

Grupo 14

ENTREGA FINAL

# PROYECTO DAW



Ana de la Torre Macías

Jéssica Marrón Carmona

Patricia Morato Jiménez

## **MEMORIA TÉCNICA DEL PROYECTO DE DESARROLLO DE APLICACIONES WEB**

### **Título del Proyecto: HomeTaskManager**

### **Aplicación Integral para la Gestión del Hogar**

Ciclo Formativo: Grado Superior en Desarrollo de Aplicaciones Web (DAW)

Proyecto final de Desarrollo de Aplicaciones Web

Integrantes del Equipo (Grupo 14):

Ana De La Torre Macías

Jéssica Marrón Carmona

Patricia Morato Jiménez



## **ÍNDICE**

### **1. INTRODUCCIÓN Y DESCRIPCIÓN DEL PROYECTO, 4**

- 1.1. Definición del Problema, 4
- 1.2. Objetivo y Alcance del Proyecto, 5

### **2. INVESTIGACIÓN Y ANÁLISIS, 6**

- 2.1. Mapa de Empatía (Identificación de Necesidades), 6
- 2.2. Estudio de Mercado (Viabilidad y Competencia), 10
- 2.3. Entrevistas Cualitativas y Técnicas de Ideación, 11
- 2.4. Modelo de Negocio, 12

### **3. PLANIFICACIÓN Y METODOLOGÍA, 13**

- 3.1. Metodología de Trabajo (SCRUM), 14
- 3.2. Roles del Equipo y Comunicación, 18
- 3.3. Planificación de Fases e Hitos, 19
- 3.4. Objetivos y Resultados Clave (OKR), 22

### **4. DISEÑO Y ARQUITECTURA TÉCNICA, 25**

- 4.1. Arquitectura de la Solución, 25
- 4.2. Herramientas y Tecnologías, 25
- 4.2. Base de Datos, 28
- 4.3. Funcionalidades Clave 31
- 4.4. Prototipado y Experiencia de Usuario (UX/UI), 39
- 4.5. Análisis de Usabilidad (10 Heurísticas de Nielsen), 44

## **5. DOCUMENTACIÓN E IMPLEMENTACIÓN, 50**

- 5.1. Software básico, AWS y Control de versiones, 50
- 5.2. Configuración del Backend, 58
- 5.3. Desarrollo de código, 64

## **6. PRUEBAS, DESPLIEGUE Y LANZAMIENTO, 68**

- 6.1. Técnicas de Testeo (Matriz de Feedback, Evaluación en Contexto), 68
- 6.2. Bancos y Casos de Prueba, 70
- 6.3. Planificación del Lanzamiento y Estrategia de Ingresos, 78
- 6.4. Plan de Recursos Humanos (RRHH), 80

## **7. UNA APP EN DESARROLLO, 82**

- 7.1. Últimos retoques, 82
- 7.2. Funciones Premium, 84
- 7.3. Resultado final, 85
- 7.4. Mirada hacia el futuro, 87

## **8. CONCLUSIÓN Y APRENDIZAJES, 88**

## **9. ANEXOS, 97**

- ANEXO 1. Entrevistas cualitativas y técnicas de ideación, 97
- ANEXO 2. Plan de viabilidad, 100
- ANEXO 3. Base de datos, 103
- ANEXO 4. Instalación y documentación, 103
- ANEXO 5. Desarrollo del código, 103
- ANEXO 6. Casos de prueba, 104

## **10. REFERENCIAS BIBLIOGRÁFICAS Y ENLACES, 121**

# 1. INTRODUCCIÓN Y DESCRIPCIÓN DEL PROYECTO

El presente proyecto tiene como finalidad el diseño y desarrollo de **HomeTaskManager**, una aplicación integral orientada a optimizar la gestión del hogar. Está dirigida principalmente a **familias, parejas jóvenes y compañeros de piso**, quienes suelen enfrentar dificultades para coordinar responsabilidades domésticas, administrar recursos y mantener un flujo de comunicación eficiente dentro del hogar.

Para el desarrollo de la solución se emplearán metodologías centradas en el usuario, como **Design Thinking**, así como diversas técnicas de investigación cualitativa que permitan comprender en profundidad las necesidades reales de los usuarios. A partir del análisis de estos insights, HomeTaskManager se perfila como una herramienta moderna, accesible y colaborativa que busca mejorar la organización diaria, reducir el estrés asociado a la gestión del hogar y fomentar dinámicas más equilibradas entre los residentes.

## 1.1. Definición del Problema

En muchos hogares, la **falta de coordinación** y la **ausencia de un sistema centralizado** para organizar tareas y responsabilidades genera numerosas complicaciones. Los usuarios suelen tener dificultades para:

- **Gestionar y distribuir tareas domésticas** de manera justa y eficiente.
- **Almacenar y localizar documentos importantes** (facturas, garantías, contratos, etc.).
- **Recordar fechas clave** como pagos, citas médicas, vencimientos o eventos familiares.
- **Mantener una comunicación fluida** sobre los asuntos cotidianos del hogar.

Estas dificultades provocan problemas como **estrés, sobrecarga mental, discusiones, falta de organización** y pérdida de tiempo. Además, la multiplicidad de aplicaciones actuales (recordatorios, notas, calendarios, gestores de documentos) fragmenta aún más el proceso, obligando a los responsables del hogar a cambiar constantemente entre herramientas, sin lograr una visión unificada.

En este contexto, resulta evidente la necesidad de una solución **única, colaborativa y flexible**, capaz de centralizar toda la información y facilitar la convivencia mediante una gestión más eficiente del hogar.

## 1.2. Objetivo y Alcance del Proyecto

### Objetivo General

Desarrollar una aplicación móvil y web que se posicione como la herramienta líder en **gestión integral del hogar**, permitiendo a los usuarios organizar tareas, almacenar documentos, coordinar eventos y administrar recursos de manera sencilla y colaborativa.

### Objetivos Específicos

- Diseñar una interfaz intuitiva que facilite el acceso a todas las funcionalidades.
- Implementar un sistema de **gamificación** que motive a los usuarios a completar las tareas asignadas.
- Permitir la **colaboración en tiempo real** entre los miembros del hogar.
- Centralizar la gestión de tareas, eventos, recordatorios, citas y documentos.
- Ofrecer herramientas para la **gestión de presupuestos** y gastos del hogar.
- Integrar sistemas de notificaciones inteligentes que reduzcan olvidos.

### Alcance del Proyecto

El proyecto incluirá:

- **Análisis de usuarios y definición de requisitos** mediante técnicas de investigación cualitativa y Design Thinking.
- **Diseño UX/UI** basado en prototipos de baja y alta fidelidad.
- Desarrollo de funcionalidades clave:
  - Gestión de tareas y listas colaborativas.
  - Sistema de recompensas y niveles (gamificación).
  - Calendario compartido y recordatorios automáticos.
  - Almacén digital para documentos del hogar.
  - Gestión de citas y presupuestos.
- Validación con usuarios mediante pruebas de usabilidad.
- Propuesta de mejoras futuras y posibilidad de ampliación (integración con dispositivos inteligentes del hogar, analíticas de uso, etc.).

HomeTaskManager aspira a convertirse en una herramienta esencial para cualquier tipo de hogar, fomentando una convivencia más organizada, eficiente y armoniosa.

## 2. INVESTIGACIÓN Y ANÁLISIS

### 2.1. Mapa de Empatía (Identificación de Necesidades)

**Usuario Objetivo:** *Familias y personas responsables de la gestión del hogar*

El mapa de empatía es una herramienta que sirve para descubrir cómo piensa, siente y se comporta un posible público objetivo, para comprender mejor sus deseos y necesidades. Para ello analizaremos **cinco esferas del usuario**: ver, necesitar, oír, pensar y sentir, y decir y hacer.

Hemos creado un perfil de usuario que representa a nuestro público objetivo. Una persona "tipo" de nuestra actual sociedad, que a la vez que trabaja desea organizar mejor su hogar y su vida personal, a la cual le falta tiempo para llevarlo todo a cabo y así mejorar su calidad de vida.

#### ¿Qué piensa y siente?

Se preocupa por mantener todo organizado, pero a menudo siente **estrés** por olvidar tareas importantes.

Le **frustra** perder documentos esenciales o no encontrarlos cuando los necesita.

**Desea** una solución fácil y accesible para gestionar responsabilidades familiares.

Siente **alivio** cuando todo está en orden y sin imprevistos.

#### ¿Qué ve?

Un hogar con **tareas acumuladas** y **responsabilidades** mal distribuidas.

Muchas **aplicaciones separadas** para distintas funciones (recordatorios, notas, almacenamiento en la nube).

**Soluciones poco integradas** que no cubren todas sus necesidades.

**Publicaciones y contenido** sobre productividad y organización familiar.

### **¿Qué dice y hace?**

Expresa que **necesita más tiempo** y ayuda para gestionar el hogar.

**Comenta** con familiares o amigos sobre su carga de tareas y la falta de herramientas adecuadas.

**Intenta** usar listas de tareas, agendas o notas adhesivas, pero no siempre son efectivas.

**Busca en internet** aplicaciones o métodos para mejorar la organización.

### **¿Qué oye?**

**Recomendaciones** de amigos o familiares sobre apps de organización.

**Consejos** en redes sociales sobre métodos de productividad.

**Quejas** de otros miembros del hogar sobre tareas olvidadas o mal coordinadas.

**Publicidad** de herramientas de planificación y almacenamiento en la nube.

### **¿Qué le duele?**

**Falta de tiempo** para organizar documentos y tareas.

**No recordar fechas** importantes como pagos, vencimientos o reuniones familiares.

Desorden en la **asignación de responsabilidades** dentro del hogar.

Dificultad para **acceder rápidamente a documentos** importantes cuando los necesita.

### **¿Qué necesita o qué le motiva?**

Un **sistema unificado** donde pueda gestionar tareas y almacenar documentos.

Recordatorios automáticos que le avisen de fechas importantes.

Una **interfaz sencilla y accesible** para todos los miembros del hogar.

Sentir más **control** de su tiempo y organización.

## **2.2. Estudio de Mercado (Viabilidad y Competencia)**

Con tal de apoyar nuestro proyecto, hemos realizado una primera prospección del mercado para ver qué opciones existen actualmente relacionadas con nuestra idea de proyecto y el público al que vamos a dirigirnos:

### **1. Objetivo del Estudio**

El propósito de este estudio es **entender las necesidades** de las personas responsables de la gestión del hogar, **analizar la competencia** e **identificar oportunidades** en el mercado para desarrollar una aplicación que solucione estos problemas de forma eficaz.

### **2. Público Objetivo**

La aplicación está dirigida a:

- **Familias con hijos:** Que necesitan organizar tareas, responsabilidades y documentos importantes.
- **Parejas jóvenes:** Que buscan una herramienta para gestionar su hogar de manera colaborativa.
- **Compañeros de piso:** Que desean coordinar tareas y gastos de manera equitativa.
- **Personas que viven solas:** Que necesitan organizar sus tareas y documentos de manera eficiente.

### **3. Análisis de la Competencia**

Existen múltiples apps disponibles relacionadas con la gestión de tareas del hogar. Algunas de ellas son:

- **Choremonster**, una app disponible para iOS donde los padres incluyen tareas domésticas que sus hijos deben realizar, para así otorgarles una puntuación, un tiempo para llevarlas a cabo, y una recompensa.

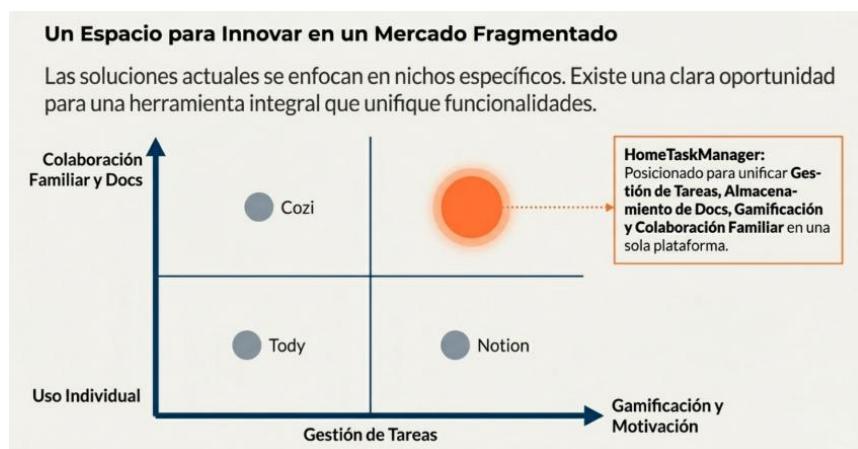
- **Sweeypy** ayuda a repartir las tareas del hogar creando una competición entre los miembros de la unidad familiar para generar motivación.

- **OurHome** administra las tareas creando listas, cuadrantes compartidos y calendarios, y ofrece un sistema de notificaciones internas.

- **Tody** es otra app diseñada para la gestión de tareas de limpieza del hogar en función de fechas específicas donde podemos ver claramente el progreso de los objetivos.

- **Cozi Family Organizer**, aporta funciones como listas de la compra, ingredientes, recetas, además de una versión “Gold” con otras funcionalidades más avanzadas.

- **Notion** es un gran sistema de productividad que permite adaptar el software a las necesidades de cada usuario mediante módulos y tareas muy avanzadas. Ofrece también la posibilidad de incluir bases de datos y generar vistas. Una herramienta compleja pero que puede servir de inspiración.



Así pues, vemos distintas aplicaciones con múltiples funciones. Algunas enfocadas al orden y la limpieza, otras enfocadas a la alimentación, otras a motivar a toda la familia, y otras para personas que quieren un orden más completo en la organización de todos los aspectos.

Nuestra idea es reunir en una aplicación estas funcionalidades, creando un software que combine las mejores características de estas aplicaciones, como la

flexibilidad en la gestión de tareas, un sistema de recompensas, una interfaz amigable, y un lugar donde reunir toda la documentación importante de forma práctica y accesible.

#### 4. Tendencias del Mercado

Aumento de la demanda de aplicaciones de productividad:

Según un informe de Statista, el mercado de **aplicaciones de productividad** ha crecido un **20% anual** en los **últimos 5 años**.

*Enfoque en la colaboración familiar:*

Las aplicaciones que fomentan la colaboración y la organización familiar están ganando popularidad, especialmente tras el aumento del teletrabajo.

*Preferencia por soluciones integrales:*

Los usuarios buscan aplicaciones que resuelvan múltiples problemas en una sola plataforma, en lugar de usar varias apps separadas.



#### 5. Conclusiones

El mercado muestra una clara oportunidad para una aplicación que:

- Integre gestión de tareas y almacenamiento seguro de documentos.
- Permita colaboración familiar y reparto de responsabilidades.
- Incluya recordatorios automáticos personalizables.
- Ofrezca un diseño limpio y fácil de usar.

## 2.3. Entrevistas Cualitativas y Técnicas de Ideación

Con tal de apoyar la necesidad real de nuestro proyecto y obtener una clara orientación sobre lo que realmente necesita el público objetivo, se realizaron entrevistas cualitativas y dos diagramas de técnicas de ideación (“**5 para qué**” y “**¿Qué, cómo y por qué?**”).

- Las entrevistas corroboraron problemas comunes como la **falta de colaboración entre miembros del hogar**, la necesidad de gestionar el tiempo para preparar comidas por adelantado, y la demanda de un sistema que recuerde actividades sin necesidad de preguntar.
- Los diagramas de técnicas de ideación revelaron que los usuarios necesitan una aplicación que les permita **ahorrar tiempo** y **reducir el estrés**, lo que llevó a la definición de funcionalidades clave, como la **gamificación** y la **personalización**.



Diagrama de los 5 Para qué, Design Thinking

\* Las entrevistas cualitativas y el diagrama de “¿Qué, cómo y por qué?” se adjuntan en el ANEXO 1.

## 2.4. Modelo de Negocio

El modelo de negocio se estructuró mediante el Business Model Canvas. El gasto total estimado para la creación de la aplicación asciende a **16.200€**, que incluye costes internos (personal, infraestructura) y externos (software/herramientas, consultoría, I+D, legales y marketing).

A continuación se muestra el Business Model Canvas, sobre el cual trabajamos para estimar costes:

## Business Model Canvas



\* El estudio de viabilidad completo se encuentra en el ANEXO 2.

### 3. PLANIFICACIÓN Y METODOLOGÍA

Nuestra app persigue ser la **aplicación líder** en la gestión integral del hogar, transformando la manera en que familias, parejas y compañeros de piso organizan sus tareas y documentos. Nuestra meta es mejorar la calidad de vida a través de una solución innovadora, que integre colaboración, personalización y herramientas intuitivas, fomentando hogares más organizados, armoniosos y felices.



Así pues, podemos resumir las necesidades del cliente de la siguiente manera:

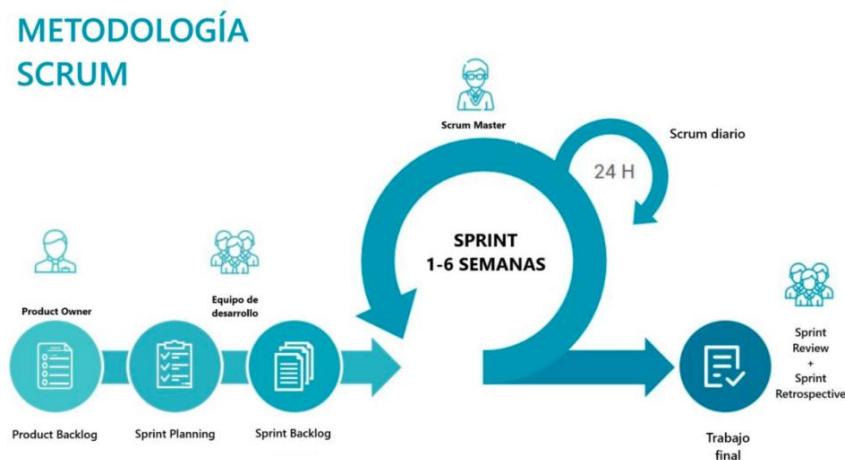
- Motivación** para realizar las **tareas** del hogar
- Motivación para realizar dichas tareas de forma **colaborativa**
- Organización** del hogar y/o del núcleo familiar
- Organización de **documentos** y simplificar su almacenamiento y su búsqueda
- Recordatorios** y notificaciones
- Calendario** compartido
- Elaborar **presupuestos** y controlar gastos e ingresos

- Elaborar **listas**
- Reducir el estrés y mejorar su **calidad de vida**
- Disponer de una app que **reúna** todas estas funcionalidades
- Una accesibilidad a la aplicación desde **múltiples dispositivos** para facilitar su uso por parte de todos los miembros del hogar.
- Que sea una interfaz **fácil e intuitiva** para todos los miembros de la familia.

Partiendo de esta base, y teniendo muy presente estas necesidades en todo momento, desarrollamos una estrategia y una metodología de trabajo para llevar a cabo nuestro objetivo.

### 3.1. Metodología de Trabajo (SCRUM)

El equipo decidió utilizar el modelo autoincremental implementado a través de la metodología **SCRUM**. Este enfoque permite el desarrollo en sprints cortos, entregando versiones funcionales periódicas (incrementos), lo que garantiza la transparencia, la adaptabilidad a los cambios, y la participación activa del cliente. Las fases se solapan y los equipos son autogestionados para agilizar la toma de decisiones.



## **Puntos clave de SCRUM:**

**Sprints:** Periodos cortos donde se desarrolla una funcionalidad completa.

**Product Backlog:** Lista priorizada de tareas y requisitos.

**Sprint Backlog:** Selección de tareas del backlog a completar durante el sprint.

## **Reuniones clave:**

- o **Sprint Planning:** Planificación de objetivos del sprint.
- o **Daily Stand-up:** Reuniones breves para sincronizar el equipo.
- o **Sprint Review:** Presentación del incremento desarrollado.
- o **Sprint Retrospective:** Reflexión para identificar mejoras.

**Incremento:** Entregables funcionales al final de cada sprint.

**Transparencia y adaptabilidad:** Visibilidad del proyecto y capacidad de ajustar prioridades según necesidades.

Se comienza con la visión general del resultado que se desea, y a partir de ella se especifica y da detalle a las funcionalidades que se desean obtener en primer lugar. Cada ciclo de desarrollo o iteración (sprint) finaliza con la entrega de una parte operativa del producto (incremento). La duración de cada sprint puede ser de una hasta seis semanas, aunque se recomienda que no exceda de un mes.

Con este enfoque, el equipo puede trabajar de manera ágil, asegurando resultados óptimos y entregas regulares que satisfacen las expectativas del cliente y del proyecto en su conjunto.

## **Revisión de las iteraciones**

Al finalizar cada sprint se revisa funcionalmente el resultado, con todos los implicados en el proyecto. Por tanto, la duración del sprint es el período de tiempo máximo para descubrir planteamientos erróneos, mejorables, o malinterpretaciones en las funcionalidades del producto.

## **Desarrollo incremental**

No se trabaja con diseños o abstracciones. El desarrollo incremental ofrece al final de cada iteración una parte de producto operativa, que se puede usar, inspeccionar y evaluar.

## **Solapamiento de fases**

Durante la construcción se depura el diseño y la arquitectura, y no se cierran en una primera fase del proyecto. Las distintas fases que el desarrollo en cascada realiza de forma secuencial, una tras otra, en scrum se solapan y avanzan de forma simultánea.

## **Autogestión**

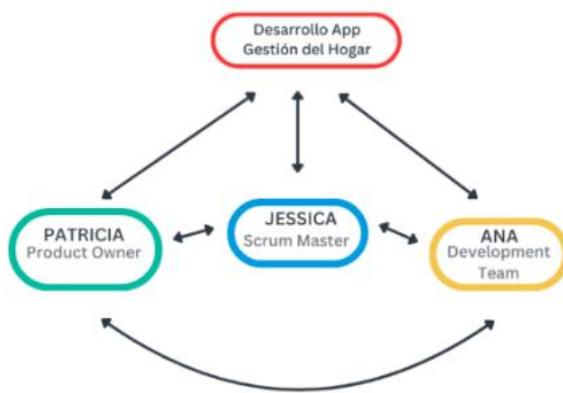
La gestión predictiva asigna al rol de gestor del proyecto la responsabilidad de su gestión y resolución. En scrum los equipos son autogestionados, con un ámbito de decisión suficiente para adoptar las resoluciones que consideren oportunas. Esto agiliza la toma de decisiones y permite responder con rapidez ante imprevistos.

## **Colaboración**

Todos los miembros del equipo colaboran de forma abierta con los demás, según sus capacidades y no según su rol o su puesto. Mediante la autogestión y la colaboración se puede gestionar con solvencia la labor que de otra forma realizaría un gestor de proyectos.

### 3.2. Roles del Equipo y Comunicación

#### ORGANIGRAMA



Como el equipo es pequeño (3 miembros), en el organigrama aparecen flechas bidireccionales que indican la interacción de todos los miembros en todas las tareas (cada miembro puede desempeñar en algún momento múltiples roles o tareas).

#### Scrum Master

Responsabilidades principales: Facilitar el proceso Scrum, eliminar impedimentos, asegurar que el equipo siga las prácticas de Scrum.

#### Product Owner

Responsabilidades principales: Gestionar el Product Backlog, priorizar las funcionalidades y definir los objetivos del sprint.

#### Development Team

Responsabilidades principales: Desarrollar funcionalidades, participar en la planificación y revisión de los sprint.

## Plan de comunicación

| CANAL              | USO  | FRECUENCIA                                | RESPONSABLE  |
|--------------------|--|---|--|
| Campus virtual     | Publicación de documentos, entregables y debates formales. Aportación de ideas, revisión de documentos.                          | Establecer fechas límite para cada tarea. | Todos los miembros   |
| Correo electrónico | Tablero de discusión. Distribución de tareas. Compartir archivos   | Según necesidad                           | Jéssica, comunicación con el tutor.<br>Todos los miembros, otras tareas. |
| WhatsApp           | Mensajes rápidos, coordinación diaria, resolución de dudas y recordatorios. Avisos cuando se publique algo en el campus virtual. | Diario o según necesidad                  | Todos los miembros   |
| Collaborate        | Debatir ideas importantes y trabajar en las tareas   | Semanal o según necesidad                 | Todos los miembros   |

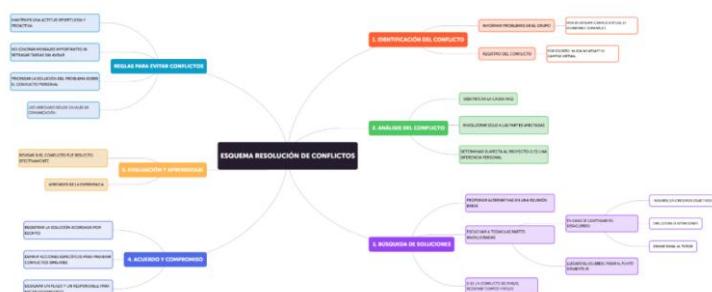
**Jéssica:** Coordinación de reuniones, creación de índices, compartir documentos y abrir debates. Creación de tareas e hilos durante las videollamadas.

**Ana:** Encargada de la Minuta. Recoge la interacción, la comunicación y los detalles del proceso para elaborar el documento final.

**Patricia:** Unifica y resume el entregable con todo el material creado. Realiza la entrega final.

**Creación del Portfolio:** Todos los miembros.

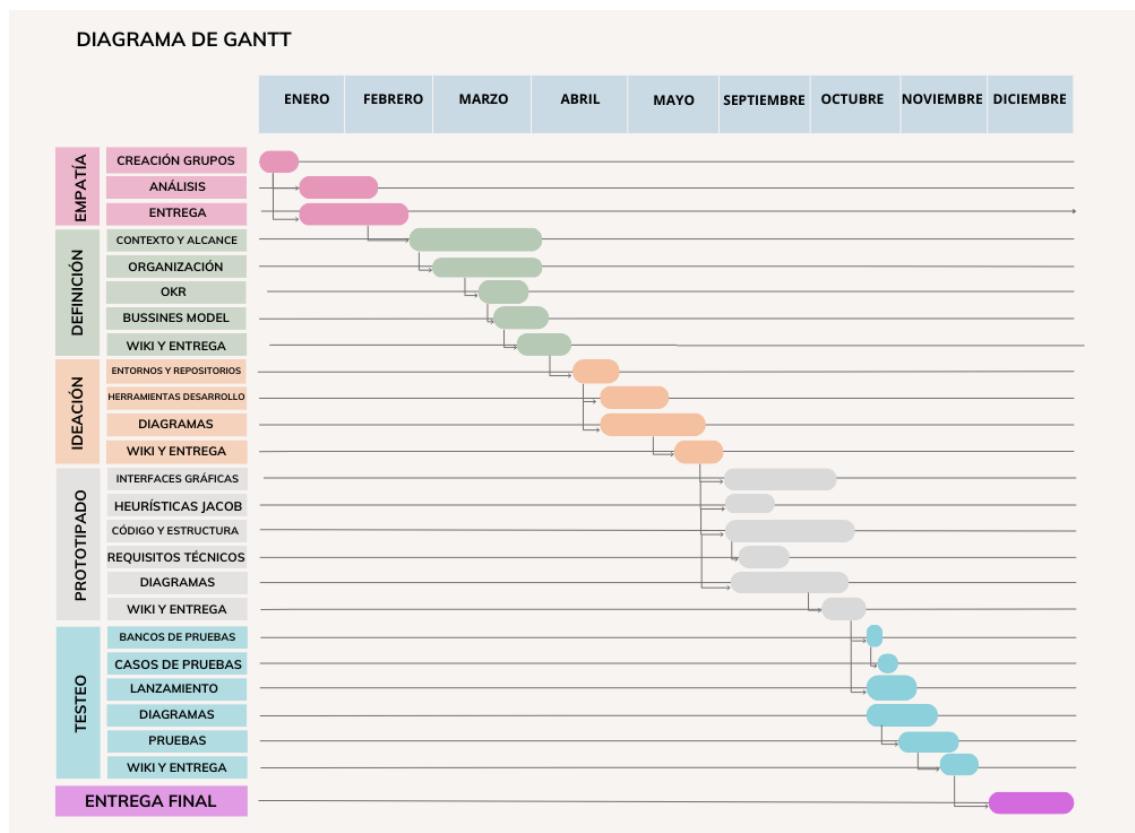
Además de definir roles y elaborar un plan de comunicación, se elaboró un esquema de resolución de conflictos y unas reglas de comunicación que incluye la frecuencia de reuniones y una serie de compromisos y acuerdos.



### 3.3. Planificación de Fases y Hitos

La ejecución del proyecto se dividió en dos partes principales: la Primera Parte (Febrero-Junio), enfocada en la Empatía, Definición e Ideación, y la Segunda Parte (Septiembre-Diciembre), dedicada al Prototipado, Testeo y la Entrega Final.

Se realizó un Diagrama de Gantt para exponer de forma visual los tiempos establecidos:



### PRIMERA PARTE DEL PROYECTO (Febrero - junio)

#### Fase empatía:

- Evaluación de ideas y creación de los grupos de trabajo.
- Análisis realizado por medio de dos de las técnicas propuestas por el docente.

#### Fase definición:

- Organización del Proyecto (contexto y alcance del proyecto)
- Planificación del Proyecto

- Organización del trabajo grupal: (Definición de los roles del equipo, plan de comunicación y resolución de conflictos)
- Objetivos OKR ó SMART
- Business Model Canvas
- Diagramas con las técnicas propuestas por el docente

#### **Fase ideación:**

- Selección de un entorno integrado de desarrollo y de un sistema de control de versiones o repositorio, e incorporar al docente en el mismo.
- Descripción de las herramientas de desarrollo y construcción del Producto (Sistema Operativo, Sistema Gestor de Bases de Datos, Frameworks...)
- Diagrama de estructura de la base de datos (Diagrama Entidad/Relación), diagrama de Despliegue y diagrama de estructura del producto.

### **SEGUNDA PARTE DEL PROYECTO ( Septiembre- diciembre)**

#### **Fase prototipado:**

- Mockups de las interfaces gráficas del Producto.
- Análisis de las Heurísticas de Jakob Nielsen.
- Resumen del código desarrollado, estructura en el entorno de desarrollo.
- Especificaciones y/o requisitos técnicos del Producto para el despliegue de la aplicación en un entorno de producción
- Diagramas con las técnicas propuestas por el docente

#### **Fase testeo:**

- Definición de los “tipos” de bancos de pruebas a implementar.
- Plantilla de los Casos de Prueba y los resultados del testeo utilizando herramientas específicas.
- Planificación del lanzamiento del producto.
- Diagramas con las técnicas propuestas por el docente.
- Pruebas con usuarios reales, recopilación de feedback sobre los prototipos, identificación de mejoras necesarias y ajustes para optimizar la app.

#### **Entrega Final:**

- Entrega de la memoria técnica del proyecto/producto.
- Entrega del video con demostración del funcionamiento del producto.

### **3.4. Objetivos y Resultados Clave (OKR)**

OKR son las siglas en inglés de **Objectives and Key Results** (Objetivos y Resultados Clave), una metodología de gestión para establecer objetivos claros y medibles en una empresa, equipo o a nivel individual. Se compone de dos partes: Objetivos, que son metas cualitativas, ambiciosas y de alto nivel, y Resultados Clave, que son métricas cuantificables que indican el progreso hacia esos objetivos.

Con estos objetivos alcanzaremos las metas propuestas del proyecto. Se irán revisando a lo largo del desarrollo para comprobar que se están llevando a cabo de forma correcta.

#### **Objetivo (1): Mejorar la experiencia de usuario de la aplicación.**

- o KR1: Realizar pruebas de usuario con al menos 10 familias y obtener una valoración positiva del 90% en usabilidad.
- o KR2: Incorporar notificaciones inteligentes antes de la fase de prueba del prototipo.
- o KR3: Identificar y resolver al menos 4 problemas de usabilidad detectados en las pruebas de usuario.
- o KR4: Implementar mejoras en la interfaz basadas en el feedback del 80% de los usuarios participantes.

#### **Objetivo (2): Crear una base de datos eficiente y estructurada.**

- o KR1: Diseñar y documentar el esquema de la base de datos con al menos 5 tablas principales.
- o KR2: Implementar la base de datos y validar su correcto funcionamiento con consultas básicas.
- o KR3: Definir y establecer relaciones entre las tablas asegurando integridad referencial.
- o KR4: Garantizar la seguridad de los datos aplicando encriptación y controles de acceso adecuados.

**Objetivo (3): Elaborar prototipos funcionales y representativos.**

- o KR1: Diseñar un prototipo completo que represente la funcionalidad principal.
- o KR2: Implementar las principales interacciones y navegación dentro del prototipo.
- o KR3: Asegurar que el prototipo incluya todos los elementos clave definidos en el diseño inicial.
- o KR4: Documentar el proceso de prototipado con especificaciones y justificación de cambios.

**Objetivo (4): Mejorar el sistema de acceso y autenticación de usuarios.**

- o KR1: Implementar un sistema de contraseña con requisitos básicos de seguridad.
- o KR2: Configurar autenticación multifactor para los usuarios.
- o KR3: Reducir los errores de inicio de sesión.
- o KR4: Asegurar que todos los usuarios pueden recuperar sus cuentas fácilmente en caso de problemas.

**Objetivo (5): Cumplir con los objetivos establecidos en el proyecto.**

- o KR1: Completar el desarrollo de todos los puntos del proyecto antes de la fecha límite.
- o KR2: Documentar las etapas del proyecto de manera detallada y clara.
- o KR3: Lograr un nivel de comprensión y ejecución óptimo en todas las tareas realizadas.
- o KR4: Recibir retroalimentación positiva sobre el proyecto por parte del profesor.

**Objetivo (6): Implementar un sistema de almacenamiento de documentos**

- o KR1: Implementar una opción para digitalizar los documentos físicos.
- o KR2: Implementar un sistema de clasificación que permita reducir el tiempo de búsqueda de documentos.
- o KR3: Asegurar que los documentos subidos sean indexados correctamente y se puedan buscar de forma rápida.
- o KR4: Recopilar y aplicar retroalimentación de los primeros usuarios para mejorar la experiencia de uso.

**Objetivo (7): Diseñar una interfaz gráfica intuitiva y visualmente atractiva.**

- o KR1: Desarrollar y probar un prototipo interactivo.
- o KR2: Implementar un diseño consistente con la guía de estilo definida, asegurando una coherencia en todos los elementos visuales.
- o KR3: Usar herramientas de diseño como Figma o Adobe XD para trabajar colaborativamente en el desarrollo.
- o KR4: Lograr una valoración promedio de satisfacción superior a 8/10 en encuestas realizadas a usuarios tras la implementación del diseño.

**Objetivo (8): Implementar estrategias de gamificación para aumentar el compromiso y la participación de los usuarios.**

- o KR1: Diseñar e integrar al menos tres mecánicas de juego (como puntos, niveles, o recompensas) en la aplicación en un plazo de X semanas.
- o KR2: Incrementar el tiempo promedio de interacción de los usuarios en la aplicación tras la implementación.
- o KR3: Aumentar el número de usuarios recurrentes dentro de los primeros tres meses después de aplicar las estrategias de gamificación.
- o KR4: Realizar un seguimiento semanal de las gamificaciones.

**Objetivo (9): Incrementar la visibilidad y atracción de usuarios hacia el producto o servicio.**

- o KR1: Diseñar e implementar al menos 3 campañas creativas en redes sociales enfocadas en aumentar las interacciones.
- o KR2: Desarrollar estrategias de marketing digital que impulsen el tráfico al sitio web o aplicación durante un periodo definido.
- o KR3: Publicar testimonios o casos de éxito de los primeros usuarios.
- o KR4: Recibir feedback positivo sobre la comunicación de marca por parte de los clientes tras las iniciativas de marketing.

**Objetivo (10): Aumentar la eficacia y el éxito en la consecución de metas estratégicas.**

- o KR1: Desarrollar y completar un plan de acción detallado para varias metas clave dentro de un plazo definido.
- o KR2: Optimizar los procesos actuales para reducir los tiempos de cumplimiento de las metas.
- o KR3: Mejorar la tasa de cumplimiento de metas estratégicas mediante la implementación de nuevas metodologías de trabajo.
- o KR4: Documentar los procesos y funcionalidades implementados en la aplicación.

## 4. DISEÑO Y ARQUITECTURA TÉCNICA

### 4.1. Arquitectura de la Solución

#### 1. Introducción

Esta aplicación utilizará **Apache Cordova** para desarrollar una solución híbrida con un solo código base. El backend, la base de datos, y los servicios relacionados estarán alojados en **Amazon Web Services** (AWS), aprovechando el Free Tier que ofrece recursos gratuitos por 12 meses.

**Las herramientas y los lenguajes utilizados son:**

- *Apache Cordova*
- *Android Studio*
- *Node.js*
- *Visual Studio Code*
- *Gradle*
- *Jdk*
- *AWS: Amazon Cognito, AppSync, IAM...*
- *DynamoDB*
- *S3*
- *Lambda*
- *GraphQL, JavaScript, HTML5, CSS*

#### 2. Infraestructura del Sistema

**Windows con Visual Studio:**

- Sistema Operativo: Windows 10 o 11 como entorno principal.

Requisitos:

- Procesador: Intel Core i5/i7 o equivalente.
- RAM: 16 GB para ejecutar Visual Studio
- Almacenamiento: SSD con al menos 50 GB libres.

**Razón:** Windows facilita la integración de Visual Studio y la gestión del código para Cordova, además de herramientas de AWS como el AWS CLI.

### **Servidor Backend (En la Nube con AWS):**

- Servicio: AWS Amplify Backend, usando AppSync (GraphQL API) y DynamoDB.

### Configuración del backend:

- API: gestionada con **AWS AppSync**.
- Base de datos: Tablas **DynamoDB** autogeneradas desde schema.graphql mediante amplify pull.
- Orígenes de datos: Directamente conectados a DynamoDB mediante resolvers GraphQL.
- Entorno: Backend serverless (sin necesidad de administrar instancias EC2).

**Razón:** Amplify permite construir y desplegar backends escalables y sin servidor de forma rápida, con integración automática de API, base de datos y autenticación. Ideal para aplicaciones Node.js que consumen APIs modernas y necesitan alta disponibilidad sin administrar infraestructura.

### **Base de Datos (NoSQL simulando relaciones):**

- Servicio: **AWS AppSync + DynamoDB**
- Modelo de Datos: Usamos **GraphQL** como interfaz para definir el esquema y AppSync lo traduce automáticamente a tablas DynamoDB.
- Configuración: La base de datos se crea y gestiona automáticamente al ejecutar amplify add api y definir modelos como Tarea, Evento, etc.
- Simulación de relaciones: Aunque DynamoDB es NoSQL, AppSync permite simular relaciones mediante claves foráneas manuales (@connection o campos de referencia). Esto nos ha permitido relacionar, por ejemplo, tareas con usuarios, o tareas con eventos.

### **Almacenamiento de Archivos:**

- Servicio: **Amazon S3** (Simple Storage Service)

**Razón:** S3 es ideal para almacenar imágenes o documentos asociados a tareas o usuarios. Es altamente escalable, seguro y puede integrarse fácilmente con Amplify y AppSync.

## Autenticación y Usuarios

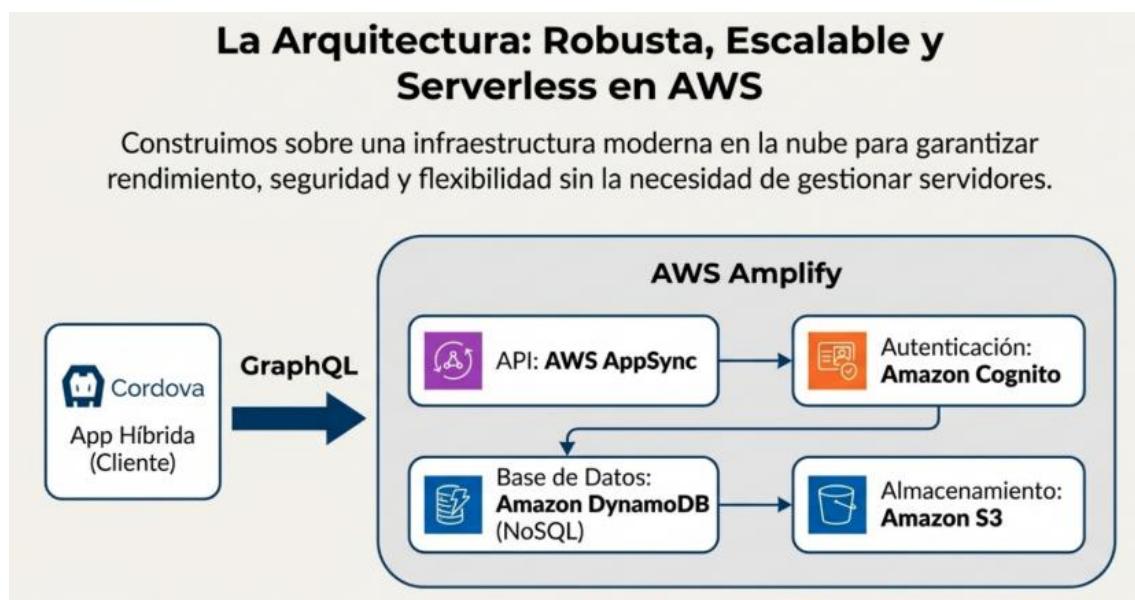
- Servicio: **Amazon Cognito**

Proporciona autenticación de usuarios (login) y gestión de sesiones de forma segura y sin servidor. Se integró automáticamente al configurar Amplify con amplify add auth.

## Notificaciones (planificadas pero todavía no integradas):

- Servicio planificado: Amazon SNS (Simple Notification Service)

**Razón:** Nos permitirá enviar recordatorios automáticos a usuarios por correo electrónico o notificaciones push en el futuro. Es escalable, económico y fácil de integrar con otras funciones de AWS.



### 3. Sistema de Control de Versiones

- **GitHub:** para coordinar el desarrollo entre el equipo.
- Visual Studio: Realizar commits directamente desde el IDE.
- Utilizar comandos Git desde la terminal para gestionar versiones, ramas y fusiones de forma más precisa y flexible.

### **Ventajas del Proyecto con AWS:**

- La infraestructura en AWS asegura **flexibilidad y escalabilidad**.
- Apache Cordova permite **reutilizar el código** web para crear aplicaciones móviles multiplataforma.
- Visual Studio proporciona un **entorno de desarrollo completo** para gestionar el proyecto.

Tras haber valorado el uso de distintos servidores y bases de datos, creemos que esta sería una buena solución.

Aun así, hemos valorado otras opciones como GPC o Render, o incluso hacer una app multiplataforma con Flutter, pero esta nos parece la mejor opción.

### **4.2. Base de Datos**

La base de datos DynamoDB se ha construido con sumo cuidado para dar solución a todos los requerimientos de nuestra app.

Primeramente, se elaboró un modelo con todas las tablas y conexiones, que fue refinándose durante la implementación para cubrir todos los requisitos.

Todos los tipos o modelos fueron creados con sus correspondientes atributos y funciones mediante el archivo **schema.graphql**, el cual se adjunta como anexo a parte dada su extensión.

Mediante el comando **amplify push** se fue creando y actualizando este modelo dentro de AWS como una base de datos DynamoDB y se crearon de forma automática las consultas, mutaciones y suscripciones.

A continuación, un breve ejemplo (*ver código completo en el anexo 4*), y más adelante desarrollaremos este apartado en profundidad.

```

type Usuario @model @auth(rules: [{ allow: private }]) {
  id: ID!
  nombre: String!
  # Tareas asignadas a este usuario (usa el índice byUsuario de Tarea)
  tareas: [Tarea] @hasMany(indexName: "byUsuario", fields: ["id"])
  # Membresías del usuario en hogares (join)
  hogares: [UsuarioHogar] @hasMany(indexName: "byUsuario", fields: ["id"])
}

type Hogar
  @model
  @auth(rules: [{ allow: private, operations: [create, read, update, delete] }])
  id: ID!
  nombre: String!
  miembros: [UsuarioHogar] @hasMany(indexName: "byHogar", fields: ["id"])
  juniors: [JuniorHogar] @hasMany(indexName: "byHogar", fields: ["id"])
  tareas: [Tarea] @hasMany(indexName: "byHogar", fields: ["id"])

  recompensas: [Recompensa] @hasMany(indexName: "byHogar", fields: ["id"])

  # Acceso directo a movimientos
  ingresos: [Ingreso] @hasMany(indexName: "byHogarByMesIngreso", fields: ["id"])
  gastos: [Gasto] @hasMany(indexName: "byHogarByMesGasto", fields: ["id"])
}

```

**GraphQL es un lenguaje de consultas** para APIs y un entorno de ejecución que permite a los clientes solicitar exactamente los datos que necesitan. Fue desarrollado por **Facebook** como una **alternativa a las arquitecturas REST**, ofreciendo una forma más eficiente, poderosa y flexible de interactuar con los datos.

## Características Clave de GraphQL

### 1. Consulta precisa de datos

A diferencia de REST, donde cada endpoint devuelve una estructura fija de datos, GraphQL permite a los clientes definir con precisión qué datos desean recibir. Esto evita la sobrecarga (**overfetching**) y la falta de datos (**underfetching**).

### 2. Consulta de múltiples recursos en una sola petición

En REST, se requieren múltiples llamadas para obtener datos de diferentes recursos relacionados. Con GraphQL, todos esos datos se pueden obtener en una sola consulta estructurada, **reduciendo el número de peticiones al servidor**.

### **3. Esquema fuertemente tipado**

GraphQL utiliza un sistema de tipos definido mediante un **esquema**. Este esquema describe todos los tipos de datos posibles en una API, incluyendo sus **relaciones**. Gracias a esto, es posible validar consultas antes de ejecutarlas y generar **documentación automática**.

### **4. Mutations para modificar datos**

Además de leer datos con "queries", GraphQL permite modificar la información del servidor a través de "mutations". Estas operaciones están definidas en el esquema y permiten acciones como crear, actualizar o eliminar registros.

### **5. Suscripciones para datos en tiempo real**

GraphQL soporta "subscriptions", que permiten a los clientes recibir datos en tiempo real cuando ocurren ciertos eventos. Esto es útil en aplicaciones que requieren actualizaciones dinámicas, como chats o dashboards.

## **Estructura básica del lenguaje**

El lenguaje de GraphQL se basa en tres tipos principales de operaciones:

### **Query**

Usada para leer datos.

### **Mutation**

Usada para modificar datos.

### **Subscription**

Usada para recibir actualizaciones en tiempo real.

## **Defición de Tipos ('type')**

En GraphQL, los tipos (`type`) se utilizan para definir la estructura de los datos que se pueden consultar. Cada tipo representa un objeto del mundo real y sus propiedades.

### Ejemplo de definición de tipos:

```
type Usuario {
```

```
    id: ID!
```

```
    nombre: String!
```

```
    email: String!
```

```
}
```

### ALMACENAMIENTO EN AWS MEDIANTE S3

Dado que también se necesitaba almacenar documentos, se creó un **espacio en la nube** dentro de AWS con S3.

Para ello se redefinió un Cognito User Pool y se configuraron sus permisos correspondientes para garantizar la seguridad y privacidad de los documentos.

### 4.3. Funcionalidades Clave

Aquí mostramos un primer prototipo muy básico a partir del cual iremos desglosando la idea punto por punto:





## USUARIOS Y HOGARES

### 1. Registro de Usuarios

Primero necesitamos un **sistema** que permita que los **usuarios se registren e inicien sesión**:

Frontend:

- Pantallas de registro e inicio de sesión, donde los usuarios ingresan su correo electrónico, contraseña y nombre.

Backend:

- Base de datos que almacene la información de los usuarios (nombre, correo, contraseña encriptada).

API para manejar:

- Registro
- Inicio de sesión

## 2. Crear un Hogar

Funcionalidad:

- Una vez registrado, el usuario puede **crear un hogar** al proporcionar un nombre (por ejemplo, “Hogar Pérez”).
- Asigna al **creador** como el **administrador del hogar por defecto**.

Base de datos:

- Tabla hogares: Guarda los hogares creados, con un ID único y el nombre.
- Relación usuario-hogar: Vincula cada hogar con su creador y otros miembros.

Frontend:

- Pantalla para crear un hogar.

## 3. Compartir el Hogar

Generación de enlace único:

- Un administrador del hogar podrá crear un **enlace de invitación** al hogar para otro usuario a partir de su email y asignarle un rol en el hogar. Puede eliminar esta invitación antes de ser aceptada. El otro usuario recibirá la invitación y podrá o no aceptarla.

Backend:

- Enlace que permitirá que los nuevos usuarios se unan al hogar utilizando las ventajas de AWS.

Frontend:

- Crear una página para generar y administrar el enlace y otra para aceptarlo o rechazarlo.

#### 4. Gestión de Permisos

Roles:

- Definir **roles** básicos como **administrador, miembro o junior**. También podrá crearse un **subperfil junior** que cuelgue de un administrador.
- El administrador puede elegir y modificar los roles de los miembros del hogar.

Backend:

- Tabla para roles de usuarios dentro del hogar.
- API para gestionar roles.

Frontend:

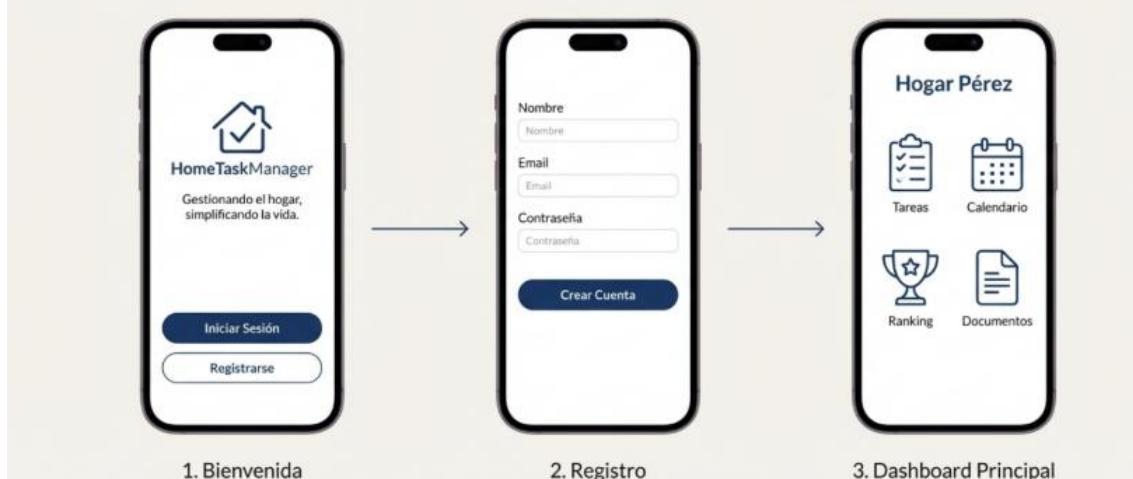
- Pantallas para el administrador donde pueda ver a todos los miembros del hogar y gestionar sus permisos.

#### 5. Flujo Completo

1. El usuario se registra o inicia sesión.
2. Crea un hogar, convirtiéndose automáticamente en administrador.
3. Genera un enlace único para invitar a otros miembros.
4. Los invitados se unen al hogar a través del enlace.
5. El administrador puede asignar o revocar permisos para cada miembro.

### Del Boceto a la Realidad: El Flujo de Bienvenida

Un **diseño limpio** y funcional que guía al usuario de forma natural desde el primer contacto con la aplicación.



## **CALENDARIO**

Los miembros del hogar tienen un **calendario compartido** donde pueden poner eventos.

En un futuro, trataremos de dar la opción de activar notificaciones de recordatorio y asociarlo además a Google calendar.

## **LISTAS**

Se pueden crear listas, como lista de la compra. Estas listas pueden ser **públicas o privadas**.

## **TAREAS**

Podrán crearse tareas completamente **personalizadas**, además de **reutilizarlas** y hacerlas **recurrentes**.

A estas tareas se les puede dar un plazo de tiempo para realizarlas o no, asignarles usuarios, etc.

Cada vez que un miembro del hogar complete una tarea como realizada, debe ser un administrador quien la dé como válida para que se le asignen los puntos.

## **PRESUPUESTO**

Ingresos y Gastos, con vista mensual y acceso a meses anteriores.

Además de contabilizar los ingresos y gastos totales, se añadirán mensajes de motivación para mejorar la gestión de la economía.

## **DOCUMENTOS**

Almacenamiento en la nube. Los usuarios podrán subir sus documentos y hacerlos públicos o privados. Tendrán opción de verlos, borrarlos y descargarlos. Una posible mejora a futuro es clasificarlos y facilitar su búsqueda.

## GAMIFICACIÓN

### Objetivo General

**Motivar** a los usuarios a colaborar, competir sanamente y mantener sus responsabilidades dentro del hogar mediante un sistema de:

- **Tareas con puntos**
- **Retos diarios**
- **Rankings semanales y mensuales**
- **Recompensas canjeables**

### 1. Tareas con Puntos

Cada **tarea** tiene una dificultad asignada que se traduce en una **cantidad de puntos**.

Al completar una tarea, el usuario recibe esa cantidad de puntos.

Los puntos **se registran y actualizan**.

Las tareas se marcan como completadas, se validan por un administrador y se almacenan con una relación a usuarioHogar.

### 2. Ranking Semanal y Mensual

La clase Clasificacion gestiona dos tipos de rankings:

- **Semanal**: reinicia cada lunes.
- **Mensual**: reinicia cada primero del mes.

El ranking se almacena como una lista de instancias ligadas a usuarioHogar.

### 3. Retos Diarios

El sistema genera automáticamente **3 retos por día para cada usuario y hogar**, por ejemplo:

- Realizar 1 tarea hoy
- Completar una tarea en menos de 3 horas
- Ser el primero en completar una tarea hoy

Los usuarios que cumplan el reto ganan **puntos extra**.

#### 4. Sistema de Recompensas

Los usuarios pueden **canjear sus puntos** por recompensas definidas en la clase Recompensa.

Ejemplos de recompensas:

- Elegir la próxima tarea
- Obtener un día libre;

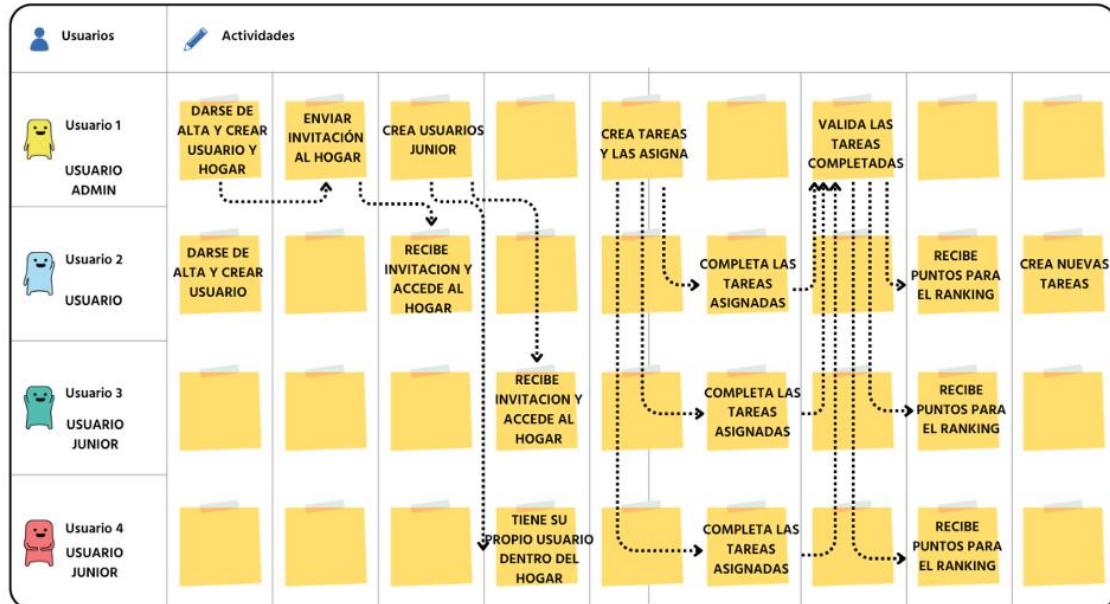
Validación y asignación:

Los administradores pueden crear **recompensas personalizadas**.

#### 5. Completar logros para subir de nivel

Los usuarios que completen logros propuestos pueden ir subiendo de nivel y obtener así una insignia especial.

Con esto se pretende fomentar el uso de la app de forma complementaria y hacer la gamificación escalable.



#### **Resumen de Flujo de Gamificación**

Usuario → realiza Tarea → gana puntos → Ranking

└ participa en Reto / cumple Logro → gana más puntos → canjea Recompensa

## Resultado Esperado

Fomentar la **participación activa**, el compromiso diario y la colaboración saludable entre miembros de un hogar a través de mecánicas de juego claramente integradas al sistema actual.

Para lograr estos objetivos, hemos tratado de abarcar varios ámbitos de la vida diaria desde un punto de vista **saludable y responsable**. Como muestra de ello estos son algunos de los retos diarios (creados en JSON) que se muestran de forma aleatoria:

⚡ Sé el primero 🤸 Triple logro 🏆 Día perfecto 🤝 Ayuda extra

📚 Tiempo de lectura ✌️ Desconexión digital 💬 Motivador

⌚ Eficiencia 🛠 Tarea extra 🤝 Equipo unido ☀️ Buen ambiente

🧠 Aprendizaje exprés 🍋 Comida saludable 🌸 Día zen

❤️ Amable del día 🎥 Comparte tu logro 💧 Hidratado

🌙 Noche tranquila 🏠 Hogar feliz 🏰 Líder del día💡 Creador

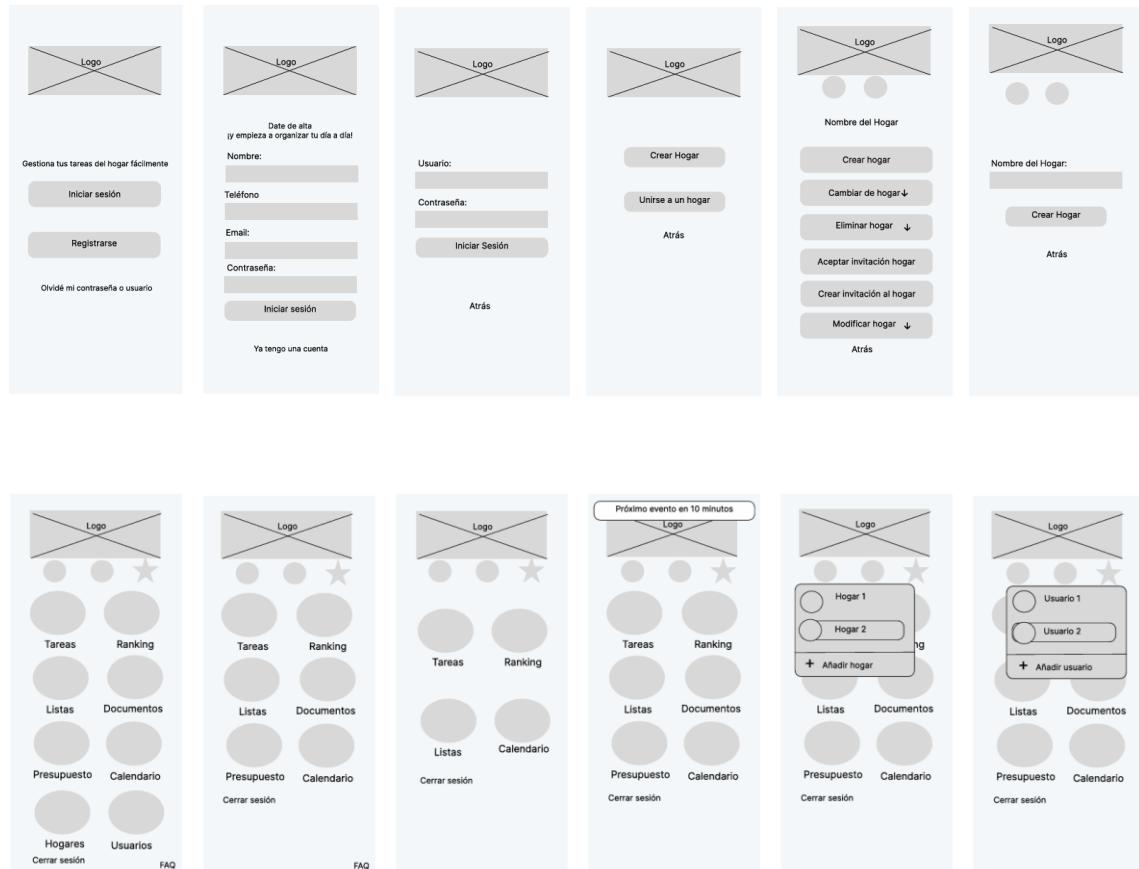
## 4.4. Prototipado y Experiencia de Usuario (UX/UI)

### DISEÑO MOCKUPS DE INTERFACES

Se realizó un **prototipo** que incluye todas las pantallas principales de la aplicación HomeTaskManager, orientada a la gestión colaborativa de tareas domésticas dentro de un hogar.

Estos son algunos de ejemplos de las pantallas creadas mediante Figma:

### WIREFRAME:



## HIGH FIDELITY:

The image displays four screenshots of the Home TaskManager mobile application, illustrating the user flow from login to the main dashboard.

- Screenshot 1: Login Screen**  
Shows the app logo "Home TaskManager" and the tagline "Gestiona tus tareas del hogar fácilmente". It features two buttons: "Iniciar sesión" and "Registrarse". Below the buttons is a link "Olvidé mi contraseña o usuario".
- Screenshot 2: Registration Screen**  
Shows the app logo and the text "Date de alta y empieza a organizar tu día a día". It has fields for "Nombre", "Teléfono", "Email", and "Contraseña", each with an associated icon. A "Iniciar sesión" button is at the bottom, and a "Ya tengo una cuenta" link is below it.
- Screenshot 3: Login with Existing Account Screen**  
Shows the app logo and fields for "Usuario:" and "Contraseña:". A "Iniciar Sesión" button is present. Below the form is a "Atrás" link.
- Screenshot 4: Home Screen (Logged In)**  
Shows the app logo and a sidebar with icons for "Crear hogar", "Cambiar de hogar", "Eliminar hogar", "Aceptar invitación hogar", "Crear invitación al hogar", and "Modificar hogar". Below the sidebar is another "Atrás" link.
- Screenshot 5: Main Navigation Menu**  
Shows the app logo and a grid of icons: Tareas, Ranking, Listas, Documentos, Presupuesto, Calendario, Hogares, and Usuarios. At the bottom are "Cerrar sesión" and "FAQ" buttons.
- Screenshot 6: Main Navigation Menu (with sidebar)**  
Shows the app logo and a sidebar with icons for "Tareas", "Ranking", "Listas", "Documentos", "Presupuesto", "Calendario", "Hogares", and "Usuarios". At the bottom are "Cerrar sesión" and "FAQ" buttons.
- Screenshot 7: Main Dashboard**  
Shows the app logo and a grid of icons: Tareas, Ranking, Listas, Documentos, Presupuesto, Calendario, Hogares, and Usuarios. A "Próximo evento en 10 minutos" notification is visible at the top right. At the bottom are "Cerrar sesión" and "FAQ" buttons.

Y el enlace al prototipo completo en Wireframe y High Fidelity:

[https://www.figma.com/design/V4TbpZDvjQCcA2FXVLsygj/Prototipado\\_proyecto-Copia?node-id=310-2&p=f&t=KkEJBy0VxoZAzyil-0](https://www.figma.com/design/V4TbpZDvjQCcA2FXVLsygj/Prototipado_proyecto-Copia?node-id=310-2&p=f&t=KkEJBy0VxoZAzyil-0)

## Propósito de cada pantalla diseñada:

- **Pantalla de inicio / bienvenida:** Presenta el nombre de la app y permite al usuario registrarse o iniciar sesión.
- **Pantalla de registro / login:** Facilita el acceso personalizado mediante correo y contraseña.
- **Pantalla principal del hogar:** Muestra las secciones clave como calendario, tareas, listas, ranking, presupuesto y documentos.
- **Pantalla de calendario:** Visualiza eventos del hogar, con opción de añadir nuevos y activar recordatorios.
- **Pantalla de tareas:** Permite crear, asignar y validar tareas, con sistema de puntos y niveles.
- **Pantalla de recompensas y ranking:** Muestra el progreso de los usuarios, retos diarios y recompensas disponibles.
- **Pantalla de configuración / perfil:** Gestiona datos del usuario, permisos y opciones del hogar.
- **Pantalla de insignias y retos:** Permite visualizar los retos que llevas conseguidos y sus insignias correspondientes por niveles.

## El Corazón de HomeTaskManger: Funcionalidades Clave

Lato Regular. Herramientas poderosas diseñadas para ser intuitivas y accesibles para todos los miembros de la familia.



## Navegación entre pantallas:

La navegación se realiza mediante un **menú** que conecta las secciones principales y un **botón inferior** que lleva al menú o sección anterior. Tras iniciar sesión, el usuario accede al hogar, desde donde puede moverse entre calendario, tareas, listas y presupuesto, el cual te permite al pulsar los distintos iconos crear eventos, tareas, listas, etc.

Los administradores tienen **acceso adicional** a la gestión de usuarios y permisos.

Una vez que el usuario ha iniciado sesión o creado una cuenta con permisos de administrador, accede al **menú principal de la aplicación**. Esta pantalla es el **centro de navegación** desde donde se pueden gestionar todas las funciones disponibles. Tareas del administrador:



- **Tareas:** permite crear, asignar y dar seguimiento a las tareas del hogar.
- **Ranking:** muestra un sistema de puntuación o clasificación de los usuarios según el cumplimiento de tareas.
- **Listas:** espacio para generar listas de compras, pendientes o actividades.
- **Documentos:** gestión de archivos compartidos relacionados con la organización del hogar.
- **Presupuesto:** control de gastos y administración financiera del hogar.
- **Calendario:** visualización de eventos, plazos de tareas y organización de fechas importantes.
- **Hogares:** opción para configurar un nuevo hogar dentro de la aplicación, cambiar de hogar.
- **Usuarios:** permite añadir nuevos miembros al hogar para que participen en la gestión de tareas , cambiar de usuario o invitar a un usuario.
- **Cerrar sesión:** opción para salir de la cuenta y volver a la pantalla inicial.

Uno de los mayores retos para facilitar la navegación ha sido la creación de un **topbar** desde el cual el usuario puede cambiar de hogar y de perfil (admin/junior) sin necesidad de cambiar de sesión. Cada pantalla responde individualmente a este cambio y muestra la información adecuada respetando los permisos del rol seleccionado.

Además de todo esto, el **logo** siempre lleva a inicio y en cada pantalla existe un enlace para volver a la pantalla anterior en función del rol seleccionado.

## Criterios de diseño utilizados:

### COLOR / ESTILO “HomeTaskManager

El estilo visual de HomeTaskManager combina una apariencia limpia, moderna y accesible con una experiencia de usuario intuitiva y cercana



#### Descripción de la paleta de colores

|                          |                             |   |
|--------------------------|-----------------------------|---|
| Fondo general            | #F5F8FA                     | Gris claro suave que aporta luminosidad y contraste con las tarjetas. |
| Tarjetas / contenedores  | #FFFFFF / #4A6584           | Fondo blanco con borde azul pizarra                                   |
| Botón                    | #E1EBF9                     | Botones visibles en las pantallas, con texto en gris oscuro.          |
| Botón activo (al pulsar) | #3D5873 / #E1EBF9           | Estado al presionar; mantiene coherencia y mejora contraste.          |
| Botón deshabilitado      | #E5E5E5 / #697586           | Gris claro con texto gris oscuro.                                     |
| Texto                    | #697586                     | Color usado en todos los textos, incluidos botones y etiquetas.       |
| Iconos y bordes          | #4A6584                     | Se usa para separar secciones o resaltar elementos sin sobrecargar.   |
| Estado de tarea          | #A7D9C9 / #F6E7A1 / #E8A1A1 | Indica acciones completadas o en curso o fuera de plazo.              |

#### Tipografía

Inter, sans-serif, con peso Bold y tamaño 20 px en títulos y botones para resaltar acciones, y Regular en textos informativos.

## 4.5. Análisis de Usabilidad (10 Heurísticas de Nielsen)

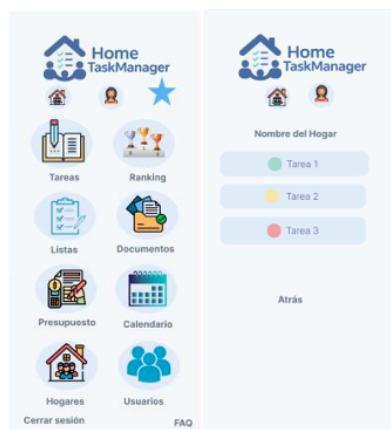
Los heurísticos de Nielsen son un conjunto de 10 **principios generales de usabilidad** propuestos por **Jakob Nielsen** (experto en usabilidad y experiencia de usuario) en 1994. Estos principios sirven como guías para evaluar interfaces de usuario y detectar problemas comunes de diseño que afectan la facilidad de uso.

A continuación, los detallamos mediante una breve explicación y de qué manera serán implementados en nuestra app. Hemos decidido mostrarlos todos ya que nos parece una parte muy importante del desarrollo en una app orientada a un público tan general.

### 1. Visibilidad del estado del sistema

**Explicación:** La aplicación debe mantener siempre **informado** al usuario sobre lo que está ocurriendo, con retroalimentación rápida y clara.

**Aplicación en la app:**



- Se han creado los botones Usuario y Hogar arriba a la izquierda que aparecen en todo momento para que el usuario sepa fácilmente en qué hogar está y con qué usuario está funcionando.
- Al gestionar tareas, aparecerá el nombre del hogar y un listado de las tareas que están en marcha para verlo de un vistazo.

### 2. Correspondencia entre el sistema y el mundo real

**Explicación:** La app debe hablar el **mismo lenguaje** que usan los usuarios en su vida cotidiana y representar conceptos de manera intuitiva.

**Aplicación en la app:**

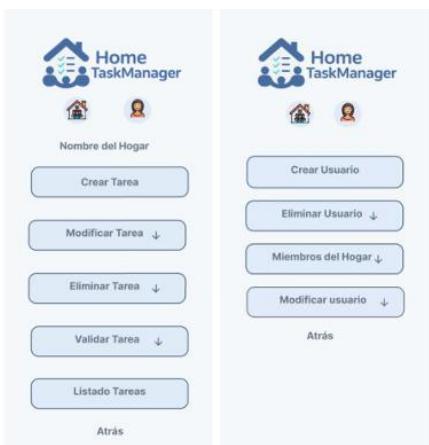


- Usar términos familiares como "sacar la basura", "hacer la cama" en lugar de expresiones técnicas.
- Incorporar iconos que representen objetos reales (persona, casa, calendario...).
- Agrupar las listas de compras o tareas según categorías prácticas (cocina, baño, supermercado).

### 3. Control y libertad del usuario

**Explicación:** Los usuarios deben sentir que tienen el **control** y poder corregir fácilmente errores o decisiones accidentales.

#### Aplicación en la app:



- Permitir deshacer al borrar una tarea ("¿Quieres deshacer esta acción?").
- Hacer sencillo reasignar una tarea a otra persona del hogar.
- Modificar los permisos de usuarios en todo momento e incluso eliminar usuarios u hogares por parte de los administradores.

### 4. Consistencia y estándares

**Explicación:** La interfaz debe ser **coherente** en todo momento y seguir los patrones comunes que los usuarios ya conocen en sus dispositivos.

#### Aplicación en la app:

- Usar los **mismos colores** para estados (verde = creado, rojo = importante, amarillo = pendiente).
- Mantener la **misma estructura** de menús y botones en todas las secciones.

## 5. Prevención de errores

**Explicación:** Es mejor evitar que ocurran errores antes que solucionarlos después.

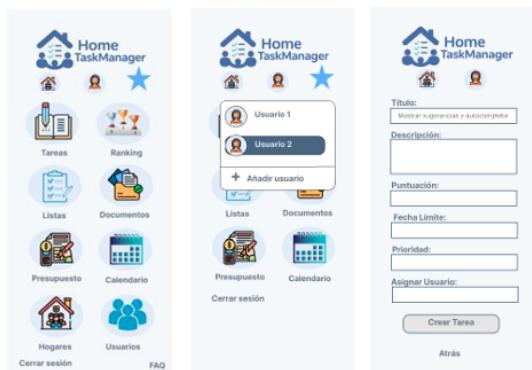
### Aplicación en la app:

- **Deshabilitar** botones hasta que los datos introducidos completos.
- **Solicitar confirmación** antes de eliminar tareas, listas, usuarios...

## 6. Reconocimiento mejor que recuerdo

**Explicación:** El usuario debe **reconocer opciones** en pantalla en lugar de depender de su memoria.

### Aplicación en la app:



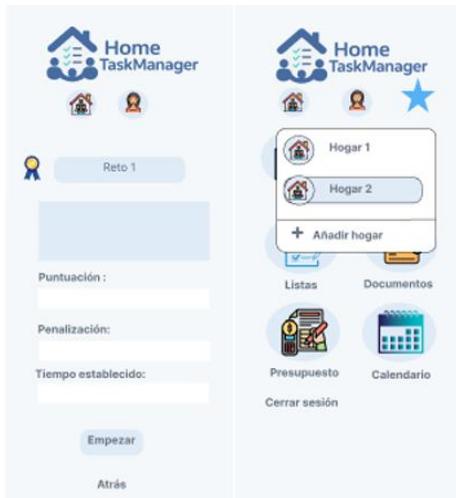
- Ofrecer menús visuales con fotos o iconos de los miembros de la familia y distintos hogares e iconos visuales de los distintos apartados del menú principal.
- Incluir historial o sugerencias de tareas usadas anteriormente.

## 7. Flexibilidad y eficiencia de uso

**Explicación:** La app debe ser útil tanto para principiantes como para usuarios avanzados, ofreciendo atajos y opciones de personalización.

### Aplicación en la app:

- En cualquier momento es posible **modificar** los datos de los distintos usuarios y hogares, así como las distintas tareas, eventos, etc.
- Permitir **filtrar** las tareas por persona, por prioridad, hogar, etc.

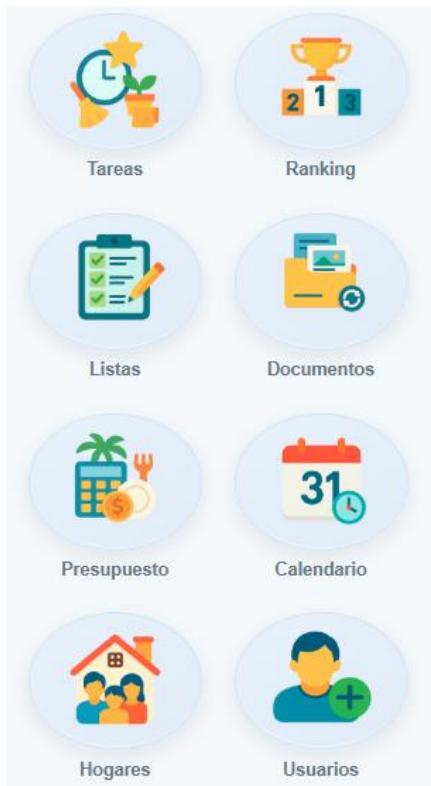


- Menús desplegables y botón para volver al inicio o a la pantalla anterior en cualquier momento.

## 8. Diseño estético y minimalista

**Explicación:** El diseño debe ser **claro**, con la información justa y sin elementos innecesarios que distraigan.

**Aplicación en la app:**



- Evitar notificaciones irrelevantes, mostrando solo lo esencial.
- Usar iconos simples y colores suaves que no saturen visualmente.

*\*Al implementar la app y revisar este principio se reevaluaron los iconos y se diseñaron otros personalizados que siguen un estilo más unificado y minimalista.*

## 9. Ayudar a reconocer, diagnosticar y recuperarse de errores

**Explicación:** Cuando ocurre un error, los mensajes deben ser claros, comprensibles y ayudar al usuario a solucionarlo.

### Aplicación en la app:

- Mensajes simples y prácticos: “No hay conexión. Reintenta cuando estés en línea.”
- Aparecerán mensajes cuando los campos estén incorrectos o incompletos.
- Esta parte se desarrollará principalmente con el código (js, try/cath, required ...) pero también mediante botones y opciones como “Atrás”, “Ya tengo una cuenta”, FAQ, modificar/eliminar, “Olvidé mi usuario o contraseña”, etc.

## 10. Ayuda y documentación

**Explicación:** Aunque lo ideal es que la app sea intuitiva, siempre debe existir un apoyo al que recurrir.

### Aplicación en la app:



- Tutorial inicial que guíe al usuario en sus primeros pasos.
- Sección de preguntas frecuentes con ejemplos prácticos.

## 5. DOCUMENTACIÓN E IMPLEMENTACIÓN

Comenzamos ahora a desarrollar la parte más técnica de la implementación de nuestra app.

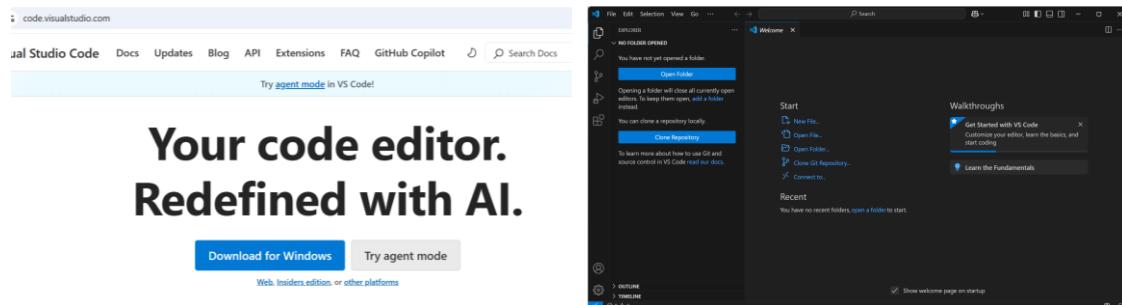
En el **ANEXO 5** se adjunta la documentación completa dada su extensión.

No obstante, a continuación, haremos un breve repaso por todos los pasos que se han dado hasta la consecución del producto final.

### 5.1. SOFTWARE BÁSICO, AWS Y CONTROL DE VERSIONES

#### VISUAL STUDIO CODE

Descargamos Visual Studio Code de **code.visualstudio.com** y ejecutamos el archivo de instalación.

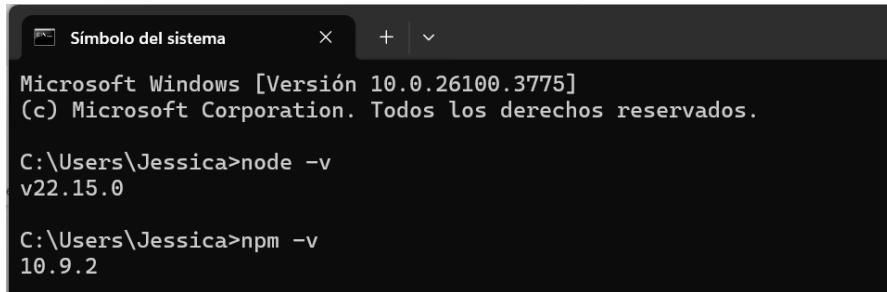


#### NODE.JS / CORDOVA

Descargamos Node.js para poder utilizar el framework Cordova de nodejs.org/es y ejecutamos el archivo de instalación:



Una vez instalado, desde la consola comprobamos que está instalado Node y su gestor de paquetes:

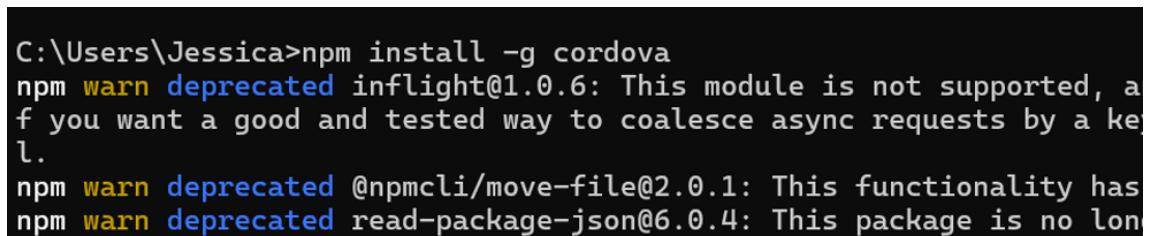


```
Símbolo del sistema
Microsoft Windows [Versión 10.0.26100.3775]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Jessica>node -v
v22.15.0

C:\Users\Jessica>npm -v
10.9.2
```

Por último, ejecutamos el comando para instalar Cordova y comprobamos que se ha instalado correctamente:



```
C:\Users\Jessica>npm install -g cordova
npm warn deprecated inflight@1.0.6: This module is not supported, a
f you want a good and tested way to coalesce async requests by a ke
l.
npm warn deprecated @npmcli/move-file@2.0.1: This functionality has
npm warn deprecated read-package-json@6.0.4: This package is no lon
```

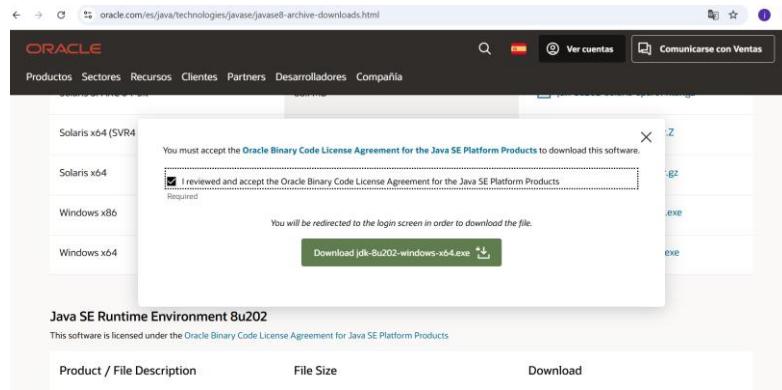
## ANDROID STUDIO

Descargamos Android Studio de developer.android.com para poder obtener previsualizaciones de nuestra app:



## JDK

Procedemos a instalar JDK descargándolo de Oracle.com y ejecutando el instalador:



\*Posteriormente lo hemos actualizado a la versión 17 para que funcione.

## AMAZON WEB SERVICES

Desde la página de [aws.amazon.com](https://aws.amazon.com/es/free/) accedemos a una cuenta gratuita con funcionalidades limitadas:



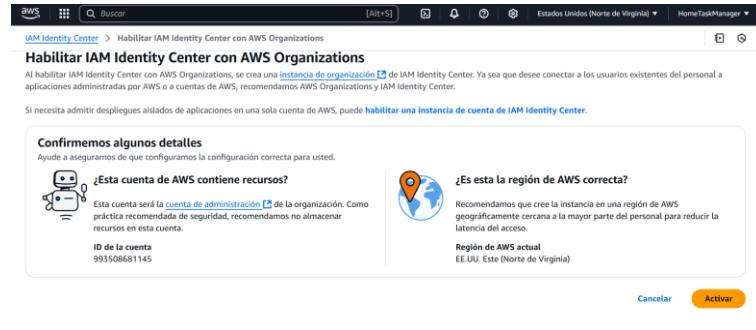
Más de 100 productos de AWS están disponibles actualmente en el nivel gratuito de AWS. Hay tres tipos diferentes de ofertas gratuitas disponibles en función del producto usado. Haga clic en el icono siguiente para explorar nuestras ofertas.

- Pruebas gratuitas**
- 12 meses de uso gratuito**
- Siempre gratis**

Estas son algunas de las características:

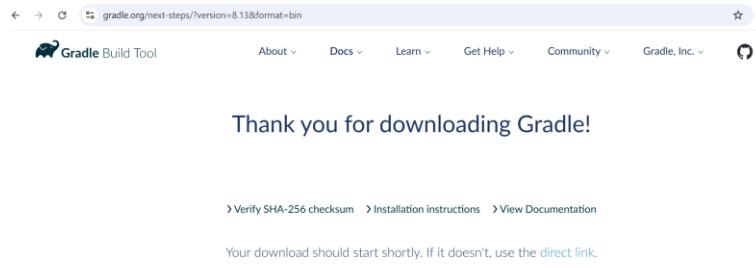
|  |                     |
|--|---------------------|
| <b>ANÁLISIS</b>  | <b>NOVEDAD</b>      |
| Nivel gratuito   | 12 MESES GRATIS     |
| <a href="#">Amazon OpenSearch Service</a>  |                     |
| <b>750 horas</b>   |                     |
| por mes de una instancia t2.small.search o t3.small.search single-AZ   |                     |
| Servicio administrado que facilita la  |                     |
|  |                     |
| <b>SOLUCIONES MÓVILES</b>  | <b>NOVEDAD</b>      |
| Nivel gratuito   | 12 MESES GRATIS     |
| <a href="#">Amazon API Gateway</a>   |                     |
| <b>1 millón</b>  |                     |
| de llamadas a la API recibidas por mes   |                     |
| Publique, mantenga, monitoree y proteja API a cualquier escala.  |                     |
| 1 millón de llamadas de API recibidas por  |                     |
|  |                     |
| <b>SEGURIDAD, IDENTIDAD Y CONFORMIDAD</b>  | <b>NOVEDAD</b>      |
| Capa gratuita  | 12 MESES GRATIS     |
| <a href="#">Amazon Cloud Directory</a>   |                     |
| <b>1 GB</b>  |                     |
| de almacenamiento al mes   |                     |
| Servicio de creación de directorios nativos en la nube y totalmente administrados para datos con varias jerarquías     |                     |
|  |                     |
| <b>ENTREGA DE CONTENIDO</b>  |                     |
| Nivel gratuito   | GRATIS PARA SIEMPRE |
| <a href="#">Amazon CloudFront</a>  |                     |
| <b>1 TB</b>  |                     |
| de transferencia saliente de datos   |                     |
| Servicio web para distribuir contenido a usuarios finales con baja latencia y una gran velocidad de transferencia.     |                     |
|  |                     |
| <b>MACHINE LEARNING</b>  | <b>NOVEDAD</b>      |
| Nivel gratuito   | 12 MESES GRATIS     |
| <a href="#">Amazon Comprehend</a>  |                     |
| <b>50 000</b>  |                     |
| unidades de texto (5 millones de caracteres) por cada API al mes   |                     |
| Procesamiento de lenguaje natural (NLP) completamente administrado y con entrenamiento continuo.                       |                     |
|  |                     |
| <b>INTERACCIÓN CON CLIENTES</b>  | <b>NOVEDAD</b>      |
| Nivel gratuito   | 12 MESES GRATIS     |
| <a href="#">Amazon Connect</a>   |                     |
| <b>90 minutos</b>  |                     |
| al mes de uso de Amazon Connect  |                     |
| Amazon Connect es un práctico centro de contacto basado en la nube que puede adaptarse a empresas de cualquier tamaño. |                     |
|  |                     |

Creamos la cuenta y accediendo a IAM configuramos los usuarios que tendrán acceso:

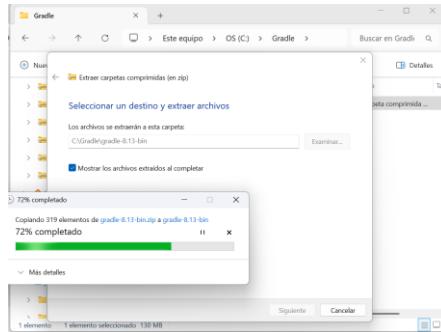


## GRADLE

Para que Android Studio funcione correctamente, vamos a instalar Gradle:



Extraemos el archivo en una carpeta dentro de C:



En variables del sistema agregamos:

- Nombre: GRADLE\_HOME
- Valor: la ruta donde la descomprimimos
- 

Agregamos al Path la ruta:

%GRADLE\_HOME%\bin

Y comprobamos desde la terminal que se ha configurado correctamente:

```

Microsoft Windows [Versión 10.0.26100.3775]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Jessica>gradle -v

Welcome to Gradle 8.13!

Here are the highlights of this release:
- Daemon JVM auto-provisioning
- Enhancements for Scala plugin and JUnit testing
- Improvements for build authors and plugin developers

For more details see https://docs.gradle.org/8.13/release-notes.html

-----
Gradle 8.13

-----
Build time: 2025-02-25 09:22:14 UTC
Revision: 073314332697ba45c16c0a0ce1891fa6794179ff

Kotlin: 2.0.21
Groovy: 3.0.22
Ant: Apache Ant(TM) version 1.10.15 compiled on August 25 2024
Launcher JVM: 1.8.0_202 (Oracle Corporation 25.202-b08)
Daemon JVM: C:\Program Files\Java\jdk1.8.0_202 (no JDK specified, using current Java home)
OS: Windows 10 10.0 amd64

C:\Users\Jessica>

```

## CONTROL DE VERSIONES

Nos situamos en el directorio en el que queremos crear el proyecto y lo creamos como un proyecto Cordova:

```

Microsoft Windows [Versión 10.0.26100.3775]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Jessica>d "OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO"
C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO>cordova create HomeTaskManager
com.tuempresa.hometaskmanager "HomeTaskManager"
Creating a new cordova project.

C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO>cd HomeTaskManager
C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>code .
C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>

```

Lo configuramos para Android:

```

C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>cordova platform
m add android
Using cordova-fetch for cordova-android
Adding android project...
Creating Cordova project for the Android platform:
  Path: platforms\android
  Package: com.tuempresa.hometaskmanager
  Name: HomeTaskManager
  Activity: MainActivity
  Android Target SDK: android-35
  Android Compile SDK: 35
  Subproject Path: CordovaLib
  Subproject Path: app
Android project created with cordova-android@14.0.0

C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>

```

Instalamos los plugins básicos:

```

C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>cordova plugin
add cordova-plugin-device
Installing "cordova-plugin-device" for android
Adding cordova-plugin-device to package.json

C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>cordova plugin
add cordova-plugin-splashscreen
Unmet project requirements for latest version of cordova-plugin-splashscreen:
  cordova-android (14.0.0 in project, >=3.6.0 <11.0.0 required)
Fetching highest version of cordova-plugin-splashscreen that this project supports: 6.0.1 (la
test is 6.0.2)
Installing "cordova-plugin-splashscreen" for android
Adding cordova-plugin-splashscreen to package.json

C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>cordova plugin
add cordova-plugin-inappbrowser
Installing "cordova-plugin-inappbrowser" for android
Adding cordova-plugin-inappbrowser to package.json

C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>cordova prepare

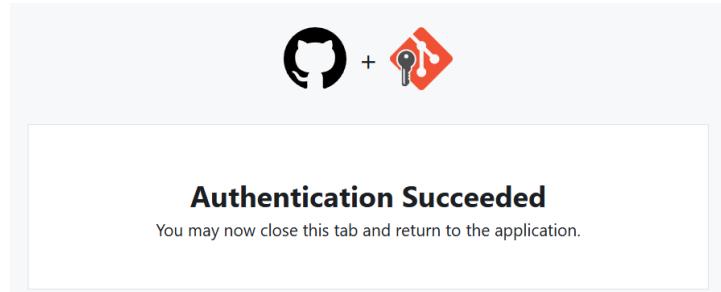
C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>

```

Y sobre este proyecto creamos el repositorio local:

```
C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>git init  
Initialized empty Git repository in C:/Users/Jessica/OneDrive/Escritorio/IFP ASIGNATURAS/PROY  
ECTO/HomeTaskManager/.git/  
C:\Users\Jessica\OneDrive\Escritorio\IFP ASIGNATURAS\PROYECTO\HomeTaskManager>
```

Añadimos a Git todo el proyecto, realizamos un primer commit para comprobar que funciona correctamente, nos autentificamos:



Y ya tenemos el repositorio listo:

A screenshot of a GitHub repository page for "HomeTaskManager". The page shows a list of files: "www", ".gitignore", "config.xml", "package-lock.json", and "package.json", all added in the first commit. The commit was made by "JessicaMarroniIP" 22 minutes ago. On the right side, there are sections for "About", "Activity" (0 stars, 1 watching, 0 forks), "Releases" (none published), "Packages" (none published), and "Languages".

Creamos nuevos usuarios y compartimos los enlaces:

A screenshot of the GitHub user invite management interface. It shows three pending invites: "Anaditm" (Collaborator), "jpcerro" (Pending Invite), and "patrimj95" (Pending invite). Each invite has a small checkbox and a trash can icon.

El siguiente comando permite a otras personas clonar el repositorio en su ordenador:

```
Bash Copiar
git clone https://github.com/JessicaMarronIFP/HomeTaskManager.git
```

Para realizar commits y push en futuras ocasiones:

1. Agrega los cambios:  

```
Bash Copiar
git add .
```
2. Realiza un commit con un mensaje explicativo:  

```
Bash Copiar
git commit -m "Descripción de los cambios"
```
3. Sube los cambios al repositorio:  

```
Bash Copiar
git push
```

## Cómo funciona el flujo de trabajo en equipo

### 1. Modificaciones locales:

- Cada colaborador trabaja en su propia copia local del repositorio, hace cambios y los guarda con un commit.
- Estos cambios son locales y no afectan a otros hasta que se hace un push al repositorio remoto.

### 2. Subir los cambios al remoto:

- Para sincronizar los cambios en GitHub, el colaborador hace un push.
- Ahora los cambios están disponibles en el repositorio remoto.

### 3. Actualizar las copias locales de los demás:

- Los demás colaboradores deben ejecutar **git pull** para descargar los cambios del remoto a sus propias computadoras.

## **\*Guía rápida para conectar al proyecto Amplify (HometaskManager)**

Esta guía permite que cualquier miembro del equipo se conecte al entorno backend de AWS Amplify para el proyecto compartido.

### **Requisitos previos:**

- Tener Node.js y Amplify CLI instalados.
- Tener una cuenta IAM con Access Key y Secret Access Key proporcionadas por el propietario del proyecto.
- Tener git instalado para clonar el repositorio.

### **PASO 1:**

Clonar el repositorio del proyecto git clone cd HometaskManager

### **PASO 2:**

Configurar credenciales de AWS

aws configure

- AWS Access Key ID: (proporcionado por el administrador)
- AWS Secret Access Key: (proporcionado por el administrador)
- Region: eu-west-1
- Output format: json

### **PASO 3:**

Vincular el proyecto con el backend de Amplify

amplify pull --appId \*\*\*\*\*er7i9jm --envName dev

### **PASO 4:**

Ya puedes trabajar con el backend (API GraphQL, autenticación, etc.) Puedes usar los archivos generados en src/graphql y src/aws-exports.js para interactuar con AWS desde tu aplicación.

## 5.2. CONFIGURACIÓN DEL BACKEND

### Pasos realizados para actualizar la API GraphQL con AWS Amplify

#### 1. Accedimos al proyecto local

Abrimos una terminal y nos ubicamos en la carpeta del proyecto llamada 'HometaskManager'.

#### 2. Ejecutamos el comando para actualizar la API

Usamos el siguiente comando para iniciar la actualización de la API GraphQL configurada con AWS Amplify:

```
amplify update api
```

#### 3. Seleccionamos el tipo de servicio

Elegimos el servicio 'GraphQL', ya que es el tipo de API que estamos utilizando para gestionar datos.

#### 4. Revisamos la información de la API actual

Se nos mostró la siguiente información:

- Nombre de la API: hometaskmanager
- Endpoint de la API: una URL de AWS AppSync que permite interactuar con la API GraphQL.

#### 5. Configuramos el tipo de autorización

- Elegimos API Key como método de autorización predeterminado.
- Añadimos una descripción para la clave de API: apikeybdd.
- Indicamos que la clave tenga una duración de 365 días antes de expirar.
- No activamos métodos adicionales de autorización (como Cognito o IAM).

## **6. Configuración de detección de conflictos**

Dejamos la opción de detección de conflictos deshabilitada, ya que no estamos usando sincronización en tiempo real con DataStore.

## **7. Compilamos el esquema GraphQL**

La herramienta compiló correctamente el esquema del backend GraphQL sin errores.

## **8. Confirmación de actualización exitosa**

Recibimos un mensaje confirmando que los cambios fueron aplicados exitosamente.

### **Compartir con el grupo:**

El comando para conectarse con este backend:

```
amplify pull --appId <tu AppId> --envName dev
```

O el aws-exports.js actualizado.

## **Configuración de autenticación**

### **1. Agregamos autenticación al proyecto**

- Se añadió un recurso de tipo Auth al proyecto usando AWS Amplify.
- Esto creó un pool de usuarios con Amazon Cognito, que se usará para gestionar el inicio de sesión y los permisos de los usuarios.

### **2. Actualizamos la API con control de acceso por usuario**

- Se actualizó el recurso de la API para aplicar restricciones de acceso a ciertos modelos.
- Los modelos Usuario, Tarea y Evento están configurados con la propiedad [owner], lo que significa que cada usuario sólo puede acceder a sus propios registros.

### 3. Advertencias de seguridad informativas

- El sistema mostró una advertencia indicando que los modelos están configurados con control de acceso basado en el propietario ([owner]), y que podríamos necesitar aplicar reglas de autorización más detalladas a nivel de campo si lo consideramos necesario.
- Se proporcionó un enlace con documentación para aplicar esas reglas: Amplify Auth Rules.

### 4. Compilación correcta del esquema GraphQL

- El esquema se compiló correctamente después de estos cambios.

| Category | Resource name           | Operation | Provider plugin   |
|----------|-------------------------|-----------|-------------------|
| Auth     | hometaskmanagerf423293d | Create    | awscloudformation |
| Api      | hometaskmanager         | Update    | awscloudformation |

```
/ Are you sure you want to continue? (Y/n) · yes
Cognito UserPool configuration
Use a Cognito user pool configured as a part of this project.
⚠ WARNING: owners may reassign ownership for the following model(s) and role(s): Usuario: [owner], Tarea: [owner], Evento: [owner]. If this is not intentional, you may want to apply field-level authorization rules to these fields. To read more: https://docs.amplify.aws/cli/graphql/authorization-rules/#per-user--owner-based-data-access.
☒ GraphQL schema compiled successfully.

Edit your schema at C:\Proyecto\HometaskManager\amplify\backend\api\hometaskmanager\schema.graphql or place .graphql files in a directory at C:\Proyecto\HometaskManager\amplify\backend\api\hometaskmanager\schema
⚠ WARNING: owners may reassign ownership for the following model(s) and role(s): Usuario: [owner], Tarea: [owner], Evento: [owner]. If this is not intentional, you may want to apply field-level authorization rules to these fields. To read more: https://docs.amplify.aws/cli/graphql/authorization-rules/#per-user--owner-based-data-access.
☒ GraphQL schema compiled successfully.

Edit your schema at C:\Proyecto\HometaskManager\amplify\backend\api\hometaskmanager\schema.graphql or place .graphql files in a directory at C:\Proyecto\HometaskManager\amplify\backend\api\hometaskmanager\schema
| Building resource api/hometaskmanager⚠ WARNING: owners may reassign ownership for the following model(s) and role(s): Usuario: [owner], Tarea: [owner], Evento: [owner]. If this is not intentional, you may want to apply field-level authorization rules to these fields. To read more: https://docs.amplify.aws/cli/graphql/authorization-rules/#per-user--owner-based-data-access.
/ Building resource api/hometaskmanager☒ GraphQL schema compiled successfully.
```

## Publicación del backend en la nube con AWS Amplify

### 1. Sincronizamos los cambios con la nube

- Ejecutamos el comando amplify push para aplicar en AWS todos los cambios realizados localmente.
- Este paso es esencial para que las actualizaciones del API y del sistema de autenticación se reflejen en el entorno dev de la nube.

### 2. Compilación y verificación final

- El sistema compiló nuevamente el esquema GraphQL sin errores.

- Se verificó que los recursos a aplicar eran:
- **Auth:** creación del sistema de autenticación (hometaskmanagerf423293d).
- **API:** actualización del API GraphQL (hometaskmanager).

### 3. Confirmación de implementación exitosa

- Se confirmó que el backend fue actualizado correctamente en la nube, quedando listo para ser usado desde el frontend de la aplicación.

```
C:\Proyecto\HometaskManager>amplify push
- Fetching updates to backend environment: dev from the cloud. ✅ GraphQL schema compiled successfully
Edit your schema at C:\Proyecto\HometaskManager\amplify\backend\api\hometaskmanager\schema.graphql
es in a directory at C:\Proyecto\HometaskManager\amplify\backend\api\hometaskmanager\schema
✓ Successfully pulled backend environment dev from the cloud.
/ Building resource api/hometaskmanager ✅ GraphQL schema compiled successfully.

Edit your schema at C:\Proyecto\HometaskManager\amplify\backend\api\hometaskmanager\schema.graphql
es in a directory at C:\Proyecto\HometaskManager\amplify\backend\api\hometaskmanager\schema

Current Environment: dev



| Category | Resource name           | Operation | Provider plugin   |
|----------|-------------------------|-----------|-------------------|
| Auth     | hometaskmanagerf423293d | Create    | awscloudformation |
| Api      | hometaskmanager         | Update    | awscloudformation |


? Are you sure you want to continue? (y/n) -> yes
```

### Sincronización desde la nube con amplify pull

#### 1. Descargamos la configuración del backend desde AWS

- Ejecutamos el comando amplify pull para traer a nuestro entorno local toda la configuración del backend almacenada en la nube.
- Esto garantiza que el proyecto local esté completamente sincronizado con el entorno dev.

#### 2. Verificación del estado previo a la sincronización

- Se nos mostró el estado actual del entorno remoto, indicando que:
  - El recurso de autenticación hometaskmanagerf423293d fue creado.
  - El recurso API hometaskmanager fue actualizado.
- Ambos están gestionados por AWS CloudFormation.

```
C:\Proyecto\HometaskManager>amplify pull
```

```
Pre-pull status:
```

```
  Current Environment: dev
```

| Category | Resource name           | Operation | Provider plugin   |
|----------|-------------------------|-----------|-------------------|
| Auth     | hometaskmanagerf423293d | Create    | awscloudformation |
| Api      | hometaskmanager         | Update    | awscloudformation |

*POR MODIFICACIONES SUSTANCIALES EN EL PROYECTO Y EN EL TIPO DE BBDD*

*(3 NIVELES) SE REHIZO EL BACKEND Y SE CREÓ UNA NUEVA API*

*(los datos finales se encuentran en el documento anexo por protección de datos sensibles)*

## Generación automática de operaciones GraphQL

Después de desplegar correctamente la API, utilizamos la funcionalidad de AWS Amplify para **generar automáticamente las operaciones GraphQL** (consultas, mutaciones y suscripciones) a partir del esquema definido.

### Proceso realizado:

1. Se eligió el lenguaje de generación: **JavaScript**.
2. Se definió la carpeta de salida: `www/graphql/**/*.js`.
3. Se habilitó la opción de generar/actualizar todas las operaciones automáticamente.
4. Se estableció una profundidad de consulta de **3 niveles** para soportar estructuras de datos anidadas.

Amplify generó exitosamente los archivos necesarios y los guardó en la carpeta especificada, lo que permite al frontend interactuar fácilmente con la API sin escribir las operaciones manualmente.

Aquí tenemos unas capturas desde AWS funcionando:

Página de inicio:

The screenshot shows the AWS Home page. On the left, under 'Visitados recientemente', there are links to AWS AppSync, IAM, DynamoDB, AWS Amplify, EC2, Aurora and RDS, and Administración de facturación y costos. On the right, under 'Aplicaciones (0)', it says 'Región: Europe (Ireland)' and 'eu-west-1 (Región actual)'. There is a search bar for 'Buscar aplicaciones' and a button to 'Crear aplicación'.

AppSync con la API creada y en funcionamiento:

The screenshot shows the AWS AppSync API creation interface. It has sections for 'AWS AppSync' (with a link to 'API'), 'Documentación', and 'API (1)'. The 'API (1)' section lists one API named 'hometask...'. The interface includes buttons for 'Crear la API de GraphQL', 'Crear la API de eventos', and 'Crear la API'.

Esquema creado a partir de schema.graphql:

The screenshot shows the AWS AppSync schema editor. The left sidebar shows 'AWS AppSync' and 'API: hometaskmanager-dev'. Under 'Esquema', there are sections for 'Orígenes de datos', 'Funciones', 'Consultas', 'Almacenamiento en caché', 'Ajustes', 'Monitoreando', and 'Nombres de dominio personalizados'. The main area shows the schema code:

```
1  input CreateEventoInput {
2     id: ID!
3     fecha: AWSDate!
4     titulo: String!
5 }
6
7  input CreateTareaInput {
8     id: ID!
9     titulo: String!
10    descripcion: String!
11    eventoId: ID!
12    usuaarioId: ID!
13 }
14
15  input CreateUsuarioInput {
16     id: ID!
17     nombre: String!
18 }
19
20  input DeleteEventoInput {
```

To the right, there is a 'Solucionadores' section with a note about how it connects fields to data sources and a 'Filtrar los tipos...' search bar. Below that is an 'Evento (6)' section with a list of event types.

## Usuarios con permisos de administrador:

|                          | Nombre de usuario | Ruta | Grupo: | Última actividad | MFA | Antigüedad de |
|--------------------------|-------------------|------|--------|------------------|-----|---------------|
| <input type="checkbox"/> | amplify-user      | /    | 0      | Hace 5 días      | -   | 7 días        |
| <input type="checkbox"/> | AnaTorre          | /    | 1      | Hace 6 días      | -   | 7 días        |
| <input type="checkbox"/> | JessicaMarron     | /    | 1      | Hace 4 horas     | -   | 8 días        |
| <input type="checkbox"/> | PatriciaMorato    | /    | 1      | Hace 6 días      | -   | 7 días        |

## Entornos de backend:

| Entorno      | Opciones                  |
|--------------|---------------------------|
| dev >        | Configurar Amplify Studio |
| pruebasapp > | Configurar Amplify Studio |

## Tablas de la base de datos DynamoDB:

|                          | Nombre                                      | Estado | Clave de partición | Clave de ordenación | Índices |
|--------------------------|---|--------|--------------------|---------------------|---------|
| <input type="checkbox"/> | Evento-o2rcod2s7jcmhbhf5qhvp6vvxi-dev       | Activo | id (S)             | -                   | 1       |
| <input type="checkbox"/> | Evento-s22h6cqbfqwfjurelxqhkkuhi-pruebasapp | Activo | id (S)             | -                   | 0       |
| <input type="checkbox"/> | Evento-zigaoj4mrhmvkejvj3yz7rvrm-dev        | Activo | id (S)             | -                   | 0       |
| <input type="checkbox"/> | Tarea-o2rcod2s7jcmhbhf5qhvp6vvxi-dev        | Activo | id (S)             | -                   | 0       |
| <input type="checkbox"/> | Tarea-s22h6cqbfqwfjurelxqhkkuhi-pruebasapp  | Activo | id (S)             | -                   | 0       |
| <input type="checkbox"/> | Tarea-zigaoj4mrhmvkejvj3yz7rvrm-dev         | Activo | id (S)             | -                   | 2       |
| <input type="checkbox"/> | Usuario-o2rcod2s7jcmhbhf5qhvp6vvxi-dev      | Activo | id (S)             | -                   | 0       |

### **5.3. DESARROLLO DE CÓDIGO**

A continuación, exponemos un **resumen**. En el **ANEXO 5** se encuentra el desarrollo completo.

#### **1. Arquitectura y entorno**

- Configuramos el SDK de Android, el PATH y las herramientas necesarias (Platform-Tools, Build-Tools, Command-line Tools).
- Verificamos binarios, levantamos el emulador y logramos ejecutar cordova run android con éxito.
- Revisamos Logcat, corregimos archivos faltantes, regeneramos artefactos con *cordova clean/prepare*.
- Integramos AWS Amplify, solucionamos la ausencia de aws-exports.js y completamos el flujo de amplify pull.

#### **2. Backend y autenticación**

- Configuramos Cognito (User Pool y App Client) e integramos Amplify con la app Cordova.
- Creamos un sistema completo de registro, login, persistencia de sesión y recuperación de contraseña.
- Unificamos la configuración, reestructuramos topbar.js y sincronizamos selectores HTML/JS.
- Simplificamos la navegación, mejoramos la experiencia visual y ajustamos CSP y mensajes de error.

#### **3. Módulo Tareas I**

- Creamos las pantallas de creación, listado, edición y eliminación de tareas.
- Cargamos dinámicamente miembros del hogar, evitando duplicados.

- Implementamos filtros, paginado y sincronización con cambios de hogar.
- Integramos GraphQL (get, update, delete) y añadimos estados visuales.

#### **4. Módulo Tareas II**

- Rediseñamos *mistareas.html* con organización por hogar, estado, prioridad y color.
- Añadimos iconos de urgencia, roles (admin/usuario/junior), restricciones y redirecciones automáticas.
- Mejoramos el topbar y preparamos funciones futuras de control de roles.
- Refinamos la estética y accesibilidad.

#### **5. Módulo Presupuestos**

- Diseñamos pantallas de ingresos/gastos y definimos los modelos GraphQL asociados a Hogar.
- Implementamos consultas mensuales, totales y mensajes motivacionales dependiendo del balance.
- Solucionamos duplicaciones y sincronizamos todo con el hogar activo.

#### **6. Módulo Retos**

- Creamos el catálogo base y las páginas de retos diarios.
- Definimos modelos Reto, RetoDelDia y RetoCompletado y actualizamos el backend.
- Asignamos retos diarios determinísticos por hogar/usuario y gestionamos su completado.
- Añadimos mutex, debounce y poda automática de retos antiguos.

## **7. Módulo Ranking**

- Implementamos ranking semanal y mensual con gráficos, barras y estilos oro/plata/bronce.
- Añadimos avatares, medallas, corona y tarjeta de "Tu puntuación".
- Conectamos el ranking a AppSync y mejoramos la visualización responsive.

## **8. Módulo Recompensas**

- Definimos el modelo Recompensa (permanente/un\_uso, expiración, puntos, etc.).
- Creamos pantallas de creación, listado, modificación y eliminación.
- Implementamos la lógica de canje según puntos del mes y mantuvimos sincronía mensual.
- Añadimos panel de puntos ganados, gastados y disponibles.

## **9. Módulo Calendario**

- Diseñamos modelos de EventoPublico y EventoPrivado con reglas @auth correctas.
- Implementamos calendario mensual, creación de eventos, visibilidad por rol y detalle de evento.
- Añadimos soporte para allDay, colores, conversión ISO y depuración de GraphQL.

## **10. Módulo Insignias**

- Conectamos la página insignias con la base de datos y creamos insignias dinámicas.

- Implementamos BadgeProgress, cálculo automático de logros y niveles secuenciales.
- Integración del topbar para mostrar la medalla correspondiente.
- Añadimos tracking de visitas en cada página.

## 11. Módulo Listas

- Implementamos listas públicas y privadas con control por perfil.
- Creamos páginas de creación, listado, visualización y CRUD de ítems.
- Añadimos bloqueo para junior, ownerId correcto, sincronización con hogar y seguridad completa.

## 12. Módulo Documentos

- Revisamos S3 (CORS, identity pool, roles IAM).
- Integramos S3 en Amplify y confirmamos credenciales correctas.
- Implementamos subida y descarga real de archivos con Storage.put.

## 13. Instalación en dispositivos reales

- Generamos la app con Cordova, activamos depuración USB y usamos adb install.
- Probamos la app en dispositivos Samsung con resultados exitosos.

## 14. Revisión final

- Iniciamos refactorización, añadimos comentarios y probamos todo en navegador y dispositivos.

## 6. PRUEBAS, DESPLIEGUE Y LANZAMIENTO

### Asegurando la Calidad: Nuestro Proceso de Testeo

Verificamos rigurosamente que cada funcionalidad cumple sus objetivos y ofrece una experiencia sin fallos a través de bancos de pruebas específicos.

|  |  |
|--|--|
| <b>Login y Autenticación</b><br>Acceso seguro, registro y recuperación de cuenta 100% funcionales.<br><br><br>Estado: Satisfactorio.  | <b>Base de Datos y Gestión de Tareas</b><br>Operaciones CRUD (Crear, Leer, Actualizar, Borrar) de tareas validadas.<br><br><br>Estado: Satisfactorio.       |
| <b>Gamificación y Recompensas</b><br>Asignación de puntos y ranking correctos. Logros funcionales en web, pendientes en móvil.<br><br><br>Estado: Parcialmente Satisfactorio. | <b>Almacenamiento de Documentos</b><br>Subida y visualización funcionales. Descarga local pendiente de implementación.<br><br><br>Estado: No Satisfactorio. |

### 6.1. Técnicas de Testeo

En esta fase hemos seleccionado dos técnicas:

#### Matriz de Feedback:

La Matriz de Feedback es una técnica que permite recopilar de forma estructurada las opiniones de los usuarios tras probar el prototipo.

A través de los cuadrantes “Qué funciona”, “Qué se puede mejorar”, “Qué preguntas nos hacen” y “Qué ideas nos proponen”, se analizan fortalezas, debilidades y sugerencias reales.

En el proyecto, esta técnica se aplicará tras las pruebas de la app para identificar mejoras en usabilidad, navegación y funcionalidad antes del despliegue final.

|   |   |
|---|---|
|  ¿Qué funciona?  |  ¿Qué se puede mejorar?  |
| <ul style="list-style-type: none"> <li>• Inicio de sesión de usuarios</li> <li>• Creación de tareas</li> <li>• Menú de navegación(Entre distintas funcionalidades)</li> <li>• El sistema de puntos motiva a los usuarios a completar tareas.</li> </ul> | <ul style="list-style-type: none"> <li>• Iconos personalizados en cada tarea</li> <li>• Opción para eliminar varias tareas a la vez</li> <li>• Que el usuario pueda elegir las notificaciones que recibe</li> <li>• Que la opción de presupuestos tenga más opciones para organizar y establecer metas</li> <li>• Opción de pago con actualizaciones para ampliar las funcionalidades.</li> </ul> |

### **Evaluación en Contexto:**

La Evaluación en Contexto permite analizar cómo los usuarios interactúan con la aplicación en su entorno real, identificando obstáculos y observaciones sobre la experiencia de uso. En el proyecto, esta técnica se empleará para evaluar la app HomeTaskManager durante pruebas domésticas, observando comportamientos y recogiendo conclusiones para optimizar la interfaz y la usabilidad.

|  Usuario |  Fecha y hora |  Contexto o entorno de prueba |  Observaciones y conclusiones |
|---|--|--|--|
| Usuario 1 : Administrador   | 20/10/25<br>20:00h   | Prueba en casa,<br>desde móvil Android.  | Logra crear el hogar sin dificultad. El calendario es muy útil, le ha gustado.                                   |
| Usuario 2 : Miembro   | 21/10/25<br>15:30h   | Ordenador  | Comprende fácilmente el sistema de tareas. Le gustaría recibir un sonido al completar una tarea.                 |
| Usuario 3 : Junior  | 25/10/25<br>18:00h   | Tablet infantil con<br>control parental.   | Completa las tareas asignadas, pero no entiende bien el sistema de puntos.                                       |
| Usuario 4 : Administrador   | 29/10/25<br>12:10h   | Prueba individual en su hogar.   | Considera la app intuitiva   |
| Usuario 5 : Miembro   | 07/11/25<br>09:06h   | Móvil  | Piensa que puede añadir más opciones en la parte de presupuestos para organizarte mejor.                         |
| Usuario 6 : Junior  | 08/11/25<br>19:38h   | Tablet infantil.   | Le anima a hacer tareas, sugiere añadir algún minijuego.   |
| Usuario 7 : Miembro   | 10/11/25<br>11:02h   | Tablet   | Usa la app sin dificultad  |
| Usuario 8 : Miembro   | 12/11/25<br>13:20h   | Ordenador  | Valora la facilidad para asignar tareas  |

## **6.2. Bancos y Casos de Prueba**

En esta etapa se han definido **bancos de pruebas** que permitirán verificar el cumplimiento de los principales **OKR** establecidos en el proyecto HomeTaskManager.

Cada banco de pruebas reúne los tests necesarios para comprobar el rendimiento, la funcionalidad y la experiencia de usuario en las áreas clave del sistema.

Dada la extensión de los casos de prueba, se muestran a continuación los bancos de prueba y se adjuntan capturas del resultado, dejando los casos completos en el ANEXO 6.

### **Banco de Pruebas 1: Login**

*OKR – Mejorar el sistema de acceso y autenticación de usuarios.*

KR1: Contraseñas seguras.

KR2: Autenticación multifactor.

KR3: Reducir los errores de inicio de sesión.

KR4: Recuperación de cuenta funcional.

Se ha implementado un sistema de autenticación mediante Cognito dentro de AWS. Al darse de alta, el usuario recibe un mail con un código que deberá introducir en la app. El navegador mantendrá la sesión abierta durante el tiempo establecido desde el servidor para facilitar el acceso, y el topbar de la app permitirá cambiar de usuario y hogar fácilmente para una mejor navegación y reducción de complicaciones a la hora de cambiar de cuenta. Además si el usuario olvida su contraseña puede recuperarla fácilmente mediante un mail.

#### **Propósito:**

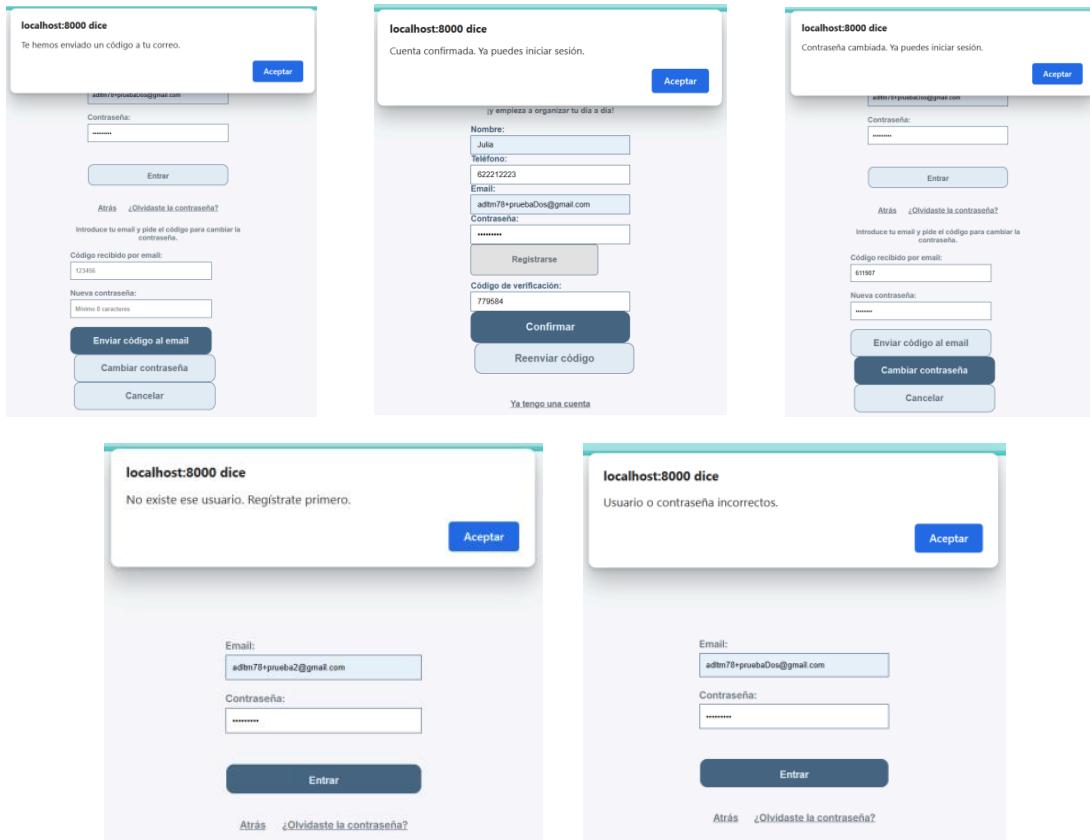
Garantizar un acceso seguro, evitando vulnerabilidades y errores de inicio de sesión.

| Test | Descripción             | Datos de entrada                        | Resultado                                      |
|------|-------------------------|---|--|
| 1    | Registrar nuevo usuario | Formulario de nuevo registro de usuario | “Cuenta confirmada. Ya puedes iniciar sesión”. |
| 2    | Contraseña incorrecta   | Correo válido con clave errónea         | Mensaje: “Usuario o contraseña incorrecta”.    |
| 3    | Inicio correcto         | Usuario y contraseña válidos            | Acceso permitido al panel principal.           |
| 4    | Usuario no registrado   | Correo no válido                        | “No existe ese usuario, regístrate primero”.   |
| 5    | Recuperar contraseña    | Solicitud de recuperación               | Usuario recibe enlace y restablece acceso.     |

**Tipo de prueba:** Funcional y de seguridad.

**Momento:** Durante la integración del módulo de login y cada vez que se realice una actualización de la app.

**El resultado de esta prueba fue satisfactorio:**



## Banco de Pruebas 2: Base de Datos y Gestión de Tareas

OKR – Crear una base de datos eficiente y estructurada.

KR1: Diseñar y documentar el esquema de la base de datos con al menos 5 tablas principales.

KR2: Implementar la base de datos y validar su funcionamiento.

KR3: Establecer relaciones y asegurar integridad de datos.

KR4: Garantizar la seguridad de los datos aplicando encriptación y controles de acceso adecuados.

Se ha creado una base de datos DynamoDB con varias tablas principales que relacionan usuarios, hogares, ranking, tareas...

En el archivo schema.graphql se han generado las consultas mediante amplify push y se tratará de implementar un sistema para reutilizarlas, de forma que el código se reduzca y que los cambios que se generen en el esquema se modifiquen de forma automática y no se rompa el funcionamiento de la app. Esta fase está en proceso de investigación.

En cuanto a los archivos y documentos, se ha creado un bucket de almacenamiento con S3 en AWS con acceso mediante Cognito UserPool para garantizar la seguridad. Esto será analizado en el siguiente caso de prueba.

### Propósito:

Comprobar que las tareas se guardan, actualizan y eliminan correctamente en la base de datos (DynamoDB).

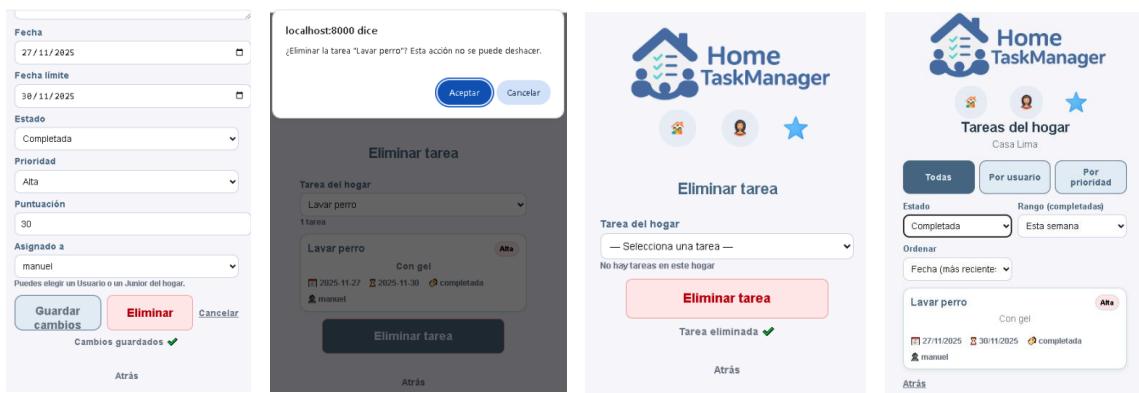
| Test | Descripción  | Datos de entrada            | Resultado                            |
|------|--------------|-----------------------------|--------------------------------------|
| 1    | Crear tarea  | “Lavar platos”, fecha 29/10 | Tarea registrada correctamente.      |
| 2    | Editar tarea | Cambiar nombre y fecha      | Actualización reflejada al instante. |

|   |                  |                               |  |
|---|------------------|-------------------------------|--|
| 3 | Eliminar tarea   | Eliminar tarea “Lavar platos” | Se elimina de la base de datos y la interfaz aunque se buscarán alternativas (por ej. Marcar una tarea como eliminada sin borrarla de la base de datos). |
| 4 | Consultar tareas | Recuperar tareas desde        | Se muestran todas las tareas del usuario activo.   |

**Tipo de prueba:** Funcional e integración.

**Momento:** Se ha de comprobar el funcionamiento de la bbdd después de cada cambio que haya en la app, durante cualquier cambio de actualización o cambio de funcionamiento.

**El resultado de esta prueba fue satisfactorio:**



### Banco de Pruebas 3: Gamificación y Recompensas

OKR – Implementar estrategias de gamificación para aumentar el compromiso y la participación de los usuarios.

KR1: Diseñar e integrar al menos tres mecánicas de juego (puntos, niveles o recompensas).

KR2: Incrementar el tiempo promedio de interacción de los usuarios en la aplicación tras la implementación.

KR3: Aumentar el número de usuarios recurrentes dentro de los primeros tres meses después de aplicar las estrategias de gamificación.

KR4: Realizar un seguimiento semanal de las gamificaciones.

Ya se han creado las mecánicas básicas, se generan puntos por tareas, hay logros diarios, insignias por niveles y recompensas que se pueden canjear y crear. Esta es la forma para que los usuarios interactúen y estén motivados.

El administrador tiene el poder de crear las tareas, elegir los puntos, modificar la prioridad, asignar a un usuario, validar las tareas.

### Propósito:

Evaluar que las mecánicas de juego estén correctamente integradas y que la experiencia de usuario incentive la participación continua dentro de la aplicación.

| Test | Descripción                                   | Datos de entrada  | Resultado esperado   |
|------|---|---|--|
| 1    | Asignación de puntos por tarea completada     | Completar tarea “Recoger habitación”                                  | El usuario recibe la cantidad de puntos asignada tras ser validado por un administrador. |
| 2    | Actualización del ranking                     | Completar varias tareas durante la semana                             | El ranking semanal se actualiza mostrando la nueva posición del usuario.                 |
| 3    | Obtención de insignias por logros completados | Usuario alcanza un nuevo nivel e insignia completando logros          | Una nueva insignia o medalla aparece en el topbar  |
| 4    | Canjear puntos por recompensas                | Canjear puntos obtenidos por recompensas creadas por un administrador | El usuario solicita la recompensa y un administrador lo valida.                          |

**Tipo de prueba:** Funcional, integración, usabilidad y seguridad (anti-abuso)

**Momento:** Durante los sprints de desarrollo al completar cada módulo para comprobar que la gamificación tiene sentido y que está relacionado con la bbdd y entre los distintos módulos.

El resultado de esta prueba fue satisfactorio:

## Banco de Pruebas 4: Sistema de Almacenamiento de Documentos

OKR – Implementar un sistema de almacenamiento de documentos.

KR1: Implementar una opción para digitalizar los documentos físicos.

KR2: Implementar un sistema de clasificación que permita reducir el tiempo de búsqueda de documentos.

KR3: Asegurar que los documentos subidos sean indexados correctamente y se puedan buscar de forma rápida.

KR4: Recopilar y aplicar retroalimentación de los primeros usuarios para mejorar la experiencia de uso.

Por el momento los archivos son visibles para admins y miembros (no juniors), y se pretende generar una primera clasificación basada en archivos públicos y privados, y una posible mejora para crear carpetas que permitan clasificarlos de forma personalizada y filtrarlos para mejorar la búsqueda.

**Propósito:**

Verificar la correcta implementación de las funcionalidades de almacenamiento, búsqueda y organización de documentos dentro de la aplicación, asegurando rapidez, accesibilidad y una buena experiencia de usuario.

| Test | Descripción              | Datos de entrada                               | Resultado esperado  |
|------|--------------------------|--|---|
| 1    | Subir archivos a la nube | Subir un archivo desde el móvil.               | El sistema guarda el archivo en la nube correctamente y el usuario recibe una notificación. |
| 2    | Accesibilidad al archivo | Botón para visualizar un documento almacenado. | El usuario puede ver el archivo.  |
| 3    | Descargar archivo        | Botón para descargar un documento almacenado.  | El usuario puede descargar un documento almacenado en la nube.                              |
| 4    | Eliminar archivo         | Botón para eliminar un documento almacenado.   | El usuario puede eliminar un documento subido previamente.                                  |

**Tipo de prueba:** Funcional, de integración y usabilidad.

**Momento:** Durante la fase de integración del sistema de almacenamiento y gestión documental y en cualquier momento en el que se modifique el tipo de autenticación y los permisos de los distintos tipos de usuario.

**Durante la ejecución de la prueba el resultado fue satisfactorio, salvo por la opción de descarga que todavía no estaba implementada:**

The image displays three screenshots of a web-based application named "Home TaskManager".

**Screenshot 1 (Left):** Shows the main document management screen. It features a header with the app logo and title. Below the header is a section titled "¡Gestiona tus Documentos!" with the sub-instruction "Sube, consulta o elimina tus archivos del hogar". A "Subir documento" button is present. A file preview card for "Captura de pantalla 2025-04-13 112732.png" is shown, along with "Ver", "Descargar", and "Borrar" buttons. At the bottom is a "Atrás" link.

**Screenshot 2 (Middle):** Shows a confirmation dialog box titled "localhost:8000 dice" with the question "¿Eliminar este documento?". It contains "Aceptar" and "Cancelar" buttons. Below the dialog is a preview card for the same file as in Screenshot 1, with "Ver", "Descargar", and "Borrar" buttons. At the bottom is a "Atrás" link.

**Screenshot 3 (Right):** Shows the result of the deletion. The "Borrar" button is highlighted in blue. The "Subir documento" button is now prominent. At the bottom is a "Atrás" link.

### **6.3. Planificación del Lanzamiento y Estrategia de Ingresos**

En esta etapa se organizan las acciones, recursos y tiempos necesarios para llevar el producto al público de forma ordenada, asegurando un lanzamiento funcional, atractivo y bien recibido por los usuarios.

#### **PLANIFICACIÓN DEL LANZAMIENTO:**

La estrategia de lanzamiento contempla tres etapas progresivas:

##### **1.Fase Beta (Pruebas internas):**

- Lanzamiento de una versión funcional entre los miembros del grupo y entorno cercano.
- Detección y corrección de errores en usabilidad, carga de datos y rendimiento.
- Feedback directo de usuarios domésticos reales.

##### **2.Fase Piloto ( Difusión limitada) :**

- Publicación en Play Store mediante canal interno de pruebas de google.
- Recopilación de métricas iniciales (descargas, uso medio, errores reportados)
- Ajustes de rendimiento, textos y diseño visual.

##### **3.Lanzamiento Público :**

- Difusión en redes sociales (Instagram, TikTok y Facebook) enfocada a familias y jóvenes adultos
- Publicación del enlace en el portfolio y en comunidades educativas.
- Incorporación de actualizaciones automáticas a través de AWS Amplify.

#### **Recursos necesarios:**

- Cuentas de desarrollador en Google Play
- Infraestructura AWS (Amplify, Cognito, DynamoDB y S3)
- Herramientas de análisis

## **FUENTES DE INGRESOS O FINANCIACIÓN:**

El objetivo es garantizar una base económica sólida que financie la infraestructura, mantenimiento, actualizaciones y futuras mejoras del sistema.

Tenemos pensado varias **estratégias de ingresos**:

### **-Funcionalidades limitadas:**

Al descargar la app habrá funciones limitadas como por ejemplo un máximo de hogares, de insignias, de niveles... y al intentar acceder aparecerá un mensaje ofreciendo al usuario acceder a una **versión premium**.

### **-Versión Premium:**

El sistema ofrecerá una **versión gratuita** con las funcionalidades esenciales (creación de hogares, asignación de tareas, calendario y ranking básico), y una **versión Premium** con herramientas avanzadas.

El beneficio esperado sería atraer usuarios a través del modelo gratuito y que se pasen a versión de pago mediante suscripción mensual o anual.

### **-Colaboraciones y patrocinios:**

Buscaremos alianzas con marcas, influencers y empresas vinculadas a la vida doméstica, sostenibilidad y organización familiar. Tendremos ingresos y apoyo para dar visibilidad a nuestra app.

Contaremos con **programas de ayudas** de nuevos emprendedores digitales donde podremos tener una financiación del desarrollo de la app.

## **Listo para el Lanzamiento y los Próximos Pasos**

El proyecto está preparado para salir al mercado con una estrategia clara de crecimiento y un modelo de negocio sostenible.

### **Plan de Lanzamiento**



#### **Beta Interna**

Pruebas con entorno cercano para feedback inicial.



#### **Piloto Limitado**

Publicación en canal interno de Google Play Store para recopilar métricas.



#### **Lanzamiento Público**

Difusión en redes sociales y comunidades.

### **Modelo de Ingresos**

#### **Freemium**

Funcionalidades esenciales gratuitas para atraer a una amplia base de usuarios y demostrar el valor de la aplicación.

#### **Suscripción Premium**

Acceso a herramientas avanzadas como presupuestos detallados, mayor almacenamiento de documentos y personalización avanzada para usuarios intensivos.

## **6.4. Plan de Recursos Humanos (RRHH)**

### **DEFINICIÓN PERFILES REQUERIDOS:**

#### **Desarrollador Full Stack:**

Encargado de construir, mantener y mejorar la aplicación en todas sus capas: frontend (interfaz de usuario) y backend (servidor, base de datos y lógica). Debe dominar JavaScript, Cordova, AWS Amplify, AppSync y DynamoDB. Su función principal sería garantizar la estabilidad, seguridad y rendimiento de la aplicación.

#### **Diseñador UX/UI:**

Responsable de optimizar la experiencia del usuario, diseñar interfaces atractivas y mantener la coherencia visual del producto. Se encargará de transformar la retroalimentación de las pruebas en mejoras visuales y de usabilidad. Debe dominar herramientas como Figma o Canva y aplicar principios de diseño centrado en el usuario.

#### **QA TESTER:**

Encargado de ejecutar pruebas funcionales, unitarias y de usabilidad para garantizar la calidad del producto antes de su lanzamiento. Documenta errores, valida correcciones y asegura que las actualizaciones no generen nuevos fallos. Debe conocer metodologías ágiles, herramientas de testing y buenas prácticas de control de calidad.

#### **Especialista en Marketing:**

Responsable de definir la visión del producto, establecer prioridades en el desarrollo y coordinar la comunicación entre los equipos técnico, de diseño y de marketing. Debe tener conocimiento de metodologías ágiles (Scrum) y habilidades de liderazgo. Asegura que el producto evolucione conforme a las necesidades del mercado y de los usuarios.

### **Especialista en seguridad:**

Garantiza la estabilidad del entorno en la nube (AWS), automatiza los procesos de despliegue y controla los permisos y datos de los usuarios. Es esencial para la versión final del producto y su escalabilidad.

### **ESTRATEGIA DE BÚSQUEDA:**

Nuestra estrategia de búsqueda de perfiles profesionales para HomeTaskManager se centrará en identificar candidatos cualificados mediante canales digitales, redes profesionales y comunidades tecnológicas. El objetivo es atraer talento con conocimientos técnicos y habilidades de trabajo en equipo dentro de un entorno ágil y colaborativo.

Se publicarán ofertas en sitios especializados que permitan filtrar por perfil y experiencia en proyectos tecnológicos como linkedIn , InfoJobs y Tecnoempleo.

### **ESTABLECIMIENTO DE NETWORKING:**

Para asegurar el crecimiento del proyecto HomeTaskManager y facilitar futuras contrataciones y colaboraciones, se establecerá una estrategia activa de networking. El objetivo es crear una red sólida de contactos profesionales dentro del sector tecnológico, diseño digital y marketing, con el fin de atraer talento, obtener asesoramiento especializado y generar oportunidades de colaboración a largo plazo.

Acciones para crear y mantener la red de contactos:

- Participación en eventos y ferias tecnológicas
- Participación en programas de emprendimiento
- Publicaciones y visibilidad de proyecto
- Participación en comunidades digitales
- Uso de LinkedIn

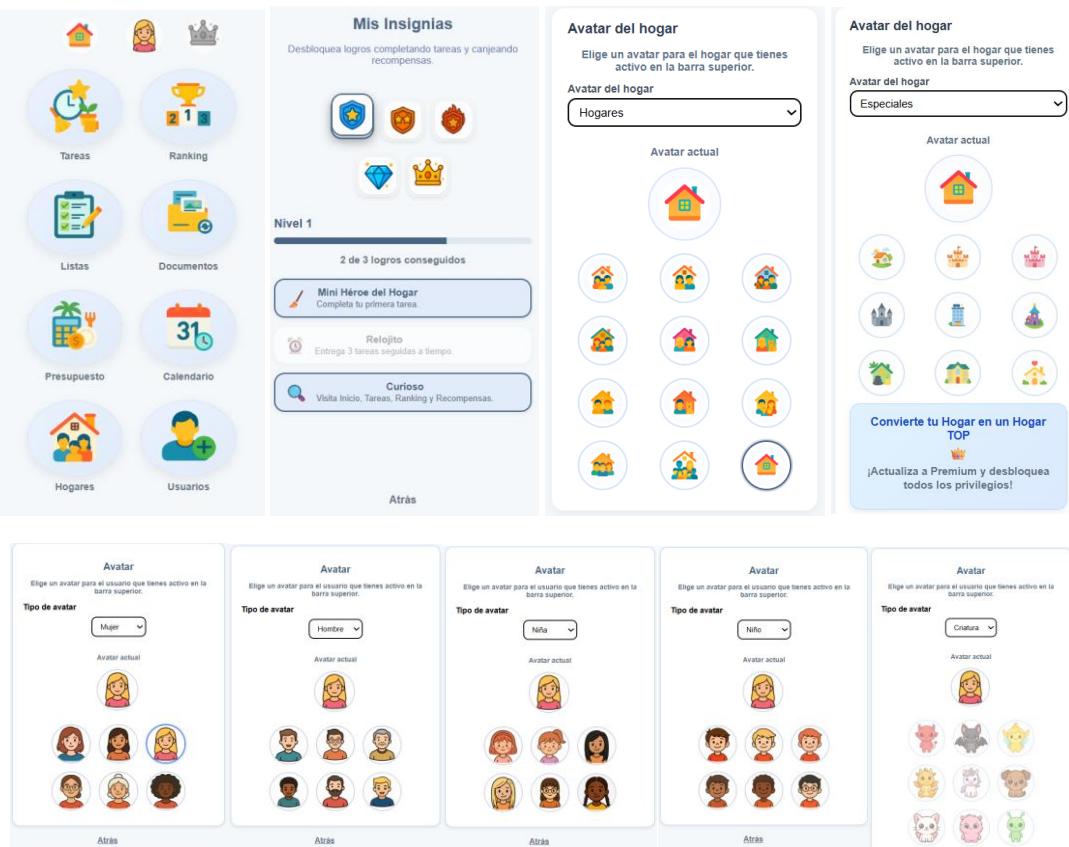
## 7. UNA APP EN DESARROLLO

### 7.1. Últimos retoques

Durante la última etapa del proyecto se han llevado a cabo una serie de acciones enfocadas a:

- Mejorar el **código**
- Mejorar la **experiencia de usuario** y la **navegación**
- Mejorar la implementación de la **Gamificación** como eje central de nuestra app
- Detección y corrección de **errores y fallos**
- Mejora de **roles de hogar** y permisos (administradores, miembros y juniors)
- Creación de **avatares e iconos** con una misma línea de estilo cartoon para toda la familia
- Creación de **FAQ** y un **tutorial** inicial para comenzar a usar la app

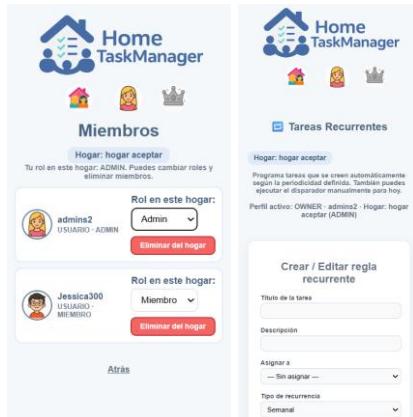
### AVATARES E ICONOS



## TUTORIAL INICIAL



## ROLES Y TAREAS RECURENTES

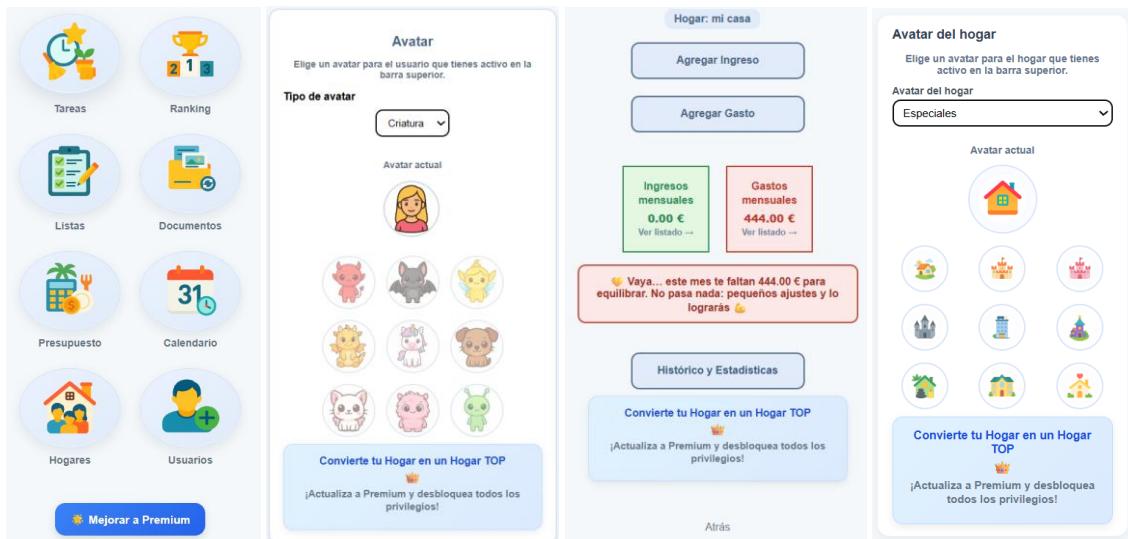


\*Estas mejoras se encuentran detalladas en el último apartado del Anexo 5.

## 7.2. Funciones Premium

Con tal de permitir el crecimiento y obtener ingresos, se ha planteado una versión Premium que incluya nuevas y mejores funcionalidades.

Se ha creado un enlace accesible desde varias páginas de la app informando de las mejoras que pueden obtenerse convirtiendo el hogar en un “**Hogar Top Premium**”:



Estos enlaces nos llevan a una página en la cual se describen las próximas funcionalidades previstas de implementar:



**PRÓXIMAMENTE!**

Con Hogar TOP Premium tendrás todas las funciones avanzadas pensadas para familias que quieren tenerlo TODO bajo control.

**¿Qué incluirá la versión Hogar TOP Premium?**

- Avatares especiales de hogar y usuario**  
Desbloquea todos los avatares especiales para que tu hogar tenga una identidad única y exclusiva.
- Documentos compartidos**  
Sube y consulta documentos compartidos por todos los adultos del hogar.
- Notificaciones emergentes**  
Recibe Notificaciones personalizadas para estar al día de todo ¡Tu eliges!

**Resumen avanzado del hogar**  
Históricos, estadísticas y consejos para seguir mejorando tu hogar.

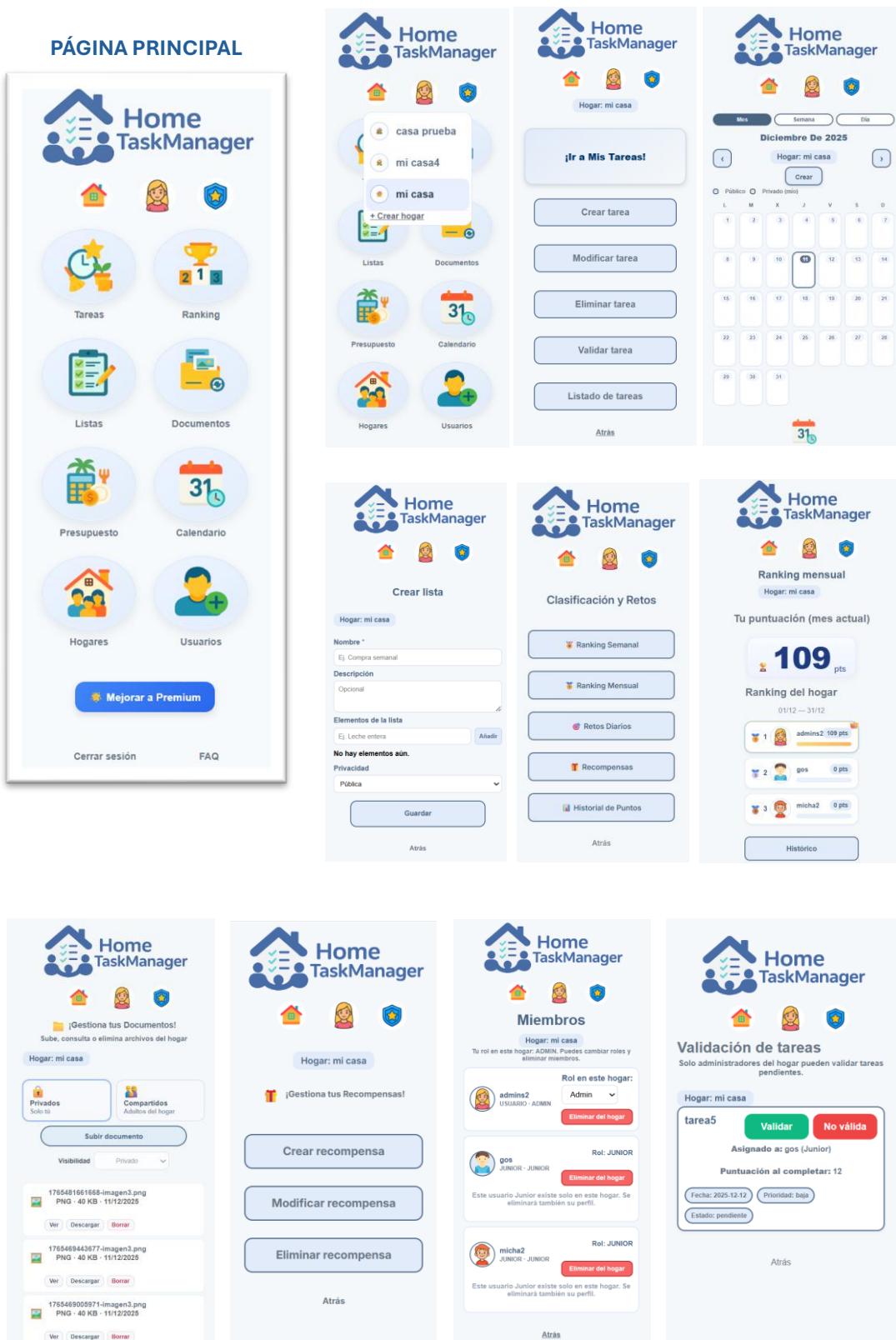
| Comparativa: Gratis vs Premium | Plan Gratis | Hogar TOP Premium |
|--------------------------------|-------------|-------------------|
| Avatares especiales            | ✗           | ✓                 |
| Documentos compartidos         | ✗           | ✓                 |
| Notificaciones emergente       | ✗           | ✓                 |
| Históricos y estadísticas      | ✗           | ✓                 |
| Más opciones de organización   | Limitadas   | Ampliadas         |

**Preguntas frecuentes**

- ▶ ¿Puedo seguir usando la versión gratis?
- ▶ ¿Qué pasa con mis datos si dejo de ser Premium?
- ▶ ¿Dónde activo Hogar TOP Premium?

**¿Listo para llevar tu hogar al siguiente nivel?**

### **7.3. Resultado final**



Estas capturas de pantalla muestran a modo de ejemplo algunas de las principales páginas creadas, así como el diseño final.

Se ha optado por mantener una misma gama de colores, un diseño intuitivo, visual, motivador y para toda la familia.

El equipo ha realizado además de este documento y sus anexos, un **portfolio** y un **video** mostrando la app en funcionamiento y todo el proceso realizado para obtener el resultado final.

#### 7.4. Mirada hacia el futuro

Llegados a este punto, podemos ver que nuestra app parte de una base sólida y funcional, la cual nos permitirá continuar **ampliando funcionalidades** mediante un desarrollo en el tiempo basado en nuestra experiencia y en la de los usuarios que comiencen a utilizarla en su día a día.

Con tal de hacerla más profesional, habiendo adquirido un gran conocimiento durante estos meses, algunos de los pasos siguientes que pretendemos dar son los siguientes:

- Seguir mejorando la **refactorización**
- Pulir y realizar algunos ajustes en la **BBDD**
- Incluir las **consultas a GraphQL** directamente generadas por amplify sin necesidad de repetir el código
- Implementar las **notificaciones S3** en AWS
- Optimizar el código y hacer nuestra app menos pesada
- Utilizar esta base de app **híbrida y escalable** con tal de hacerla funcional también para IOS y navegadores
- Seguir incorporando **nuevas funcionalidades** (avatares personalizados, competición entre hogares, recordatorios, integrar disparadores Lambda para tareas recurrentes, estadísticas e históricos, más niveles de juego, posibilidad por parte de los usuarios de hacernos llegar incidencias e ideas de mejora...)

Una vez obtenido un resultado con el cual nos sintamos satisfechas y cómodas, la idea es proceder a adaptar nuestra app no solamente para Android mediante la subida a Google Play, sino también adaptarla para hacerla disponible en IOS y ponerla en marcha para que cualquier persona pueda utilizarla, ¡y empezar a disfrutar de ella!

## 8. CONCLUSIÓN Y APRENDIZAJES

### 8.1. Evolución

El equipo ha ido progresando a lo largo de todo un año, no solamente en lo que a la arquitectura de software y desarrollo de código se refiere, sino también en la forma de trabajar y colaborar para la consecución de unos objetivos comunes.

La forma en la que todo este trabajo se ha plasmado ha sido a partir de la elaboración de un documento resumen en cada una de las fases, la **Minuta**.

En dicho documento se han ido recogiendo todas las conversaciones, decisiones, reuniones, avances y dificultades que han ido surgiendo.

Por ello nos parece interesante mostrar un resumen de dichos documentos para plasmar en este documento dicha evolución.

### MEMORIA UNIFICADA DEL PROCESO DE TRABAJO EN EQUIPO

El trabajo del equipo comenzó con una primera toma de contacto en marzo, donde se asentaron los cimientos del proyecto HomeTaskManager. En estas primeras reuniones se revisaron los **requisitos iniciales**, se intercambiaron ideas sobre la solución y se establecieron **roles generales** entre los integrantes.

A lo largo de las distintas minutas se observa cómo el equipo fue madurando su forma de organización: pasó de una división básica de tareas a una planificación más estratégica basada en roles estables, metodologías como **Scrum**, y herramientas como **GitHub**, la **wiki** del campus y **WhatsApp**.

La comunicación del grupo se mantuvo constante durante todo el proyecto, combinando:

- **Videollamadas** periódicas para revisar avances y resolver problemas.
- Mensajes de WhatsApp para **coordinación diaria** y emergente.
- **Hilos del campus** para documentar decisiones, dudas al tutor y actividades entregables.
- **GitHub** como repositorio oficial del código.

## **1. Inicio del Proyecto — Organización básica y primeras decisiones**

**(Entregable 1)**

### **Situación inicial**

- Formación del equipo y primeros contactos.
- Selección del proyecto: **una app para la gestión de tareas y documentos del hogar.**
- Lectura de rúbricas, actividades y revisión de contenidos necesarios.

### **Decisiones clave**

- Reparto inicial de tareas.
- Elección de medios de comunicación (campus como canal formal, WhatsApp como apoyo).
- Primeras aclaraciones con el profesor.
- Metodologías iniciales: estudio de mercado, entrevistas, mapa de empatía y técnicas *Design Thinking*.

### **Problemas iniciales**

- **Comunicación desorganizada**, dudas constantes y coordinación compleja.
- Diferencias de ritmo y expectativas → reconfiguración del equipo.

## **2. Consolidación y estructura de trabajo — Organización formal**

**(Entregable 2)**

### **Mejoras implementadas**

- Reuniones semanales estructuradas.
- Reparto detallado de responsabilidades.
- Uso más responsable del campus para dejar trazabilidad.

### **Decisiones destacadas**

- Introducción de **Scrum** como metodología interna.
- Revisión de roles, comunicación, plan de resolución de conflictos.
- Solicitud de tutoría para resolver dudas técnicas y de normativa.

## **Avances del equipo**

- División de tareas más clara.
- Documentos mejor estructurados.
- Integración de aportes individuales en materiales conjuntos como:
  - Business Model Canvas
  - Cronograma
  - OKR
  - Metodologías

## **Obstáculos**

- Necesidad de **aclarar alcance** y requisitos con el tutor.
- Definir programas y herramientas adecuadas para el proyecto.

## **3. Fase técnica — Primer prototipo funcional y crecimiento tecnológico (Entregable 3)**

### **Gran salto técnico**

Aquí empieza la parte más exigente del proyecto:

- Se decide construir **una app híbrida**, considerando Flutter y finalmente usando Cordova + AWS.
- Creación del **backend en la nube**:
  - AWS AppSync
  - DynamoDB
  - Amplify
  - IAM
- Se crea un prototipo funcional con:
  - Registro
  - Creación de hogar
  - Calendario básico

### **Decisiones clave**

- Enviar documento al profesor para validar idea tecnológica.
- Revisión de diagramas, arquitectura y wiki.
- Uso de GitHub como repositorio principal.

### **Obstáculos enfrentados**

- Curva de aprendizaje muy elevada en tecnologías desconocidas.
- Errores de compilación, problemas con plugins, configuración de SDK y entornos.

### **Hitos logrados**

- Compilación correcta en varios equipos.
- Backend operativo.
- Diagramas revisados y corregidos.
- Wiki en construcción.

## **4. Diseño y usabilidad — Prototipo visual completo**

**(Entregable 4)**

### **Foco del periodo**

- Prototipado visual completo en **Figma**.
- Revisión visual, usabilidad, heurísticos, coherencia gráfica.

### **Decisiones clave**

- Reparto de tareas muy especializado:
  - Patri: Wireframe, wiki y despliegue.
  - Ana: Alta fidelidad, minuta y navegación.
  - Jessica: Heurísticos, repositorio y prototipado.

### **Avances importantes**

- Pantallas principales del sistema: login, registro, tareas, listas, calendario, recompensas.
- Guía visual: colores, iconos, tipografías.

- Integración del diseño en Cordova.
- Revisión de la navegación y roles (administrador, usuario, menor).

## **Pendientes**

- Pantallas secundarias.
- Integración definitiva en la app real.
- Refinamiento del diseño antes de entregar.

## **5. Cierre técnico — Integración, pruebas y documentación final**

### **(Entregable 5)**

#### **Fase más sólida del equipo**

El grupo llega aquí con un sistema casi completo:

#### **Funcionalidades implementadas**

- Login + hogares ,Tareas (I y II), Ranking semanal/mensual, Recompensas, Retos e insignias dinámicas, Calendario, Listas, Documentos (subida en S3).

#### **Decisiones clave**

- Unificar estilos visuales.
- Priorizar pruebas en distintos dispositivos.
- Consolidar la versión oficial en GitHub.
- Elaborar el banco de pruebas y documentación final.

#### **Obstáculos**

- Sincronización del sistema de puntos entre ranking/retos/recompensas.
- Errores del SDK, CORS, permisos IAM.
- Sobrecarga por otras asignaturas.
- Ajustes visuales entre pantallas.

## **6. Estado final del proyecto — HomeTaskManager**

**(Entregable final)**

### **Foco del periodo**

- Consolidación de una versión final estable de la aplicación.
- Corrección de errores, coherencia visual y optimización para dispositivos móviles.
- Preparación de la entrega, documentación y evaluación final.

### **Decisiones clave**

- Priorizar estabilidad y correcto funcionamiento frente a nuevas funcionalidades.
- Documentar incidencias técnicas y soluciones aplicadas.
- Mantener las mejoras no esenciales como propuestas futuras.

### **Avances importantes**

- Mejora de la coherencia visual entre pantallas.
- Corrección de errores de visualización en móvil.
- Pruebas realizadas en entorno Cordova e integración con AWS Amplify.

### **Estado del proyecto**

- Proyecto estable, sin bloqueos críticos.
- Listo para entrega, presentación y evaluación final.
- Posibles mejoras evolutivas identificadas (visuales y funcionales no esenciales).

## Resultado

El equipo cierra con una **app funcional, documentada, probada y estable**, lista para entrega final.

A lo largo del proyecto, el Grupo 14 ha vivido un proceso de transformación significativo tanto en su forma de trabajar como en su cohesión interna.

Comenzaron como un grupo que aún buscaba **entender las exigencias** del proyecto, con **dudas constantes**, problemas de comunicación y una organización inicial poco definida. Con el paso del tiempo, fueron aprendiendo a coordinarse, estableciendo canales claros, **roles más definidos** y una dinámica de trabajo más estable. La comunicación, que al principio generaba confusión, se convirtió en una herramienta de apoyo mutuo y en un espacio seguro para debatir decisiones, aclarar dudas y repartir tareas de forma más equilibrada.

El **crecimiento** no fue solo organizativo, sino también técnico. El equipo pasó de enfrentarse con inseguridad a tecnologías desconocidas a ser capaz de configurar entornos complejos, trabajar con AWS, manejar Cordova, gestionar repositorios en GitHub y construir un prototipo real y funcional. Los obstáculos, lejos de desmotivar, fortalecieron la colaboración: cuando surgían errores, pruebas fallidas o bloqueos técnicos, el grupo utilizaba la comunicación y el apoyo mutuo para superarlos. Esa capacidad de adaptación marcó un antes y un después.

Con el avance de los entregables, el equipo no solo mejoró su metodología, sino también su **madurez**. Empezaron a **anticipar problemas**, organizar reuniones más **productivas, documentar** mejor sus avances y tomar decisiones de manera consensuada y fundamentada. La carga de trabajo, que en ocasiones generó **estrés**, se equilibró gracias a una distribución más consciente y a la consolidación de cada integrante en tareas específicas donde podía aportar más valor.

Finalmente, el grupo llegó a la última fase convertido en un equipo cohesionado, autónomo y capaz de desarrollar un producto completo, funcional y bien documentado. Pasaron de la incertidumbre y el desorden inicial a una coordinación efectiva, una comunicación constante y una capacidad técnica que fueron construyendo juntas. Su evolución demuestra no solo aprendizaje, sino también **resiliencia, compromiso** y una mejora notable en su forma de trabajar como **verdadero equipo**.

## **REFLEXIONES PERSONALES**

### **Patricia Morato Jiménez**

Este trabajo me ha servido para comprender mejor cómo funciona el desarrollo de una aplicación real en equipo.

He aprendido a usar los diferentes programas utilizados de forma más consciente, resolviendo errores y conflictos que al principio parecían complicados.

He mejorado en la organización del código y en la maquetación de las pantallas con HTML, CSS y JavaScript.

A lo largo del proyecto he notado una evolución clara en mi forma de trabajar y de afrontar los problemas.

Aunque ha habido momentos de dificultad, han sido clave para aprender.

En general, este proyecto me ha ayudado a ganar confianza y experiencia práctica.

Agradezco al equipo lo unidas que hemos estado siempre y sin ellas nada hubiera sido posible.

### **Ana de la Torre Macías**

Este proyecto no solo me ha ayudado a afianzar conocimientos, sino también a descubrir capacidades que antes no tenía tan claras. El apoyo de mis compañeras ha sido fundamental: juntas hemos demostrado que somos un equipo fuerte y unido. En los momentos difíciles supimos acompañarnos y sacar adelante lo que parecía complicado, y eso ha marcado la diferencia. Gracias a todo este proceso ahora confío más en mí y en lo que puedo lograr.

### **Jéssica Marrón Carmona**

Ha sido una experiencia muy enriquecedora tanto en lo profesional como en lo personal. Además de aprender a desarrollar una app desde cero con todo lo que ello conlleva, el aprendizaje de trabajar en grupo ha sido un punto clave que marca un antes y un después en este ciclo.

He descubierto que puedo hacer muchas más cosas de las que me creía capaz, y siento que es un punto de partida desde donde puedo llegar mucho más lejos.

Agradezco a mis compañeras por haber realizado juntas este viaje con tanto esfuerzo, y a nuestro tutor por haber sabido llevarnos a buen puerto desde la empatía y la libertad.

# HomeTaskManager:

## Más que una App, un Aliado para el Hogar

*Hemos completado un viaje de desarrollo integral, creando una solución viable y bien ejecutada que responde a una necesidad real del mercado.*

### Logros Clave del Proyecto



#### 1. Validamos

una necesidad de mercado y diseñamos una solución centrada en el usuario.

#### 2. Implementamos

una arquitectura cloud moderna, serverless y escalable en AWS.

#### 3. Creamos

una experiencia de usuario atractiva y funcional, validada con prototipos y pruebas.

#### 4. Completamos

un ciclo de desarrollo profesional de principio a fin, superando retos técnicos y metodológicos.

## 9. ANEXOS

### ANEXO 1. Entrevistas cualitativas y técnicas de ideación

#### Entrevistas cualitativas:

Cada uno de los miembros del equipo ha realizado entrevistas a diferentes personas de nuestro entorno y estos son los resultados obtenidos:

##### ❖ Entrevista nº1:

###### **¿Qué necesidades tiene el sujeto?**

-Preparar la comida para toda la semana por adelantado.

###### **¿Qué problemas se lo impiden?**

-Falta de tiempo.

-Restricciones e intolerancias alimentarias.

###### **¿Cuáles son sus deseos?**

-Reducir el tiempo necesario para cocinar

-Abaratar los gastos de la compra.

#### Observaciones:

-Repite comidas para cocinar menos.

-Se interesa por el Bach cooking.

##### ❖ Entrevista nº 2:

###### **¿Qué necesidades tiene el sujeto?**

-Necesita tener la casa organizada.

###### **¿Qué problemas se lo impiden?**

-No colaboran.

###### **¿Cuáles son sus deseos?**

-Me gustaría animar a que cambien la actitud y colaboren más.

**Observaciones:**

-Se nota que está cansada.

❖ Entrevista nº 3:

**¿Qué necesidades tiene el sujeto?**

-Que le recuerden las cosas de las actividades de los niños.

**¿Qué problemas se lo impiden?**

-La falta de ganas de estar preguntando.

**¿Cuáles son sus deseos?**

-Que se lo recuerden sin tener que preguntar.

**Observaciones:**

-Tener una aplicación común le vendría bien.

## Técnica ¿Qué, cómo y por qué?

La técnica ¿Qué, cómo y por qué? es una metodología utilizada para estructurar y analizar ideas, problemas o proyectos.

Observando al público objetivo y sus necesidades, completamos los siguientes diagramas entre todas para resumir las necesidades básicas y entender mejor al usuario final.

| Fotos   | ¿Qué está sucediendo?                 | ¿Cómo está sucediendo?                    | ¿Por qué está sucediendo?             |
|---|---------------------------------------|---|---------------------------------------|
|    | A La mascota quiere pasear            | Está esperando que alguien lo saque       | Porque nadie hace cuenta              |
|   | B Se han olvidado de la mascota       | Nadie de la casa puede atenderle          | Porque están cansados los dueños      |
|   | C Esta triste                         | Por olvido y desinterés                   | Porque han perdido el interés         |
|   | A No colabora en las tareas del hogar | La pareja le pide ayuda                   | Porque esta acostumbrado a no hacerlo |
|   | B No tiene interés por nada           | Se niega a todo                           | Porque no tiene motivación            |
|   | C Prefiere ver tele a ayudar          | La pareja insiste                         | Porque no tiene interés               |
|  | A La madre está estresada             | Se lleva las manos a la cabeza por estrés | No lo tenía bien planificado          |
|   | B Los niños ya están listos           | Tienen sus ropa preparadas para irse      | Porque es la hora                     |
|   | C Lleva a sus hijos a sus actividades | Es a la misma hora                        | No tiene ayuda                        |

| Fotos   | ¿Qué está sucediendo?                | ¿Cómo está sucediendo?     | ¿Por qué está sucediendo?       |
|---|--------------------------------------|----------------------------|---------------------------------|
|  | A Estoy tirada en el sofá            | Me genera estrés           | No tengo orden                  |
|   | B No tengo ganas de moverme          | Nadie me ayuda             | Estoy cansada                   |
|   | C Todo está sucio                    | Me agoto de pensar         | Llevo días así                  |
|  | A Desorganización de documentos      | Me genera estrés           | No tengo orden                  |
|   | B No encuentro el papel que necesito | Estoy desordenando la casa | Tengo muchos papeles            |
|   | C Tengo prisa                        | Estoy dando vueltas        | Mis hijos me lo desordenan todo |
|  | A La madre está enfadada             | No recogen                 | No reparte tareas               |
|   | B Hay desorden                       | Los niños no están         | No se le ha asignado tareas     |
|   | C El padre no está                   | No quiere ayudar           | No sabía que tocaba limpiar     |

## **ANEXO 2. VIABILIDAD DEL PROYECTO**

### **La capacidad de ejecución y los recursos disponibles.**

La capacidad de ejecución y los recursos disponibles representan los elementos clave que determinarán si nuestro proyecto puede desarrollarse con éxito.

#### **1. Capacidad de ejecución:**

- o Equipo humano: Contamos con un equipo especializado en desarrollo de software, diseño , marketing y gestión de proyectos.
- o Metodologías de trabajo: Tenemos un cronograma de la ejecución de cada fase con objetivos alcanzables.

#### **2. Recursos disponibles:**

- o Tecnológicos: Tenemos una Infraestructura para desarrollo, como entornos de programación, herramientas de diseño, servidores y sistemas de prueba.
- o Financieros: Nuestro presupuesto para cubrir gastos de desarrollo, marketing, almacenamiento en la nube, etc.
- o Herramientas y plataformas: Contamos con un sistema de control de versiones (Git), bases de datos, frameworks de desarrollo y otros sistemas que faciliten la creación de la app.

### **Un cálculo aproximado de nuestros costes sería:**

#### **1. COSTES INTERNOS:**

##### **PERSONAL:**

- Desarrolladores (backend y frontend)
- Diseñador UX/UI

Total de personal: 0€

## **INFRAESTRUCTURA Y HERRAMIENTAS INTERNAS:**

- Actualización de equipos: 1.500€
  - Licencias de software de diseño (Adobe, Figma, etc.) : 1.000€
  - Plataformas de colaboración : 300€
- Total de infraestructura: 2.800€

## **2.COSTES EXTERNOS:**

### **SOFTWARE Y HERRAMIENTAS EXTERNAS:**

- Servicios en la nube : 1.000€
  - APIs: 750€
  - Certificados SSL y dominios: 200€
- Total de software y herramientas externas: 1.950€

### **CONSULTORÍA Y SERVICIOS:**

- Consultoría técnica para optimización: 2.500€
  - Evaluación legal específica: 1.500€
- Total de consultorías: 4.000€

### **COSTES DE INVESTIGACIÓN Y DESARROLLO:**

- Pruebas de usuarios: 1.000€
  - Despliegue inicial: 500€
- Total de I+D: 1.500€

### **COSTES LEGALES Y ADMINISTRATIVOS:**

- Registro de marca y propiedad intelectual: 750€
  - Honorarios de asesoramiento legal: 1.500€
- Total de costos legales: 2.250€

**COSTES DE MARKETING Y PROMOCIÓN:**

- Campañas de publicidad en redes: 2.200€
  - Banners : 500€
  - Optimización para motores de búsqueda (SEO): 1.000€
- Total de marketing y promoción : 3.700€

**El gasto total para la creación de la aplicación sería de 16.200€**

### **ANEXO 3. BASE DE DATOS**

Se adjunta en un documento a parte dada su extensión.

Dicho documento contiene el schema graphql utilizado para la creación de la base de datos DynamoDB en AWS.

### **ANEXO 4. MANUAL DE INSTALACIÓN Y DOCUMENTACIÓN**

Se adjunta en un documento a parte dada su extensión.

### **ANEXO 5. DESARROLLO DE CÓDIGO**

Se adjunta en un documento a parte dada su extensión.

## ANEXO 6. CASOS DE PRUEBA

### CASO DE PRUEBA 1: LOGIN

#### • INSTANCIA N° 1: REGISTRAR NUEVO USUARIO

##### Descripción del test:

Comprobar que el sistema permite registrar correctamente un nuevo usuario y almacenar sus datos en la base de autenticación.

##### Condiciones de ejecución:

- La aplicación debe estar instalada y abierta.
- Acceso a la pantalla “Registro”.
- Conexión a Internet activa.

##### Entrada:

1. Pulsar en “Registrarse”.
2. Introducir un correo electrónico válido.
3. Escribir una contraseña que cumpla los requisitos mínimos.
4. Confirmar la contraseña y pulsar “Registrar”.

##### Resultado esperado:

- El sistema crea la cuenta correctamente.
- El usuario recibe código de verificación a su correo, lo introduce y confirma.
- Se redirige al login para iniciar sesión.

##### Evaluación de la prueba: Satisfactoria

The figure consists of two side-by-side screenshots of a web application's registration process. The left screenshot shows the initial registration form. It includes fields for 'Email' (admin@pruebaDosis@gmail.com), 'Contraseña:' (password), and 'Repetir contraseña'. Below these are buttons for 'Entrar' (Login), 'Atrás', 'Olvidaste la contraseña?' (Forgot password?), and 'Envir código al email' (Send code to email). There are also links for 'Código recibido por email:' (Code received via email:) and 'Nueva contraseña:' (New password:). The right screenshot shows a confirmation page. It displays the user's name ('Julia'), phone number ('622212223'), email ('admin78@pruebaDosis@gmail.com'), and a generated verification code ('776584'). It features a large 'Confirmar' (Confirm) button and a smaller 'Reenviar código' (Resend code) button. At the bottom, there is a link 'Ya tengo una cuenta' (I have an account).

- **INSTANCIA N° 2: CONTRASEÑA INCORRECTA**

#### **Descripción del test**

Verificar que el sistema bloquee el acceso al introducir una contraseña incorrecta.

#### **Condiciones de ejecución:**

- Usuario registrado en el sistema.
- Acceso a pantalla de inicio de sesión.

#### **Entrada :**

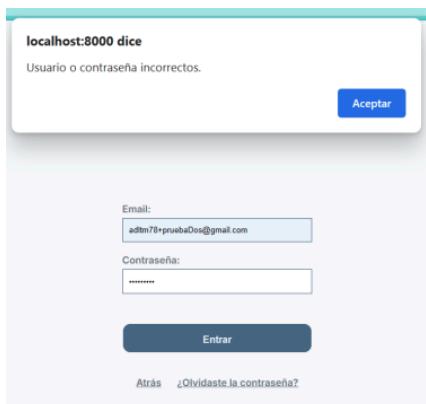
1. Abrir la app y Pulsar "Iniciar sesión".
2. Escribir email válido.
3. Introducir contraseña incorrecta.
4. Pulsar "Entrar".

#### **Resultado esperado:**

- El sistema muestra el mensaje: “Usuario o contraseña incorrectos”.
- El acceso NO se permite.
- El campo contraseña no se limpia, se tiene que borrar e introducir otra.

#### **Evaluación de la Prueba:**

**Estado:** Satisfactoria



- **INSTANCIA N° 3: INICIO CORRECTO**

**Descripción del test:**

Validar que un usuario pueda iniciar sesión en la aplicación utilizando credenciales correctas y que el sistema rechace accesos no válidos.

**Condiciones de ejecución:**

- La aplicación debe estar instalada y abierta.
- El usuario debe estar previamente registrado con email y contraseña válidos.
- Conexión a Internet activa para validar credenciales en AWS Cognito.

**Entrada :**

1. Abrir la aplicación.
2. Acceder a la pantalla de login y Pulsar "Iniciar sesión".
3. Introducir email válido.
4. Introducir contraseña válida.
5. Pulsar botón "Entrar".

**Resultado esperado:**

- El sistema valida las credenciales.
- El usuario accede correctamente a la pantalla principal de la app.
- No se muestran mensajes de error.

**Evaluación de la Prueba:**

**Estado:** Satisfactoria



- **INSTANCIA N° 4: USUARIO NO REGISTRADO**

**Descripción del test:**

Verificar que el sistema impide el acceso cuando se intenta iniciar sesión con un usuario no registrado.

**Condiciones de ejecución:**

- No debe existir un usuario con el correo utilizado.
- Acceso a la pantalla de login.

**Entrada:**

1. Abrir la aplicación.
2. Acceder a la pantalla de login y Pulsar "Iniciar sesión".
3. Introducir correo electrónico no registrado.
4. Introducir una contraseña cualquiera.
5. Pulsar "Entrar".

**Resultado esperado:**

- El sistema muestra el mensaje: "No existe ese usuario. Regístrate primero"
- El acceso no se permite.
- El usuario permanece en la pantalla de login.

**Evaluación de la prueba:**

**Estado:** Satisfactoria

The screenshot shows a modal dialog box at the top with the following text:  
localhost:8000 dice  
No existe ese usuario. Regístrate primero.  
A blue 'Aceptar' button is visible on the right side of the dialog.

Below the dialog is a login form with the following fields:  
Email:   
Contraseña:   
A blue 'Entrar' button is at the bottom of the form.  
At the very bottom left, there are small links: 'Atrás' and '[¿Olvidaste la contraseña?](#)'.

- **INSTANCIA N° 5: RECUPERAR CONTRASEÑA**

**Descripción del test:**

Comprobar que el sistema permite recuperar el acceso a una cuenta mediante el proceso de restablecimiento de contraseña.

**Condiciones de ejecución:**

- El usuario debe tener una cuenta registrada.
- Debe disponer de conexión a Internet activa.

**Entrada:**

1. Acceder a la pantalla de login.
2. Pulsar en “Olvidé mi contraseña o usuario”.
3. Introducir correo electrónico registrado.
4. Confirmar la solicitud de recuperación.
5. Revisar el correo para código de recuperación.

**Resultado esperado:**

- El sistema envía un correo con un código de recuperación.
- El usuario puede establecer una nueva contraseña.
- El acceso vuelve a ser posible con la nueva clave.

**Evaluación de la prueba:**

**Estado:** Satisfactoria

The image shows two side-by-side screenshots of a web application interface. Both screenshots have a light gray background and a white central form area. At the top of each screenshot, there is a header bar with the text "localhost:8000 dice".

**Screenshot 1 (Left):** This screenshot shows a step where a password reset code has been sent via email. The message reads: "Te hemos enviado un código a tu correo." Below this, there is a login form with fields for "Usuario" (containing "admin@pruebasacnogmail.com") and "Contraseña". A blue "Aceptar" button is at the bottom right. Below the form, there is a link "Atrás" and a note "«Olvidaste la contraseña?». Introduce tu email y pide el código para cambiar la contraseña." At the bottom, there is a field labeled "Código recibido por email:" containing "123456", a field for "Nueva contraseña:" (with a note "Mínimo 8 caracteres"), and three buttons: "Enviar código al email", "Cambiar contraseña", and "Cancelar".

**Screenshot 2 (Right):** This screenshot shows the result of changing the password. The message reads: "Contraseña cambiada. Ya puedes iniciar sesión." Below this, there is a login form with fields for "Usuario" (containing "admin@pruebasacnogmail.com") and "Contraseña". A blue "Aceptar" button is at the bottom right. Below the form, there is a link "Atrás" and a note "«Olvidaste la contraseña?». Introduce tu email y pide el código para cambiar la contraseña." At the bottom, there is a field labeled "Código recibido por email:" containing "611997", a field for "Nueva contraseña:", and three buttons: "Enviar código al email", "Cambiar contraseña", and "Cancelar".

## CASO DE PRUEBA 2: BASE DE DATOS Y GESTIÓN DE TAREAS

- **INSTANCIA Nº 1: CREAR TAREA**

### Descripción del test:

Comprobar que el usuario puede crear una tarea doméstica y que esta se guarda correctamente.

### Condiciones de ejecución:

- Usuario logueado en la app.
- Acceso a pantalla “Crear tarea”.

### Entrada:

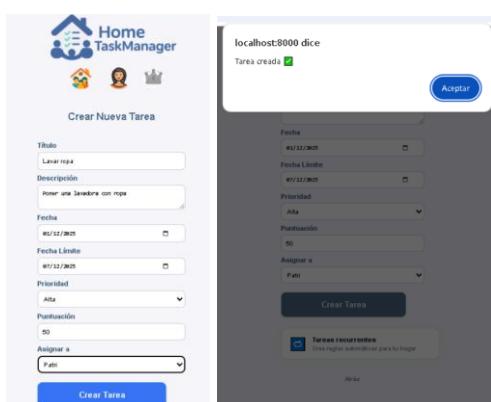
1. Pulsar botón “Crear tarea”.
2. Introducir título: “Fregar platos”.
3. Seleccionar fecha: 28/11/2025.
4. Pulsar botón “Crear Tarea”.

### Resultado esperado:

- La tarea aparece en la lista de tareas.
- Se visualiza en el calendario asignado.
- Se guarda en la base de datos sin errores.

### Evaluación de la Prueba:

**Estado:** Satisfactoria



- **INSTANCIA N° 2: EDITAR TAREA**

**Descripción del test:**

Comprobar que el usuario puede editar correctamente una tarea existente y que los cambios se actualizan en la base de datos.

**Condiciones de ejecución:**

- Usuario logueado.
- Debe existir al menos una tarea creada previamente.

**Entrada:**

1. Desde Tareas.
2. Pulsar la opción “Modificar Tarea”.
3. Modificar el nombre o la fecha de la tarea.
4. Guardar los cambios.

**Resultado esperado:**

- La tarea se actualiza con los nuevos datos.
- Se reflejan los cambios en la lista y el calendario.
- La información se guarda en la base de datos sin errores.

**Evaluación de la prueba:**

**Estado:** Satisfactoria

The screenshot shows a modal dialog for editing a task. It contains the following fields:

- Fecha: 27/11/2025
- Fecha límite: 30/11/2025
- Estado: Completada
- Prioridad: Alta
- Puntuación: 30
- Asignado a: manuel

Below these fields is a note: "Puedes elegir un Usuario o un Junior del hogar." At the bottom of the dialog are three buttons: "Guardar cambios" (highlighted in blue), "Eliminar", and "Cancelar". A message "Cambios guardados" with a checkmark is displayed at the bottom right. There is also a link "Atrás" at the very bottom.

- **INSTANCIA N° 3: ELIMINAR TAREA**

**Descripción del test:**

Verificar que el usuario puede eliminar una tarea existente y que esta desaparece del sistema.

**Condiciones de ejecución:**

- Usuario logueado.
- Al menos una tarea creada previamente.

**Entrada:**

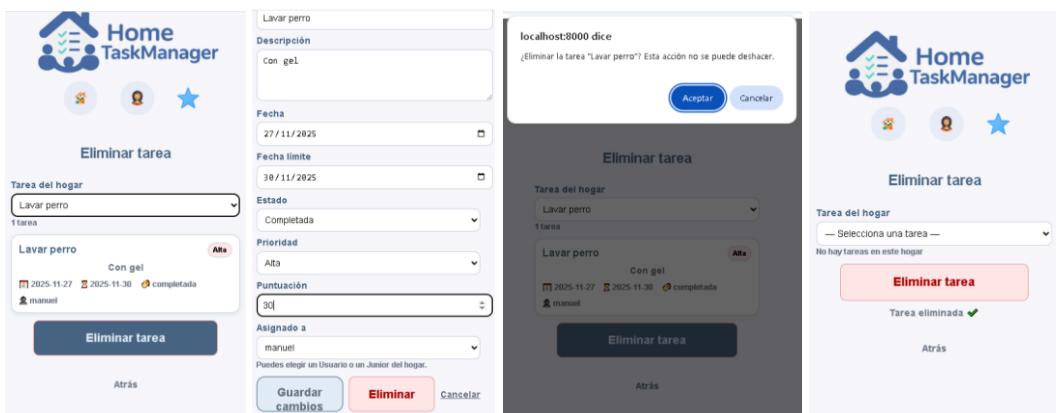
1. Desde Tareas.
2. Pulsar la opción “Eliminar Tarea”.
3. Confirmar la eliminación cuando el sistema lo solicite.

**Resultado esperado:**

- La tarea desaparece de la lista y del calendario.
- La base de datos elimina el registro sin errores.

**Evaluación de la prueba:**

**Estado:** Satisfactoria



- **INSTANCIA N° 4: CONSULTAR TAREAS**

**Descripción del test:**

Comprobar que el usuario puede visualizar correctamente las tareas almacenadas en la base de datos.

**Condiciones de ejecución:**

- Usuario logueado.
- Debe haber tareas registradas.

**Entrada:**

1. Abrir la aplicación e iniciar sesión.
2. Acceder a la pantalla donde se muestra la lista de tareas del hogar.
3. Esperar la carga automática de datos.

**Resultado esperado:**

- Se muestran todas las tareas registradas en el hogar.
- La información coincide con los datos almacenados en la base de datos.
- No aparecen errores de carga.

**Evaluación de la prueba:**

**Estado:** Satisfactoria



## CASO DE PRUEBA 3: GAMIFICACIÓN Y RECOMPENSAS

### • INSTANCIA Nº 1: ASIGNACIÓN DE PUNTOS POR TAREA COMPLETADA

#### Descripción del test:

Verificar que, al completar una tarea, la aplicación asigna correctamente la cantidad de puntos correspondiente al usuario.

#### Condiciones de ejecución:

- Usuario logueado correctamente.
- Al menos una tarea pendiente asignada al usuario.
- Conexión activa para sincronizar los cambios con la base de datos.

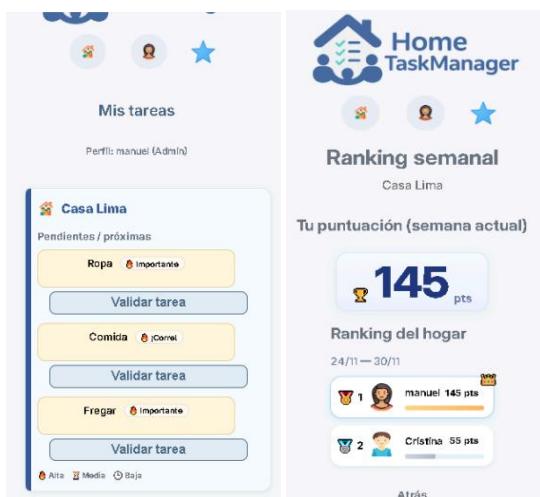
#### Entrada:

1. Acceder a la lista de tareas.
2. Seleccionar una tarea pendiente.
3. Marcar la tarea como “Completada”.
4. Acceder al ranking semanal, ranking mensual o historial de puntos.

#### Resultado esperado:

- Se puede completar una tarea.
- Se actualizan los puntos en la base de datos.
- El ranking o historial se actualizan correctamente.

#### Evaluación de la prueba: Satisfactoria



## • INSTANCIA N° 2: ACTUALIZACIÓN DEL RANKING

### Descripción del test:

Comprobar que el ranking de usuarios se actualiza correctamente cuando un usuario gana nuevos puntos.

### Condiciones de ejecución:

- Múltiples usuarios registrados en el mismo hogar.
- Todos los usuarios tienen un saldo de puntos inicial.
- Conexión activa con la base de datos.

### Entrada:

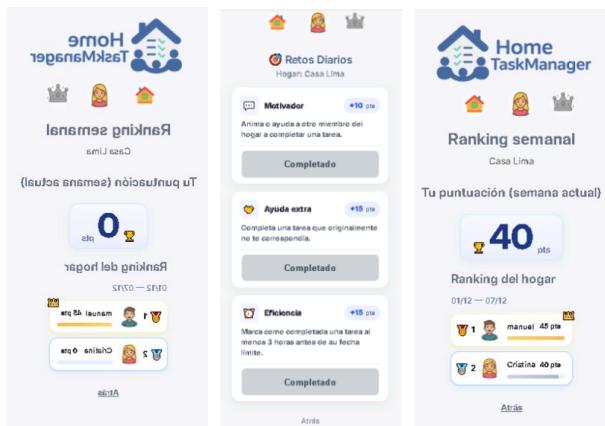
1. Iniciar sesión con un usuario que esté por debajo en el ranking.
2. Completar varias tareas para aumentar sus puntos.
3. Acceder a la sección de “Ranking”.

### Resultado esperado:

- El total de puntos del usuario aumenta.
- La posición del usuario en el ranking se recalcula.
- Si supera a otros usuarios, su nueva posición se refleja correctamente.

### Evaluación de la prueba:

**Estado:** Satisfactoria



### • INSTANCIA N° 3: OBTENCIÓN DE INSIGNIAS

#### Descripción del test:

Verificar que las insignias (logros) se otorgan correctamente al cumplir las condiciones definidas (por ejemplo, completar X tareas, consultar secciones, etc.).

#### Condiciones de ejecución:

- Usuario logueado.
- Sistema de logros activo.
- Existen insignias configuradas, como el logro “Curioso”.

#### Entrada:

1. Realizar las acciones necesarias para desbloquear una insignia (por ejemplo, visitar varias secciones o completar cierto número de tareas).
2. Acceder a la sección de logros o perfil del usuario.

#### Resultado esperado:

- La insignia correspondiente se marca como desbloqueada.
- El usuario puede ver visualmente la insignia ganada.

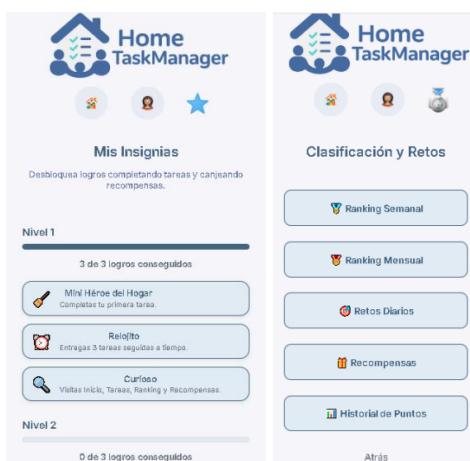
#### Resultado real observado:

- Algunas insignias, como el logro “Curioso”, funcionan correctamente en el navegador del ordenador, pero no se activan en la app móvil.

#### Evaluación de la prueba:

Parcialmente satisfactoria

La lógica de logros funciona en entorno web, pero presenta errores en la versión móvil, por lo que requiere revisión e implementación específica para dispositivo.



## INSTANCIA N° 4: CANJEAR PUNTOS POR RECOMPENSAS

### Descripción del test:

Comprobar que el usuario puede canjear sus puntos por recompensas y que el saldo de puntos se actualiza correctamente.

### Condiciones de ejecución:

- Usuario logueado con saldo suficiente de puntos.
- Módulo de recompensas activo y con recompensas disponibles.

### Entrada:

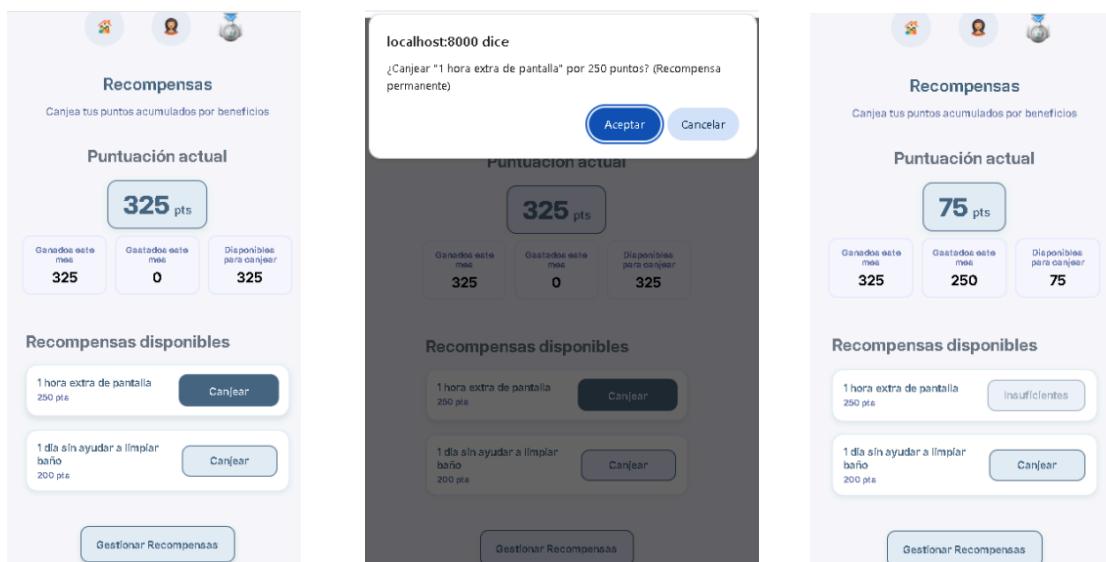
1. Acceder a la sección “Recompensas”.
2. Seleccionar una recompensa disponible.
3. Pulsar en “Canjear” .
4. Confirmar la acción si el sistema lo solicita.

### Resultado esperado:

- La recompensa se marca como obtenida o se añade al historial del usuario.
- El sistema descuenta del saldo del usuario la cantidad de puntos correspondiente.
- El nuevo total de puntos se muestra actualizado.

### Evaluación de la prueba:

**Estado:** Satisfactoria



## CASO DE PRUEBA 4: SISTEMA DE ALMACENAMIENTO DE DOCUMENTOS

- **INSTANCIA N° 1: SUBIR ARCHIVO A LA NUBE**

### Descripción del test:

Comprobar que el usuario puede subir correctamente un archivo al sistema de almacenamiento en la nube y que este queda disponible para su consulta posterior.

### Condiciones de ejecución:

- Usuario autenticado y dentro del módulo “Documentos”.
- Permiso de acceso a cámara o galería de archivos.
- Conexión a Internet activa para subir el archivo a la nube.

### Entrada:

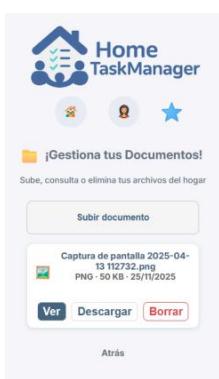
1. Acceder al módulo “Documentos”.
2. Pulsar el botón “Subir archivo”.
3. Seleccionar un archivo desde la galería (PDF, imagen o documento escaneado).
4. Confirmar la subida.

### Resultado esperado:

- El archivo se sube correctamente a la nube.
- El sistema muestra el documento en la lista de archivos disponibles.
- El archivo mantiene su formato original y se puede visualizar sin errores.

### Evaluación de la prueba:

Satisfactoria



La subida de archivos funciona correctamente. El archivo se sincroniza, aparece en la lista y puede abrirse sin errores.

## INSTANCIA N° 2: ACCESIBILIDAD AL ARCHIVO

### Descripción del test:

Verificar que un documento previamente almacenado puede ser abierto sin errores desde el módulo de documentos.

### Condiciones de ejecución:

- Usuario autenticado.
- Al menos un documento subido previamente.
- Conexión a Internet estable.

### Entrada:

1. Acceder al apartado “Documentos”.
2. Seleccionar un archivo subido anteriormente.
3. Pulsar “Abrir documento”.

### Resultado esperado:

- El documento se visualiza correctamente dentro de la aplicación.
- El archivo mantiene su formato (PDF, imagen...).
- No aparecen mensajes de error ni problemas de carga.

### Evaluación de la prueba:

**Estado:** Satisfactoria



- **INSTANCIA N° 3: DESCARGAR AL ARCHIVO**

**Descripción del test:**

Comprobar si la aplicación permite descargar un documento almacenado en la nube al dispositivo local del usuario.

**Condiciones de ejecución:**

- Usuario autenticado.
- Documento disponible en la nube.
- Opción “Descargar” visible en la interfaz.

**Entrada:**

1. Acceder al módulo “Documentos”.
2. Seleccionar un archivo.
3. Pulsar el botón “Descargar”.

**Resultado esperado:**

- Se inicia la descarga del archivo y aparece en el almacenamiento del dispositivo.

**Resultado real:**

- El sistema solo permite visualizar el documento, pero NO descarga el archivo en el dispositivo.

**Evaluación de la prueba:**

**Estado:** No satisfactoria (Función no implementada)

La funcionalidad de descarga está pendiente de desarrollo. Actualmente solo se abre en modo visualización.

**\*ESTA FUNCIONALIDAD SE IMPLEMENTÓ POSTERIORMENTE CON SATISFACTORIO RESULTADO**

- **INSTANCIA N° 4: ELIMINAR ARCHIVO**

**Descripción del test:**

Validar que un usuario puede eliminar un documento almacenado en la nube y que este desaparece del sistema.

**Condiciones de ejecución:**

- Usuario autenticado.
- Documento previamente subido.
- Conexión a Internet.

**Entrada:**

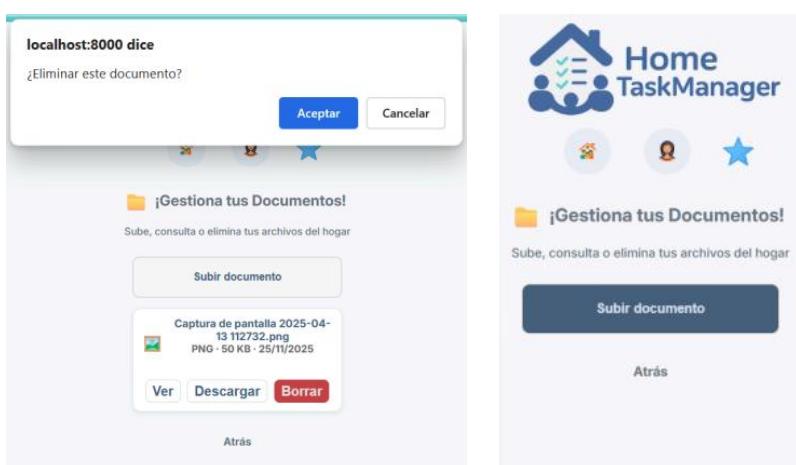
1. Acceder al módulo “Documentos”.
2. Seleccionar un archivo existente.
3. Pulsar el botón “Eliminar”.
4. Confirmar la acción cuando la app lo solicite.

**Resultado esperado:**

- El documento desaparece de la lista de archivos.
- El archivo se elimina en la nube.
- No se puede volver a abrir ni visualizar.

**Evaluación de la prueba:**

**Estado:** Satisfactoria



## 10. REFERENCIAS BIBLIOGRÁFICAS Y ENLACES

- Chaffey, D., & Ellis-Chadwick, F. (2019). Digital Marketing: Strategy, Implementation and Practice (7th ed.). Pearson.
- Fisher, R., Ury, W., & Patton, B. (2011). Getting to Yes: Negotiating Agreement Without Giving In. Penguin Books.
- García, A. (2019). El diagrama de Ishikawa: Un enfoque práctico para resolver problemas. Ediciones Innovación.
- Kotler, P., & Keller, K. L. (2016). Marketing Management (15th ed.). Pearson.
- López, C. (2021). Trabajo colaborativo en entornos virtuales. Plataforma de aprendizaje colaborativo.
- Osterwalder, A., & Pigneur, Y. (2010). Business Model Generation. John Wiley & Sons.
- Pérez, M. (2020). Organización del hogar y herramientas digitales. Revista Digital de Productividad.
- Project Management Institute (PMI). (2021). A Guide to the Project Management Body of Knowledge (PMBOK Guide) (7th ed.).
- Ries, A., & Trout, J. (2001). Positioning: The Battle for Your Mind. McGraw-Hill Education.
- Rodríguez, L. (2021). Desarrollo de aplicaciones móviles: De la idea a la implementación. TechBooks.
- Smith, J. (2022). Gestión del tiempo y productividad personal. Editorial Proyectos Eficientes.
- Tuckman, B. W. (1965). Developmental Sequence in Small Groups. Psychological Bulletin.
- Apple Inc. (2023). App Store Review Guidelines.
- Chaffey, D., & Ellis-Chadwick, F. (2019). Digital Marketing: Strategy, Implementation and Practice.
- Cozi. (2023). Cozi Family Organizer.

Google Play Console. (2023). App Naming Best Practices.

Kotler, P., & Keller, K. L. (2016). Marketing Management.

Nielsen, J. (2012). Usability 101: Introduction to Usability.

Notion. (s.f.). Guía para crear un dashboard de trabajo personal en Notion.

Osterwalder, A., & Pigneur, Y. (2010). Business Model Generation.

Project Management Institute (PMI). (2021). A Guide to the Project Management Body of Knowledge (PMBOK Guide).

Ries, A., & Trout, J. (2001). Positioning: The Battle for Your Mind.

Scrum.org. (2023). Scrum Guide.

Statista. (2023). Number of available apps in the Apple App Store from 2008 to 2023.

LinkedIn, InfoJobs, Tecnoempleo (Canales de RRHH).

<https://developer.apple.com/app-store/review/guidelines/>

<https://www.cozi.com/>

<https://www.decoora.com/6-aplicaciones-para-llevar-las-tareas-domesticas-al-dia/>

<https://designthinking.es/que-como-por-que/>

<https://www.educaciontrespuntocero.com/familias/apps-gratuitas-organizar-tareas-domesticas-familia/>

<https://support.google.com/googleplay/android-developer/answer/9866151>

<https://www.entrepreneur.com/article/350000>

<https://copilot.microsoft.com/chats/FewdHw2iuk7B1vjeXqx4e>

<https://miro.com/>

<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

<https://www.notion.com/es-es/help/guides/personal-work-dashboard>

<https://www.whatmatters.com/>

<https://www.scrum.org/resources/scrum-guide>

<https://www.mindtools.com/pages/article/smарт-goals.htm>

<https://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/>

<https://trello.com/>

<https://asana.com/es/resources/communication-plan>

<https://www.doofinder.com/es/blog/plan-de-comunicacion>

<https://grupo-pya.com/como-mejorar-la-capacidad-de-ejecucion/>

<https://blog.hubspot.es/marketing/crear-plan-comunicacion>

[https://www.plandemejora.com/como-elaborar-un-diagrama-pert-cpm/#google\\_vignette](https://www.plandemejora.com/como-elaborar-un-diagrama-pert-cpm/#google_vignette)

<https://ingenieroenproceso.com/kanban-un-ejemplo-practico/>

<https://www.productiviza.com/scrum-para-ninos/>

<https://millennialsconsulting.es/trabajo-en-equipo-en-una-empresa/>

<https://www.ingeniovirtual.com/tipos-de-graficos-y-diagramas-para-la-visualizacion-de-datos/>

<https://www.salesforce.com/mx/blog/okr/>

<https://cards.algoreducation.com/app/home/base>

<https://chatgpt.com/>

<https://www.canva.com/templates/?query=gantt>

<https://img.freepik.com/>

<https://copilot.microsoft.com/chats/>

<https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/>

[https://www.initiumsoft.com/blog\\_initium/programacion-web/](https://www.initiumsoft.com/blog_initium/programacion-web/)

<https://innevo.com/blog/metodologia-scrum>

<https://winred.es/management/metodologia-scrum-que-es/gmx-niv116-con24594.htm>

<https://github.com/>

[https://www.ganttproject.biz/#google\\_vignette](https://www.ganttproject.biz/#google_vignette)

<https://es.software.fish/project-management-software/best-project-management-software>

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