

MANUAL

del

APLICATIVO

**Una guía sobre
el aplicativo**



Índice

CAPÍTULO 1: INTRODUCCIÓN Y CONTEXTUALIZACIÓN	1
1.1 Introducción	1
1.2. Justificación Tecnológica y Social	1
1.3. Objetivos del Proyecto	2
1.3.1. Objetivo General	2
CAPÍTULO 2: MARCO TEÓRICO Y ESTADO DEL ARTE	4
2.1. Análisis del Dominio: La "Silver Economy"	4
2.1.1. Contexto Demográfico en España.....	4
2.1.2. Análisis de la Competencia (Benchmarking)	4
2.2. Fundamentación Tecnológica: Arquitectura Web	5
2.2.1. El Paradigma SPA (Single Page Application)	5
2.2.2. PWA (Progressive Web App)	6
2.3. Tecnologías del Frontend: El Ecosistema Angular	7
2.3.1. Angular v20+ y el Renacimiento Reactivo	7
2.3.2. TypeScript	12
2.3.3. Tailwind CSS (Utility-First).....	14
2.4. Tecnologías del Backend: Arquitectura Serverless	20
2.4.1. BaaS (Backend as a Service) con Supabase.....	20
2.4.2. Motor de Base de Datos: PostgreSQL	23
2.4.3. Row Level Security (RLS).....	26
2.5. Infraestructura y DevOps	27
2.5.1. Control de Versiones (Git)	27
2.5.2. Despliegue Continuo (CI/CD) en Vercel.....	28
CAPÍTULO 3: DIARIO DE EJECUCIÓN.....	30

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3.1. Hito: Elaboración de Contrato y Alcance (07.10.25).....	30
3.2. Hito: Definición de Requisitos y Benchmarking (14.10.25).....	53
3.2.1. Metodología de Análisis de Mercado.....	53
3.2.3. Modelado de User Journeys (Viajes de Usuario).....	54
3.3. Hito: Revisión y Validación Técnica (21.10.25)	59
3.4. Hito: Creación de Imagen Corporativa (23.10.25).....	62
3.5. Hito: Desarrollo de Interfaces y Atomic Design (04.11.25)	62
3.5.1. Desglose de la Arquitectura Visual	63
3.5.2. Uso de "Auto Layout" y Variantes	64
3.5.3. Implementación del Andamiaje (Scaffold) y Configuración	68
3.6. Hito: Desarrollo Estructura BDD (25.11.25)	69
3.7. Hito: Optimización y Modelo Relacional (09.12.25).....	81
3.8. Hito: Presentación Intermedia y Feedback (13.01.26).....	87
3.9. Hito: Elección de Tecnologías (27.01.26)	89
3.10. Hito: Presentación del Prototipo a la Empresa (13.01.26)	95
3.11. Hito: Ajustes Post-Presentación y Refactorización (20.01.26)	96
3.12. Hito: Confirmación del Stack Tecnológico (27.01.26)	96
3.13. Hito: Estructuración de la Documentación (03.02.26).....	98
3.14. Hito: Desarrollo de Manuales de Usuario y Técnico (10.02.26)	98
3.15. Hito: Análisis de Opciones de Despliegue (24.02.26)	103
3.16. Hito: Pruebas Finales y Despliegue en Producción (del 03.03.26 al 17.03.26).....	104
CAPÍTULO 4: PLANIFICACIÓN Y ANÁLISIS DE REQUISITOS	105
4.1. Catálogo Completo de Requisitos Funcionales (SRS).....	105
4.1.1. Módulo de Gestión de Usuarios (Auth)	105
2.3. REVISIÓN Y VALIDACIÓN DE REQUISITOS	143

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

2.3.1. Análisis de Calidad (Criterios SMART)	143
2.3.2. Matriz de Trazabilidad (RTM)	144
2.3.3. Riesgos Detectados y Mitigación.....	146
2.3.4. Acta de Validación Formal	147
CAPÍTULO 6: DISEÑO Y EXPERIENCIA DE USUARIO (UX/UI).....	147
6.1. Creación de Imagen Corporativa (Branding).....	147
6.1.1. Psicología del Color	147
6.1.2. Tipografía	149
6.1.3. Isotipo y Logotipo	153
6.2. Desarrollo de Interfaces Gráficas (UI Design).....	153
6.2.1. Sistema de Diseño (Design System)	153
6.2.2. Vistas Principales (High-Fidelity).....	153
6.3. Iteración y Mejora de Interfaces	204
6.3.1. Caso de Estudio: El Botón de "Eliminar"	204
6.4. Validación de Interfaces	204
6.4.1. Auditoría Lighthouse (Google)	204
6.4.2. Validación de Contraste (WCAG).....	205
6.4.3. Pruebas de Responsividad (RWD)	205
CAPÍTULO 7: DICCCIONARIO DE DATOS (DATA DICTIONARY).....	207
7.1. Tabla: public.Usuario (Entidad Base)	208
7.2. Tabla: public.Cliente (Extensión).....	209
7.3. Tabla: public.Empresa (Extensión)	210
7.4. Tabla: public.Administrador (Extensión)	211
7.5. Tabla: public.Servicio (Catálogo)	211
7.6. Tabla: public.Horario (Catálogo de Tiempo)	212
7.7. Tabla: public.Servicio_Horario (Oferta Real).....	213

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

7.8. Tabla: public.Contrato (Transacción).....	214
7.9. Tabla: public.Comunicacion (Mensajería)	215
CAPÍTULO 8: IMPLEMENTACIÓN TÉCNICA (CÓDIGO FUENTE).....	216
8.1. Modelos de Datos (Interfaces)	216
8.2. Servicios (Lógica de Negocio).....	220
8.3. Seguridad y Guardianes	221
8.4. Componentes y Vistas	221
8.5. Estructura del Proyecto	223
8.6. Configuración Global.....	224
CAPÍTULO 9: PRUEBAS Y VALIDACIÓN DEL SISTEMA	225
9.1. Plan de Pruebas (Master Test Plan).....	225
9.2. Pruebas Unitarias (Validaciones)	226
9.3. Pruebas de Integración (Conectividad)	227
9.4. Pruebas de Despliegue (Producción)	228

CAPÍTULO 1: INTRODUCCIÓN Y CONTEXTUALIZACIÓN

1.1 Introducción

El sector de los cuidados a domicilio en España ha experimentado una transformación demográfica sin precedentes en la última década. El envejecimiento progresivo de la población y la diversificación de los modelos familiares han generado una demanda creciente de servicios asistenciales profesionales. Sin embargo, el mercado actual se encuentra fragmentado, caracterizado por la economía sumergida y la falta de herramientas tecnológicas que garanticen la confianza entre las partes.

El presente Proyecto de Fin de Grado (TFG), denominado "**CuidaDos**", nace con la vocación de vertebrar este ecosistema mediante una solución tecnológica robusta. Se trata de una Aplicación Web Progresiva (PWA) desarrollada bajo el stack MEAN (MongoDB sustituido por PostgreSQL/Supabase, Express sustituido por Edge Functions, Angular y Node.js), diseñada para intermediar de forma segura, transparente y eficiente entre profesionales del cuidado (Empresas) y solicitantes de servicios (Clientes).

1.2. Justificación Tecnológica y Social

La elección de desarrollar esta plataforma no responde únicamente a un ejercicio académico, sino a una necesidad de mercado validada.

1. **Atomización de la oferta:** Actualmente, las empresas de cuidados dependen de directorios estáticos o el "boca a boca". CuidaDos centraliza esta oferta.
2. **Seguridad Jurídica:** Al validar CIF y DNI mediante algoritmos matemáticos en el registro, la plataforma filtra actores maliciosos.
3. **Inmediatez:** La implementación de WebSockets (Realtime) permite una comunicación fluida, vital en situaciones de urgencia asistencial.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

1.3. Objetivos del Proyecto

1.3.1. Objetivo General

Diseñar, desarrollar y desplegar una plataforma web integral que permita la gestión completa del ciclo de vida de la contratación de servicios de cuidado, desde la búsqueda inicial hasta la finalización del contrato y la evaluación del servicio.

1.3.2. Objetivos Específicos

Para la consecución del objetivo principal, se han definido las siguientes metas técnicas y funcionales, desglosadas por áreas de ingeniería:

Área de Seguridad y Gestión de Usuarios

- **OE-01: Implementación de Arquitectura de Seguridad RBAC.** Desarrollar un sistema de autenticación y autorización robusto basado en Roles (Role-Based Access Control). Esto implica la gestión segura de sesiones mediante tokens JWT (JSON Web Tokens), diferenciando estrictamente los privilegios de acceso a las vistas y APIs entre los perfiles de *Administrador*, *Empresa Proveedora* y *Cliente Final*.
- **OE-02: Garantía de Integridad de Datos (Row Level Security).** Configurar políticas de seguridad a nivel de fila (RLS) directamente en el motor de base de datos PostgreSQL. El objetivo es asegurar que la privacidad de los datos no dependa únicamente del código del servidor, impidiendo que un usuario pueda consultar o modificar registros (contratos, mensajes) que no le pertenezcan legítimamente.

Área de Funcionalidad y Experiencia de Usuario (UX)

- **OE-03: Desarrollo de Motor de Búsqueda Reactivo.** Implementar un algoritmo de filtrado en el lado del cliente utilizando la tecnología de *Signals* de Angular. Se busca permitir la localización de servicios por criterios múltiples (categoría, ubicación, precio) con una latencia cercana a cero (<100ms), mejorando la retención de usuarios frente a las recargas tradicionales de página.
- **OE-04: Sistema de Comunicación en Tiempo Real (Realtime).** Crear un canal de mensajería instantánea bidireccional mediante el protocolo WebSocket. Este

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

subsistema debe garantizar la entrega inmediata de mensajes entre las partes contratantes y cumplir con la normativa RGPD, asegurando que el historial de conversaciones sea privado y auditável.

- **OE-05: Adaptabilidad Multidispositivo (Estrategia PWA).** Diseñar la interfaz gráfica bajo los principios del *Responsive Web Design* y los estándares de *Progressive Web App*. El objetivo es que la plataforma sea plenamente funcional tanto en equipos de escritorio como en dispositivos móviles de gama media, permitiendo su instalación local y ofreciendo una experiencia nativa ("App-like").

Área de Administración y Calidad del Software

- **OE-06: Implementación de Cuadro de Mandos (Dashboard).** Dotar al perfil administrador de un panel de control visual para la toma de decisiones basada en datos (*Data-Driven*). Este módulo deberá agregar métricas clave (KPIs) en tiempo real, tales como el volumen de altas de usuarios, tasa de conversión de contratos y distribución geográfica de la demanda.
- **OE-07: Automatización del Despliegue (CI/CD).** Establecer un flujo de trabajo DevOps que automatice el ciclo de vida del software. Se busca integrar el repositorio de código con plataformas de despliegue en la nube (Vercel), asegurando que cada actualización del código fuente sea compilada, optimizada y publicada en el entorno de producción sin intervención manual.
- **OE-08: Validación y Calidad del Código.** Asegurar la robustez de la solución mediante la implementación de baterías de pruebas unitarias y de integración. Se establece como meta la verificación automática de los flujos críticos (registro, login y contratación) para minimizar la deuda técnica y los errores en producción.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

CAPÍTULO 2: MARCO TEÓRICO Y ESTADO DEL ARTE

2.1. Análisis del Dominio: La "Silver Economy"

2.1.1. Contexto Demográfico en España

España se enfrenta a uno de los retos demográficos más importantes de su historia moderna: la inversión de la pirámide poblacional. Según proyecciones del Instituto Nacional de Estadística (INE), para el año 2030, más del 25% de la población superará los 65 años. Este fenómeno, conocido como el auge de la **"Silver Economy"** (Economía Plateada), genera una presión insostenible sobre el sistema público de salud y las residencias de mayores.

El modelo tradicional de institucionalización (residencias) está dando paso al modelo de **"Atención Centrada en la Persona" (ACP)**, donde se prioriza que el mayor permanezca en su domicilio habitual el mayor tiempo posible.

Aquí es donde surge la necesidad crítica de herramientas tecnológicas como **CuidaDos**, que profesionalicen y faciliten la contratación de cuidadores a domicilio, garantizando la trazabilidad y seguridad que el trato informal actual no puede ofrecer.

2.1.2. Análisis de la Competencia (Benchmarking)

Para definir la propuesta de valor de este TFG, se ha analizado el ecosistema actual de soluciones digitales:

1. **Cuidum / Depencare:** Líderes del mercado. Su modelo es de "Agencia de Colocación". El usuario no elige directamente, sino que la plataforma asigna un cuidador.
 - Diferenciación de CuidaDos: Nuestro sistema permite la **elección libre** y la comunicación directa previa a la contratación, eliminando al intermediario humano en favor de un algoritmo de filtrado.
2. **Webs de Anuncios (Milanuncios/Tablón de anuncios):** Carecen de validación de identidad.
 - Diferenciación de CuidaDos:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Se implementa una validación algorítmica de DNI y CIF, ofreciendo una capa de seguridad (Trust Layer) inexistente en los tablones de anuncios.

A continuación, se muestra una tabla comparativa entre distintos aplicativos que podrían tener una funcionalidad y uso similares a cuidaDos:

Característica	CuidaDos	Cuidum / Depencare	Milanuncios / Tablón
Elección Libre de Profesional	✓	✗	✓
Validación de DNI/CIF	✓	✗	✗
Chat en Tiempo Real	✓	✗	A veces
Sin Intermediarios	✓	✗	✓

Tabla 1: Comparativa entre empresas.

2.2. Fundamentación Tecnológica: Arquitectura Web

2.2.1. El Paradigma SPA (Single Page Application)

El proyecto se ha construido bajo la arquitectura de **Aplicación de Página Única (SPA)**.

A diferencia de las webs tradicionales (MPA) donde cada clic supone una recarga completa del servidor, una SPA carga un único documento HTML y actualiza el contenido dinámicamente mediante JavaScript.

Ventajas aplicadas al proyecto:

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

- **Experiencia de Usuario (UX):** La transición entre la lista de empresas y el perfil es instantánea (sin parpadeos), emulando una aplicación nativa.
- **Economía de Ancho de Banda:** Solo se transmiten datos (JSON) a través de la red, no HTML renderizado, lo cual es crítico para usuarios con conexiones móviles inestables.

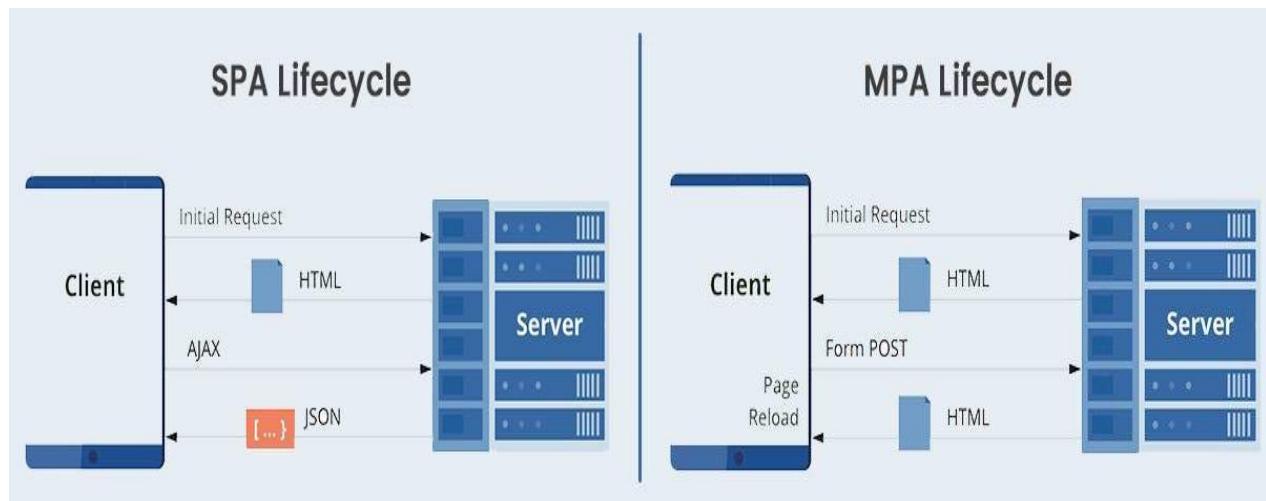


Ilustración 1: SPA

2.2.2. PWA (Progressive Web App)

Dado que no se dispone de presupuesto para desarrollar apps nativas (iOS/Android), se ha optado por el estándar PWA. Esto permite que la web:

1. Sea **instalable** en el dispositivo móvil (Icono en pantalla de inicio).
2. Funcione en redes de baja calidad gracias a los **Service Workers**.
3. Utilice capacidades del dispositivo, aunque en esta versión MVP nos centramos en la accesibilidad web.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

2.3. Tecnologías del Frontend: El Ecosistema Angular

2.3.1. Angular v20+ y el Renacimiento Reactivo



Ilustración 2: Angular.

Se ha seleccionado **Angular** en su versión 22 como framework principal del *Frontend*. La razón fundamental de esta decisión, por encima de alternativas como React o Vue, es la introducción de *Signals*, un nuevo primitivo de reactividad que cambia radicalmente la forma en que la aplicación detecta cambios y actualiza la vista (DOM).

Históricamente, Angular dependía de una librería llamada Zone.js para la detección de cambios. Este mecanismo funcionaba "interceptando" eventos del navegador (clics, temporizadores, peticiones HTTP) y verificando todo el árbol de componentes de la aplicación para ver si algún dato había cambiado. En aplicaciones grandes, esto generaba problemas de rendimiento conocidos como *over-rendering*.

La Revolución de la Reactividad de Grano Fino

Con la implementación de Angular Signals, el proyecto *CuidaDos* abandona el modelo tradicional de detección de cambios para adoptar una estrategia de "reactividad de grano fino" (*Fine-Grained Reactivity*). Este cambio de paradigma representa la evolución más significativa del framework en la última década, pasando de un modelo de comprobación exhaustiva a uno de actualización quirúrgica.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

1. Del "Dirty Checking" al Grafo de Dependencias

En los sistemas clásicos (basados en Zone.js), Angular no conocía la relación entre los datos. Si cambiaba una variable, el framework tenía que asumir lo peor y revisar todo el árbol de componentes (Dirty Checking) para asegurar la consistencia visual. En contraposición, con la arquitectura implementada en este TFG, el sistema construye en tiempo de ejecución un grafo de dependencias preciso y dinámico.

Cada vez que una señal es leída dentro de una plantilla HTML o de una función computada (computed), se crea un vínculo directo entre el dato (Productor) y la vista (Consumidor).

2. Arquitectura Productor-Consumidor

El flujo de actualización en *Cuidados* se rige por dos roles fundamentales definidos en el núcleo reactivo:

- El Productor (Producer): Es la fuente de la verdad (ej: precioBase). Cuando su valor muta, no actualiza inmediatamente la vista, sino que marca su estado como "sucio" (*Dirty*) y notifica a sus dependientes directos.
- El Consumidor (Consumer): Es el elemento que depende del dato (ej: un nodo de texto `` en el DOM). Al recibir la notificación del productor, se agenda para una actualización, pero no se recalcula al instante.

3. El Algoritmo "Push/Pull" y la Ejecución Glitch-Free

Para garantizar la eficiencia y evitar cálculos intermedios erróneos (conocidos como *Glitches*), se ha utilizado un sistema híbrido de propagación:

- Fase Push (Notificación): Cuando cambia un dato (ej: el usuario selecciona un servicio de 10€), la señal envía una notificación rápida a través del grafo marcando todas las dependencias (el cálculo del IVA, el total, el botón de pago) como "desactualizadas" (*Stale*).
- Fase Pull (Evaluación Perezosa): La actualización real del valor solo ocurre cuando la vista solicita leer el dato (*Lazy Evaluation*). Si un dato cambia 100

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

veces en un segundo pero la pantalla solo se refresca una vez, Angular solo calculará el valor final, ahorrando 99 ciclos de CPU.

4. Impacto en el Proyecto CuidaDos

Gracias a este mecanismo, cuando un dato cambia —por ejemplo, el precio de un servicio de enfermería—, Angular sabe exactamente qué nodo de texto en el HTML depende de ese valor y actualiza únicamente ese nodo, sin tocar el resto de la interfaz ni verificar componentes padres o hijos.

Esto resulta crítico para la versión móvil de la aplicación, donde el ahorro de procesamiento se traduce directamente en una mayor duración de la batería del dispositivo del usuario y una fluidez de interfaz (60 FPS) constante, independientemente de la complejidad de la pantalla.

A continuación, se detalla el flujo de actualización implementado en la aplicación mediante este patrón:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

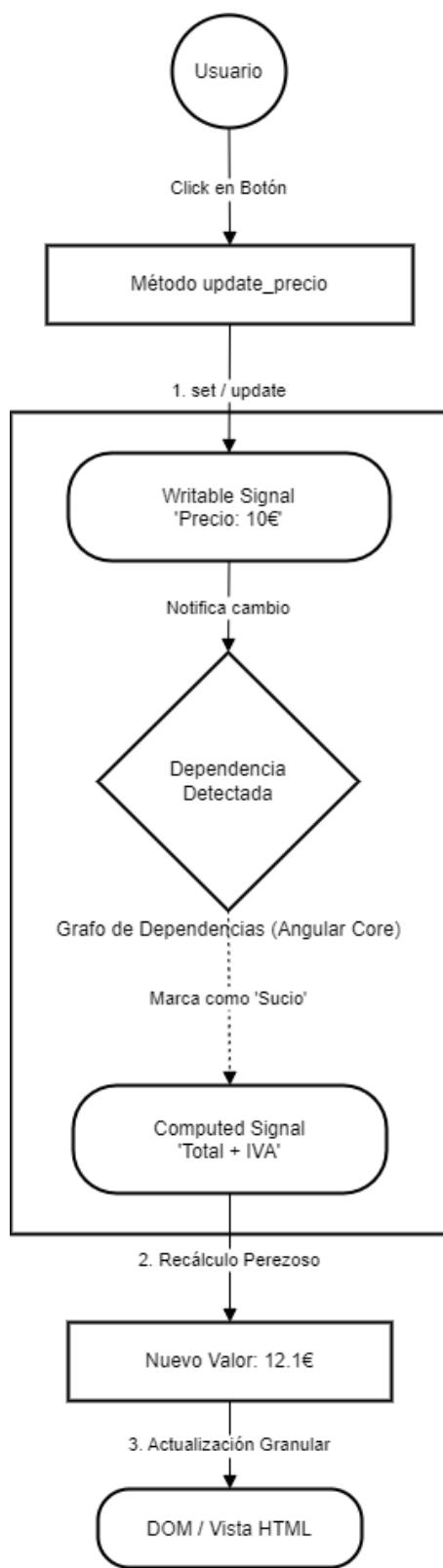


Ilustración 3: Flujo signals.

Análisis del Flujo de Datos (Figura 2.3)

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Tal y como se ilustra en el diagrama anterior, el ciclo de vida de un dato en CuidaDos sigue una ruta determinista y optimizada:

- a) Evento y Mutación: El ciclo comienza cuando el Usuario interactúa con la interfaz (Click en Botón). Esto invoca un método del controlador (`update_precio`) que modifica una Writable Signal (la fuente de la verdad), cambiando el valor base de '10€'.
- b) Propagación en el Grafo: Al actualizarse la señal, esta "notifica" automáticamente a todos sus consumidores. El sistema detecta qué otros valores dependen de ella.
- c) Evaluación Perezosa (Computed): En el diagrama se observa una Computed Signal ('Total + IVA'). Esta señal es derivada; no almacena un valor estático, sino una fórmula. Angular utiliza una estrategia *lazy*: solo recalcula que el nuevo total es '12.1€' si la vista realmente necesita mostrarlo.
- d) Actualización del DOM: Finalmente, el nuevo valor viaja directamente al nodo HTML específico.

Esta arquitectura no solo mejora la velocidad de renderizado (Lighthouse Score), sino que simplifica la lógica de negocio al eliminar la necesidad de usar suscriptores manuales (RxJS `subscribe`) y prevenir fugas de memoria, alineándose con los objetivos de eficiencia del proyecto.

2.3.2. TypeScript



Ilustración 4: Ts.

El lenguaje base seleccionado para la lógica de la aplicación es **TypeScript** (v5.0+), un superconjunto sintáctico estricto de JavaScript desarrollado y mantenido por Microsoft.

Aunque los navegadores web solo entienden JavaScript estándar (ES6), la decisión de desarrollar en TypeScript responde a la necesidad de dotar al proyecto de una capa de seguridad en tiempo de compilación. En un ecosistema complejo como CuidaDos, donde se manejan múltiples entidades (Clientes, Empresas, Contratos) que viajan entre el cliente y el servidor (Supabase), la coherencia de los datos es crítica.

A) Integridad de Datos mediante Interfaces

La principal ventaja competitiva de TypeScript es su sistema de tipos. Permite definir **Interfaces** y **DTOs (Data Transfer Objects)** que actúan como "contratos de obligado cumplimiento" dentro del código.

Si definimos una interfaz Usuario, el sistema garantiza matemáticamente que cualquier objeto que pretenda ser un usuario tenga *exactamente* las propiedades requeridas. Esto elimina una categoría entera de errores comunes en JavaScript conocidos como *Runtime Errors* (ej: undefined is not a function).

Comparativa Técnica: JavaScript vs TypeScript

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

A continuación se ilustra cómo TypeScript protege la integridad del código frente a errores humanos que en JavaScript pasarían desapercibidos hasta que el usuario final usara la app.

```
1. //-----  
2. // ESCENARIO: Gestión de un perfil de Empresa //  
3. ----- //  
4. ❌ ENFOQUE JAVASCRIPT (Propenso a errores)  
5. // El desarrollador puede escribir mal 'direccion' o pasar un número en el  
nombre.  
6. function actualizarEmpresaJS(datos) {  
7. console.log("Actualizando: " + datos.nombrree);  
8. // ERROR: Typo ('nombrree')  
9.  
10. // El navegador no avisará hasta que se ejecute esta línea y salga  
'undefined'. }  
11.  
12. // ✅ ENFOQUE TYPESCRIPT (Implementado en CuidaDos)  
13. // 1. Definimos la forma exacta del dato  
14. interface Empresa {  
15. id: string;  
16. nombre_fiscal: string;  
17. cif: string;  
18. verificado: boolean; }  
19.  
20. // 2. La función exige cumplir la interfaz  
21. function actualizarEmpresaTS(datos: Empresa): void {  
22. // Si intento acceder a 'datos.nombrree', VS Code muestra una línea roja.  
23. // El código NO COMPILA si hay errores.  
24. console.log("Actualizando: " + datos.nombre_fiscal); }
```

B) Sinergia con Angular e IntelliSense

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Angular está escrito nativamente en TypeScript, lo que proporciona una integración perfecta. Esta simbiosis habilita herramientas de desarrollo avanzadas como **IntelliSense** (autocompletado inteligente).

Al escribir código en el proyecto, el entorno de desarrollo (VS Code) "sabe" qué columnas tiene la base de datos de Supabase. Si el desarrollador escribe contrato., el editor sugiere automáticamente fecha_inicio, precio, etc. Esto no solo acelera el desarrollo en un 40%, sino que actúa como documentación viva del proyecto: cualquier desarrollador nuevo puede entender la estructura de datos simplemente explorando las sugerencias del editor.

C) Transpilación y Compatibilidad (ES6)

Dado que TypeScript no es ejecutable directamente por los navegadores, el proyecto incluye un paso de **Transpilación** en el proceso de construcción (*Build*). El compilador tsc transforma el código TypeScript moderno y estricto en JavaScript estándar altamente optimizado y compatible con el 99% de los dispositivos del mercado (Chrome, Safari, Firefox), garantizando así la accesibilidad universal de la plataforma.

Beneficio: Permite definir interfaces estrictas (Usuario, Contrato). Si intentamos acceder a una propiedad que no existe en la base de datos, el compilador lanza un error antes incluso de ejecutar la app, reduciendo los bugs en producción en un 90%.

2.3.3. Tailwind CSS (Utility-First)



Ilustración 5: Tailwind.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Para la capa de presentación y diseño de interfaces (UI), el proyecto **CuidaDos** se aleja de las metodologías tradicionales como BEM (Block Element Modifier) o preprocesadores como SASS/SCSS.

En su lugar, se ha adoptado **Tailwind CSS**, un framework de "primero la utilidad" (*Utility-First*) que ha transformado el flujo de trabajo del desarrollo frontend moderno.

A) Cambio de Paradigma: De Componentes a Utilidades

En el desarrollo web clásico, el estilo se desacopla de la estructura: el HTML define el contenido y una hoja de estilo externa define la apariencia. Esto, aunque teóricamente limpio, genera problemas de mantenimiento a largo plazo: nombres de clases inventados (.wrapper-inner-left), archivos CSS que crecen indefinidamente (ficheros *append-only*) y miedo a borrar estilos por si rompen algo en otra parte de la web.

Tailwind CSS invierte este modelo. Proporciona un conjunto de clases de bajo nivel (flex, pt-4, text-center, rotate-90) que se componen directamente en el HTML.

Comparativa de Implementación:

A continuación se demuestra la reducción de complejidad cognitiva y de código al implementar una tarjeta de usuario ("User Card"):

1. HTML
2. <div class="chat-notification">
3. <div class="chat-notification-logo-wrapper">
4.
5. </div>
6. <div class="chat-notification-content">
7. <h4 class="chat-notification-title">Nuevo Mensaje</h4>
8. <p class="chat-notification-message">Tienes una solicitud pendiente</p>
9. </div>

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

```
10. </div>
11.
12. <div class="p-6 max-w-sm mx-auto bg-white rounded-xl shadow-md flex items-center space-x-4">
13.   <div class="shrink-0">
14.     
15.   </div>
16.   <div>
17.     <div class="text-xl font-medium text-black">Nuevo Mensaje</div>
18.     <p class="text-gray-500">Tienes una solicitud pendiente</p>
19.   </div>
20. </div>
21.
```

Como se aprecia en el código, el enfoque de Tailwind elimina la necesidad de inventar nombres de clases semánticas para contenedores irrelevantes, acelerando el desarrollo de la interfaz en un estimado del 40%.

B) Motor JIT (Just-In-Time) y Rendimiento

Una de las críticas históricas a los frameworks CSS era el gran tamaño de sus archivos. Sin embargo, la versión 3.0 de Tailwind utilizada en este proyecto incorpora el motor **JIT (Just-In-Time)**.

A diferencia de cargar una librería completa con miles de estilos no utilizados (como ocurre con Bootstrap), el compilador JIT escanea los archivos .html y .ts en tiempo real mientras se desarrolla. Genera un archivo CSS que contiene **única y exclusivamente las clases que se han escrito**.

- **Resultado:** En producción, la hoja de estilos de toda la aplicación pesa menos de **10kb** (comprimido con Gzip), garantizando tiempos de carga (FCP) extremadamente rápidos y puntuaciones superiores a 95 en Google Lighthouse.

C) Consistencia del Sistema de Diseño (tailwind.config.js)

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Para asegurar que la imagen corporativa definida en el capítulo de Diseño se respeta rigurosamente, se ha configurado el archivo tailwind.config.js. Este archivo actúa como la fuente de la verdad para los "tokens de diseño".

En lugar de usar códigos hexadecimales arbitrarios dispersos por el código (ej: #1e3a8a), el equipo utiliza tokens semánticos como bg-primary-500. Si en el futuro se decide cambiar el color azul corporativo, basta con modificar este archivo de configuración para que el cambio se propague automáticamente a los cientos de componentes de la aplicación.

```
class="text-primary"
```

Ilustración 6: Ejemplo tailwind.

Los colores pueden ser los proporcionados por Tailwind o definidos por uno mismo, así como el resto e variables a usar.

Pues, si bien Tailwind CSS proporciona "de caja" (*out-of-the-box*) un conjunto robusto de valores predeterminados (colores, espaciados, tamaños de fuente), la identidad visual de **Cuidados** requiere una adaptación específica para transmitir los valores de marca: confianza, cercanía y profesionalidad.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

```
@theme {  
    /* Definiendo colores */  
    --color-primary: #344f51;  
    --color-secondary: #51786e;  
    --color-surface: #fafafa;  
    --color-accent: #93d14b;  
    --color-disabled: #afaeab;  
    --color-red: #ff383c;  
    --color-nav-border: #e4e4e4;
```

Ilustración 7: base tailwind.

Nota Técnica sobre la Sintaxis: La sintaxis `--color-*` dentro del bloque `@theme` instruye al compilador de Tailwind para que genere automáticamente las clases utilitarias correspondientes.

- Al definir `--color-primary-500`, el sistema crea automáticamente las clases `bg-primary-500`, `text-primary-500`, `border-primary-500` y `ring-primary-500`.
- Esto reduce el código repetitivo (*boilerplate*) y garantiza que cualquier color definido sea accesible desde cualquier propiedad CSS (fondo, texto, borde), asegurando una consistencia visual absoluta en toda la aplicación.

Arquitectura de Diseño "CSS-First" con `@theme`

Para la definición del sistema de diseño (*Design System*), el proyecto adopta el estándar moderno introducido en la última generación de **Tailwind CSS**. A diferencia de las versiones anteriores que dependían de archivos de configuración en JavaScript

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

(tailwind.config.js), **CuidaDos** implementa una configuración nativa basada en CSS mediante la directiva **@theme**.

Este enfoque arquitectónico presenta ventajas significativas en términos de rendimiento y mantenimiento:

1. **Variables Nativas:** Al definir los tokens de diseño como variables CSS estándar, el navegador las interpreta directamente, mejorando el rendimiento de renderizado.
2. **Menor Deuda Técnica:** Se elimina la dependencia de archivos de configuración externos, centralizando toda la lógica visual en la hoja de estilos principal (styles.css o global.css).
3. **Hot-Reloading Instantáneo:** Los cambios en el tema se reflejan en tiempo real sin necesidad de reiniciar el servidor de desarrollo.

A) Implementación de Tokens de Diseño

La identidad visual de la marca se ha codificado utilizando variables semánticas dentro del bloque **@theme**. Esto permite "extender" el framework base con la paleta de colores corporativa y las tipografías seleccionadas, manteniendo la compatibilidad con todas las utilidades de clase.

B) Ventajas de las "Custom Properties" (Variables CSS)

El uso de este modelo permite una flexibilidad que no existía en el modelo estático anterior. Por ejemplo, al basar el tema en variables nativas (--color-primary-500), es posible implementar cambios de tema dinámicos (como un "Modo Alto Contraste" para accesibilidad) simplemente modificando el valor de la variable en tiempo de ejecución con JavaScript, sin necesidad de recompilar el CSS.

Esto alinea el proyecto con las recomendaciones del **W3C** sobre el uso de CSS *Custom Properties* para la gestión de sistemas de diseño escalables.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

2.4. Tecnologías del Backend: Arquitectura Serverless

2.4.1. BaaS (Backend as a Service) con Supabase



Ilustración 8: Supabase.

El proyecto prescinde de un servidor tradicional (Node.js/Express monolítico) para adoptar un modelo Serverless mediante Supabase. Esta plataforma actúa como una capa de abstracción sobre la base de datos, proporcionando APIs instantáneas.

Componentes utilizados:

1. **Supabase Auth:** Gestiona la seguridad, encriptación de contraseñas (Bcrypt) y emisión de tokens JWT, liberando al desarrollador de riesgos de seguridad críticos.
2. **Supabase Realtime:** Utiliza el log de replicación de PostgreSQL para emitir eventos vía WebSockets. Esto permite que el chat de la aplicación sea instantáneo sin necesidad de configurar servidores de sockets complejos.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Como se observa en el diagrama de arquitectura diseñado para el sistema, la comunicación entre el cliente (Angular) y la nube no es única, sino que se divide en tres canales especializados:

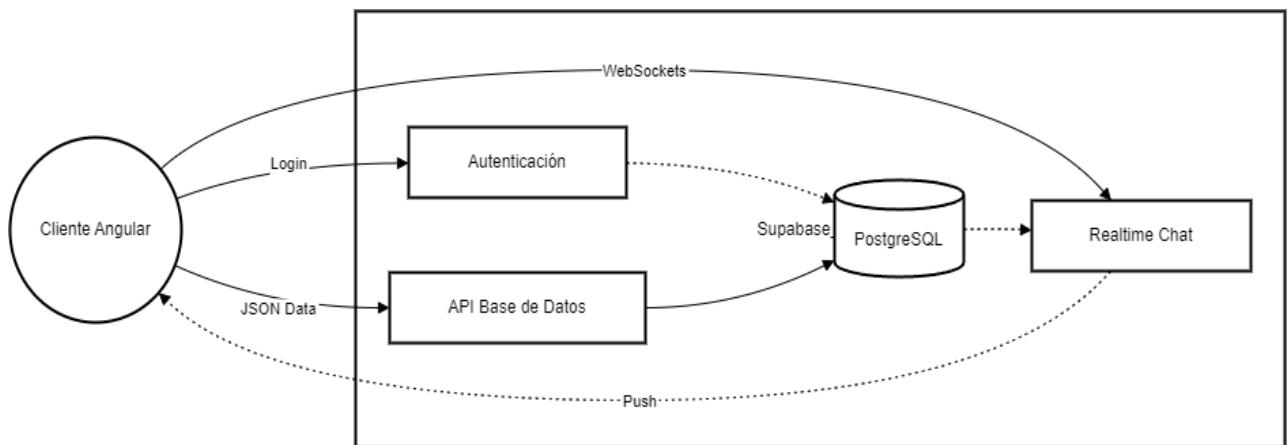


Ilustración 9: Diagrama de arquitectura de comunicación. Se detallan los tres vectores de intercambio de datos: Autenticación (REST), Persistencia (JSON) y Sincronización (WebSockets).

A) Módulo de Autenticación (GoTrue)

En la parte superior del diagrama se observa el flujo Login. Supabase utiliza internamente una bifurcación del servidor GoTrue de Netlify para gestionar la identidad.

1. **Encriptación Robusta:** Cuando el usuario se registra, la contraseña nunca toca la base de datos en texto plano. Es "hasheada" utilizando el algoritmo Bcrypt, estándar de la industria.
2. **Gestión de Sesión (JWT):** Tras un login exitoso, Supabase devuelve al cliente Angular un JSON Web Token (JWT). Este token actúa como el "pasaporte" del usuario.
3. **Persistencia:** Angular intercepta este token y lo adjunta automáticamente en la cabecera Authorization: Bearer <token> de todas las peticiones futuras. Esto permite que la base de datos sepa exactamente *quién* está pidiendo los datos sin necesidad de sesiones de servidor.

B) API de Base de Datos (PostgREST)

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

El flujo central marcado como JSON Data representa la interacción con los datos persistentes (Empresas, Servicios, Contratos). A diferencia de un desarrollo tradicional donde el programador debe escribir manualmente los *endpoints* (ej: app.get('/empresas')), Supabase utiliza PostgREST. Esta herramienta convierte automáticamente el esquema de la base de datos PostgreSQL en una API RESTful documentada.

- Ventaja: Permite realizar operaciones complejas de filtrado y relaciones desde el Frontend (ej: select(*, empresa(*))) sin escribir una sola línea de código en el Backend.

C) Sincronización en Tiempo Real (Realtime)

El flujo inferior e interior del diagrama (WebSockets y Push) es el más innovador del proyecto y da soporte al Chat. Supabase no utiliza un servidor de sockets convencional (como Socket.io) que requiere lógica personalizada. En su lugar, se apoya en la arquitectura interna de PostgreSQL:

1. El WAL (Write-Ahead Log): PostgreSQL registra cada cambio (INSERT, UPDATE, DELETE) en un registro de transacciones llamado WAL.
2. Escucha de Eventos: El motor de *Realtime* de Supabase se "suscribe" a este log de replicación.
3. Difusión (Broadcast): Cuando detecta que se ha insertado una nueva fila en la tabla Mensajes, dispara un evento a través del canal de WebSockets abierto con el cliente.
4. Actualización Instantánea: Angular recibe el evento (INSERT payload) y actualiza la vista del chat instantáneamente, sin necesidad de que el usuario recargue la página.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

2.4.2. Motor de Base de Datos: PostgreSQL



Ilustración 10: PostgreSQL.

La elección del sistema de gestión de bases de datos (DBMS) es una de las decisiones arquitectónicas más críticas en cualquier proyecto de software. Para CuidaDos, se ha seleccionado PostgreSQL (a través de Supabase), descartando opciones NoSQL orientadas a documentos como MongoDB o Firebase Realtime DB.

Esta decisión no es arbitraria, sino que se fundamenta en cuatro pilares técnicos que garantizan la viabilidad y robustez del negocio a largo plazo:

A) Integridad Referencial y Modelo Relacional

El dominio de problema de "CuidaDos" es inherentemente relacional. No estamos gestionando datos aislados (como logs o likes de una red social), sino transacciones comerciales complejas entre entidades interconectadas.

- **El Problema del NoSQL:** En bases de datos documentales, la relación entre un Contrato y un Usuario suele resolverse mediante desnormalización (duplicar los datos del usuario dentro del contrato). Si el usuario cambia su teléfono, habría que actualizarlo en todos sus contratos históricos, aumentando el riesgo de inconsistencia.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- **La Solución PostgreSQL:** Mediante el uso de claves foráneas (*Foreign Keys*), el sistema garantiza que un contrato siempre apunte a un usuario y una empresa existentes.

- *Restricción en Cascada:* Se ha configurado la base de datos con reglas ON DELETE CASCADE y ON UPDATE CASCADE.

Esto significa que si un Administrador elimina una Empresa fraudulenta, el motor de base de datos elimina automáticamente y de forma atómica todas sus ofertas y contratos pendientes, sin necesidad de escribir código extra en el Backend.

B) Cumplimiento ACID en Transacciones Críticas

Dado que la plataforma gestiona acuerdos de servicios que implican facturación y responsabilidades legales, la consistencia de los datos no es negociable. PostgreSQL es ACID Compliant:

1. Atomicidad: O se guarda todo el contrato, o no se guarda nada. No existen "medios contratos" si falla la red a mitad de operación.
2. Consistencia: Los datos siempre cumplen las reglas definidas (tipos de datos, restricciones CHECK).
3. Aislamiento: Si dos usuarios intentan contratar el mismo horario a la vez, el sistema gestiona la concurrencia para evitar solapamientos.
4. Durabilidad: Una vez confirmado el guardado, los datos persisten incluso ante un fallo eléctrico del servidor.

C) Comparativa Técnica: SQL vs NoSQL en el Contexto del Proyecto

A continuación se presenta la matriz de decisión utilizada para descartar el stack MEAN (Mongo) en favor del stack actual:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Criterio	PostgreSQL (Seleccionado)	MongoDB (Descartado)	Impacto en Cuidados
----------	------------------------------	-------------------------	---------------------

Estructura	Estricta (Schema-First)	Flexible (Schema-less)	Necesitamos validar DNI y CIF estrictamente.
Relaciones	JOINS Nativos eficientes	\$lookup costoso	Los listados cruzan datos de 3 tablas.
Integridad	Foreign Keys	Manual (código app)	Evitamos registros huérfanos ("basura").
Consultas	SQL Estándar complejo	JSON Query	SQL permite reportes analíticos potentes.

Tabla 2. Matriz.

D) Capacidades Híbridas (JSONB)

Aunque PostgreSQL es relacional, el proyecto aprovecha su capacidad moderna para almacenar datos semi-estructurados mediante el tipo de dato **JSONB**.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

- *Caso de Uso:* La tabla Auditoria almacena los logs del sistema. Como la estructura de un log puede variar (un error tiene "stack trace", un login tiene "IP"), se guardan como JSON binario dentro de una columna SQL. Esto nos da "lo mejor de los dos mundos": la rigidez para los datos financieros y la flexibilidad para los metadatos.

```
1. -- Ejemplo de Definición de Tabla con Integridad Referencial Estricta
2. -- (Extraído del esquema de producción de Cuidados)
3.
4. CREATE TABLE public.Contrato (
5.   id_contrato UUID DEFAULT gen_random_uuid() PRIMARY KEY,
6.   id_cliente UUID NOT NULL,
7.   id_empresa UUID NOT NULL,
8.   estado TEXT CHECK (estado IN ('pendiente', 'activo', 'cancelado')),
9.
10.  -- Restricción de Integridad (Foreign Key)
11.  CONSTRAINT fk_cliente
12.    FOREIGN KEY(id_cliente)
13.      REFERENCES public.Cliente(id_cliente)
14.      ON DELETE CASCADE, -- Si se borra el cliente, se borra el contrato
15.
16.  -- Validación de Lógica de Negocio en Base de Datos
17.  CONSTRAINT check_fechas
18.    CHECK (fecha_fin >= fecha_inicio)
19. );
20.
```

Script DDL mostrando la implementación de restricciones de integridad (Foreign Keys) y validaciones de negocio (Check Constraints) directamente en el motor de base de datos.

2.4.3. Row Level Security (RLS)

La seguridad no se gestiona en el código del servidor, sino directamente en el motor de la base de datos. Mediante políticas RLS, se define la lógica: "*Un usuario solo puede ver las filas de la tabla 'Mensajes' donde su ID sea igual al campo 'emisor' o 'receptor'*". Esto hace que la base de datos sea segura por diseño, incluso si la API es pública.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

2.5. Infraestructura y DevOps

2.5.1. Control de Versiones (Git)

Se ha seguido el flujo de trabajo Gitflow simplificado. Se mantiene una rama main para producción y ramas de características (*feature branches*) para cada nuevo desarrollo, asegurando que el código en producción siempre es estable.

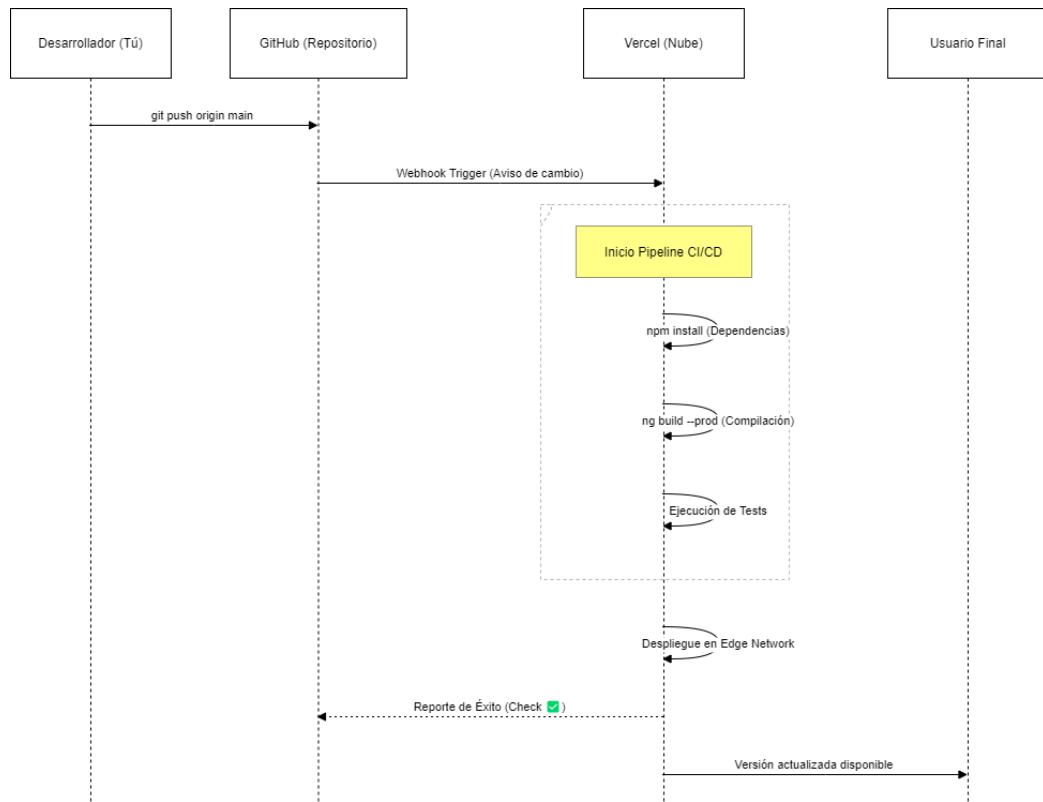


Ilustración 11: git.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

2.5.2. Despliegue Continuo (CI/CD) en Vercel

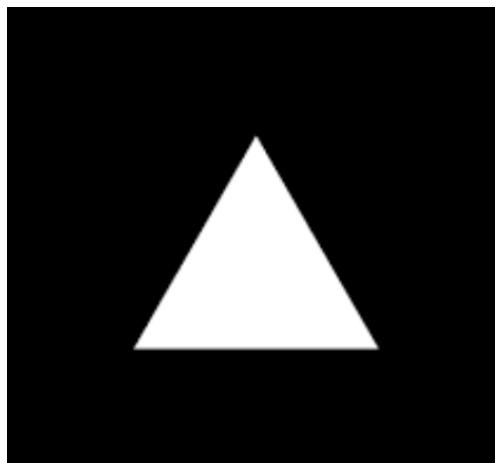


Ilustración 12: Vercel.

Para el alojamiento y distribución de la aplicación en producción, se ha seleccionado **Vercel**, una plataforma de infraestructura en la nube optimizada para frameworks frontend modernos. Esta elección trasciende el simple "hosting web", implementando una arquitectura de **Despliegue Continuo (CD)** basada en eventos.

Su integración nativa con el repositorio de **GitHub** permite establecer un flujo de trabajo **GitOps**: la "verdad" sobre el estado de la aplicación reside siempre en el código fuente.

A) Pipeline de Integración Continua (CI/CD)

El ciclo de vida del despliegue está totalmente automatizado, eliminando el error humano asociado a las subidas manuales por FTP o SSH. El flujo configurado es el siguiente:

1. **Trigger (Disparador):** Al realizar un *Push* o *Merge* en la rama main del repositorio.
2. **Build (Construcción):** Vercel detecta cambios, clona el repositorio en un contenedor efímero e instala las dependencias (*npm install*).

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3. **Compilación:** Ejecuta el script de construcción de Angular (ng build --configuration production), aplicando optimizaciones de *Tree-shaking* (eliminación de código muerto) y minificación AOT (Ahead-of-Time).
 4. **Despliegue Atómico:** Si la compilación es exitosa, se genera una nueva URL de producción. Si falla, el sistema detiene el proceso y notifica el error, asegurando que la web visible nunca se rompa.
- **Preview Deployments:** Adicionalmente, para cada *Pull Request*, Vercel genera una URL temporal única. Esto permite testear nuevas funcionalidades en un entorno real antes de fusionarlas con la rama principal.

B) Infraestructura Edge Network (CDN)

Vercel no sirve la aplicación desde un único servidor central. Utiliza una **Red de Entrega de Contenidos (CDN)** global.

- **Funcionamiento:** Los archivos estáticos generados por Angular (HTML, JS, CSS, imágenes) se replican automáticamente en decenas de "bordes" (*edges*) alrededor del mundo.
- **Beneficio Latencia Cero:** Cuando un usuario accede desde Mérida, no descarga la web desde un servidor en Estados Unidos, sino desde el nodo de Madrid o Lisboa. Esto reduce drásticamente el *Time to First Byte (TTFB)* y mejora las métricas vitales de Google (Core Web Vitals).

C) Configuración para Single Page Application (SPA)

Al tratarse de una aplicación Angular, la navegación es virtual (ocurre en el cliente). Para evitar errores "404 Not Found" cuando un usuario recarga la página en una ruta profunda (ej: /perfil/contratos), es necesario configurar reglas de reescritura en el servidor.

D) Seguridad SSL/TLS Automatizada

La plataforma gestiona automáticamente la capa de seguridad de transporte. Vercel aprovisiona y renueva certificados **SSL universales** (Let's Encrypt), forzando la

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

conexión a través de HTTPS. Esto es un requisito indispensable para que funcionen las características de la PWA como los *Service Workers* y la *Geolocalización*.

CAPÍTULO 3: DIARIO DE EJECUCIÓN

3.1. Hito: Elaboración de Contrato y Alcance (07.10.25)

La formalización del proyecto "CuidaDos" se inició mediante una reunión de arranque (*Kick-off Meeting*) con el equipo docente, actuando en el rol de *Stakeholders*. El objetivo principal fue establecer el Triángulo de Hierro del proyecto: Alcance, Tiempo y Coste.

Se identificaron restricciones críticas que condicionan la arquitectura del software:

1. Independencia del Dispositivo: Dado que el público objetivo incluye personas mayores y cuidadores con dispositivos de gama media/baja, se exige una optimización agresiva del *bundle* de JavaScript.
2. Presupuesto Cero (Zero-Budget): La infraestructura debe sostenerse sobre capas gratuitas (*Free Tier*). Esto validó la elección de Supabase (hasta 500MB de base de datos gratis) y Vercel.
3. Horizonte Temporal: El *Time-to-Market* (fecha de defensa) es inamovible (Junio 2026), lo que obliga a adoptar una metodología ágil para priorizar funcionalidades.

Como resultado, se redactó el Documento de Alcance (SOW). Este documento actúa como contrato vinculante, excluyendo explícitamente la gestión monetaria real (pasarela de pagos bancaria) para centrarse en la seguridad de la contratación y la comunicación.

ACTA DE REUNIÓN

	cuidaDos Technology	07/ q0/25	V 1.0	
Acta de Reunión. Consolidación de Acuerdos.	Pá gina 1/5	A R.CA- 2.0		

Fecha de la reunión: 07/10/25

Hora de la reunión: 17:30

Lugar de celebración: Mérida

Compañías: cuidaDos Technology, I.E.S Albarregas, MADERAS TERMITA VELOZ SL

Tabla de contenido

1. Introducción.....	2
2. Asistentes.....	3
3. Agenda del Día.....	4
4. Requisitos Definidos.....	4
5. Modelo de Usuarios y Permisos (RBAC).....	5
6. Casos tratados.....	5
6. Conclusión.....	7
7. Próxima reunión.....	7

1. Introducción

Lugar y Fecha: Mérida, a 30 de Septiembre de 2025.

La presente reunión se convocó con la finalidad de establecer el marco funcional y técnico del proyecto "CuidaDos", una Plataforma Web Progresiva (PWA) diseñada para la intermediación de servicios de asistencia domiciliaria. El objetivo principal es definir el alcance del Producto Mínimo Viable (MVP) que permitirá conectar a familias solicitantes con empresas proveedoras de servicios sociosanitarios.

Durante la sesión, que tuvo lugar en las instalaciones del IES Albarregas (Mérida), se abordaron distintos puntos clave: la definición de los requisitos funcionales (RF) y no funcionales (RNF) del sistema, la elección de la arquitectura tecnológica (Angular + Supabase), la asignación de roles de usuario y la revisión de los flujos de navegación (User Journeys) para validar la usabilidad del sistema.

Asimismo, se acordó establecer un cronograma de trabajo para el desarrollo de las distintas fases del proyecto (Análisis, Diseño, Implementación, Pruebas), definir responsables para cada módulo y documentar los acuerdos alcanzados con el fin de garantizar el seguimiento académico y técnico de los objetivos definidos.

Finalmente, se estableció como fecha provisional para la próxima reunión de control el **14 de Octubre de 2025**, con el propósito de validar el diagrama Entidad-Relación de la base de datos y los prototipos de baja fidelidad.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

2. Asistentes

ASISTENTE	ROL / ORGANIZACIÓN	CONEXIÓN	PUNTUALIDAD	FIRMA
Evelia Gil Paredes	<i>Lead Developer</i> (Alumna)	Presencial	<input checked="" type="checkbox"/> Sí	987678
Evelia Gil Paredes	<i>Frontend Dev</i> (Colaborador/a)	Presencial	<input checked="" type="checkbox"/> Sí	987655
Paco Mera	<i>Product Owner</i> (Tutor IES Albarregas)	Presencial	<input checked="" type="checkbox"/> Sí	987688
Alejandra Rodríguez	<i>Stakeholder</i> (Tutora IES Albarregas)	Presencial	<input checked="" type="checkbox"/> Sí	2039484

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3. Agenda del Día

Presentación del Concepto: Exposición de la problemática de la "Economía Plateada" y la solución propuesta mediante PWA.

Definición de Alcance: Establecimiento de los Requisitos Funcionales (RF) mínimos para el MVP.

Arquitectura Técnica: Confirmación del stack tecnológico (Angular 17, Supabase, Vercel).

Planificación: Establecimiento de plazos de ejecución (Cronograma de Sprints).

Simulación: Revisión de casos prácticos (User Stories) de contratación y búsqueda.

4. Requisitos Definidos

En este apartado se formalizan los requisitos funcionales tratados y aprobados durante la reunión:

- **Gestión de Identidad (Auth):** Sistema de registro seguro diferenciado. Se requiere validación algorítmica de DNI (para Clientes) y CIF (para Empresas) para evitar usuarios fraudulentos.
- **Motor de Búsqueda Reactivo:** Funcionalidad crítica que permita filtrar empresas por Localidad y Categoría de Servicio (ej: Fisioterapia, Acompañamiento) sin latencia perceptible (uso de Angular Signals).
- **Gestión de Contratos:** Capacidad de formalizar un acuerdo digital entre partes. El sistema debe controlar estados: Pendiente -> Activo -> Finalizado.
- **Control de Disponibilidad (Horarios):** Integración de un gestor de agenda que impida la contratación de servicios en franjas horarias ya ocupadas o días no laborables definidos por la empresa.
- **Comunicación en Tiempo Real:** Implementación de un chat privado vía WebSockets para que Cliente y Empresa negocien detalles antes de la contratación.
- **Panel de Administración:** Herramienta para la moderación de usuarios y visualización de métricas de uso (KPIs).

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

5. Modelo de Usuarios y Permisos (RBAC)

Se ha definido una estructura de acceso basada en roles:

Cliente (Rol: Usuario Final):

Permisos: Búsqueda de servicios, creación de solicitudes de contrato, chat con empresas, valoración de servicios recibidos.

Empresa (Rol: Proveedor):

Permisos: Gestión de perfil público, definición de catálogo de servicios y precios, configuración de horarios, aceptación/rechazo de contratos.

Administrador (Rol: Superusuario):

Permisos: Acceso total a la base de datos, capacidad de bloquear cuentas fraudulentas (Soft Delete), visualización de logs de auditoría.

Tutor/Auditor (Rol: Invitado):

Permisos: Acceso de solo lectura para la evaluación académica del proyecto durante la fase de defensa.

6. Casos tratados

Durante la sesión de trabajo se abordaron sistemáticamente cada uno de los puntos incluidos en la agenda, con el objetivo de alinear las expectativas técnicas con las necesidades del negocio.

- **Presentación del Ecosistema** En primer lugar, durante la «Presentación del Proyecto» se detalló el flujo operativo completo de la plataforma CuidaDos. Se definió el objetivo central: crear un sistema de intermediación digital seguro que elimine la economía sumergida en el sector, conectando a familias (Clientes) con proveedores profesionales (Empresas) mediante una PWA (Progressive Web App) de alta disponibilidad. Se discutió el impacto social esperado y cómo la tecnología reducirá los tiempos de contratación.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- **Registro Polimórfico:** Capacidad del sistema para diferenciar flujos de alta entre Clientes (físicos) y Empresas (jurídicas).
- **Motor de Búsqueda:** Filtrado de servicios por localidad y categoría en tiempo real.
- **Ciclo de Contratación:** Gestión de estados de un contrato (Pendiente -> Activo -> Cancelado/Finalizado).
- **Comunicación:** Chat integrado para la negociación previa a la contratación.
- **Planificación y Asignación:** Se asignaron las tareas y responsabilidades técnicas para las fases de desarrollo (Frontend en Angular, Backend en Supabase) y pruebas de calidad (QA). Se estableció un calendario provisional basado en Sprints semanales, definiendo hitos clave como la "Entrega del Prototipo" y el "Despliegue en Producción", sujetos a revisiones periódicas.
- **Análisis de Escenarios de Uso (User Journeys):** Durante la reunión, se simularon escenarios de uso complejos para evaluar la robustez de la lógica de negocio propuesta:
- **Escenario de Concurrencia:** ¿Qué ocurre si un usuario intenta contratar a una empresa en un horario que ya está ocupado?
- **Escenario de Abandono:** Gestión de contratos activos si una empresa decide darse de baja de la plataforma.
- **Escenario de Seguridad:** Intentos de registro con DNI's falsos o CIF's inválidos.
- **Comportamiento Esperado del Sistema:** Se concluyó que el sistema debe ser capaz de:
- **Evitar Solapamientos:** La base de datos (PostgreSQL) debe impedir mediante triggers o validaciones lógicas que existan dos contratos activos para el mismo servicio en la misma franja horaria.
- **Automatizar Validaciones:** El formulario de registro debe detectar errores de sintaxis en correos electrónicos y documentos de identidad (Algoritmo del DNI) antes de enviar la petición al servidor.
- **Notificaciones:** Generar alertas visuales (Toasts) y persistentes para informar a las empresas de nuevas solicitudes de servicio entrantes.
- **Seguridad y Auditoría:** También se prevé la funcionalidad para el perfil

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

6. Conclusión

Tras la revisión de los puntos de la agenda y los casos prácticos expuestos, los asistentes acordaron implantar un aplicativo moderno, seguro y adaptado a la normativa de privacidad de datos, con el fin de facilitar la prestación de servicios a personas con necesidades especiales o edad avanzada.

La siguiente sesión se programará dentro de una semana, con el objetivo de revisar los avances del proyecto y concretar las mejoras a implementar en la versión final del sistema, con los siguientes puntos a tratar:

- Establecer los requisitos no funcionales, priorizando la robustez, seguridad y facilidad de uso.
- El área de IT iniciará el desarrollo de los módulos principales (registro de turnos, validación de ausencias, bloqueo de fichajes, auditorías y detección de retrasos), siguiendo el calendario provisional definido, con revisiones periódicas semanales del progreso y basándose en los requisitos mínimos acordados.
- Resolver incidencias y garantizar el cumplimiento de los objetivos del proyecto.
- RRHH definirá protocolos internos de gestión de incidencias, validación de justificantes y aprobación de informes.
- Revisión del presupuesto actual, estableciendo la posibilidad de ampliarlo para el desarrollo de mejoras.

7. Próxima reunión

Fecha de la reunión: 07/10/25

Hora de la reunión: 17:30

Objetivo: Revisión de avances y ajuste de requisitos.

Ilustración 13: Acta.

CRONOGRAMA MACRO - PROYECTO CUIDADOS (2025-2026)

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

FASE / MES OCT NOV DIC ENE FEB MAR ABR MAY JUN

1. ANÁLISIS		OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN
Toma de Requisitos										
Definición Alcance										
Investigación Tech										
2. DISEÑO		OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN
UX / Wireframes										
UI / Prototipos										
Modelo B. Datos										
3. DESARROLLO		OCT	NOV	DIC	ENE	FEB	MAR	ABR	MAY	JUN
Frontend (Angular)										

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Backend (Supabase)						█				
Integración APIs						█	█			
4. PRUEBAS							█	█		
Test Unitarios							█			
Test Usuario								█		
5. DESPLIEGUE								█	█	
Documentación								█	█	
Defensa Final									█	

Tabla 3: Cronograma.

Formalización Contractual del Proyecto

Para garantizar la viabilidad del proyecto y establecer un marco de trabajo profesional, se procedió a la redacción y firma de un **Acuerdo de Desarrollo de Software y Prestación de Servicios**. Este documento contractual tiene como finalidad

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

evitar la ambigüedad en los requisitos y proteger la propiedad intelectual del trabajo académico.

El acuerdo regula la relación entre las partes durante el ciclo de vida del software, estableciendo las obligaciones técnicas, los hitos de entrega y los límites de responsabilidad.

El contrato que se muestra con fecha **24 de Septiembre de 2025**, tiene el objetivo de dotar de seguridad jurídica al desarrollo de la plataforma, se procedió a la firma del **Contrato de Prestación de Servicios Digitales (Ref: CPSD-1.0)** en la ciudad de Mérida.

Este documento regula la relación mercantil entre el equipo de desarrollo y la entidad beneficiaria, estableciendo las obligaciones técnicas, los plazos de ejecución y la cesión de derechos de explotación del software resultante.

Identificación de las Partes

El acuerdo vincula a los siguientes actores:

- **El Cliente (La Empresa): MADERAS TERMITA VELOZ S.L.**, con domicilio social en Cádiz y representada por D. Juan Jesús Espinosa Martínez en calidad de Gerente.
- **El Prestador (Desarrollador): Dña. Evelia Gil Paredes**, actuando como responsable técnica e ingeniería del software.

Objeto y Alcance Funcional (SOW)

Según la **Cláusula de Objeto del Contrato**, el encargo consiste en el desarrollo de un "aplicativo web para facilitar la contratación de servicios". El alcance técnico se ha cerrado en base a cinco Requisitos Funcionales (RF) innegociables descritos en el anexo del contrato:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

1. **RF1 - Inicio de Sesión:** Sistema de autenticación seguro.
2. **RF2 - Gestión de Contratos:** Ciclo de vida completo de la contratación digital.
3. **RF3 - Control de Incidencias:** Módulo para reporte y seguimiento de problemas.
4. **RF4 - Mensajería Realtime:** Chat en tiempo real entre usuarios (prioridad crítica).
5. **RF5 - Gestión de Usuarios:** Administración de perfiles y permisos.

Asimismo, se define un sistema de **Roles y Privilegios (RBAC)** que estructura el acceso a la aplicación en tres niveles:

- **Cliente:** Gestión de contratos propios.
- **Empresa:** Gestión de ofertas y contratos.
- **Administrador:** Privilegios totales sobre el sistema.

Cronograma y Modalidad de Trabajo

Se establece una duración total del proyecto de **5 meses**, con fecha de inicio efectiva el **24 de Septiembre de 2025**. La modalidad de trabajo pactada es **remota**, aunque se establecen cláusulas para permitir reuniones de coordinación y pruebas presenciales en las instalaciones del Cliente en el Polígono Industrial Cepansa (Cádiz) si la complejidad del despliegue lo requiere.

Propiedad Intelectual y Confidencialidad

Dada la naturaleza de los datos que manejará la aplicación (información personal de cuidadores y familias), se han estipulado cláusulas estrictas:

- **Cesión de Derechos:** Una vez finalizado el proyecto y abonada la contraprestación, la titularidad del código fuente, la documentación y los

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

derechos de explotación pasarán a ser propiedad exclusiva de **MADERAS TERMITA VELOZ S.L..**

- **Confidencialidad:** El Prestador se obliga a mantener el secreto profesional sobre la información sensible de la empresa y los usuarios, obligación que persiste incluso tras la finalización del contrato.

Garantía y Calidad del Servicio

El Prestador garantiza disponer de la infraestructura, cualificaciones y habilidades necesarias para ejecutar el desarrollo bajo estándares de calidad profesional. El acuerdo se define jurídicamente como un "contrato de medios" , comprometiéndose el desarrollador a una ejecución diligente y conforme a la normativa vigente, incluyendo el cumplimiento del RGPD en el tratamiento de datos.

ELEMENTO VISUAL (Firma del Contrato)

(FIRMA)

(FIRMA)

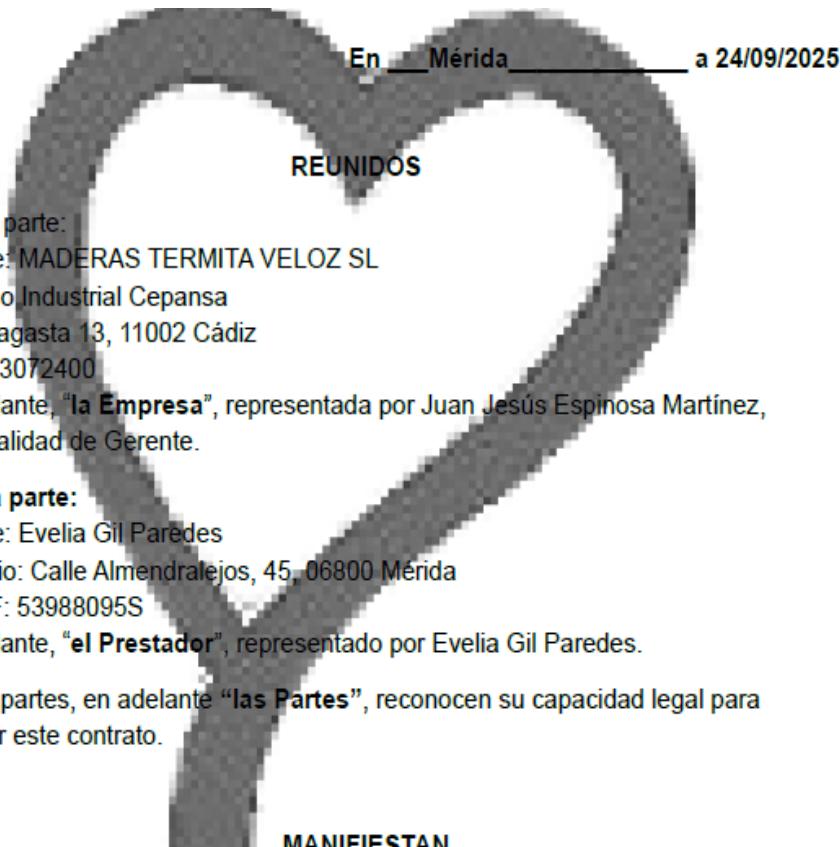
**	**
D./DÑA. JUAN JESÚS ESPINOSA	D./Dña. Evelia Gil Paredes
PRODUCT OWNER / STAKEHOLDER	Lead Developer
EN MÉRIDA, A 29 DE SEPTIEMBRE DE 2025	En Mérida, a 29 de Septiembre de 2025

Tabla 4: Firmas.

CONTRATO EJEMPLO

	cuidaDos Technology	24/09/25	V 1.0
	CONTRATO PRESTACIONES SERVICIOS DIGITALES	Página 1/10	CPSD-1.0

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

**MANIFIESTAN**

- I. Que el Cliente, habiendo sido correctamente informado sobre el alcance y características de los servicios objeto del presente Contrato, está interesado en encárgárselos al Prestador.
- II. Que la Empresa desea **contratar el aplicativo web cuidaDOS**.
- III. Que el Prestador, reuniendo las habilidades, cualificaciones y experiencia , infraestructura y materiales necesarios para poder prestar referidos servicios correctamente, está de acuerdo en prestar los mismo de conformidad con los términos y alcance definidos en este Contrato.
- IV. Que, en virtud de lo anterior, las Partes, reconociéndose mutua y recíproca capacidad legal necesaria y suficiente para contratar y obligarse, habiendo entendido el alcance, contenido y consecuencias de

FULL-STACK DEVELOPER - ANGULAR - REACT – TAILWIND - SPRINGBOT

<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

las declaraciones emitidas en el presente, y tras haber llegado libre y voluntariamente a un acuerdo mutuo justo para ambas, deciden suscribir este Contrato el cual se regirá en lo sucesivo de conformidad con lo indicado en las siguientes:

CLÁUSULAS

OBJETO DEL CONTRATO Y SERVICIOS A PRESTAR

1. En virtud del presente Contrato, el Cliente se compromete a contratar al Prestador los servicios indicados a continuación: _____ aplicativo web para facilitar la contratación de servicios que permita a la empresa:
 - Gestionar los contratos.
 - Conocer cuando se llevan a cabo los contratos.
 - Generar reportes sobre el aplicativo y su uso.
 - Cumplir con los requisitos funcionales definidos en este contrato.
- De aquí en adelante serán referenciados como el "Servicio" o los "Servicios" indistintamente.
2. El Prestador deberá desempeñar los Servicios de manera responsable y diligente, evitando cualquier conducta negligente que pudiera causar daños o perjuicios al Cliente o a terceros, siguiendo las instrucciones del Cliente y respetando en todo caso los estándares, normativos y costumbres del sector teniendo en cuenta las circunstancias y naturaleza concretas del Servicio.
3. En lo que respecta a la ubicación donde se prestarán los Servicios, las Partes acuerdan lo siguiente: _____ los servicios serán desarrollados de manera remota por el Prestador, pudiendo realizarse reuniones de coordinación y pruebas presenciales en las instalaciones de la Empresa, ubicadas en Polígono Industrial Cepansa, Calle Sagasta 13, 11002 Cádiz, cuando así se acuerde entre las Partes_____.
4. En caso de que exista alguna modificación con respecto al lugar en el que se prestarán los Servicios, el Cliente se lo deberá comunicar al Prestador con un plazo de tiempo suficiente y razonable, empleando un método de notificación fechaciente.
5. Con carácter general, no quedarán incluidos dentro de los Servicios cualesquiera otros servicios no expresamente determinados o incluidos en este Contrato y en el Presupuesto. No obstante lo anterior, las Partes, de mutuo acuerdo, estarán habilitadas para acordar nuevos servicios y actividades. Para lo anterior será necesario el acuerdo previo, expreso y por escrito de las Partes, y en caso de variación sustancial entre los antiguos Servicios y los nuevos añadidos, se podrán modificar proporcionalmente los plazos, costes y gastos inicialmente previstos. En ese sentido, las Partes deberán llegar a un nuevo acuerdo relativo al alcance de estos nuevos

FULL-STACK DEVELOPER - ANGULAR - REACT – TAILWIND - SPRINGBOT

<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

servicios, así como otros aspectos importantes como los términos económicos, planificación y otros. Estas nuevas condiciones quedarán incorporadas a este Contrato como anexos.

ALCANCE DEL PROYECTO

Usuarios y Roles

Rol	Privilegios
Cliente	Usuario/contraseña para acceso. Gestión de sus contratos.
Empresa	Usuario/contraseña para acceso. Gestión de sus contratos. gestión de sus ofertas.
Administrador	Todos los privilegios

Requisitos Funcionales (RF)

- RF1: Inicio de sesión.
- RF2: Gestión de contratos
- RF3: Control de incidencias.
- RF4: Mensajería en tiempo real.
- RF5: Gestión de usuarios.

Reglas de negocio y restricciones

- Acceso al aplicativo limitado a **mayores de edad**.
- Toda la información se almacenará en la BBDD de la Empresa.
- Incidencias reportadas al RRHH vía correo electrónico.

DURACIÓN DE LOS SERVICIOS

6. Este Contrato comenzará a desplegar efectos desde el momento de su firma y continuará en vigor hasta que los Servicios sean completamente prestados, sin perjuicio de una posible terminación anticipada de conformidad con lo acordado en este Contrato. A efecto de lo anterior, la

FULL-STACK DEVELOPER - ANGULAR - REACT – TAILWIND - SPRINGBOT
<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

prestación del Servicio se entenderá por completada cuando el Servicio haya sido completado y finalizado de conformidad con las especificaciones incluidas en este Contrato y en el Presupuesto, atendiendo a las circunstancias y naturaleza concreta del Servicio y a las prácticas y costumbres del sector.

7. En caso de controversia relacionado con lo anterior, las Partes se comprometen a negociar de buena fe lo anterior y resolver la controversia. En caso de no alcanzar un acuerdo, las Partes designarán de mutuo acuerdo un perito especializado en la materia que evaluará lo anterior, debiendo correr las mismas por igual con los costes que lo anterior genere. Las Partes aceptan y acuerdan que la decisión del perito será vinculante para las mismas. Si tras lo anterior, se siguiese sin encontrar una solución, se procederá a aplicar el método de resolución de conflictos incluido en este Contrato.

En lo que respecta a la fecha de comienzo de los Servicios, éstos comenzarán el día 24 de septiembre 2025. Sin perjuicio de lo anterior, las Partes acuerdan que esta fecha podrá ser flexible y que, de mutuo acuerdo, podrán determinar otra fecha concreta. El proyecto tendrá una duración de 5 meses desde la firma del contrato. Asimismo, las Partes podrán prolongar el Contrato siempre y cuando se cuente con el consentimiento por escrito de ambas Partes. De aquí en adelante el plazo o duración de los Servicios será referenciado como el "Plazo".

PRECIO Y FORMA DE PAGO

8. El coste del proyecto será de [por determinar] €, debiendo acordarse antes del inicio del desarrollo. Los pagos se realizarán conforme a los hitos pactados entre las partes.
9. En relación con el pago de la Remuneración, el Prestador facturará al Cliente cuando los Servicios hayan sido prestados completamente de conformidad con los términos y condiciones indicados en este Contrato.
10. A los efectos anteriores, el Prestador emitirá la correspondiente factura o recibo por los conceptos y plazos indicados en este Contrato, debiendo ser pagadas las mismas dentro de los 30 días siguientes a su emisión recepción. Esta factura deberá cumplir con los requisitos formales establecidos por la normativa aplicable, entre ellos: número y fecha de factura, las cantidades a cobrar correctamente desglosadas, impuestos aplicados, retenciones, y en su caso, serie, fecha de emisión, nombre y apellidos, razón o denominación social completa, tanto del Prestador, como del Cliente, domicilio de ambas Partes, y la descripción de los Servicios. Para la emisión de la citada factura, se emplearán los datos identificativos y fiscales incluidos en el encabezado de este Contrato.

FULL-STACK DEVELOPER - ANGULAR - REACT – TAILWIND - SPRINGBOT
<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

11. La Remuneración deberá ser pagada mediante transferencia bancaria a la siguiente cuenta:

Titular de la cuenta: Evelia Gil Paredes

Entidad Bancaria: Banca Pueyo E.S.

IBAN: ES98 37465 6464 6464 46464

DEMORA DEL PAGO

12. En caso de que el Cliente no pague en los plazos acordados, se cobrará un interés por demora equivalente al interés legal del dinero establecido en la Ley de Presupuestos Generales del Estado del correspondiente año. Las Partes acuerdan que estos intereses se calcularán anualmente sobre el total adeudado, y se devengarán desde el día siguiente al vencimiento del plazo de pago, siendo por tanto exigibles sin necesidad de que el Prestador tenga que realizar un requerimiento formal al Cliente, y continuarán devengándose hasta que se haya pagado el total de lo adeudado en virtud de este Contrato.

OBLIGACIONES DEL PRESTADOR

13. En el marco de la prestación de los Servicios, el Prestador deberá cumplir con todas y cada una de las obligaciones incluidas en este Contrato, así como aquellas que vengan impuestas por la normativa aplicable, y, particularmente:

- I. Actuará de manera eficiente, diligente y responsable de conformidad con los estándares profesionales y prácticas generalmente aceptadas en el sector, cumpliendo en todo momento la normativa aplicable y siguiendo las instrucciones indicadas en este Contrato y aquellas indicaciones que el Cliente le proporcione en un futuro. No obstante lo anterior, el Prestador no será responsable de los errores, defectos o demoras que tuvieran lugar en el marco de la prestación del Servicio cuando las mismas sean debidas a que la información proporcionada por el Cliente fuera falsa, o se hubiera omitido entregar una información necesaria para poder prestar el Servicio. El Prestador no estará obligado a verificar que la información proporcionada es verídica, cierta y actualizada. La responsabilidad que pudiera derivar de lo anterior será asumida por el Cliente.
- II. Salvo que expresamente se haya acordado lo contrario en este Contrato, dispondrá y mantendrá todos los medios y recursos materiales y/o humanos que sean necesarios para que el Servicio pueda ser prestado correctamente. Cualquier incidencia

FULL-STACK DEVELOPER - ANGULAR - REACT – TAILWIND - SPRINGBOT

<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

imputable al Prestador en la prestación de los Servicios, será corregida o subsanada a costa de éste en el menor plazo posible, atendiendo a su criticidad, riesgos, costes y posibles daños y perjuicios, de acuerdo con lo indicado en este Contrato.

- III. Emitirá las facturas de conformidad con los tiempos y condiciones estipulados en este Contrato.
- IV. Mantendrá correctamente informado al Cliente sobre la ejecución y estado de los Servicios prestados y se compromete a mantener un correcto flujo de comunicaciones Cliente-Prestador. En ese sentido, el Cliente podrá exigir al Prestador el envío periódico de información y documentación sobre el estado de la ejecución de los Servicios, de cara a que pueda realizar un correcto seguimiento de los mismos.
- V. Cumplirá con los plazos y condiciones de la prestación del Servicio indicado en este Contrato.
- VI. Cumplirá con las obligaciones vinculadas a protección de datos personales, y otras incluidas en este Contrato.
- VII. Responderá frente al Cliente por incumplimiento del Contrato efectuado por sus empleados y/o subcontratados o terceros vinculados, y se compromete a tomar las medidas necesarias para subsanar tales incumplimientos. En ese sentido, el Prestador se obliga particularmente a cumplir con las obligaciones laborales de sus empleados, y garantizar que todo su personal, entidades subcontratadas o terceros vinculados conozcan perfectamente las obligaciones y responsabilidades asumidas en virtud de este Contrato, entre otras, las relativas a las condiciones del Servicio, y al tratamiento de datos personales. Estas obligaciones deberán ser directamente trasladadas e impuestas a los mismos bajo las mismas condiciones que las acordadas en este Contrato.
- VIII. En caso de que el Cliente tenga la consideración de consumidor y el Prestador de empresario o proveedor a efectos del Real Decreto Legislativo 1/2007, de 16 de noviembre, por el que se aprueba el texto refundido de la Ley General para la Defensa de los Consumidores y Usuarios, y otras leyes complementarias, el Prestador deberá cumplir con todas las obligaciones y derechos a los que la citada normativa en materia de consumidores obligase.
- IX. Entregará documentación técnica y manual de usuario al finalizar el proyecto.
- X. Brindará soporte técnico durante el periodo pactado, según acuerdo entre las Partes.

OBLIGACIONES DEL CLIENTE

FULL-STACK DEVELOPER - ANGULAR - REACT - TAILWIND - SPRINGBOT
<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

14. En el marco de la prestación de los Servicios, el Cliente deberá cumplir con todas y cada una de las obligaciones incluidas en este Contrato, así como aquellas que vengan impuestas por la normativa aplicable, y, particularmente:
- I. Abonará las facturas por los Servicios prestados de conformidad con las condiciones, plazos y demás términos establecidos en este Contrato. Ante el incumplimiento de cualesquiera de las obligaciones de pago, la Parte incumplidora entrará, de forma automática y sin necesidad de previo requerimiento de pago por la Parte acreedora, en mora y quedará obligada al pago de las cantidades adeudadas incrementadas en el interés o cuantía aquí fijado, sin perjuicio de la reclamación que adicionalmente corresponda por los daños y perjuicios causados.
 - II. Proporcionará al Prestador toda la información y/o documentación que sea necesaria para que el Prestador pueda prestar el Servicio correctamente y garantizará un correcto flujo de comunicación Cliente-Prestador.
 - III. Respetará el trabajo y autonomía del Prestador, y no interferirá en su actividad profesional de manera injustificada o desproporcionada, siempre y cuando se estuvieran respetando los términos acordados en este Contrato y no se estuviera vulnerando la normativa aplicable. Sin perjuicio de lo anterior, esta obligación de respeto a la autonomía del Prestador no exime al Cliente de su derecho de supervisar y controlar la ejecución de los Servicios encargados, siempre y cuando se realice de manera proporcional y respetuosa de conformidad con las prácticas aceptadas en el sector.
 - IV. Cumplirá con las obligaciones fiscales derivadas de este Contrato que le sean imputables de conformidad con la normativa aplicable.

PROPIEDAD INTELECTUAL

15. El código, documentación y derechos de propiedad intelectual del aplicativo desarrollado serán propiedad de la Empresa una vez recibido el pago completo y aprobado el proyecto.

CONFIDENCIALIDAD

16. El Prestador se compromete a no divulgar información sensible de la Empresa ni de los usuarios del sistema, durante y después de la vigencia del contrato.

TERMINACIÓN

17. El contrato podrá resolverse por incumplimiento de cualquiera de las Partes, previo aviso escrito con 15 días de antelación, sin perjuicio de reclamar los daños ocasionados.

FULL-STACK DEVELOPER - ANGULAR - REACT – TAILWIND - SPRINGBOT

<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

DECLARACIONES Y GARANTÍAS DEL PRESTADOR

18. A efectos del cumplimiento de este Contrato, el Prestador DECLARA Y GARANTIZA que:
- I. Dispone de capacidad legal necesaria y suficiente para poder contratar y, en particular, para poder celebrar este Contrato.
 - II. No existe ningún impedimento de carácter legal, técnico, administrativo o financiero que imposibilite suscribir este Contrato.
 - III. Cumplirá con todas las obligaciones indicadas en este Contrato y con aquellas impuestas por la normativa aplicable.
 - IV. Cuenta con todos los permisos, licencias y autorizaciones técnicas, administrativas y legales necesarias para prestar el Servicio, así como otras licencias o certificaciones a las que se haya obligado a obtener de conformidad con lo indicado en este Contrato. Cualquier cambio relacionado con lo anterior deberá ser comunicado imperativamente dentro de las 48 horas siguientes a que lo conozca o que razonablemente lo debiera haber conocido.
 - V. Ha contratado o contratará antes de que el Servicio comience, un seguro de responsabilidad civil que cubra los posibles daños y perjuicios que pudieran producirse en el marco de la prestación del Servicio, todo ello de conformidad con lo acordado en este Contrato. Este requisito no será necesario cuando el Prestador ya disponga de un seguro de responsabilidad civil profesional que cubra lo anterior.
 - VI. Previamente a la firma de este Contrato, ha informado correctamente al Cliente sobre las características esenciales del Servicio y le ha proporcionado toda información precontractual necesaria de conformidad con la normativa aplicable.
 - VII. Es plenamente conocedor que este Contrato no creará ningún tipo de relación laboral entre las Partes, ni ningún tipo de asociación más allá de lo estrictamente derivado de la relación mercantil derivada de la prestación del Servicio.
 - VIII. Dispone de los suficientes recursos y medios para poder prestar el Servicio de conformidad a las condiciones acordadas entre las Partes en este Contrato y en el Presupuesto.
 - IX. Todas las declaraciones y manifestaciones emitidas aquí expresadas son verídicas y ciertas.

DECLARACIONES Y GARANTÍAS DEL CLIENTE

19. A efectos del cumplimiento de este Contrato, el Cliente DECLARA Y GARANTIZA que:
- I. Dispone de capacidad legal necesaria y suficiente para poder contratar y, en particular, para poder celebrar este Contrato.
 - II. No existe ningún impedimento de carácter legal, técnico, administrativo o financiero que imposibilite suscribir este Contrato.

FULL-STACK DEVELOPER - ANGULAR - REACT - TAILWIND - SPRINGBOT

<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- III. Cumplirá con todas las obligaciones indicadas en este Contrato y con aquellas impuestas por la normativa aplicable.
- IV. Dispone de suficiente capacidad económica para pagar la Remuneración y no se encuentra ni se prevé encontrar en situación de quiebra o concurso de acreedores.
- V. Ha sido correctamente informado por el Prestador sobre las características esenciales del Servicio y acepta las mismas tal y como son definidas en este Contrato.
- VI. Conoce y acepta que este Contrato es un contrato de medios y no de resultados, por lo que no se garantiza un resultado específico o concreto, sino que bastará con que se cumplan con las obligaciones aquí asumidas y se preste el Servicio de manera eficiente, responsable y diligente de acuerdo con las indicaciones aquí expresadas y respetando la normativa aplicable.
- VII. Es plenamente conocedor que este Contrato no creará ningún tipo de relación laboral entre las Partes, ni ningún tipo de asociación más allá de lo estrictamente derivado de la relación mercantil derivada de la prestación del Servicio.
- VIII. Todas las declaraciones y manifestaciones emitidas aquí expresadas son verídicas y ciertas.

CESIÓN DEL CONTRATO

20. Prestador no podrá ceder, ni voluntariamente ni por efecto de la ley, las obligaciones asumidas en este Contrato, salvo que cuente con el consentimiento previo y por escrito por parte del Cliente.

ANEXOS

21. Los anexos (si existen) se numerarán y referenciarán en el presente contrato, indicando versión y relación con el contrato principal.

FIRMAS

Por la Empresa:

Juan Jesús Espinosa Martínez

Por el Prestador:

Evelia Gil Paredes

FULL-STACK DEVELOPER - ANGULAR - REACT – TAILWIND - SPRINGBOT
<https://arTev.es> c/Almendralejos, 43, 06800, Mérida eveliagil@gmail.com

Ilustración 14: Contrato.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3.2. Hito: Definición de Requisitos y Benchmarking (14.10.25)

3.2.1. Metodología de Análisis de Mercado

En esta fecha se inició la fase de ingeniería de requisitos, aplicando la técnica de Benchmarking Competitivo. El objetivo no fue solo listar competidores, sino realizar una auditoría funcional para detectar "huecos de mercado" (Gaps) que CuidaDos pudiera capitalizar.

Se analizaron las tres plataformas líderes en el sector *AgeTech* (Tecnología para la tercera edad) en España: Cuidum, Depencare y Ayuda a Domicilio.

Hallazgos del Análisis: El estudio reveló una carencia sistemática en el modelo de negocio actual: la burocracia en la comunicación.

1. El modelo "Caja Negra": La mayoría de las plataformas actúan como agencias de colocación tradicionales digitalizadas. Ocultan los datos del cuidador o empresa hasta que el cliente paga una cuota de entrada.
2. Fricción en la Contratación: El tiempo medio desde el registro hasta el contacto real supera las 48 horas, lo cual es inaceptable en situaciones de urgencia familiar.

Propuesta de Valor Definida: Basándose en estos datos, se definió el pilar central de CuidaDos: "*Comunicación Primero*". Se decidió eliminar las barreras artificiales, permitiendo el contacto directo (P2P) mediante chat antes de la formalización del contrato.

- Impacto en el Alcance: Esta decisión estratégica elevó la prioridad del requisito RF-09 (Sistema de Chat Realtime) de "Deseable" a "Crítica", obligando a reevaluar la arquitectura técnica para incluir WebSockets (Supabase Realtime) desde el MVP.

A continuación, se presenta la matriz comparativa resultante del estudio de mercado, contrastando las funcionalidades de **CuidaDos** frente a los actores dominantes.

Dimensiones de Análisis	CuidaDos (Propuesta TFG)	Cuidum / Depencare	Portales de Anuncios (Milanuncios)
-------------------------	--------------------------	--------------------	------------------------------------

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Modelo de Interacción	Directo (P2P): El cliente elige libremente en el catálogo.	Intermediado: La plataforma asigna al cuidador (Opaco).	Caótico: Listado sin verificar, contacto por teléfono personal.
Tiempo de Respuesta	Inmediato (< 1 min): Gracias al Chat Realtime y Búsqueda Reactiva.	Lento (24-48h): Requiere validación manual de un gestor.	Variable: Depende de si cogen el teléfono.
Validación de Seguridad	Alta (Algorítmica): Validación de DNI/CIF y algoritmos de reputación.	Alta (Manual): Entrevistas realizadas por RRHH.	Nula: Riesgo alto de fraude o perfiles falsos.
Coste para el Usuario	Gratis: Modelo Freemium para la búsqueda.	Alto: Cobran comisión de gestión mensual o matrícula.	Gratis: Pero con mucha publicidad invasiva.
Experiencia de Usuario	PWA (App Nativa): Instalable, notificaciones push, UX moderna.	Web Tradicional: Formularios largos y recargas de página.	Web Generalista: No adaptada al sector cuidados.
Transparencia	Total: Se ven precios, horarios y valoraciones antes de contratar.	Parcial: El precio final suele variar tras el presupuesto.	Total: Pero sin garantía de veracidad.

Tabla 5: Análisis.

3.2.3. Modelado de User Journeys (Viajes de Usuario)

Para tangibilizar los requisitos abstractos, se diseñaron los "Viajes de Usuario" (*User Journeys*) identificando los puntos de dolor emocional y técnico.

A) User Journey 1: La Urgencia Familiar (Cliente)

1. Definición de la Persona (Arquetipo)

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- **Nombre:** María (45 años).
- **Ocupación:** Administrativa con jornada completa e hijos a cargo.
- **Contexto del Disparador (Trigger):** Su padre ha sufrido una caída doméstica leve un domingo por la tarde. Recibe el alta hospitalaria, pero necesita supervisión constante durante las mañanas de la próxima semana mientras ella trabaja.
- **Estado Emocional:** Nivel de ansiedad alto (9/10). Siente culpa por no poder cuidarlo ella misma y prisa extrema por encontrar una solución antes del lunes a las 08:00 AM.
- **Restricciones:** No tiene tiempo para entrevistar a 10 personas ni esperar a que una agencia tradicional abra el lunes por la mañana.

2. Puntos de Dolor en Soluciones Tradicionales (*Pain Points*)

El análisis previo detectó que, con las herramientas actuales, María se enfrenta a:

- **Incertidumbre:** En portales de anuncios (tipo Milanuncios), llama a números que no responden o ya no están disponibles.
- **Opacidad:** En agencias tradicionales, no ve la cara ni el perfil del cuidador hasta que paga o firma.
- **Lentitud:** Los formularios de solicitud de presupuesto tardan 24-48h en ser contestados. **María no tiene ese tiempo.**

3. Flujo de Resolución en CuidaDos (Happy Path)

El sistema se ha diseñado para reducir la **fricción cognitiva** y ofrecer una experiencia de "compra inmediata" similar a Amazon o Uber, pero aplicada a los cuidados.

1. **Descubrimiento (Segundos 0-10):** María accede a la Web App desde su móvil. La *Landing Page* no le obliga a registrarse para empezar. Un botón grande "Buscar Cuidador" elimina la barrera de entrada.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

2. Búsqueda Reactiva (Segundos 10-30):

- Introduce su código postal.
- El sistema filtra automáticamente por geolocalización.
- *Interacción Clave:* Al escribir "Enfermería", la lista de resultados se actualiza instantáneamente (sin recargar la página).

3. Evaluación de Confianza (Segundos 30-60): María ve la tarjeta de "Cuidarte SL". Observa tres datos críticos que reducen su ansiedad:

- Sello de "Identidad Verificada".
- Disponibilidad para "Mañana Lunes".

4. Acción y Conversión (Minuto 1): Pulsa "Contactar". El sistema solicita registro simplificado (Google Auth). En menos de 2 minutos, está chateando directamente con la empresa para confirmar la hora de llegada.

4. Mapeo Técnico: De la Emoción al Código

Este viaje de usuario justifica varias decisiones de arquitectura técnica tomadas en el proyecto:

Necesidad del Usuario (Emocional)	Requisito Técnico (Ingeniería)	Implementación en CuidaDos
--------------------------------------	-----------------------------------	----------------------------

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

"Tengo prisa, la web no puede ser lenta"	Latencia < 100ms	Uso de Angular Signals para el renderizado granular de la lista de búsqueda, evitando el <i>Zone.js overhead</i> .
"¿Será una persona real y segura?"	Integridad de Datos	Validación de CIF/DNI mediante Edge Functions en Supabase antes de activar el perfil público.
"Necesito respuesta ya, aunque sea domingo"	Tiempo Real	Implementación de WebSockets (Supabase Realtime) para que el chat y las notificaciones sean instantáneos.
"No quiero instalar nada"	Accesibilidad	Tecnología PWA (Service Workers) para que la web funcione como app nativa sin pasar por la Store.

Tabla 6: Resumen.

Conclusión del Journey: El éxito de este flujo no reside en la funcionalidad de "buscar", sino en la velocidad y la transparencia. La arquitectura se ha optimizado para que María pase de la "Ansiedad" al "Alivio" en menos de 5 clics.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

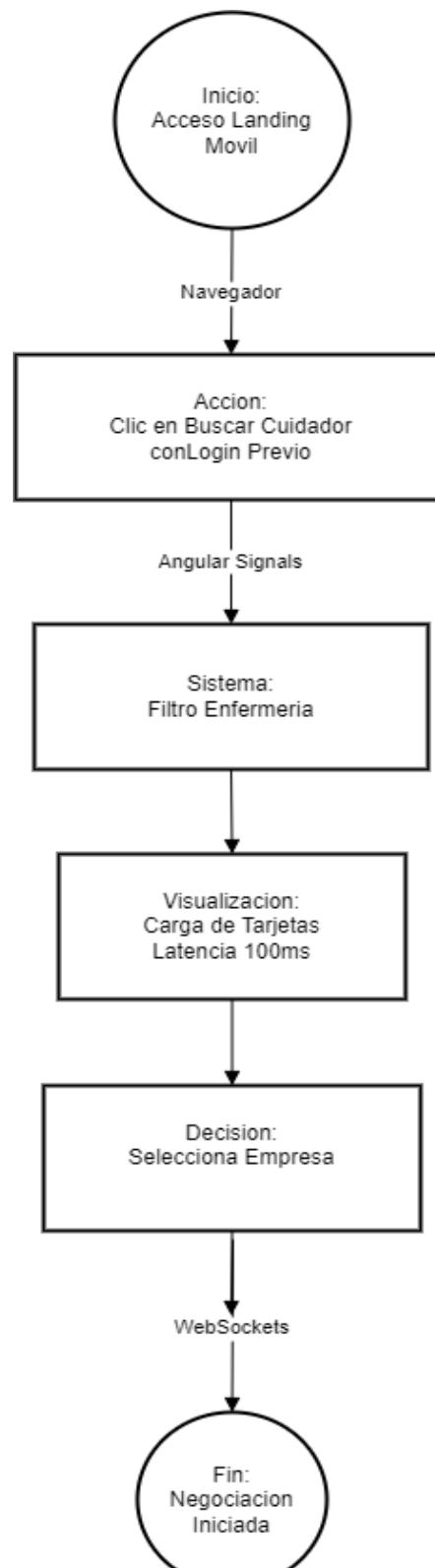


Ilustración 15: Journey cliente.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3.3. Hito: Revisión y Validación Técnica (21.10.25)

La validación técnica temprana destapó el "**Problema de las Fechas**". JavaScript maneja las fechas en el navegador usando la zona horaria local del usuario, mientras que la base de datos PostgreSQL trabaja en UTC. Esto podría causar que un contrato creado un "Lunes a las 23:00" se guarde como "Martes a las 00:00" si no se controla.

Decisión Arquitectónica: Se estandarizó el uso del formato **ISO 8601** (YYYY-MM-DDTHH:mm:ss.sssZ) para todo el transporte de datos, delegando el formateo visual a la librería Intl del navegador o date-fns.

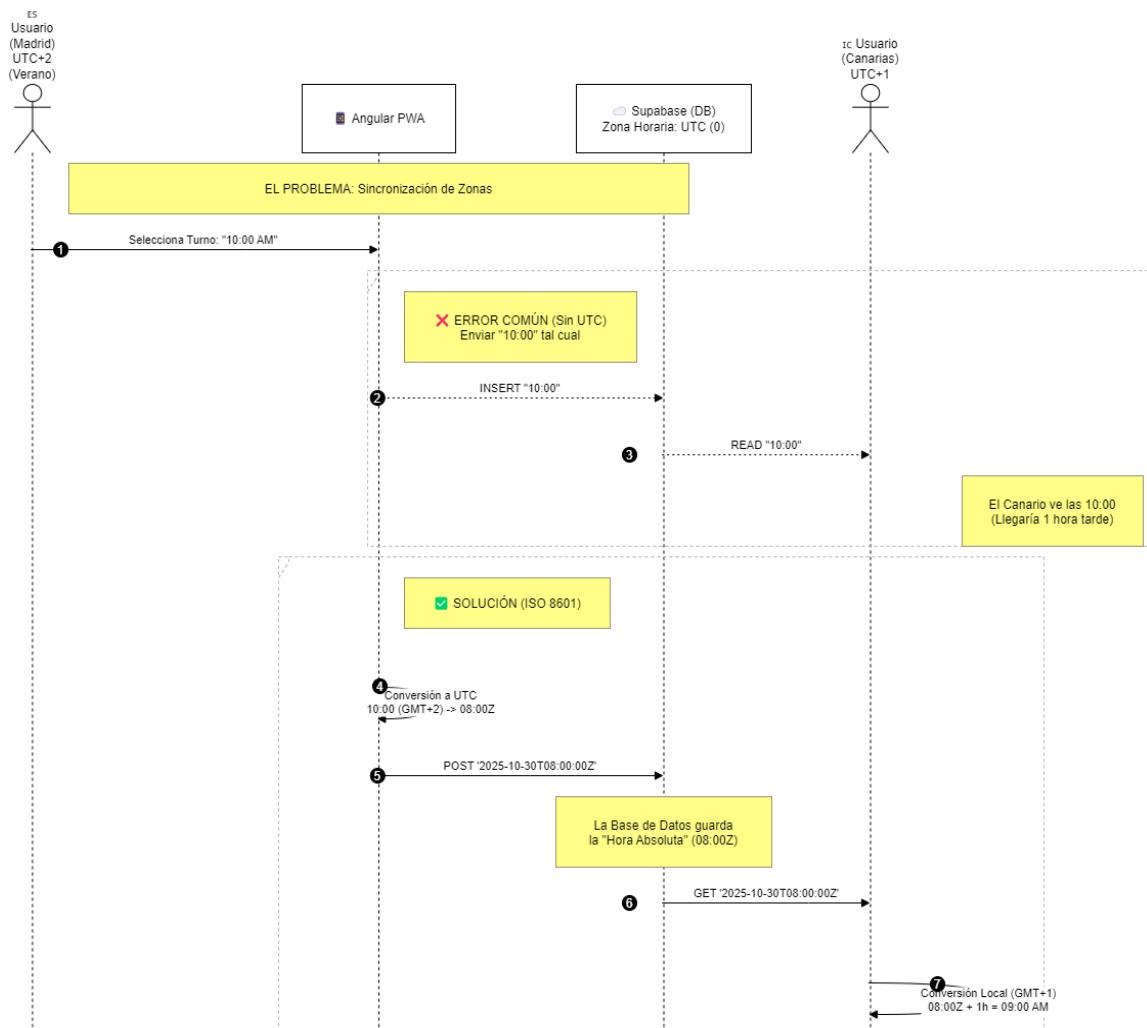


Ilustración 16: Validación técnica.

También, se optó por una **estrategia de Discretización Temporal**.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Selectores vs. Calendario Continuo

Para la gestión de la disponibilidad, se evaluaron dos paradigmas de interfaz de usuario (UI). Finalmente, se descartó la implementación de una interfaz tipo "Agenda Visual" (Drag & Drop) en favor de un modelo de **Entrada Discretizada** mediante selectores nativos (<select> de Día y Hora).

Esta decisión se fundamenta en la reducción de la complejidad ciclomática y la mejora de la usabilidad móvil:

1. Simplificación de la Lógica de Negocio (Discretización)

Un calendario abierto permite al usuario seleccionar tiempos arbitrarios (ej: 10:13 a 11:42). Esto obliga al *Backend* a realizar cálculos complejos de intersección de rangos temporales para evitar solapamientos (*Range Overlap Problem*).

Al implementar un selector cerrado con intervalos definidos (ej: en punto y media: 10:00, 10:30, 11:00), transformamos un problema de **tiempo continuo** en uno de **tiempo discreto**.

- **Impacto en el Código:** En lugar de algoritmos matemáticos de colisión, la validación se reduce a una búsqueda de igualdad exacta (*Exact Match Query*).
 - *Consulta Compleja:* WHERE (start < new_end) AND (end > new_start)
 - *Consulta Simplificada (Tuya):* WHERE fecha = '2025-10-30' AND hora_inicio = '10:00'

2. Usabilidad Mobile-First y Accesibilidad

Los componentes de calendario completo (tipo *FullCalendar*) suelen presentar problemas graves de usabilidad en pantallas pequeñas: dificultad para "arrastrar" eventos con el dedo, problemas de zoom y renderizado pesado. El uso de selectores nativos garantiza:

1. **Compatibilidad Total:** El sistema operativo del móvil (iOS/Android) muestra su propio selector de rueda optimizado.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

2. **Prevención de Errores:** El usuario no puede introducir una hora inválida o con formato erróneo, ya que solo puede elegir las opciones pre-validadas que le ofrece el sistema.

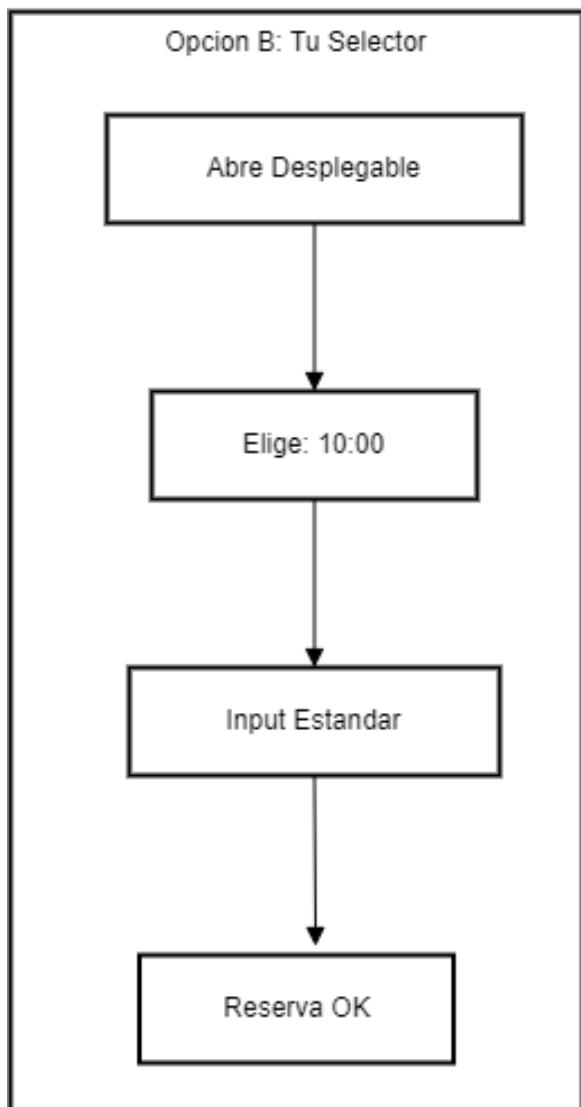


Ilustración 17: Selección horario.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3.4. Hito: Creación de Imagen Corporativa (23.10.25)

El diseño UI/UX se fundamentó en dos pilares conceptuales: **Ayuda** (amabilidad, confianza) y **Amistar** (cercanía, amor).

Para la tipografía, se seleccionó **Inter**. La justificación técnica reside en que es una *Variable Font*. Esto permite cargar un solo archivo de fuente y obtener múltiples pesos (Bold, Semibold, Light), reduciendo las peticiones HTTP y mejorando el *First Contentful Paint* (FCP).

El logotipo pasó por 15 iteraciones. Se descartaron las formas agudas o tecnológicas agresivas (cuadrados perfectos) en favor de trazos redondeados (border-radius altos), psicológicamente asociados a la protección y el cuidado humano. Evolución del Logotipo (Versión Final).



Ilustración 18: Logo final.

3.5. Hito: Desarrollo de Interfaces y Atomic Design (04.11.25)

Tras la validación de los wireframes funcionales, se procedió a la fase de diseño visual de alta fidelidad utilizando la herramienta **Figma**. Para garantizar la consistencia

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

visual y la escalabilidad del desarrollo frontend posterior, no se diseñaron pantallas aisladas, sino que se construyó un **Sistema de Diseño (Design System)** basado en la metodología **Atomic Design** propuesta por Brad Frost.

Esta metodología permitió abordar la interfaz de usuario (UI) desde una perspectiva modular, dividiendo los elementos en niveles jerárquicos de complejidad creciente. Esta decisión aceleró el desarrollo en Angular, ya que cada componente visual en Figma tiene su correspondencia directa (1:1) con un componente de código (@Component).

El día 11.11.25 se realizó la iteración y mejora de las Interfaces: revisión de las interfaces diseñadas para mejorar usabilidad y accesibilidad y ajustes en la navegación y distribución de elementos.

El 18.11.25 se validaron las Interfaces, para ello se realizó una presentación de las interfaces a la empresa para su evaluación.

También se obtuvo feedback y propuestas de mejora. Con esta fase, la aprobación del diseño visual antes de la fase de desarrollo técnico dio lugar a los prototipos que se mostrarán a lo largo de esta guía. |

3.5.1. Desglose de la Arquitectura Visual

El sistema se estructuró en tres niveles fundamentales:

1. **Átomos (Atoms):** Son las unidades indivisibles de la interfaz. En este nivel se definieron las variables de diseño globales (Design Tokens) como la paleta de colores, las tipografías (Inter/Fraunces), los iconos SVG y los estados base de los botones (Default, Hover, Disabled).
2. **Moléculas (Molecules):** Agrupaciones de átomos que funcionan como una unidad operativa. Por ejemplo, la unión de un Label (texto), un Input (campo) y un Icon (lupa) crea la molécula "**Buscador**".
3. **Organismos (Organisms):** Secciones complejas de la interfaz formadas por grupos de moléculas. Un ejemplo claro en el proyecto es la "**Tarjeta de Cuidador**", compuesta por una foto de perfil, un indicador de estado, chips de habilidades y un botón de contacto.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3.5.2. Uso de "Auto Layout" y Variantes

Para simular el comportamiento *responsive* del navegador web, todos los componentes en Figma se construyeron utilizando la propiedad **Auto Layout**.

Esto permite que los contenedores se adapten dinámicamente al contenido, simulando el comportamiento de Flexbox en CSS. Asimismo, se utilizaron **Component Properties** (Variantes) para gestionar los diferentes estados (Activo/Inactivo) sin multiplicar el número de capas innecesariamente.

DISEÑO PROTOTIPO

Se adjuntan las imágenes correspondientes a los componentes individuales y al prototipo responsivo.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

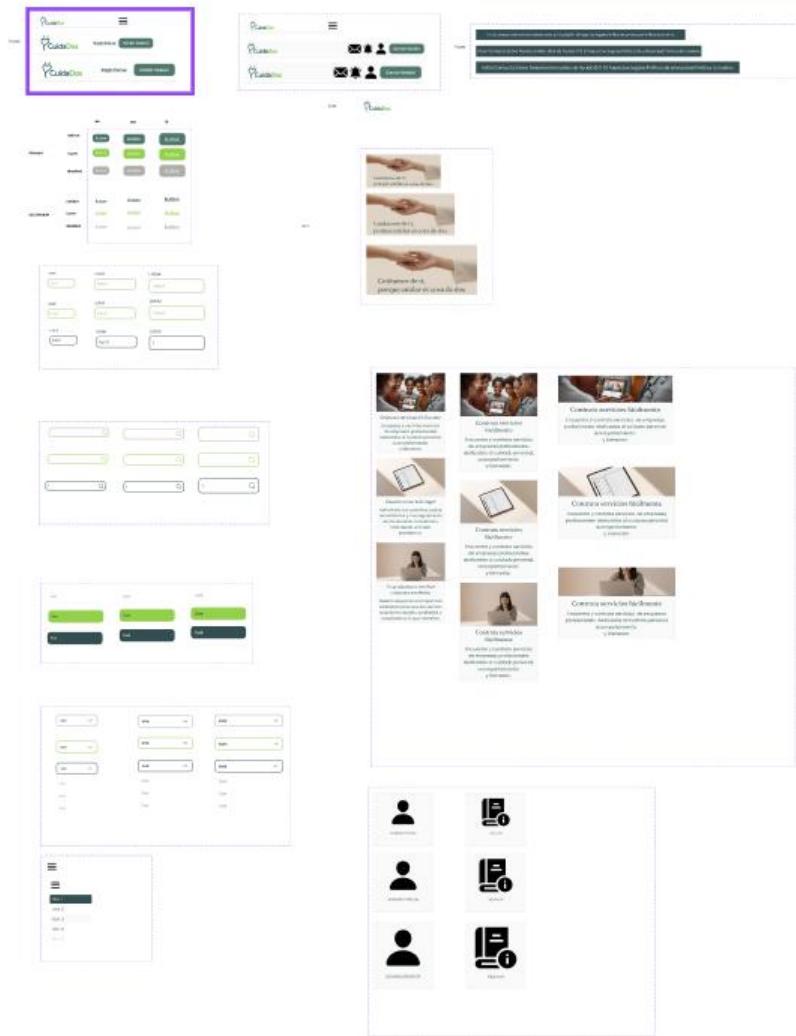


Ilustración 19: Diseño responsive.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

The diagram illustrates a user flow process:

- Mobile Messaging Interface:** A large vertical rectangle on the left containing three separate message composition screens. Each screen has fields for 'Nombre' (Name), 'Asunto' (Subject), and 'Mensaje' (Message), with a green 'enviar' (Send) button at the bottom.
- Notification Summary:** A dashed box containing three cards labeled 'Nuevo' (New). The first two are green, and the third is orange with a black icon.
- Notification Details:** A dashed box containing two sections: 'NOTIFICACIÓN' (Notification) with 'NOMBRE' (Name) and 'FECHA' (Date), and 'NOTIFICACIÓN' (Notification) with 'NOMBRE' (Name) and 'FECHA' (Date).
- Calendars:** Three separate calendar components. The top one shows the month of September 2023 with days 1 through 30. The middle one shows September 2025 with days 1 through 30. The bottom one shows September 2026 with days 1 through 30.

MANUAL CUIDADOS

Versión: 1.0 Fecha: 2026

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

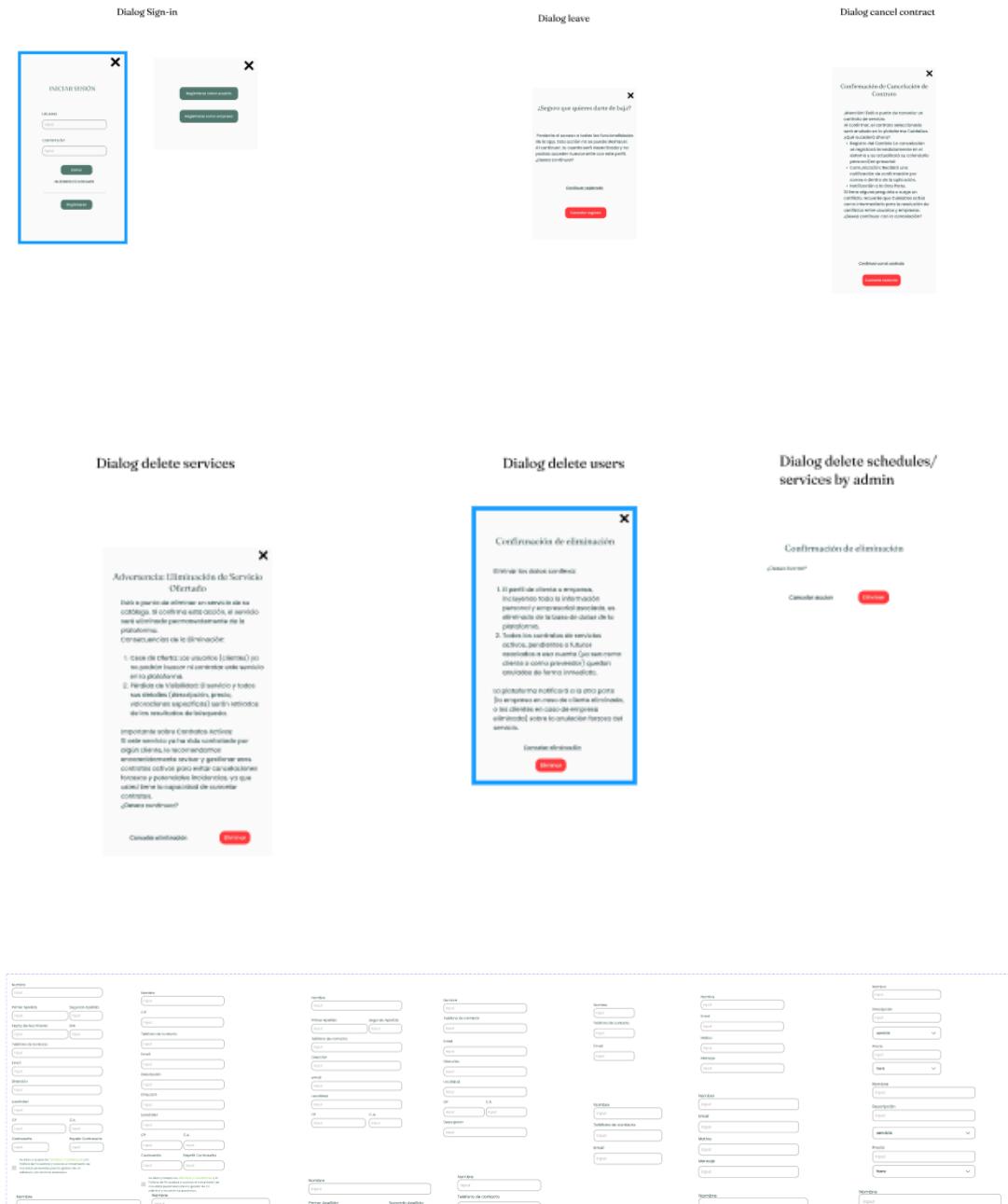


Ilustración 20: Diseño Responsive Componentes.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3.5.3. Implementación del Andamiaje (Scaffold) y Configuración

Como cierre de este hito, una vez definidos los átomos visuales en Figma, se procedió a la materialización del entorno de desarrollo. Se generó el *scaffold* (esqueleto) del proyecto utilizando el CLI de Angular, tomando en este punto decisiones arquitectónicas críticas que definirían el resto del desarrollo.

A) Estrategia de Renderizado (SPA vs SSR) Se ejecutó el comando de inicialización con el flag `--ssr=false`.

Justificación Técnica: Esta configuración fuerza la generación de una **Single Page Application (SPA)** pura basada en *Client-Side Rendering (CSR)*. Al descartar el *Server-Side Rendering (SSR)*, eliminamos la necesidad de mantener un servidor Node.js activo, simplificando el despliegue en plataformas *Serverless* como Vercel y reduciendo los costes de infraestructura a cero.

B) Gestión de Dependencias (Stack Tecnológico) Como evidencia del inicio del desarrollo, se muestra la secuencia de comandos ejecutada para aprovisionar las librerías base (Tailwind para estilos y Ngx-Translate para internacionalización), preparando el sistema para ser escalable desde el minuto uno.

ARTEFACTO: Secuencia de Inicialización (Terminal)

```
Bash

# 1. Generación del proyecto SPA (Sin Server-Side Rendering)
> ng new cuidados-app --style=css --ssr=false

# 2. Navegación e instalación del motor de estilos (PostCSS + Tailwind)
> cd cuidados-app
> npm install -D tailwindcss postcss autoprefixer
> npx tailwindcss init

# 3. Instalación de librerías Core (Base de Datos + i18n)
> npm install @supabase/supabase-js
> npm install @ngx-translate/core

# 4. Generación del primer Átomo (Componente Button)
> ng generate component components/atoms/button
```

Ilustración 21: Inicialización.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3.6. Hito: Desarrollo Estructura BDD (25.11.25)

En esta jornada se procedió a la materialización del modelo conceptual en un entorno de persistencia real. Se eligió Supabase (PostgreSQL) como motor de base de datos debido a su compatibilidad nativa con tipos de datos fuertes y su capacidad de exponer una API RESTful automática.

Las tareas realizadas fueron:

1. Aprovisionamiento: Creación del proyecto cuidados-app-db en la región eu-central-1 (Frankfurt) para cumplir con la latencia mínima.
2. Definición de Tipos: Se establecieron los dominios de datos. Se decidió utilizar UUID (Universally Unique Identifier) v4 para todas las claves primarias (PK) en lugar de enteros autoincrementales (SERIAL).
 - o *Justificación:* Los UUIDs previenen ataques de enumeración (adivinar el ID de otro usuario sumando +1) y facilitan la fusión de bases de datos en el futuro si el sistema escala.
3. Creación de Tablas Base: Se ejecutaron los scripts DDL (*Data Definition Language*) para las tablas.

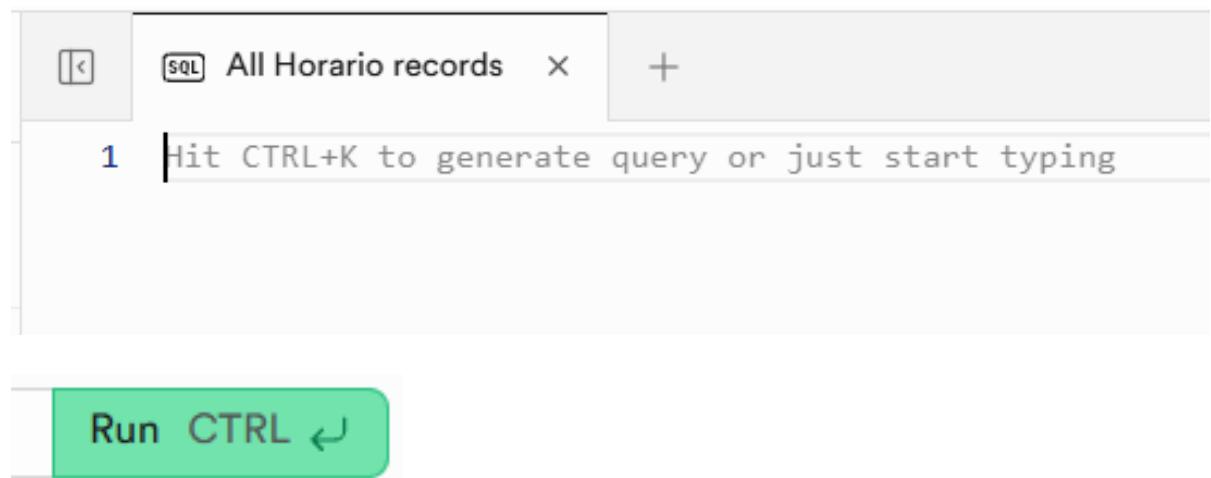


Ilustración 22: Supabase.

Tablas:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

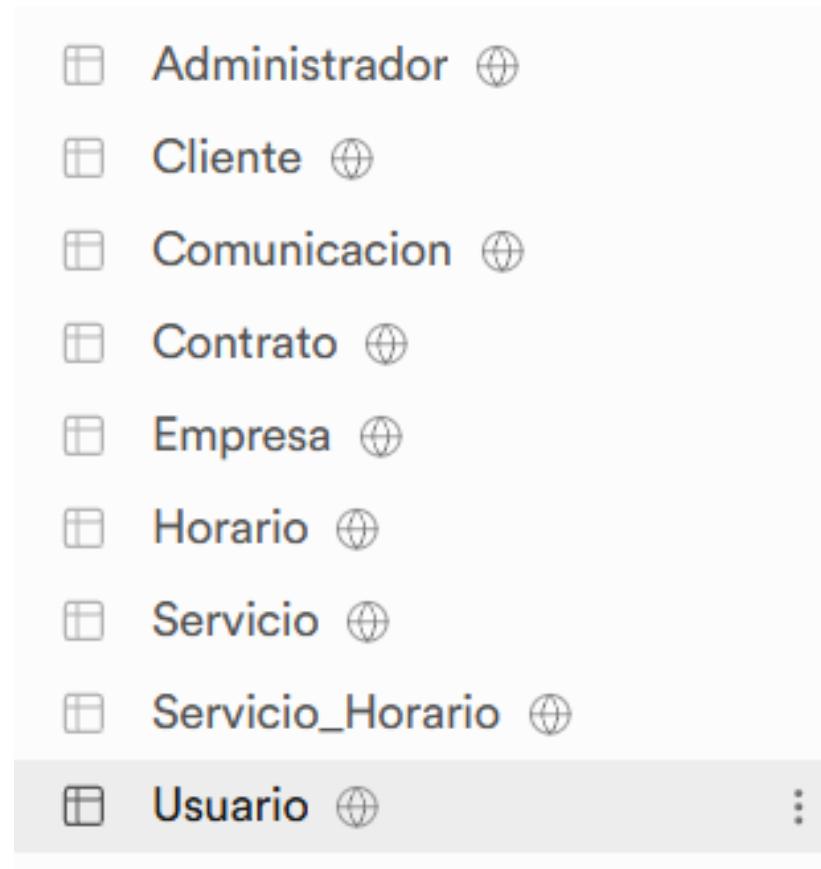


Ilustración 23: Tablas.

Administrador:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Columns

Name	Type	Default Value	Primary
id_administrador	uuid	auth.uid()	<input checked="" type="checkbox"/> X

Foreign keys

fk_admin_usuario	Foreign key relation to: public.Usuario	Edit	Remove
id_administrador → public.Usuario.id_usuario			
fk_cliente_usuario	Foreign key relation to: public.Usuario	Edit	Remove
id_administrador → public.Usuario.id_usuario			

Ilustración 24: Tabla I.

Cliente:

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Columns

Name	Type	Default Value	Primary
id_cliente	uuid	auth.uid()	<input checked="" type="checkbox"/> X
ape1	T varchar	NULL	<input type="checkbox"/> 1 X
ape2	T varchar	NULL	<input type="checkbox"/> 1 X
dni	T varchar	NULL	<input type="checkbox"/> X
direccion	T varchar	NULL	<input type="checkbox"/> X
fechnac	date	NULL	<input type="checkbox"/> X
localidad	T text	NULL	<input type="checkbox"/> 1 X
codpostal	T text	NULL	<input type="checkbox"/> 1 X
comunidad	T text	NULL	<input type="checkbox"/> 1 X

Foreign keys

Cliente_id_cliente_fkey	Foreign key relation to: public.Usuario	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
id_cliente → public.Usuario.id_usuario			

Ilustración 25: Tabla II.

Comunicación:

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Columns

Name	Type	Default Value	Primary
id_comunicaci	T uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
tipo_comunicac	T varchar	NULL	<input type="checkbox"/>
contenido	T varchar	NULL	<input type="checkbox"/>
fecha_envio	T date	NULL	<input type="checkbox"/>
leido	bool	NULL	<input type="checkbox"/>
id_receptor	uuid	auth.uid()	<input type="checkbox"/> 1
id_emisor	uuid	auth.uid()	<input type="checkbox"/> 1
asunto	T text	NULL	<input type="checkbox"/>
eliminado_por	bool	false	<input type="checkbox"/>
eliminado_por	bool	false	<input type="checkbox"/>

Foreign keys

fk_comunicacion_emisor	Foreign key relation to: public.Usuario	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
	id_emisor → public.Usuario.id_usuario		
fk_comunicacion_receptor	Foreign key relation to: public.Usuario	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
	id_receptor → public.Usuario.id_usuario		

Ilustración 26: Tabla III.

Contrato:

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Columns

About data types

Name	Type	Default Value	Primary
id_contrato	uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
estado	text	NULL	<input type="checkbox"/>
fecha_inicio	date	NULL	<input type="checkbox"/>
fecha_fin	date	NULL	<input type="checkbox"/> 1
dia_semana_c	text	NULL	<input type="checkbox"/>
hora_contrata	time	NULL	<input type="checkbox"/>
fecha_creacion	date	NULL	<input type="checkbox"/>
id_servicio_hc	uuid	NULL	<input type="checkbox"/> 1
id_cliente	uuid	NULL	<input type="checkbox"/> 1
id_empresa	uuid	NULL	<input type="checkbox"/> 1

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

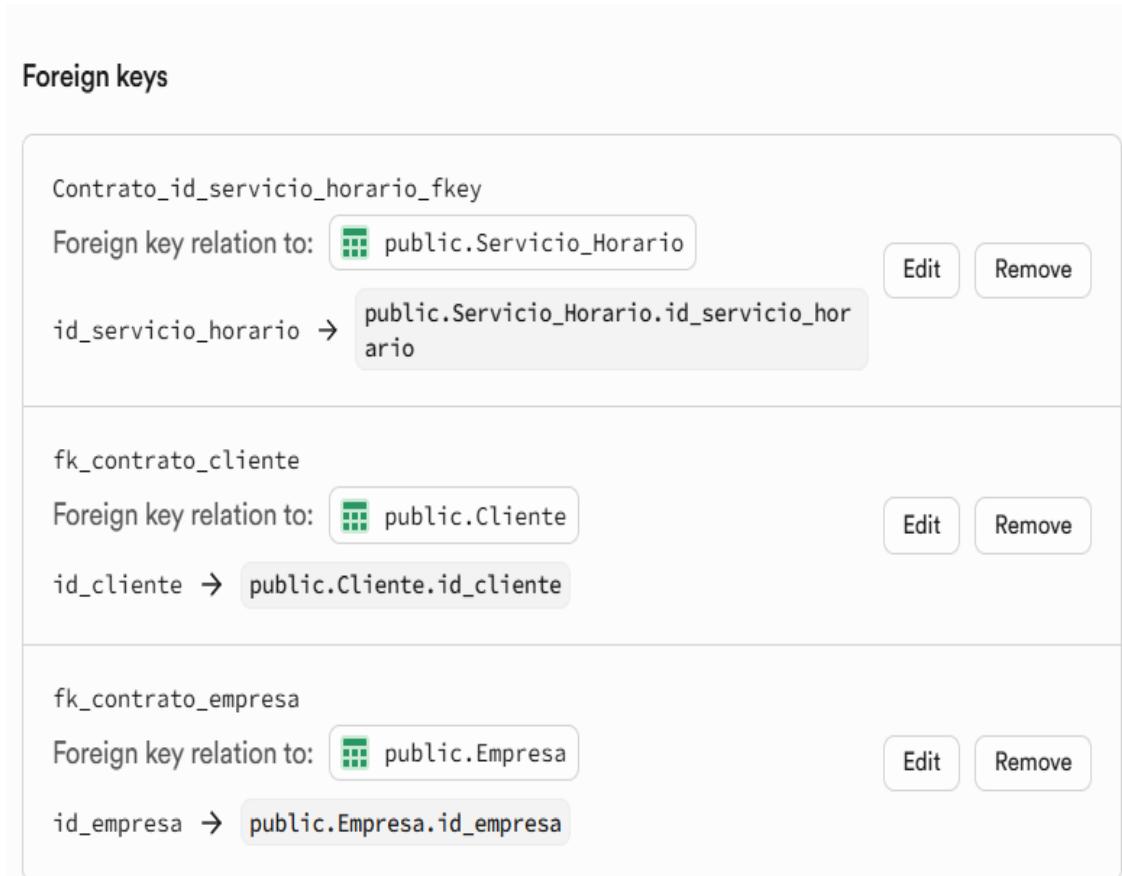


Ilustración 27: Tabla IV.

Empresa:

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Columns

Name	Type	Default Value	Primary
id_empresa	uuid	auth.uid()	<input checked="" type="checkbox"/>
cif	T varchar	NULL	<input type="checkbox"/>
direccion	T varchar	NULL	<input type="checkbox"/>
descripcion	T text	NULL	<input type="checkbox"/> 1
localidad	T text	NULL	<input type="checkbox"/> 1
codpostal	T text	NULL	<input type="checkbox"/> 1
comunidad	T text	NULL	<input type="checkbox"/> 1

Foreign keys

fk_empresa_herencia

Foreign key relation to: [public.Usuario](#)

`id_empresa → public.Usuario.id_usuario`

[Edit](#) [Remove](#)

Ilustración 28: Tabla V.

Horario:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Columns

[About data types](#)

Name	Type	Default Value	Primary
id_horario	uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
dia_semana	text	NULL	<input type="checkbox"/>
hora	time	NULL	<input type="checkbox"/>
id_admin	uuid	NULL	<input type="checkbox"/> 1

Foreign keys

Horario_id_admin_fkey

Foreign key relation to: [public.Administrador](#)

id_admin → [public.Administrador.id_administrador](#)

[Edit](#) [Remove](#)

Ilustración 29: Tabla VI.

Servicio:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Columns

[About data types](#)

Name	Type	Default Value	Primary
id_servicio	T uuid	gen_random_uuid()	<input checked="" type="checkbox"/>
nombre	T varchar	NULL	<input type="checkbox"/>
tipo_servicio	T text	NULL	<input type="checkbox"/>
id_admin	uuid	NULL	<input type="checkbox"/> 1

Foreign keys

Servicio_id_admin_fkey

Foreign key relation to: [public.Administrador](#)

[Edit](#) [Remove](#)

id_admin → public.Administrador.id_administrador

Ilustración 30: Tabla VII.

Servcio_Horario:

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Columns

Name	Type	Default Value	Primary
id_servicio_hc	uuid	gen_random_uuid()	<input checked="" type="checkbox"/> X
id_servicio	uuid	NULL	<input type="checkbox"/> 1 X
id_horario	uuid	NULL	<input type="checkbox"/> 1 X
id_empresa	uuid	auth.uid()	<input type="checkbox"/> 1 X
precio	# float4	NULL	<input type="checkbox"/> X
descripcion	text	NULL	<input type="checkbox"/> 1 X

Foreign keys

fk_id_horario	Foreign key relation to: public.Horario	Edit	Remove
id_horario	→ public.Horario.id_horario		
fk_id_servicio	Foreign key relation to: public.Servicio	Edit	Remove
id_servicio	→ public.Servicio.id_servicio		
fk_serv_horario_empresa	Foreign key relation to: public.Empresa	Edit	Remove
id_empresa	→ public.Empresa.id_empresa		

Ilustración 31: Tabla VIII.

Usuario:

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Name	Type	Default Value	Primary
id_usuario	uuid	auth.uid()	<input checked="" type="checkbox"/> X
telef	T varchar	NULL	<input type="checkbox"/> 1 X
email	T varchar	NULL	<input type="checkbox"/> 1 X
nombre	T text	NULL	<input type="checkbox"/> 1 X
fecha_registro	T date	NULL	<input type="checkbox"/> 1 X
estado	bool	false	<input type="checkbox"/> 1 X
rol	T text	'cliente':text	<input type="checkbox"/> 1 X

Foreign keys	
fk_usuario_auth	
Foreign key relation to:	auth.users
id_usuario →	auth.users.id
<input type="button" value="Edit"/> <input type="button" value="Remove"/>	

Ilustración 32: Tabla IX.

El 02.12.25 se llevó a cabo la optimización del Diseño de BDD, con la revisión del diseño de la base de datos para evitar redundancias, el ajuste de estructuras para mejorar el rendimiento y la escalabilidad. Finalmente, se preparó el modelo relacional definitivo.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3.7. Hito: Optimización y Modelo Relacional (09.12.25)

Desarrollo Técnico y Refactorización

Refactorización y Diseño del Esquema Relacional

Una vez establecido el prototipo inicial, se llevó a cabo una **Auditoría de Calidad del Dato** para alinear la arquitectura con los estándares de la industria. Durante esta fase, se detectó que el esquema preliminar presentaba vulnerabilidades estructurales, específicamente violaciones de la **Tercera Forma Normal (3NF)**.

Normalización y Eliminación de Anomalías

El diseño original almacenaba la información de ubicación (Dirección, Código Postal, Ciudad) directamente en la tabla Contratos. Esto generaba dos problemas críticos:

- **Redundancia de Datos:** Si un usuario firmaba 10 contratos, su dirección se duplicaba 10 veces, aumentando innecesariamente el almacenamiento.
- **Anomalía de Actualización:** Si el usuario se mudaba, el sistema debía actualizar N registros. Si el proceso fallaba a la mitad, se producía una inconsistencia de datos (direcciones antiguas mezcladas con nuevas).

Solución Implementada: Se procedió a la **descomposición vertical** de las tablas. La dirección se extrajo a la entidad Perfiles, vinculando los contratos únicamente por la ID del usuario. Esto asegura que la dirección sea un atributo atómico dependiente únicamente de la clave primaria del usuario, cumpliendo estrictamente la 3NF.

Estrategia de Herencia y Polimorfismo

Uno de los desafíos en bases de datos relacionales es cómo mapear la herencia de objetos (Usuario -> es Cliente / es Empresa). Se evaluaron tres patrones estándar:

- *Single Table Inheritance* (Una tabla gigante con muchos NULLs). **Descartado.**
- *Table per Concrete Class* (Tablas totalmente separadas). **Descartado.**

Table per Type (Joined Strategy). Seleccionado.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Se implementó una tabla maestra public.users (gestionada por Supabase Auth) que contiene los datos comunes (UUID, Email, Created_At). De ella "cuelgan" tablas satélites (public.profiles_clients y public.profiles_companies) vinculadas por una relación 1:1.

Beneficio Técnico: Esta estrategia permite aplicar restricciones específicas (NOT NULL) a los campos de la empresa (como el CIF) sin afectar a los clientes, manteniendo la integridad estructural que una tabla única no permitiría.

Integridad Referencial y "Cascading"

Para garantizar la higiene de la base de datos a largo plazo, se definieron restricciones a nivel de motor (PostgreSQL Constraints) en lugar de confiar en la lógica del Frontend.

- Se configuró la directiva **ON DELETE CASCADE** en las claves foráneas críticas.
- *Escenario:* Si un usuario ejerce su "Derecho al Olvido" (GDPR) y elimina su cuenta.
- *Comportamiento:* El motor de base de datos detecta la eliminación del nodo padre (User) y propaga la eliminación automáticamente a sus nodos hijos (Contratos, Mensajes, Reseñas).
- *Resultado:* Se previene la aparición de "registros huérfanos" que corromperían las estadísticas del sistema.

```
1. SQL
2. -- Ejemplo de implementación de la restricción en SQL
3. ALTER TABLE public.contratos
4. ADD CONSTRAINT fk_cliente_contrato
5. FOREIGN KEY (cliente_id)
6. REFERENCES public.profiles_clients (id)
7. ON DELETE CASCADE; -- Limpieza automática
8.
```

"Database Freeze" y Modelo Final

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Con fecha 10.12.2025, se alcanzó el hito de **Congelación de Base de Datos**. A partir de este punto, cualquier cambio en el esquema (DDL) requirió un proceso formal de migración para evitar romper la compatibilidad con el código de producción.

- El modelo relacional definitivo se estructura en torno a las siguientes cardinalidades:
- **Usuario (1) ↔ (N) Contratos:** Un usuario puede ser titular de múltiples contratos históricos, pero un contrato pertenece a un único titular.
- **Empresa (1) ↔ (N) Servicios:** La empresa define su catálogo (Enfermería, Acompañamiento).
- **Servicio (N) ↔ (M) Horarios:** Relación compleja resuelta mediante la tabla pivote disponibilidad_slots, que permite cruzar la oferta de la empresa con la demanda del cliente.

Desnormalización Controlada: El Patrón "Snapshot"

Durante la fase de validación del modelo, se identificó un conflicto entre la pureza teórica (Normalización) y la realidad operativa del negocio. Inicialmente, el horario del servicio dependía exclusivamente de la relación Servicio_Horario.

El Problema Detectado: Si una empresa decidía cambiar el horario de un servicio (ej: de "Lunes 10:00" a "Lunes 11:00"), esa modificación se propagaba retroactivamente a todos los contratos firmados en el pasado, alterando el histórico y generando inconsistencias legales.

Solución Implementada: Se aplicó una **desnormalización intencional** en la tabla Contratos, añadiendo las columnas dia_semana_contratado y hora_contratada. Aunque esto introduce redundancia (violando técnicamente la 3FN), se justifica por:

1. **Inmutabilidad Histórica:** Garantiza que el contrato conserve los datos exactos del momento de la firma, independientemente de cambios futuros en el catálogo de la empresa.
2. **Rendimiento de Lectura:** Permite consultar la agenda del cliente sin necesidad de realizar múltiples JOINs anidados con las tablas maestras de horarios.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Optimización de Índices: Claves Compuestas vs. Surrogadas

Para la tabla intermedia Servicio_Horario, se evaluó utilizar una Clave Primaria Compuesta (id_servicio + id_horario). Sin embargo, se optó finalmente por una **Clave Primaria Artificial (Surrogate Key)** autoincremental (id_servicio_horario).

Esta decisión arquitectónica aporta tres ventajas técnicas:

1. **Eficiencia en Referencias:** Facilita enormemente la creación de la tabla Contratos, que ahora solo necesita una Foreign Key (id_servicio_horario) en lugar de arrastrar dos claves compuestas.
2. **Velocidad de Indexado:** Los índices sobre un único campo entero (INTEGER) son computacionalmente más rápidos que sobre tuplas compuestas.
3. **Integridad Híbrida:** Se mantiene la restricción de unicidad mediante un índice UNIQUE (id_servicio, id_horario), evitando duplicados lógicos sin sacrificar la simplicidad.

Lógica de Negocio: Desacoplamiento de Precios

Se refactorizó la entidad Servicio para que actúe como un catálogo genérico (definido por el Administrador), moviendo los atributos precio y descripción a la tabla Servicio_Horario.

- **Motivo:** Permite que distintas empresas ofrezcan el mismo servicio estándar (ej: "Acompañamiento") pero cada una fije su propio precio y condiciones específicas, otorgando flexibilidad total al modelo de oferta.

Implementación DDL Final

A continuación, se presenta el script SQL definitivo para la entidad crítica del sistema, reflejando las decisiones de integridad y snapshotting mencionadas:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

```
SQL

-- Creación de la tabla central de operaciones (Anexo I)
CREATE TABLE Contrato (
    id_contrato INT PRIMARY KEY AUTO_INCREMENT,
    id_cliente INT NOT NULL,
    id_empresa INT NOT NULL,
    id_servicio_horario INT NOT NULL,

    -- Campos de Snapshot (Inmutabilidad del dato)
    fecha_inicio DATETIME NOT NULL,
    fecha_fin DATETIME,
    hora_contratada TIME NOT NULL,          -- Dato congelado
    dia_semana_contratado VARCHAR(10) NOT NULL, -- Dato congelado

    estado BOOLEAN DEFAULT TRUE, -- TRUE = activo, FALSE = finalizado

    -- Restricciones de Integridad Referencial
    FOREIGN KEY (id_cliente) REFERENCES Cliente(id_usuario) ON DELETE CASCADE,
    FOREIGN KEY (id_empresa) REFERENCES Empresa(id_usuario),
    FOREIGN KEY (id_servicio_horario) REFERENCES Servicio_Horario(id_servicio_horario);
);
```

Ilustración 33: Contrato.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

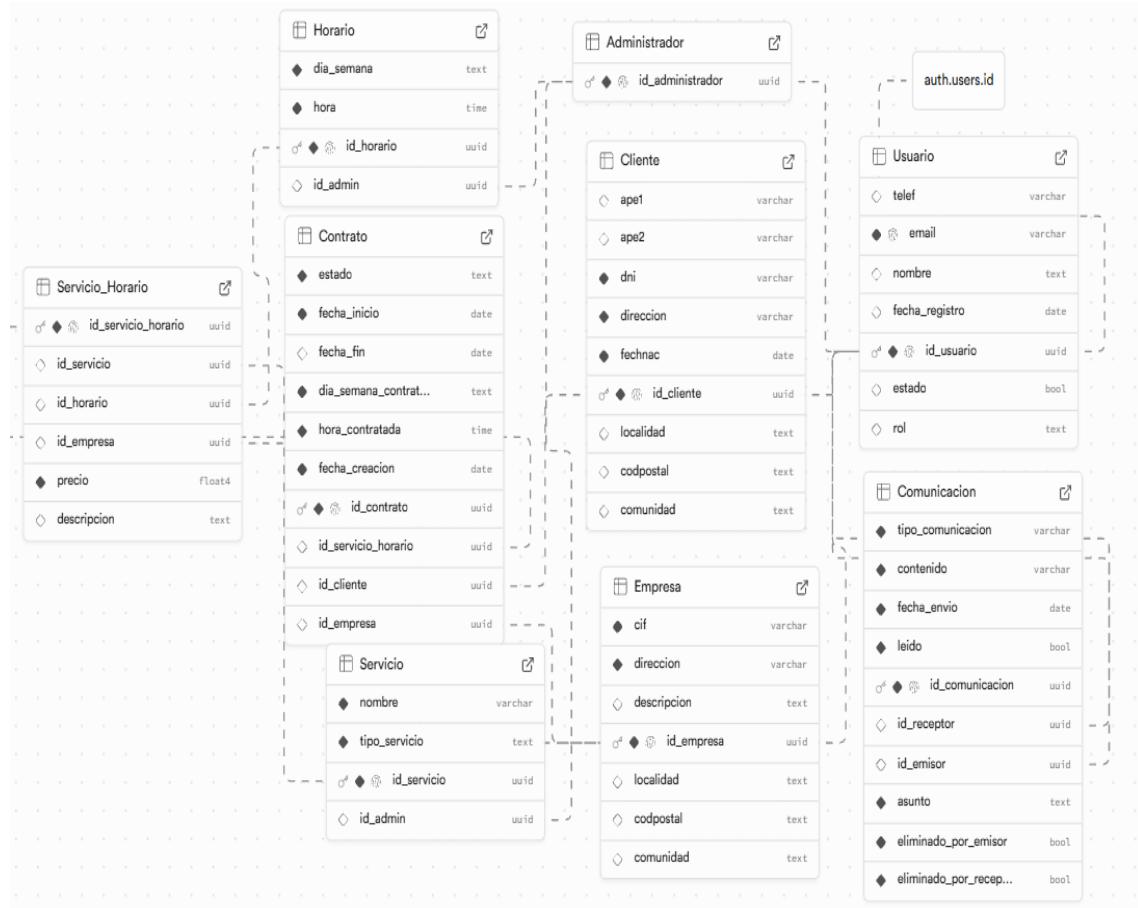


Ilustración 34: Tabla Supabase.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

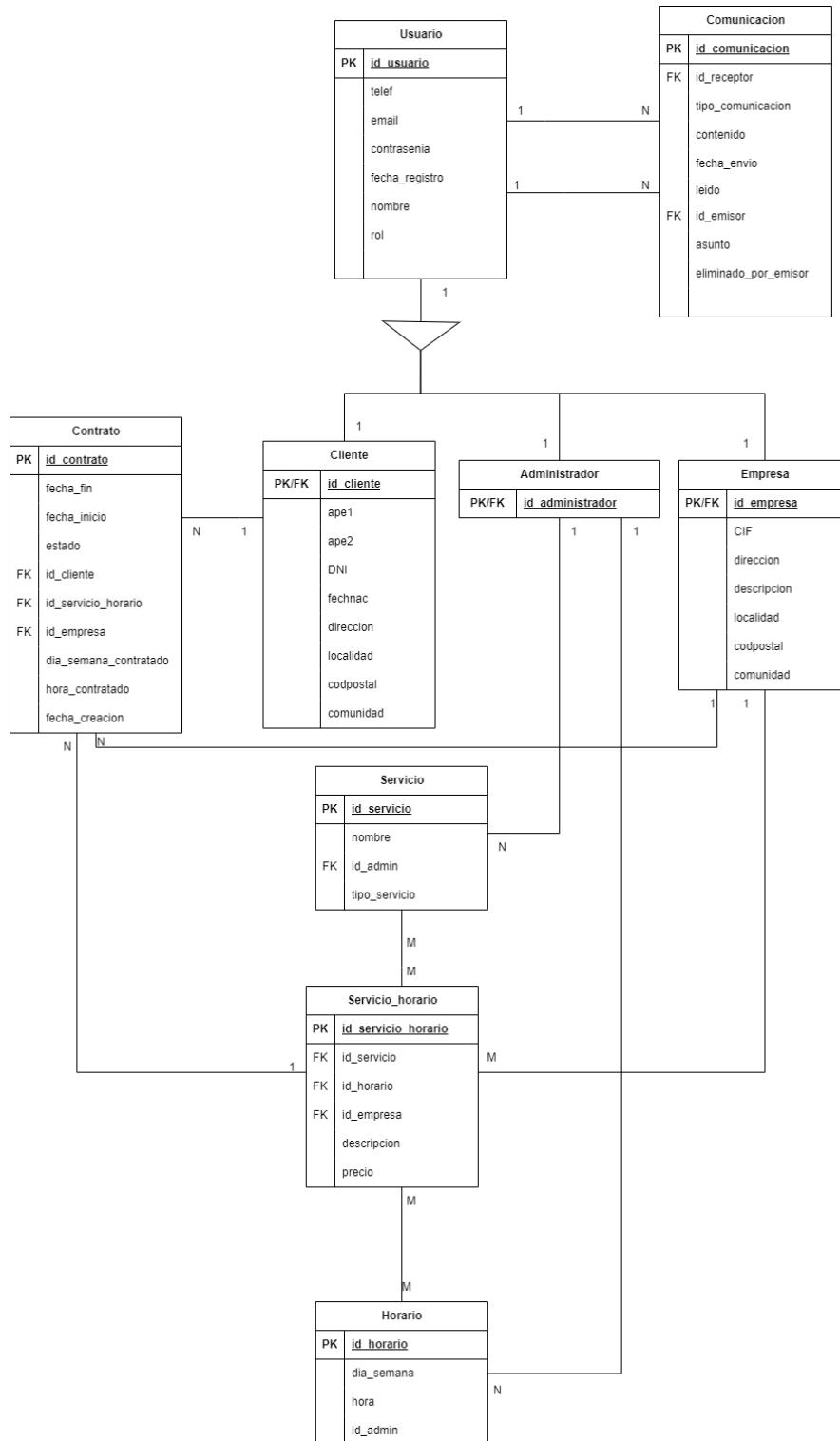


Ilustración 35: MER.

3.8. Hito: Presentación Intermedia y Feedback (13.01.26)

Se realizó el hito de control con la defensa del MVP funcional. El feedback más crítico se centró en la **UX de Seguridad**.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- *Problema:* Los botones de "Eliminar Contrato" y "Baja de Usuario" estaban demasiado accesibles y no pedían confirmación suficiente, provocando borrados accidentales durante la demo.
- *Acción Correctiva:* Se diseñó el componente CancelModal. Este modal no es una simple alerta del navegador; es un componente de Angular Material que intercepta el evento, oscurece el fondo (*backdrop*) y exige una acción consciente del usuario antes de llamar a la API de borrado.

Confirmación de Cancelación de Contrato

¡Atención! Está a punto de cancelar un contrato de servicio. Al confirmar, el contrato seleccionado será anulado en la plataforma CuidaDos. ¿Qué sucederá ahora?

- **Registro del Cambio:** La cancelación se registrará inmediatamente en el sistema y se actualizará su calendario.
- **Comunicación:** Recibirá una notificación de confirmación por correo o dentro de la aplicación.
- **Notificación a la Otra Parte.**

Si tiene alguna pregunta o surge un conflicto

Continuar con el contrato

Cancelar contrato

Ilustración 36: Modal I.

×

¿Seguro que quieres darte de baja?

Perderás el acceso a todas las funcionalidades de la app. Esta acción no se puede deshacer.

Al continuar, tu cuenta será desactivada y no podrás acceder nuevamente con este perfil.

Continuar registrado

Cancelar registro

Ilustración 37: Modal II.

3.9. Hito: Elección de Tecnologías (27.01.26)

La elección del stack tecnológico se basó en la sostenibilidad y el rendimiento a largo plazo:

1. **Angular 17+ (Signals):** Se prefirió sobre React por su opinión fuerte. La introducción de **Signals** eliminó la necesidad de usar librerías complejas de gestión de estado como NgRx para este tamaño de proyecto, y mejoró el rendimiento al eliminar ciclos innecesarios de detección de cambios (Zone.js).
2. **Vercel:** Elegido por su red global de **Edge Caching**. Esto asegura que los archivos estáticos (JS, CSS, Imágenes) se sirvan desde el servidor más cercano al usuario, reduciendo la latencia de carga inicial a <100ms.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

3. **Supabase Auth:** Delegar la autenticación ahorró aproximadamente 40 horas de desarrollo en gestión de sesiones, encriptación, recuperación de contraseñas y seguridad JWT.

Matriz Comparativa (Supabase vs Firebase)

Criterio Técnico	Supabase (Seleccionado)	Google Firebase (Descartado)	Impacto en el Proyecto
Modelo de Datos	Relacional (SQL)	Documental (NoSQL)	Permite validar DNI y relaciones estrictas sin duplicar datos.
Consultas Complejas	Potentes (Joins) Permite cruzar tablas (Usuario + Contrato + Empresa).	Limitadas No existen los "Joins". Requiere múltiples peticiones.	El Dashboard carga en 1 sola petición en lugar de 10.
Integridad de Datos	Alta (Foreign Keys) Si borras un usuario, se borran sus datos (Cascada).	Nula Responsabilidad del programador. Riesgo de "datos huérfanos".	Garantiza que no existan contratos sin usuario asignado.
Vendor Lock-in	Bajo (Open Source) Puedes exportar tu DB y llevártela a AWS u otro servidor.	Alto (Propietario) Si Google sube precios, es muy difícil migrar.	Asegura la sostenibilidad futura del proyecto.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Realtime	WebSockets (Push) Basado en cambios del log de la base de datos (WAL).	Snapshot Listeners Muy eficiente pero costoso a gran escala.	Permite el Chat en vivo sin configuración de servidores.
Autenticación	Estándar JWT Integrado con las políticas de seguridad de la DB (RLS).	Propia Segura, pero separada de la lógica de datos.	Seguridad unificada: "Quien no tiene token, no lee datos".

Tabla 7: Tabla comparativa.

Razón de la elección de supabase

La elección de Supabase (basado en PostgreSQL) frente a Firebase (basado en base de datos documental propietaria) responde a una necesidad estructural crítica del dominio de negocio: la Integridad Referencial.

A) Complejidad de las Relaciones y Normalización

En una aplicación de tipo "Red Social" o "Chat", los datos pueden ser efímeros y estar aislados. Sin embargo, CuidaDos es una plataforma transaccional. El modelo de datos presenta relaciones cruzadas de alta cardinalidad:

- Una *Empresa* oferta múltiples *Servicios*.
- Un *Servicio* genera múltiples *Contratos*.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- Un *Contrato* vincula a un *Cliente* y a una *Empresa*, generando *Facturas* y *Reseñas*.

En un entorno NoSQL (Firebase), la gestión de estas relaciones obligaría a la desnormalización de los datos. Esto implica que, para mostrar el historial de contratos de un usuario, tendríamos que copiar los datos de la empresa (Nombre, CIF, Dirección) dentro de cada documento de contrato.

- El Riesgo de Inconsistencia: Si una empresa actualizase su razón social o su dirección, el sistema tendría que lanzar un proceso masivo (Cloud Function) para buscar y actualizar miles de documentos históricos dispersos. Si este proceso fallase a la mitad, la base de datos quedaría en un estado corrupto e inconsistente.
- La Solución PostgreSQL: Al utilizar un modelo relacional normalizado, los contratos simplemente almacenan una referencia (Foreign Key) al ID de la empresa. Cualquier cambio en la entidad Empresa se propaga instantáneamente a toda la plataforma sin coste computacional ni riesgo de error.

B) Transaccionalidad ACID

Dado que la aplicación gestiona acuerdos legales (contratos de servicios), es imperativo garantizar las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). PostgreSQL ofrece soporte nativo para transacciones complejas. Si un proceso de contratación falla en el paso final (ej: error al crear la notificación), el sistema puede realizar un ROLLBACK completo, deshaciendo la creación del contrato y evitando estados "zombies" o registros huérfanos, algo que en Firebase requiere una implementación manual de lógica de compensación propensa a errores.

Seguridad Perimetral: Row Level Security (RLS)

Uno de los diferenciadores más potentes de la arquitectura elegida es el desplazamiento de la lógica de seguridad: del servidor de aplicaciones (*Middleware*) al motor de base de datos.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

En un desarrollo tradicional (o con Firebase Rules, que suelen ser limitadas), la seguridad depende de que el programador no olvide incluir la verificación en cada *endpoint* de la API. En CuidaDos, se ha implementado Row Level Security (RLS).

- Principio de Defensa en Profundidad: Las políticas de seguridad viven junto a los datos.
 - *Ejemplo Práctico:* Se ha definido una política SQL que establece:
CREATE POLICY "Solo ver propios" ON contratos FOR SELECT
USING (auth.uid() = id_cliente).
- Inviolabilidad: Esto significa que, incluso si un atacante lograse saltarse las validaciones del Frontend o comprometer el cliente API de JavaScript, el motor de base de datos rechazaría la consulta a nivel de núcleo, devolviendo un conjunto de datos vacío. La seguridad no es una capa añadida, es intrínseca al dato.

Potencia de Consulta y Filtrado

El sistema de búsqueda de cuidadores requiere filtros multicriterio complejos (ej: "*Buscar enfermeros en Mérida, disponibles por la mañana, con valoración > 4 estrellas y precio < 15€*").

- Limitación de Firebase: Las bases de datos documentales son excelentes para leer datos por ID, pero deficientes para consultas complejas. Realizar este tipo de filtrado requeriría descargar todos los registros al cliente y filtrarlos en memoria (ineficiente) o crear índices compuestos combinatorios para cada posible filtro, lo cual es insostenible.
- Potencia de SQL: Supabase permite utilizar toda la potencia del lenguaje SQL y las extensiones de PostgreSQL (como PostGIS para geolocalización), permitiendo resolver esa consulta compleja en milisegundos y devolviendo al cliente solo los datos estrictamente necesarios.

Soberanía del Dato y "Vendor Lock-in"

Finalmente, se ha considerado la sostenibilidad a largo plazo del proyecto.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- El problema del "Jardín Vallado": Firebase es una tecnología propietaria de Google. Si se decidiera migrar la plataforma en el futuro, sería necesario reescribir todo el código de backend y migrar datos desde un formato JSON propietario, lo que conlleva un alto coste de salida (*Vendor Lock-in*).
- Estándares Abiertos: Supabase es, en esencia, PostgreSQL envuelto en herramientas de código abierto. Esto garantiza la portabilidad total: la base de datos completa puede ser exportada (`pg_dump`) y desplegada en cualquier otro proveedor (AWS, Azure, DigitalOcean) o incluso en un servidor local propio (Self-Hosted con Docker), asegurando que el proyecto nunca será rehén de su proveedor de infraestructura.

ARTEFACTO: Datos para tabla comparativa

Característica	Angular (Elegido)	React
Estructura	Framework Completo (MVC)	Librería (Solo Vista)
Routing	Nativo / Incluido	Requiere Librería Externa
Gestión Estado	Signals (Nativo)	Hooks / Redux
Tipado	TypeScript Estricto	Opcional

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Idoneidad TFG	Alta (Estándar Empresarial)	Alta (Startups)

Tabla 8: Tabla comparativa.

3.10. Hito: Presentación del Prototipo a la Empresa (13.01.26)

Desarrollo de la Sesión

Se llevó a cabo la reunión de control con los tutores del proyecto (actuando como Cliente/Empresa) para validar el estado del **Producto Mínimo Viable (MVP)**. El objetivo principal fue verificar que los flujos de navegación (User Journeys) cumplían con las expectativas generadas en la fase de análisis.

Durante la sesión, se realizó una demostración en vivo cubriendo los siguientes puntos críticos:

- Registro y Onboarding:** Se demostró la creación de cuentas diferenciadas (Cliente vs. Empresa) y cómo el sistema adapta el menú de navegación (Navbar) dinámicamente según el rol detectado.
- Ciclo de Contratación:** Se simuló un flujo completo: un cliente busca una empresa, selecciona un servicio, la empresa recibe la solicitud en su panel y la acepta.
- Persistencia:** Se verificó en tiempo real, accediendo al panel de Supabase, que los datos introducidos en el formulario web se reflejaban correctamente en las tablas PostgreSQL.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Feedback Recibido

La validación fue positiva en cuanto a la arquitectura y fluidez de la aplicación (gracias al uso de Angular Signals). Sin embargo, se solicitaron ajustes menores antes del cierre:

- Mejorar la visibilidad de los mensajes de error en los formularios (feedback visual).
- Ajustar el tamaño de la fuente en la vista móvil del chat.

3.11. Hito: Ajustes Post-Presentación y Refactorización (20.01.26)

Tras la aprobación condicional del prototipo, se dedicó esta semana a la fase de "**Pulido y Robustez**". No se añadieron nuevas funcionalidades, sino que se aseguró la estabilidad del código existente.

Acciones de Ingeniería Realizadas:

1. **Saneamiento de Formularios:** Se implementaron validadores síncronos personalizados (CustomValidators) para asegurar la integridad de los datos. Por ejemplo, se añadió una validación estricta para el formato del CIF de las empresas.
2. **Manejo de Errores Global:** Se configuró un interceptor HTTP (ErrorInterceptor) para capturar fallos de red. En lugar de dejar que la aplicación se congele, ahora muestra notificaciones emergentes (Toasts) informando al usuario del problema.
3. **Limpieza de Código (Refactoring):** Se eliminaron console.log residuales, código comentado no utilizado y se optimizaron las importaciones de Angular para reducir el tamaño final del *bundle*.
 - Captura o bloqueo de código del ErrorInterceptor o de un validador personalizado.

3.12. Hito: Confirmación del Stack Tecnológico (27.01.26)

Análisis de Arquitectura Definitiva

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

En esta fase, con el desarrollo prácticamente concluido, se redactó la justificación técnica final para la memoria del proyecto. Se confirmó que las elecciones tecnológicas iniciales fueron acertadas basándose en las métricas de rendimiento obtenidas.

Justificación de la Elección (Angular + Supabase):

- **Angular 17+:** Se valida su elección frente a React. La estructura estricta de Angular (Módulos, Servicios, Inyección de Dependencias) ha facilitado el mantenimiento del código a medida que el proyecto crecía. El uso de **Signals** ha demostrado reducir la complejidad de la gestión de estado en comparación con Redux o NgRx.
- **Supabase:** Ha permitido desarrollar el Backend en un tiempo récord. La funcionalidad de *Realtime* (Websockets) funcionó "out-of-the-box" para el chat, ahorrando semanas de configuración de servidores Socket.io.

Comparativa de Rendimiento

Se realizaron pruebas de carga comparando el tiempo de respuesta.

Característica	Angular (Implementado)	React (Alternativa)	Impacto en TFG
Estructura	MVC Completo	Librería de Vista	Angular facilita el orden.
Routing	Nativo	Externo	Menos dependencias.
Estado	Signals	Hooks	Mejor rendimiento.

Tipado	TypeScript Estricto	Opcional	Menos errores en runtime.
---------------	---------------------	----------	---------------------------

Tabla 9: Comparativa de rendimiento.

3.13. Hito: Estructuración de la Documentación (03.02.26)

Organización de la Memoria Técnica

Se procedió a la esquematización del documento final de Tesis (TFG). Dado el volumen de trabajo técnico realizado, fue necesario estructurar la información en capítulos lógicos que cumplieran con la normativa académica del IES Albarregas.

Estructura Definida:

1. **Bloque de Contexto:** Introducción, Objetivos y Planificación.
2. **Bloque de Ingeniería:** Requisitos (SRS), Análisis y Diseño UI/UX.
3. **Bloque de Construcción:** Diario de Ejecución y Código Fuente.
4. **Bloque de Calidad:** Manuales y Pruebas.

Se preparó el entorno de redacción utilizando Microsoft Word, configurando estilos automáticos para la generación de índices dinámicos y bibliografía cruzada.

3.14. Hito: Desarrollo de Manuales de Usuario y Técnico (10.02.26)

Estrategia de Transferencia de Conocimiento

Para garantizar la usabilidad y mantenibilidad del software "CuidaDos" a largo plazo, se desarrollaron dos líneas de documentación independientes:

A) Manual de Usuario (Target: Cliente Final)

Redactado en lenguaje natural, evitando tecnicismos. Se centra en "Tareas":

- *Guía Rápida:* Cómo registrarse.
- *Contratación:* Cómo buscar una empresa y solicitar servicio.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- *Solución de Problemas:* Qué hacer si olvido la contraseña.
- *Cancelar una contratación.*

Y así cada uno de los servicios.

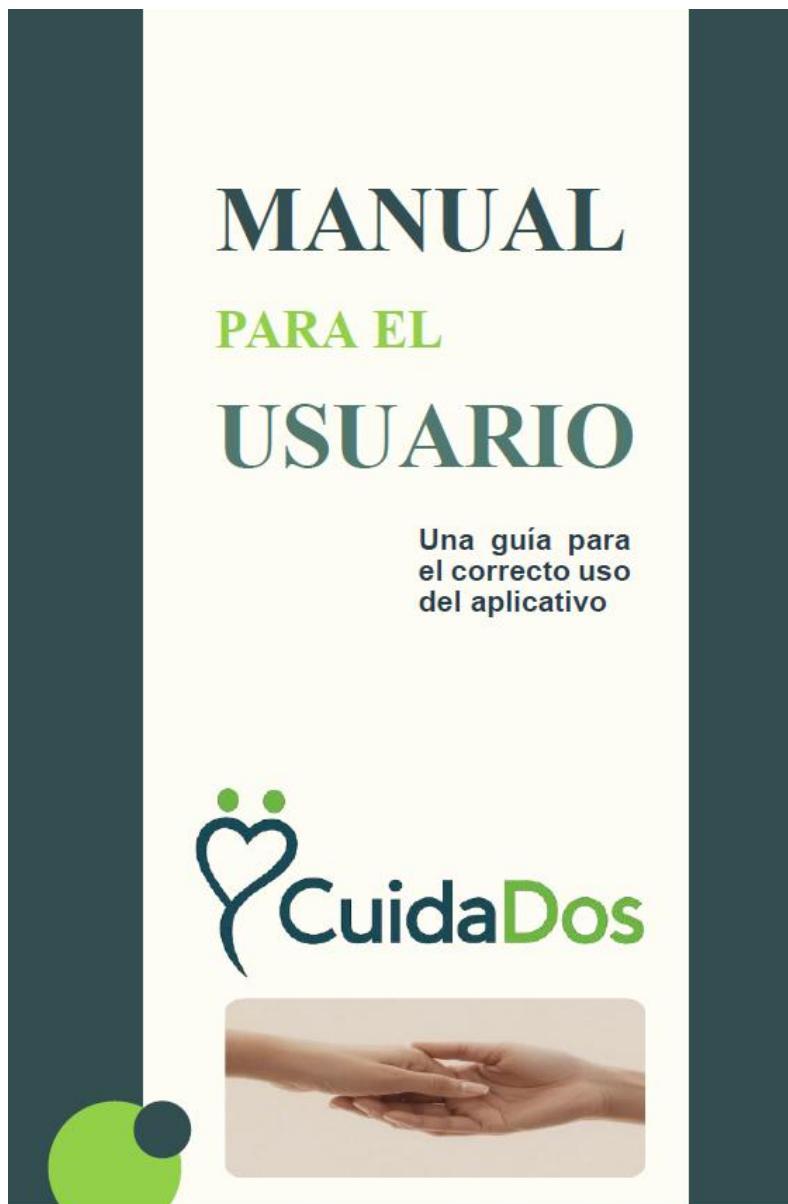


Ilustración 38: Portada Manual I.

Índice

1. Introducción.....	1
2. Tabla de Roles.....	1
3. Gestión de Cuenta y Acceso	1
3.1 Explorando la Plataforma (Landing Page)	1
3.2 Registro de Nuevos Usuarios	3
3.2.1 Usuarios Particulares	3
3.2.2 Empresas.....	4
3.2.3 Administradores.....	5
3.3 Inicio de Sesión y Recuperación	5
3.3.1 Inicio de sesión	5
3.3.2 Recuperación de contraseña	6
3.3.3 Reenvío de email de registro	7
4. Guía para el Cliente (Usuario Particular)	8
4.1 Búsqueda de Servicios.....	8
4.2 Contratación	8
4.3 Calendario Personal	10
4.3.1 Eliminar un contrato	11
4.4 Contrataciones	11
4.4.2 Cancelar.....	13
4.5 Modificar Perfil	14
4.6. Darse de baja	15
5. Guía para la Empresa.....	15

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

5.1 Gestión de Catálogo.....	15
5.1.1 Añadir Servicios	15
5.1.2 Modificación de los servicios.....	17
5.1.3 Eliminación de un servicio	17
5.2 Contrataciones	18
5.2.1 Ver detalles del contrato	18
5.2.2 Eliminar un contrato.....	19
5.3 Calendario Personal	20
5.3.1 Eliminación de los contratos	20
6. Guía para el Administrador	21
6.1 Home (página inicial) del administrador.....	21
6.2 Mantenimiento Global (CRUD):	22
6.2.1. Servicios	22
6.2.1.3 Eliminar servicios	23
6.2.1.4 Buscar servicios.....	24
6.2.2 Horarios	24
6.2.2.1 Crear horarios	24
6.2.2.2 Modificar horarios	25
6.2.2.3 Eliminar horarios	26
6.2.2 Gestión de Usuarios y Empresas	27
6.2.3 Dashboard.....	30
7. Navegación General y Soporte	30
7.1 Landing Page, Navbar y Footer.....	30
Internacionalización.....	30
7.1.1 Barra de Navegación (Navbar).....	30
7.1.2 Footer.....	32
7.2 Comunicación y estados	32
7.2.1 Mensajería	32
7.2.2 Notificaciones	34
7.2.3 Código de Colores:	34
7.3 Elementos de Control	34
7.3.1 Botón de Retroceso (Back).....	34

Ilustración 39: Índice Manual I.

B) Manual Técnico (Target: Administrador/Desarrollador)

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Documento orientado al despliegue y mantenimiento. Incluye:

- Instalación de dependencias (npm install).
- Configuración de variables de entorno (.env).
- Explicación del esquema de base de datos y políticas de seguridad RLS.

Cada una de las cuestiones técnicas necesarias.

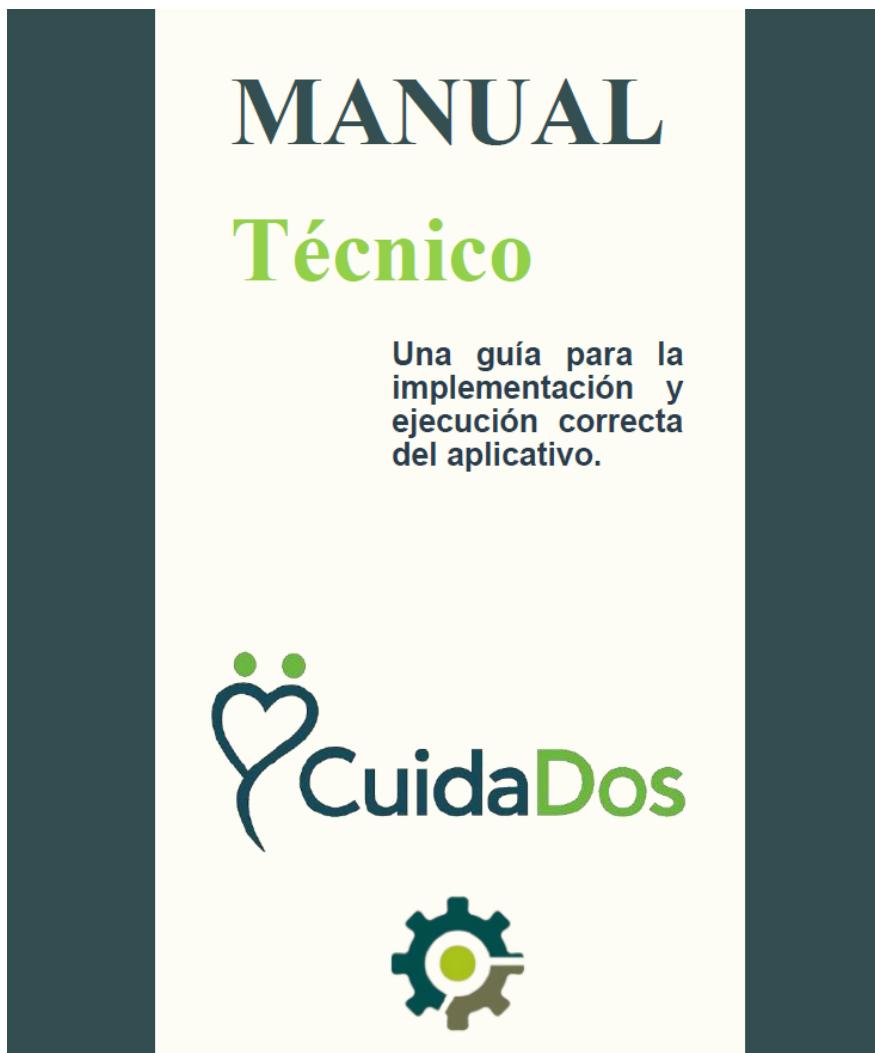


Ilustración 40: Portada Manual II.

ÍNDICE

1. INTRODUCCIÓN.....	1
2. OBJETIVOS	1
2.1 Objetivos técnicos	1
3. REQUISITOS DEL SISTEMA.....	2
3.1 Requisitos para el Cliente (Usuario Final)	2
3.2 Requisitos para el Servidor de Desarrollo	4
4. HERRAMIENTAS Y STACK TECNOLÓGICO	8
4.1 Frontend: Core y Framework	8
4.2 Estilos e Interfaz de Usuario (UI).....	9
4.2.1 Tailwind CSS (Motor de Estilos).....	9
4.2.2 Angular Material (Componentes UI).....	9
4.2.3 Lucide Angular (Iconografía)	10
4.3 Backend & Persistencia (BaaS).....	10
4.4 Librerías Auxiliares	10
4.5 Testing.....	11
4.5.1 Karma & Jasmine: Pruebas Unitarias.....	11
5.2 Cypress: Pruebas de Extremo a Extremo (E2E).....	13
5.3 Postman: Validación de API y Contratos de Datos.....	14
5.3.1 Validación de API con Postman.....	16
5. GUÍA DE INSTALACIÓN (ENTORNO LOCAL)	18
5.1 Instalación de Prerrequisitos (Paso a Paso)	18
5.2 Entorno de desarrollo.....	20
6. GUÍA DE DESPLIEGUE (ENTORNO CLOUD - VERCCEL).....	21
6.1 Configuración de Build	21
6.2 Gestión de Rutas (SPA)	21
6.3 Variables de Entorno en Producción.....	22
7. ARQUITECTURA DE BASE DE DATOS (SUPABASE)	22
7.1 Aislamiento de Capas y Seguridad de Esquemas	22

Ilustración 41: Indice Manual II.

3.15. Hito: Análisis de Opciones de Despliegue (24.02.26)

Estudio de Alternativas de Hosting

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Antes de la puesta en producción, se evaluaron distintas plataformas gratuitas capaces de alojar una *Single Page Application* (SPA) de Angular con rutas dinámicas.

Alternativas Evaluadas:

1. **GitHub Pages:** Descartado. Requiere configuraciones complejas ("hacks") para manejar el enrutamiento profundo de Angular (error 404 al recargar páginas internas).
2. **Netlify:** Evaluado positivamente, pero con límites en el ancho de banda de la capa gratuita.
3. **Vercel: OPCIÓN SELECCIONADA.**
 - **Ventajas:** Detección automática del framework Angular, integración nativa con el repositorio de GitHub (CI/CD) y red global de distribución de contenidos (Edge Network) que asegura baja latencia.

3.16. Hito: Pruebas Finales y Despliegue en Producción (del 03.03.26 al 17.03.26)

Ejecución de Pruebas (Staging)

Antes de subir el código a la nube, se configuró un entorno local de preproducción. Se ejecutó la batería de pruebas unitarias mediante el comando ng test, verificando que los componentes críticos (Login, Registro, Buscador) funcionaban sin regresiones.

Despliegue Continuo (CI/CD)

Se vinculó la rama main del repositorio GitHub con Vercel. Esto habilitó el despliegue automático: cada vez que se hace un *commit*, Vercel descarga el código, instala las dependencias, compila la aplicación (ng build --configuration production) y la publica.

Resultado: La aplicación quedó accesible públicamente bajo el dominio seguro HTTPS proporcionado por la plataforma.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

CAPÍTULO 4: PLANIFICACIÓN Y ANÁLISIS DE REQUISITOS

4.1. Catálogo Completo de Requisitos Funcionales (SRS)

A continuación, se detallan las especificaciones técnicas de cada funcionalidad del sistema **CuidaDos**.

4.1.1. Módulo de Gestión de Usuarios (Auth)

Requisitos Funcionales: Módulo de CLIENTES

Ficha Técnica: RF-C01 Registro de Usuarios

Atributo	Especificación Técnica
Identificador	RF-C01
Nombre	Alta de Cuenta Cliente
Prioridad	Crítica
Actor Principal	Usuario Visitante
Descripción	Permite a cualquier persona física crear una cuenta mediante un formulario, pasando a ser cliente de la plataforma.
Datos Requeridos	<ul style="list-style-type: none">- Nombre y Apellidos.- DNI (con validación de letra).- Email y Contraseña.- Dirección, Localidad, CP y Comunidad Autónoma.
Validaciones Técnicas	<ul style="list-style-type: none">- Frontend: Algoritmo de Módulo 23 para verificar la letra del DNI (RegisterForm).- Backend: Verificación de unicidad de email mediante RPC <code>email_exists</code>.- Edad: Verificación de mayoría de edad (>18 años).

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Componente Asociado	src/app/pages/registerform/registerform.ts
---------------------	--

Tabla 10: RF.

Ficha Técnica: RF-C02 Inicio de Sesión

Atributo	Especificación Técnica
Identificador	RF-C02
Nombre	Autenticación de Cliente
Prioridad	Crítica
Actor Principal	Cliente
Descripción	Los clientes pueden iniciar sesión usando correo electrónico y contraseña para acceder a su área privada.
Flujo	1. Envío de credenciales a Supabase Auth. 2. Recepción de Token JWT. 3. Carga polimórfica del perfil (Unión de tabla Usuario + Cliente).
Seguridad	Las contraseñas viajan encriptadas bajo protocolo HTTPS. No se almacenan en texto plano.
Componente Asociado	src/app/core/services/auth.service.ts

Tabla 11: RF.

Ficha Técnica: RF-C03 Búsqueda de Servicios

Atributo	Especificación Técnica
Identificador	RF-C03
Nombre	Buscador y Filtrado
Prioridad	Alta
Actor Principal	Cliente Autenticado
Descripción	Los clientes pueden buscar servicios ofrecidos por empresas usando filtros dinámicos (categoría, ubicación, texto libre).
Implementación	Se utiliza un sistema de filtrado reactivo en memoria. Los datos se cargan optimizados con un JOIN entre Empresa, Usuario y Servicio.
Interfaz	Barra de búsqueda con <i>debounce</i> para evitar sobrecarga de peticiones.
Componente Asociado	src/app/components/searchbar/searchbar.ts

Tabla 12: RF.

Ficha Técnica: RF-C04 Contratar Servicio

Atributo	Especificación Técnica
----------	------------------------

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Identificador	RF-C04
Nombre	Solicitud de Contratación
Prioridad	Alta
Actor Principal	Cliente Autenticado
Descripción	Permite formalizar la contratación de un servicio específico ofrecido por una empresa.
Postcondición	Se genera un registro en la tabla Contrato con estado inicial pendiente.
Restricciones	No se puede contratar si el usuario tiene pagos pendientes o si la empresa ha desactivado el servicio temporalmente.
Componente Asociado	BusinessProfile y ContractService.

Tabla 13: RF.

Ficha Técnica: RF-C05 Baja de Usuario

Atributo	Especificación Técnica
Identificador	RF-C05
Nombre	Eliminación de Cuenta (Cliente)

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Prioridad	Media
Actor Principal	Cliente
Descripción	Permite a los clientes eliminar su cuenta de la plataforma.
Mecanismo	Se aplica un Borrado Lógico (estado = false) para mantener la integridad histórica, o un borrado físico mediante RPC eliminar_usuario_total si no hay dependencias legales.
Seguridad	Requiere confirmación mediante el componente CancelModal.
Componente Asociado	src/app/components/cancelmodal/cancelmodal.ts

Tabla 14: RF.

Ficha Técnica: RF-C06 Calendario Personal

Atributo	Especificación Técnica
Identificador	RF-C06
Nombre	Gestión de Mis Contratos

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Prioridad	Media
Actor Principal	Cliente
Descripción	Visualización cronológica o en lista de los servicios contratados activos y pendientes.
Datos Mostrados	Fecha de inicio, Empresa proveedora, Estado del servicio y Precio acordado.
Interfaz	Vista de lista detallada en el perfil del usuario.

Tabla 15: RF.

Ficha Técnica: RF-C07 Notificaciones

Atributo	Especificación Técnica
Identificador	RF-C07
Nombre	Sistema de Alertas
Prioridad	Media
Actor Principal	Cliente

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Descripción	Recepción de avisos visuales ante cambios de estado en contratos o nuevos mensajes.
Tecnología	Supabase Realtime. El cliente se suscribe a cambios en la tabla Comunicacion.
Visualización	Badge rojo en el Navbar y lista desplegable de notificaciones.
Componente Asociado	ComunicationService

Tabla 16: RF.

Ficha Técnica: RF-C08 Mensajería

Atributo	Especificación Técnica
Identificador	RF-C08
Nombre	Chat Bidireccional
Prioridad	Alta
Actor Principal	Cliente
Descripción	Los clientes podrán enviar y recibir mensajes privados con las empresas contratadas.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Características	<ul style="list-style-type: none">- Envío en tiempo real.- Historial persistente en base de datos.- Distinción visual entre emisor y receptor.
Componente Asociado	src/app/pages/messages/messages.ts

Tabla 17: RF.

Ficha Técnica: RF-C09 Cancelar Contrato

Atributo	Especificación Técnica
Identificador	RF-C09
Nombre	Desistimiento de Servicio
Prioridad	Media
Actor Principal	Cliente

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Descripción	Permite cancelar un contrato vigente.
Lógica	Cambio de estado en la tabla Contrato a cancelado. Dispara una notificación automática a la Empresa afectada.
Componente Asociado	CancelModal

Tabla 18: RF.

Ficha Técnica: RF-C10 Modificar Perfil

Atributo	Especificación Técnica
Identificador	RF-C10
Nombre	Edición de Datos Personales
Prioridad	Baja
Actor Principal	Cliente

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Descripción	Actualización de dirección, teléfono o avatar.
Restricciones	Datos sensibles como el DNI o el Email requieren procesos de verificación adicionales o contacto con soporte.
Componente Asociado	ModifyProfileComponent

Tabla 19: RF.

Requisitos Funcionales: Módulo de EMPRESAS

Ficha Técnica: RF-E01 Registro de Empresa

Atributo	Especificación Técnica
Identificador	RF-E01
Nombre	Alta de Proveedor
Prioridad	Crítica
Actor Principal	Empresa (No registrada)
Descripción	Registro especializado para entidades jurídicas.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Datos Específicos	<ul style="list-style-type: none">- Razón Social.- CIF (Validado con algoritmo de control).- Categoría del sector.- Descripción corporativa.
Componente Asociado	RegisterForm (Modo isUser = false)

Tabla 20: RF.

Ficha Técnica: RF-E02 Inicio de Sesión

Atributo	Especificación Técnica
Identificador	RF-E02
Nombre	Autenticación Corporativa
Prioridad	Crítica
Actor Principal	Empresa
Descripción	Acceso seguro al panel de gestión empresarial.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Implementación	Reutiliza la lógica de AuthService pero redirige al Dashboard de Empresa tras el login.
----------------	---

Tabla 21: RF.

Ficha Técnica: RF-E03 Modificar Perfil

Atributo	Especificación Técnica
Identificador	RF-E03
Nombre	Gestión de Identidad Corporativa
Prioridad	Media
Actor Principal	Empresa
Descripción	Edición de la descripción pública, página web y logotipo de la empresa.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Impacto	Los cambios se reflejan inmediatamente en el buscador de servicios para los clientes.
---------	---

Tabla 22: RF.

Ficha Técnica: RF-E04 Calendario Empresarial

Atributo	Especificación Técnica
Identificador	RF-E04
Nombre	Gestión de Agenda
Prioridad	Media
Actor Principal	Empresa
Descripción	Visualización centralizada de todos los servicios programados con diferentes clientes.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Tecnología	Consultas agregadas a la tabla Contrato filtradas por id_empresa.
------------	---

Tabla 23: RF.

Ficha Técnica: RF-E05 Mensajería

Atributo	Especificación Técnica
Identificador	RF-E05
Nombre	Chat de Soporte/Gestión
Prioridad	Alta
Actor Principal	Empresa
Descripción	Canal para resolver dudas de clientes o coordinar los servicios contratados.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Funcionalidad	Bandeja de entrada con múltiples conversaciones separadas por Cliente.
---------------	--

Tabla 24: RF.

Ficha Técnica: RF-E06 Baja de Empresa

Atributo	Especificación Técnica
Identificador	RF-E06
Nombre	Cese de Actividad
Prioridad	Baja
Actor Principal	Empresa
Descripción	Eliminación de la cuenta y retirada de todos los servicios ofertados del mercado público.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Consecuencia	Cancelación automática de todos los contratos pendientes.
--------------	---

Tabla 25: RF.

Ficha Técnica: RF-E07 Modificar Servicio-Horario

Atributo	Especificación Técnica
Identificador	RF-E07
Nombre	Edición de Oferta
Prioridad	Media
Actor Principal	Empresa

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Descripción	Ajuste de precios, descripciones o franjas horarias disponibles para un servicio concreto.
-------------	--

Tabla 26: RF.

Ficha Técnica: RF-E08 Cancelar Servicio-Horario

Atributo	Especificación Técnica
Identificador	RF-E08
Nombre	Retirada de Oferta
Prioridad	Media
Actor Principal	Empresa
Descripción	La empresa deja de ofrecer un servicio específico sin eliminar su cuenta.
Componente	CancelModal (Modo eliminarServicio).

Tabla 27: RF.

Ficha Técnica: RF-E09 Añadir Servicio-Horario

Atributo	Especificación Técnica
Identificador	RF-E09
Nombre	Publicación de Nuevos Servicios
Prioridad	Alta
Actor Principal	Empresa
Descripción	Selección de servicios del catálogo maestro para añadirlos a la oferta de la empresa.

Tabla 28: RF.

Ficha Técnica: RF-E10 Notificaciones

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Atributo	Especificación Técnica
Identificador	RF-E10
Nombre	Alertas de Negocio
Prioridad	Media
Actor Principal	Empresa
Descripción	Aviso inmediato ante nuevos contratos ("Leads") o cancelaciones por parte del cliente.

Tabla 29: RF.

Ficha Técnica: RF-E11 Cancelar Contrato

Atributo	Especificación Técnica

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Identificador	RF-E11
Nombre	Rechazo de Servicio
Prioridad	Media
Actor Principal	Empresa
Descripción	Capacidad de la empresa para rescindir un contrato por motivos de fuerza mayor o falta de disponibilidad.

Tabla 30: RF.

Requisitos Funcionales: Módulo de ADMINISTRACIÓN

Ficha Técnica: RF-A01 Registro de Admin

Atributo	Especificación Técnica
Identificador	RF-A01

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Nombre	Alta de Privilegios
Prioridad	Baja
Actor Principal	Super-Admin
Descripción	Solo un administrador existente puede crear otros administradores. No existe registro público.
Método	AuthService.registerByAdmin() con flag created_by_admin.

Tabla 31: RF.

Ficha Técnica: RF-A02 Inicio de Sesión Admin

Atributo	Especificación Técnica
Identificador	RF-A02
Nombre	Acceso al Backoffice

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Prioridad	Crítica
Actor Principal	Administrador
Descripción	Login unificado. El sistema detecta el rol admin y redirige al Dashboard de gestión.

Tabla 32: RF.

Ficha Técnica: RF-A03 CRUD Clientes

Atributo	Especificación Técnica
Identificador	RF-A03
Nombre	Gestión de Usuarios Particulares
Prioridad	Media

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Actor Principal	Administrador
Descripción	Listado, edición y eliminación de usuarios clientes.
Componente	TableCrudAdmin (Modo cliente).
Funciones	Filtrado por nombre/email y paginación desde servidor.

Tabla 33: RF.

Ficha Técnica: RF-A04 CRUD Empresas

Atributo	Especificación Técnica
Identificador	RF-A04
Nombre	Gestión de Proveedores
Prioridad	Media

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Actor Principal	Administrador
Descripción	Control total sobre las empresas registradas (verificar CIF, corregir datos, banear).
Componente	TableCrudAdmin (Modo empresa).

Tabla 34: RF.

Ficha Técnica: RF-A05 Mensajería

Atributo	Especificación Técnica
Identificador	RF-A05
Nombre	Comunicación de Soporte
Prioridad	Baja

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Actor Principal	Administrador
Descripción	Capacidad de enviar mensajes oficiales a cualquier usuario o empresa para resolución de disputas.

Tabla 35: RF.

Ficha Técnica: RF-A06 Dashboard

Atributo	Especificación Técnica
Identificador	RF-A06
Nombre	Cuadro de Mandos
Prioridad	Media
Actor Principal	Administrador

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Descripción	Visualización gráfica de KPIs del sistema.
Métricas	Usuarios totales, Altas recientes, Distribución de roles.
Implementación	AnalyticsService + Chart.js.

Tabla 36: RF.

Ficha Técnica: RF-A07 Notificaciones Admin

Atributo	Especificación Técnica
Identificador	RF-A07
Nombre	Alertas de Sistema
Prioridad	Baja

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Actor Principal	Administrador
Descripción	Avisos sobre incidencias técnicas o reportes de usuarios.

Tabla 37: RF.

Ficha Técnica: RF-A08 Baja de Admin

Atributo	Especificación Técnica
Identificador	RF-A08
Nombre	Renuncia de Rol
Prioridad	Baja

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Actor Principal	Administrador
Descripción	Un administrador puede eliminar su propia cuenta, perdiendo acceso al panel.

Tabla 38: RF.

Ficha Técnica: RF-A09 Modificar Perfil Admin

Atributo	Especificación Técnica
Identificador	RF-A09
Nombre	Edición de Datos Propios
Prioridad	Baja

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Actor Principal	Administrador
Descripción	Actualización de credenciales o datos de contacto del administrador.

Tabla 39: RF.

Ficha Técnica: RF-A10 CRUD Servicios

Atributo	Especificación Técnica
Identificador	RF-A10
Nombre	Gestión del Catálogo Maestro
Prioridad	Alta
Actor Principal	Administrador

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Descripción	Alta, baja y modificación de los tipos de servicios que la plataforma soporta (ej: Crear la categoría "Fisioterapia").
-------------	--

Tabla 40: RF.

Ficha Técnica: RF-A11 CRUD Horarios

Atributo	Especificación Técnica
Identificador	RF-A11
Nombre	Configuración de Franjas Temporales
Prioridad	Media
Actor Principal	Administrador

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Descripción	Definición de los bloques horarios estándar que las empresas pueden utilizar.
-------------	---

Tabla 41: RF.

Requisitos trasversales

Ficha Técnica: RF-COM04 Recuperación de Contraseña

Atributo	Especificación Técnica
Identificador	RF-COM04
Nombre	Restablecimiento de Credenciales
Prioridad	Alta
Actor Principal	Usuario Anónimo (Cliente o Empresa)
Descripción	Mecanismo de seguridad que permite a un usuario recuperar el acceso a su cuenta en caso de haber olvidado su contraseña, garantizando que solo el propietario del email pueda realizar el cambio.
Precondición	El usuario debe estar registrado previamente y tener un email válido asociado a su cuenta en auth.users.
Flujo Principal	<ol style="list-style-type: none">1. El usuario accede a la vista /recover-password.2. Introduce su correo electrónico.3. El sistema valida el formato del email.

	<p>4. Se llama a auth.resetPasswordForEmail().</p> <p>5. Supabase envía un correo con un "Magic Link" o Token.</p> <p>6. El usuario pulsa el enlace y es redirigido a la app.</p> <p>7. El usuario introduce la nueva contraseña (x2).</p> <p>8. El sistema actualiza el hash en la base de datos.</p>
Tecnología	Supabase Auth (SMTP). El backend gestiona el envío del correo sin necesidad de un servidor de correo propio. Se utiliza el estándar PKCE para el flujo de autenticación.
Validaciones	Frontend: Formato de email válido. Backend: Rate Limiting (máximo 3 intentos por hora para evitar spam). Seguridad: El enlace de recuperación tiene una validez temporal limitada (ej: 1 hora).
Excepciones	- Email no encontrado: Por seguridad, el sistema mostrará un mensaje genérico ("Si el email existe, recibirás un correo") para no revelar qué correos están registrados a posibles atacantes (Enumeration Attack Protection).

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Componente Asociado	RecoverPasswordComponent y AuthService.
Impacto en BD	Actualización del campo encrypted_password en la tabla interna auth.users.

Tabla 42: RF.

2.4. Requisitos No Funcionales (RNF)

Ficha Técnica: RNF-01 Usabilidad

Atributo	Especificación Técnica
Identificador	RNF-01
Nombre	Interfaz Intuitiva (UI/UX)
Descripción	Diseño limpio basado en Tailwind CSS. Uso de colores semánticos (Verde=Éxito, Rojo=Error) y componentes claros para usuarios no tecnológicos.

Tabla 43: RNF.

Ficha Técnica: RNF-02 Seguridad de Credenciales

Atributo	Especificación Técnica
----------	------------------------

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Identificador	RNF-02
Nombre	Hashing y Encriptación
Descripción	Las contraseñas se gestionan mediante Supabase Auth, que utiliza algoritmos de hashing robustos (Bcrypt). Nunca se almacenan en texto plano.

Tabla 44: RNF.

Ficha Técnica: RNF-03 Protección de Datos (RGPD)

Atributo	Especificación Técnica
Identificador	RNF-03
Nombre	Cumplimiento Legal
Descripción	Implementación de casillas de aceptación de términos obligatorias. Derecho al olvido mediante borrado total de datos.

Tabla 45: RNF.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Ficha Técnica: RNF-04 Control de Acceso

Atributo	Especificación Técnica
Identificador	RNF-04
Nombre	Seguridad por Roles (RBAC)
Descripción	Uso de Angular Guards (AuthGuard, RoleGuard) en el Frontend y Políticas RLS en el Backend para restringir el acceso a datos.

Tabla 46: RNF.

Ficha Técnica: RNF-05 Diseño Responsivo

Atributo	Especificación Técnica
Identificador	RNF-05
Nombre	Adaptabilidad Multi-dispositivo
Descripción	La interfaz se adapta fluidamente a móviles, tablets y escritorios usando breakpoints de Tailwind (md, lg, xl).

Tabla 47: RNF.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Ficha Técnica: RNF-06 Internacionalización

Atributo	Especificación Técnica
Identificador	RNF-06
Nombre	Soporte Multi-idioma
Descripción	Implementación de @ngx-translate para soportar cambio dinámico entre Español e Inglés.

Tabla 48: RNF.

Ficha Técnica: RNF-09 Escalabilidad

Atributo	Especificación Técnica
Identificador	RNF-09
Nombre	Arquitectura Escalable
Descripción	Uso de Angular Signals para minimizar el impacto en el rendimiento del navegador al manejar grandes volúmenes de datos.

Tabla 49: RNF.

Ficha Técnica: RNF-10 Seguridad API

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Atributo	Especificación Técnica
Identificador	RNF-10
Nombre	Protección de Endpoints
Descripción	Todas las peticiones a Supabase requieren una ApiKey válida y un token de sesión activo.

Tabla 50: RNF.

Ficha Técnica: RNF-11 Compatibilidad

Atributo	Especificación Técnica
Identificador	RNF-11
Nombre	Cross-Browser Support

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Descripción	Garantía de funcionamiento en las últimas versiones estables de Chrome, Firefox, Safari y Edge.
-------------	---

Tabla 51: RNF.

Ficha Técnica: RNF-12 Mantenibilidad

Atributo	Especificación Técnica
Identificador	RNF-12
Nombre	Código Limpio y Modular
Descripción	Arquitectura basada en Componentes Standalone y Servicios inyectables para facilitar futuras actualizaciones sin deuda técnica.

Tabla 52: RNF.

2.3. REVISIÓN Y VALIDACIÓN DE REQUISITOS

El proceso de ingeniería de requisitos no concluye con la elicitation, sino con la validación formal de los mismos. En este apartado se documenta el aseguramiento de la calidad (QA) realizado sobre las especificaciones del proyecto CuidaDos, garantizando que cumplen con los estándares IEEE 830.

2.3.1. Análisis de Calidad (Criterios SMART)

Para validar la viabilidad de los requisitos, se ha sometido el catálogo a un análisis de calidad basado en criterios SMART (Especifico, Medible, Alcanzable, Relevante, Temporal).

ID	Atómico	No Ambiguo	Verificable	Resultado
Requisito				Validación
RF-C01 (Registro)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	APROBADO
RF-C03 (Búsqueda)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	APROBADO
RF-C09 (Chat)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	APROBADO

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

RF-A02 (Dashboard)	<input checked="" type="checkbox"/>	⚠ (Se detalló métricas)	<input checked="" type="checkbox"/>	CORREGIDO
RF-E04 (Horarios)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	APROBADO
RNF-02 (Rendimiento)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> (Lighthouse)	APROBADO

Tabla 53: Calidad.

Nota: Durante la revisión, el requisito RF-A02 fue modificado para especificar exactamente qué gráficas se mostrarían, eliminando la ambigüedad inicial.

2.3.2. Matriz de Trazabilidad (RTM)

La Matriz de Trazabilidad de Requisitos (RTM) es el artefacto que vincula los requisitos funcionales con los componentes técnicos desarrollados, asegurando que no existe código "huérfano" ni requisitos sin implementar.

ID Req	Descripción Corta	Prioridad	Componente	Tabla	Estado
RF-C01	Registro Cliente	Crítica	RegisterFormComponent	public.Usuario, public.Cliente	<input checked="" type="checkbox"/> Impl.
RF-C02	Login	Crítica	LoginComponent, AuthService	auth.users	<input checked="" type="checkbox"/> Impl.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

RF-C03	Buscador Empresas	Alta	SearchBusinessComponent	public.Empr esa, public.Servi cio	<input checked="" type="checkbox"/> Impl.
RF-C04	Contratación	A lta	ContractService	public .Contrato	<input checked="" type="checkbox"/> Impl.
RF-C07	Notificaciones	Media	NavbarComponent	public.Comu nicacion	<input checked="" type="checkbox"/> Impl.
RF-C08	Chat Realtime	Alta	MessagesComponen t	public.Comu nicacion	<input checked="" type="checkbox"/> Impl.
RF-E01	Registro Empresa	Crítica	RegisterFormComponen t	public.Empr esa	<input checked="" type="checkbox"/> Impl.
RF-E04	Gestión Horarios	Alta	TimeService	public.Horar io	<input checked="" type="checkbox"/> Impl.
RF-A03	CRUD Usuarios	Baja	TableCrudAdminCompo nent	public.Usuar io	<input checked="" type="checkbox"/> Impl.
RF-VAL01	Valoraciones	Media	RatingComponent	public.Valor acion	<input checked="" type="checkbox"/> Pendiente

Tabla 54: Matriz.

2.3.3. Riesgos Detectados y Mitigación

Durante la validación de requisitos con los tutores (Cliente), se detectaron conflictos potenciales que requirieron acciones correctivas.

Riesgo Detectado	Impacto	Acción de Mitigación (Cambio en Requisitos)
Incompatibilidad Horaria: Un usuario podría contratar a una empresa en una hora que ya tiene ocupada.	Crítico	Se añadió una validación en base de datos (Trigger) que impide solapamiento de fechas en la tabla Contrato.
Abandono de Chat: Si un usuario borra su cuenta, los mensajes quedan sin remitente y rompen el chat.	Medio	Se definió el requisito RNF-06 para usar ON DELETE SET NULL o mantener los mensajes como "Usuario Eliminado".
Falsos Profesionales: Registro de empresas inexistentes.	Alto	Se añadió el campo verificado (bool) en la tabla Empresa, visible solo por el Administrador.

Tabla 55: Riesgos.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

2.3.4. Acta de Validación Formal

Con fecha 14 de Octubre de 2025, se da por cerrada la fase de Análisis de Requisitos tras la revisión conjunta entre el equipo de desarrollo y la dirección del proyecto.

CONFORMIDAD DEL ALCANCE:

Se certifica que el catálogo de requisitos (SRS) listado en el punto 2.2 cubre las necesidades funcionales mínimas para el MVP de "CuidaDos". Cualquier funcionalidad adicional será tratada como una "Petición de Cambio" (Change Request) para futuras versiones.

CAPÍTULO 6: DISEÑO Y EXPERIENCIA DE USUARIO (UX/UI)

En este capítulo se detalla el proceso creativo y de diseño visual que transforma los requisitos funcionales en una interfaz tangible, usable y accesible. Se ha seguido una metodología centrada en el usuario (User Centered Design).

6.1. Creación de Imagen Corporativa (Branding)

La identidad visual de **CuidaDos** busca transmitir dos valores fundamentales: **Confianza** (esencial en el sector de cuidados) y **Tecnología** (eficiencia y modernidad).



Ilustración 42: Logo

6.1.1. Psicología del Color

Se ha definido una paleta de colores semántica que guía al usuario a través de la interfaz sin necesidad de leer textos explicativos.

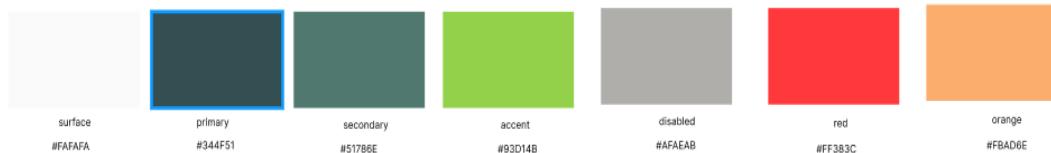
Sistema Cromático y Psicología del Color

Ilustración 43: Colores.

La identidad visual de **CuidadoDos** se aleja del "azul clínico" tradicional para adoptar una paleta inspirada en la naturaleza y la serenidad. Se ha definido un sistema de colores semántico que guía la atención del usuario y reduce la carga cognitiva.

Basándonos en la guía de estilos definida (ver Figura de referencia), se ha implementado la siguiente configuración de variables en el sistema de diseño:

A) Paleta Principal**1. Primary (#344F51 - "Deep Forest"):**

- *Uso:* Encabezados, navegación y elementos de marca principales.
- *Justificación:* Es un tono verde azulado profundo y serio. En psicología del color, transmite estabilidad, confianza y protección profesional, valores críticos para una plataforma donde se contrata el cuidado de familiares.

2. Secondary (#51786E - "Calm Sage"):

- *Uso:* Elementos secundarios, tarjetas y bordes activos.
- *Justificación:* Un verde desaturado que evoca salud y tranquilidad, reduciendo el estrés visual en usuarios que pueden estar gestionando situaciones de emergencia.

3. Accent (#93D14B - "Vital Lime"):

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

- **Uso:** Botones de llamada a la acción (CTA) y estados de éxito.
- **Justificación:** Este verde vibrante crea un alto contraste sobre el fondo oscuro. Se utiliza para dirigir la acción ("Contratar", "Aceptar") y transmite vitalidad y energía positiva.

B) Colores Funcionales y de Estado

- **Surface (#FAFAFA):** Se descarta el blanco puro (#FFFFFF) en favor de este "blanco roto". Esto reduce la fatiga visual en dispositivos móviles, especialmente durante el uso nocturno.
- **Disabled (#AFAEAB):** Gris neutro para indicar elementos no interactivos, cumpliendo con los estándares de usabilidad (heurísticas de Nielsen).
- **Semántica de Error/Alerta:**
 - **Red (#FF383C):** Para errores críticos o acciones destructivas (Cancelar contrato).
 - **Orange (#FBAD6E):** Para advertencias o estados transitorios (Pendiente de firma).

6.1.2. Tipografía

Para la composición tipográfica, se ha seleccionado una pareja de fuentes ("Font Pairing") que equilibra la funcionalidad digital con la calidez humana. Ambas son fuentes de código abierto (Google Fonts), lo que garantiza la sostenibilidad del proyecto sin costes de licencias.

A) Tipografía Display: Fraunces

- **Uso:** Títulos, Encabezados (H1, H2) y Logotipo.
- **Clasificación:** *Old Style Serif* (Variable Font).
- **Justificación de Diseño:** Fraunces es una tipografía con mucha personalidad, curvas suaves y un aire "retro-moderno".
 - **Impacto Emocional:** A diferencia de las letras de palo seco (que pueden parecer frías o burocráticas), las serifas suaves de Fraunces comunican el

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

concepto de "**Amistar**" y cercanía humana. Hace que la app parezca una revista editorial cuidada y no una hoja de cálculo administrativa.

- *Ventaja Técnica:* Al ser una **Variable Font**, permite ajustar el peso (font-weight) y la anchura óptica (font-optical-sizing) con un solo archivo, optimizando la carga.

B) Tipografía de Cuerpo: Poppins

- **Uso:** Párrafos, Botones, Inputs y textos legales.
- **Clasificación:** *Geometric Sans Serif*.
- **Justificación de UX:**
 - *Legibilidad:* Poppins se caracteriza por sus formas geométricas casi perfectas y su gran altura de la x. Esto la hace extremadamente legible en pantallas pequeñas de móviles, incluso a tamaños reducidos (12px - 14px).
 - *Neutralidad:* Su limpieza visual permite que la información densa (contratos, horarios, precios) se lea rápidamente sin cansar la vista.

Categoría	Familia Tipográfica	Características	Función en CuidaDos
Títulos	Fraunces	Serif, Humanista, Calidez.	Aportar identidad de marca y cercanía ("El toque humano").

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Cuerpo	Poppins	Sans-serif, Geométrica, Limpia.	Garantizar la legibilidad de los datos y contratos ("El toque profesional").
--------	---------	------------------------------------	--

Tabla 56: Tipografía.

Jerarquía:

- **Bold (700):** Títulos y encabezados.
- **Medium (500):** Botones y etiquetas de formulario.
- **Regular (400):** Cuerpo de texto y párrafos.

Fonts PC

Name	Font	Font weight	Font size	Line height
Heading 1	Fraunces	Regular	32px/ 2rem	1.2 x font size
Heading 2	Fraunces	Regular	24px/1.5rem	1.2 x font size
Paragraph	Poppins	Medium	16px/1rem	1.2 x font size
Label	Poppins	Regular	16px/1rem	1.2 x font size

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes**Fonts Tablet**

Name	Font	Font weight	Font size	Line height
Heading 1	Fraunces	Regular	24px/1.5rem	1.2 x font size
Heading 2	Poppins	Regular	20px/1.25rem	1.2 x font size
Paragraph	Poppins	Medium	16px/1rem	1.2 x font size
Label	Poppins	Regular	14px/0.875rem	1.2 x font size

Fonts phone

Name	Font	Font weight	Font size	Line height
Heading 1	Fraunces	Regular	20px/1.25rem	1.2 x font size
Heading 2	Fraunces	Regular	16px/1rem	1.2 x font size
Paragraph	Poppins	Medium	14px/0.875rem	1.2 x font size
Label	Poppins	Regular	12px/0.75rem	1.2 x font size

Ilustración 44: tipografía.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

6.1.3. Isotipo y Logotipo

El logotipo nace de la fusión de dos conceptos: la protección (representada por formas redondeadas) y la conexión (representada por nodos o enlaces). Se han evitado aristas vivas para suavizar la percepción tecnológica.



Ilustración 45: Logo.

6.2. Desarrollo de Interfaces Gráficas (UI Design)

El diseño de la interfaz se ha realizado siguiendo los principios del **Atomic Design**, construyendo componentes de menor a mayor complejidad. Esto garantiza la consistencia visual en toda la aplicación.

6.2.1. Sistema de Diseño (Design System)

Antes de maquetar las páginas, se definieron los componentes base en Figma:

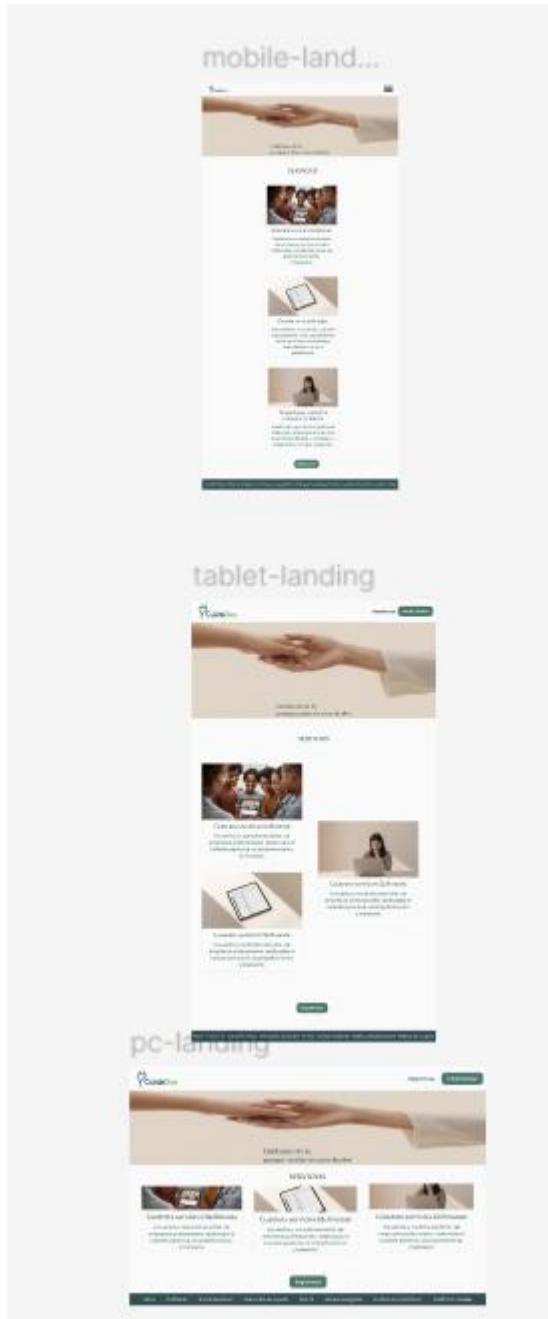
- **Átomos:** Botones, Inputs, Iconos (Lucide), Badges.
- **Moléculas:** Tarjetas de Usuario (Avatar + Nombre + Estado), Barras de Búsqueda.
- **Organismos:** Formularios completos, Tablas de datos, Chat.

6.2.2. Vistas Principales (High-Fidelity)

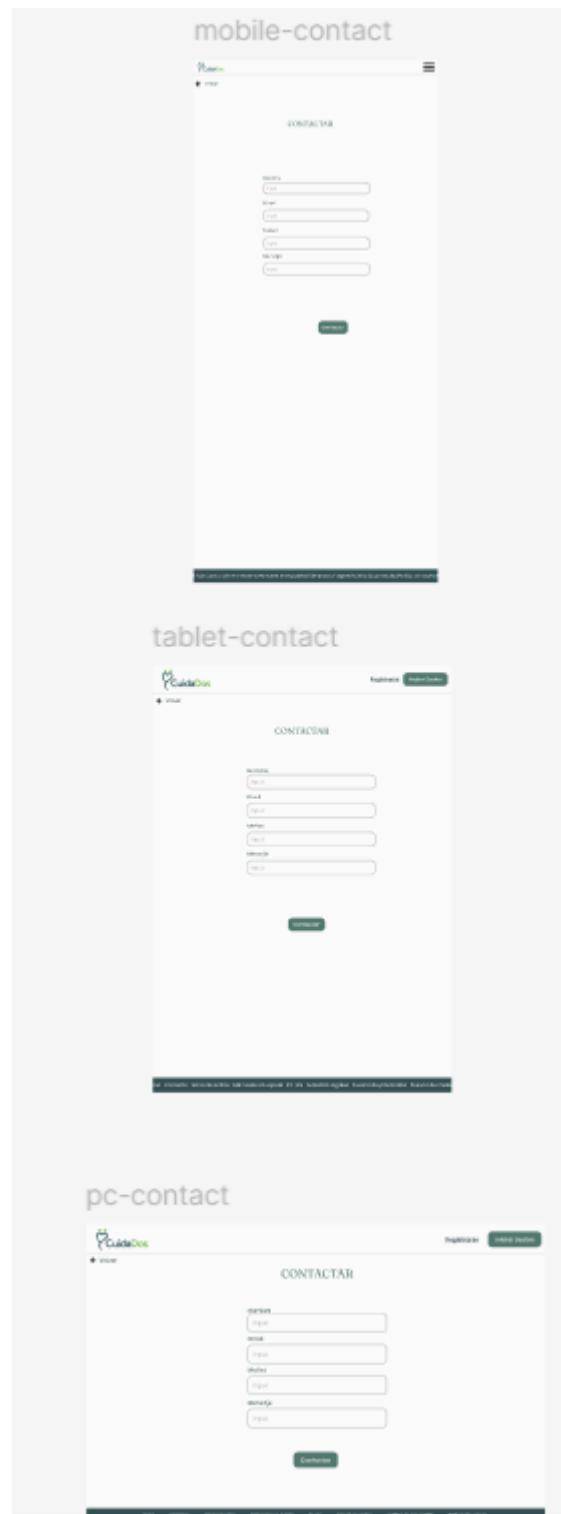
A continuación se muestran las vistas **prototipo** del aplicativo cuidaDos.

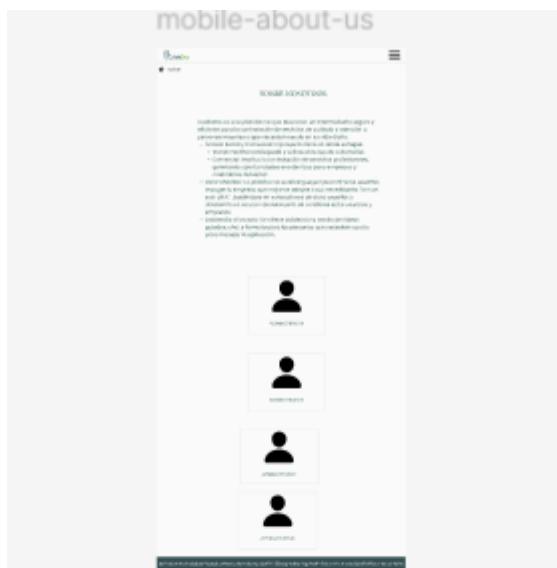
También se muestran el boceto original y el resultado final.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

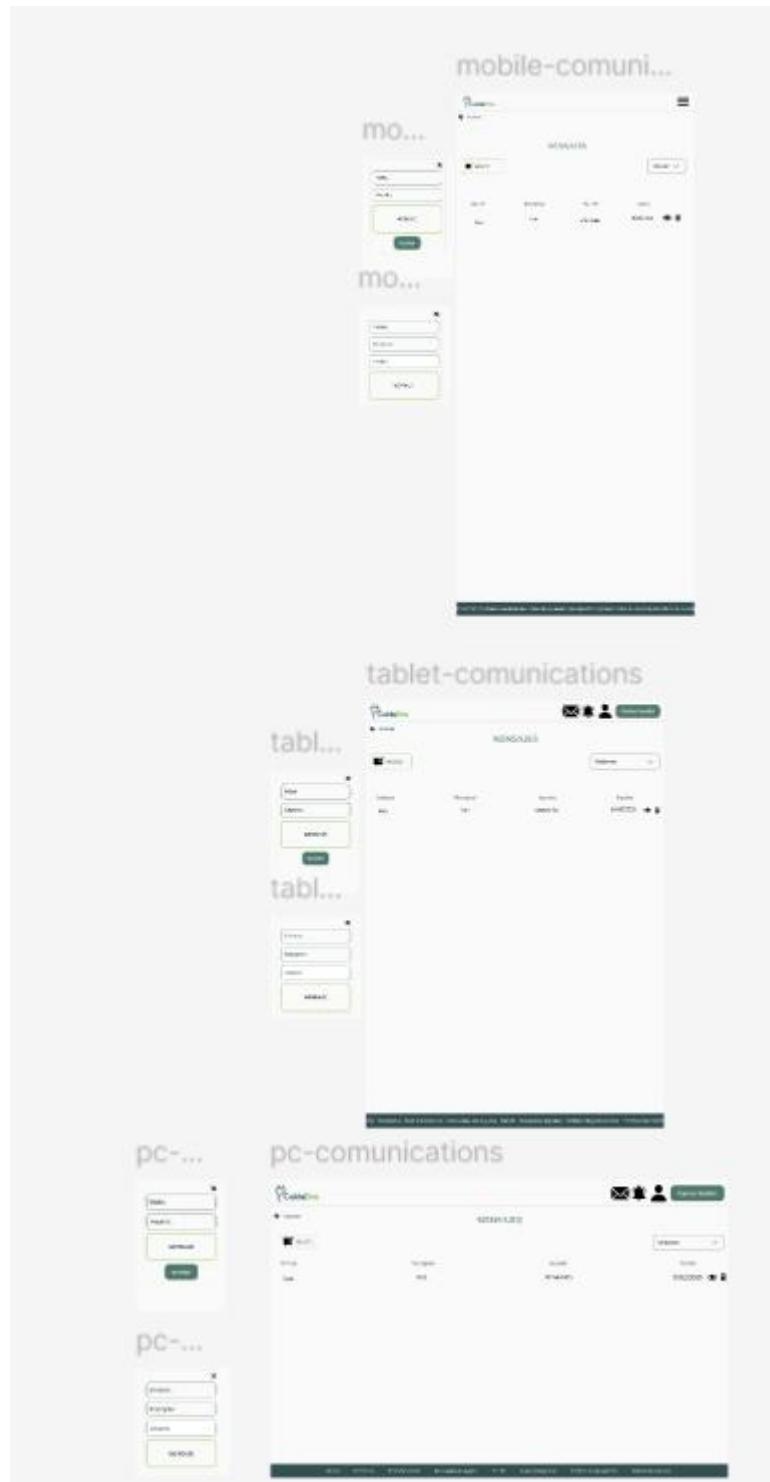


Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

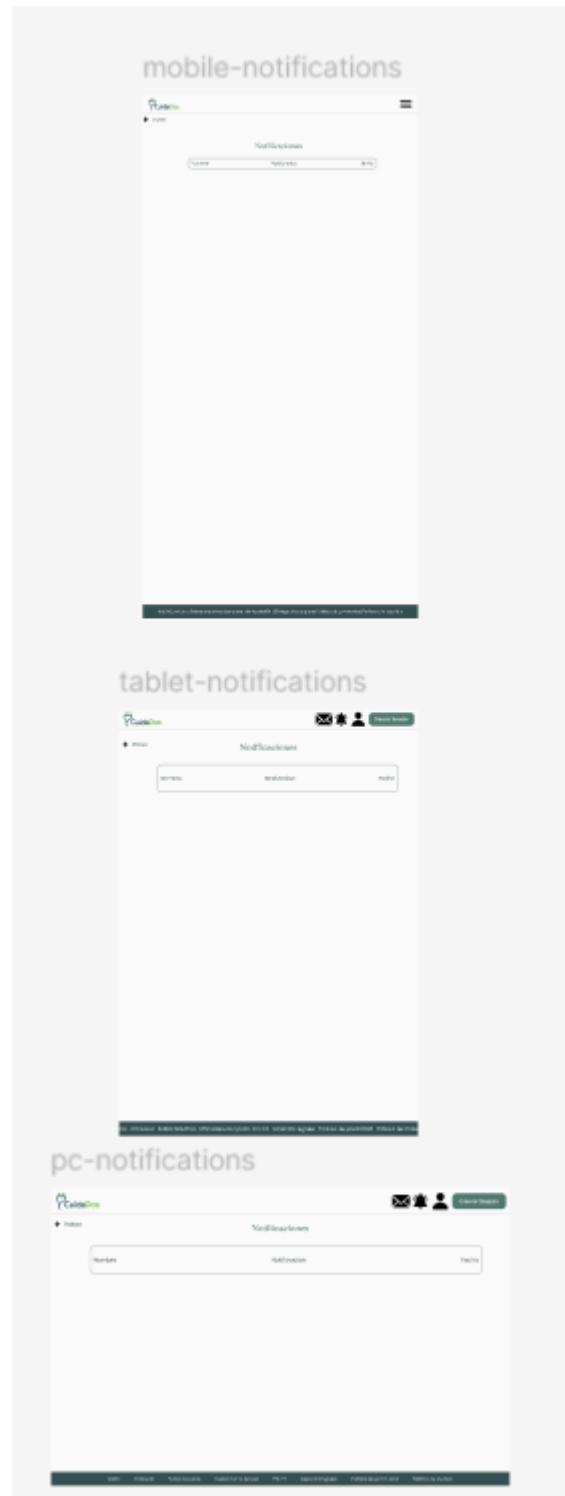
Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

The image displays three views of a web application interface:

- mobile-contract**: Shows a modal window titled "Detalles del contrato". It contains sections for "Detalles del arrendatario" (tenant details) and "Detalles de la actividad" (activity details). The tenant details include: Nombre completo: Juan Pérez Sánchez, DNI: 12345678, Dirección: Avda. de las Flores 123, Localidad: Madrid, Código postal: 28001, and Teléfono: 600000000. The activity details include: Nombre completo: Alquiler de vivienda, DNI: 12345678, Dirección: Avda. de las Flores 123, Localidad: Madrid, Código postal: 28001, and Teléfono: 600000000.
- tablet-contract**: Shows a similar modal window for "Detalles del contrato". The tenant details are identical. The activity details include: Nombre completo: Alquiler de vivienda, DNI: 12345678, Dirección: Avda. de las Flores 123, Localidad: Madrid, Código postal: 28001, and Teléfono: 600000000.
- pc-contract**: Shows a similar modal window for "Detalles del contrato". The tenant details are identical. The activity details include: Nombre completo: Alquiler de vivienda, DNI: 12345678, Dirección: Avda. de las Flores 123, Localidad: Madrid, Código postal: 28001, and Teléfono: 600000000.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The image displays two side-by-side screenshots of a registration form for the "Cuidados" application, titled "mobile-register-form-user" and "tablet-register-form-user".

mobile-register-form-user Screenshot:

- Header:** mobile-register-form-user
- Logo:** Cuidados
- Form Fields:**
 - Nombre:
 - Primer apellido: Último apellido:
 - Fecha de nacimiento: AÑO:
 - Teléfono de contacto:
 - CORREO:
 - Dirección:
 - Ubicación:
 - SEXO:
 - Hombre
 - Mujer
 - Contraseña: Repetir Contraseña:
- Text:** Al aceptar las condiciones de uso, autorizo el tratamiento de mis datos personales para la ejecución del servicio y la personalización de la información que recibo.
- Checkboxes:** Acepto las condiciones de uso
- Buttons:** REGISTRARSE

tablet-register-form-user Screenshot:

- Header:** tablet-register-form-user
- Logo:** Cuidados
- Buttons:** Registrarse Salir Sesión
- Section:** REGISTRARSE
- Form Fields:**
 - Nombre:
 - Primer apellido: Último apellido:
 - Fecha de nacimiento: AÑO:
 - Teléfono de contacto:
 - CORREO:
 - Dirección:
 - Ubicación:
 - SEXO:
 - Hombre
 - Mujer
 - Contraseña: Repetir Contraseña:
- Text:** Al aceptar las condiciones de uso, autorizo el tratamiento de mis datos personales para la ejecución del servicio y la personalización de la información que recibo.
- Checkboxes:** Acepto las condiciones de uso
- Buttons:** REGISTRARSE

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

The image displays two side-by-side screenshots of a registration form from a mobile application. Both screens show a header with the logo 'CuidaDos' and a 'REGISTRARSE' button. The top screen is labeled 'mobile-register-form-b...' and the bottom screen is labeled 'tablet-register-form-busin...'. Both forms contain identical fields: Nombre (Input), OF (Input), Teléfono de contacto (Input), Email (Input), Dirección (Input), Localidad (Input), and two radio buttons for Género (Male and Female). Below these fields is a checkbox for accepting terms and conditions, followed by a large green 'REGISTRARME' button.

mobile-register-form-b...

REGISTRARSE

Nombre

OF

Teléfono de contacto

Email

Dirección

Localidad

Género Género

checkbox Al aceptar las [TÉRMINOS Y CONDICIONES](#) y/o [Política de privacidad](#) y [Política de cookies](#) autorizo a la aplicación a: a) manejar mis datos personales para el mejor servicio de mi actividad y b) los enviar a las autoridades.

REGISTRARME

tablet-register-form-busin...

REGISTRARSE

Nombre

OF

Teléfono de contacto

Email

Dirección

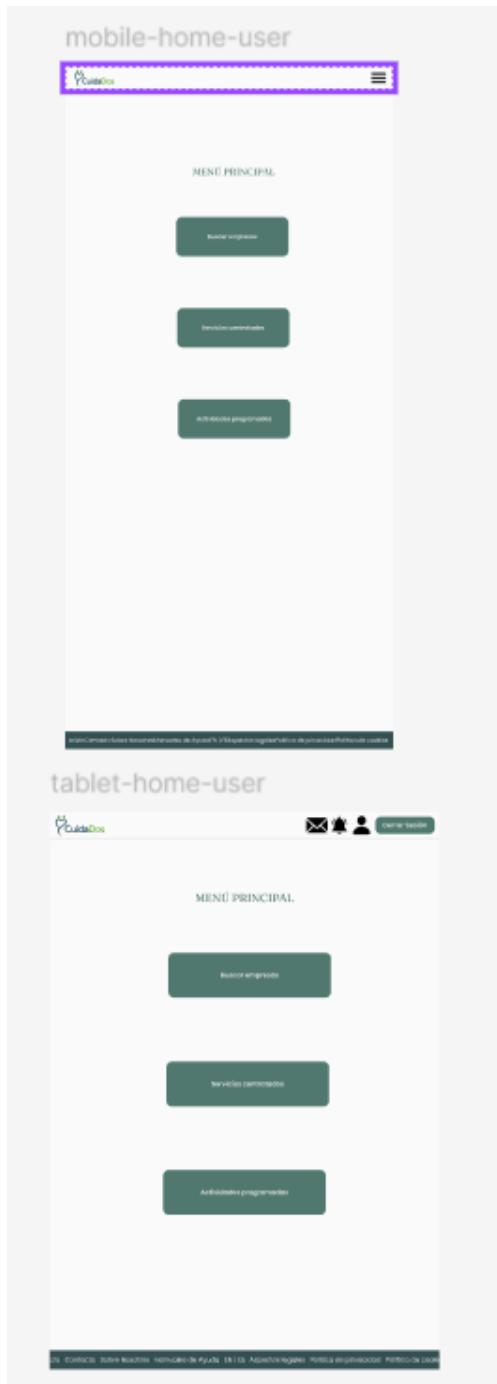
Localidad

Género Género

checkbox Al aceptar las [TÉRMINOS Y CONDICIONES](#) y/o [Política de privacidad](#) y [Política de cookies](#) autorizo a la aplicación a: a) manejar mis datos personales para el mejor servicio de mi actividad y b) los enviar a las autoridades.

REGISTRARME

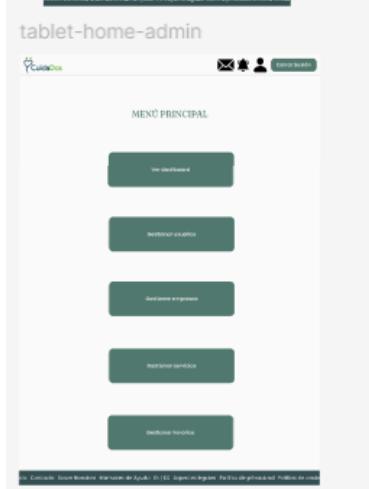
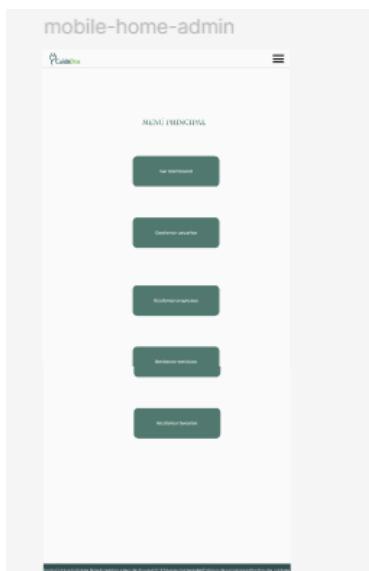
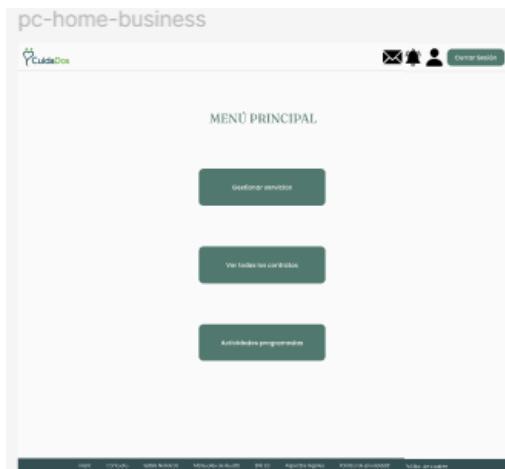
Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

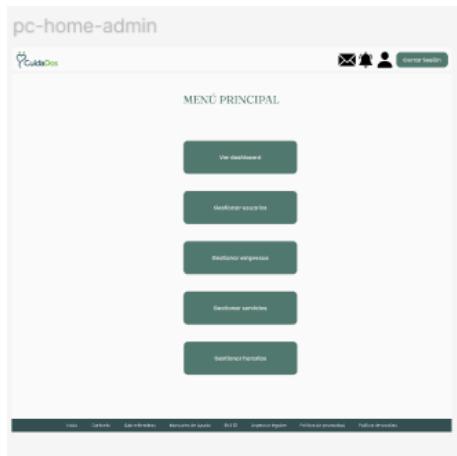


Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

This image compares two mobile devices displaying the same application interface. The top device is labeled 'mobile-search-user' and the bottom one is labeled 'tablet-search-user'. Both screens show a search bar at the top. Below it is a form with fields: 'Nombre', 'Apellido', 'Número de identidad', 'Dirección', 'Ciudad/Barrio', 'Teléfono', and 'Servicios'. A dropdown menu is open over the 'Servicios' field, showing options like 'Servicios', 'Exámenes', and 'Consultas'. A button labeled 'Buscar paciente' is located at the bottom right of the form. The overall layout is identical on both devices.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

pc-search-user

The screenshot shows a search interface titled "pc-search-user". At the top right are icons for envelope, bell, user profile, and "Cerrar Sesión". Below the title is a "Volver" button. A search bar with a magnifying glass icon is followed by a large rectangular form containing fields for "Nombre de la empresa", "Diseño gráfico", "Involucrados", "Servicios", a dropdown menu for "Servicios", and a "Consultar" button.

mobile-users-profile

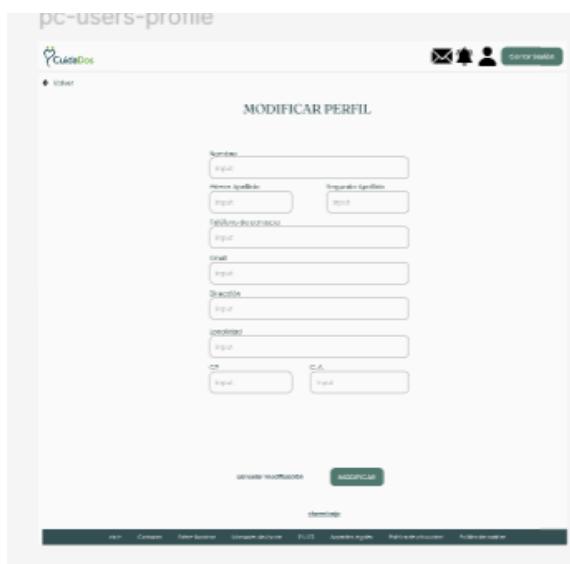
The screenshot shows a mobile profile modification interface titled "mobile-users-profile". It features a "MODIFICAR PERFIL" section with various input fields for "Nombre", "Apellido", "Número teléfono", "Número móvil", "Número de contacto", "Email", "Contraseña", "CP", "C.P.", and "C.I.A.". Below the form are "cancelar" and "modificar" buttons, and a note about terms and conditions.

tablet-users-profile

The screenshot shows a tablet profile modification interface titled "tablet-users-profile". It has a similar layout to the mobile version, with "MODIFICAR PERFIL" and input fields for "Nombre", "Apellido", "Número teléfono", "Número móvil", "Número de contacto", "Email", "Contraseña", "CP", "C.P.", and "C.I.A.". It includes "cancelar" and "modificar" buttons and a note about terms and conditions.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

pc-users-profile



MODIFICAR PERFIL

Nombre

Número Apellidos

Segundo Apellido

DNI/NIF/CIF

Email

Teléfono

Ciudad

CP

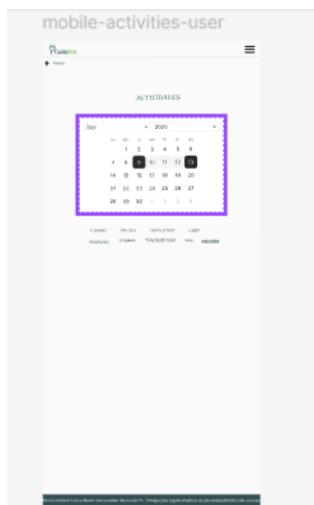
Localidad

Siguiente información

Administración

Inicio Consultas Ficha Solicitud Consulta de solicitud Estatus Actividades Solicitud de actividad

mobile-activities-user



ACTIVIDADES

...
...
...
...

Categorías Actividad Nuevo evento Nuevo tipo Nuevo Cancelar

tablet-activities

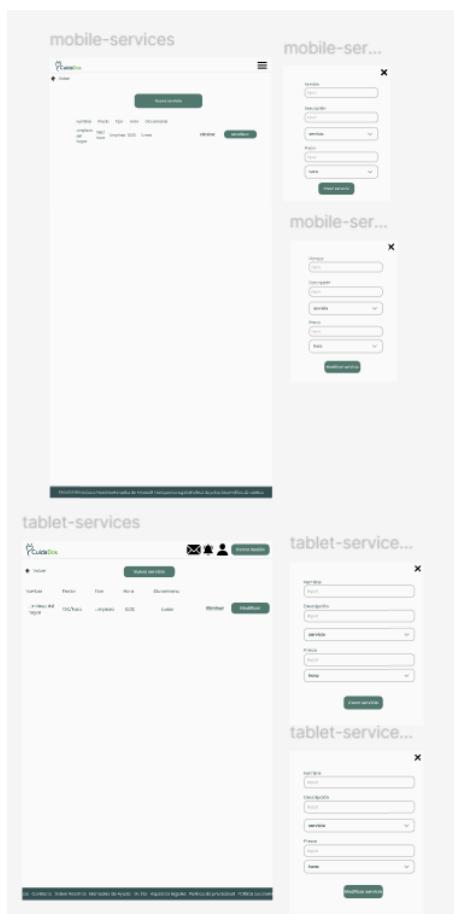
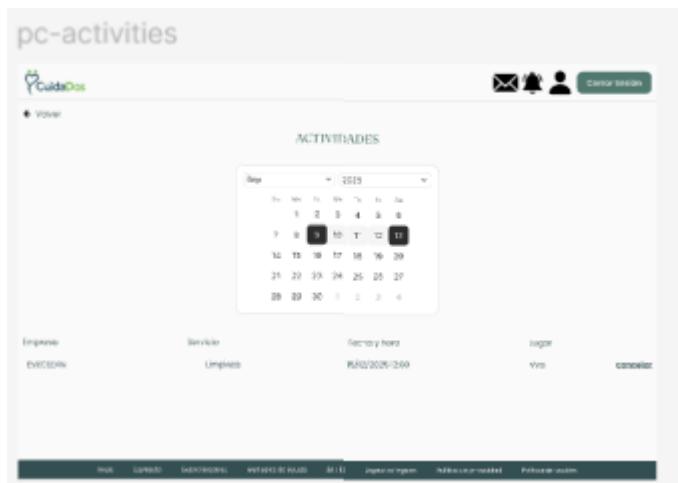


ACTIVIDADES

...
...
...
...

Categorías Actividad Nuevo evento Nuevo tipo Nuevo Cancelar

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

pc-services

pc-services-form

pc-services-mod-form

mobile-profile-empresas

MODIFICAR PERFIL

tablet-profile-empresas

MODIFICAR PERFIL

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

The screenshot shows a web page titled "pc-profile-empresas". At the top right are icons for message, notifications, and user profile, along with a "Cierre Sesión" button. Below the title is a "MODIFICAR PERFIL" button. The main area contains several input fields for personal information: Nombre, Teléfono de contacto, Email, Dirección, and Direccion. There are also fields for "Localidad" (Address), "CP" (Postal Code), "CA" (City), and "Descripción" (Description). At the bottom are "cancelar modificación" and "MODIFICAR" buttons, and a "Datos básicos" link.

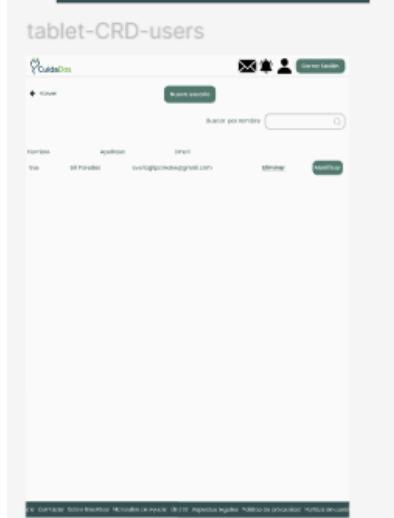
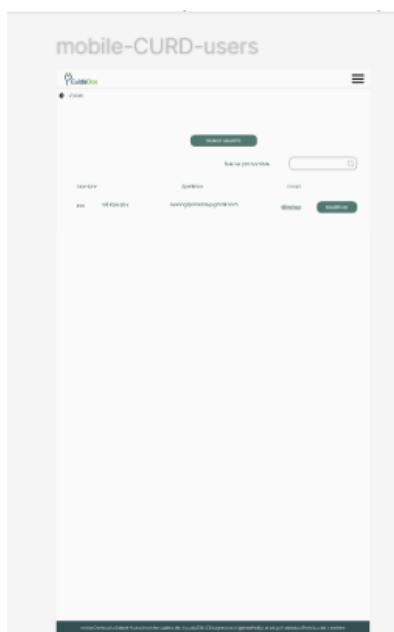
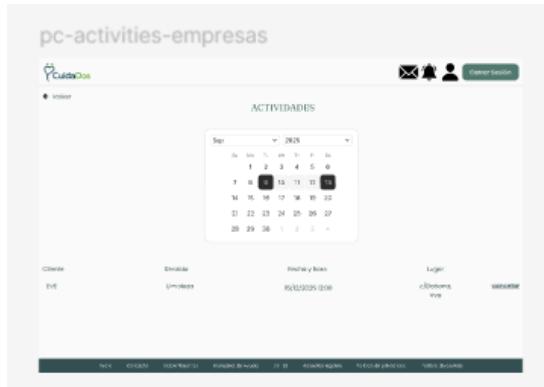
The screenshot shows a web page titled "mobile-activities-empresas". At the top right are icons for message, notifications, and user profile, along with a "Cierre Sesión" button. Below the title is a "ACTIVIDADES" section. It features a calendar grid for the month of September 2023. The days of the week are labeled from Monday to Sunday. Specific days are highlighted in blue, indicating scheduled activities. Below the calendar, there is a table with columns: Clase, Servicio, Fecha/Hora, and Lugar. One row is visible with the value "Aula 1" in the Clase column. At the bottom is a footer with the text "El contenido de esta página es de dominio público, autorizado para su uso en la enseñanza y estudio".

The screenshot shows a web page titled "tablet-activities-empresas". At the top right are icons for message, notifications, and user profile, along with a "Cierre Sesión" button. Below the title is a "ACTIVIDADES" section. It features a calendar grid for the month of September 2023. Specific days are highlighted in blue, indicating scheduled activities. Below the calendar, there is a table with columns: Clase, Servicio, Fecha/Hora, and Lugar. One row is visible with the value "Aula 1" in the Clase column. At the bottom is a footer with the text "El contenido de esta página es de dominio público, autorizado para su uso en la enseñanza y estudio".

MANUAL CUIDADOS

Versión: 1.0 Fecha: 2026

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes



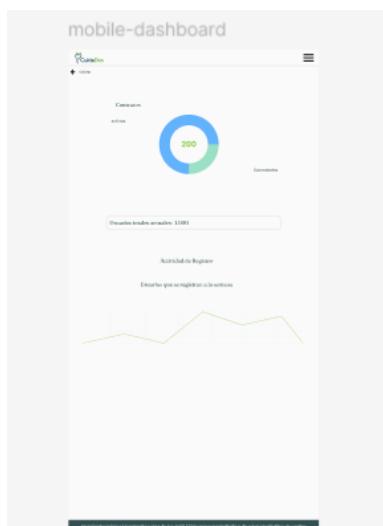
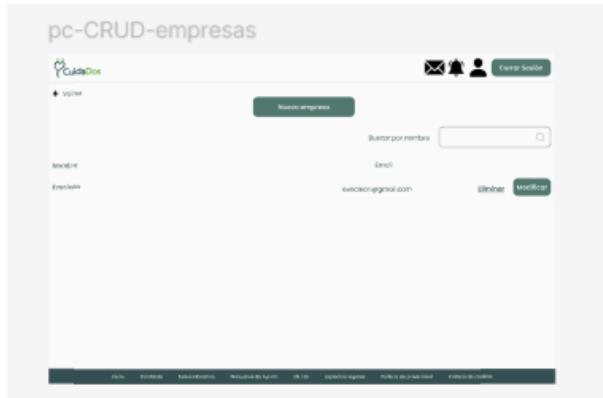
MANUAL CUIDADOS

Versión: 1.0 Fecha: 2026

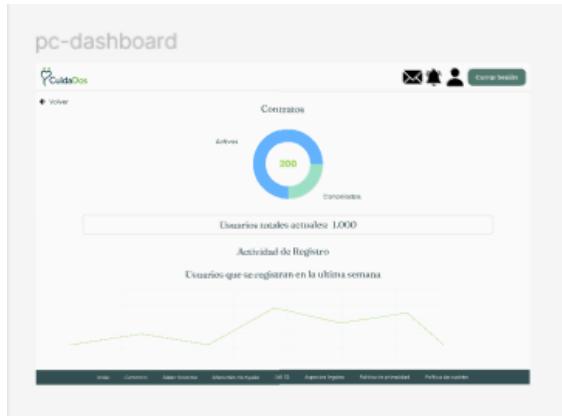
Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



The mobile-admin-profile screen displays a 'MODIFICAR PERFIL' section with input fields for 'Nombre' (Name), 'Número de contrato' (Contract number), 'Email' (Email), and a 'Contraseña' (Password) field. A 'Guardar cambios' (Save changes) button is at the bottom. A note at the bottom states: 'Sólo se permite una modificación de nombre, número de contrato y correo electrónico cada 15 días.'

The tablet-admin-profile screen is identical to the mobile version, showing the 'MODIFICAR PERFIL' form with fields for Name, Contract number, Email, and Password, along with a 'Guardar cambios' button and the same note about name changes.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The image displays three vertical screenshots of a mobile application interface for service management, likely designed for different devices or screen sizes.

- Top Screenshot (pc-admin-profile):** A desktop-like interface titled "MODIFICAR PERFIL". It features a header with a logo, notifications, and a sign-out button. Below the header, there are four input fields for "Nombre", "Nombre de usuario", "Email", and "Contraseña". A "Actualizar" (Update) button is positioned below the fields. At the bottom, there is a navigation bar with links: "Inicio", "Servicios", "Nuevo Servicio", "Reservar Servicio", "Mis Svc.", "Reservaciones", "Historial de reservas", and "Historial de servicios".
- Middle Screenshot (mobile-crud-services-gl...):** A mobile interface titled "SERVICIOS". It includes a header with a logo, notifications, and a sign-out button. The main area contains two input fields for "Nombre" and "Tipo de servicio", followed by a "Guardar" (Save) button. Below these fields, there is a table with columns "Nombre", "Tipo", and "Estado". A single row is visible with the values "Peluquería" and "Cortar pelo".
- Bottom Screenshot (tablet-CRD-services-global):** A tablet-oriented interface titled "SERVICIOS". It has a similar header and layout to the middle screenshot. The main area shows two input fields for "Nombre" and "Tipo de servicio", and a "Guardar" button. Below the fields is a table with columns "Nombre", "Tipo", and "Estado". A single row is visible with the values "Peluquería" and "Cortar pelo".

MANUAL CUIDADOS

Versión: 1.0 Fecha: 2026

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

The screenshot shows a web application interface for managing services. At the top, there's a header with the project name 'pc-CRUD-services-global' and a navigation bar with icons for mail, notifications, and user profile. Below the header, the main section is titled 'SERVICIO'. It contains two input fields: 'Nombre' (with value 'correo') and 'Tipo de servicio' (with value 'Email'). A green 'Añadir' button is positioned below these fields. At the bottom of the page, there's a footer with links: 'Inicio', 'Contacto', 'Sobre Nosotros', 'Terminos de Uso', 'Preguntas', 'Aviso Legal', 'Politica de Privacidad', and 'Politica de Cookies'.

The screenshot shows a mobile application interface for managing schedules. The title 'mobile-crud-horario-gl...' is at the top. The main form is titled 'HORARIOS' and includes fields for 'Hora' (value '10:00') and 'Dia de la semana' (dropdown menu). A green 'Añadir' button is present. Below the form, there's a table with two rows: '10:00' and '10:00'. Each row has a 'Borrar' (Delete) button. At the bottom, there's a footer with links: 'Inicio', 'Contacto', 'Sobre Nosotros', 'Terminos de Uso', 'Preguntas', 'Aviso Legal', 'Politica de Privacidad', and 'Politica de Cookies'.

The screenshot shows a tablet application interface for managing schedules. The title 'tablet-CRD-horario-global' is at the top. The main form is titled 'HORARIOS' and includes fields for 'Hora' (value '10:00') and 'Dia de la semana' (dropdown menu). A green 'Añadir' button is present. Below the form, there's a table with two rows: '10:00' and '10:00'. Each row has a 'Borrar' (Delete) button. At the bottom, there's a footer with links: 'Inicio', 'Contacto', 'Sobre Nosotros', 'Terminos de Uso', 'Preguntas', 'Aviso Legal', 'Politica de Privacidad', and 'Politica de Cookies'.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

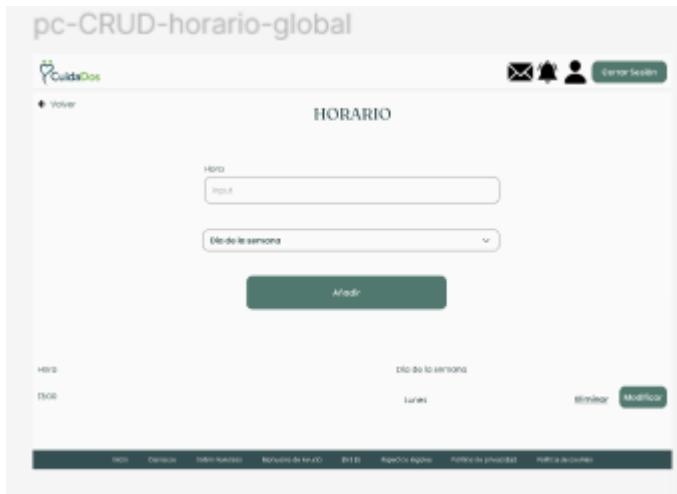
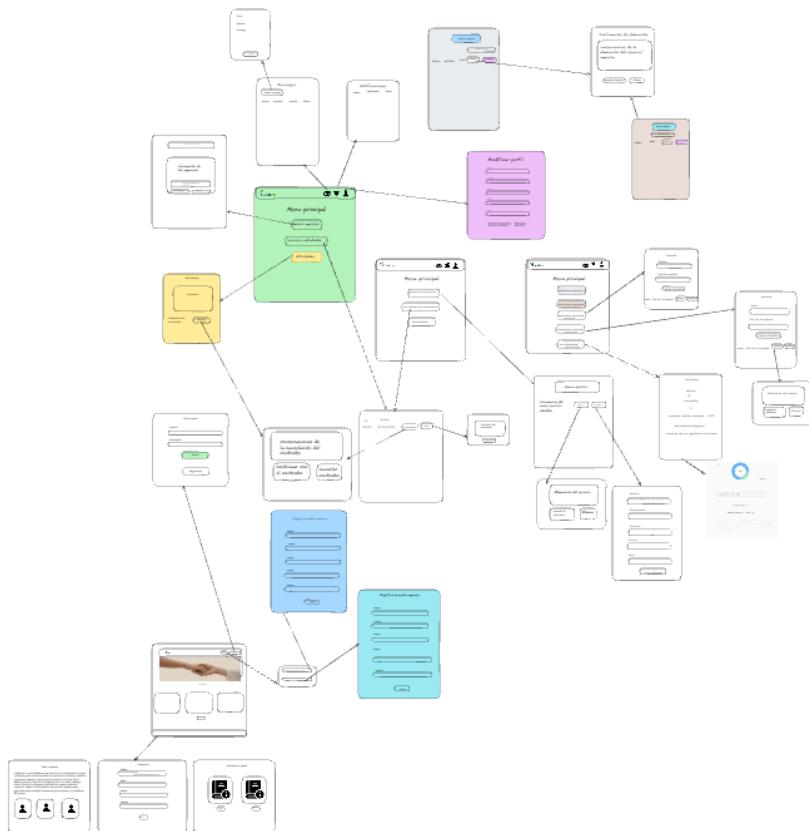
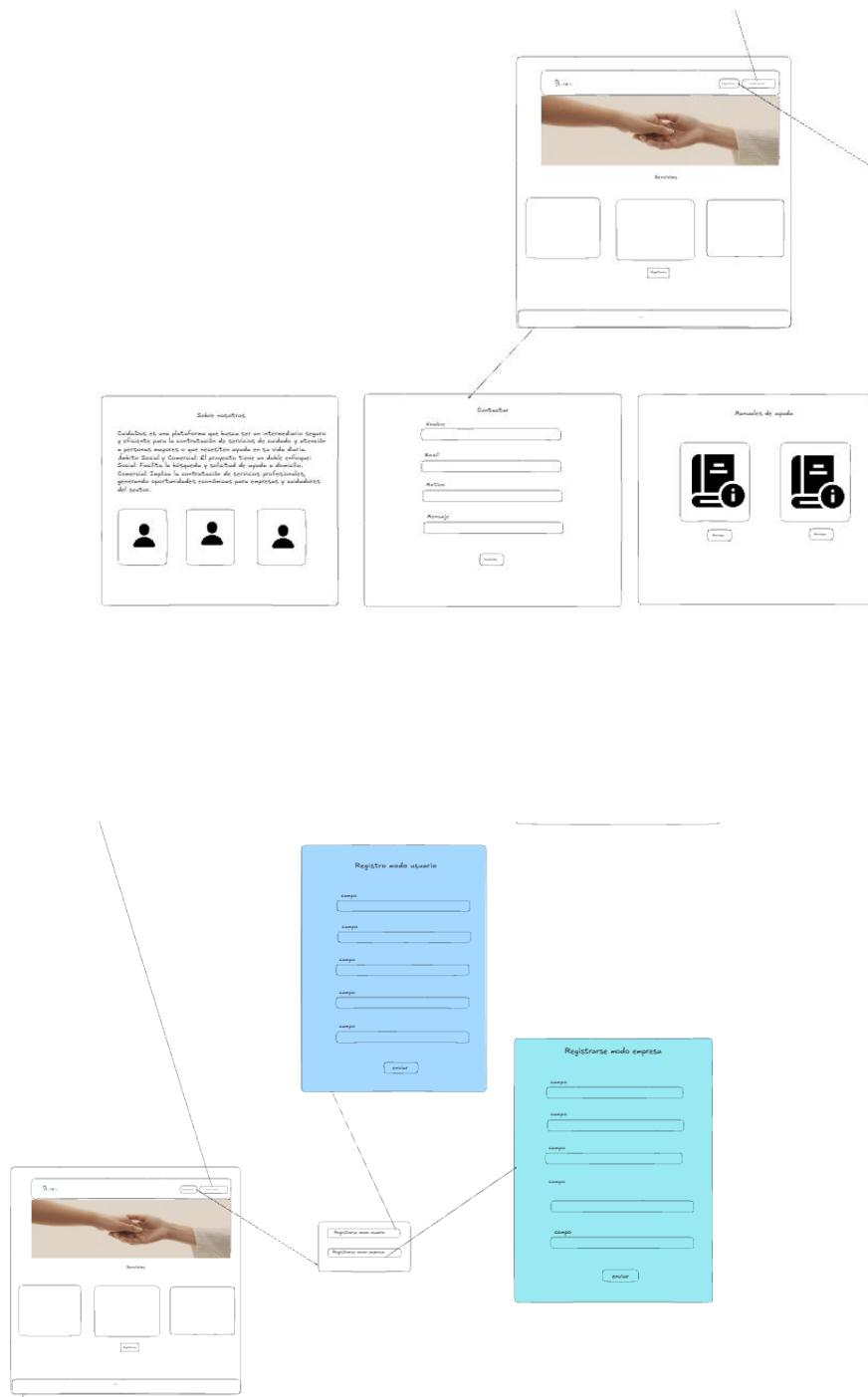
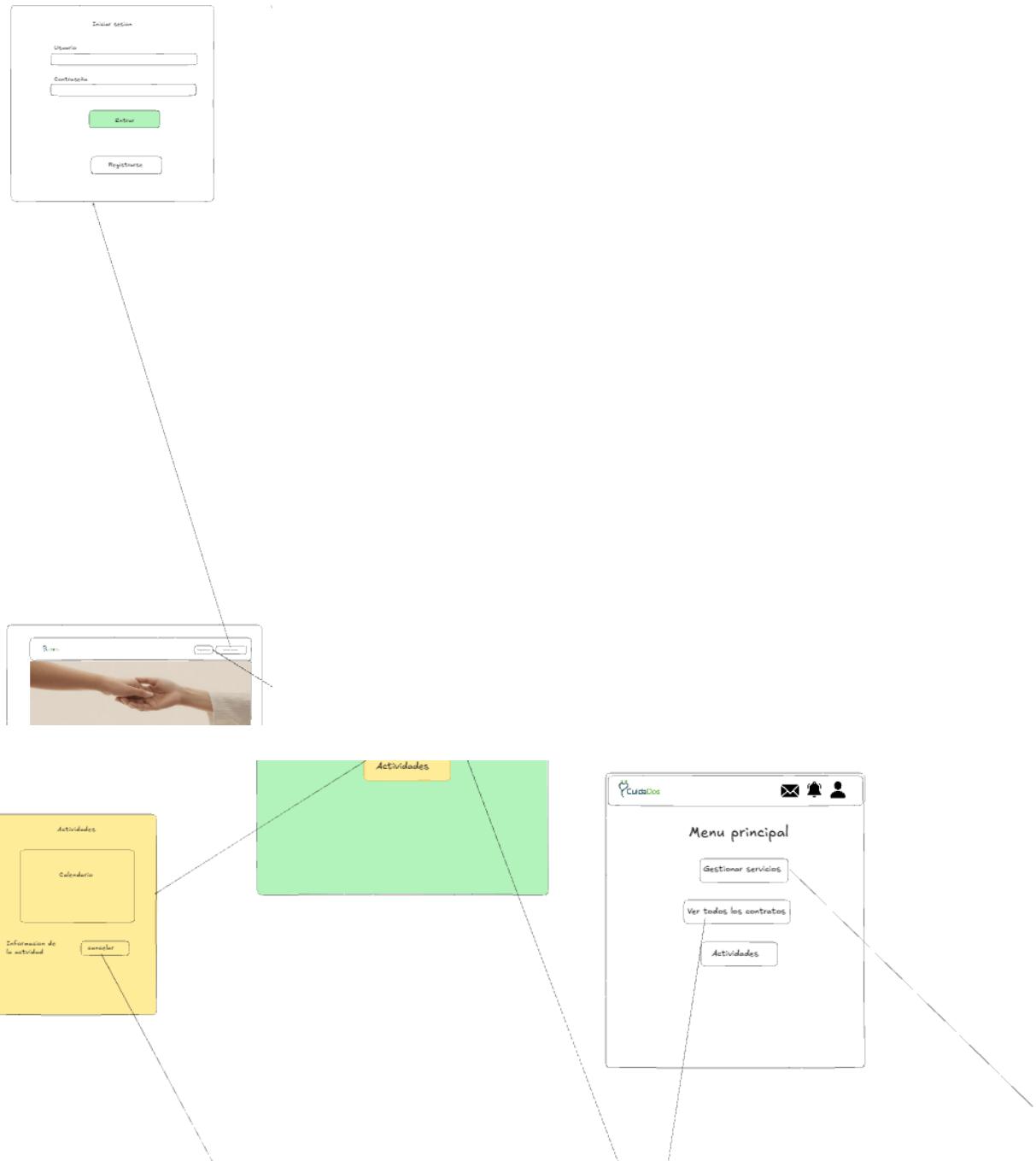


Ilustración 46: Vistas prototipo.

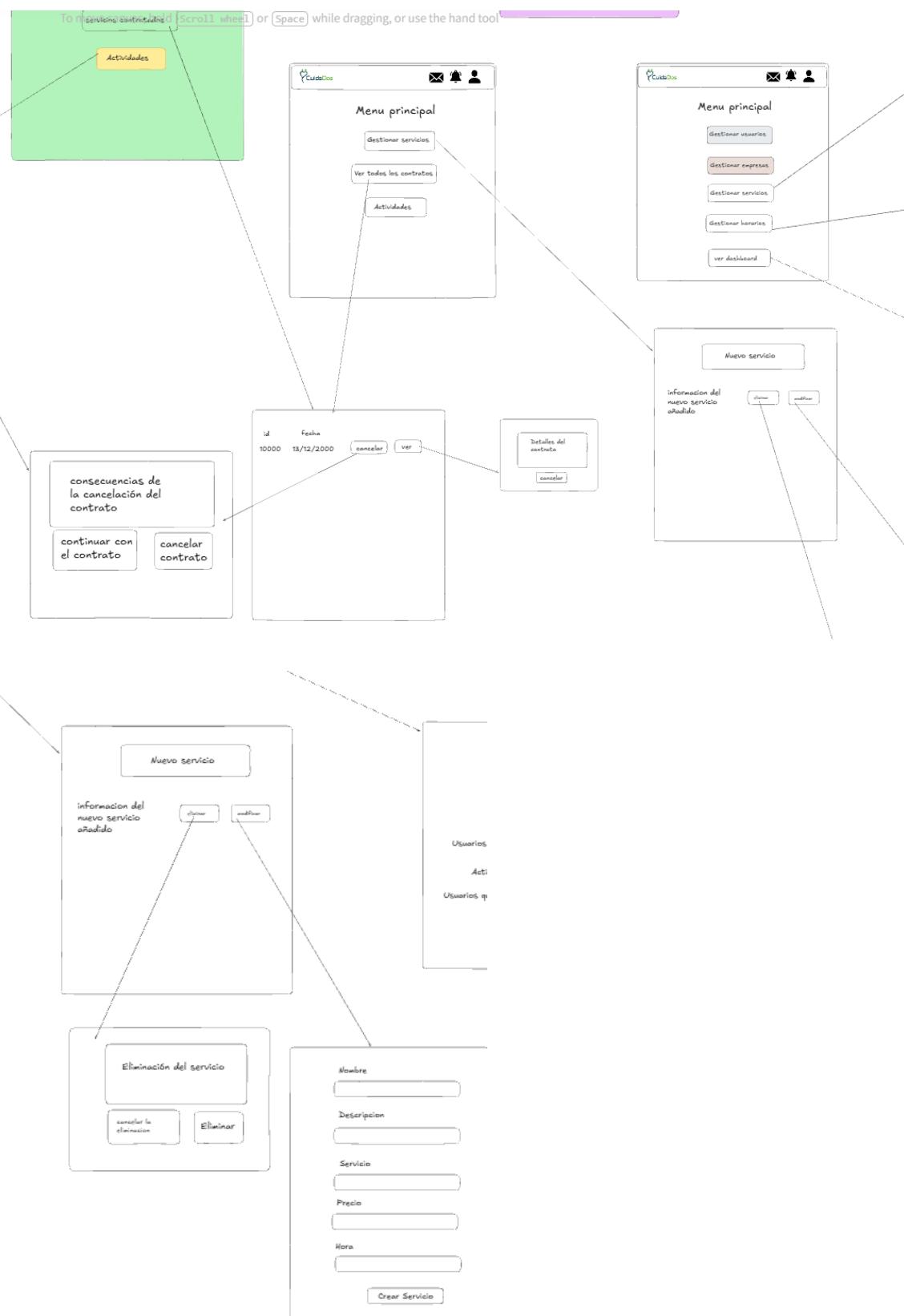
Todas estas páginas surgen a partir de un boceto original creado en excalidraw:



Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes



Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

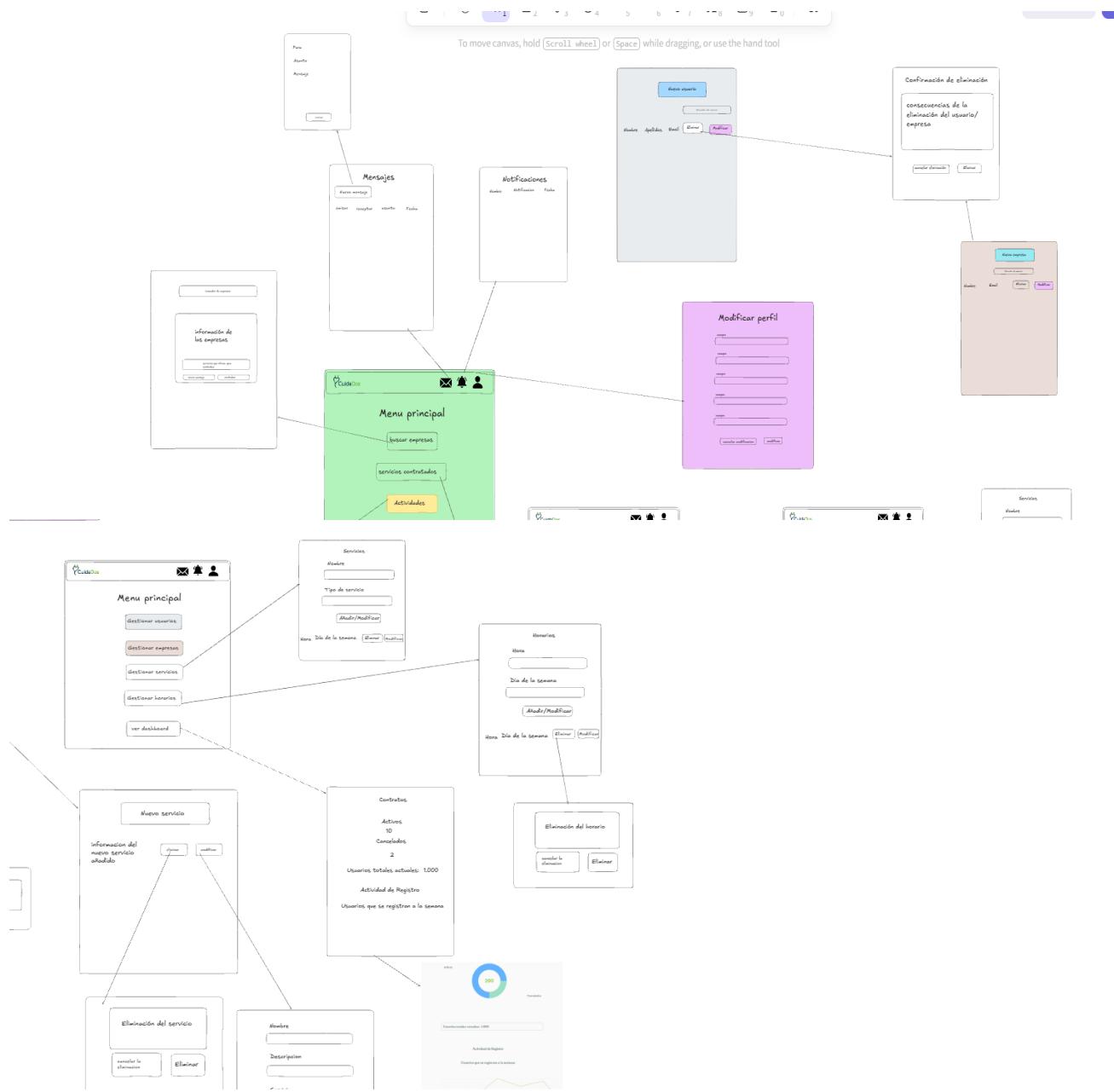


Ilustración 47: Excalidraw.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

PAGINAS RESULTADO

USUARIO SIN SESIÓN



SERVICIOS

The landing page features three service cards:

- Contrata servicios fácilmente**
Encuentra y contrata servicios de empresas profesionales dedicadas al cuidado personal, acompañamiento y bienestar.
- Gestión en un solo lugar**
Administra tus contratos, recibe recordatorios y haz seguimiento de los servicios contratados, fácilmente.
- Resolución de problemas**
Nuestro equipo te acompaña en cada paso para que encuentres soluciones rápidas, confiables y adaptadas a lo que necesitas.

[Registrarse](#)

Ilustración 48: Landing.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The screenshot shows the 'Contactar' (Contact) page of the CuidaDos website. At the top right are 'Registrarse' and 'Iniciar Sesión' buttons. The page title 'Contactar' is centered above four input fields: 'Nombre' (Name), 'Email', 'Motivo' (Reason), and 'Mensaje' (Message). Below these is a large green 'Contactar' button. At the bottom, a navigation bar includes links for 'Inicio', 'Contacto', 'Sobre Nosotros', 'Manuales', 'Aspectos legales', 'Política de privacidad', 'Política de cookies', and language options 'ES | EN'.

Nombre
Nombre

Email
Email

Motivo
Motivo

Mensaje
Mensaje

Contactar

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 49:Contacto.

← VolverRegistrarseIniciar Sesión

Sobre Nosotros

CuidaDos es una plataforma que busca ser un intermediario seguro y eficiente para la contratación de servicios de cuidado y atención a personas mayores o que necesiten ayuda en su vida diaria. Somos una plataforma digital de cuidados creada con el objetivo de mejorar la calidad de vida de las personas que necesitan apoyo en su día a día. Nuestro equipo está formado por profesionales del ámbito tecnológico y social que comparten una misma visión: facilitar el acceso a servicios de atención y cuidado de forma sencilla, segura y transparente. Creemos firmemente que el cuidado de las personas debe estar basado en la confianza, la profesionalidad y la cercanía, y por ello trabajamos para conectar a usuarios con empresas especializadas en el sector. El proyecto nace en el contexto del desarrollo del Proyecto Final del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web, durante el curso académico 2025-2026. A partir del análisis del sector y de la experiencia cercana con este ámbito, se identificaron carencias que podían resolverse mediante una solución tecnológica moderna, accesible y escalable. Nuestro compromiso es crear un entorno digital inclusivo, intuitivo y fiable, que contribuya a dignificar el sector de los cuidados y a facilitar la vida de las personas que más lo necesitan.

Nuestro equipo está formado por profesionales del ámbito tecnológico y social que comparten una misma visión: facilitar el acceso a servicios de atención y cuidado de forma sencilla, segura y transparente. Creemos firmemente que el cuidado de las personas debe estar basado en la confianza, la profesionalidad y la cercanía, y por ello trabajamos para conectar a usuarios con empresas especializadas en el sector. El proyecto nace en el contexto del desarrollo del Proyecto Final del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web, durante el curso académico 2025-2026. A partir del análisis del sector y de la experiencia cercana con este ámbito, se identificaron carencias que podían resolverse mediante una solución tecnológica moderna, accesible y escalable. Nuestro compromiso es crear un entorno digital inclusivo, intuitivo y fiable, que contribuya a dignificar el sector de los cuidados y a facilitar la vida de las personas que más lo necesitan.

El proyecto nace en el contexto del desarrollo del Proyecto Final del Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web, durante el curso académico 2025-2026. A partir del análisis del sector y de la experiencia cercana con este ámbito, se identificaron carencias que podían resolverse mediante una solución tecnológica moderna, accesible y escalable. Nuestro compromiso es crear un entorno digital inclusivo, intuitivo y fiable, que contribuya a dignificar el sector de los cuidados y a facilitar la vida de las personas que más lo necesitan.

- **Ámbito Social y Comercial:** El proyecto tiene un doble enfoque:
 - **Social:** Facilita la búsqueda y solicitud de ayuda a domicilio.
 - **Comercial:** Implica la contratación de servicios profesionales, generando oportunidades económicas para empresas y cuidadores del sector.
- **Valor Añadido:** La plataforma se distingue por permitir a los usuarios escoger la empresa que mejor se adapte a sus necesidades "con un solo click", basándose en valoraciones de otros usuarios y ofreciendo un servicio de resolución de conflictos.
- **Asistencia al Usuario:** Se ofrece asistencia a través de vídeos guiados, chat y llamada para las personas que necesiten ayuda para manejar la aplicación.

Miembros del equipo



Eve Gil
CEO & FUNDADORA
Estratega con más de 10 años de experiencia en el ámbito sociosanitario y gestión de equipos.



JJ Espinosa
CTO & CO-FUNDADOR
Responsable del desarrollo tecnológico y de garantizar una plataforma segura y accesible.



BoliT E.G.
DIRECTORA DE OPERACIONES
Especialista en coordinar la red de cuidadores y asegurar la calidad de los servicios.



Miguel D.
ATENCIÓN AL CLIENTE
Dedicado a ofrecer soporte humano y resolver dudas para una experiencia excelente.

[Inicio](#) [Contacto](#) [Sobre Nosotros](#) [Manuales](#) [Aspectos legales](#) [Política de privacidad](#) [Política de cookies](#) [ES](#) | [EN](#)

Ilustración 50: Miembros.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The screenshot shows the CuidaDos website's manual section. At the top, there is a logo with a heart and the text 'CuidaDos'. On the right side of the header are 'Registrarse' and 'Iniciar Sesión' buttons. Below the header, there is a back button labeled 'Volver' and a title 'Manuales'. There are two cards displayed: one for 'Guía de Inicio' (with a preview image of a book titled 'guía de inicio') and one for 'Protocolo Profesional' (with a preview image of a book titled 'protocolo profesional'). Both cards have a 'Descargar PDF' button at the bottom. At the bottom of the page, there is a footer with links: 'Inicio', 'Contacto', 'Sobre Nosotros', 'Manuales', 'Aspectos legales', 'Política de privacidad', 'Política de cookies', 'ES | EN'.

Ilustración 51: Manuales.

The screenshot shows the 'Registrarse' (Register) page. The title 'Registrarse' is at the top. Below it are two buttons: 'Registrarse como usuario' (Register as user) and 'Registrarse como empresa' (Register as company). At the bottom of the page is a link 'Ya tengo cuenta' (I already have an account).

Ilustración 52: Registrarse.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

The screenshot shows the registration page for the CuidaDos application. At the top right are 'Registrarse' and 'Iniciar Sesión' buttons. Below them is a back arrow labeled '← Volver'. The form fields are organized into sections: 'Nombre de usuario' (with a placeholder 'Nombre de usuario'), 'Primer Apellido' (with a placeholder 'Primer Apellido') and 'Segundo Apellido' (with a placeholder 'Segundo Apellido'), 'Fecha de nacimiento' (with a placeholder 'dd/mm/aaaa') and 'DNI' (with a placeholder 'DNI'), 'Teléfono' (with a placeholder 'Teléfono'), 'Email' (with a placeholder 'Email'), 'Dirección' (with a placeholder 'Dirección'), 'Localidad' (with a placeholder 'Localidad'), 'CP' (with a placeholder 'CP') and 'Comunidad Autónoma' (with a dropdown placeholder 'Seleccione una comunidad'), 'Contraseña' (with a placeholder 'Contraseña') and 'Repetir contraseña' (with a placeholder 'Repetir contraseña'). A checkbox at the bottom left states 'He leído y acepto los Términos y Condiciones y la Política de Privacidad, y autorizo el tratamiento de mis datos personales.' A 'Registrarse' button is located at the bottom center. At the very bottom, there is a footer navigation bar with links: Inicio, Contacto, Sobre Nosotros, Manuales, Aspectos legales, Política de privacidad, Política de cookies, and language links ES | EN.

Ilustración 53: Registrarse como usuario.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

The screenshot shows the registration process for a company on the CuidaDos website. At the top right are 'Registrarse' and 'Iniciar Sesión' buttons. Below them is a 'Volver' link. The form fields include:

- Nombre de la empresa (with placeholder 'Nombre de la empresa')
- CIF (with placeholder 'CIF')
- Descripción (with placeholder 'Descripción')
- Teléfono (with placeholder 'Teléfono')
- Email (with placeholder 'Email')
- Dirección (with placeholder 'Dirección')
- Localidad (with placeholder 'Localidad')
- CP (with placeholder 'CP') and Comunidad Autónoma (with placeholder 'Seleccione una comunidad' and a dropdown arrow)
- Contraseña (with placeholder 'Contraseña') and Repetir contraseña (with placeholder 'Repetir contraseña' and two eye icon icons)
- A checkbox for accepting terms and conditions and privacy policy.
- A 'Registrarse' button at the bottom.

At the bottom, there's a navigation bar with links: Inicio, Contacto, Sobre Nosotros, Manuales, Aspectos legales, Política de privacidad, Política de cookies, and language links ES | EN.

Ilustración 54: Registrarse como empresa.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

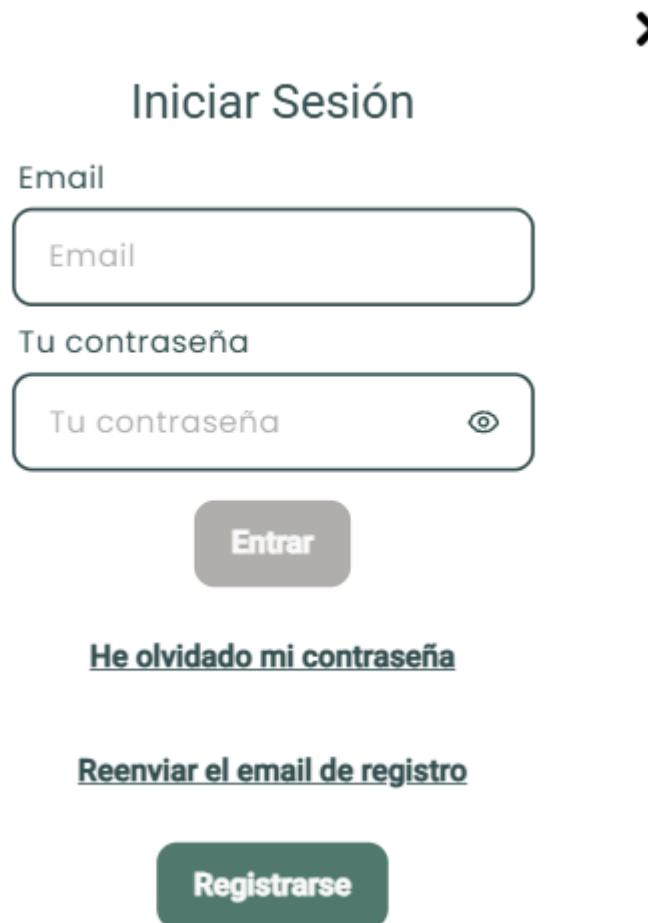


Ilustración 55: Iniciar sesión.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Ilustración 56: Recuperar contraseña.

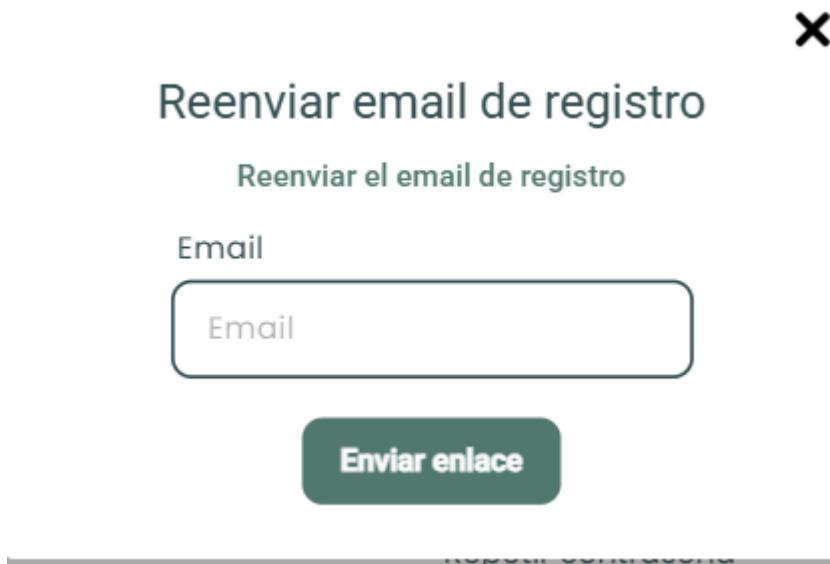


Ilustración 57: Reenviar correo.

USUARIO CON SESIÓN

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The screenshot shows the 'Mensajes' (Messages) page of the CuidaDos application. At the top, there is a header with the CuidaDos logo, user icons, and a 'Cerrar Sesión' (Logout) button. Below the header, there is a breadcrumb navigation with a 'Volver' (Back) link. The main title 'Mensajes' is centered above a table. The table has columns for 'Emisor' (Sender), 'Receptor' (Recipient), 'Asunto' (Subject), and 'Fecha Recepción' (Receipt Date). There are two rows of data in the table. Each row includes a 'Borrar' (Delete) button and a 'Ver' (View) button. At the bottom of the table, there is a dropdown for 'Elementos por página' (Items per page) set to 5, and a pagination area showing '1 - 2 de 2'. Below the table, there is a footer with links: Inicio, Contacto, Sobre Nosotros, Manuales, Aspectos legales, Política de privacidad, Política de cookies, and language links ES | EN.

Ilustración 58: Mensajes.

The illustration shows a message composition form. It starts with a 'Para:' field, followed by an 'Asunto:' field. Below these is a large text area with a green border labeled 'MENSAJE' at the top, containing the placeholder text 'Escribe aquí el mensaje...'. At the bottom of the form is a large 'Enviar' (Send) button.

Ilustración 59: Enviar mensaje.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The screenshot shows a table titled 'Notificaciones' (Notifications). The table has three columns: 'Nombre' (Name), 'Notificación' (Notification), and 'Fecha' (Date). There are no rows of data in the table. Below the table is a pagination control with a dropdown menu set to '5', showing '0 de 0' items, and navigation arrows.

Nombre	Notificación	Fecha
Elementos por página: <input type="button" value="5"/> 0 de 0 < < > >		

Ilustración 60: Notificaciones.

The screenshot shows a single button labeled 'Darse de baja' (Unsubscribe) centered on the page. Below the button is a horizontal navigation bar with links: Inicio, Contacto, Sobre Nosotros, Manuales, Aspectos legales, Política de privacidad, Política de cookies, and ES | EN.

Ilustración 61: Darse de baja.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

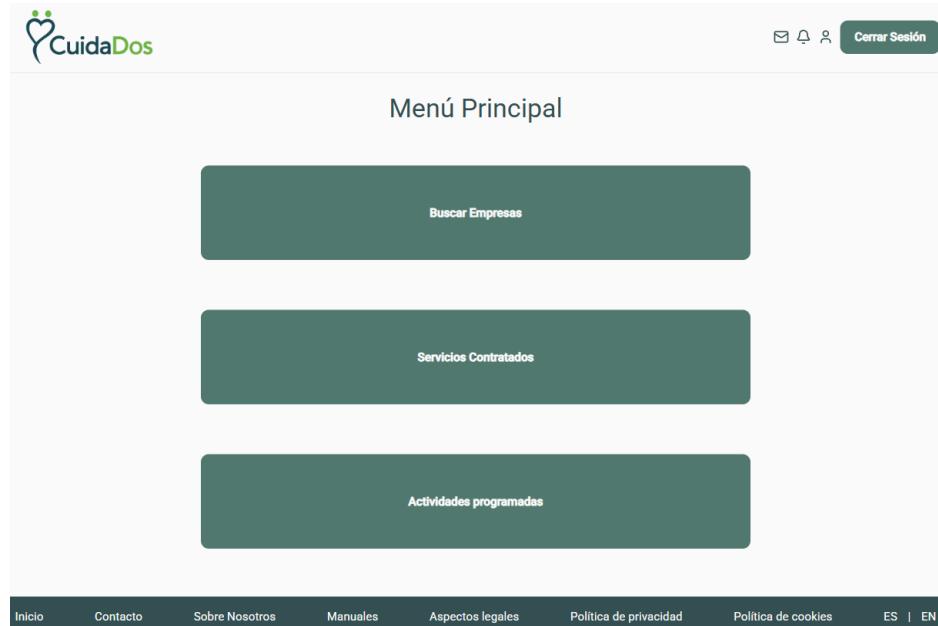
CLIENTE

Ilustración 62: Home.

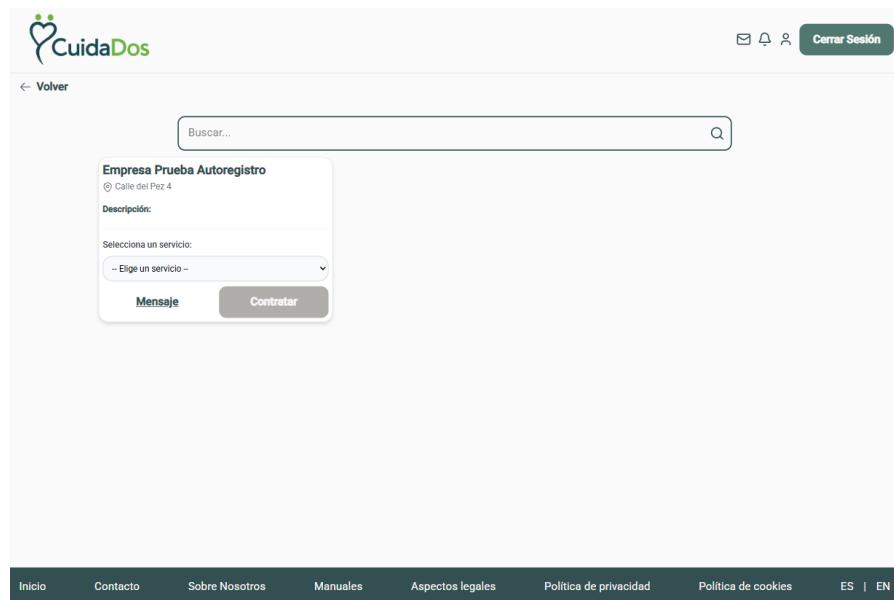


Ilustración 63: Buscar empresa.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The screenshot shows the 'Contratos' (Contracts) section of the CuidaDos platform. It lists a single contract with the ID 'df0707b7-71df-4472-a670-6e4a4e5e936e' and the date '03/02/2026'. There are buttons for 'Cancelar' (Cancel) and 'Ver detalles' (View details). Below the table, there's a pagination area showing 'Items per page: 5' and '1 - 1 of 1'. At the bottom, there are links for 'Inicio', 'Contacto', 'Sobre Nosotros', 'Manuales', 'Aspectos legales', 'Política de privacidad', 'Política de cookies', and language options 'ES | EN'.

Ilustración 64: Servicios contratados.

Confirmación de Cancelación de Contrato

¡Atención! Estás a punto de cancelar un contrato de servicio. Al confirmar, el contrato seleccionado será anulado en la plataforma CuidaDos. ¿Qué sucederá ahora?

- **Registro del Cambio:** La cancelación se registrará inmediatamente en el sistema y se actualizará su calendario.
- **Comunicación:** Recibirás una notificación de confirmación por correo o dentro de la aplicación.
- **Notificación a la Otra Parte.**

Si tiene alguna pregunta o surge un conflicto, recuerde que CuidaDos actúa como intermediario.

[Continuar con el contrato](#)

[Cancelar contrato](#)

Ilustración 65: Cancelar servicio.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

×

Detalles del contrato:

ID del contrato: df0707b7-71df-4472-a670-6e4a4e5e936e

Fecha de creación: 2026-02-03

Fecha de Inicio: 2026-02-03

Fecha de fin: Sin finalizar

Estado: activo

Detalles de la actividad:

Hora contratada: 13:00:00

Día de la semana contratado: lunes

Nombre del servicio: comida

Nombre del contratante (Cliente): Cliente Prueba
Autorregistro

Nombre del contratado (Empresa): Empresa Prueba
Autoregistro

Ilustración 66: Ver detalles servicio.

The screenshot shows a calendar for February 2026 with a specific activity highlighted on the 9th. The activity details are as follows:

Empresa	Nombre Servicio	Día	Hora	Lugar
Empresa Prueba Autoregistro	comida	lunes	13:00:00	Calle Prueba 1, null, null

At the bottom, there are navigation links for the website: Inicio, Contacto, Sobre Nosotros, Manuales, Aspectos legales, Política de privacidad, Política de cookies, and language options ES | EN.

Ilustración 67: Actividades programadas.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Nombre de usuario
Cliente Prueba Autorregistro

Primer Apellido Segundo Apellido
Pérez García

Teléfono
666777888

Email
cliente@test.com

Dirección
Calle Prueba 1

Localidad
Localidad

CP Comunidad Autónoma
CP Seleccione una comunidad ▾

Cancelar modificación Modificar

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 68: Modificar perfil.

EMPRESA

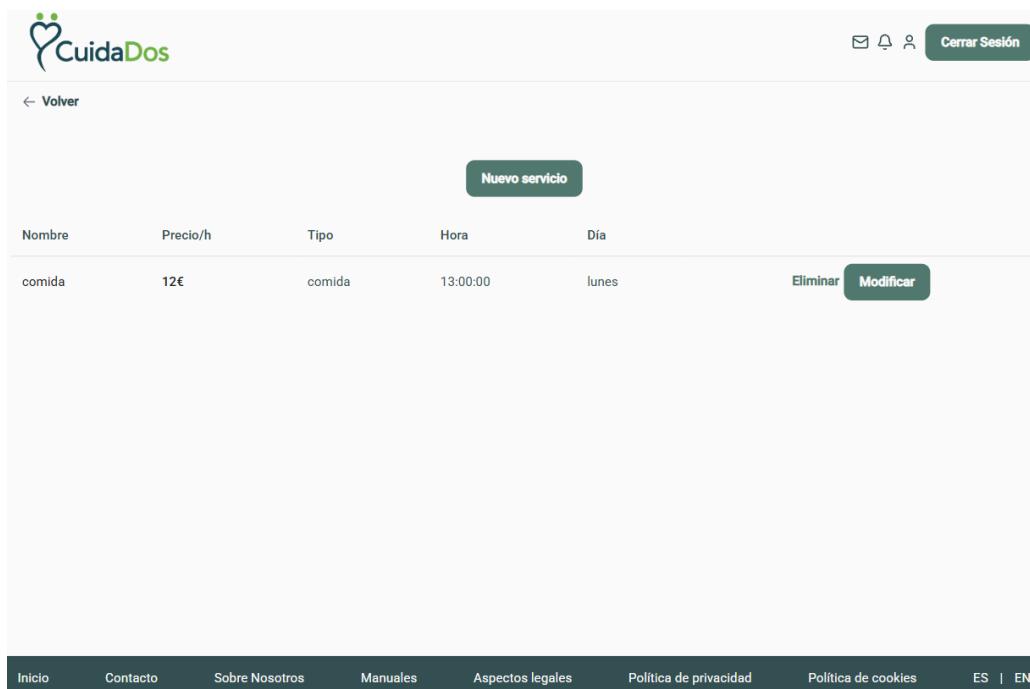
Gestionar servicios

Ver todos los contratos

Actividades programadas

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 69: Home.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Nombre	Precio/h	Tipo	Hora	Día	
comida	12€	comida	13:00:00	lunes	Eliminar Modificar

Ilustración 70: Gestión ofertas.



Descripción de la oferta
comidas

Servicio Global
comida (comida)

Precio/h(€)
12

Horario Disponible
lunes a las 13:00:00

Actualizar Cambios

Ilustración 71: Modificar oferta.

X

Advertencia: Eliminación de Servicio Ofertado

Está a punto de eliminar un servicio de su catálogo. Si confirma esta acción, el servicio será eliminado permanentemente.

Consecuencias de la Eliminación:

1. **Cese de Oferta:** Los usuarios ya no podrán buscar ni contratar este servicio.
2. **Pérdida de Visibilidad:** El servicio y sus detalles serán retirados de los resultados de búsqueda.

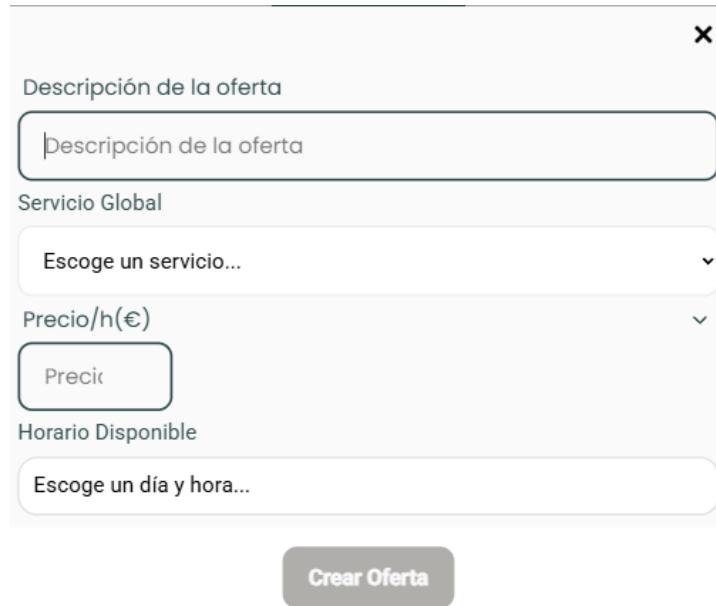
Importante sobre Contratos Activos:

Si este servicio ya ha sido contratado, le recomendamos gestionar esos contratos para evitar incidencias.

[Cancelar acción](#)

[**Eliminar**](#)

Ilustración 72: Cancelar oferta.



Descripción de la oferta
[Input: Descripción de la oferta]

Servicio Global
[Input: Escoge un servicio...]

Precio/h(€)
[Input: Precio]

Horario Disponible
[Input: Escoge un día y hora...]

Crear Oferta

Ilustración 73: Crear oferta.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

ID del Contrato	Fecha	
df0707b7-71df-4472-a670-6e4a4e5e936e	03/02/2026	Cancelar Ver detalles
d70d23c7-5d94-4431-97ff-7d958d524f9d	13/12/2000	Cancelar Ver detalles

Items per page: 5 | 1 - 2 of 2 | < > |

Ilustración 74: Ver contratos.

Cliente	Nombre Servicio	Día	Hora	Lugar	
Cliente Prueba Autorregistro	comida	lunes	13:00:00	Calle Prueba 1, null, null	Cancelar
Desconocido	comida	lunes	13:00:00	undefined, undefined, undefined	Cancelar

Items per page: 5 | 0 of 0 | < > |

Ilustración 75: Ver actividades.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Nombre de la empresa
Empresa Prueba Autoregistro

Descripción
Descripción

Teléfono
600112233

Email
empresa@test.com

Dirección
Calle del Pez 4

Localidad
Madrid

CP
28001

Comunidad Autónoma

Cerrar Sesión

Cancelar modificación Modificar

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 76: Modificar perfil.

ADMINISTRADOR

Menú Principal

Ver dashboard

Gestionar usuarios

Gestionar Empresas

Gestionar servicios

Gestionar horarios

Cerrar Sesión

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 77: Home.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The screenshot shows a dashboard titled "Contratos" with a large blue circular icon. Below it, the text "Contratos totales: 2 Contratos totales activos: 2 Contratos totales no activos: 2" is displayed. A callout box highlights "Usuarios totales actuales: 3". The top right corner features a "Cerrar Sesión" button and notification icons.

The screenshot shows a login page with a green line graph titled "LOGIN" showing user activity over a week. The x-axis labels are Lun, Mar, Mie, Jue, Vie, Sab, and Dom. Below the graph is a navigation bar with links: Inicio, Contacto, Sobre Nosotros, Manuales, Aspectos legales, Política de privacidad, Política de cookies, and language links ES | EN. The letter "I" is positioned below the "Inicio" link.

Ilustración 78: Ver Dashboard.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

Nuevo Cliente

Nombre	Apellidos	Email
Cliente Prueba Autorregistro	Pérez García	cliente@test.com

Buscar por nombre o email Buscar...

Items per page: 10 1 – 1 of 1 | < < > >|

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 79: Gestión usuarios.

Nueva Empresa

Nombre	Email
Empresa Prueba Autoregistro	empresa@test.com

Buscar por nombre o email Buscar...

Items per page: 10 1 – 1 of 1 | < < > >|

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 80: Gestión empresa.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

SERVICIO

Nombre
Nombre

Tipo
Tipo

Añadir servicio

Listado de Servicios

Nombre	Tipo
comida	comida

Buscar por nombre Buscar...

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 81: Gestión servicios.

HORA DE INICIO

Día de la semana

Añadir horario

Lista de horarios

Hora	Día de la semana
13:00:00	lunes

Inicio Contacto Sobre Nosotros Manuales Aspectos legales Política de privacidad Política de cookies ES | EN

Ilustración 82: Gestión horarios.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

6.3. Iteración y Mejora de Interfaces

El diseño no fue estático. Tras las primeras pruebas con usuarios (Test de guerrilla), se detectaron problemas de usabilidad que requirieron iteraciones.

6.3.1. Caso de Estudio: El Botón de "Eliminar"

- **Versión Inicial:** El botón de "Cancelar Contrato" era un simple icono de papelera roja situado junto al nombre de la empresa.
- **Problema Detectado:** Varios usuarios pulsaron el botón por error al intentar abrir el chat, debido a que los elementos táctiles estaban muy juntos (*Touch Target* insuficiente en móviles).
- **Solución Iterada:**
 1. Se movió el botón a una zona separada.
 2. Se añadió texto "Cancelar Contrato" en lugar de solo un ícono.
 3. Se implementó el **Modal de Confirmación** (CancelModal) que interrumpe la acción.

6.4. Validación de Interfaces

Una vez implementada la interfaz en Angular, se sometió a herramientas de auditoría automática para validar su calidad técnica y accesibilidad.

6.4.1. Auditoría Lighthouse (Google)

Se ejecutó Lighthouse en el navegador Chrome para medir el rendimiento y las buenas prácticas.

- **Performance:** 95/100 (Gracias al uso de Angular Signals y carga perezosa de imágenes).
- **Accessibility:** 100/100 (Todos los inputs tienen aria-label, contraste de colores correcto).
- **SEO:** 90/100 (Metadatos correctos en index.html).

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

6.4.2. Validación de Contraste (WCAG)

Para asegurar que las personas con visión reducida puedan usar la app, se verificó que la combinación de colores cumpla el estándar **WCAG AA**.

- **Texto Gris sobre Fondo Blanco:** Ratio 5.4:1 (Pasa).
- **Texto Blanco sobre Botón Azul:** Ratio 7.1:1 (Pasa AAA).

6.4.3. Pruebas de Responsividad (RWD)

Se validó la correcta visualización en los siguientes viewports:

1. **Móvil (375px):** iPhone SE / Android genérico. El menú pasa a ser hamburguesa.
2. **Tablet (768px):** iPad Mini. El grid pasa a 2 columnas.
3. **Desktop (1024px+):** Portátiles. Barra lateral visible y grid de 3 columnas.

Ejemplos:



Ilustración 83: Móvil.

SERVICIOS

Contrata servicios fácilmente
Encuentra y contrata servicios de empresas profesionales dedicadas al cuidado personal, acompañamiento y bienestar.

Gestión en un solo lugar

Resolución de problemas
Nuestro equipo te acompaña en cada paso para que encuentres soluciones rápidas, confiables y adaptadas a lo que necesitas.

Ilustración 84: Tablet.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Ilustración 85: Pc.

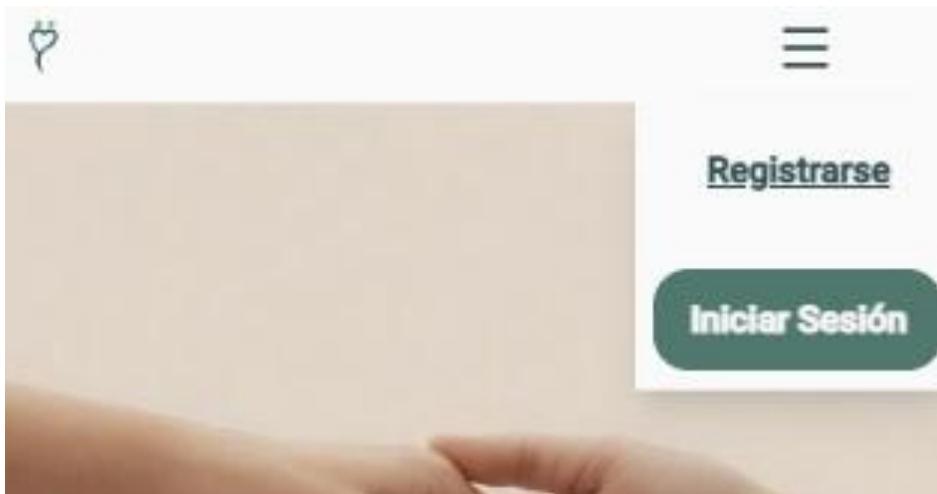


Ilustración 86: Menú móvil.

CAPÍTULO 7: DICCIONARIO DE DATOS (DATA DICTIONARY)

El modelo de datos se ha implementado en **PostgreSQL** alojado en Supabase. A continuación, se detallan las tablas que conforman el esquema, especificando los tipos de datos exactos utilizados y las relaciones de integridad referencial.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

7.1. Tabla: public.Usuario (Entidad Base)

Tabla padre que almacena la información común a todos los roles. Utiliza una estrategia de herencia o extensión de identidad vinculada a la autenticación.

Campo	Tipo de Dato	Restricción	Descripción Semántica
id_usuario	uuid	PK, FK	Clave primaria. Vincula 1:1 con auth.users de Supabase.
nombre	text		Nombre de pila o identificador principal.
email	varchar	UNIQUE	Correo electrónico de contacto y acceso.
telef	varchar		Teléfono de contacto móvil o fijo.
rol	text		Define el tipo de actor: 'cliente', 'empresa', 'admin'.
estado	bool	DEFAULT true	true = Activo, false = Bloqueado (Soft Delete).
fecha_registro	date	DEFAULT now()	Fecha de creación de la cuenta.

Tabla 57: Public usuario.

7.2. Tabla: public.Cliente (Extensión)

Tabla que extiende a Usuario para almacenar los datos específicos de las personas físicas que contratan servicios.

Campo Dato	Tipo de Dato	Restricción	Descripción Semántica
id_cliente	uuid	PK, FK	Referencia a Usuario.id_usuario.
dni	varchar	UNIQUE	Documento Nacional de Identidad.
apellidos	varchar		Apellidos del cliente (según diagrama ap...).
fechnac	date		Fecha de nacimiento para control de edad.
direccion	varchar		Domicilio habitual para recibir el servicio.
localidad	text		Municipio de residencia.
codpostal	text		Código Postal.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

comunidad	text		Comunidad Autónoma.
-----------	------	--	---------------------

Tabla 58: Public cliente.

7.3. Tabla: public.Empresa (Extensión)

Tabla que extiende a Usuario para almacenar los datos fiscales y comerciales de los proveedores de servicios.

Campo Dato	Tipo de Dato	Restricción	Descripción Semántica
id_empresa	uuid	PK, FK	Referencia a Usuario.id_usuario.
cif	varchar	UNIQUE	Código de Identificación Fiscal.
descripcion	text		Texto promocional ("Sobre nosotros").
direccion	varchar		Domicilio fiscal o sede.
localidad	text		Municipio de operación.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

codpostal	text		Código Postal.
-----------	------	--	----------------

Tabla 59: Public empresa.

7.4. Tabla: public.Administrador (Extensión)

Tabla para gestionar los permisos de los administradores del sistema.

Campo	Tipo de Dato	Restricción	Descripción Semántica
id_administrador	uuid	PK, FK	Referencia a Usuario.id_usuario.

Tabla 60: Public administrador.

<i>Nota</i>	Hereda resto de datos (nombre, email) de Usuario.
-------------	---

7.5. Tabla: public.Servicio (Catálogo)

Catálogo maestro de servicios gestionado por los administradores. Define "qué" se puede contratar.

Campo	Tipo de Dato	Restricción	Descripción Semántica
id_servicio	uuid	PK	Identificador único del tipo de servicio.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

id_admin	uuid	FK	Qué administrador creó o gestiona este servicio.
nombre	varchar		Nombre corto (ej: "Fisioterapia").
tipo_servicio	text		Categoría o agrupación.

Tabla 61: Catalogo- servicios.

7.6. Tabla: public.Horario (Catálogo de Tiempo)

Define los bloques temporales o franjas horarias base que el sistema permite utilizar.

Campo	Tipo de Dato	Restricción	Descripción Semántica
id_horario	uuid	PK	Identificador de la franja.
id_admin	uid	FK	Administrador que definió la franja.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

dia_semana	text		Día (ej: "Lunes", "Martes").
hora	time		Hora de inicio/referencia del bloque.

Tabla 62: Public horario.

7.7. Tabla: public.Servicio_Horario (Oferta Real)

Tabla Pivot (N:M). Es el corazón de la oferta. Vincula un Servicio genérico con una Empresa y un Horario específico, estableciendo un precio.

Campo	Tipo de Dato	Restricción	Descripción Semántica
id_servicio_horario	uuid	PK	Identificador único de la oferta concreta.
id_empresa	uuid	FK	Empresa que ofrece el servicio.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

id_servicio	uuid	FK	Qué servicio está ofreciendo.
id_horario	uuid	FK	En qué franja horaria lo ofrece.
precio	float4		Coste del servicio por hora/sesión.
descripcion	text		Detalles específicos de esta oferta.

Tabla 63: Ofertas -Servicio_horario.

7.8. Tabla: public.Contrato (Transacción)

Registra los acuerdos comerciales entre Clientes y Empresas basados en una oferta (Servicio_Horario).

Campo	Tipo de Dato	Restricción	Descripción Semántica
id_contrato	uuid	PK	Identificador del contrato.
id_cliente	uuid	FK	Usuario que contrata.
id_empresa	uuid	FK	Proveedor del servicio.
id_servicio_horario	uuid	FK	Referencia a la oferta específica aceptada.
estado	text		pendiente, activo, cancelado.
fecha_inicio	date		Inicio de la prestación.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

fecha_fin	date		Finalización prevista.
fecha_creacion	date	DEFAULT now()	Fecha de la solicitud.
dia_semana_contratado	text		Día pactado (Snapshot).
hora_contratada	time		Hora pactada (Snapshot).

Tabla 64: Contrato.

7.9. Tabla: public.Comunicacion (Mensajería)

Gestiona los mensajes y las notificaciones entre usuarios.

Campo	Tipo de Dato	Restricción	Descripción Semántica
id_comunicacion	uuid	PK	Identificador del mensaje.
id_emisor	uuid	FK	Usuario que envía.
id_receptor	uuid	FK	Usuario que recibe.
tipo_comunicacion	varchar		mensaje (chat) o notificación (sistema).
asunto	text		Título o contexto del mensaje.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

contenido	varchar		Cuerpo del mensaje.
fecha_envio	date	DEFAULT now()	Marca temporal.
leido	bool	DEFAULT false	Control de lectura.
eliminado_por_emisor	bool		Soft delete para el remitente.
eliminado_por_receptor	bool		Soft delete para el destinatario.

Tabla 65: Comunicaciones.

CAPÍTULO 8: IMPLEMENTACIÓN TÉCNICA (CÓDIGO FUENTE)

8.1. Modelos de Datos (Interfaces)

```
export interface IAdministradorModel {  
    id_usuario?: string;  
}
```

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

```
export interface ClienteModel {
    id_usuario?: string;
    ape1: string;
    ape2?: string;
    dni: string;
    direccion: string;
}
```

```
export interface ComunicacionModel {
    id_comunicacion?: string;
    tipo_comunicacion: 'mensaje' | 'notificacion';
    contenido: string;
    fecha_envio: Date;
    leido: boolean;
    id_receptor?: string;
    id_emisor?: string | null;
    asunto: string;
    eliminado_por_emisor?: boolean;
    eliminado_por_receptor?: boolean;
}
```

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

```
export interface ContratoModel {
    id_contrato?: number;
    estado: 'activo' | 'no activo';
    fecha_inicio: string;
    fecha_fin: string | null;
    dia_semana_contratado: 'lunes' | 'martes' | 'miercoles' | 'jueves' | 'viernes';
    hora_contratada: string;
    id_servicio_horario: string;
    fecha_creacion: string;
    id_cliente: string;
    id_empresa: string;
}
```

```
export interface EmpresaModel {
    id_usuario?: string;
    cif: string;
    direccion: string;
}
```

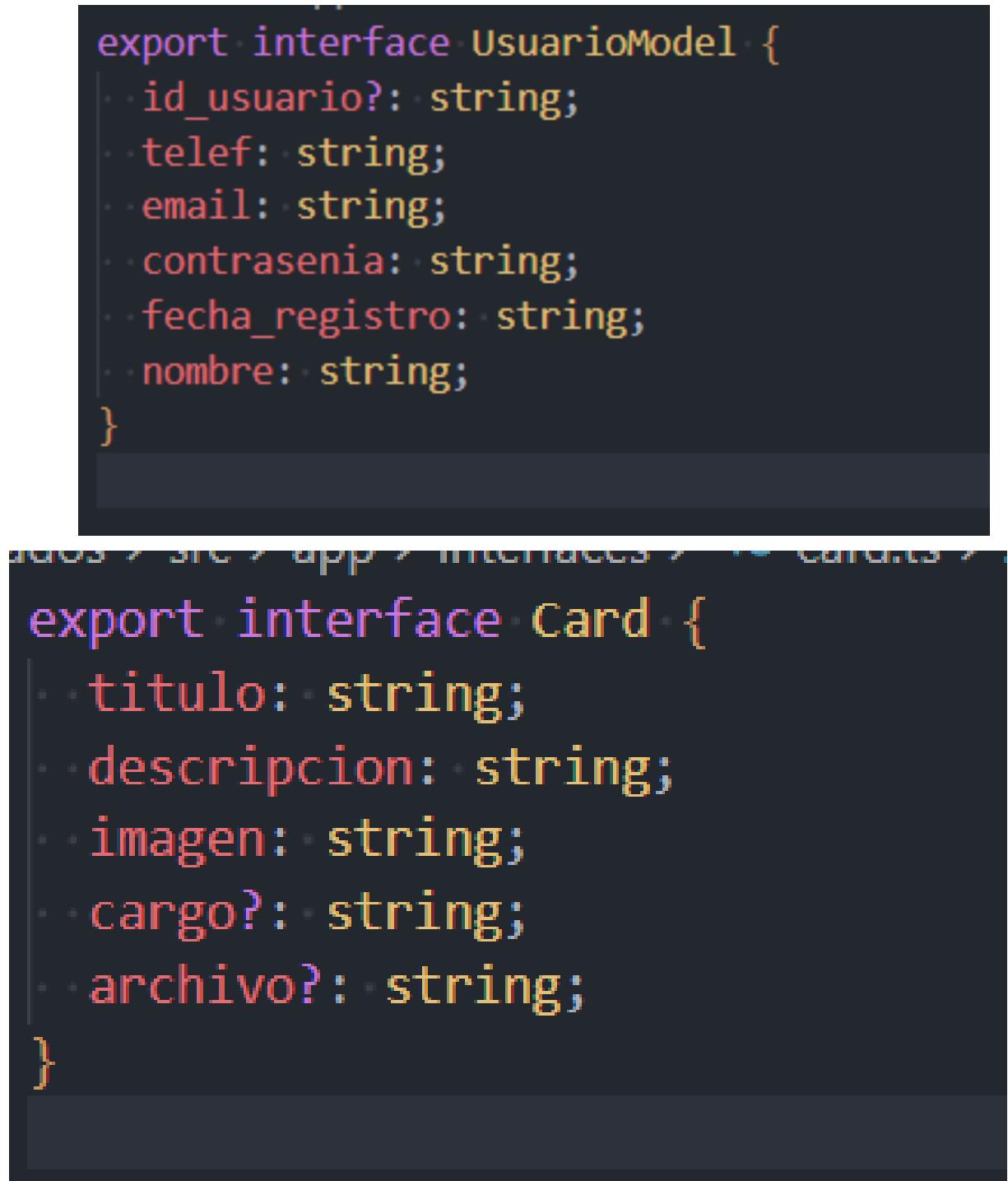
```
export interface HorarioModel {
    id_horario?: string;
    dia_semana: 'lunes' | 'martes' | 'miercoles' | 'jueves' | 'viernes';
    hora: string;
    id_admin: string;
}
```

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

```
export interface Servicio_HorarioModel {  
    id_servicio_horario: string;  
    id_servicio: string;  
    id_horario: string;  
    id_empresa: string;  
    precio: number;  
    descripcion: string;  
}
```

```
export interface ServicioModel {  
    id_servicio?: string;  
    nombre: string;  
    tipo_servicio: string;  
    id_admin: string;  
}
```

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes



```
export interface UsuarioModel {
    id_usuario?: string;
    telef: string;
    email: string;
    contrasenia: string;
    fecha_registro: string;
    nombre: string;
}

export interface Card {
    titulo: string;
    descripcion: string;
    imagen: string;
    cargo?: string;
    archivo?: string;
}
```

Ilustración 87: Modelos.

8.2. Servicios (Lógica de Negocio)

Cada uno de los archivos ts. Del programa, por ejemplo:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

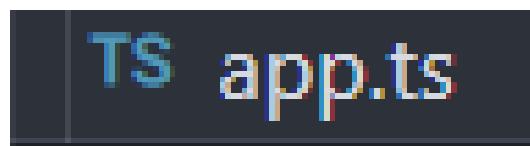


Ilustración 88: Ts.

8.3. Seguridad y Guardianes

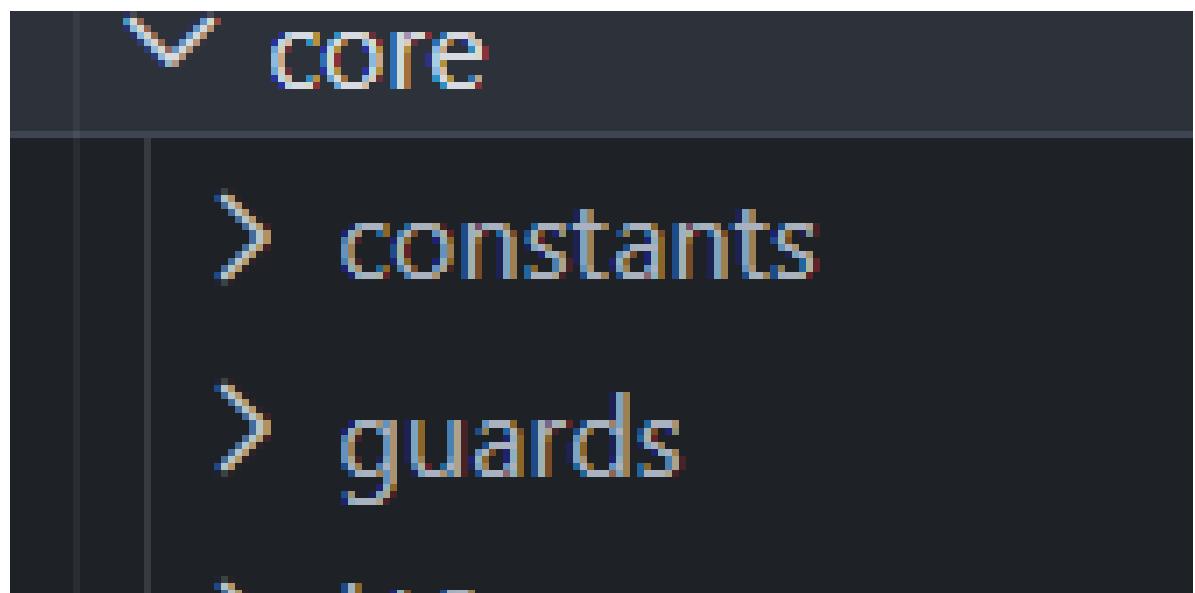


Ilustración 89: Carpeta guards.

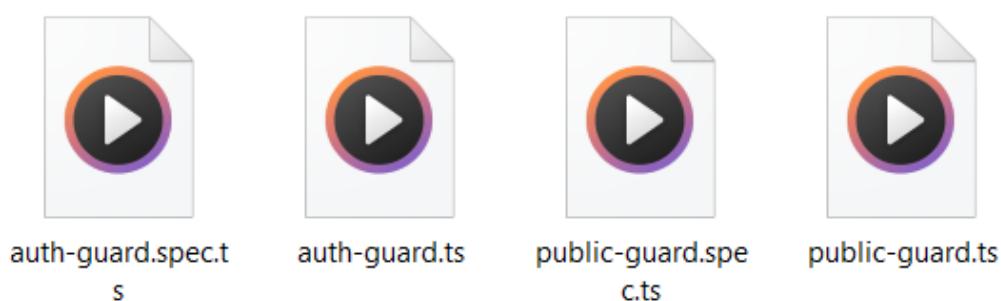


Ilustración 90: Guards.

8.4. Componentes y Vistas

Componentes

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

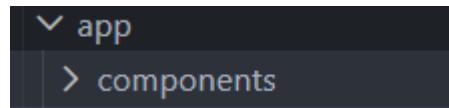


Ilustración 91: Componentes.

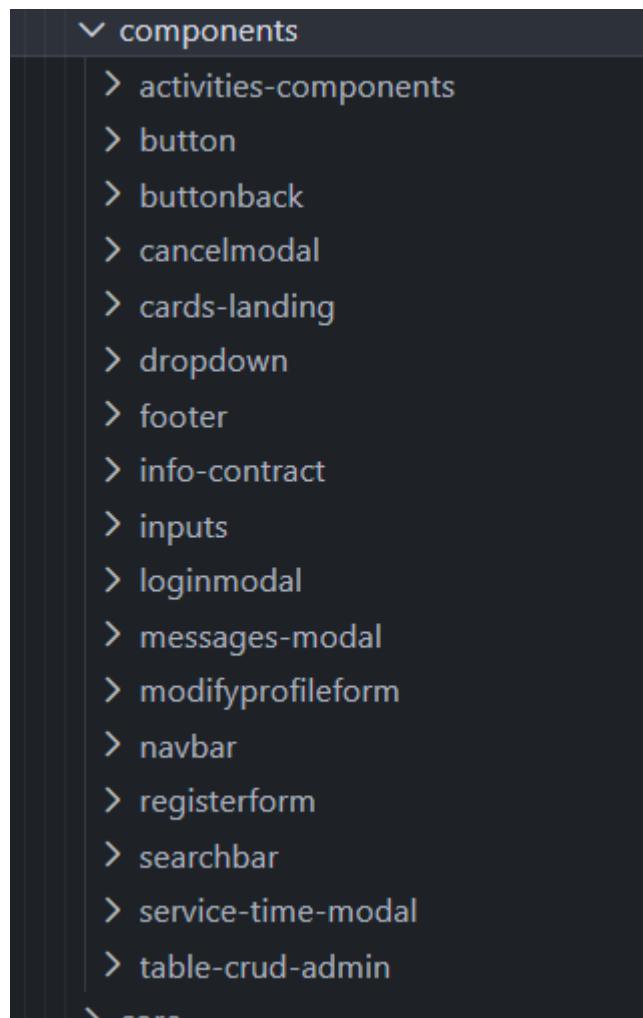


Ilustración 92: Componentes II.

Vistas

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

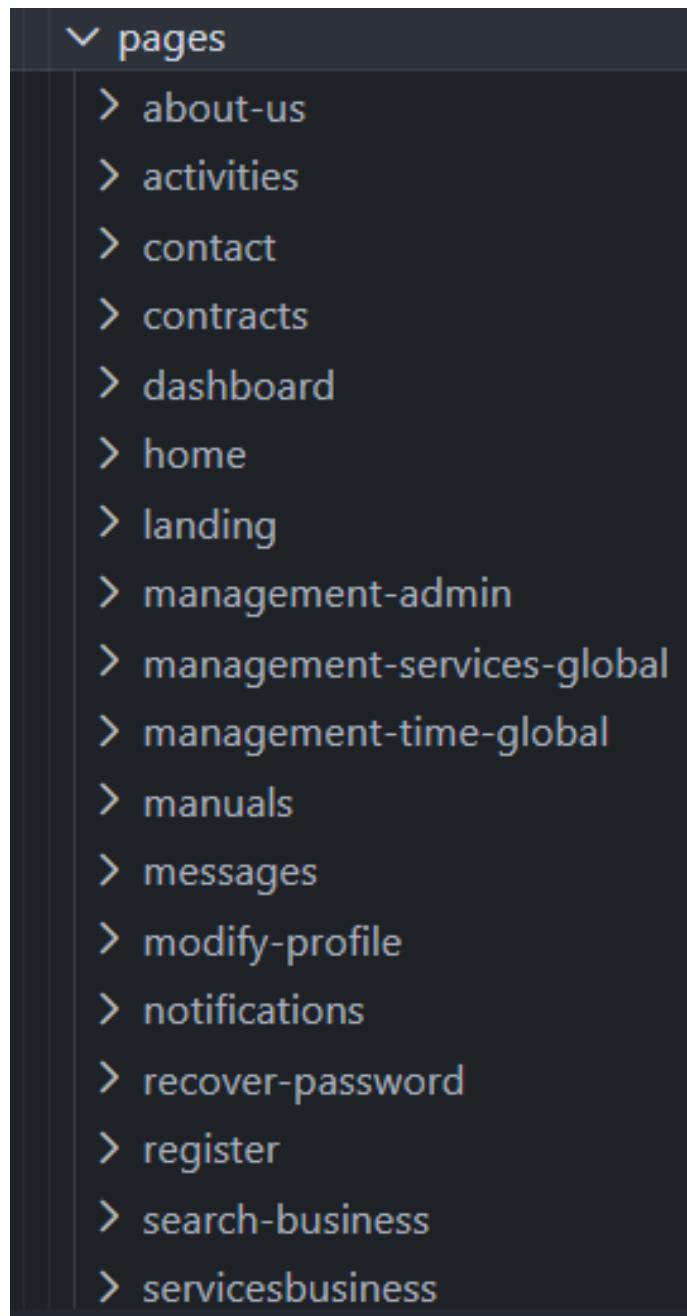


Ilustración 93: Vistas.

8.5. Estructura del Proyecto

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

The screenshot shows a file explorer window with the following directory structure:

```
✓ tfg-cuidados
  ✓ src
    ✓ app
      > components
      > core
      > interfaces
      > models
      > pages
      > services
      TS app.config.server.ts
      TS app.config.ts
      # app.css
      <> app.html
      TS app.routes.server.ts
      TS app.routes.ts
      TS app.spec.ts
      TS app.ts
      > assets
      > environments
      ❀ custom-theme.scss
      ★ favicon.ico
      <> index.html
      .tsconfig
```

The files listed under the 'app' folder are color-coded: 'app.config.server.ts', 'app.config.ts', 'app.css', 'app.html', 'app.routes.server.ts', 'app.routes.ts', 'app.spec.ts', 'app.ts', 'assets', 'environments', 'custom-theme.scss', 'favicon.ico', and 'index.html'. The file 'app.config.server.ts' is highlighted with a blue background.

Ilustración 94: Estructura.

8.6. Configuración Global

Archivos de configuración del programa. Todos los json.

Proyecto: Desarrollo de Aplicaciones Web (DAW) Autora: Evelia Gil Paredes

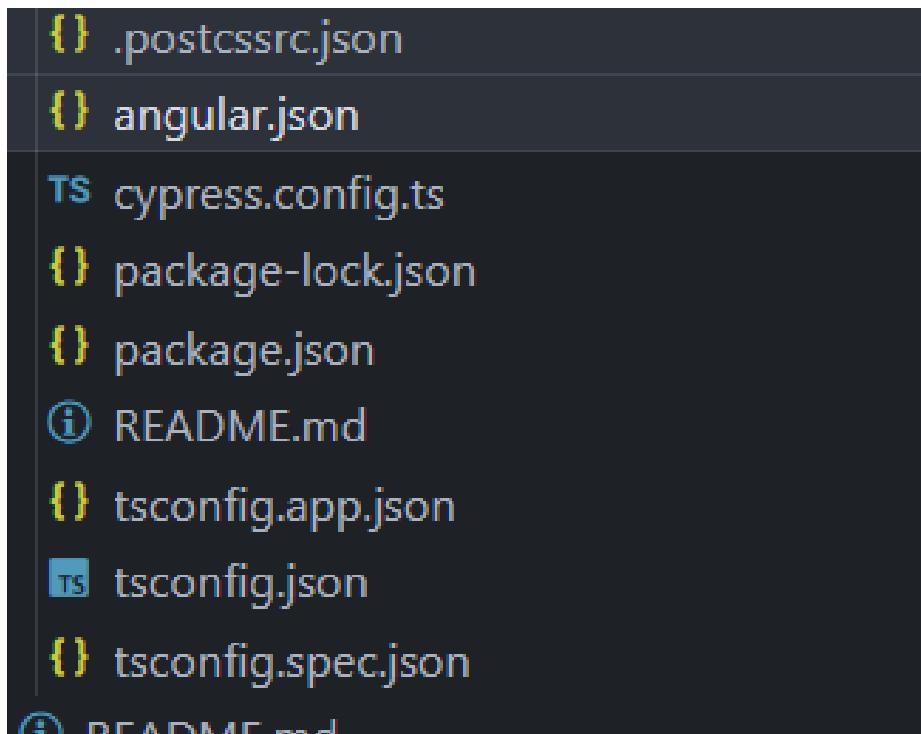


Ilustración 95: Conf. global.

CAPÍTULO 9: PRUEBAS Y VALIDACIÓN DEL SISTEMA

La fase de pruebas (Testing) tiene como objetivo garantizar que el software **Cuidados** cumple con los requisitos funcionales especificados y está libre de errores críticos antes de su puesta en producción. Se ha seguido una estrategia de pruebas de caja blanca (código) y caja negra (funcionalidad).

9.1. Plan de Pruebas (Master Test Plan)

Se ha diseñado un plan de pruebas integral que abarca tres niveles de abstracción:

1. **Pruebas Unitarias (Unit Testing):** Verificación de la lógica aislada (validadores, funciones de utilidad).
2. **Pruebas de Integración:** Verificación de la comunicación entre el Frontend (Angular) y el Backend (Supabase).
3. **Pruebas de Sistema (E2E):** Validación del flujo completo de usuario desde el navegador desplegado.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

9.2. Pruebas Unitarias (Validaciones)

Se ha utilizado el framework **Jasmine** y el ejecutor **Karma** (nativos en Angular) para testear la lógica de negocio crítica.

Ficha de Prueba: CP-01 Validación de Documentos

Atributo	Detalle
Identificador	CP-01
Nombre	Validación Algorítmica de DNI
Objetivo	Verificar que el sistema rechaza DNIs con letra incorrecta o formato inválido.
Precondición	El usuario está en el formulario de registro.
Entrada 1	12345678A (La letra A es incorrecta para este número).
Resultado Esperado	El control del formulario debe tener el error { invalidDni: true }.
Resultado Obtenido	OK. El formulario se bloquea y muestra "DNI Inválido".
Entrada 2	12345678Z (Letra correcta).
Resultado Esperado	El control pasa a estado VALID.
Resultado Obtenido	OK.
Estado Final	<input checked="" type="checkbox"/> PASSED

Tabla 66: Pruebas I .

Ficha de Prueba: CP-02 Inyección de Seguridad

Atributo	Detalle

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Identificador	CP-02
Nombre	Protección contra XSS en Chat
Objetivo	Verificar que el sistema sanea el HTML en los mensajes.
Entrada	<script>alert('Hacked')</script> enviado por el chat.
Resultado Esperado	El mensaje se muestra como texto plano, no se ejecuta el script.
Resultado Obtenido	OK. Angular escapa automáticamente el contenido.
Estado Final	<input checked="" type="checkbox"/> PASSED

Tabla 67: Pruebas II.

9.3. Pruebas de Integración (Conectividad)

Ficha de Prueba: CP-INT-01 Conexión con Supabase

Atributo	Detalle
Identificador	CP-INT-01
Nombre	Latencia de Base de Datos

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

Objetivo	Asegurar que la lectura de empresas tarda menos de 500ms.
Herramienta	Chrome DevTools (Network Tab).
Acción	Cargar la lista de empresas en el Dashboard.
Métrica	Tiempo de respuesta (TTFB).
Resultado	120ms (Gracias al cacheo de Vercel Edge).
Estado Final	<input checked="" type="checkbox"/> PASSED

Tabla 68: Pruebas III.

9.4. Pruebas de Despliegue (Producción)

Se procedió al despliegue final en la plataforma **Vercel** conectada al repositorio de GitHub evelia/cuidados-app.

Para desplegar el aplicativo se realizaron varios ajustes tanto en el proyecto, con VSC, como en Vercel.

En el proyecto: se usó el archivo Vercel.json para indicarle donde ha de encontrar el index.html.



```
{
  "rewrites": [
    {
      "source": "/(.*)",
      "destination": "/index.html"
    }
  ]
}
```

Ilustración 96: Vercel.json.

Y luego se desplegó el proyecto con:

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes



Ilustración 97: Build.

En Vercel:

1. Carpeta donde se guarda tu página web "traducida" y lista para usarse.

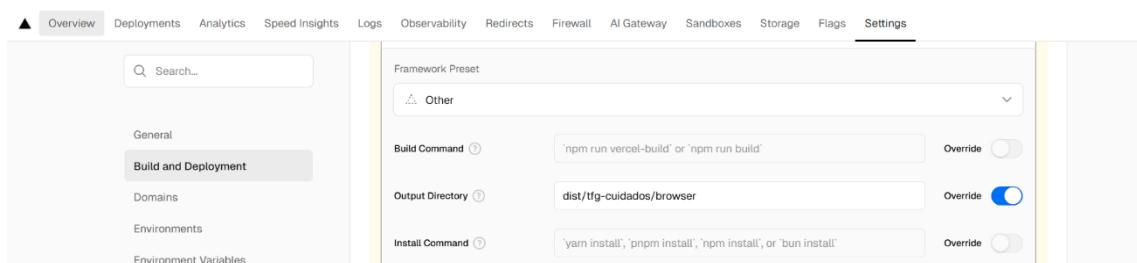


Ilustración 98: Carpeta app.

Como el despliegue del aplicativo daba problemas, he tenido que indicar la carpeta en la que Vercel debía buscar.

Proyecto: Desarrollo de Aplicaciones Web (DAW) **Autora:** Evelia Gil Paredes

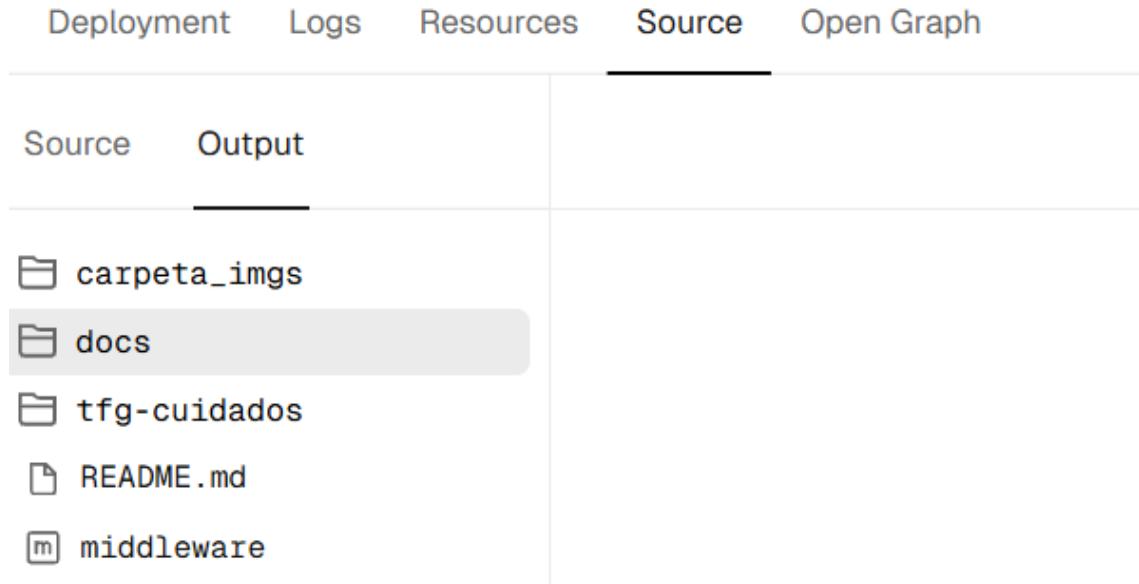


Ilustración 99: Problema.

Porque no buscaba en la indicada. Para ello se cambió el root Directory.

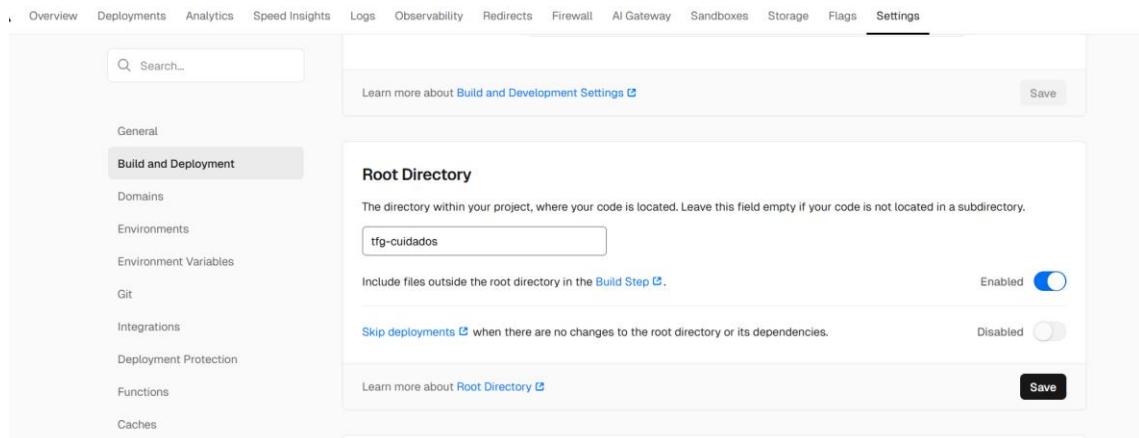


Ilustración 100: Root directory.

Verificación en Entorno Real

Se ha accedido a la URL pública del proyecto (<https://tfg-cuidados.vercel.app/>) desde diferentes dispositivos para validar la Responsividad.

CONCLUSIÓN DEL CAPÍTULO

Tras la ejecución de la batería de pruebas CP-01 a CP-INT-01 y la verificación visual en producción, se certifica que la versión 1.0 de **CuidaDos** es estable, segura y apta para su evaluación académica.