, linear interaction flow aligned with accessibility guidelines.

Exercise Personalization. The application allows patients to generate personalized therapy content using their autobiographical profile. Because the personalization flow involves several nested interactions (context selection, verb selection, and background generation), its full screen sequence is provided in Annex A.4. Functionally, the module operates as follows:

- **Context and verb selection.** When the patient enters the personalization module, they first see a list of semantic contexts. Selecting a context immediately expands the interface to display the verbs associated with it within the same screen, minimizing navigation depth. This inline expansion reduces cognitive load and supports patients who may struggle with multi-level menus.
- **Generation of personalized exercises.** After selecting a verb, the application automatically generates a customized VNeST or SR exercise based on the patient’s autobiographical data (e.g., family members, routines, meaningful objects). The generation process takes a few seconds, during which the patient sees a progress indicator to avoid confusion or premature navigation (Annex A.4).
- **Post-generation options.** Once the exercise is successfully generated, the application notifies the patient and offers two options: (i) navigate directly to the newly generated exercise to begin working on it immediately, or (ii) return to the personalization screen to generate additional exercises or explore other contexts. This functionality keeps therapy content personally meaningful and dynamically adaptable, supporting motivation and long-term engagement in aphasia rehabilitation.

3.5.2 Web Application Functionality

The web platform enables therapists and administrators to perform essential tasks that complement and extend the patient workflow. While the mobile application is the delivery environment for therapy, the web interface is the professional workspace through which exercises are curated, reviewed, and assigned. Full flow diagrams and screens appear in Annex B.

Login and Therapist Registration. Therapists authenticate through a unified login page. New therapists complete a registration form containing identity, credentials, and motivation. Their request remains pending until a system administrator reviews and approves it (Annex B.1).

Therapist Dashboard. After authentication, therapists access a dashboard summarizing patient counts, pending items, and shortcuts to patient management, exercise review, and exercise creation (Annex B.2).

Patient Management. Therapists inspect the profiles and exercises of assigned patients, filter tasks by status, verb, context, or personalization, and review detailed exercise content (Annex B.3). Two additional tools enhance this workflow:

- **Personalize Exercise for a Patient:** therapists generate a customized VNeST exercise for a specific patient using their profile data.
- **Assign Existing Exercise:** therapists select a public or previously created exercise and assign it manually.

Exercise Management and Review. Therapists browse, filter, and review all exercises in the system, including those generated by AI. The module supports detailed inspection, verification, and marking an exercise as reviewed (Annex B.4).

AI-Assisted Exercise Creation. Therapists can create new VNeST exercises using a streamlined form that specifies context, difficulty level, and visibility. The AI engine generates structured sentences, verb–argument pairs, and contextual expansions (Annex B.5).

Administrator Approval. Administrators oversee therapist registration requests, verify credentials, and approve or reject applicants. Approved users gain immediate system access, while rejected applications are archived (Annex B.6).

Together, the mobile and web applications provide a complete therapeutic ecosystem that integrates evidence-based rehabilitation protocols with AI-assisted content generation and clinician supervision, ensuring both accessibility for patients and control for therapists.

3.6 Technical Overview

3.6.1 Web Application (Therapist and Administrator Interface)

The web application is designed for two user roles: *therapists* and *administrators*. Its primary goal is to support the management of patients and therapy exercises while providing administrative oversight for new therapist registrations. The platform is developed in *React* using *JavaScript* and integrates directly with *Firebase Firestore*, *Firebase Authentication*, and *Firebase Cloud Functions* through the modular SDK (v9). Each React component is styled individually using *CSS* and *Bootstrap*.

Therapist functionality. Therapists can register on the platform and await administrative approval before gaining access. Once approved, they enter a personalized dashboard that centralizes their clinical and content management activities. Within this interface, therapists can:

- View and manage their assigned patients.
- Assign existing VNeST exercises to patients.
- Create new VNeST exercises using the AI agent integrated through the backend (*FastAPI*), specifying the contextual scenario, difficulty level, and visibility (public or private).
- Personalize existing exercises for individual patients by invoking the AI service.
- Review, edit, and approve exercises to ensure clinical quality.

The therapist dashboard displays all exercises that are either (i) public, (ii) private and created by the logged-in therapist, or (iii) private but assigned to one of their patients. Exercises created by other therapists remain inaccessible, maintaining privacy and authorship boundaries. Each exercise modification or creation that involves AI triggers a backend request to the FastAPI service, which processes the generation logic and writes the resulting exercise into Firestore. In contrast, operations that do not require AI—such as assigning or retrieving exercises—interact directly with Firestore via the frontend SDK.

Therapists can also monitor patient progress by viewing completion status and recent activity within assigned exercises. Although no statistical dashboards are currently implemented, the system provides visibility into which exercises have been completed and the most recent interactions per patient.

Administrative functionality. Administrators have access to a dedicated interface that lists all pending therapist registration requests. From this dashboard, they can review submissions and either approve or reject them. Upon performing either action, a corresponding Firebase Cloud Function (`aprobarTerapeuta` or `rechazarTerapeuta`) is triggered, automatically sending a customized confirmation or rejection email to the applicant. Only approved therapist accounts can subsequently authenticate and access the main web platform.

Data flow and integration. The web client communicates with both Firebase and the backend. Data persistence and retrieval occur primarily through Firestore, while AI-based exercise generation, personalization, and profile structuring are handled by REST API calls to the FastAPI backend. Once generated, exercises are stored in Firestore, from which they become accessible to both the therapist and corresponding patient accounts. This hybrid communication model ensures that the platform leverages real-time updates through Firebase while maintaining scalability for computationally intensive AI tasks.

Access control and data protection. User authentication is managed by Firebase Authentication. Role distinction is enforced at the client level: the web application accepts logins from accounts existing in the `terapeutas` or `admins` collections, while the mobile client validates access only for users in the `pacientes` collection. There is no redirection between applications—each client enforces its own access policies independently. As only verified and approved therapist or administrator accounts can access the platform, patient data exposure is restricted to authorized users, ensuring basic privacy and ethical compliance within the therapeutic workflow.

3.6.2 Mobile Application (Patient Interface)

The mobile application is designed exclusively for *patients* and serves as the primary interface through which users complete therapeutic exercises, personalize content, and manage their autobiographical profile. The app is developed in *Flutter* following an *MVVM* architecture, ensuring a clear separation between view logic, state management, and data operations. All information displayed to the patient: exercises, progress, and personalized content, is retrieved in real time from *Firebase Firestore*, while *Firebase Authentication* enforces secure user access. The application also integrates native device capabilities such as speech-to-text, allowing patients with limited writing ability to input text using their voice.

Patient functionality. Patients interact with the mobile application through a set of guided flows that cover registration, therapy execution, personalization, and progress review:

- **Registration and profile structuring.** Upon creating an account, patients complete a guided registration process consisting of four sections: *personal*, *familia*, *rutinas*, and *objetos*. Each section offers the option to fill in structured fields or provide a free-form paragraph that can be automatically processed by the Profile Structuring AI agent. Once registration is completed and the profile is structured, the system automatically generates a set of Spaced Retrieval (SR) exercises tailored to the patient’s autobiographical information.
- **Therapy dashboard and VNeST interaction.** After logging in, the patient is presented with a therapy dashboard displaying the two available interventions: VNeST and Spaced Retrieval. In the VNeST module, users first select among available contextual categories; each context then displays its associated verbs. Verbs for which a personalized exercise has been generated are marked with a lightbulb icon, encouraging engagement with individualized content. Selecting a verb loads the corresponding exercise, including subject–verb–object pairs, contextual “where/when/why” expansions, and the final sentence evaluation activity. If no personalized

or assigned exercise exists for a given verb, the system automatically retrieves a public exercise from the shared pool, assigns it to the patient, and then loads it in the interface.

- **Spaced Retrieval interaction.** In the SR module, the app presents one autobiographical recall question at a time. The client manages the full spacing logic locally—interval progression, success streaks, lapse counts, and computation of the next scheduled review—while synchronizing the updated state with Firestore after each interaction. The interface prioritizes cognitive simplicity by displaying a single card per screen and providing immediate feedback on the patient’s response.
- **Exercise personalization.** Patients can also request exercise personalization directly from the mobile application. Through a dedicated interface, they browse available contexts and verbs, select an exercise, and trigger a personalization request. The app sends the patient ID, selected exercise ID, and profile to the backend, which generates a personalized version and stores it in Firestore. These personalized exercises are then given highest display priority and are visually highlighted with the lightbulb icon.
- **Progress overview.** The profile section summarizes the patient’s progress, including the total number of completed exercises and the most recent activity. All values are computed from Firestore records and updated in real time as new exercises are completed.

Data flow and integration. The mobile client reads all exercises, progress markers, and personalized content directly from Firestore. Assigned exercises take precedence when determining which version of an exercise to display; personalized versions have even higher priority. Exercise completions are recorded by updating the corresponding Firestore documents. AI-driven operations—such as profile structuring and exercise personalization—are handled via REST calls to the FastAPI backend, while all non-AI operations (retrieval, assignment, progress updates) are executed through Firestore.

Access control and data protection. Authentication is managed entirely through Firebase Authentication. The mobile application accepts logins only from accounts stored in the `pacientes` collection, ensuring strict role separation between therapist and patient environments. The app enforces local checks to guarantee that patients can access only their own exercises and profile information. This isolation prevents unauthorized cross-account access and ensures compliance with ethical and privacy guidelines for clinical data.

3.7 Backend and AI Agent

3.7.1 Overview

The backend is a lightweight FastAPI service that coordinates the web app, mobile app, AI workflows, and Firestore. FastAPI provides a clean, type-safe setup (via Pydantic), low-latency JSON APIs, and smooth integration with Python’s AI stack, especially our LangGraph pipelines, for efficient prompt processing.

The service exposes a small set of focused endpoints for: (i) generating VNeST exercises, (ii) generating Spaced Retrieval (SR) cards, (iii) personalizing existing exercises for a given patient profile, and (iv) structuring free text into a normalized patient profile schema. Each of these routes delegates core reasoning and content generation to dedicated LangGraph entry points, while the backend handles request validation, dispatch, and returning structured responses to the clients.

To enable cross-origin access from our web front end during integration, the service includes a permissive CORS configuration. A Firestore client (via the Firebase Admin SDK) is initialized at startup, allowing downstream LangGraph nodes and helper functions to persist generated artifacts and read the metadata required for therapy workflows.

3.7.2 Backend Service Architecture

The backend is implemented as a single FastAPI application that wires four business endpoints to their corresponding LangGraph “main” functions and helper routines. Each main function encapsulates the orchestration logic for a specific therapeutic workflow, while a separate set of `prompts_*.py` files (seen in Annex C) defines the LLM instructions, examples, and context templates used by the agent.

The core structure of the backend codebase can be summarized as follows:

- `main.py`: Initializes the FastAPI application, defines all routes, configures CORS, and establishes the Firestore client connection.
- `main_langgraph_vnest.py`: Entry point for the VNeST generation agent. Imports LangGraph nodes and prompt definitions from `prompts_vnest.py`.
- `main_langgraph_sr.py`: Entry point for the Spaced Retrieval (SR) generation agent, which builds retrieval cards and memory cues. Uses prompt templates from `prompts_sr.py`.
- `main_personalization.py`: Handles personalization of existing exercises according to patient profiles. Calls the prompt defined in `prompts_personalization.py`.
- `main_profile_structure.py`: Processes free-form textual information (e.g., patients routines) into structured profile data using the definitions in `prompts_profile_structure.py`.

- **prompts_*.py files:** Each prompt file defines the templates, contextual examples, and role-based instructions that guide the behavior of its corresponding agent. This modular design allows each therapeutic workflow to evolve independently and facilitates rapid prompt iteration without modifying the backend code.

The main endpoints are described below.

Root Health Check

- **Route:** GET /
- **Behavior:** Returns a simple JSON payload to verify service availability.

VNeST Exercise Generation

- **Route:** POST /context/generate
- **Input schema (Pydantic ContextGeneratePayload):**
 - context: str
 - nivel: str
 - creado_por: str
 - tipo: str
- **Dispatch:** main.langraph_vnest(context, nivel, creado_por, tipo)
- **Behavior:** Triggers the LangGraph pipeline that composes a VNeST exercise based on the requested scenario and difficulty, then returns the generated structure (and, downstream, persists it to Firestore).

Spaced Retrieval (SR) Card Generation

- **Route:** POST /spaced-retrieval/
- **Input schema (Pydantic SRPayload):**
 - user_id: str
 - profile: dict
- **Dispatch:** main.langraph_sr(user_id, profile)
- **Behavior:** Produces a set of SR prompts/cards tailored to the patient profile; results are returned to the caller and stored for subsequent assignment.

Exercise Personalization

- **Route:** POST /personalize-exercise/
- **Input schema (Pydantic PersonalizePayload):**
 - user_id: str
 - exercise_id: str
 - profile: dict
- **Dispatch:** main_personalization(user_id, exercise_id, profile)
- **Behavior:** Adapts an existing exercise to the patient’s profile (e.g., vocabulary, context, difficulty), returning the personalized artifact and updating persistence.

Profile Structuring

- **Route:** POST /profile/structure/
- **Input schema (Pydantic ProfileStructurePayload):**
 - user_id: str
 - raw_text: str
- **Dispatch:** On-demand import and call: main_profile_structure(user_id, raw_text)
- **Behavior:** Converts unstructured clinician/caregiver notes into a normalized profile representation usable by the generation pipelines.

Request Validation and Typing All routes use Pydantic models to validate incoming payloads and provide clear, self-documenting contracts, keeping client-backend integration stable even as the agent graphs evolve.

Integration with Firestore The application initializes `firebase_admin.firestore.client()` once at startup; LangGraph routines and helper functions use this client to write generated exercises, personalization outputs, and structured profiles. Non-generative operations (e.g., reads) can be served directly by the clients via the Firebase SDK, while generative workflows go through the backend to centralize AI orchestration.

Separation of Concerns The backend keeps business logic minimal: it validates input, routes to the appropriate LangGraph entry point (`main_langraph_vnest`, `main_langraph_sr`, `main_personalization`, `main_profile_structure`), and returns the structured result. All reasoning steps, prompt assembly, tool calls, and guardrails live inside the LangGraph nodes, which simplifies maintenance and enables independent evolution of agent behavior without changing the HTTP surface.

3.7.3 System and AI Agent Workflows

This section unifies the request flow and internal AI agent logic for the four main backend processes: (1) exercise generation, (2) spaced retrieval creation, (3) exercise personalization, and (4) patient profile structuring. Each workflow begins with a client request (from the web or mobile interface), continues through the FastAPI backend, and reaches a specific AI agent that orchestrates language model reasoning and Firestore persistence.

The overall pattern is consistent across agents:

1. **Request initiation:** triggered from the web or mobile client.
2. **Backend processing:** validation and routing via FastAPI.
3. **AI agent execution:** modular reasoning steps implemented through LangGraph or sequential pipelines.
4. **Model inference:** calls to Azure OpenAI's GPT-4.1 deployment.
5. **Persistence:** structured JSON results stored in Firestore and linked to patient records.

Only the VNeST agent is implemented using *LangGraph* to explicitly define node transitions and state propagation. The remaining agents (Spaced Retrieval, Personalization, and Profile Structuring) follow a linear, LangChain-inspired pipeline, executed as sequential function calls.

1. VNeST Exercise Generation

When a therapist creates a new VNeST exercise from the web platform, the request follows the end-to-end path shown in Figure 15. The left lane captures the system request flow (client → FastAPI → persistence), while the right lane “zooms in” on the agent’s internal reasoning implemented with LangGraph.

External request flow.

1. The therapist triggers a creation action in the web app.
2. The frontend sends `POST /context/generate` with the parameters `context`, `nivel`, `creado_por`, and `tipo`.
3. FastAPI validates the payload (Pydantic) and dispatches the task to the VNeST agent.
4. After the agent returns a structured JSON, the backend persists both the general document in `ejercicios` and the extended contents in `ejercicios_VNEST`.
5. The new exercise becomes available to the assigned patient in the mobile app.

Internal agent flow (LangGraph). Before executing the five-node LangGraph workflow, the system constructs a sequence of prompt templates that guide the model through each reasoning stage. These templates define the linguistic constraints, JSON formats, difficulty rules, and expansion requirements used throughout the VNeST generation pipeline. The full prompt set used by the agent is included in Annex C.1, where each template is presented in its raw form for reproducibility and transparency.

Then, the agent decomposes generation into five nodes:

```
step1_generate_verbs proposes seven transitive verbs tailored to the context.  
step2_classify_verbs organizes them by lexical difficulty (easy/medium/hard).  
step3_select_pairs chooses one verb at the requested level and builds three  
S-V-O triplets under prototypicality constraints.  
step4_expand_sentences adds “where/when/why” expansions and generates  
ten evaluation sentences (mixing correct/incorrect).  
step5_save_db validates fields and writes both the general and extended exer-  
cise documents to Firestore.
```

Interface contract. Input:

```
{context, nivel, creado_por, tipo}.
```

Output: structured JSON with `id`, `verbo`, `pares`, and `oraciones`, plus metadata (`reviewed=false`, `personalizado=false`). This format guarantees the exercise can be listed, personalized later, and audited.

Rationale and benefits. Using LangGraph for VNeST improves traceability (clear intermediate states), robustness (JSON validation before persistence), and extensibility (adding nodes for scoring or therapist review requires only connecting new edges). The separation between request flow and agent flow makes the system observable end-to-end while keeping the generation logic modular.

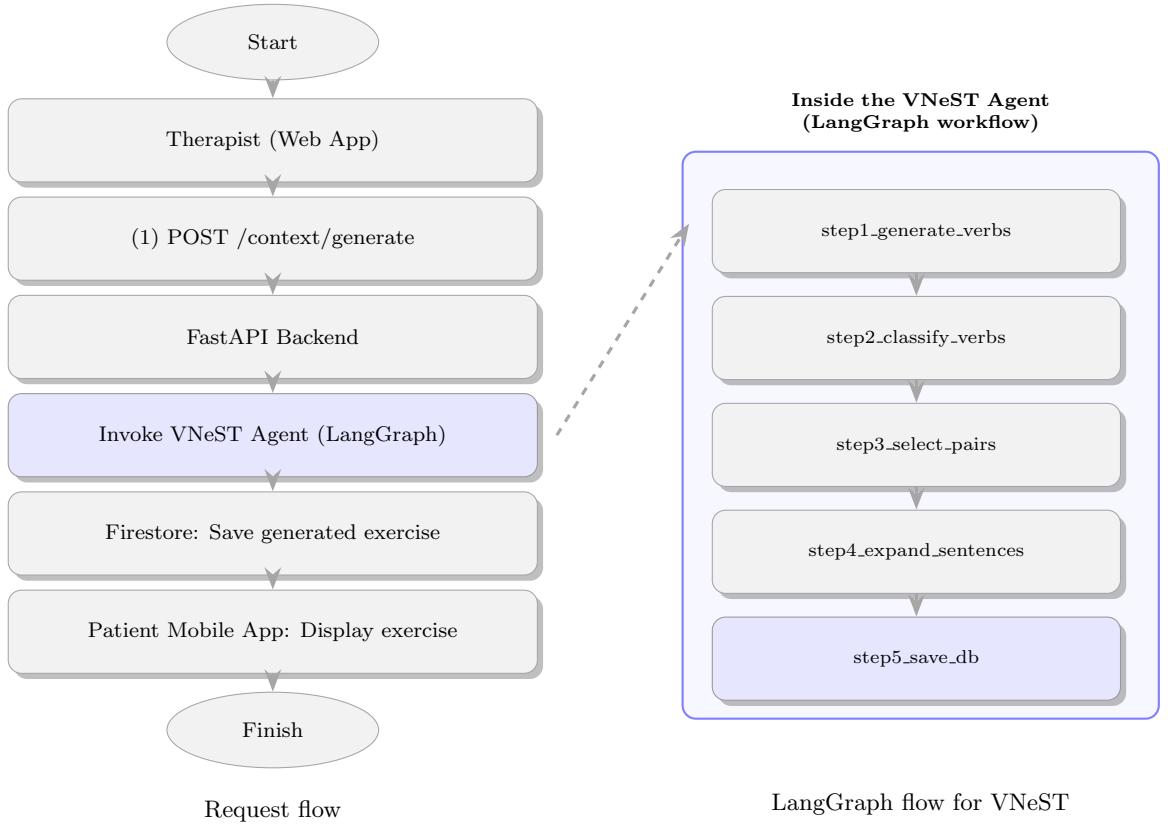


Figure 15: Combined request flow and internal LangGraph workflow for VNeST exercise generation.

2. Spaced Retrieval (SR) Generation

The *Spaced Retrieval* (SR) agent automates the creation of memory-based recall exercises derived from each patient's autobiographical profile. These exercises are used to reinforce memory retention through repeated recall intervals adapted to the patient's personal context. The workflow integrates both the external request flow and the internal AI pipeline, ensuring that personalized recall content is generated and stored automatically once a patient completes registration.

External request flow. As illustrated in Figure 16, the SR generation process is triggered immediately after a new patient finishes registration in the mobile application. At that moment:

1. The client sends a `POST /spaced-retrieval/` request containing the patient's `user_id` and structured profile.

2. The FastAPI backend validates the payload and calls the function `main_langraph_sr(user_id, profile)`.
3. The backend delegates the content creation task to the SR agent.
4. Once the SR cards are generated, the backend stores them in Firestore and links them to the corresponding patient’s record.
5. These exercises become visible immediately in the patient’s mobile therapy section.

Internal agent flow. Before executing the four-stage SR pipeline, the system constructs a dedicated prompt template that encodes the patient’s autobiographical profile and specifies the constraints for generating memory–recall items. This prompt guides the model in creating short, simple questions directly grounded in the patient’s personal history. The complete SR prompt template is included in Annex C.1 to ensure transparency and reproducibility of the generation process.

Inside the backend, the SR agent follows a linear pipeline composed of four functional stages:

- **build_prompt:** Constructs a personalized prompt based on the patient’s stored profile. The prompt uses autobiographical data such as personal, family, routine, and object-related details to generate meaningful recall questions.
- **call_model:** Sends the constructed prompt to the Azure OpenAI GPT-4.1 model via the `AzureOpenAI` client, requesting a structured JSON containing recall “cards”.
- **parse_and_validate:** Parses the model response and verifies that each card contains the correct schema: a `stimulus` (question), `answer`, and `category`.
- **persist:** Saves each generated card into two Firestore collections: `ejercicios` (for general exercise data) and `ejercicios_SR` (for SR-specific data). Each exercise is also assigned to the corresponding patient under `pacientes/<id>/ejercicios_asignados`.

Data structure and automation. The SR agent’s output is a list of short recall prompts that encourage the patient to retrieve personally meaningful information, such as family members, favorite activities, or objects of emotional significance. Each card is enriched with interval metadata used for spaced repetition scheduling (e.g., 15s, 30s, 60s, 120s, 300s). Because this process runs automatically after registration, therapists are relieved from manually configuring recall content for each new user.

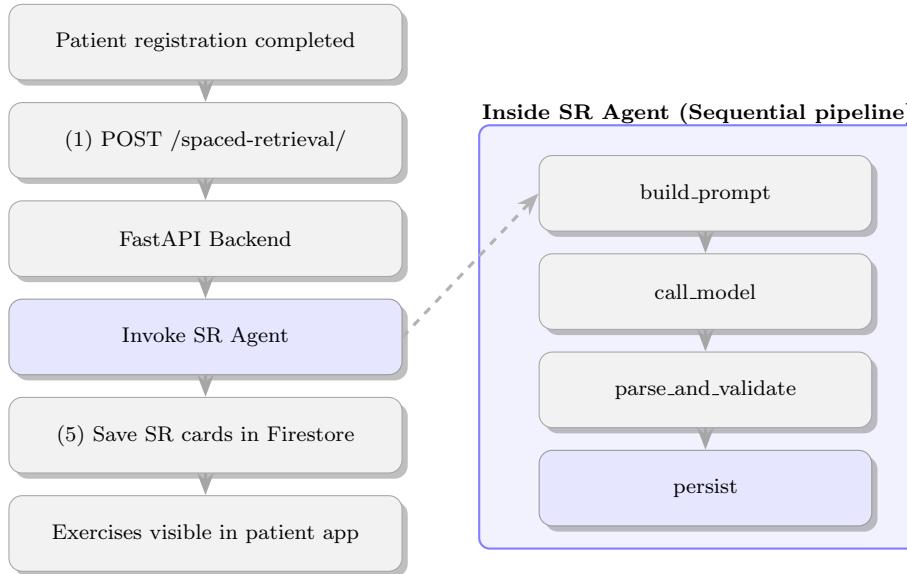


Figure 16: Combined request and internal pipeline for Spaced Retrieval generation.

3. Exercise Personalization

Exercise personalization enables the system to adapt previously generated VNeST or Spaced Retrieval (SR) exercises to the specific linguistic, cognitive, and biographical profile of a patient. This mechanism allows therapists, and in some cases patients themselves, to request customized versions of existing exercises that retain the original therapeutic structure while incorporating personally meaningful details. The complete process, shown in Figure 17, integrates the general request flow with the internal reasoning steps executed by the personalization agent.

External request flow. A personalization request may originate from either the therapist’s web interface or the patient’s mobile application. The sequence is as follows:

1. The user (patient or therapist) initiates a personalization request, selecting an existing exercise from their available list.
2. The frontend sends a `POST /personalize-exercise/` request containing the `user_id`, the `exercise_id` to adapt, and the patient’s structured profile.

3. FastAPI validates the payload and invokes `main_personalization`, which orchestrates the adaptation pipeline.
4. After the exercise is personalized and saved, the backend links the new version to the patient under `pacientes/{uid}/ejercicios_asignados`.
5. The updated exercise becomes immediately available in the patient's mobile application.

Internal agent flow. Before executing the five-stage personalization pipeline, the system constructs a specialized prompt that integrates both the base exercise and the patient's structured profile into a unified instruction set. This prompt explicitly defines which elements may be adapted, which must remain unchanged, and how the patient's autobiographical context should be incorporated. The complete personalization prompt template is provided in Annex C.1 to ensure transparency, reproducibility, and interpretability of the adaptation process.

The personalization agent follows a linear, LangChain-inspired pipeline that performs five key operations:

- **get_exercise_base:** Retrieves the original exercise from Firestore. This function merges its general metadata from `ejercicios` with its extended content found in `ejercicios_VNEST` or `ejercicios_SR`, depending on the therapy type.
- **generate_prompt:** Builds a detailed prompt that integrates the patient's profile—family members, routines, preferences, and meaningful objects—into the base exercise. The prompt specifies which parts of the exercise must remain unchanged and which elements may be adapted.
- **call_model:** Sends the composed instructions to the Azure OpenAI GPT-4.1 model, requesting a JSON object that preserves the original structure while replacing selected entities or sentences with personalized equivalents.
- **save_personalized_exercise:** Saves the adapted content back into Firestore under a new exercise ID. The function writes both the general document (in `ejercicios`) and the therapy-specific document (in `ejercicios_VNEST` or `ejercicios_SR`).
- **assign_exercise:** Associates the personalized exercise with the patient by creating an entry in `pacientes/{uid}/ejercicios_asignados`.

Preservation of structure and metadata. A crucial aspect of personalization is that the adapted exercise must remain structurally compatible with its base version. The agent therefore enforces:

- conservation of the therapy type and difficulty level,
- preservation of mandatory fields (*verbo, pares, oraciones* for VNeST; *stimulus, answer* for SR),

- metadata inheritance (creator, reference to the base exercise, timestamps, visibility),
- controlled adaptation: only specific elements are modified, and no global restructuring is allowed.

Design rationale. This pipeline enables large-scale reuse of previously created exercises, reduces therapist workload, and provides patients with material that is semantically aligned with their daily life. Although the agent does not use LangGraph explicitly, the design of its sequential modules mirrors the clarity and modularity of graph-based reasoning. Each step is independently testable, and the final output is always a valid, Firestore-compatible JSON document ready for client consumption.

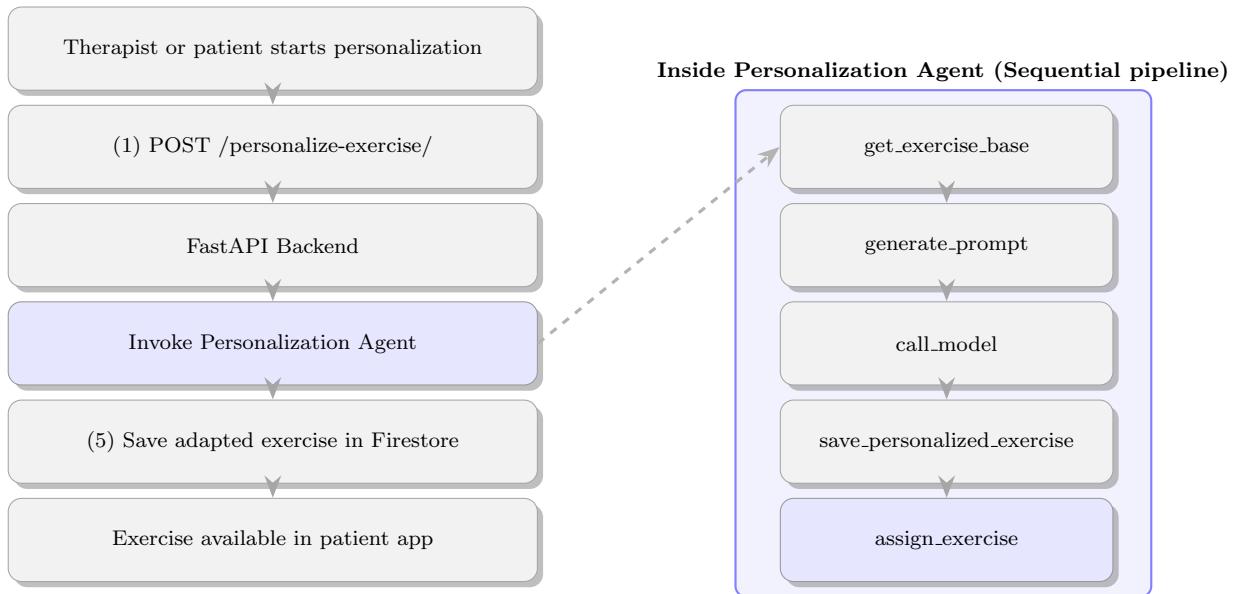


Figure 17: Combined request and internal pipeline for exercise personalization.

4. Patient Profile Structuring

The patient profile structuring workflow converts open-ended autobiographical text into a standardized JSON schema suitable for storage and later use in personalization or SR task generation. This process is triggered during the patient registration flow in the mobile application, where users may choose to describe personal, family, or daily routine information using free-form natural language. The complete interaction between the frontend, backend, and the structuring agent is shown in Figure 18.

External request flow. During registration, the mobile client gathers paragraph-level responses describing relevant autobiographical details. Once the user submits the form:

1. The frontend sends a POST `/profile/structure/` request containing the `user_id` and the raw, unstructured text.
2. The FastAPI backend validates the request and calls `main.profile_structure(user_id, raw_text)`.
3. The backend forwards the text to the Profile Structuring agent for interpretation and schema construction.
4. After the agent returns a structured JSON profile, the backend optionally persists it under the corresponding patient document in Firestore.

Internal agent flow. Before executing the three-stage structuring pipeline, the system constructs a dedicated prompt that specifies the target schema, relationship constraints, and semantic requirements for transforming unstructured autobiographical text into a structured patient profile. This prompt ensures that the resulting JSON conforms to the fields and subsections required by the system (*personal*, *familia*, *rutinas*, and *objetos*). The complete prompt template used for this transformation is included in Annex C.1, providing a transparent and reproducible reference for the profile-structuring process.

The structuring agent uses a concise, three-stage pipeline inspired by LangChain's sequential design:

- **generate_prompt:** Builds a detailed instruction prompt that specifies the target schema, required fields, allowed relationship categories, and constraints on semantic consistency (e.g., distinguishing people from places or activities).
- **call_model:** Sends the prompt and raw text to Azure OpenAI's GPT-4.1 model, requesting a structured JSON object divided into `personal`, `familia`, `rutinas`, and `objetos`.
- **persist_profile (optional):** Stores the resulting structured profile in Firestore under a patient-specific document for future personalization or SR generation.

Structured schema. The agent produces a JSON profile with consistent sections designed to support downstream therapeutic workflows:

- **personal:** name, birth information, places of residence, and other identifying attributes.
- **familia:** list of family members with relationship labels (*Hijo/a*, *Cónyuge/Pareja*, *Padre/Madre*, etc.) and short descriptions.
- **rutinas:** daily activities or meaningful habits relevant for therapy personalization.

- **objetos:** objects of emotional, practical, or autobiographical significance.

Design rationale. This workflow ensures that patient-specific content—otherwise expressed in free text—can be reliably extracted, normalized, and reused by downstream agents. By enforcing a consistent schema and validating semantic alignment within each field, the system enables personalized therapy generation at scale while maintaining clean, machine-interpretable patient records. The structuring agent also reduces the burden on therapists, who no longer need to manually translate patient narratives into structured data.

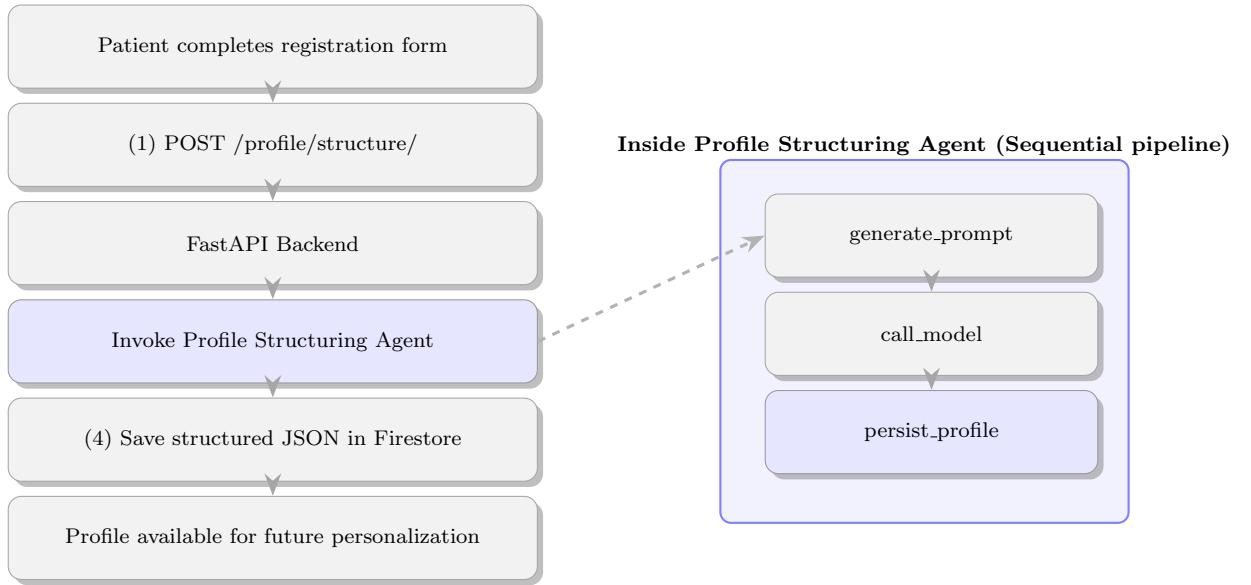


Figure 18: Combined request and internal pipeline for patient profile structuring.

3.8 Database Design

The platform employs *Firebase Firestore* as its primary database solution, chosen for its scalability, real-time synchronization, and native integration with Firebase Authentication and Cloud Functions. As a NoSQL document-oriented database, Firestore offers a flexible schema that supports nested structures and collections, allowing efficient representation of the hierarchical relationships between therapists, patients, and exercises. The database design follows a hybrid model where general exercise data is centralized, and specialized therapy data is distributed across distinct collections.

Overview of collections. The system defines eight main collections: `contextos`, `ejercicios`, `ejercicios_SR`, `ejercicios_VNEST`, `pacientes`, `solicitudes`, `terapeutas`,

and `terapias`. In addition, each patient document includes a subcollection named `ejercicios_asignados`, which stores the exercises currently linked to that individual. This design enables efficient querying, fine-grained access, and real-time synchronization of patient progress through Firestore's snapshot listeners.

Table 1: Main collections in Firebase Firestore.

Collection	Purpose	Description / Key Fields
terapeutas	Therapist registry	Stores information about verified therapists including <code>nombre</code> , <code>email</code> , <code>profesion</code> , <code>celular</code> , and list of assigned <code>pacientes</code> .
pacientes	Patient data	Contains personal, clinical, and contextual data such as <code>nombre</code> , <code>ciudad_residencia</code> , <code>fecha_nacimiento</code> , <code>terapeuta</code> , <code>familia</code> , <code>objetos</code> , and <code>rutinas</code> . Includes subcollection <code>ejercicios_asignados</code> .
ejercicios	General exercises	Central repository for all exercise metadata (e.g., <code>id</code> , <code>terapia</code> , <code>tipo</code> , <code>creado_por</code> , <code>fecha_creacion</code> , <code>revisado</code> , <code>personalizado</code>).
ejercicios_VNEST	VNeST therapy exercises	Stores specialized data for Verb Network Strengthening Treatment, including <code>contexto</code> , <code>nivel</code> , <code>oraciones</code> , and <code>pares</code> . References <code>id_ejercicio_general</code> .
ejercicios_SR	Spaced Retrieval exercises	Stores SR tasks with recall intervals and responses (<code>pregunta</code> , <code>rta_correcta</code> , <code>intervals_sec</code> , <code>status</code>). References <code>id_ejercicio_general</code> .
contextos	Context categories	List of available semantic contexts for exercise generation (e.g., “Educación”, “Deportes”).
solicitudes	Registration requests	Contains pending therapist applications with <code>nombre</code> , <code>email</code> , <code>profesion</code> , <code>motivacion</code> , and approval <code>estado</code> .
terapias	Therapy catalog	Defines supported therapy types (currently VNEST and SR).

General exercises. The `ejercicios` collection acts as a base layer that contains metadata common to all types of therapy exercises. Each document represents a single exercise instance and includes fields summarized in Table 2. This general schema promotes reusability across therapy modalities and allows each specific therapy collection to inherit common attributes.

Table 2: Structure of documents in the `ejercicios` collection.

Field	Type	Description
<code>creado_por</code>	String	Creator identifier or “IA” if generated automatically by the AI agent.
<code>fecha_creacion</code>	Timestamp	Exercise creation date.
<code>id</code>	String	Unique exercise identifier.
<code>id_paciente</code>	String / Null	Reference to the patient if the exercise is assigned.
<code>terapia</code>	String	Therapy type (<code>VNEST</code> or <code>SR</code>).
<code>personalizado</code>	Boolean	Indicates whether the exercise has been customized for a specific patient.
<code>revisado</code>	Boolean	Indicates whether the exercise has been clinically reviewed.
<code>tipo</code>	String	Visibility scope (<code>publico</code> or <code>privado</code>).

VNeST exercises. The `ejercicios_VNEST` collection stores data for Verb Network Strengthening Treatment exercises, referencing their corresponding general entry through the field `id_ejercicio_general`. Table 3 details the main fields stored for each VNeST exercise. This structure allows storing both the generated exercise content and the logical explanations that support therapeutic feedback, facilitating adaptive exercise generation and review.

Table 3: Structure of documents in the `ejercicios_VNEST` collection.

Field	Type	Description
<code>id_ejercicio_general</code>	String	Reference to the corresponding document in <code>ejercicios</code> .
<code>contexto</code>	String	Thematic context (e.g., “Servicios de Transporte”).
<code>nivel</code>	String	Difficulty level of the exercise.
<code>oraciones</code>	Array	List of sentences annotated with grammatical correctness, explanations, and semantic context.
<code>pares</code>	Array	Verb–argument pairs, each with subfields such as <code>sujeto</code> , <code>objeto</code> , and <code>expansiones</code> (“when”, “where”, “why”).

Spaced Retrieval exercises. The `ejercicios_SR` collection represents memory-based therapy tasks aligned with the Spaced Retrieval (SR) technique. Each document references its general record in `ejercicios` via `id_ejercicio_general`. Table 4 summarizes the fields used to represent SR training state.

Table 4: Structure of documents in the `ejercicios_SR` collection.

Field	Type	Description
<code>id_ejercicio_general</code>	String	Reference to the corresponding document in <code>ejercicios</code> .
<code>pregunta</code>	String	Recall prompt presented to the patient.
<code>rta_correcta</code>	String	Expected correct answer.
<code>intervals_sec</code>	Array of integers	Sequence of time intervals (in seconds) used for spaced retrieval.
<code>interval_index</code>	Integer	Index of the current interval in <code>intervals_sec</code> .
<code>next_due</code>	Number / Timestamp	Time when the next retrieval attempt is scheduled.
<code>status</code>	String	Current learning state (e.g., <code>learning</code> , <code>mastered</code>).

Patients and assigned exercises. The `pacientes` collection holds demographic and contextual information about each registered patient. The main fields are summarized in Table 5. It also contains arrays representing `familia`, `objetos`, and `rutinas`, which capture personalized information used for exercise adaptation. Each patient document contains a subcollection `ejercicios_asignados` with entries of the structure described in Table 6.

Table 5: Structure of documents in the `pacientes` collection.

Field	Type	Description
<code>nombre</code>	String	Patient's full name.
<code>email</code>	String	Contact email address.
<code>ciudad_residencia</code>	String	City of residence.
<code>fecha_nacimiento</code>	Timestamp / Date	Patient's date of birth.
<code>terapeuta</code>	String	Identifier or UID of the assigned therapist.
<code>familia</code>	Array	List of key family members or contacts relevant for personalization.
<code>objetos</code>	Array	Personally meaningful objects used to adapt exercises.
<code>rutinas</code>	Array	Daily routines and activities used as contextual cues.

Table 6: Structure of the `ejercicios_asignados` subcollection in each patient document.

Field	Type	Description
<code>id_ejercicio</code>	String	Reference to the general exercise document.
<code>contexto</code>	String	Thematic context of the exercise (e.g., "Deportes").
<code>tipo</code>	String	Therapy type (<code>VNEST</code> or <code>SR</code>).
<code>fecha_asignacion</code>	Timestamp	Date when the exercise was assigned.
<code>ultima_fecha_realizado</code>	Timestamp	Last time the exercise was completed.
<code>estado</code>	String	Current completion status (<code>pendiente</code> , <code>completado</code>).
<code>personalizado</code>	Boolean	Indicates if the exercise was personalized by AI.
<code>prioridad</code>	Integer	Priority level assigned by the therapist.
<code>veces_realizado</code>	Integer	Number of times the patient has completed the exercise.

This embedded structure supports real-time tracking of patient activity without requiring a separate "progress" collection.

Therapists and registration requests. Therapist information is stored in the `terapeutas` collection, whose main fields are shown in Table 7. The collection `solicitudes` records pending therapist registration requests, with the

fields summarized in Table 8. Administrative approval triggers a Firebase Cloud Function that updates the request’s status and sends an automated confirmation email.

Table 7: Structure of documents in the `terapeutas` collection.

Field	Type	Description
<code>nombre</code>	String	Therapist’s full name.
<code>email</code>	String	Contact email address used for authentication.
<code>profesion</code>	String	Professional title (e.g., “Fonoaudiólogo”).
<code>celular</code>	String	Phone number for contact and verification.
<code>pacientes</code>	Array	List of identifiers of patients assigned to this therapist.

Table 8: Structure of documents in the `solicitudes` collection.

Field	Type	Description
<code>nombre</code>	String	Name of the applicant.
<code>email</code>	String	Contact email for follow-up and login.
<code>profesion</code>	String	Declared profession or specialization.
<code>celular</code>	String	Phone number provided in the application.
<code>motivacion</code>	String	Short free-text description of motivation and experience.
<code>estado</code>	String	Validation state (<code>pendiente</code> , <code>aprobada</code> , <code>rechazada</code>).
<code>fecha_creacion</code>	Timestamp	Date when the request was submitted.

Administrative approval triggers a Firebase Cloud Function that updates the request’s status and sends an automated confirmation email.

Therapies and contexts. The `terapias` collection maintains the catalog of available therapy types—currently limited to VNeST and SR—while the `contextos` collection stores contextual categories used for generating exercises (e.g., “Educación”, “Deportes”, “Servicios de Transporte”). Both collections facilitate the scalability of the system by allowing the addition of new therapies or linguistic domains without altering the overall database structure. Their fields are summarized in Table 9.

Table 9: Structure of documents in the `terapias` and `contextos` collections.

Collection	Field	Type	Description
<code>terapias</code>	<code>id</code>	String	Unique identifier of the therapy type.
<code>terapias</code>	<code>nombre</code>	String	Human-readable name (e.g., “VNEST”, “SR”).
<code>terapias</code>	<code>descripcion</code>	String	Short description of the therapeutic approach.
<code>contextos</code>	<code>id</code>	String	Unique identifier of the context.
<code>contextos</code>	<code>nombre</code>	String	Name of the semantic context (e.g., “Deportes”).
<code>contextos</code>	<code>descripcion</code>	String	Optional descriptive label or examples of situations.

Example schema. The following simplified diagram summarizes the hierarchical organization of the database:

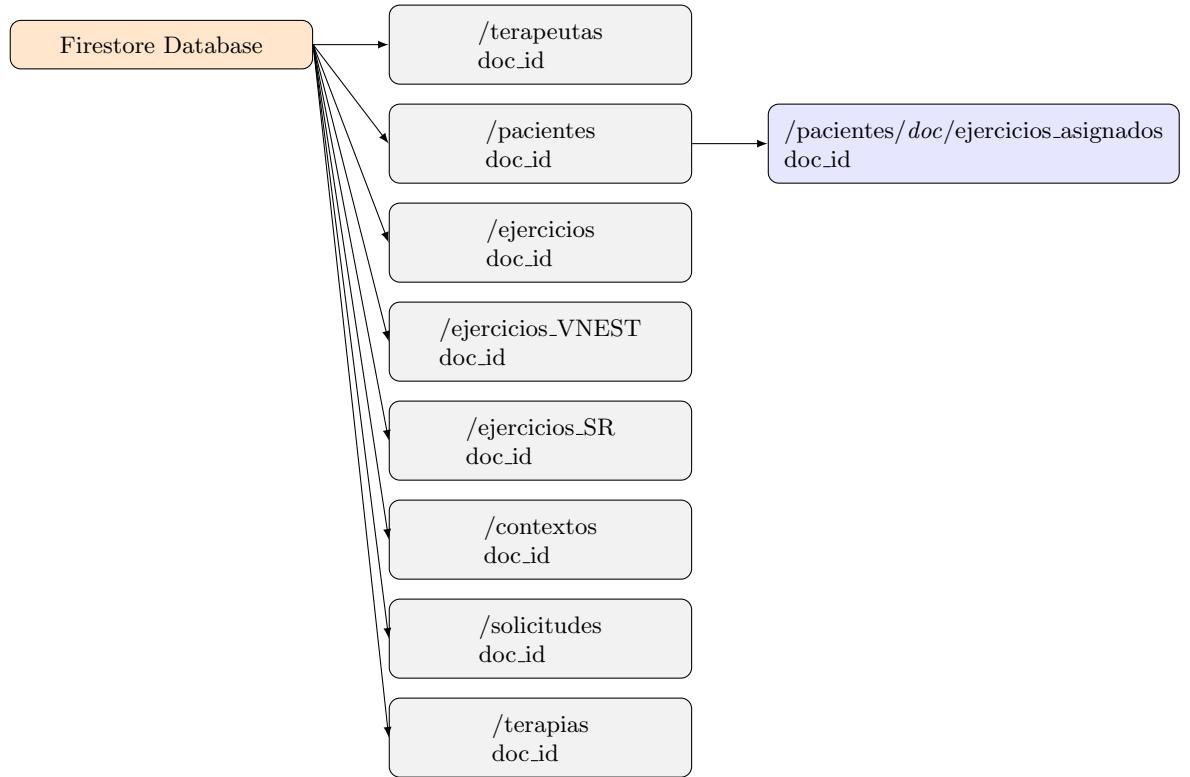


Figure 19: Firestore schema: root, main collections, and subcollection.

3.9 Cloud Functions for Therapist Verification

To support secure therapist onboarding and ensure that only verified professionals gain access to patient information and clinical management tools, the platform incorporates two backend Cloud Functions deployed on Firebase. These functions automate all privileged operations associated with the approval or rejection of therapist registration requests. By placing this logic on the server side, the system prevents unauthorized actions, avoids exposing sensitive credentials to the client, and guarantees consistency in the administrative workflow.

Therapist Approval Function. When an administrator approves a therapist request through the web platform (see Annex B.6), the system invokes a dedicated Cloud Function responsible for transforming the pending request into a fully active therapist account. This function performs several coordinated tasks:

- It creates a new authenticated user profile for the therapist within Firebase Authentication, ensuring that account creation occurs only through secure server-side operations.
- It generates a password-creation link that allows the therapist to set their credentials without transmitting or storing passwords in the administrator interface.
- It registers the therapist in the `terapeutas` collection, storing their name, contact details, professional background, motivation statement, and initial status.
- It updates the original request in the `solicitudes` collection, marking it as “approved” and recording the corresponding timestamp.
- Finally, it sends an automated confirmation email that includes the secure password-creation link and guidance for accessing the platform.

This workflow ensures that therapist accounts are provisioned safely, that no sensitive operations occur in the client applications, and that the approval process remains auditable and consistent across the system.

Therapist Rejection Function. When a registration request is not approved, a second Cloud Function manages the rejection process. Its responsibilities include:

- Updating the request entry in the `solicitudes` collection to reflect its “rejected” status and storing the corresponding timestamp.
- Sending an automated email notification to the applicant, informing them of the decision and inviting them to reapply in the future.

By centralizing this process in a backend-controlled environment, the platform ensures that no inconsistent states occur (e.g., rejected requests remaining active) and that communication with applicants is uniform and timely.

4 Conclusion and Future Work

4.1 Extensibility and Future Work

The system was intentionally designed with modularity and extensibility in mind, so that future contributors can build upon the existing architecture and expand the platform with new therapeutic protocols, interaction models, or agentic workflows. Rather than encoding each therapy as a monolithic block, the platform structures every intervention around two core components: a *logic* file that encodes the sequence of steps in the therapeutic flow, and a *prompt* file used to generate or support content through LLM-driven agent workflows. This separation of concerns simplifies maintenance, clarifies responsibilities, and provides a concrete template for adding new therapies following the same methodology.

Future collaborators can introduce additional therapies by creating a new logic module that defines the full patient interaction flow, as well as an associated prompt file that aligns with the style, structure, and conventions of the existing ones. Because the application backend exposes well-defined endpoints for each therapeutic function—documented in earlier chapters—developers can model their new endpoints after the current implementation for VNeST and Spaced Retrieval (SR), ensuring consistency across the system’s API design and behavior.

To fully integrate a new therapy into the platform, several coordinated updates are required across the system:

- **Mobile application.** A new therapy option must be added to the home screen, following the visual and navigational structure of the existing VNeST and Spaced Retrieval modules. Corresponding screens need to be registered in the navigation stack, and the new therapy logic must be connected to the appropriate backend endpoints so that content, feedback, and progress are synchronized in real time.
- **Database integration.** A new therapy type must be defined within the database schema, including storage formats for exercise instances, result logs, progress tracking, and any personalized content. This involves extending the Firestore collections (or subcollections) with additional fields or documents that capture therapy-specific data while preserving the overall structure described in the database design.
- **Web therapist interface.** The therapist dashboard must be expanded to include management tools for the new therapy. Clinicians and researchers should be able to assign exercises for this modality, monitor patient performance, and review detailed results in a way that is consistent with the existing modules. This may require adding new filters, tables, or detail views to the web interface.
- **Backend and agent workflows.** A new set of backend endpoints should be implemented following the conventions established for VNeST and SR,

including naming, authentication, and payload structure. If the therapy relies on adaptive or generated content, an additional agent workflow—comprising a dedicated prompt file and orchestration logic—must be created so that the LLM can generate or adapt exercises in line with the therapeutic goals.

The modular nature of the current system—spanning mobile, web, and back-end components—positions the project to grow organically as new contributors join. Clear boundaries between therapy logic, interface components, and AI-driven content generation make it possible to introduce new treatment protocols without disrupting existing functionality. This architecture also provides a natural entry point for future expansions such as advanced analytics dashboards, clinical data visualizations, or more autonomous AI agents that can recommend exercise adjustments based on aggregated performance data, as training data and therapeutic requirements evolve.

Overall, the project has been structured not as a closed, single-purpose system, but as a foundation for a broader collaborative effort. Its architecture encourages continued expansion, allowing researchers, clinicians, and engineers to extend the platform with additional evidence-based therapies, richer interaction models, and decision-support tools that support Spanish-speaking individuals with aphasia.

4.2 Security and Ethical Considerations

Given that the platform handles sensitive personal and potentially health-related information, security and ethical safeguards were an essential component of the system’s design. Although the current implementation focuses primarily on access control and secure user authentication, the architectural foundations support future expansion toward more comprehensive data protection measures aligned with clinical and regulatory expectations.

User Authentication and Access Control. The mobile application uses Firebase Authentication (FireAuth) to ensure that each patient can securely access their therapy content. This authentication mechanism protects user accounts through encrypted credential storage and industry-standard login flows. Similarly, the web platform for therapists also relies on email-based authentication, ensuring consistent access control across both interfaces and reducing the likelihood of weak or ad-hoc credential management.

Beyond basic authentication, therapist accounts undergo a manual verification process. When a user requests access as a therapist, system administrators review the request and either approve or deny it, based on the information provided. In both cases, the applicant receives an email notification informing them of the outcome. This additional verification step prevents unauthorized individuals from accessing patient data or clinical management features and reinforces the distinction between patient and therapist roles.

User Consent and Transparency. Because therapy personalization relies on autobiographical data provided by the patient, explicit user consent is required at the time of registration. Patients are informed that the information they provide will be used exclusively for generating tailored therapeutic exercises and for tracking their progress within the platform. No data is used for research purposes, shared with external entities, or processed beyond the scope of therapy delivery without explicit permission.

This focus on consent and transparency ensures that patients retain control over how their information is employed, and that they are aware of the direct link between their personal data and the personalization of their therapy content.

4.3 Conclusions

This work presents the design and implementation of a modular, extensible platform to support aphasia therapy in Spanish, addressing the general objective of creating a web and mobile ecosystem capable of generating, personalizing, and delivering evidence-based speech-language therapy through Large Language Models (LLMs) and agent-based workflows. The resulting system integrates a patient-facing mobile application, a therapist-facing web platform, and a set of AI-driven backend services that collectively operationalize two established therapeutic methods: Verb Network Strengthening Treatment (VNeST) and Spaced Retrieval (SR).

From the perspective of the specific objectives, the project successfully delivered a mobile application that enables patients to engage with interactive, structured VNeST and Spaced Retrieval exercises. The onboarding process gathers autobiographical information used for personalization, and the interfaces for each therapy were carefully designed to follow accessibility principles appropriate for individuals with aphasia—emphasizing low cognitive load, clear visual hierarchy, multimodal input, and immediate feedback. Each therapeutic protocol was translated into a sequence of concrete, screen-based interactions that support independent use while maintaining clinical fidelity.

The development of the therapist-focused web platform also met the objectives related to personalization, supervision, and monitoring. The web interface includes secure authentication and manual verification of therapist accounts, tools for managing patients and reviewing their exercises, mechanisms for assigning therapy tasks, and an AI-assisted workflow for generating new VNeST exercises. These features create a continuous loop in which therapists guide and refine therapy, patients complete personalized tasks, and the system records progress for potential adaptation. Although advanced analytics dashboards or long-term progress visualizations were not fully implemented, the platform establishes the functional core required for therapist monitoring.

With respect to agent-based workflows, the system incorporates prompt-based generation of VNeST structures, automated composition of personalized exercises, structured scheduling logic for SR intervals, and systematic logging of user performance through Firestore. The platform demonstrates how LLMs can be embedded into a clinical-adjacent architecture in a controlled, transparent

manner. By separating therapy logic from prompt design and exposing modular backend endpoints, the system supports future extensions—such as new therapies, updated prompts, or refined generation logic—without destabilizing existing functionality.

One objective that was only partially achieved concerns collaboration with speech-language therapy experts and individuals with aphasia. While the design and specification of the platform were informed by existing literature and informal consultations during the early stages of development, a formal evaluation study with clinicians or patients could not be conducted within the timeframe of this work. This limitation highlights a central avenue for future research: structured usability and acceptability testing with therapists and people with aphasia to assess the therapeutic validity, accessibility, and long-term feasibility of the platform in real-world scenarios.

Throughout the project, security and ethical considerations guided the treatment of personal and autobiographical data. The system incorporates secure authentication, restricted access to patient information, explicit user consent for personalization, and alignment with core ethical principles such as data minimization, purpose limitation, and transparency. Although the prototype does not yet implement clinical-grade security measures such as encrypted storage at rest, audit logging, or fully granular role-based access control, the underlying architecture supports their future integration.

In summary, this work provides a solid foundation for a multi-platform, AI-assisted aphasia therapy system tailored to Spanish-speaking users. It contributes: (1) a fully functional mobile application for VNeST and SR delivery; (2) a therapist web interface for supervision, assignment, and AI-assisted generation; (3) an extensible backend architecture that supports modular therapy logic and agent workflows; and (4) an initial security and ethics framework appropriate for medical-adjacent applications. Future work should focus on expanding therapeutic coverage, incorporating richer analytics and data visualizations for clinicians, conducting empirical evaluations with end users, and progressively strengthening security guarantees to support real-world clinical or community deployment.

References

- [1] K. Berg, J. Isaksen, S. J. Wallace, M. Cruice, N. Simmons-Mackie, and L. Worrall. Establishing consensus on a definition of aphasia: an e-delphi study of international aphasia researchers. *Aphasiology*, 36(4):385–400, 2020. doi:10.1080/02687038.2020.1852003.
- [2] Seçkin Arslan, Cecilia Devers, and Silvia Martínez Ferreiro. Pronoun processing in post-stroke aphasia: A meta-analytic review of individual data. *Journal of Neurolinguistics*, 59:101005, 2021. URL: <https://www.sciencedirect.com/science/article/pii/S091160442100021X>, doi: 10.1016/j.jneuroling.2021.101005.

- [3] R. S. Husak, S. E. Wallace, R. C. Marshall, and E. G. Visch-Brink. A systematic review of aphasia therapy provided in the early period of post-stroke recovery. *Aphasiology*, 37(1):143–176, 2021. doi:10.1080/02687038.2021.1987381.
- [4] Claudia Peñaloza, Manuel Jose Marte, Anne Billot, and Swathi Kiran. Cross-language interaction during sequential anomia treatment in three languages: Evidence from a trilingual person with aphasia. *Cortex*, 189:107–130, 2025. URL: <https://www.sciencedirect.com/science/article/pii/S0010945225001546>, doi:10.1016/j.cortex.2025.05.017.
- [5] S. Maity and M. J. Saikia. Large language models in healthcare and medical applications: A review. *Bioengineering*, 12(6):631, 2025. doi:10.3390/bioengineering12060631.
- [6] X. Tan, Y. He, Y. Zhou, X. Li, Q. Ding, Y. Tang, Y. L. L. Luo, and R. Gu. Born to fear the machine? genetic and environmental influences on negative attitudes toward ai agents. *Advanced Science*, page e06262, 2025. doi:10.1002/advs.202506262.
- [7] Giuliano Gallano, Andres Giglio, and Andres Ferre. Artificial intelligence in speech-language pathology and dysphagia: A review from latin american perspective and pilot test of llms for rehabilitation planning. *Journal of Voice*, 2025. URL: <https://www.sciencedirect.com/science/article/pii/S0892199725001584>, doi:10.1016/j.jvoice.2025.04.010.
- [8] Lisa A. Edmonds. Tutorial for verb network strengthening treatment (vnest): Detailed description of the treatment protocol with corresponding theoretical rationale. *Perspectives on Neurophysiology and Neurogenic Speech and Language Disorders*, 24(3):78–88, 2014. URL: <https://pubs.asha.org/doi/abs/10.1044/nnsld24.3.78>, doi:10.1044/nnsld24.3.78.
- [9] Cameron Camp, Vincent Antenucci, Jennifer Brush, and Thomas Slomin-ski. Using spaced retrieval to effectively treat dysphagia in clients with dementia. *Perspectives on Swallowing and Swallowing Disorders (Dysphagia)*, 21(3):96–104, 2012. URL: <https://pubs.asha.org/doi/abs/10.1044/sasd21.3.96>, doi:10.1044/sasd21.3.96.
- [10] Haley Hayashi, Michelle Gravier, Kristen Gustavson, and Ellen Bernstein-Ellis. Perspectives of u.s. speech-language pathologists on supporting the psychosocial health of individuals with aphasia. *Journal of Communication Disorders*, 105:106365, 2023. URL: <https://www.sciencedirect.com/science/article/pii/S0021992423000655>, doi:10.1016/j.jcomdis.2023.106365.
- [11] L. V. Forero García, M. P. Bernal Castilla, O. M. Aguilar Mejía, and Y. M. Quique Buitrago. Tratamientos para la afasia en hispanohablantes. *Revista*

de Investigación en Logopedia, 13(1):e81535, 2023. doi:10.5209/rlog.81535.

- [12] Fundación INECO. El 30 por ciento de los pacientes que sobreviven un acv tienen afasia. <https://www.fundacionineco.org/el-30-de-los-pacientes-que-sobreviven-un-acv-tienen-afasia/>, 6 2019. Accedido: 2025-08-29.
- [13] Ministerio de Salud y Protección Social de Colombia. Mujeres: las más afectadas por accidente cerebrovascular acv en colombia, según cifras del ministerio de salud y protección social. <https://www.minsalud.gov.co/Paginas/mujeres-las-mas-afectadas-por-accidente-cerebrovascular-en-colombia.aspx>. Accedido: 2025-08-29.
- [14] Babb M. Edmonds LA. Effect of verb network strengthening treatment in moderate-to-severe aphasia. *American Journal Speech-Language Pathology*, 2011. doi:10.1044/1058-0360(2011/10-0036).
- [15] Norvig P. Russell, S. J. Artificial intelligence: A modern approach. *Pearson Prentice Hall*, 2020.
- [16] RELEASE Collaborators. Dosage, intensity, and frequency of language therapy for aphasia: A systematic review-based, individual participant data network meta-analysis. *Stroke*, 53(3):956–967, 2022. doi:10.1161/STROKEAHA.121.035216.
- [17] Lisa A. Edmonds, Stephen E. Nadeau, and Swathi Kiran. Effect of verb network strengthening treatment (vnest) on lexical retrieval of content words in sentences in persons with aphasia. *Aphasiology*, 23(3):402–424, 2009. doi:10.1080/02687030802291339.
- [18] Erica L. Middleton, Katherine A. Rawson, and Jay Verkuilen. Retrieval practice and spacing effects in multi-session treatment of naming impairment in aphasia. *Cortex*, 119:386–400, 2019. URL: <https://www.sciencedirect.com/science/article/pii/S0010945219302552>, doi: 10.1016/j.cortex.2019.07.003.
- [19] Jennifer A. Brush and Cameron J. Camp. Using spaced retrieval as an intervention during speech-language therapy. *Clinical Gerontologist*, 19(1):51–64, 1998. doi:10.1300/J018v19n01_05.
- [20] Julius Fridriksson, Audrey Holland, Pelagie Beeson, and Lisa Morrow. Spaced retrieval treatment of anomia. *Aphasiology*, 19(2):99–109, 2005. doi:10.1080/02687030444000660.
- [21] Yuliana M. Quique, Adriana Marcela-Martínez, Gabriela Almeida, Laura Chávez, and Oscar M. Aguilar. Examining aphasia rehabilitation in

- colombia from the perspective of speech-language pathologists and neuropsychologists: a pilot study. *Aphasiology*, pages 1–28, 2025. doi: 10.1080/02687038.2025.2519332.
- [22] Caitlin Brandenburg, Linda Worrall, Amy D. Rodriguez, and David Copland. Mobile computing technology and aphasia: An integrated review of accessibility and potential uses. *Aphasiology*, 27(4):444–461, 2013. doi:10.1080/02687038.2013.772293.
 - [23] M. A. Aziz et al. Development of speech therapy mobile application for aphasia patients. In *2021 IEEE National Biomedical Engineering Conference (NBEC)*, pages 89–94, Kuala Lumpur, Malaysia, 2021. IEEE. doi:10.1109/NBEC53282.2021.9618759.
 - [24] Nuno Nogueira, Henrique Pereira S. Mamede, Vitor Santos, Pedro Maia Malta, and Carolina Santos. Developing a computer system prototype to support aphasia rehabilitation. In *Proceedings of the 10th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-Exclusion*, DSAI ’22, pages 18–24, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3563137.3563172.
 - [25] A. Adikari, N. Hernandez, D. Alahakoon, M. L. Rose, and J. E. Pierce. From concept to practice: a scoping review of the application of ai to aphasia diagnosis and management. *Disability and Rehabilitation*, 46(7):1288–1297, 2023. doi:10.1080/09638288.2023.2199463.
 - [26] N. Azevedo, E. Kehayia, G. Jarema, G. Le Dorze, C. Beaujard, and M. Yvon. How artificial intelligence (ai) is used in aphasia rehabilitation: A scoping review. *Aphasiology*, 38(2):305–336, 2023. doi:10.1080/02687038.2023.2189513.
 - [27] Swathi Kiran and Zachary M. Smith. The efficacy and utility of constant therapy in poststroke aphasia rehabilitation. *Perspectives of the ASHA Special Interest Groups*, 2025. URL: https://pubs.asha.org/doi/abs/10.1044/2025_PERSP-24-00260, doi:10.1044/2025_PERSP-24-00260.
 - [28] D. S. Barbera, M. Huckvale, V. Fleming, E. Upton, H. Coley-Fisher, C. Doogan, I. Shaw, W. Latham, A. P. Leff, and J. Crinion. Nuva: A naming utterance verifier for aphasia treatment. *Computer Speech & Language*, 69:101221, 2021. doi:10.1016/j.csl.2021.101221.
 - [29] Rong Tong, Farhan Azmi, Xing Long He, Han Xiang Kee, Ying Zhen Lee, and Wen Qin Yeo. Chataphasia: A personalized end-to-end system for aphasia therapy. In *Conference contribution, Singapore Institute of Technology*, 2025. doi:10.25447/sit.29163800.v1.
 - [30] S. S. Mahmoud, R. F. Pallaud, A. Kumar, S. Faisal, Y. Wang, and Q. Fang. A comparative investigation of automatic speech recognition

- platforms for aphasia assessment batteries. *Sensors*, 23:857, 2023. doi: 10.3390/s23020857.
- [31] D. J. Bailey, F. Herget, D. Hansen, F. Burton, G. Pitt, T. Harmon, and D. Wingate. Generative ai applied to aac for aphasia: a pilot study of aphasia-gpt. *Aphasiology*, pages 1–16, 2024. doi:10.1080/02687038.2024.2445663.

A Mobile Application Flows and Screens

This annex presents the complete interaction flows and associated user interface screens for the patient mobile application. The purpose of this section is to complement the functional overview by providing detailed visual references of the registration process, VNeST therapy sequence, Spaced Retrieval (SR) progression, and exercise personalization workflow. Each subsection below includes structured placeholders for screenshots and navigation diagrams.

A.1 Registration Flow

The registration process guides new users through a linear, single-step-per-screen onboarding sequence. Screens capture autobiographical information that is later used to generate personalized exercises.

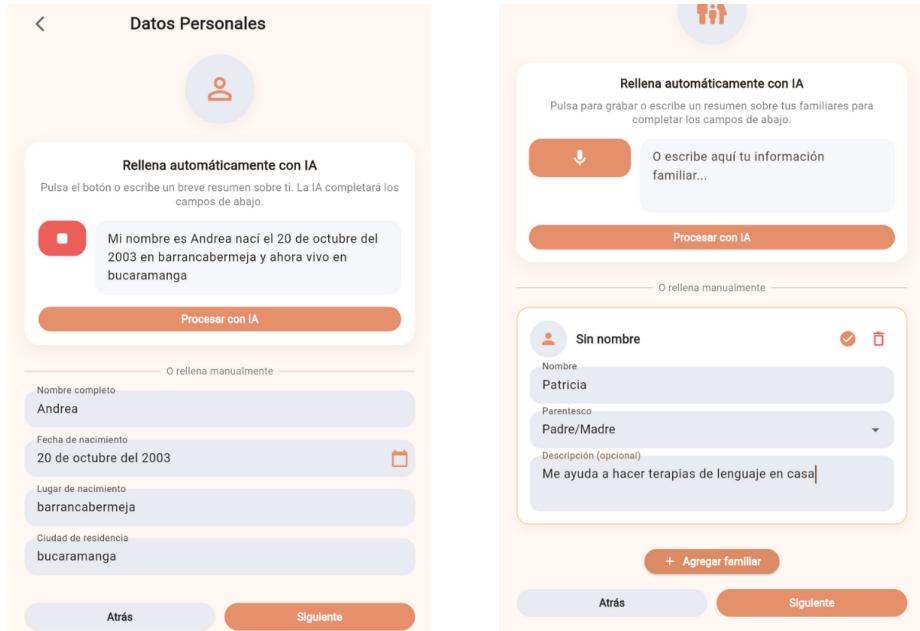


Figure 20: Personal data and family members fill screens.

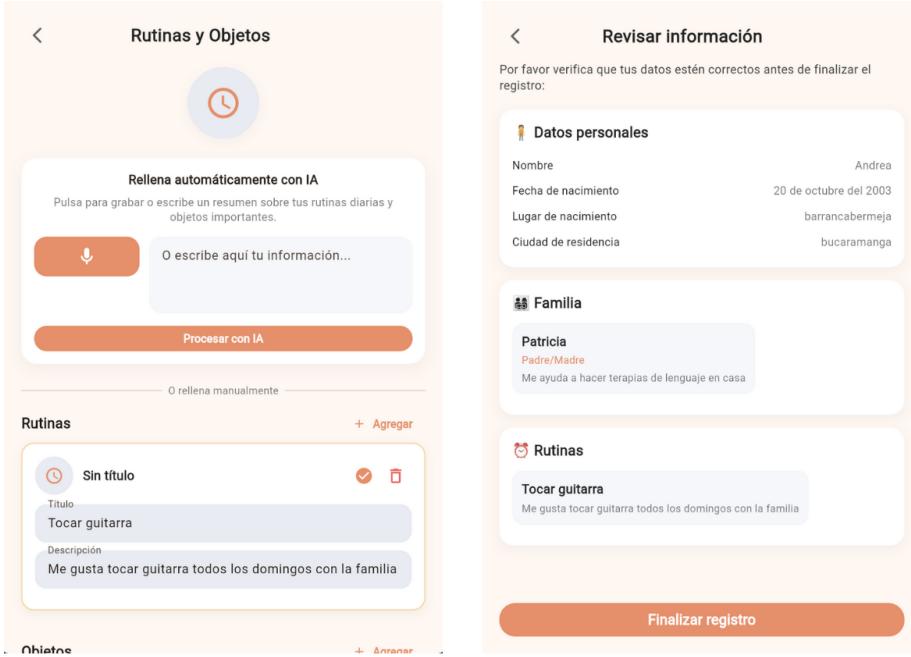


Figure 21: Objects and routines fill screens and summary screen.

A.2 VNeST Flow

The VNeST module contains multiple therapeutic phases. Each step is designed to minimize cognitive load and maintain a consistent sentence-building context.

A.2.1 Context and Verb Selection

The patient first selects a context, then the corresponding verbs. This anchors semantic expectations and guides the start of the VNeST protocol.

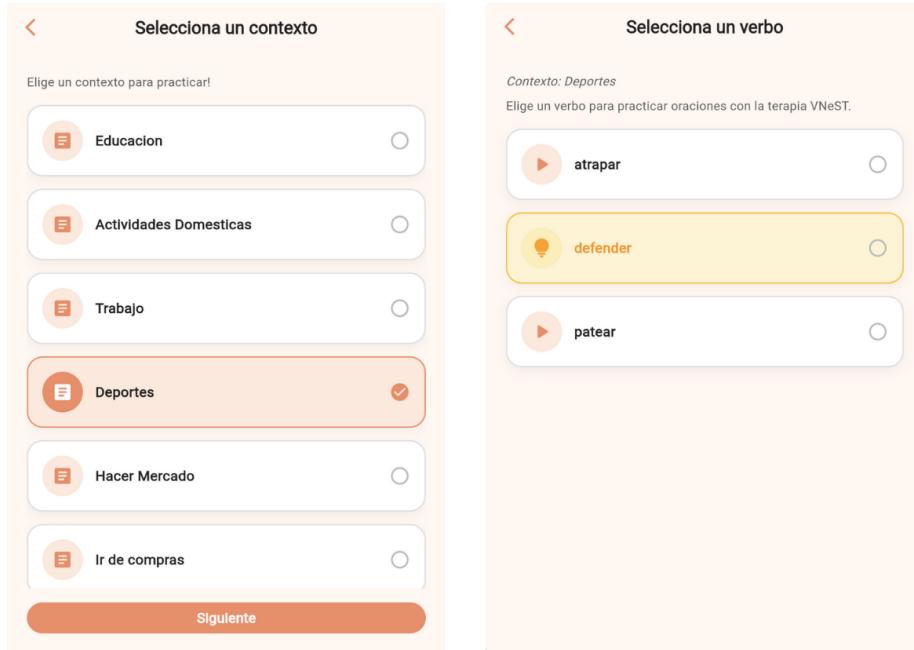


Figure 22: Context and verb selection screens in VNeST.

A.2.2 Phase 1: Agent–Object Generation

The user selects a valid agent and object pair associated with the chosen verb. Only semantically valid combinations allow progression.

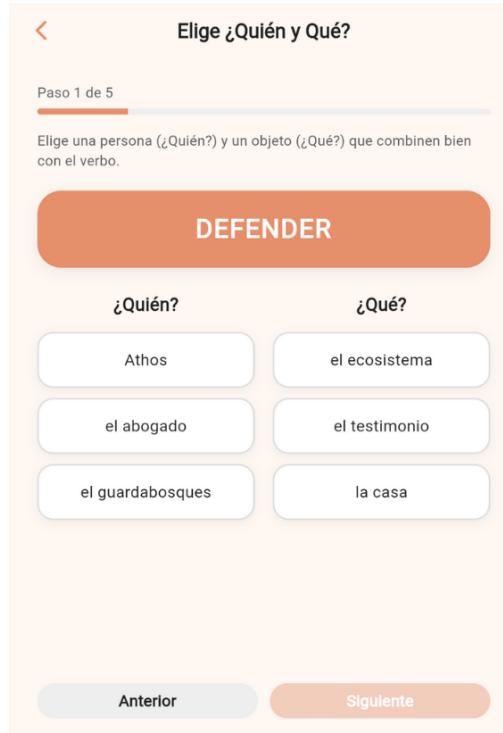


Figure 23: Phase 1: Selecting agent and object for the verb.

A.2.3 Phase 2: Contextual Expansion

Contextual elements (*where*, *when*, *why*) are added using single-choice questions. The sentence expands progressively with correct selections.

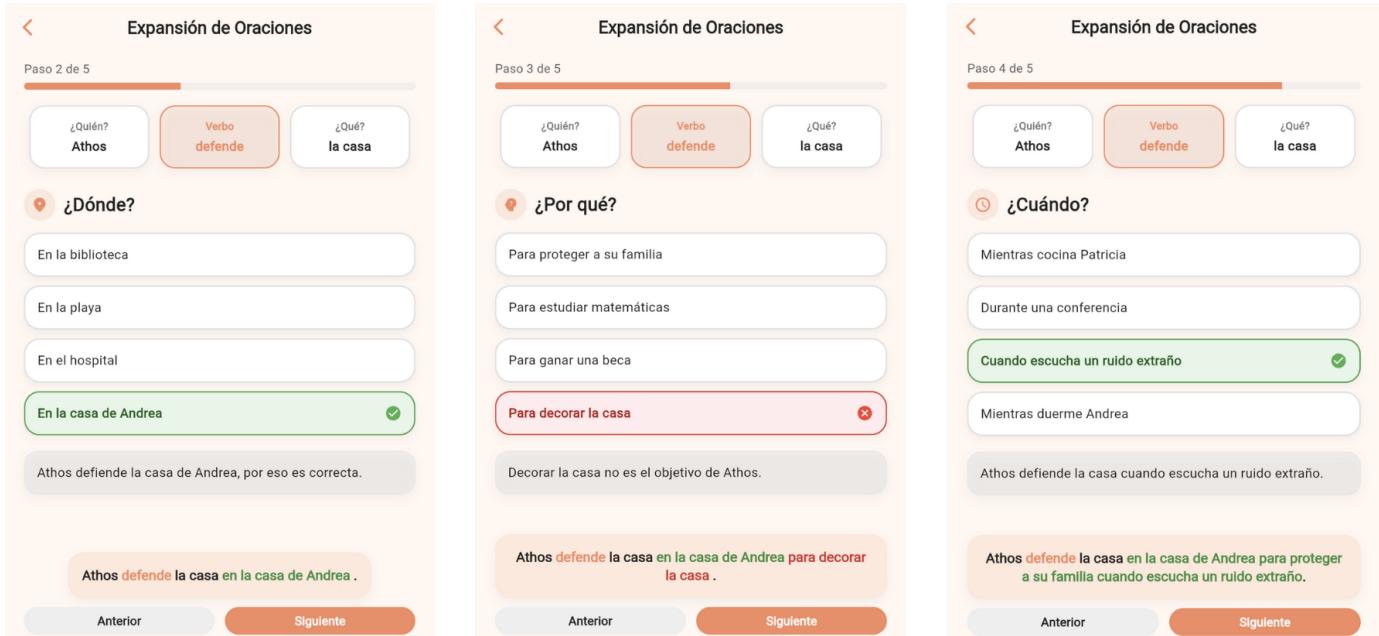


Figure 24: Phase 2: Contextual expansion of the sentence.

A.2.4 Phase 3: Plausibility Judgments

The user classifies 10 sentences as plausible or implausible by swiping right or left. Feedback is provided immediately for incorrect decisions.

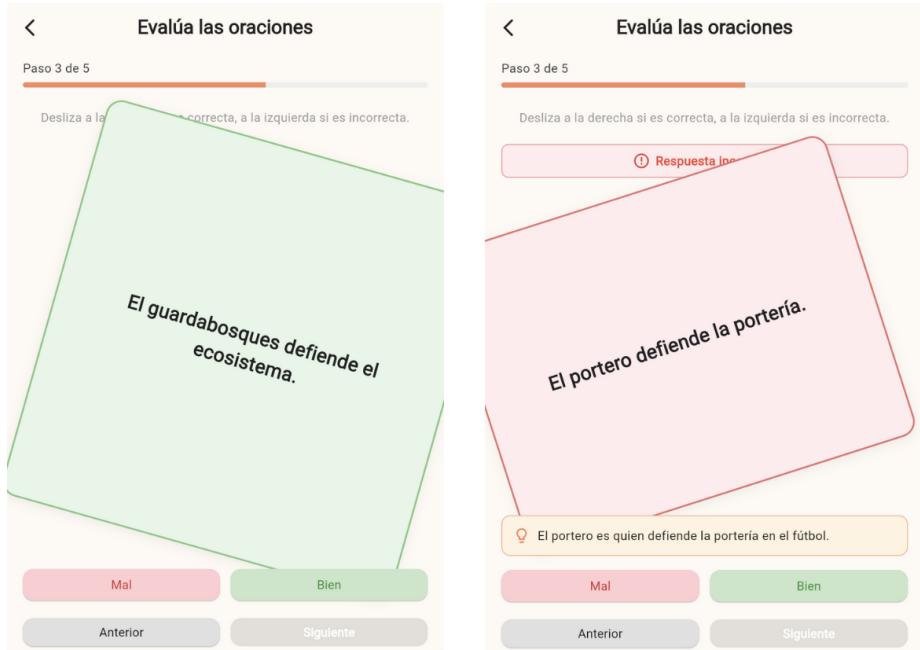


Figure 25: Phase 3: Plausibility judgment swipe interface.

A.2.5 Final Summary

A final screen displays the completed verb, agent, object, and expanded sentence.



Figure 26: Summary of plausibility judgments after 10 items and of therapy overall

A.3 Spaced Retrieval Flow

The SR module presents incremental recall intervals, doubling the delay after each correct answer and resetting to the last successful interval after an incorrect one.

A single question appears at each interval. The patient responds by typing or using voice input. Correct answers double the interval; incorrect answers reset to the previous successful interval. After five successful intervals, the exercise ends with a completion and reinforcement screen.

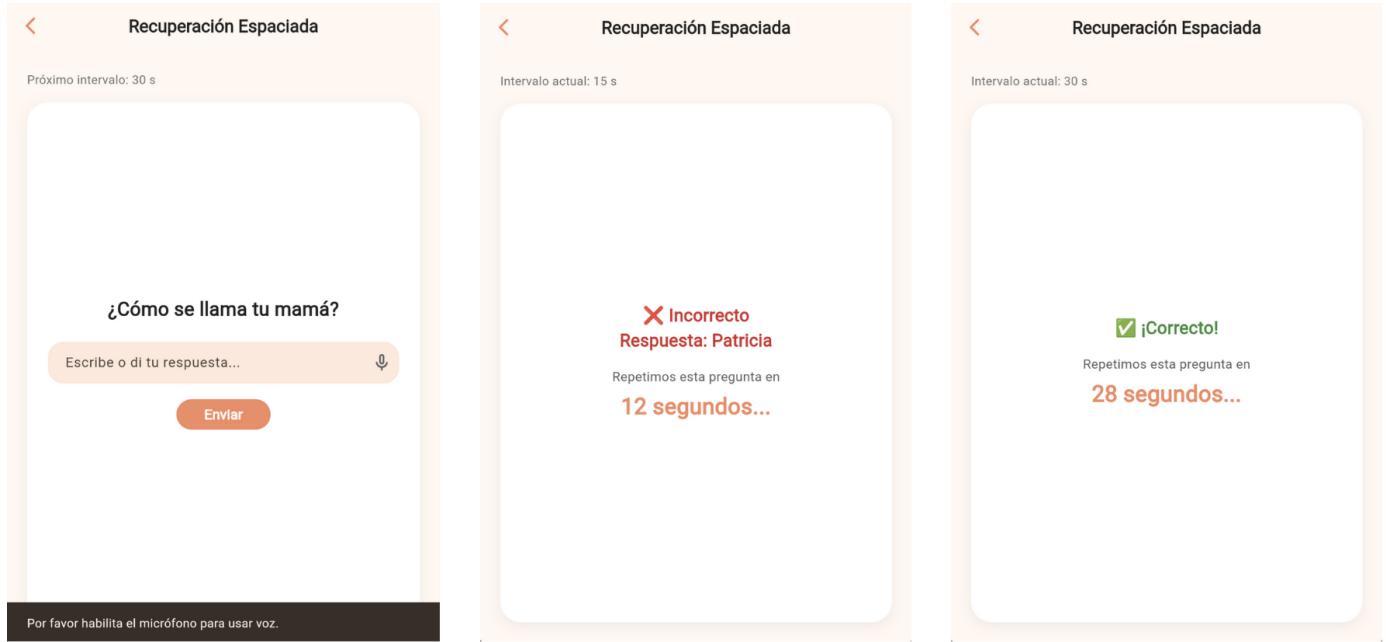


Figure 27: SR input screen with typing and audio options.

A.4 Exercise Personalization Flow

This module allows patients to generate custom exercises based on their auto-biographical profile.

Contexts appear first; selecting one expands the list of associated verbs in-line. After selecting a verb, a personalized exercise is generated using profile data. The patient may open the generated exercise or return to the main screen.

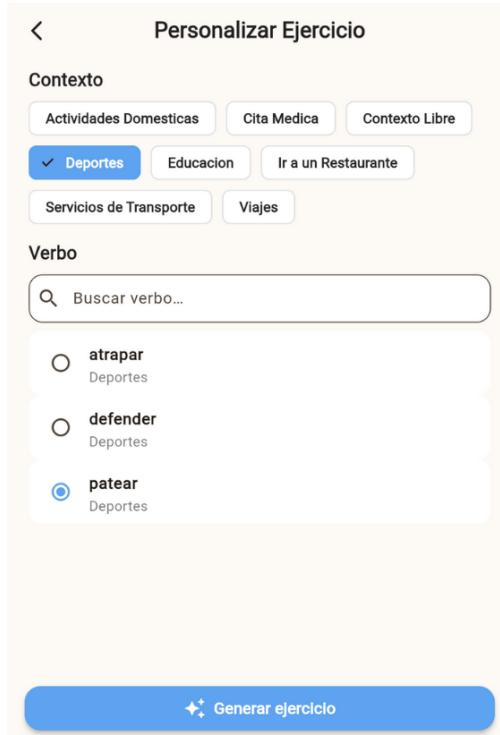


Figure 28: Context and verb selection in exercise personalization.

B Web Application Flows and Interfaces

This annex presents the main interaction flows and graphical interfaces of the web platform used by therapists and administrators. The goal is to complement the functional description of the system with concrete visual references of the login process, therapist onboarding workflow, dashboard navigation, patient and exercise management, and administrative approval of new therapist requests.

Each subsection includes a structured description of the flow and placeholders for the corresponding interface screenshots.

B.1 Login and Therapist Registration Flow

Therapists and administrators access the platform through a unified login page. From this screen, new therapists may submit a registration request, which is later evaluated by the administrator. The process consists of:

1. Opening the login page and entering credentials.
2. Navigating to the registration form (for new therapists).

3. Completing personal and professional information.
4. Submitting the request.
5. Waiting for administrative approval before accessing the platform.

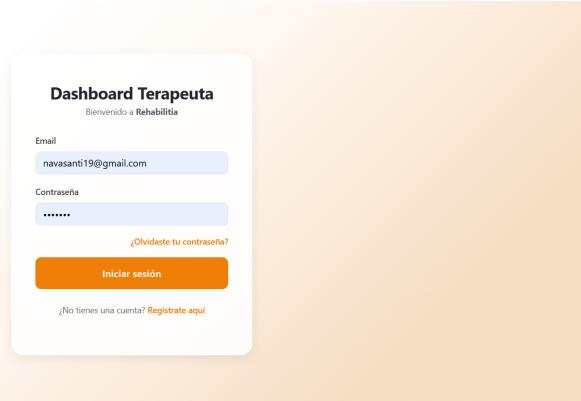


Figure 29: Login screen for therapists and administrators.

The registration form is titled 'Registro de Terapeuta' and includes the instruction 'Envía tu solicitud para ser parte de Rehabilitis'. It features fields for 'Nombre completo' (Santiago), 'Email' (navasanti19@gmail.com), 'Celular' (3212346442), 'Profesión' (Fonoaudiólogo), and a motivation text area. The motivation text reads: 'Por qué quieres ser parte de esta comunidad? Llevo trabajando con personas con Afasia desde hace más de 10 años. Tengo mucha experiencia y varios pacientes que están usando la aplicación'. At the bottom is an orange 'Enviar solicitud' button.

Figure 30: Therapist registration form with personal, contact, and motivation fields.

B.2 Therapist Dashboard Flow

The dashboard is the main entry point for therapists after authentication. It provides a high-level overview of key metrics, including the number of associated

patients and pending exercises to review. From the dashboard, the therapist can navigate to patient management, exercise review, or exercise creation.

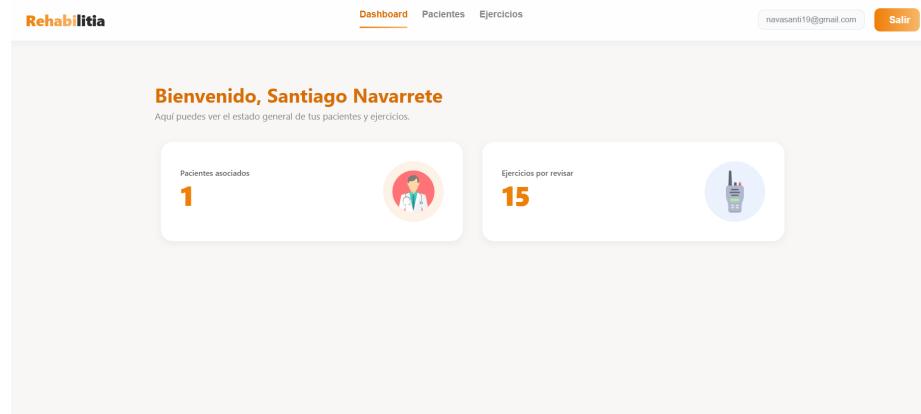


Figure 31: Therapist dashboard showing patient and exercise summary cards.

B.3 Patient Management Flow

Therapists can view and manage their assigned patients through the “Pacientes” section. The core flow includes:

1. Opening the patient list.
2. Selecting a patient to view their details.
3. Filtering and searching assigned exercises.
4. Opening individual exercises for detailed review.

Pacientes		EJERCICIOS ASIGNADOS		ACCIÓN
NOMBRE	EMAIL			
Andrea Galindo	andrealgalindoc@gmail.com	15		Ver detalles

Figure 32: List of assigned patients and navigation to the detailed patient view.

Ejercicios de Andrea Galindo		+ Crear Ejercicio Personalizado	+ Asignar Ejercicio					
VNEST	SR							
Estado:	Verbo:	Contexto:						
<input type="button" value="Todos"/>	<input type="button" value="Buscar verbo"/>	<input type="button" value="Buscar contexto"/>	<input type="button" value="Limpiar X"/>					
ID	CONTEXTO	VERBO	PERSONALIZADO	NIVEL	ESTADO	FECHA ASIGNADO	FECHA REALIZADO	ACCIÓN
EBA7695	Deportes	defender	Sí	difícil	COMPLETADO	16/11/2025	16/11/2025	Ver
E66C51D	Viajes	visitar	Sí	medio	PENDIENTE	5/11/2025	—	Ver
E286C3F	Servicios de Transporte	entregar	No	medio	PENDIENTE	5/11/2025	—	Ver

Figure 33: Patient exercise list with filters for state, verb, and context.

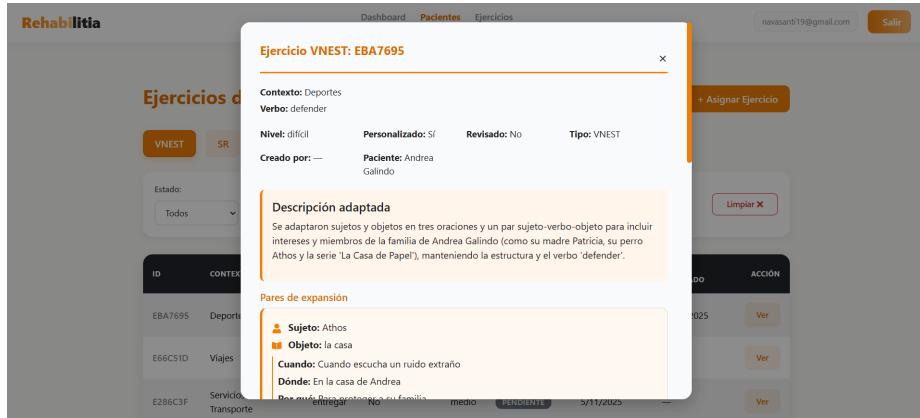


Figure 34: Exercise detail view

B.3.1 Personalizing a VNeST Exercise for a Patient

Within the patient detail view, therapists may generate a personalized VNeST exercise based on the patient's autobiographical profile. The flow is:

1. Opening the patient detail page.
2. Clicking the “Crear Ejercicio Personalizado” button.
3. Selecting a context from the list of semantic categories.
4. Reviewing the automatically generated exercise using patient-specific information.
5. (Optional) Assigning the personalized exercise directly to the patient.

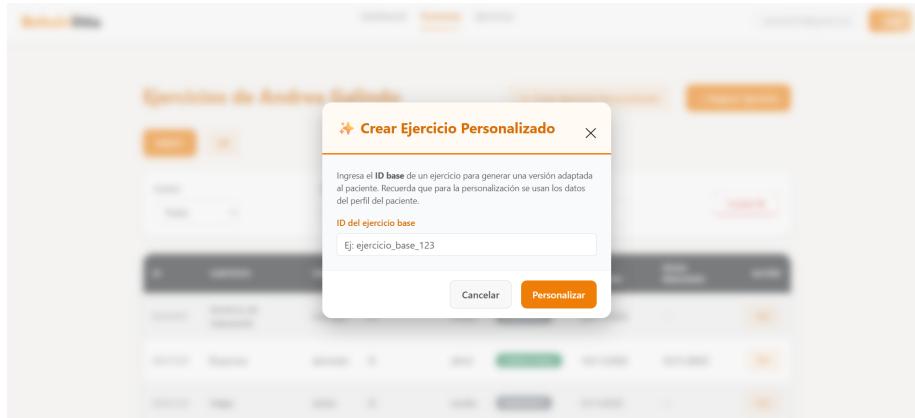


Figure 35: Flow for generating a personalized VNeST exercise for a patient.

B.3.2 Assigning an Existing Exercise to a Patient

Therapists can also assign an existing exercise—public or previously created—to a selected patient. The assignment workflow consists of:

1. Opening the patient detail screen.
2. Clicking the “Asignar Ejercicio” button.
3. Searching or filtering exercises by visibility, context, verb, or level.
4. Selecting an exercise from the list.
5. Confirming the assignment so it appears immediately in the patient’s dashboard.

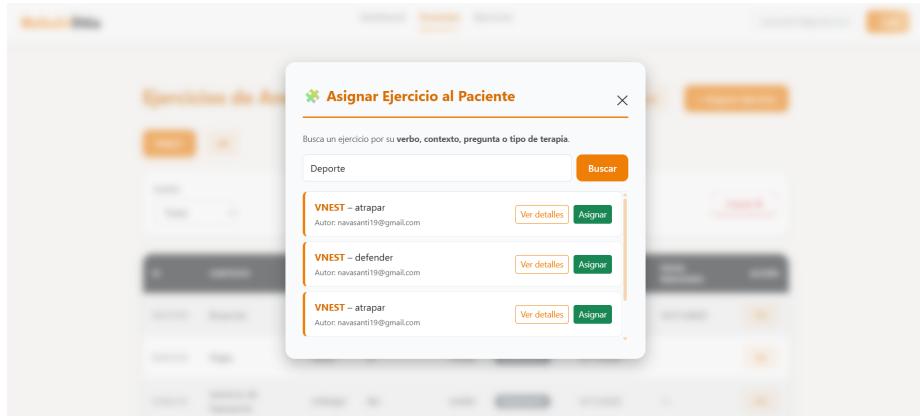


Figure 36: Flow for assigning an existing exercise to a patient.

B.4 Exercise Management and Review Flow

The “Ejercicios” section allows therapists to supervise all exercises created in the system. This includes public templates, personalized items, and patient-assigned exercises.

The interaction flow is:

1. Navigating to the exercise list.
2. Applying filters (visibility, status, personalization, verb, context, patient).
3. Opening a specific exercise.
4. Reviewing correctness, content structure, and linguistic appropriateness.
5. Marking the exercise as reviewed.

Figure 37: Exercise management interface with filtering options.

B.5 AI-Assisted Exercise Creation Flow

Therapists may generate new exercises using the built-in AI engine. This automated process requires completing a minimal form and allows the creation of both public and private VNeST exercises.

1. Opening the “Nuevo Ejercicio” page.
2. Selecting ”Generar con IA” option
3. Selecting a context, difficulty level, and visibility.
4. Triggering the generation process.
5. Reviewing the resulting exercise and optionally assigning it to a patient.

Figure 38: AI-assisted option.

Figure 39: AI-assisted VNeST exercise creation form.

B.6 Administrator Approval Flow

Administrators manage the approval of new therapist registration requests. The flow consists of:

1. Opening the “Panel de Administración”.
2. Viewing pending requests.
3. Opening an individual request to review the applicant’s information.
4. Approving or rejecting the request.
5. Tracking approved and rejected requests in separate categories.

Figure 40: Administrator panel showing pending and completed therapist requests.

C Prompt templates for agent flows

C.1 Prompt Templates for the VNeST LangGraph Agent

This appendix contains the full set of prompt templates used by the VNeST generation agent. Each prompt corresponds to one stage of the LangGraph workflow described in Section ???. The templates are shown here in their original form for transparency and reproducibility.

Prompt 1: Verb Generation

generate_verb_prompt

```
Dado el siguiente contexto: '{contexto}'. Genera una lista de
exactamente 7 verbos
transitivos que cumplan:
0) Deben ser verbos que puedan claramente usarse en el contexto dado;
1) Requieren complemento directo;
2) Son cotidianos y familiares;
3) No son genéricos (evita 'hacer', 'tener', 'llevar');
4) Son diferentes entre sí; 5) En infinitivo; 6) Mezcla de
dificultades.
Responde SOLO con JSON válido, sin texto adicional.
Formato: {"contexto":"string","verbos":["v1","v2","v3","v4",
"v5","v6","v7"]}
```

Prompt 2: Verb Classification by Difficulty

verb_by_difficulty

```
Toma este JSON y clasifica los verbos por dificultad.
Responde SOLO con JSON válido.
Reglas:
- Fácil = 1{2 sílabas
- Medio = 2{3 sílabas
- Difícil = 3+ sílabas
Debe haber distribución equilibrada.
Formato de salida:
{"contexto":"string","verbos_clasificados":{"facil":[...],
"medio":[...],"difícil":[]}}}
```

Prompt 3: SVO Pair Selection

pair_subject_object

Toma un ÚNICO verbo del nivel indicado y genera oraciones
(sujeto+verbo+objeto)
cumpliendo:
- EXACTAMENTE 3 oraciones
- Todas usan el mismo verbo
- Prototipicidad del rol agente/paciente
- Una oración debe relacionarse con el contexto dado
- Oraciones disyuntivas (no intercambiables)
Salida SOLO JSON.

Prompt 4: Sentence Expansion

sentence_expansion

Para cada par SVO, generar:
- 3 interrogantes: ¿Dónde?, ¿Cuándo?, ¿Por qué?
- Cada interrogante con 4 opciones, 1 correcta
- Explicaciones cortas sin comillas dobles
Salida SOLO JSON.

Prompt 5: Final Sentence Set Generation

generate_prompt

Agregar al JSON una última sección "oraciones":
- EXACTAMENTE 10 oraciones SVC
- Algunas correctas, otras incorrectas
Formato:
{"verbo":"...","pares":[...],"oraciones":[]}

Prompt Template for the Spaced Retrieval Agent

The following prompt is used by the SR agent to generate autobiographical memory-recall cards based on the patient's structured profile. The template implements explicit constraints on output format, question type, and permitted categories to ensure consistent, clinically meaningful results.

generate_sr_prompt

Eres un terapeuta del lenguaje. Tu tarea es crear ejercicios de Spaced Retrieval para un paciente con afasia, usando SOLO su información autobiográfica.

Perfil del paciente:
{ ... JSON del perfil del paciente ... }

Genera EXACTAMENTE 5 preguntas cortas, simples y claras en español, con su respuesta.
Las preguntas deben ser de tipo autobiográfico (ej: "¿Cómo se llama tu hijo?", "¿Dónde naciste?").

Formato de salida:
{
 "cards": [
 { "stimulus": "string",
 "answer": "string",
 "category": "string"
 }
]
}

Las categorías permitidas son: 'personal', 'familia', 'rutina', 'objetos'.
NO incluyas texto adicional ni explicaciones, solo JSON válido.

Prompt Template for the Personalization Agent

The following prompt is used by the personalization agent to adapt a base exercise (VNeST or SR) to the linguistic and autobiographical context of a specific patient. The template enforces preservation of exercise structure, controlled adaptation rules, and complete JSON validity in the final output.

generate_personalization_prompt

Eres un terapeuta experto en afasia y personalización de ejercicios de lenguaje.

Tienes un ejercicio base y el perfil de un paciente.
Tu tarea es adaptar el contenido del ejercicio al contexto del paciente, manteniendo la estructura y los campos originales.

Perfil del paciente:
{ ... JSON del perfil del paciente ... }

Ejercicio base:
{ ... JSON del ejercicio base ... }

Instrucciones:
1. Reemplaza nombres genéricos ("el agricultor", "la persona",

"el cliente") por miembros de la familia, lugares, rutinas o intereses del paciente.

2. En la sección de pares sujeto-verbo-objeto: cambia ÚNICAMENTE 1 sujeto y/o 1 objeto para que reflejen el contexto del paciente. Los pares deben seguir siendo coherentes con el verbo.
3. En la sección de oraciones: adapta ÚNICAMENTE 3 oraciones para que reflejen el contexto del paciente. Mantén las demás sin cambios.
4. No cambies el nivel ni el tipo del ejercicio.
5. Devuelve un JSON con la misma estructura, pero con el contenido adaptado.
6. Incluye estos campos adicionales:
 - "id_paciente": "<USER_ID>"
 - "personalizado": true
 - "referencia_base": "<ID_DEL_EJERCICIO_BASE>"
 - "descripcion_adaptado": breve explicación de cómo se adaptó.
7. No cambies el contexto original: debe mantenerse igual al proporcionado en "context_hint".

La salida debe ser SIEMPRE JSON válido. No incluyas explicaciones adicionales.

Prompt Template for the Profile Structuring Agent

The following prompt is used by the Profile Structuring agent to transform free-form autobiographical text provided during registration into a structured JSON profile with the fields required by the system. The template enforces semantic consistency, relationship constraints, and strict JSON formatting to ensure downstream compatibility.

generate_profile_structure_prompt

Eres un asistente experto en estructurar información personal y clínica de pacientes con afasia. Recibirás un texto libre con información autobiográfica y tu tarea será transformarlo en un JSON estructurado con los siguientes campos.

El campo 'tipo_relacion' de 'familia' debe ser uno de: ["Cónyuge/Pareja", "Hijo/a", "Padre/Madre", "Hermano/a", "Otro"]. El campo 'tipo_relacion' de 'objetos' puede ser libre. En la descripción de familia, no menciones el tipo de relación ya indicado en 'tipo_relacion'.

Estructura esperada:

```
{
  "personal": {
    "nombre": "",
    "fecha_nacimiento": "",
```

```
        "lugar_nacimiento": "",  
        "ciudad_residencia": ""  
    },  
    "familia": [  
        {"nombre": "", "tipo_relacion": "", "descripcion": ""}  
    ],  
    "rutinas": [  
        {"titulo": "", "descripcion": ""}  
    ],  
    "objetos": [  
        {"nombre": "", "tipo_relacion": "", "descripcion": ""}  
    ]  
}
```

Instrucciones:

- Usa SOLO la información presente en el texto.
- Si algún campo no se menciona, déjalo vacío o con lista vacía.
- Mantén coherencia semántica (no pongas una ciudad como nombre de persona).
- Si el texto menciona varias personas, objetos o actividades, inclúyelos.
- Devuelve ÚNICAMENTE JSON válido, sin texto adicional.

ID del paciente: <USER_ID>

Texto recibido:

```
"""<TEXTO_DEL_PACIENTE>"""
```