

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA E INFORMÁTICA
CONCEPCIÓN - CHILE



**“DISEÑO E IMPLEMENTACIÓN DEL FRONTEND DE
EVERPLANAPP: APLICACIÓN DE ORGANIZACIÓN CON
ENFOQUE INCLUSIVO”**

MICHELLE MABELLE ROCHA ESPINOZA

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA EN INFORMÁTICA**

Profesor Guía: Cristian Antonio Lara Valenzuela

Junio - 2026

DEDICATORIA

A mi mami Sussy, que siempre está lista para celebrar cada uno de mis logros y preparada para acompañarme en todo lo que venga después. Su amor, y su forma de estar presente en toda mi vida han hecho posible cada paso que he dado. Nada de lo que he conseguido ha sido solo mío. Cada logro mío es también un logro de ella, porque en cada esfuerzo, decisión y meta alcanzada está su apoyo, su enseñanza y su amor.

A Marco Riffo, por su amor, su compañía y su apoyo constante. Entre todas las personas del mundo, eligió caminar a mi lado, sostenerme en los momentos difíciles y ser un ánimo permanente en este proceso y en la vida.

Finalmente, a todas las madres y padres que me criaron y acompañaron. Dicen que para criar a un niño se necesita un pueblo, y en mi historia eso es completamente cierto. El cariño y cada valor inculcado, cada cuidado y cada enseñanza recibida han marcado profundamente a la profesional y a la persona que soy hoy.

“Llevo tu corazón conmigo (lo llevo en mi corazón);
nunca estoy sin él, tú vas dondequiera que yo voy,
y todo lo que hago por mí misma lo haces tú también, amada mía.” -E. E. Cummings

AGRADECIMIENTOS

Quiero agradecer a Cristian Lara por su orientación y apoyo durante estos años de universidad. Valoro especialmente la confianza que tuvo en mí al darme responsabilidades y permitirme participar en distintas actividades, lo que me ayudó a ganar seguridad y a sentir que podía asumir desafíos importantes. Esa experiencia fue clave para mi formación y crecimiento profesional.

A Viviana Delgado, quiero agradecerle de una forma muy especial. Siempre estuvo dispuesta a escuchar, a alentar y a felicitar cuando era necesario. Su apoyo constante, su preocupación genuina y su forma tan cercana de acompañar hicieron que muchas veces me sintiera contenida y motivada, casi como el apoyo de una mamá dentro de la universidad. Su presencia fue muy importante a lo largo de este proceso.

Agradezco profundamente a Eduardo Urrutia, por estar siempre disponible para apoyar y orientar en lo que fuera necesario. Su disposición constante, la forma en que entregó las herramientas y recursos justo como se necesitaban, y la confianza al considerarme en distintas ayudantías fueron un apoyo muy importante durante mi vida universitaria.

Finalmente, quiero agradecer a mis amigos, por estar presentes durante todo este camino. Las risas, los buenos momentos, el apoyo mutuo y la compañía hicieron que la carga académica fuera más llevadera y que esta etapa se recuerde no solo por el esfuerzo, sino también por todo lo alcanzado.

.

RESUMEN

Resumen— Esta memoria presenta el desarrollo del frontend de EverPlanApp, una aplicación orientada a apoyar la organización personal y la autorregulación de estudiantes, con especial énfasis en personas neurodivergentes. El trabajo se centró en el diseño e implementación de una interfaz móvil accesible, clara y de bajo estímulo cognitivo, que permita gestionar tareas, rutinas, horarios y herramientas de apoyo a la concentración y el bienestar emocional.

La solución fue desarrollada utilizando React Native y TypeScript, incorporando una arquitectura basada en componentes reutilizables y flujos de navegación diferenciados según el estado de autenticación del usuario. Se implementó la integración con un backend mediante consumo de APIs REST, permitiendo la gestión de información en tiempo real, así como el uso de notificaciones, temporizadores y elementos visuales de apoyo como pictogramas. El desarrollo siguió un enfoque incremental e iterativo, apoyado por control de versiones con Git y GitHub, prototipado en Figma y retroalimentación continua de usuarios y profesionales del área de inclusión.

Como resultado, se obtuvo un frontend funcional, modular y escalable, alineado con principios de accesibilidad y diseño inclusivo. La aplicación pretende ser una herramienta válida para mejorar la organización, reducir la carga cognitiva y apoyar procesos de autorregulación, destacando su potencial como apoyo tecnológico en contextos educativos y personales.

Palabras Clave— Frontend móvil; accesibilidad; neurodiversidad; organización personal; React Native.

ABSTRACT

Abstract— This thesis presents the development of the frontend of EverPlanApp, an application designed to support personal organization and self-regulation, with a particular focus on neurodivergent students. The work focused on the design and implementation of an accessible, clear, and low cognitive-stimulation mobile interface that enables task management, routines, schedules, and tools to support concentration and emotional well-being.

The solution was developed using React Native and TypeScript, incorporating a component-based architecture and differentiated navigation flows depending on the user's authentication status. Integration with a backend was implemented through REST API consumption, allowing real-time data management, as well as the use of notifications, timers, and visual support elements such as pictograms. The development followed an incremental and iterative approach, supported by version control with Git and GitHub, prototyping with Figma, and continuous feedback from users and inclusion professionals.

As a result, a functional, modular, and scalable frontend was obtained, aligned with accessibility and inclusive design principles. The application proved to be a valid tool for improving organization, reducing cognitive load, and supporting self-regulation processes, highlighting its potential as a technological aid in educational and personal contexts.

Keywords— Mobile frontend; accessibility; neurodiversity; personal organization; React Native.

GLOSARIO

API REST: Interfaz que permite la comunicación entre el frontend y el backend mediante solicitudes HTTP, utilizando métodos estándar como GET, POST, PUT y DELETE.

Arquitectura cliente-servidor: Modelo de diseño donde el frontend actúa como cliente solicitando servicios a un backend que procesa la información y responde.

Autenticación: Proceso mediante el cual se verifica la identidad de un usuario antes de permitir el acceso a las funcionalidades de la aplicación.

Autorregulación: Capacidad de una persona para gestionar su comportamiento, emociones y tiempo, apoyándose en herramientas que faciliten la organización y el autocontrol.

Backend: Parte del sistema encargada de la lógica de negocio, procesamiento de datos y gestión de solicitudes provenientes del frontend.

Carga cognitiva: Cantidad de esfuerzo mental requerido para procesar información o realizar una tarea determinada.

Componente: Unidad reutilizable de la interfaz que encapsula lógica y presentación dentro de la aplicación.

Control de versiones: Práctica que permite registrar, gestionar y recuperar cambios realizados en el código fuente durante el desarrollo.

Drawer (menú lateral): Elemento de navegación que permite acceder rápidamente a distintas secciones de la aplicación mediante un menú desplegable.

Estado de la aplicación: Información que representa la situación actual de la aplicación y que influye en el comportamiento de las vistas y componentes.

Experiencia de usuario (UX): Percepción general del usuario al interactuar con la aplicación, considerando facilidad de uso, claridad y satisfacción.

Frontend: Parte de la aplicación con la que interactúa directamente el usuario, incluyendo pantallas, botones y flujos de navegación.

Iterativo e incremental: Enfoque de desarrollo donde el sistema se construye por etapas, agregando y mejorando funcionalidades de forma progresiva.

Neurodivergencia: Concepto que reconoce la diversidad en el funcionamiento neurológico, incluyendo condiciones como TDAH y TEA.

Pictogramas: Representaciones visuales simples que facilitan la comprensión de tareas o estados, especialmente útiles para usuarios neurodivergentes.

Pomodoro: Técnica de gestión del tiempo basada en intervalos de trabajo y descanso para favorecer la concentración.

Prototipado: Proceso de creación de versiones preliminares de la interfaz para evaluar diseño y funcionalidad antes de su implementación final.

React Native: Framework que permite desarrollar aplicaciones móviles multiplataforma utilizando JavaScript y componentes reutilizables.

Stack de navegación: Estructura que organiza las pantallas de la aplicación de forma jerárquica, permitiendo avanzar y retroceder entre vistas.

TypeScript: Lenguaje basado en JavaScript que incorpora tipado estático para mejorar la organización y comprensión del código.

Usabilidad: Grado en que una aplicación resulta fácil de usar, comprender y aprender por parte del usuario.

INDICE DE CONTENIDOS

RESUMEN.....	4
ABSTRACT	5
INDICE DE FIGURAS.....	11
INTRODUCCIÓN	14
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	15
1.1 Contexto General	15
1.2 Situación Actual.....	15
1.3 Actores Involucrados.....	15
1.4 Problema específico de la memoria.....	16
1.5 Motivación e Interés del Proyecto	16
1.6 Objetivo General Subproyecto.....	17
1.7 Objetivos Específicos.....	17
CAPÍTULO 2: MARCO CONCEPTUAL	18
2.1 Frameworks de Desarrollo Frontend	18
2.1.1 ¿Qué es un Framework?	18
2.1.2 Elección de Frameworks para EverPlanApp.....	18
2.2 Accesibilidad y Neurodiversidad	18
2.2.1 Diseño Inclusivo y Accesibilidad Web.....	18
2.2.2 Consideraciones de Neurodiversidad	19
2.2.3 Diseño emocional y autorregulación	19
2.3 Herramientas y Librerías Frontend	19
2.3.1 React.js	19

2.3.2 React Native	19
2.3.3 TypeScript.....	20
2.3.4 Vite	20
2.3.5 Zustand y Context API	20
2.3.6 React Navigation	20
2.3.7 Axios para consumo de APIs y estándares de calidad	21
2.3.8 Control de versiones con Git y GitHub	21
2.4 Compatibilidad y Adaptabilidad Móvil	21
2.5 Seguridad y Buenas Prácticas en el Frontend	21
2.5.1 Cumplimiento con ISO/IEC 25010 en el frontend de EverPlanApp	21
CAPÍTULO 3: PROPUESTA DE SOLUCION	23
3.1 Enfoque general de la solución	23
3.2 Metodología de desarrollo aplicada	24
3.3 Arquitectura del frontend e integración con el backend.....	25
3.4 Navegación y estructura de la aplicación.....	25
3.5 Implementación de funcionalidades principales	26
3.6 Uso de TypeScript en el frontend.....	28
3.7 Diseño de interfaz y experiencia de usuario	28
3.8 Notificaciones y retroalimentación al usuario	30
3.9 Casos de uso implementados en la aplicación.....	30
3.10 Aporte del subproyecto frontend	41
4.1 Estrategia de validación aplicada	42
4.2 Pruebas funcionales del frontend	42
4.3 Validación de integración con backend	43

4.4 Validación con usuarios finales	46
4.5 Validación con profesionales del área de inclusión	47
4.6 Resultados de la validación	47
CAPÍTULO 5: CONCLUSIONES	48
REFERENCIAS BIBLIOGRÁFICAS.....	50
I. Bibliografía	50

INDICE DE FIGURAS

Figura 1: Visión general del Frontend de EverPlanApp Fuente: Elaboración propia	24
Figura 2: Flujo de comunicación Frontend – Backend Fuente: Elaboración propia	25
Figura 3: Navegación de EverPlanApp Fuente: Elaboración propia	26
Figura 4: Ejemplo de pantallas principales del Frontend Fuente: Elaboración propia	28
Figura 5: Paleta de colores Fuente: Procolor AkzoNobel	29
Figura 6: Primer diseño de pantalla en Figma Fuente: Elaboración propia	29
Figura 7: Pantalla de login EverPlanApp Fuente: Elaboración propia	31
Figura 8: Pantalla de Registro de usuario EverPlanApp Fuente: Elaboración propia.....	32
Figura 9: Pantalla de módulo tareas EverPlanApp Fuente: Elaboración propia	33
Figura 10: Pantalla de módulo rutinas EverPlanApp Fuente: Elaboración propia	34
Figura 11: Pantalla de módulo horario personal EverPlanApp Fuente: Elaboración propia	35
Figura 12: Pantalla de módulo pomodoro EverPlanApp Fuente: Elaboración propia	36
Figura 13: Pantalla de módulo pictograma EverPlanApp Fuente: Elaboración propia	37
Figura 14: Pantalla de módulo autorregulación EverPlanApp Fuente: Elaboración propia	38
Figura 15: Pantalla de módulo estadísticas EverPlanApp Fuente: Elaboración propia.....	39
Figura 16: Pantalla de módulo sugerencias EverPlanApp Fuente: Elaboración propia	40
Figura 17: Flujo general de navegación de EverPlanApp. Fuente: Elaboración propia	43
Figura 18: Autenticación y obtención de token JWT mediante la API REST en Postman Fuente: Elaboración propia	44
Figura 19: Consulta de tareas del usuario autenticado en Postman Fuente: Elaboración propia.....	45
Figura 20: Visualización en interfaz de la tarea listada en postman Fuente: Elaboración propia.....	45

Figura 21: Evidencia encuesta CSAT con usuarios Fuente: Elaboración propia	46
Figura 22: Sugerencias de usuarios en validación Fuente: Elaboración propia	47

INDICE DE TABLAS

Tabla 1: Objetivos específicos, marco conceptual y propuesta de solución del subproyecto Frontend de EverPlanApp. Fuente: Elaboración Propia	17
---	----

INTRODUCCIÓN

En los últimos años, el uso de aplicaciones digitales orientadas a la organización personal y académica ha aumentado de manera significativa, especialmente entre estudiantes y personas que requieren apoyo en la planificación de tareas, rutinas y manejo del tiempo. Sin embargo, muchas de estas soluciones no consideran de forma adecuada las necesidades de usuarios neurodivergentes, como personas con Trastorno por Déficit de Atención (TDAH) o Trastorno del Espectro Autista (TEA), quienes pueden experimentar dificultades adicionales frente a interfaces complejas, sobrecarga visual o flujos de uso poco claros.

En este contexto surge EverPlanApp, una aplicación móvil orientada a apoyar la organización personal mediante un enfoque accesible, claro y centrado en la neurodiversidad. El problema abordado en esta memoria se relaciona con la falta de herramientas digitales que integren funcionalidades de planificación, autorregulación y apoyo visual, diseñadas específicamente para reducir la carga cognitiva y favorecer la autonomía del usuario.

La presente memoria se enmarca en el área del desarrollo de software, específicamente en el diseño e implementación del frontend móvil de EverPlanApp. El alcance del trabajo se centra en la construcción de la interfaz de usuario, la definición de flujos de navegación, la integración con un backend mediante APIs REST y la aplicación de principios de accesibilidad y diseño inclusivo. El frontend fue desarrollado utilizando React Native y TypeScript, considerando una arquitectura modular y escalable, y manteniendo una comunicación constante con el backend para la gestión de datos y funcionalidades.

La propuesta de solución consiste en el desarrollo de un frontend móvil que permita a los usuarios gestionar tareas, rutinas, horarios personales y herramientas de apoyo como temporizador Pomodoro, pictogramas, autorregulación y visualización de estadísticas, todo ello mediante una interfaz simple, predecible y de bajo estímulo visual. La solución se apoya en una metodología de desarrollo iterativa e incremental, con investigación previa y durante el proceso, validaciones con usuarios objetivos y retroalimentación de profesionales del área de la neurodiversidad e inclusión.

Desde el punto de vista metodológico, el desarrollo se realizó bajo un enfoque ágil, utilizando prácticas inspiradas en Scrum, con definición de objetivos claros, planificación de tareas, revisión continua y control de versiones mediante Git y GitHub. Además, se incorporaron herramientas de diseño como Figma para el prototipado y validación temprana de interfaces, así como pruebas funcionales para verificar el correcto funcionamiento de los módulos implementados.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Contexto General

En el contexto de la educación superior, la organización académica representa un desafío constante para gran parte del estudiantado. Esta dificultad se acentúa en estudiantes neurodivergentes, como aquellos que presentan Trastorno por Déficit de Atención e Hiperactividad (TDAH) o Trastorno del Espectro Autista (TEA), quienes suelen enfrentar barreras adicionales relacionadas con la gestión del tiempo, la estructuración de tareas y la autorregulación emocional.

A pesar de la existencia de múltiples herramientas digitales orientadas a la planificación, muchas de ellas no consideran las particularidades cognitivas ni sensoriales de este grupo de estudiantes. Interfaces recargadas, elementos de gamificación o flujos poco intuitivos pueden generar sobrecarga sensorial o desmotivación, dificultando aún más su experiencia académica.

Frente a esta realidad, surge la necesidad de desarrollar soluciones tecnológicas inclusivas que no solo faciliten la organización académica, sino que también fomenten la autonomía, la autorregulación y el bienestar emocional de los usuarios. En este escenario, se enmarca el desarrollo de EverPlanApp, una aplicación diseñada específicamente para apoyar la planificación académica de estudiantes neurodivergentes. La presente memoria se centra en la construcción del backend y la base de datos de dicha aplicación, componentes clave para garantizar su funcionamiento robusto, seguro y escalable, en línea con los principios de accesibilidad e inclusión educativa.

1.2 Situación Actual

Actualmente, herramientas como Todoist, MyStudyLife o Focus Plant son ampliamente utilizadas por estudiantes para organizar sus tareas, rutinas y compromisos académicos. Sin embargo, muchas de estas plataformas pueden representar una barrera para personas con condiciones neurodivergentes, como aquellas dentro del espectro autista. Esto se debe a que algunas de estas aplicaciones incorporan elementos de gamificación o interfaces visualmente cargadas con el objetivo de hacer más atractiva la experiencia de usuario. No obstante, este tipo de estímulos pueden resultar sobreestimulantes o generar rechazo en usuarios autistas, quienes suelen preferir entornos visuales más simples, con menos elementos distractores, colores suaves y estructuras claras. En general, estas aplicaciones carecen de un enfoque inclusivo y un diseño sensorialmente accesible que minimice la sobrecarga cognitiva y emocional.

1.3 Actores Involucrados

Los principales beneficiarios de esta solución son estudiantes universitarios neurodivergentes, quienes enfrentan mayores desafíos en la gestión de sus actividades

académicas. No obstante, la aplicación también puede ser útil para cualquier estudiante que requiera apoyo en su organización personal. Asimismo, se consideran actores secundarios a tutores, docentes y profesionales del ámbito psicopedagógico, quienes podrían utilizar la información generada por la aplicación como una herramienta complementaria para fortalecer el acompañamiento académico y la toma de decisiones pedagógicas.

1.4 Problema específico de la memoria

El problema específico que se aborda en esta memoria es la necesidad de desarrollar un frontend funcional, accesible y adaptable para EverPlanApp, capaz de entregar una experiencia de usuario intuitiva, clara y sensorialmente adecuada a las necesidades de estudiantes neurodivergentes. Sin una interfaz correctamente diseñada, la aplicación no lograría facilitar la organización académica ni reducir la sobrecarga cognitiva, comprometiendo su propósito inclusivo.

Este frontend debe responder a criterios de usabilidad, accesibilidad, mantenibilidad y compatibilidad con el backend, permitiendo el correcto funcionamiento de las distintas funcionalidades que conforman la aplicación, como la visualización de tareas, rutinas, horarios, recordatorios, estados de ánimo, estadísticas y pictogramas. Además, debe considerar principios de diseño universal, uso de contrastes adecuados, navegación simplificada, y estructuras visuales limpias que minimicen la sobreestimulación sensorial.

Asimismo, se requiere que el frontend se adapte a distintos dispositivos mediante una arquitectura responsive y que utilice componentes reutilizables para facilitar su mantenimiento y escalabilidad futura. El enfoque centrado en el usuario neurodivergente debe guiar cada decisión de diseño, asegurando que la plataforma sea no solo funcional, sino también empática, comprensible y verdaderamente útil para su público objetivo.

1.5 Motivación e Interés del Proyecto

La motivación principal para llevar a cabo esta memoria radica en la necesidad de contribuir, desde el desarrollo de interfaces digitales, a la creación de herramientas inclusivas que respondan a la diversidad cognitiva y sensorial presente en el entorno universitario. La ausencia de aplicaciones que consideren de forma explícita las necesidades de estudiantes neurodivergentes constituye una brecha significativa en cuanto a equidad digital, accesibilidad y bienestar académico.

Desde una perspectiva tanto profesional como personal, existe un interés genuino en aportar a una solución concreta que permita mejorar la experiencia de estudio de quienes enfrentan mayores dificultades en la organización, la planificación y la autorregulación emocional. El diseño del frontend de EverPlanApp representa una oportunidad para aplicar principios de diseño universal y tecnologías modernas orientadas a la accesibilidad, la claridad visual y la empatía con el usuario final.

Asimismo, este subproyecto constituye un desafío formativo para aplicar conocimientos adquiridos durante la carrera en áreas como diseño de interfaces, desarrollo web responsivo, integración con APIs, usabilidad y patrones de interacción centrados en el usuario neurodivergente.

1.6 Objetivo General Subproyecto

Desarrollar la interfaz de usuario de EverPlanApp como una aplicación accesible, intuitiva y visualmente clara para plataformas web y móvil, que permita a estudiantes neurodivergentes organizar sus tareas, rutinas, clases y actividades personales, promoviendo su autorregulación, autonomía e independencia.

1.7 Objetivos Específicos

Tabla 1: Objetivos específicos, marco conceptual y propuesta de solución del subproyecto Frontend de EverPlanApp. Fuente: Elaboración Propia

Objetivo Específico	Marco Conceptual (Fundamento Técnico)	Propuesta de Solución
Diseñar interfaces accesibles, claras y neuroinclusivas.	WCAG 2.1, diseño universal, principios de diseño para usuarios neurodivergentes (colores suaves, estructura predecible, bajo estímulo sensorial).	Aplicar contrastes y tipografías accesibles, minimizar estímulos visuales innecesarios, respetar jerarquías visuales y favorecer la autorregulación del usuario.
Implementar componentes reutilizables, modulares y adaptables a dispositivos móviles.	React y React Native, arquitectura basada en componentes, uso de Flexbox y estilos nativos adaptativos.	Desarrollo de componentes funcionales reutilizables, uso de estilos nativos de React Native y estructuras visuales consistentes en distintas pantallas.
Integrar navegación fluida y manejo de estado consistente.	React Navigation para la gestión de flujos de navegación, Zustand y Context API para el manejo de estado global y transversal.	Implementación de navegación predecible mediante stacks y vistas, junto con gestión eficiente del estado utilizando Zustand y Context API según el caso.
Consumir APIs REST de forma segura y eficiente.	Arquitectura REST, Axios, manejo de errores, async/await, ISO/IEC 25010 (eficiencia funcional).	Consumo de la API mediante Axios, validación de respuestas, manejo de estados de carga y mensajes accesibles ante fallos de comunicación.

Implementar una aplicación móvil accesible utilizando React Native.	React Native, enfoque mobile-first, diseño adaptativo y principios de accesibilidad.	Desarrollo directo de la aplicación móvil, priorizando patrones de interacción nativos, claridad visual y coherencia con los principios de diseño neuroinclusivo.
---	--	---

CAPÍTULO 2: MARCO CONCEPTUAL

2.1 Frameworks de Desarrollo Frontend

2.1.1 ¿Qué es un Framework?

Un framework es una estructura reutilizable de software que proporciona un conjunto de herramientas, reglas y componentes predefinidos que facilitan el desarrollo de aplicaciones. En el ámbito frontend, su función principal es estandarizar el flujo de trabajo, mejorar la mantenibilidad y acelerar el proceso de desarrollo. Su uso permite centrarse en la lógica específica de la aplicación sin construir todo desde cero [AWS25].

2.1.2 Elección de Frameworks para EverPlanApp

Para este subproyecto se seleccionaron tecnologías modernas y ampliamente utilizadas en el desarrollo frontend, como React.js para la construcción de interfaces de usuario basadas en componentes reutilizables [ReactDoc]. Adicionalmente, Vite fue empleado como entorno de desarrollo durante etapas iniciales de prototipado del frontend en entorno web, permitiendo optimizar el rendimiento, acelerar la recarga en tiempo real y simplificar la configuración del proyecto [Vite25].

Estas tecnologías fueron seleccionadas debido a su enfoque moderno y flexible, que facilita la creación de interfaces modulares, mantenibles y alineadas con principios de accesibilidad. Asimismo, la compatibilidad del ecosistema React con React Native permitió orientar el desarrollo de EverPlanApp hacia una implementación móvil coherente desde las primeras etapas del proyecto, manteniendo coherencia visual y funcional entre los distintos entornos de desarrollo.

2.2 Accesibilidad y Neurodiversidad

2.2.1 Diseño Inclusivo y Accesibilidad Web

El enfoque de diseño de EverPlanApp se fundamenta en los principios de accesibilidad web propuestos por las WCAG 2.1 (Web Content Accessibility Guidelines), los cuales establecen que todo contenido digital debe ser perceptible, operable, comprensible y robusto

[W3C25]. Este enfoque busca eliminar barreras de acceso para personas con condiciones como TDAH o autismo, promoviendo interfaces simples, predecibles, de bajo estímulo sensorial y con jerarquías claras.

2.2.2 Consideraciones de Neurodiversidad

Se incorporan lineamientos sugeridos por investigaciones y guías prácticas centradas en población neurodivergente, como Autism&Uni [AutismUni16] y Fundación Descúbreme [Descubre23]. Se priorizan colores suaves, espaciado adecuado, navegación lineal, íconos comprensibles y consistencia visual. Esto mejora la experiencia de usuarios con TEA o TDAH, y también de personas sin diagnóstico formal que enfrentan desafíos similares en el entorno académico. A diferencia de otras aplicaciones de organización, EverPlanApp evita el uso de mecanismos de gamificación intensiva, rankings o recompensas visuales intrusivas, ya que estos elementos pueden generar sobreestimulación o rechazo en usuarios neurodivergentes.

2.2.3 Diseño emocional y autorregulación

El diseño emocional es un enfoque que considera el impacto de los elementos visuales y de interacción en el estado emocional del usuario. En EverPlanApp, este enfoque se traduce en la implementación de interfaces que favorecen la autorregulación, reduciendo la sobrecarga cognitiva y emocional. Se priorizan estructuras visuales simples, retroalimentación no intrusiva, uso moderado del color y ausencia de estímulos agresivos, permitiendo que los usuarios mantengan el control de su experiencia sin generar ansiedad o distracción excesiva.

2.3 Herramientas y Librerías Frontend

2.3.1 React.js

React es una biblioteca JavaScript para construir interfaces de usuario a partir de componentes reutilizables [ReactDoc]. Permite un desarrollo declarativo, un manejo eficiente del DOM mediante virtual DOM y facilita la separación de responsabilidades entre lógica y presentación. Gracias a su modularidad y amplia comunidad, es una herramienta ideal para construir interfaces accesibles, dinámicas y mantenibles.

2.3.2 React Native

React Native es el framework principal utilizado en el desarrollo de EverPlanApp, permitiendo la construcción de una aplicación móvil multiplataforma mediante el uso de componentes reutilizables basados en JavaScript y React. Su arquitectura facilita la creación de interfaces consistentes y accesibles, aprovechando patrones de interacción nativos de dispositivos móviles. La elección de React Native permitió priorizar un enfoque mobile-first, alineado con las necesidades de estudiantes neurodivergentes, garantizando una experiencia de uso clara, predecible y de bajo estímulo sensorial.

2.3.3 TypeScript

TypeScript fue utilizado en el desarrollo de EverPlanApp como lenguaje base del frontend, permitiendo trabajar sobre JavaScript con una estructura más clara y controlada. Su principal aporte en el proyecto fue facilitar la organización del código mediante la definición explícita de tipos para variables, funciones y estructuras de datos, como tareas, rutinas y estados internos de la aplicación.

El uso de TypeScript permitió reducir errores comunes asociados a valores mal definidos o inconsistentes, mejorar la lectura y comprensión del código durante el desarrollo, y apoyar la mantención del proyecto a medida que se incorporaban nuevas funcionalidades. De esta forma, TypeScript contribuyó a un desarrollo más ordenado y predecible del frontend de EverPlanApp, sin agregar complejidad innecesaria al proceso.

2.3.4 Vite

Vite es una herramienta moderna de construcción y entorno de desarrollo orientada a proyectos frontend basados en módulos ES. En el contexto de EverPlanApp, Vite fue utilizado principalmente durante etapas iniciales de prototipado y validación del frontend en entorno web, permitiendo una rápida configuración del proyecto, recarga en caliente eficiente y un flujo de desarrollo ágil. Si bien el desarrollo principal de la aplicación se orientó a plataformas móviles, el uso de Vite facilitó la experimentación temprana con componentes, estructuras visuales y principios de diseño antes de su adaptación definitiva al entorno React Native.

2.3.5 Zustand y Context API

Para el manejo del estado global de la aplicación se utilizó principalmente la librería Zustand, debido a su simplicidad, bajo acoplamiento y facilidad de integración con componentes funcionales en React Native. Zustand permitió gestionar de manera eficiente estados compartidos como tareas, rutinas y configuraciones del usuario, favoreciendo una arquitectura clara y mantenible. De forma complementaria, la Context API fue utilizada únicamente en casos específicos donde se requirió compartir información transversal, evitando sobrecargar la aplicación con contextos innecesarios y manteniendo una estructura de estado optimizada.

2.3.6 React Navigation

React Navigation es la librería utilizada para la gestión de navegación dentro de EverPlanApp en su implementación móvil con React Native. Esta herramienta permite organizar la aplicación mediante flujos de navegación basados en stacks y pestañas, facilitando una experiencia de usuario fluida y coherente sin recargar la aplicación. Su uso fue clave para diseñar una navegación predecible, simple y estructurada, alineada con las necesidades de usuarios neurodivergentes, quienes se benefician de recorridos claros, jerarquías visuales consistentes y una reducción de cambios abruptos entre pantallas.

2.3.7 Axios para consumo de APIs y estándares de calidad

Para la comunicación entre el frontend y el backend de EverPlanApp se utilizó Axios como librería para la realización de peticiones HTTP. Esta herramienta permitió manejar de forma clara y estructurada las solicitudes a la API REST, incorporando validaciones de respuesta, manejo de errores y control de estados de carga. Su implementación contribuye a la eficiencia funcional del sistema, alineándose con los criterios definidos por la norma ISO/IEC 25010, al asegurar interacciones confiables, comprensibles y oportunas para el usuario final.

2.3.8 Control de versiones con Git y GitHub

Durante el desarrollo de EverPlanApp se utilizó Git como sistema de control de versiones, permitiendo gestionar de forma ordenada los cambios en el código fuente, mantener un historial de modificaciones y facilitar la detección de errores. GitHub fue empleado como plataforma de repositorio remoto, apoyando el trabajo incremental, la organización del proyecto y la recuperación de versiones estables. El uso de control de versiones contribuyó a la mantenibilidad y trazabilidad del desarrollo del frontend.

2.4 Compatibilidad y Adaptabilidad Móvil

EverPlanApp fue desarrollada directamente como una aplicación móvil utilizando React Native, priorizando desde el inicio un enfoque mobile-first. La interfaz fue diseñada considerando distintos tamaños de pantalla, orientación vertical y patrones de interacción propios de dispositivos móviles. Este enfoque permitió garantizar una experiencia consistente y accesible, asegurando que los componentes visuales, tipografías y flujos de navegación se adapten adecuadamente a distintos dispositivos, manteniendo siempre los principios de simplicidad, claridad y bajo estímulo sensorial.

2.5 Seguridad y Buenas Prácticas en el Frontend

Durante el desarrollo del frontend de EverPlanApp se aplicaron buenas prácticas orientadas a la estabilidad y confiabilidad de la aplicación, tales como la validación de datos de entrada, el manejo controlado de errores y la separación clara de responsabilidades entre componentes. Asimismo, se dejó preparada la arquitectura del frontend para la incorporación futura de mecanismos de seguridad adicionales, como autenticación mediante tokens y almacenamiento seguro de información sensible, garantizando así la escalabilidad y robustez del sistema.

2.5.1 Cumplimiento con ISO/IEC 25010 en el frontend de EverPlanApp

El desarrollo del frontend de EverPlanApp considera los lineamientos de la norma internacional ISO/IEC 25010, enfocándose en aquellas características de calidad más relevantes para una interfaz centrada en accesibilidad, eficiencia funcional y mantenibilidad. A continuación, se detallan las principales características aplicadas y cómo se abordan en el proyecto:

- Eficiencia funcional: se asegura que cada funcionalidad de la interfaz cumpla correctamente con los requisitos definidos. Esto se traduce en una correcta integración con el backend mediante el consumo de APIs REST, validación de datos de entrada, retroalimentación inmediata al usuario y manejo adecuado de errores.
- Usabilidad: se implementan principios de accesibilidad siguiendo la norma WCAG 2.1 [W3C25], promoviendo interfaces claras, previsibles y de bajo estímulo sensorial. Se prioriza el diseño inclusivo mediante estructuras visuales consistentes, contrastes adecuados y elementos comprensibles para usuarios neurodivergentes.
- Compatibilidad: se adopta un enfoque de diseño adaptativo mediante el uso de estilos nativos de React Native, flexbox y componentes responsivos, permitiendo que la interfaz se adapte correctamente a distintos tamaños de pantalla y dispositivos móviles.
- Mantenibilidad: se estructura el sistema a través de componentes modulares en React, facilitando la separación de responsabilidades y la reutilización de lógica y estilos. El uso de Context API y Zustand para el manejo del estado refuerza la claridad del código y su facilidad de extensión futura.

Este enfoque cumple con requisitos técnicos y entregar una solución alineada con estándares de calidad reconocidos, robusta y adecuada para usuarios con necesidades diversas.

CAPÍTULO 3: PROPUESTA DE SOLUCION

Este capítulo presenta el desarrollo de la solución propuesta correspondiente al subproyecto Frontend de EverPlanApp. La solución fue implementada como una aplicación móvil orientada a la organización personal y autorregulación de estudiantes, considerando especialmente a usuarios neurodivergentes. El trabajo realizado se centra en el diseño, implementación y validación funcional de la interfaz de usuario, así como en su integración con un backend mediante el consumo de servicios API.

Si bien EverPlanApp contempla una arquitectura general que incluye backend y otros módulos, este capítulo aborda en detalle la solución desarrollada en el frontend móvil, que constituye el principal medio de interacción entre el usuario y el sistema.

3.1 Enfoque general de la solución

La solución propuesta consiste en una aplicación móvil desarrollada con React Native utilizando Expo, diseñada bajo un enfoque mobile-first. La aplicación permite a los usuarios gestionar tareas, rutinas y horarios, además de incorporar herramientas de apoyo a la concentración y autorregulación, como temporizador Pomodoro, pictogramas, estadísticas y notificaciones.

Desde el punto de vista arquitectónico, el frontend actúa como un cliente que consume servicios provistos por un backend a través de una API REST. El frontend no almacena información crítica de forma permanente, sino que obtiene y envía datos mediante solicitudes HTTP, procesando las respuestas para actualizar la interfaz y el estado de la aplicación.

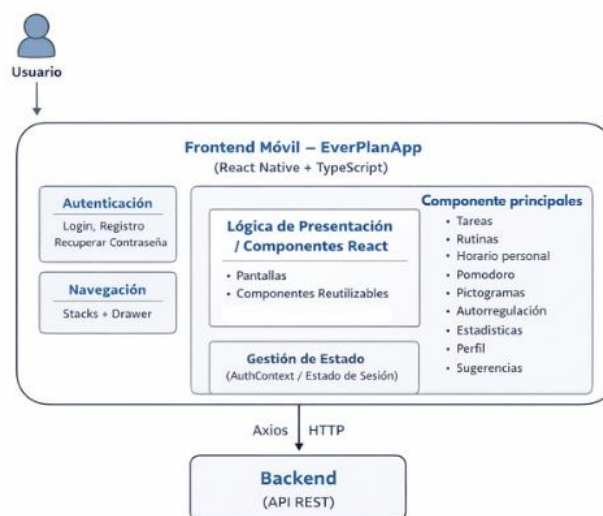


Figura 1: Visión general del Frontend de EverPlanApp

Fuente: Elaboración propia

3.2 Metodología de desarrollo aplicada

El desarrollo del frontend se realizó mediante un enfoque incremental e iterativo, alineado con principios de metodologías ágiles. Las funcionalidades fueron implementadas de manera progresiva, comenzando por pantallas base y flujos principales, para luego incorporar validaciones, navegación, conexión con backend y mejoras en la experiencia de usuario.

Previo al desarrollo técnico, se realizó una etapa de investigación inicial orientada a comprender las necesidades de estudiantes neurodivergentes, considerando dificultades asociadas a la atención, organización del tiempo, autorregulación emocional y sobrecarga cognitiva. Esta investigación incluyó revisión de material especializado y conversaciones con profesionales del área de la neurología y de la inclusión, cuyos aportes permitieron orientar las decisiones de diseño y funcionalidad del frontend.

Durante el proceso de desarrollo, esta investigación se mantuvo de forma continua, contrastando las funcionalidades implementadas con recomendaciones profesionales, especialmente en aspectos como simplificación de interfaces, uso de apoyos visuales, reducción de estímulos innecesarios y estructuración clara de flujos de interacción. Esto permitió ajustar el diseño de módulos como pictogramas, autorregulación, temporizador Pomodoro y notificaciones, asegurando que la aplicación respondiera a necesidades reales de los usuarios objetivo.

En paralelo, se diseñaron prototipos de interfaz utilizando la herramienta Figma, lo que permitió definir la estructura de las pantallas, los flujos de navegación y la jerarquía visual de los componentes antes de su implementación. Estos prototipos facilitaron la validación temprana del diseño y la realización de ajustes previos a la etapa de programación.

Asimismo, se consideró retroalimentación obtenida a partir de pruebas de uso y observaciones cualitativas realizadas con usuarios objetivo, enfocadas en la claridad de las pantallas, la facilidad de navegación y la comprensión de las funcionalidades principales. Esta retroalimentación permitió realizar ajustes en la disposición de los elementos, textos y flujos de interacción.

El trabajo se organizó siguiendo una dinámica ágil, con objetivos claros definidos por periodos de desarrollo, instancias de seguimiento del avance y ajustes continuos según los resultados obtenidos. Cada módulo fue desarrollado, probado y ajustado antes de avanzar al siguiente, lo que permitió detectar errores tempranamente y mejorar la estabilidad general de la aplicación.

El control de versiones se realizó utilizando Git, con GitHub como repositorio remoto, lo que permitió mantener un registro ordenado de los cambios, respaldar el trabajo y facilitar la organización del proyecto durante su evolución.

3.3 Arquitectura del frontend e integración con el backend

El frontend fue estructurado de forma modular, organizando la aplicación en pantallas y componentes funcionales. Cada pantalla representa una funcionalidad específica y se conecta al backend mediante solicitudes HTTP gestionadas a través de Axios.

Las llamadas a la API se centralizaron en un archivo de servicios (api.tsx), lo que permitió separar la lógica de comunicación con el backend de la lógica de presentación. Este diseño facilita la mantención del código y permite modificar endpoints o configuraciones sin afectar directamente a las pantallas.

El flujo general de interacción es el siguiente: el usuario realiza una acción en la interfaz, la aplicación valida los datos ingresados, se envía una solicitud al backend, se procesa la respuesta y finalmente se actualiza la interfaz según el resultado obtenido.

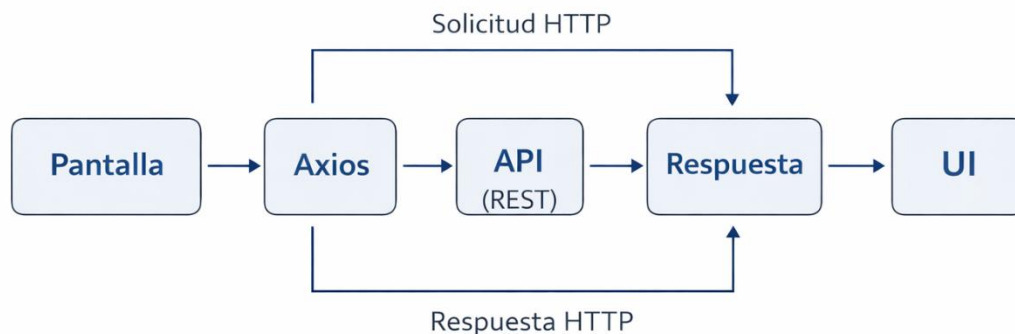


Figura 2: Flujo de comunicación Frontend – Backend

Fuente: Elaboración propia

3.4 Navegación y estructura de la aplicación

Se definieron flujos de navegación distintos según el estado de autenticación del usuario. Cuando el usuario no ha iniciado sesión, la aplicación muestra únicamente pantallas de acceso, como inicio de sesión, registro y recuperación de contraseña. Una vez autenticado, el usuario accede a un conjunto diferente de pantallas correspondientes a las funcionalidades principales de la aplicación.

Estos flujos fueron organizados mediante stacks de navegación, permitiendo un desplazamiento ordenado entre pantallas relacionadas, y un menú lateral tipo drawer que facilita el acceso directo a los distintos módulos de la aplicación. Esta estructura permite un desplazamiento claro entre las distintas secciones de la aplicación, evitando recorridos complejos o confusos. El diseño de navegación fue pensado para favorecer la claridad y la previsibilidad, aspectos especialmente relevantes para usuarios con dificultades de atención.

Entre las principales pantallas implementadas se encuentran: inicio de sesión, registro, recuperación de contraseña, pantalla principal, tareas, rutinas, horario personal, Pomodoro, autorregulación, pictogramas, estadísticas, perfil y sugerencias.

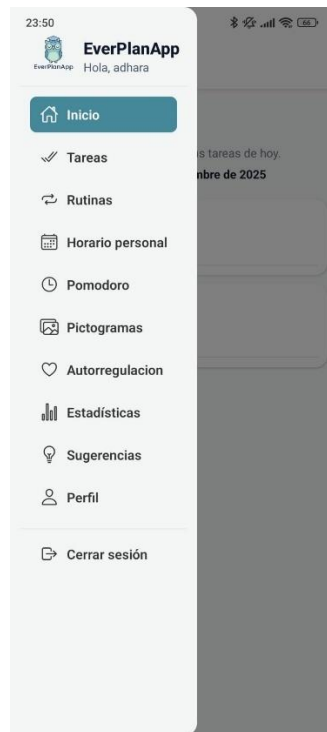


Figura 3: Navegación de EverPlanApp

Fuente: Elaboración propia

3.5 Implementación de funcionalidades principales

La funcionalidad de autenticación permite a los usuarios registrarse, iniciar sesión y recuperar su contraseña. Estas acciones se conectan directamente con el backend, validando las credenciales y controlando el acceso a las funcionalidades internas mediante el manejo del estado de sesión. Este mecanismo permite restringir el acceso a la información personal y asegurar que cada usuario visualice únicamente sus propios datos dentro de la aplicación.

El módulo de tareas permite crear, visualizar, editar y gestionar actividades pendientes de forma flexible. Cada tarea puede ser configurada con distintos atributos, como nombre, descripción, estado y color identificador, lo que facilita la categorización visual mediante chips de colores. Esta diferenciación cromática permite al usuario reconocer rápidamente el tipo de tarea o su prioridad. Además, el módulo contempla el registro del tiempo asociado a cada tarea, permitiendo al usuario llevar un control aproximado de las horas dedicadas a sus actividades, así como la programación de notificaciones de recordatorio.

El módulo de rutinas y el horario personal permiten estructurar actividades recurrentes y visualizar la planificación diaria o semanal del usuario. En el caso de las rutinas, se diseñó una funcionalidad orientada a la organización de tareas repetitivas o rituales diarios, aspecto especialmente relevante para personas neurodivergentes, quienes suelen beneficiarse de estructuras estables y predecibles. Estas rutinas permiten reducir la carga cognitiva asociada a la toma de decisiones y facilitan la adherencia a hábitos.

El horario personal ofrece una visualización clara del tiempo, permitiendo asociar información contextual a cada bloque, como notas o imágenes de referencia, por ejemplo, fotografías del aula, del entorno o del profesor supervisor. Este apoyo visual refuerza la anticipación de actividades y contribuye a una mejor orientación temporal y espacial del usuario.

El temporizador Pomodoro fue incorporado como una herramienta de apoyo a la concentración y gestión del tiempo, permitiendo al usuario alternar periodos de trabajo y descanso de forma estructurada. Su diseño prioriza la simplicidad visual, evitando estímulos excesivos y promoviendo el enfoque sostenido en una sola actividad a la vez.

El módulo de pictogramas y autorregulación fue diseñado como apoyo específico para usuarios neurodivergentes, permitiendo representar tareas, estados emocionales o pasos de una actividad mediante elementos visuales simples y comprensibles. Esta funcionalidad facilita la anticipación de actividades, la comprensión de procesos y la estructuración de rutinas, aportando una alternativa visual al contenido textual.

Finalmente, la sección de estadísticas entrega retroalimentación visual sobre el uso de la aplicación, permitiendo al usuario observar patrones de comportamiento, frecuencia de uso y hábitos de organización. Esta información apoya la toma de conciencia sobre la gestión del tiempo y favorece procesos de autorregulación, sin recurrir a mecanismos de gamificación invasivos.

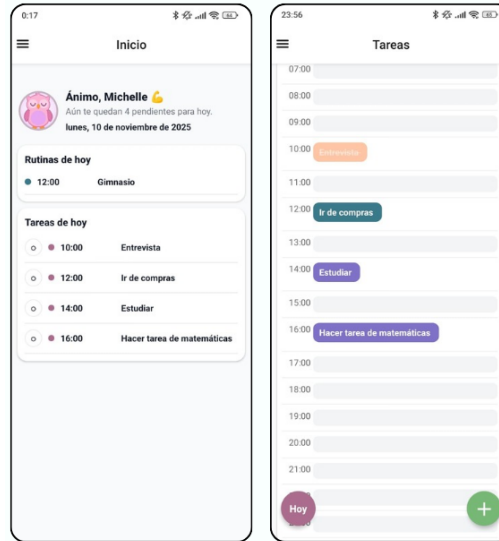


Figura 4: Ejemplo de pantallas principales del Frontend

Fuente: Elaboración propia

3.6 Uso de TypeScript en el frontend

El desarrollo del frontend se realizó utilizando TypeScript como lenguaje base. Su uso permitió definir de manera clara las estructuras de datos utilizadas en la aplicación, como usuarios, tareas, rutinas y estados internos.

TypeScript contribuyó a reducir errores durante el desarrollo, mejorar la legibilidad del código y facilitar la organización del proyecto a medida que se incorporaban nuevas funcionalidades. Su uso fue intencionado y acotado, evitando una complejidad innecesaria.

3.7 Diseño de interfaz y experiencia de usuario

El diseño de la interfaz priorizó la simplicidad, la coherencia visual y la accesibilidad. Se utilizaron colores suaves, espaciados adecuados y estructuras visuales repetibles, permitiendo que el usuario reconozca fácilmente patrones de interacción dentro de la aplicación.

Las pantallas fueron diseñadas para adaptarse a distintos tamaños de dispositivo, manteniendo siempre la legibilidad y usabilidad. Se evitó el uso de animaciones invasivas y estímulos visuales excesivos.

El diseño considera principios de accesibilidad y neurodiversidad, incorporando jerarquías claras, iconografía comprensible y flujos de interacción predecibles.

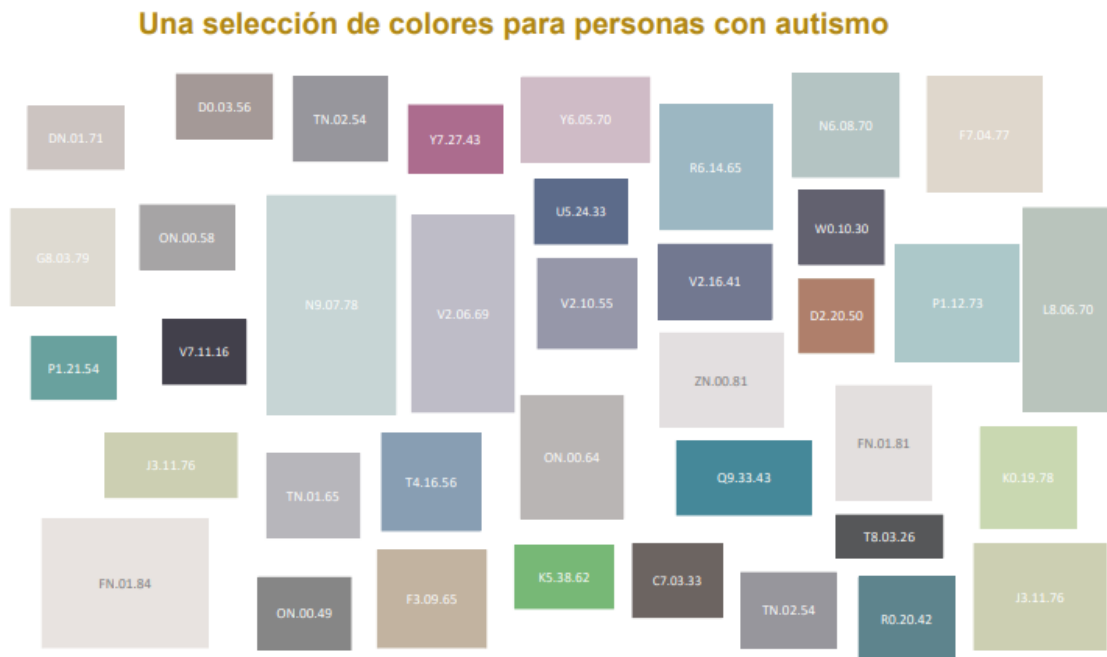


Figura 5: Paleta de colores
Fuente: Procolor AkzoNobel

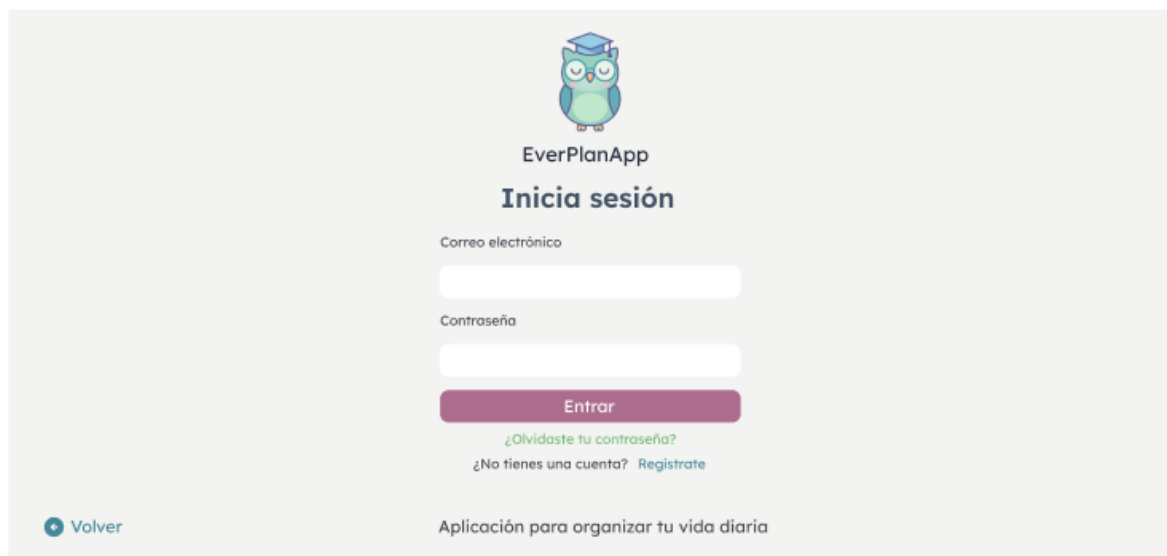


Figura 6: Primer diseño de pantalla en Figma
Fuente: Elaboración propia

3.8 Notificaciones y retroalimentación al usuario

La aplicación incorpora un sistema de notificaciones que permite recordar tareas y rutinas programadas. Estas notificaciones fueron diseñadas para ser claras y no intrusivas, entregando información relevante sin generar sobreestimulación.

Además, se implementó manejo controlado de errores, mostrando mensajes comprensibles ante fallos de conexión, errores de validación o respuestas inesperadas del backend.

3.9 Casos de uso implementados en la aplicación

Caso de uso: Iniciar sesión

Actores: Usuario.

Descripción breve: Este caso de uso permite al usuario acceder a las funcionalidades de EverPlanApp mediante sus credenciales.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario abre la app.	2. El sistema muestra el formulario de acceso.
3. El usuario ingresa su correo electrónico y contraseña.	4. El sistema valida las credenciales.
	5. El sistema permite el acceso y muestra la pantalla principal.

Excepciones

- **Paso 4:** Si las credenciales son incorrectas, el sistema informa el error y solicita nuevamente los datos.



Figura 7: Pantalla de login EverPlanApp

Fuente: Elaboración propia

Caso de uso: Registrar nuevo usuario

Actores: Usuario.

Descripción breve: Este caso de uso permite al usuario crear una cuenta en la aplicación.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la opción "Registrarse".	2. El sistema muestra el formulario de registro.
3. El usuario ingresa sus datos personales.	4. El sistema valida la información.
5. El usuario confirma el registro.	6. El sistema crea la cuenta de usuario.
	7. El sistema muestra un mensaje de confirmación.

Excepciones

- **Paso 4:** Si algún dato es inválido o está incompleto, el sistema informa el error.

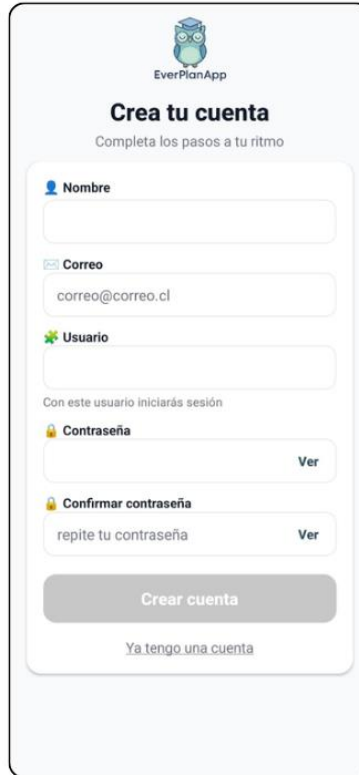


Figura 8: Pantalla de Registro de usuario EverPlanApp
Fuente: Elaboración propia

Caso de uso: Registrar tarea

Actores: Usuario.

Descripción breve: Permite al usuario crear una nueva tarea.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el botón (+)	2. El sistema muestra el formulario de tarea.
3. El usuario ingresa la información requerida.	4. El sistema valida los datos ingresados.
5. El usuario confirma el registro.	6. El sistema almacena la tarea.
	7. El sistema muestra un mensaje de confirmación.

Excepciones

- **Paso 4:** Si existen campos obligatorios vacíos, el sistema solicita completarlos.

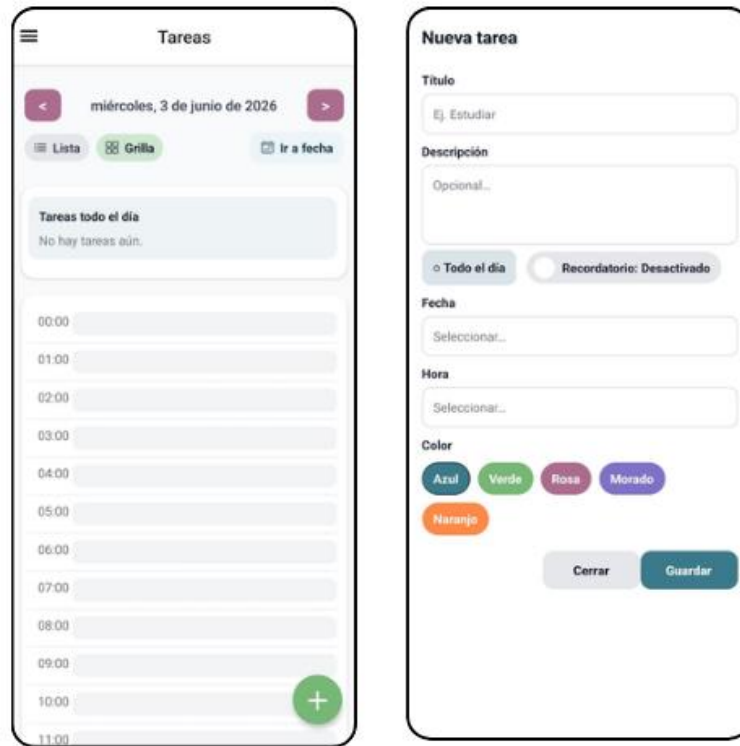


Figura 9: Pantalla de módulo tareas EverPlanApp

Fuente: Elaboración propia

Caso de uso: Crear Rutina

Actores: Usuario.

Descripción breve: Permite al usuario crear una nueva rutina.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el botón (+)	2. El sistema muestra el formulario de rutina.
3. El usuario ingresa la información requerida.	4. El sistema valida los datos ingresados.
5. El usuario confirma el registro.	6. El sistema almacena la rutina.

	7. El sistema muestra un mensaje de confirmación.
--	---

Excepciones

- **Paso 4:** Si existen campos obligatorios vacíos, el sistema solicita completarlos.

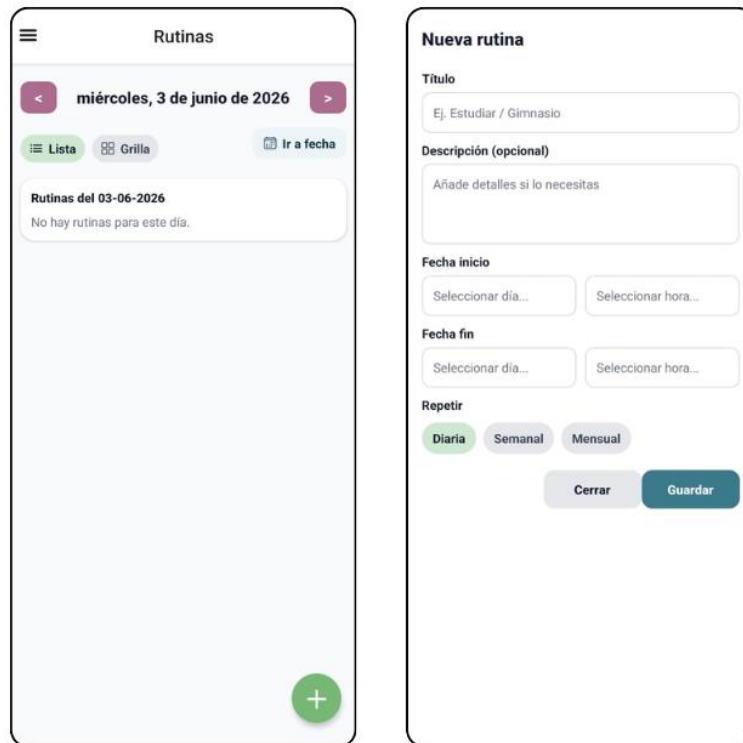


Figura 10: Pantalla de módulo rutinas EverPlanApp
Fuente: Elaboración propia

Caso de uso: Gestionar Horario Personal

Actores: Usuario.

Descripción breve: Permite organizar actividades dentro de un horario personal.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario accede al horario personal.	2. El sistema muestra el horario actual.
3. El usuario agrega o modifica un bloque horario.	4. El sistema valida la información.
5. El usuario confirma los cambios.	6. El sistema guarda la información.
	7. El sistema muestra el horario actualizado.

Excepciones

- **Paso 4:** Si existen campos obligatorios vacíos, el sistema solicita completarlos, si existe conflicto de horarios, el sistema informa el error.

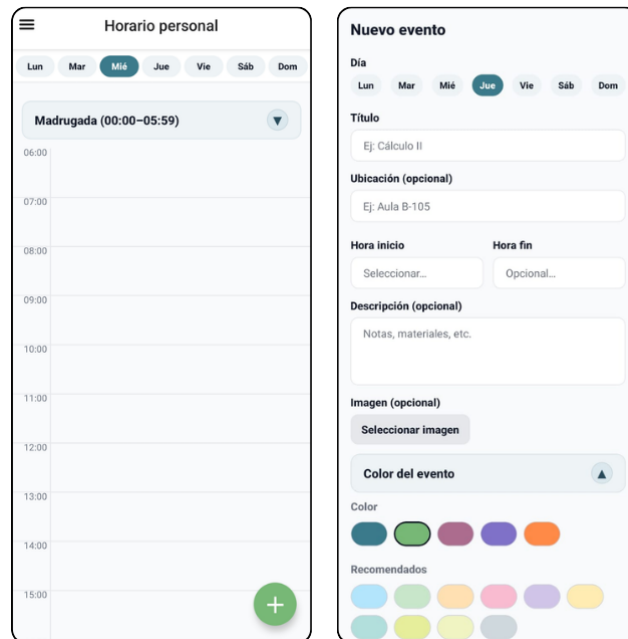


Figura 11: Pantalla de módulo horario personal EverPlanApp

Fuente: Elaboración propia

Caso de uso: Utilizar Temporizador Pomodoro

Actores: Usuario.

Descripción breve: Permite iniciar sesiones de concentración utilizando la técnica Pomodoro.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario accede al módulo Pomodoro.	2. El sistema muestra el temporizador.
3. El usuario configura la duración de la sesión.	4. El sistema registra la configuración.
5. El usuario inicia el temporizador.	6. El sistema comienza la cuenta regresiva.
	7. El sistema notifica el término de la sesión.

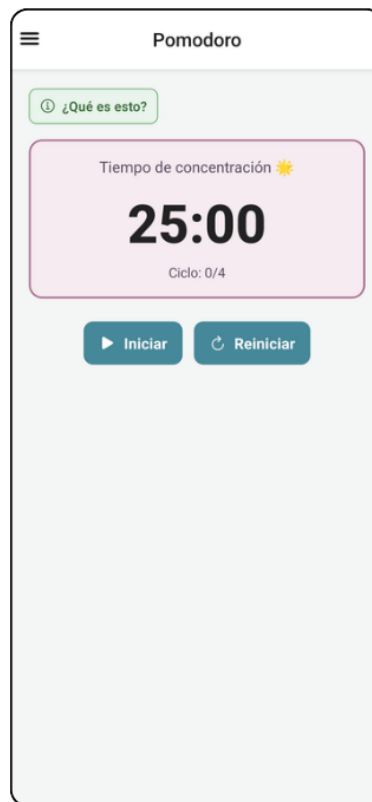


Figura 12: Pantalla de módulo pomodoro EverPlanApp

Fuente: Elaboración propia

Caso de uso: Consultar Pictogramas

Actores: Usuario.

Descripción breve: Permite visualizar pictogramas secuenciales de actividades.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario accede al módulo de pictogramas.	2. El sistema muestra las categorías disponibles.
3. El usuario selecciona una categoría.	4. El sistema muestra los pictogramas asociados.
5. El usuario revisa la información visual.	6. El sistema mantiene la visualización disponible.

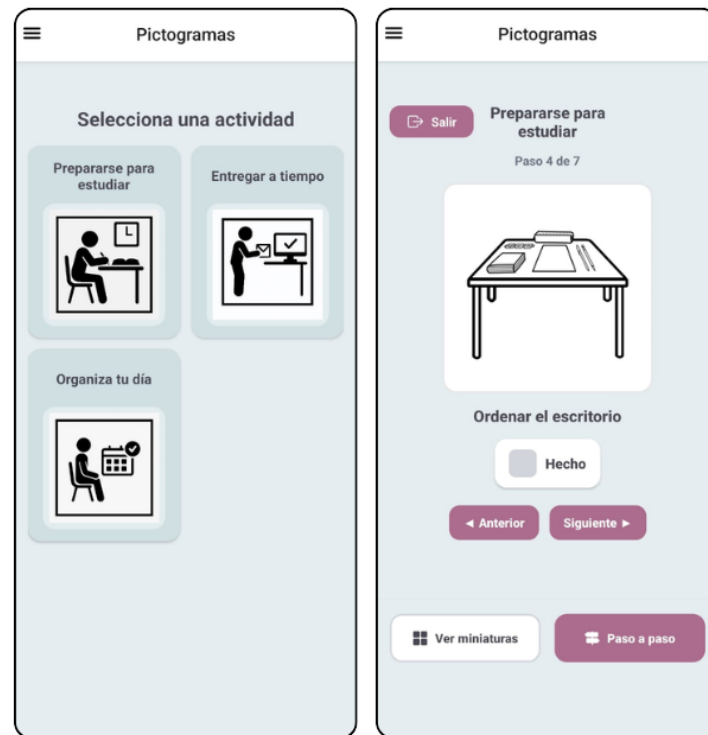


Figura 13: Pantalla de módulo pictograma EverPlanApp

Fuente: Elaboración propia

Caso de uso: Consultar Estrategias de Autorregulación

Actores: Usuario.

Descripción breve: Permite acceder a recursos orientados a la gestión emocional y autorregulación.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario accede al módulo de autorregulación.	2. El sistema muestra las estrategias disponibles.
3. El usuario selecciona una estrategia.	4. El sistema muestra la información correspondiente.
5. El usuario revisa el contenido y sigue las instrucciones.	6. El sistema mantiene la información disponible.



Figura 14: Pantalla de módulo autorregulación EverPlanApp

Fuente: Elaboración propia

Caso de uso: Visualizar Estadísticas

Actores: Usuario.

Descripción breve: Permite consultar estadísticas relacionadas con el uso de la aplicación y el cumplimiento de actividades.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario accede al módulo de estadísticas.	2. El sistema recupera la información registrada.
	3. El sistema genera gráficos e indicadores.
4. El usuario revisa las estadísticas.	5. El sistema mantiene la visualización disponible.



Figura 15: Pantalla de módulo estadísticas EverPlanApp

Fuente: Elaboración propia

Caso de uso: Consultar Sugerencias

Actores: Usuario.

Descripción breve: Permite visualizar recomendaciones relacionadas con organización.

Curso normal de los eventos

Acción del Actor	Respuesta del Sistema
1. El usuario accede al módulo de sugerencias.	2. El sistema muestra la barra de búsqueda de sugerencias.
3. El usuario escribe la tarea de la que quiere sugerencia de agendamiento horario	3. El sistema muestra las sugerencias al usuario.
4. El usuario revisa las recomendaciones.	5. El sistema mantiene la información disponible.

Excepciones:

- **Paso 2:** Si no existen sugerencias disponibles, el sistema informa la situación.



Figura 16: Pantalla de módulo sugerencias EverPlanApp

Fuente: Elaboración propia

3.10 Aporte del subproyecto frontend

El principal aporte de este subproyecto es el diseño e implementación de un frontend móvil funcional, accesible y orientado a las necesidades reales de los usuarios. El frontend desarrollado permite una interacción efectiva con un backend, facilita la gestión de información personal y apoya procesos de organización y autorregulación.

La solución propuesta cumple una función técnica, y también incorpora criterios de diseño centrados en el usuario y en la neurodiversidad, aportando valor al proyecto EverPlanApp y estableciendo una base sólida para futuras mejoras y extensiones.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

Este capítulo presenta la validación de la solución propuesta para EverPlanApp, enfocándose en comprobar que el desarrollo del frontend responde de forma adecuada a las necesidades para las cuales fue diseñado. La validación se realizó considerando tanto pruebas técnicas del funcionamiento de la aplicación como instancias de retroalimentación con usuarios reales, especialmente personas neurodivergentes o con dificultades de organización y atención.

Dado que EverPlanApp corresponde a una aplicación en etapa de desarrollo funcional, la validación se orientó a verificar la coherencia de los flujos, la usabilidad de las interfaces, la correcta integración con el backend y la utilidad percibida de las funcionalidades implementadas.

4.1 Estrategia de validación aplicada

La validación del frontend de EverPlanApp se realizó mediante una combinación de las siguientes estrategias:

- Pruebas funcionales manuales de cada módulo implementado.
- Pruebas de navegación y flujo de pantallas.
- Pruebas de integración con el backend mediante consumo de APIs.
- Validación de experiencia de usuario con personas del público objetivo.
- Retroalimentación de profesionales vinculados a neurodivergencia e inclusión.

Este enfoque permitió evaluar tanto el correcto funcionamiento técnico como la pertinencia de la solución en el contexto real de uso.

4.2 Pruebas funcionales del frontend

Las pruebas funcionales se realizaron durante todo el proceso de desarrollo, siguiendo un enfoque incremental. Cada pantalla y módulo fue probado de manera individual antes de integrarse al flujo general de la aplicación.

Entre las funcionalidades verificadas se encuentran:

- Registro, inicio de sesión y recuperación de contraseña.
- Manejo de estado de sesión y persistencia del usuario autenticado.
- Creación, edición y eliminación de tareas.
- Asignación de colores y etiquetas visuales a las tareas.

- Programación y recepción de notificaciones asociadas a tareas.
- Creación y visualización de rutinas repetitivas.
- Visualización del horario personal diario y semanal.
- Uso del temporizador Pomodoro con control de ciclos de trabajo y descanso.
- Acceso a módulos de pictogramas, autorregulación y sugerencias.
- Visualización de estadísticas de uso.

Estas pruebas permitieron detectar errores de navegación, problemas de renderizado, fallos en la carga de datos y ajustes necesarios en la interacción entre pantallas.

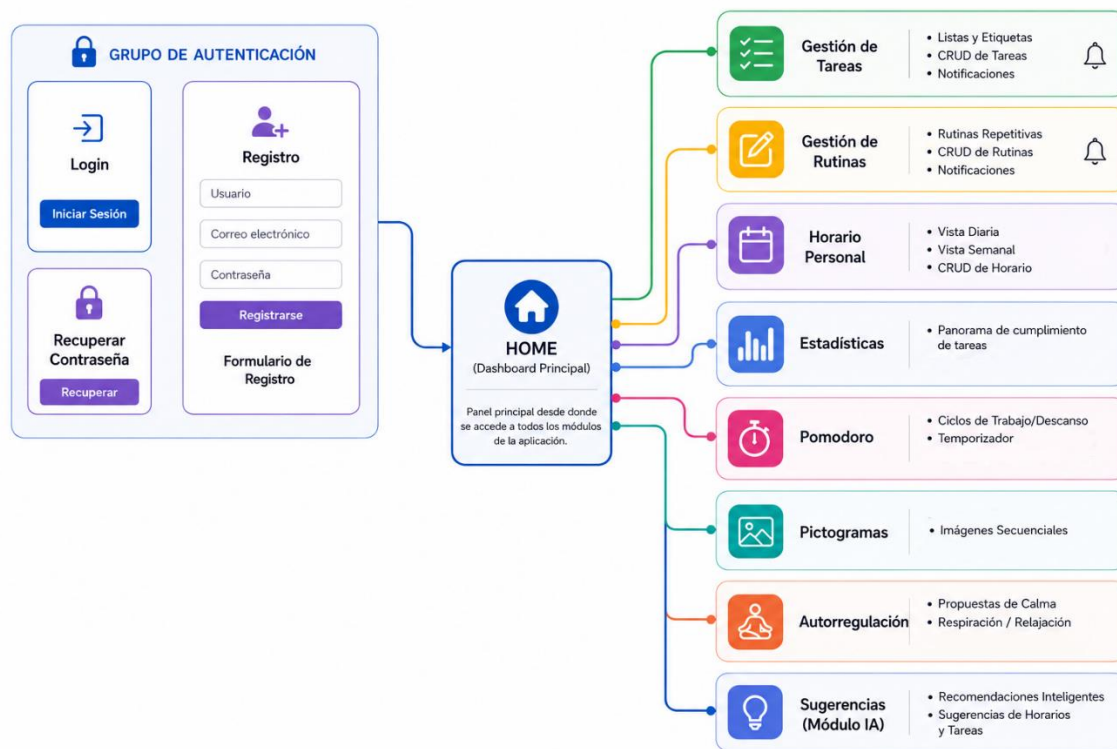


Figura 17: Flujo general de navegación de EverPlanApp.

Fuente: Elaboración propia

4.3 Validación de integración con backend

El frontend de EverPlanApp fue validado mediante pruebas de integración con los servicios REST disponibles, verificando el correcto consumo de los datos requeridos por la aplicación móvil. Estas pruebas permitieron comprobar que las distintas pantallas fueran capaces de

comunicarse adecuadamente con la API y representar la información obtenida de manera consistente.

Durante el proceso de validación se verificó:

- El envío correcto de credenciales desde la interfaz de inicio de sesión.
- La recepción y utilización de los tokens de autenticación entregados por la API.
- El consumo de endpoints protegidos para la obtención de información del usuario.
- El manejo de respuestas exitosas y mensajes de error provenientes de los servicios.
- La actualización de la interfaz en función de los datos recibidos.

Las solicitudes HTTP fueron centralizadas en un módulo de servicios, facilitando la reutilización de código y el mantenimiento de la aplicación. Asimismo, se comprobó que la información recuperada desde la API fuera mostrada correctamente en las distintas vistas de EverPlanApp, validando la integración entre la capa de presentación y los servicios consumidos.

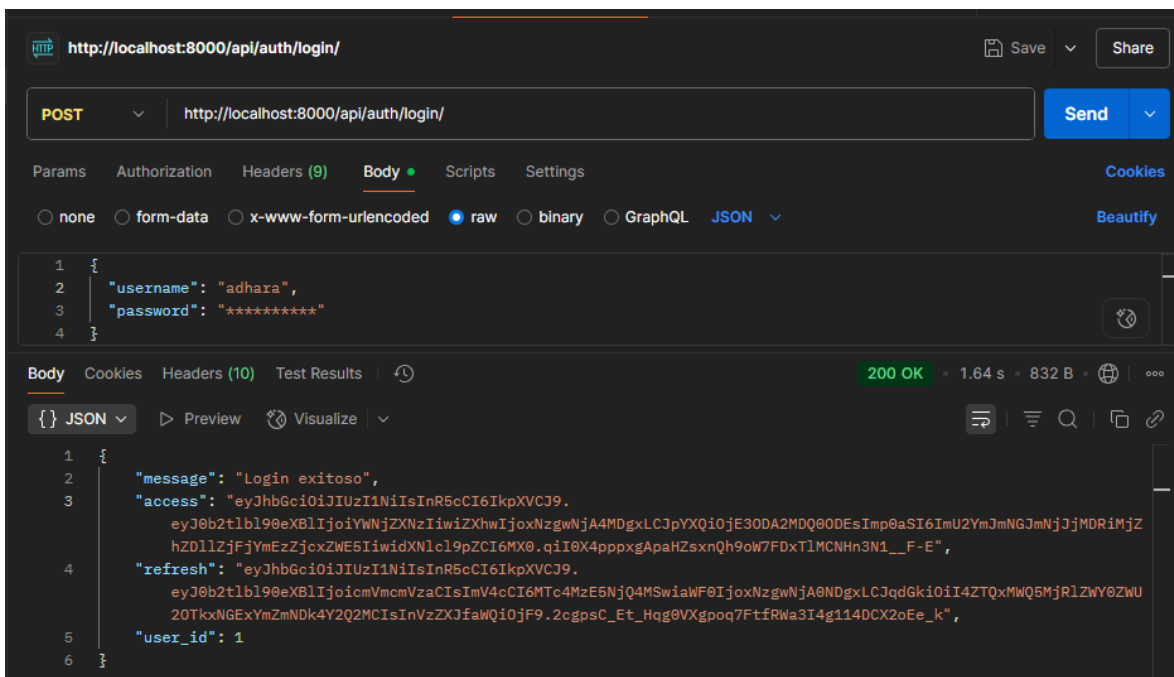


Figura 18: Autenticación y obtención de token JWT mediante la API REST en Postman

Fuente: Elaboración propia

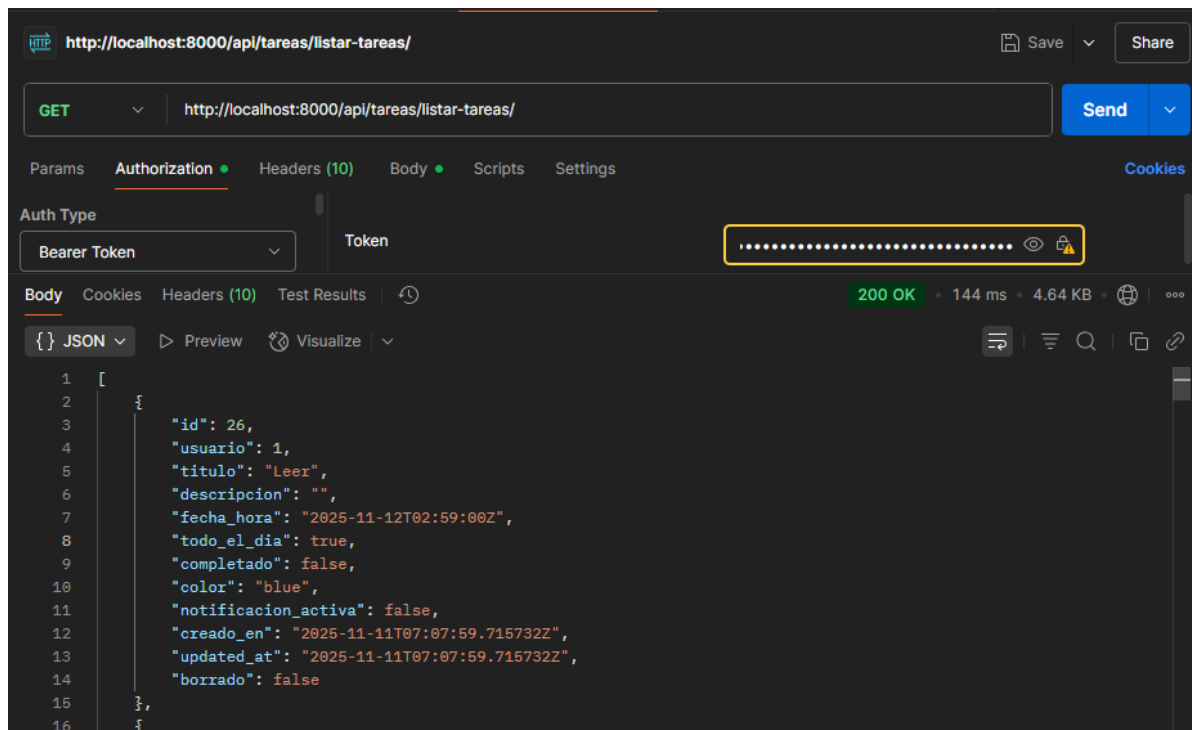


Figura 19: Consulta de tareas del usuario autenticado en Postman

Fuente: Elaboración propia



Figura 20: Visualización en interfaz de la tarea listada en postman

Fuente: Elaboración propia

4.4 Validación con usuarios finales

Una parte fundamental de la validación se realizó mediante pruebas informales con usuarios del público objetivo, incluyendo personas con dificultades de atención, organización y usuarios neurodivergentes.

Estas instancias permitieron observar el uso real de la aplicación y recoger comentarios sobre:

- Claridad de las pantallas.
- Facilidad para entender la navegación.
- Comprensión de los íconos, colores y pictogramas.
- Utilidad de las tareas, rutinas y horario personal.
- Sensación de orden y reducción de sobrecarga cognitiva.

Los usuarios destacaron positivamente la estructura clara de las pantallas, el uso de colores suaves y la separación por módulos. También se valoró el uso de pictogramas y rutinas como apoyo visual para anticipar actividades.

A partir de esta retroalimentación se realizaron ajustes menores, como simplificación de textos, mejora en el tamaño de botones y reorganización de algunos elementos visuales.

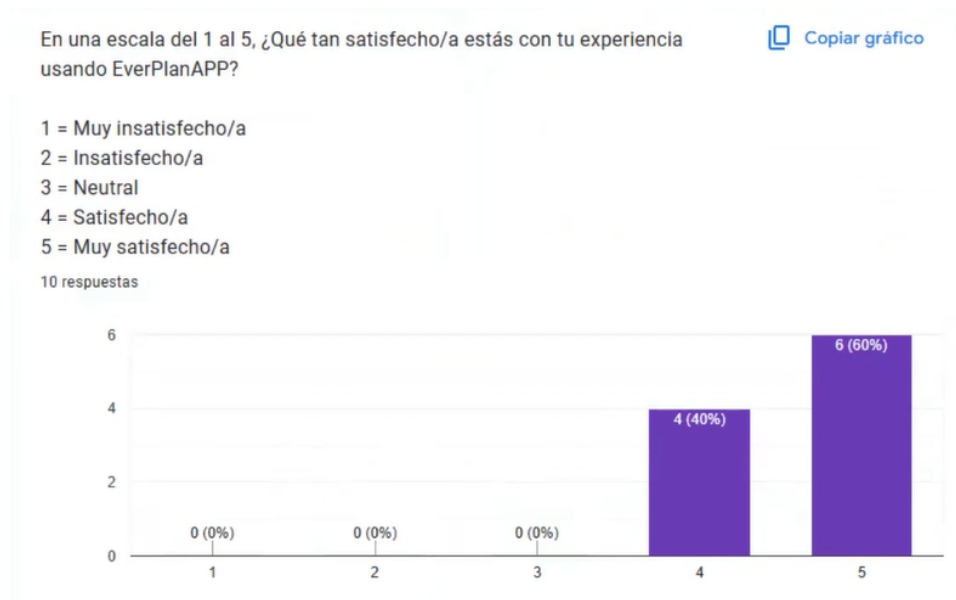


Figura 21: Evidencia encuesta CSAT con usuarios

Fuente: Elaboración propia

¿Hay algo que te gustaría mejorar o cambiar en la aplicación?

10 respuestas

La verdad me ha gustado mucho esta app ya que ayuda un problema que hoy en día es difícil de entender y que muchos estudiantes no saben organizarse

Podrían agregar los hábitos además de las rutinas como mencione en la entrevista

Faltaría que en tareas agregaran la vista como de calendario para ver el mes por lo demás todo me pareció bien

Personalización de colores similar a la que se encuentra en rutinas, pero para el apartado de tareas.

Estaría bueno poder recibir recordatorios o notificaciones antes de que empiecen las actividades.

Me gustaría poder marcar tareas como completadas desde la notificación.

Agregar una opción de modo oscuro, creo que ayudaría a muchas personas.

Me costó entender al principio cómo crear una rutina, tal vez una guía rápida ayudaría.

Sería útil poder agregar notas o detalles dentro de cada tarea o rutina.

En general la app es muy buena y fácil de usar, solo pequeños detalles por mejorar.

Figura 22: Sugerencias de usuarios en validación

Fuente: Elaboración propia

4.5 Validación con profesionales del área de inclusión

Durante el desarrollo de EverPlanApp se contó con orientación y retroalimentación de profesionales vinculados a áreas de neurología, inclusión y neurodivergencia. Esta validación permitió reforzar decisiones de diseño relacionadas con:

- Evitar estímulos visuales excesivos.
- Mantener estructuras predecibles y consistentes.
- Priorizar apoyos visuales sobre texto extenso.
- Facilitar la creación de rutinas y rituales repetitivos.

Esta investigación previa y durante el proceso permitió alinear la solución con principios de accesibilidad cognitiva y diseño inclusivo.

4.6 Resultados de la validación

Los resultados obtenidos permiten concluir que el frontend de EverPlanApp cumple con los objetivos planteados en cuanto a funcionalidad, usabilidad y enfoque inclusivo. La aplicación demostró ser estable en las pruebas realizadas y comprensible para usuarios con distintos niveles de experiencia tecnológica.

Si bien existen aspectos que pueden seguir mejorándose, como la ampliación de estadísticas o la personalización avanzada, la solución propuesta se considera válida para el contexto en el cual fue planteada.

CAPÍTULO 5: CONCLUSIONES

El desarrollo de EverPlanApp permitió aplicar de manera concreta los conocimientos adquiridos durante la formación académica, integrando aspectos técnicos, metodológicos y humanos en una solución digital orientada a una problemática real: las dificultades de organización, gestión del tiempo y autorregulación que enfrentan muchos estudiantes, especialmente aquellos con características asociadas a la neurodivergencia.

A través del diseño e implementación del frontend de la aplicación, fue posible comprender que el desarrollo de software no se limita únicamente a la construcción técnica de interfaces, sino que también requiere una comprensión profunda del contexto del usuario, sus necesidades, limitaciones y formas de interacción con la tecnología. En este sentido, EverPlanApp fue concebida como una herramienta de apoyo y acompañamiento, priorizando la claridad, la simplicidad y la reducción de estímulos visuales innecesarios.

Desde la perspectiva de los objetivos planteados, se concluye que estos fueron alcanzados de manera satisfactoria. Se logró diseñar e implementar una interfaz móvil funcional, accesible y coherente, integrando módulos como tareas, rutinas, horario personal, temporizador Pomodoro, pictogramas, herramientas de autorregulación y estadísticas, todos conectados a un backend y organizados bajo una estructura de navegación clara, intuitiva y predecible. Cada uno de estos módulos responde directamente a necesidades identificadas durante la etapa de investigación y análisis del proyecto.

Uno de los principales aportes del trabajo desarrollado corresponde a la incorporación consciente de principios de accesibilidad y diseño inclusivo en una aplicación de organización personal. A diferencia de muchas aplicaciones similares disponibles en el mercado, EverPlanApp evita la sobrecarga visual, la gamificación excesiva y los estímulos innecesarios, convirtiéndose en una alternativa especialmente adecuada para usuarios con TDAH, TEA u otras dificultades relacionadas con la atención y la regulación cognitiva. Este enfoque evidencia que es posible desarrollar soluciones tecnológicas modernas y funcionales sin comprometer la experiencia de usuarios que requieren entornos digitales más estructurados, comprensibles y amigables.

En términos técnicos, el desarrollo del frontend permitió aplicar tecnologías actuales como React Native y TypeScript, junto con librerías orientadas a la navegación, manejo de estado, consumo de APIs y gestión de notificaciones. Asimismo, la utilización de una arquitectura modular facilitó la organización del código, la reutilización de componentes y la escalabilidad futura del proyecto. La integración efectiva con el backend permitió validar flujos reales de autenticación, almacenamiento y sincronización de información, fortaleciendo la solidez y viabilidad de la solución implementada.

El proceso de validación con usuarios resultó especialmente relevante, ya que permitió contrastar las decisiones de diseño con experiencias reales de uso. La retroalimentación obtenida confirmó que la aplicación es comprensible, usable y percibida como una herramienta útil para la organización personal y académica. Del mismo modo, permitió

identificar oportunidades de mejora que fortalecen el carácter iterativo y evolutivo del proyecto. Esta etapa evidenció la importancia de considerar al usuario como un actor activo dentro del proceso de desarrollo y no únicamente como destinatario final del producto tecnológico.

No obstante, también se identificaron ciertas limitaciones. Al tratarse de un proyecto en constante evolución, existen funcionalidades que podrían ampliarse o perfeccionarse en futuras versiones, tales como una mayor personalización de la interfaz, análisis más avanzados dentro del módulo de estadísticas e integraciones adicionales con servicios externos. Asimismo, la validación fue realizada con un número acotado de usuarios, por lo que futuras etapas del proyecto podrían beneficiarse de pruebas más amplias y sistemáticas.

Como proyección futura, EverPlanApp presenta un importante potencial de crecimiento y expansión. El proyecto podría evolucionar mediante la incorporación de nuevas funcionalidades de apoyo académico, mejoras en accesibilidad e integración con calendarios y plataformas externas. De igual forma, existe la posibilidad de adaptar la aplicación a otros contextos, como entornos laborales o de planificación personal para personas adultas. Además, el proyecto abre oportunidades para continuar investigando sobre diseño de interfaces orientadas a la neurodiversidad, un área que aún presenta amplias posibilidades de innovación y desarrollo.

Finalmente, el presente trabajo constituye una experiencia significativa tanto a nivel técnico como formativo, permitiendo desarrollar una solución concreta y reflexionar sobre el impacto social del desarrollo de software. Asimismo, demuestra que la tecnología puede transformarse en una herramienta de apoyo real cuando es diseñada con empatía, responsabilidad y enfoque centrado en las personas. De esta manera, EverPlanApp no solo cumple con los objetivos académicos establecidos, sino que también representa una base sólida para futuras iniciativas profesionales y proyectos orientados a la inclusión digital y el bienestar de los usuarios.

REFERENCIAS BIBLIOGRÁFICAS

I. BIBLIOGRAFÍA

Autism&Uni, U. A. (2016). *Guía para la inclusión de estudiantes autistas en la universidad*.

Obtenido de UAM:

<https://www.uam.es/uam/media/doc/1606894726443/autismuni-guia3.pdf>

Cagiao, M. V., Fernández, R. V., Layna, C. H., Amarelo, M. M., Sanz, M. P., & Gutiérrez Ruiz, C. (2020). *La educación universitaria para estudiantes con TEA*. Obtenido de

Autismo.org.es: [https://autismo.org.es/wp-](https://autismo.org.es/wp-content/uploads/2018/09/informe_laeducacionuniversitariaparaestudiantesconte)

[content/uploads/2018/09/informe_laeducacionuniversitariaparaestudiantesconte](https://autismo.org.es/wp-content/uploads/2018/09/informe_laeducacionuniversitariaparaestudiantesconte)
[a_vf2021.pdf](https://autismo.org.es/wp-content/uploads/2018/09/informe_laeducacionuniversitariaparaestudiantesconte)

Caledu. (2023). *Participación de jóvenes con discapacidad y neurodivergentes en CFT*

chilenos. Obtenido de Caledu: <https://doi.org/10.31619/caledu.n60.1449>

Chile, M. d. (2024). *Guía de apoyo para la elaboración de un Plan de Equidad en la*

Educación Superior (PEES). Obtenido de Mineduc: [https://acceso.mineduc.cl/wp-](https://acceso.mineduc.cl/wp-content/uploads/2024/11/guia-PESD_compressed.pdf)
[content/uploads/2024/11/guia-PESD_compressed.pdf](https://acceso.mineduc.cl/wp-content/uploads/2024/11/guia-PESD_compressed.pdf)

Chile, R. H. (s.f.). *Neurodiversidad: una forma diferente de pensar y aprender*. Obtenido de

UC Christus: [https://www.redclinica.cl/especiales-de-](https://www.redclinica.cl/especiales-de-salud/especial_neurodiversidad.aspx)
[salud/especial_neurodiversidad.aspx](https://www.redclinica.cl/especiales-de-salud/especial_neurodiversidad.aspx)

Chile, U. d. (2024). *Ser una estudiante autista en la Universidad de Chile*. Obtenido de

uchile.cl: [https://uchile.cl/noticias/214655/ser-una-estudiante-autista-en-la-](https://uchile.cl/noticias/214655/ser-una-estudiante-autista-en-la-universidad-de-chile)
[universidad-de-chile](https://uchile.cl/noticias/214655/ser-una-estudiante-autista-en-la-universidad-de-chile)

Descúbreme, F. (2023). *Tecnología y neurodiversidad: herramientas inclusivas para el*

aprendizaje. Obtenido de descubreme.cl: [https://www.descubreme.cl/tecnologia-](https://www.descubreme.cl/tecnologia-y-neurodiversidad)
[y-neurodiversidad](https://www.descubreme.cl/tecnologia-y-neurodiversidad)

- Emily, E. (2024). *Challenges for autistic university students*. Obtenido de Authentically Emily: <https://www.authenticallyemily.uk/blog/challenges-for-autistic-university-students>
- Galvez, R. (2022). *Deserción universitaria mantiene cifras elevadas y salud mental asoma como una de sus causas* . Obtenido de latercera.com: <https://www.latercera.com/nacional/noticia/desercion-universitaria-mantiene-cifras-elevadas-y-salud-mental-asoma-con-fuerza-como-una-de-sus-causas/NUIDY5776FDU5MLHPLL6ZEEKWY>
- Garey, J. (2025). *¿Cómo pueden las escuelas apoyar a estudiantes neurodivergentes?* Obtenido de Child Mind Institute: <https://childmind.org/es/articulo/como-pueden-las-escuelas-apoyar-a-estudiantes-neurodiversos>
- Labs, T. (2025). *Tailwind CSS Documentation*. Obtenido de tailwindcss: <https://tailwindcss.com/docs>
- Neurodivergentes&co. (s.f.). *Población mundial de personas neurodivergentes*. Obtenido de Neurodivergentes.com: <https://neurodivergentes.com/poblacion-mundial-de-personas-neurodivergentes/>
- React. (s.f.). *React Documentation*. Obtenido de Reactjs.org: <https://reactjs.org/docs/getting-started.html>
- Services, A. W. (2025). *What is a Framework in Programming and Engineering?* Obtenido de Amazon Web Services: <https://aws.amazon.com/what-is/framework/>
- Vite. (2025). *Vite Documentation* . Obtenido de vitejs.dev: <https://vitejs.dev/>
- W3C. (2025). *Web Content Accessibility Guidelines (WCAG) 2.1* . Obtenido de W3C: <https://www.w3.org/TR/WCAG21/>
- Yáñez, C., Maira, P., Elgueta, C., Brito, M., Crockett, M. A., Troncoso, L., . . . Troncoso, M. (2021). *Estimación de la prevalencia de trastorno del Espectro Autista en población urbana chilena*. Obtenido de Scielo: https://www.scielo.cl/scielo.php?pid=S2452-60532021000400519&script=sci_arttext