IES BERNAT EL FERRER

Desenvolupament d'aplicacions web

Segon Curs



**KanFlow**



*Autores:*

*Marc Ustero*

*y*

*Aniol Carceller*

Data d'entrega: 22/04/2025

Curs acadèmic 2024 /2025

**Resumen**

Este proyecto presenta el desarrollo de una aplicación web de tipo kanban para la gestión de tareas, inspirada en herramientas como Trello. Además de ofrecer funcionalidades básicas de organización visual de tareas en columnas, la plataforma incorpora un sistema de automatizaciones mediante bloques, que permite crear flujos de trabajo personalizados. El usuario puede definir desencadenadores (como la finalización de una tarea) y acciones correspondientes (como el envío automático de correos electrónicos). El documento describe el contexto, objetivos, análisis de requisitos, diseño, implementación y resultados del proyecto, así como los beneficios que aporta a la productividad y organización del trabajo.

**Palabras clave**

Kanban, Gestión de tareas, Automatización, Flujo de trabajo, Tableros, Tareas, Desencadenador, Correo electrónico, Organización, Productividad, Aplicación web, Sistema visual, Notificaciones automáticas, Interfaz interactiva, Usuarios, Herramienta colaborativa, Sistema de bloques, Acción automatizada, Eventos, Usabilidad.

**Keywords**

Kanban, Task management, Automation, Workflow, Boards, Tasks, Trigger, Email, Organization, Productivity, Web application, Visual system, Automatic notifications, Interactive interface, Users, Collaborative tool, Block-based system, Automated action, Events, Usability.

**Índex**

[**1 Introducción 4**](#_heading=h.qssp3r2nkzmr)

[**2 Contexto y motivación del proyecto 4**](#_heading=h.s1c029h99ggu)

[2.1 Objetivos del proyecto 4](#_heading=h.ow8m0kko7lf8)

[**3 Descripción 4**](#_heading=h.i3axl2fxohbz)

[3.1 Resultados esperados 4](#_heading=h.idpa13lwxou)

[3.2 Beneficios 5](#_heading=h.3sdd4rvzrv6u)

[**4 Análisis 5**](#_heading=h.sx64spj8sbkv)

[4.1 Requisitos funcionales 5](#_heading=h.qne3k67ior47)

[4.2 Requisitos no funcionales 5](#_heading=h.542fvm1s51p4)

[**5 Requisitos de datos 6**](#_heading=h.1azbe54olhik)

[5.1 Diagrama de classes 6](#_heading=h.7j992juws635)

[5.2 Diseño de base de datos 6](#_heading=h.qm00n02pqknz)

[5.3 Diseño del sistema 7](#_heading=h.85gl91ro55f)

[**6 Diseño 8**](#_heading=h.k92is0rnnr5l)

[Investigación de referencias y definición de tu propia idea 8](#_heading=h.qlz6t6uqjqzb)

[Segunda etapa: creación del sitemap 11](#_heading=h.qtw2021370lw)

[Wireflows 11](#_heading=h.7z1mafl6dvsb)

[Cuarta etapa: Prototipo de alta fidelidad 12](#_heading=h.l2obep2x7ywl)

[**7 Resultado Final 18**](#_heading=h.72dtfb4a3p2j)

[**8 Conclusiones 27**](#_heading=h.nu6q53o5g570)

[**9 Referencias 27**](#_heading=h.acgov7lug4w)

# Introducción

El proyecto Kanflow es una aplicación web enfocada en la gestión de tareas mediante sistemas de Kanban y flujos de trabajo personalizados. Esta herramienta está diseñada para ayudar tanto a usuarios individuales como a equipos a visualizar, planificar y gestionar su flujo de trabajo de manera eficiente e intuitiva.

# Contexto y motivación del proyecto

Actualmente, la gestión del tiempo y la organización de tareas son fundamentales para el éxito tanto a nivel personal como profesional. Aplicaciones como Trello o Jira han demostrado la eficacia del sistema Kanban. Con **Kanflow**, se pretende crear una alternativa propia que permita a los usuarios gestionar su flujo de trabajo de forma visual, clara y adaptada a sus necesidades, al mismo tiempo que se aplican conocimientos adquiridos de desarrollo web.

## Objetivos del proyecto

* + 1. Objetivos generales de producción
* Desarrollar una aplicación web funcional y usable para gestionar tareas con el sistema Kanban.
* Crear una interfaz intuitiva con un diseño moderno y adaptable.
* Permitir la creación de flujos de trabajo personalizados.
* Facilitar la organización de tareas en columnas y estados.
  + 1. Objetivos de aprendizaje
* Configurar el entorno local con Laragon, instalando y gestionando el servidor web y la base de datos.
* Utilizar Laravel para crear rutas, controladores, modelos y vistas, y gestionar la base de datos con migraciones.
* Integrar Vue.js para desarrollar componentes interactivos en el front-end y gestionar el estado de la aplicación.

# Descripción

Kanflow es una aplicación web pensada para permitir a los usuarios gestionar su flujo de trabajo a través de un sistema visual basado en columnas. El usuario podrá crear diferentes tableros, añadir listas (columnas) y añadir tarjetas (tareas). También podrá configurar flujos de trabajo personalizados, mover tareas entre columnas y tener una visión clara del estado de sus proyectos.

## Resultados esperados

Se espera que, una vez finalizado el proyecto, la aplicación Kanflow permita a cualquier usuario:

* Crear y gestionar tableros y tareas.
* Organizar su trabajo de forma visual.
* Personalizar sus flujos de trabajo según sus necesidades.
* Disfrutar de una interfaz clara, rápida y fácil de usar.
* Utilizar la herramienta tanto desde ordenador como desde dispositivos móviles.

## 

## Beneficios

El uso de KanFlow aportará al usuario y al equipo de trabajo diversas ventajas:

* Mayor productividad: La organización de tareas en tableros visuales permite identificar rápidamente el estado de cada actividad y detectar cuellos de botella.
* Colaboración mejorada: Al compartir tableros en tiempo real, los miembros del equipo pueden coordinarse de forma más eficiente y sincronizar su trabajo.
* Automatización de procesos: Con el editor de bloques, los usuarios pueden crear flujos automáticos (por ejemplo, notificaciones al mover una tarjeta a "Terminado"), ahorrando tiempo y reduciendo errores humanos.

# Análisis

En el siguiente apartado se mostrará el análisis realizado por el desarrollo del proyecto.

## Requisitos funcionales

Explicar todas las funciones de la aplicación

*RF01: El usuario podrá registrarse e iniciar sesión.*

*RF02: El usuario podrá crear nuevos tableros de trabajo.*

*RF03: Cada tablero podrá contener múltiples columnas (listas).*

*RF04: Se podrán añadir tarjetas (tareas) dentro de cada columna.*

*RF05: El usuario podrá arrastrar y soltar (drag and drop) las tarjetas entre columnas.*

*RF06: Las tarjetas podrán tener título, descripción y subtareas.*

*RF07: Se podrán editar o eliminar tarjetas.*

*RF08: El usuario podrá cerrar sesión y gestionar su sesión activa.*

*RF09: Se podrá personalizar el orden de las columnas.*

*RF10: El sistema guardará automáticamente los cambios realizados.*

*RF11: Se podrán crear flujos de automatización y asignarlos a las tareas*

## Requisitos no funcionales

Los requisitos no funcionales son aquellos que imponen restricciones bajo las que debe funcionar el sistema.

En nuestro sistema tenemos los siguientes requisitos no funcionales:

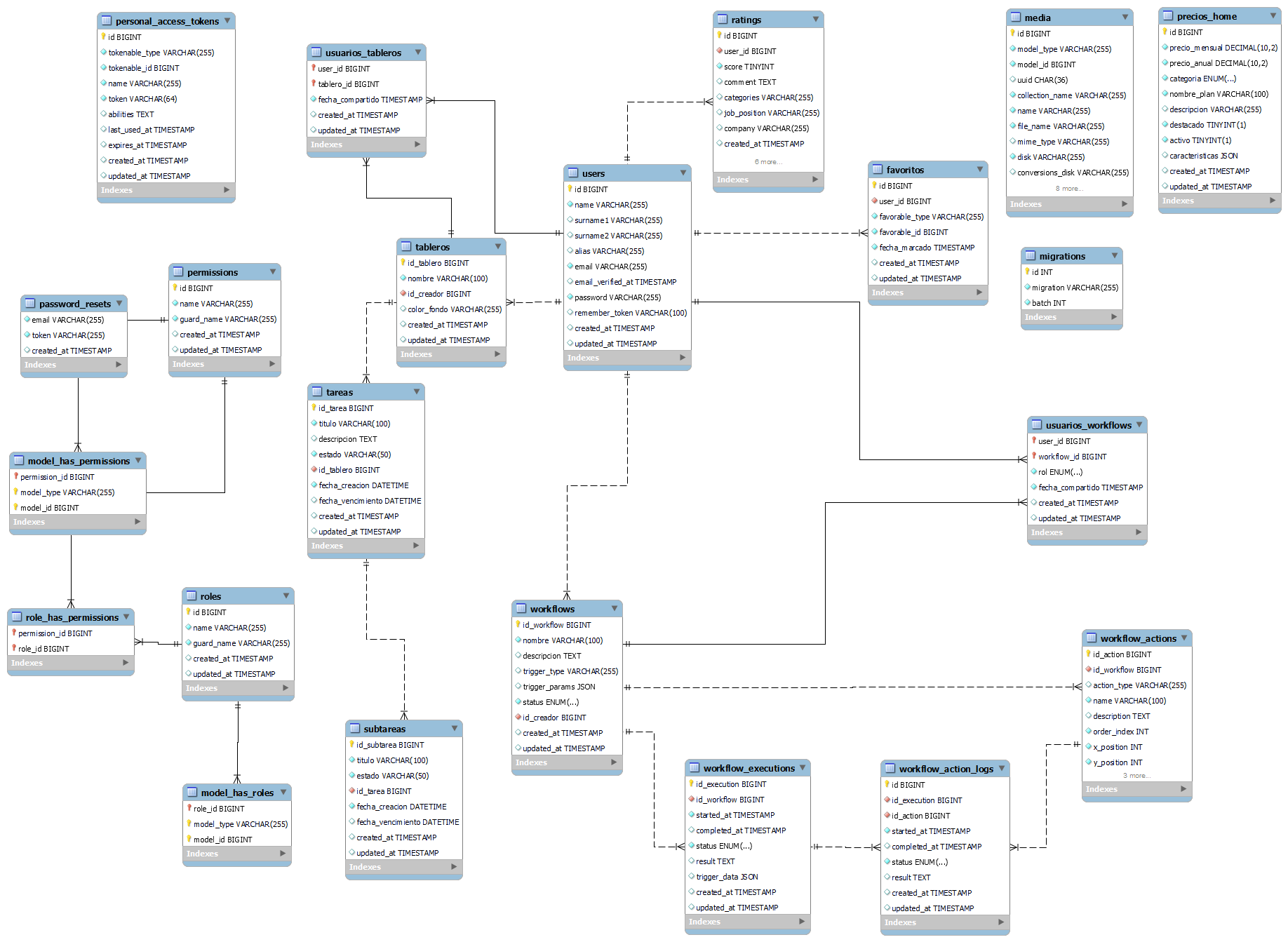
* *RNF01: Fluidez: La aplicación debe responder de forma rápida, independientemente de la cantidad de tareas o columnas, incluso si depende de la conexión a Internet.*
* *RNF02: Usabilidad: La aplicación debe ser intuitiva para que pueda ser utilizada por cualquier tipo de usuario.*
* *RNF03: Compatibilidad: La aplicación debe funcionar correctamente en diferentes navegadores y dispositivos.*
* *RNF04: Escalabilidad: El código debe estar preparado para permitir futuras ampliaciones y mejoras.*
* *RNF05: Seguridad: Los datos del usuario, especialmente la información de inicio de sesión, deben estar protegidos adecuadamente.*

# Requisitos de datos

## Diagrama de classes

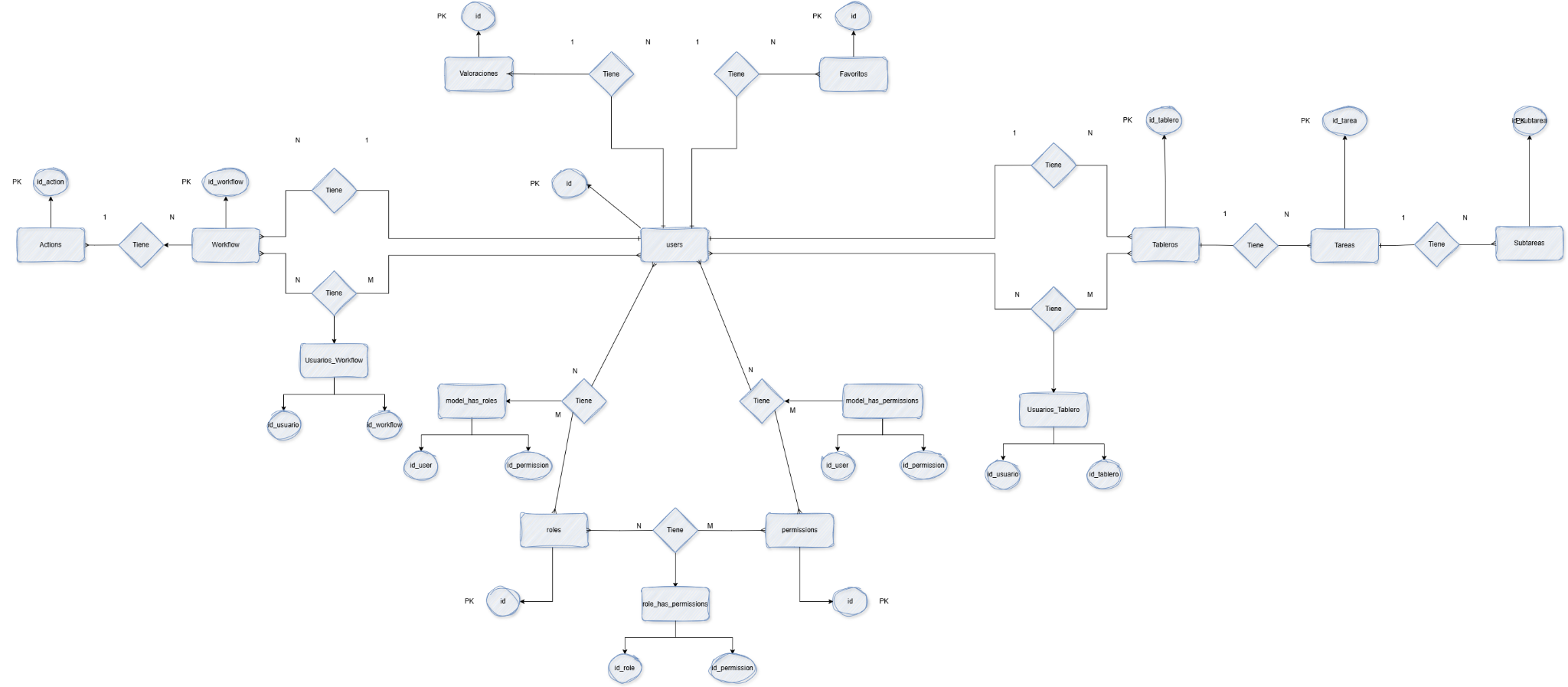
Añadiremos el diagrama de clases, las clases que posteriormente crearemos, sus atributos y métodos que creemos que serán necesarios. Enlace al diagrama de clases:

<https://drive.google.com/file/d/1G5oqCyw0RGvujvCaKvFu4kdKSakZkuLy/view?usp=sharing>



## Diseño de base de datos

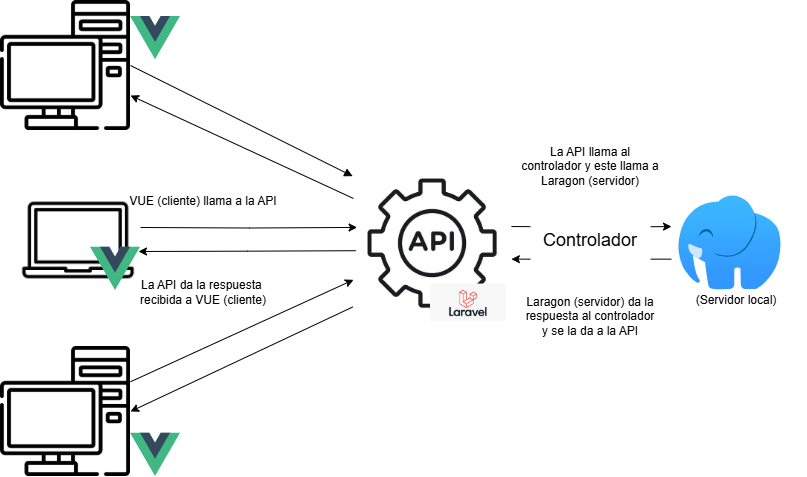
Esquema del modelo entidad-relacion enlace al modelo entidad-relacion: <https://drive.google.com/file/d/11-_W8BAfOEet72RSt8VLdsEF4tBdNC0r/view?usp=sharing>



## 

## Diseño del sistema

Esquema del model client-servidor amb les tecnologies. Enlace al modelo cliente-servidor: <https://drive.google.com/file/d/11-_W8BAfOEet72RSt8VLdsEF4tBdNC0r/view?usp=sharing>



# 

# Diseño

## Investigación de referencias y definición de tu propia idea

Estas son las tres ideas que hemos tenido:

**1-** Gestor de Proyectos

**2-** Sistema de Tareas (To-Do List)

**3-** CRM

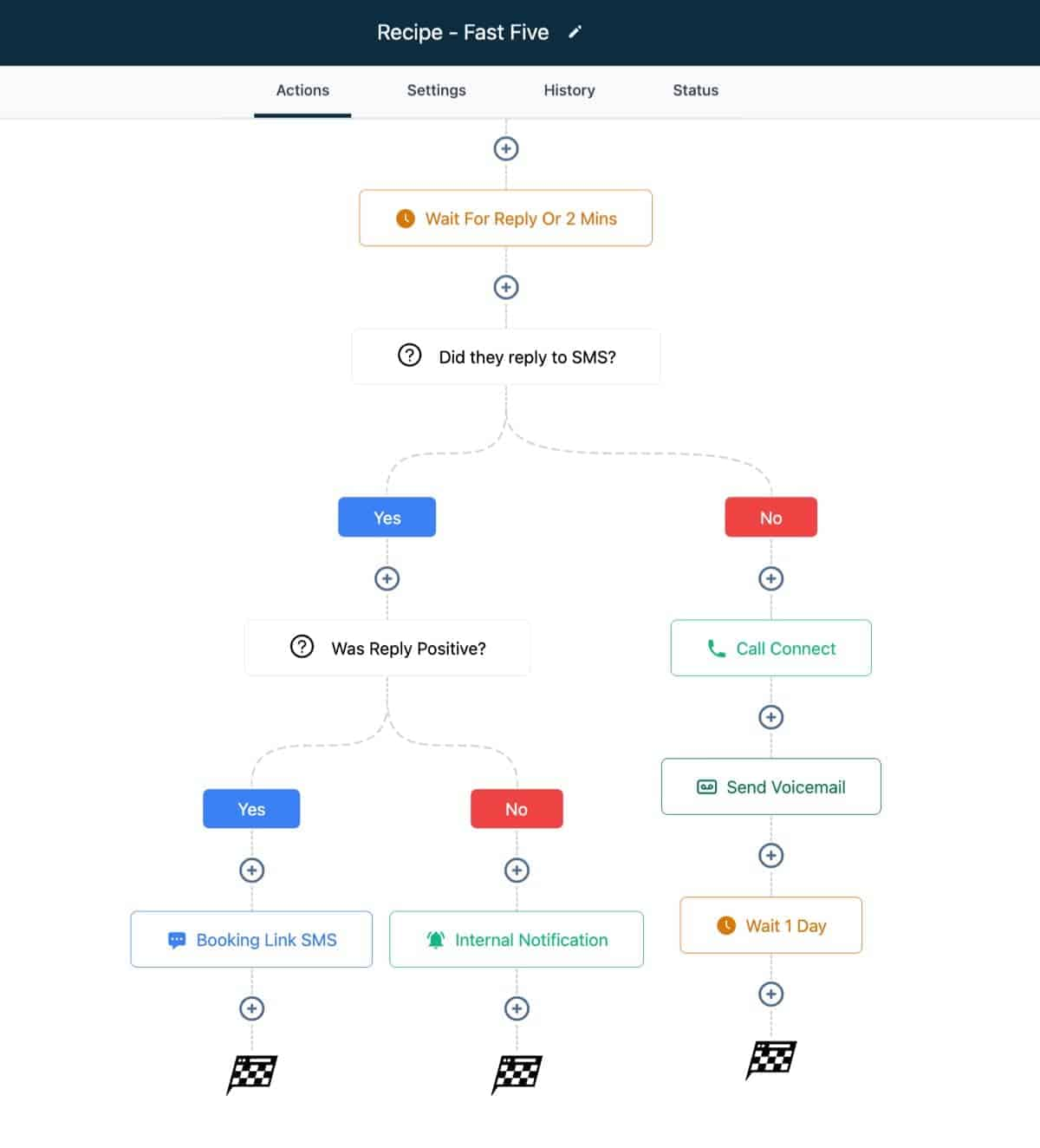
Por eso vamos a desarrollar un proyecto basado en un sistema Kanban, similar a herramientas como Trello o Monday.com, pero con una particularidad: incorporaremos una pequeña funcionalidad propia de un CRM. Este proyecto no será un CRM completo, pero integrará una fusión sencilla que combine la gestión visual de tareas del Kanban con una característica clave típica de un CRM.

Hemos identificado tres aplicaciones web: **Monday.com**, **Trello.com** y **GoHighLevel.com**.

Aunque GoHighLevel incluye una funcionalidad similar al Kanban, nos enfocaremos específicamente en su característica de Workflows (flujos de trabajo con bloques), que es la función que adoptaremos como inspiración desde el ámbito de un CRM.

Utilizaremos este formato de Workflows, añadiremos funciones básicas como automatizar el envío de correos electrónicos y la confirmación de tareas.

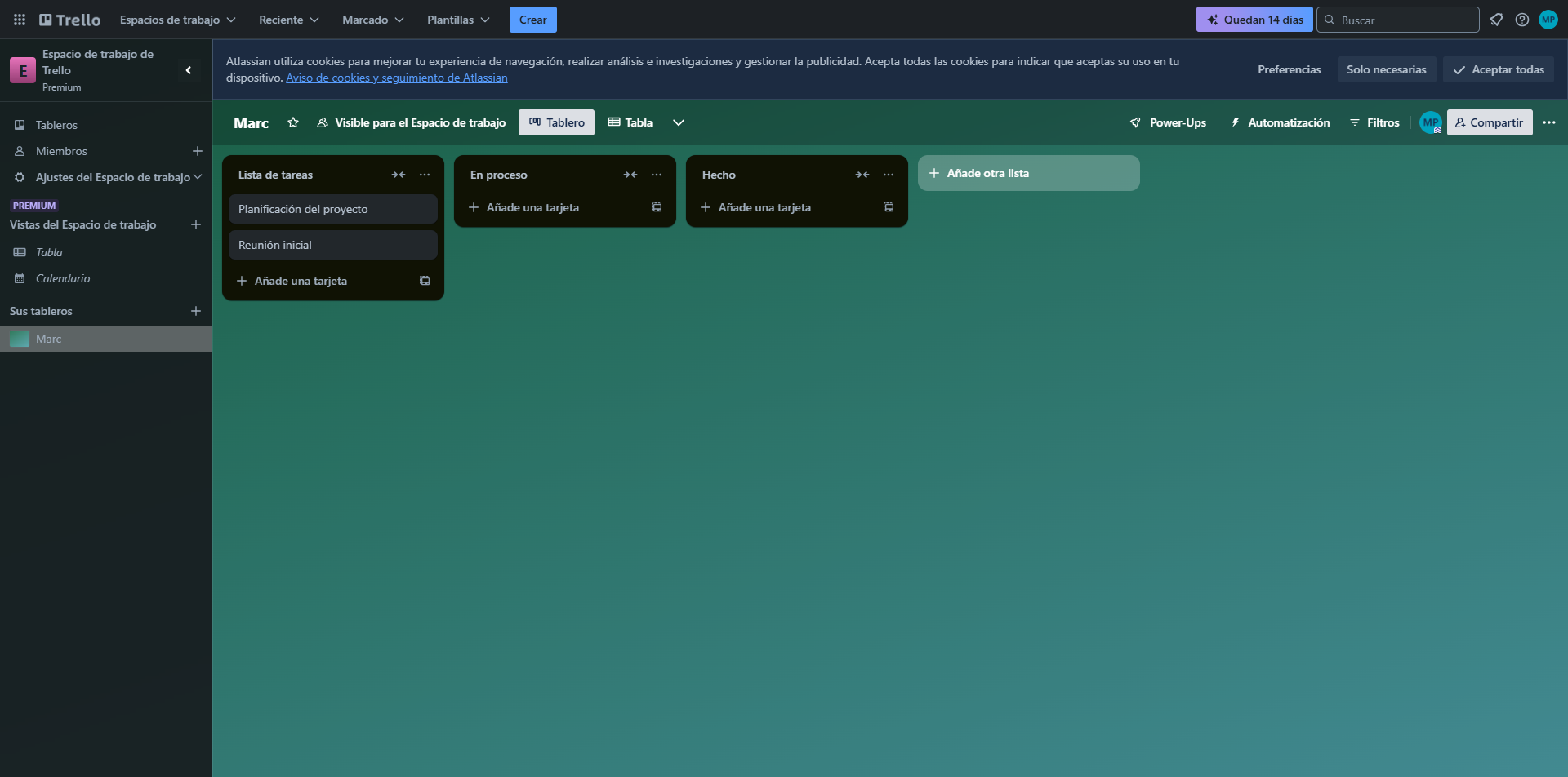
Captura de pantalla de GoHighLevel:



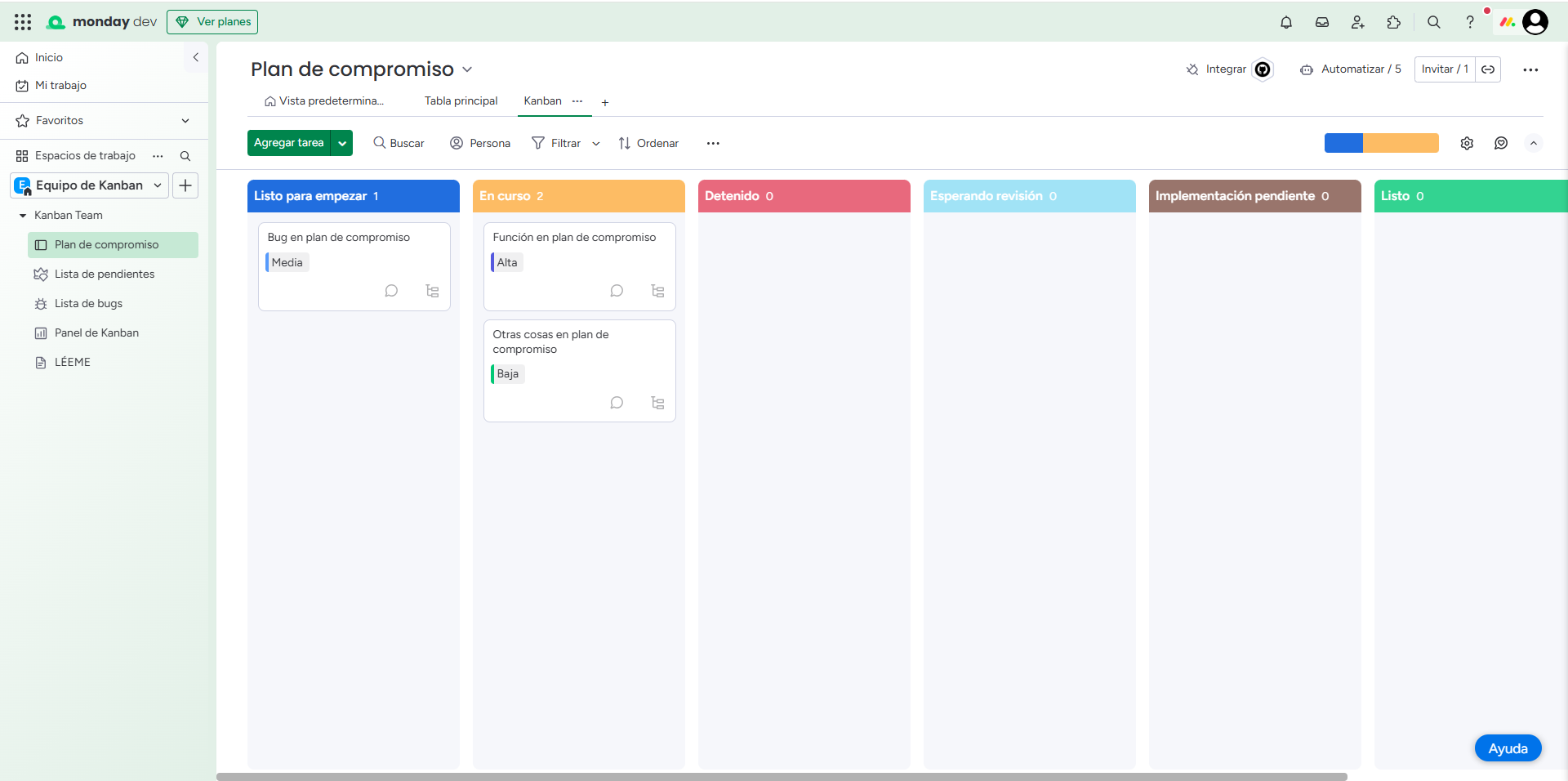
**Comparación Trello y Monday:**

En cuanto a los aspectos visuales, Trello destaca por su diseño simple y limpio, con tarjetas rectangulares que son fáciles de identificar y manejar. Por otro lado, Monday.com ofrece un diseño más colorido y moderno, con una estética más empresarial y profesional. Sin embargo, Monday.com puede presentar una ligera curva de aprendizaje al inicio, en comparación con la simplicidad de Trello, que resulta más intuitivo desde el principio.

Captura de Trello:



Captura de Monday:



**Propósito de la Aplicación:**

Desarrollar una plataforma web que combine elementos de gestión de tareas y automatización básica, dirigida a usuarios que buscan herramientas accesibles y personalizables para organizar sus proyectos y flujos de trabajo.

**Funcionalidades Principales:**

En nuestra aplicación web, tomaremos como referencia la funcionalidad de "flows" de GoHighLevel, que permite la creación de automatizaciones. Nuestro objetivo es desarrollar un tablero interactivo que permita a los usuarios arrastrar bloques para configurar acciones, similar al conocido programa Power Automate.

Dado que replicar todas las capacidades de un sistema tan avanzado sería demasiado complejo, optamos por simplificar su funcionalidad. Implementaremos acciones básicas que permitan al usuario crear sus propios "flows" o automatizaciones de forma sencilla e intuitiva.

Por ejemplo, una de las acciones que planeamos incluir será la automatización vinculada al tablero Kanban. Cuando un usuario arrastre una tarjeta a la columna de "Terminado", la aplicación podrá desencadenar una notificación automática indicando que la tarea ha sido completada. Es importante destacar que esta automatización deberá haber sido previamente configurada por el propio usuario.

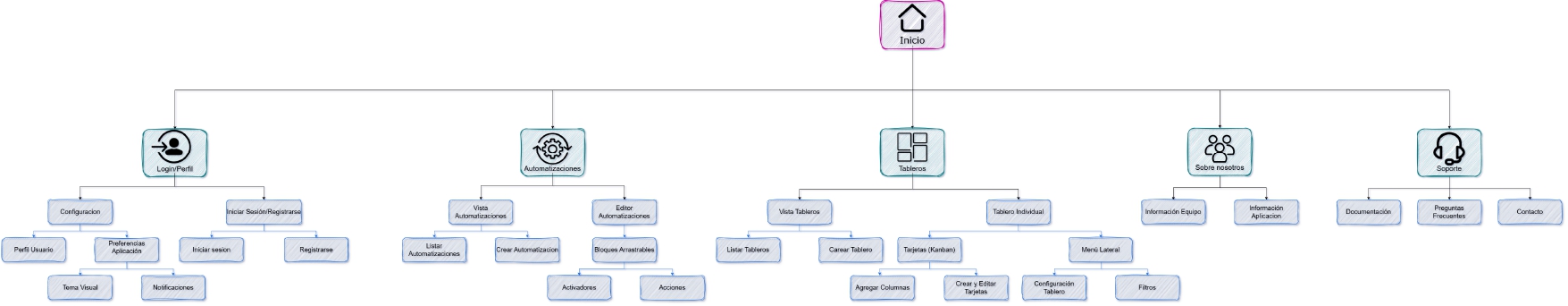
En resumen, proporcionaremos esta función en nuestra aplicación web que permita a los usuarios crear automatizaciones simples, adaptadas a sus necesidades, mediante un sistema de bloques arrastrables y acciones predeterminadas.

En las otras aplicaciones web en las que nos basaremos, como Trello y Monday, implementaremos funcionalidades específicas de cada una. De Trello, adoptaremos la funcionalidad del menú de la barra lateral y la configuración de las tarjetas, que ofrecen opciones claras y personalizables. Por otro lado, de Monday tomaremos como referencia su diseño principal, concretamente el tablero con tarjetas, caracterizado por su apariencia profesional y organizada. Estas características serán adaptadas y combinadas para optimizar la experiencia de usuario en nuestro proyecto.

## 

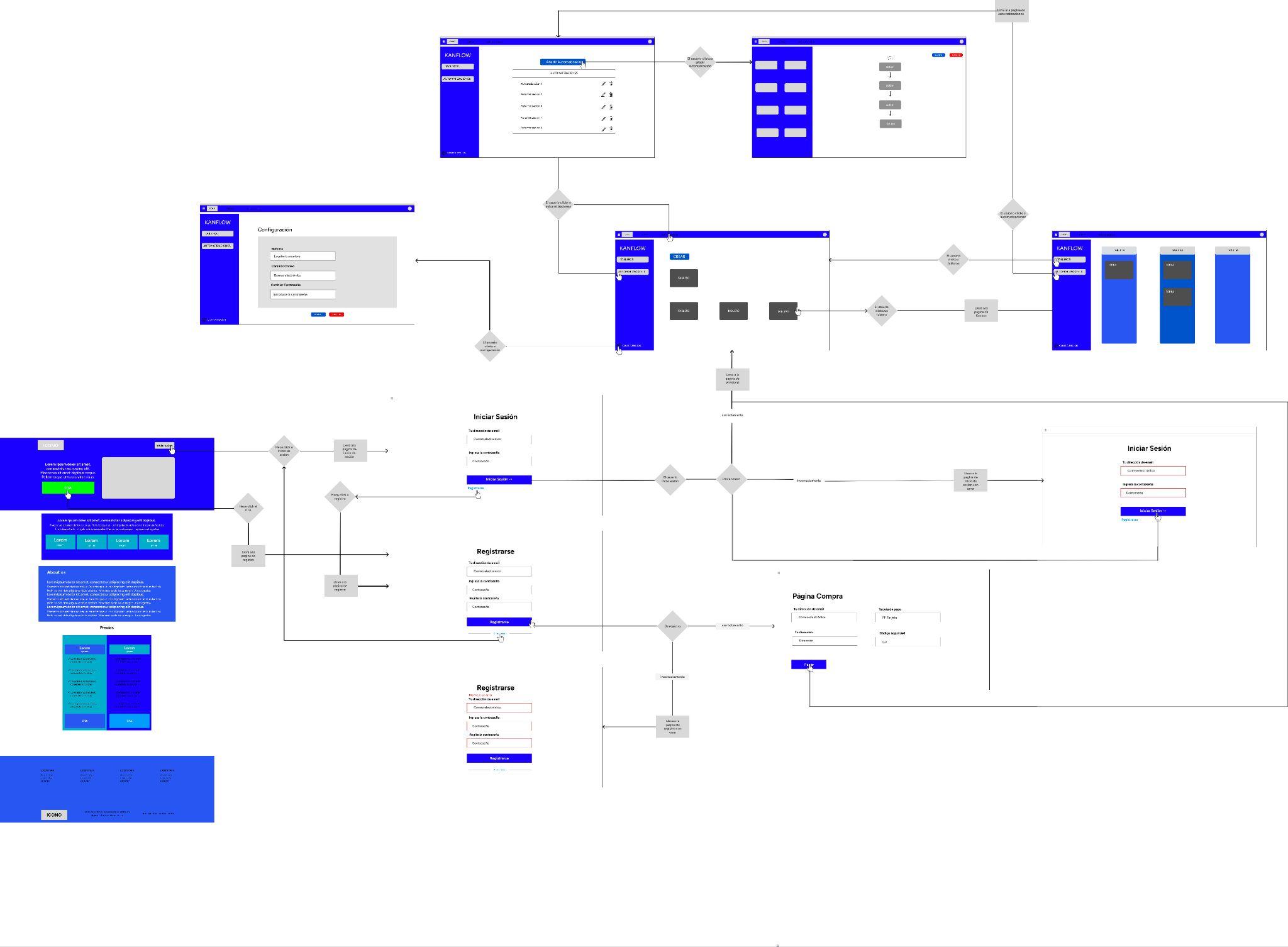
## Segunda etapa: creación del sitemap

Enlace al sitemap: <https://drive.google.com/file/d/1J5IH1s-2JXKFEslhxagnvG08-O8nnJag/view?usp=sharing>



## Wireflows

Enlace a los wireflows: <https://www.figma.com/design/KAaNQIdyO6BSst5PcOUhrB/Untitled?node-id=0-1&t=1jzZhjR3A1hPCj5S-1>



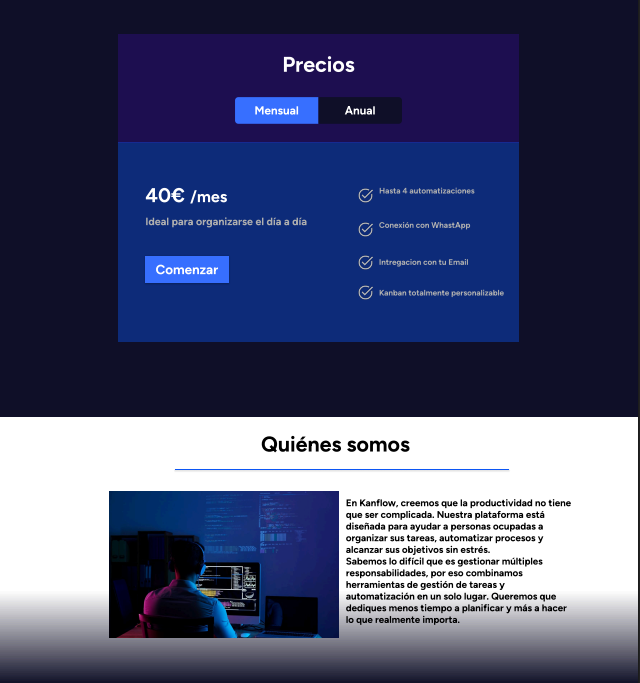
## 

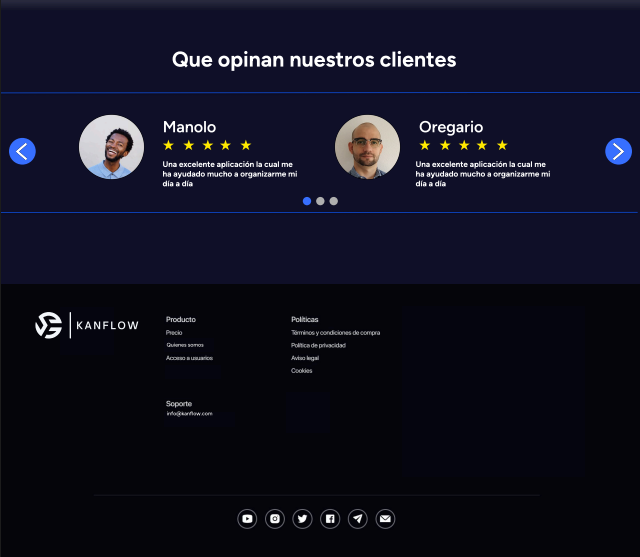
## Cuarta etapa: Prototipo de alta fidelidad

Enlace al prototipo de alta fidelidad: <https://www.figma.com/design/Ay3iKw57JwuC4o2RpGCqlW/Prototipo-de-alta-fidelidad?node-id=0-1&t=4VdEfgNJInE85e9k-1>

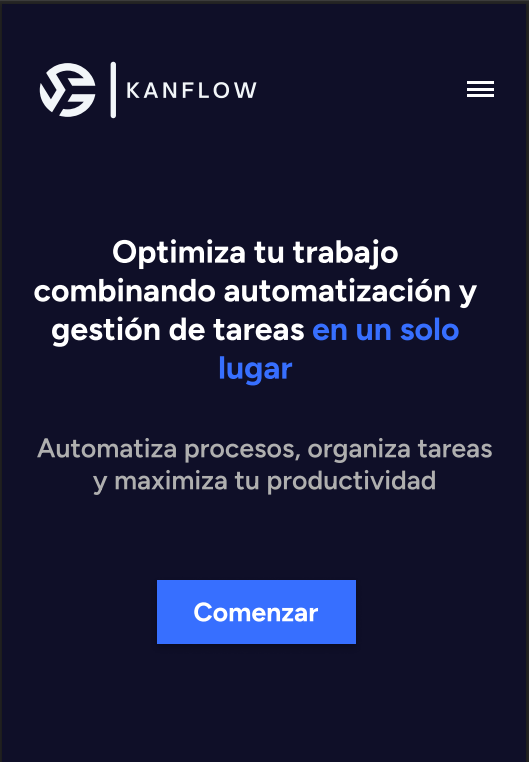
Capturas de la versión de ordenador:

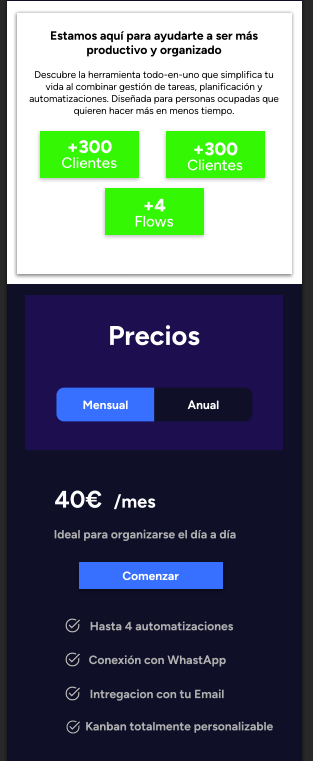






Capturas de la versión responsive



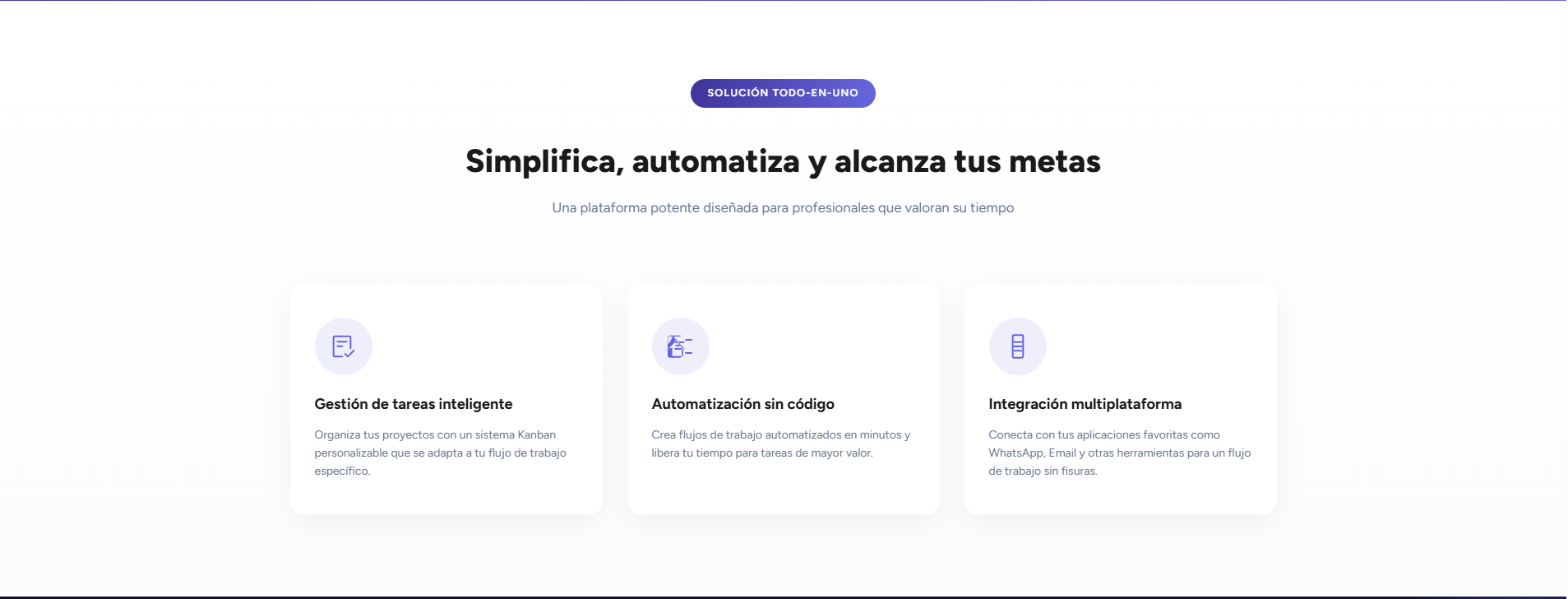




# Resultado Final

Resultado final de la web con capturas.

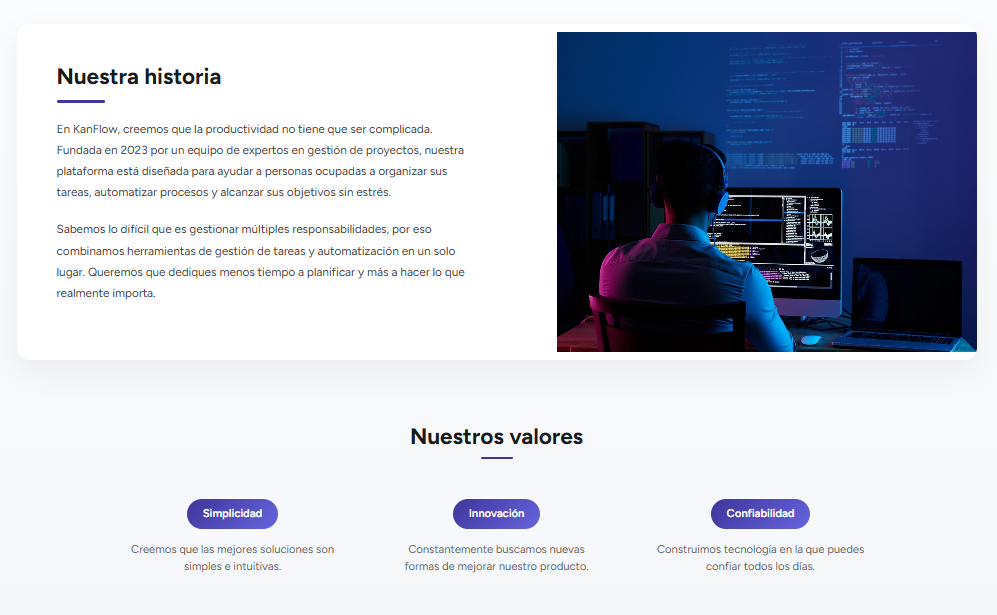






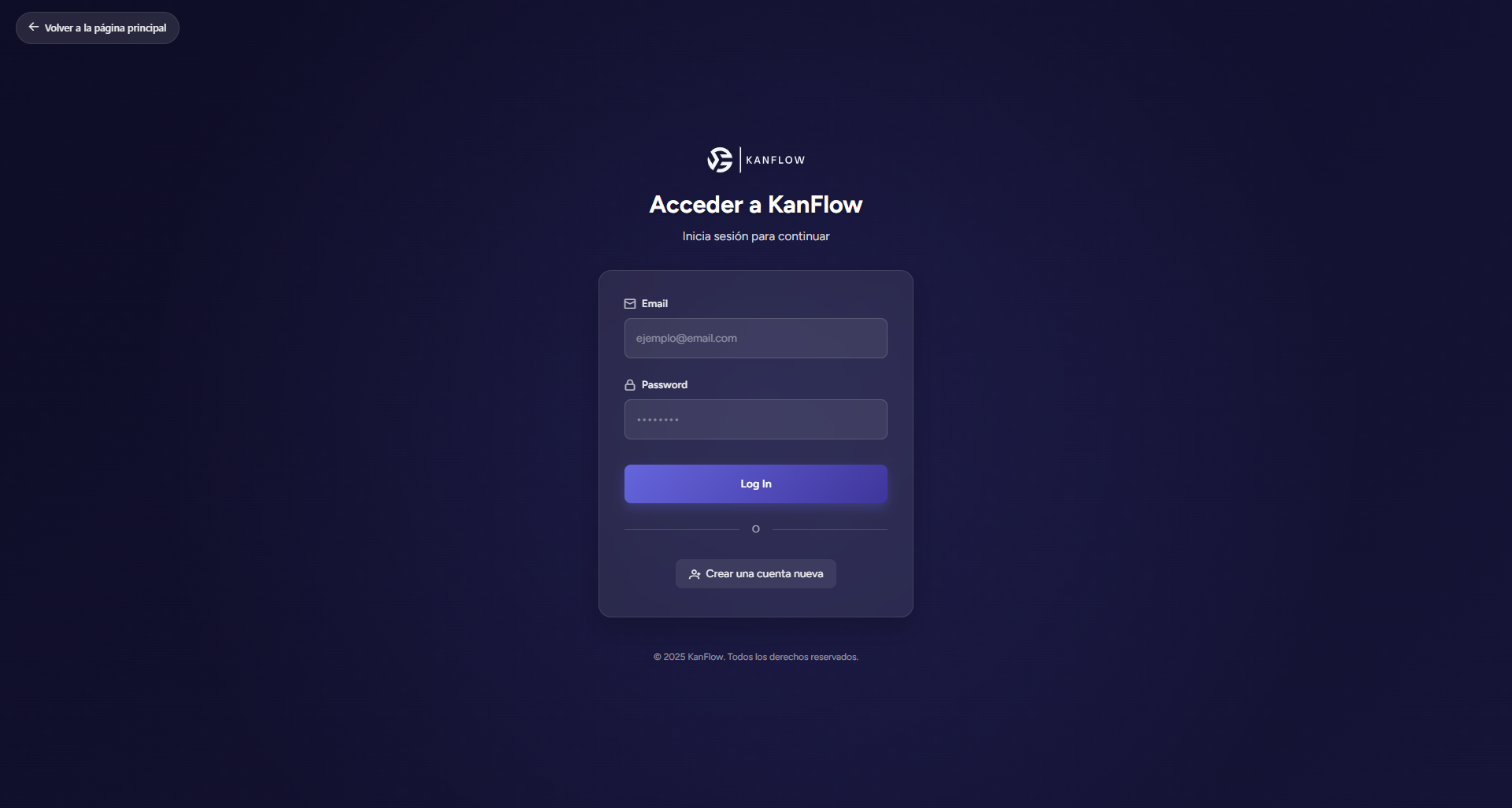


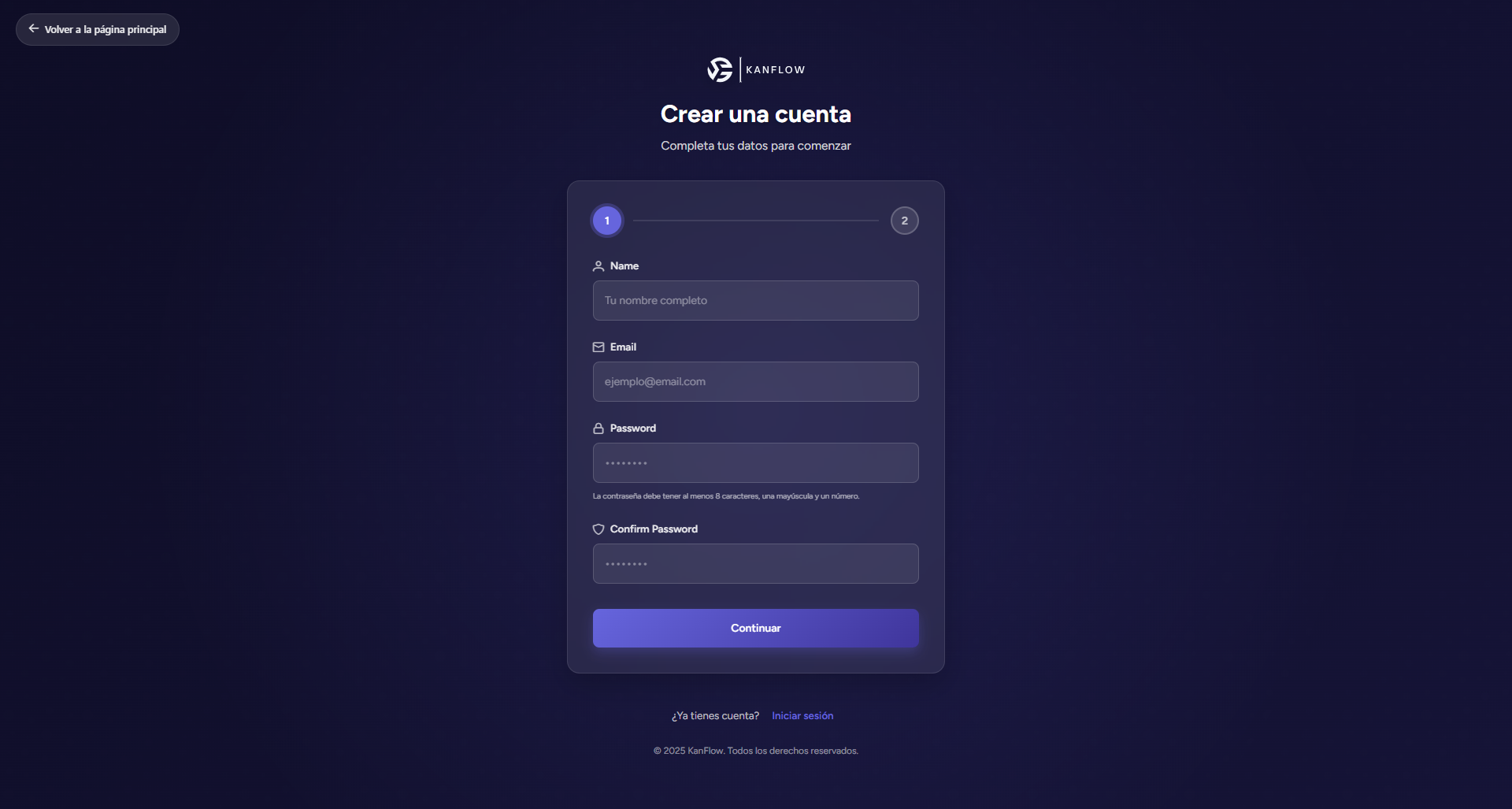


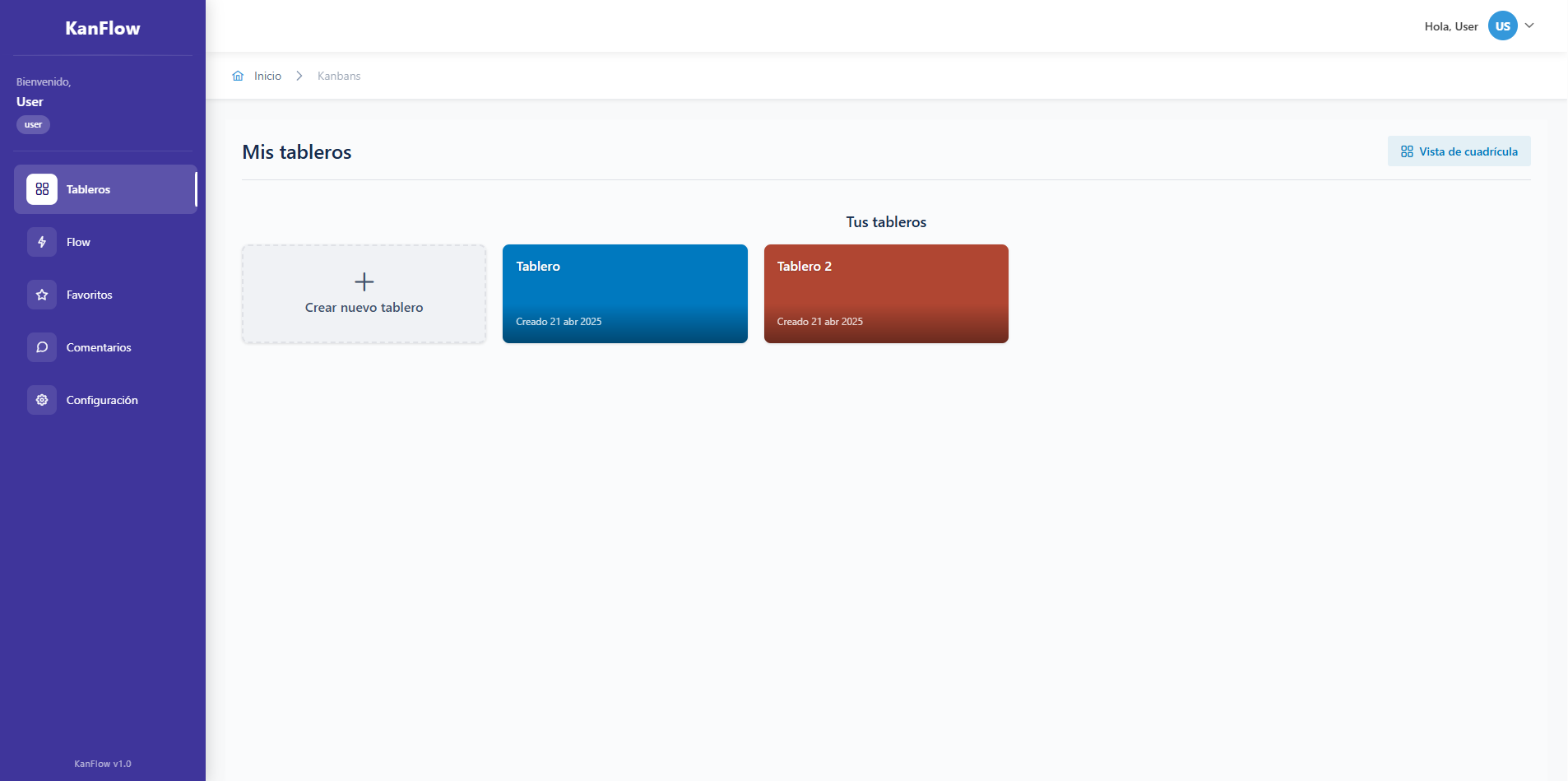


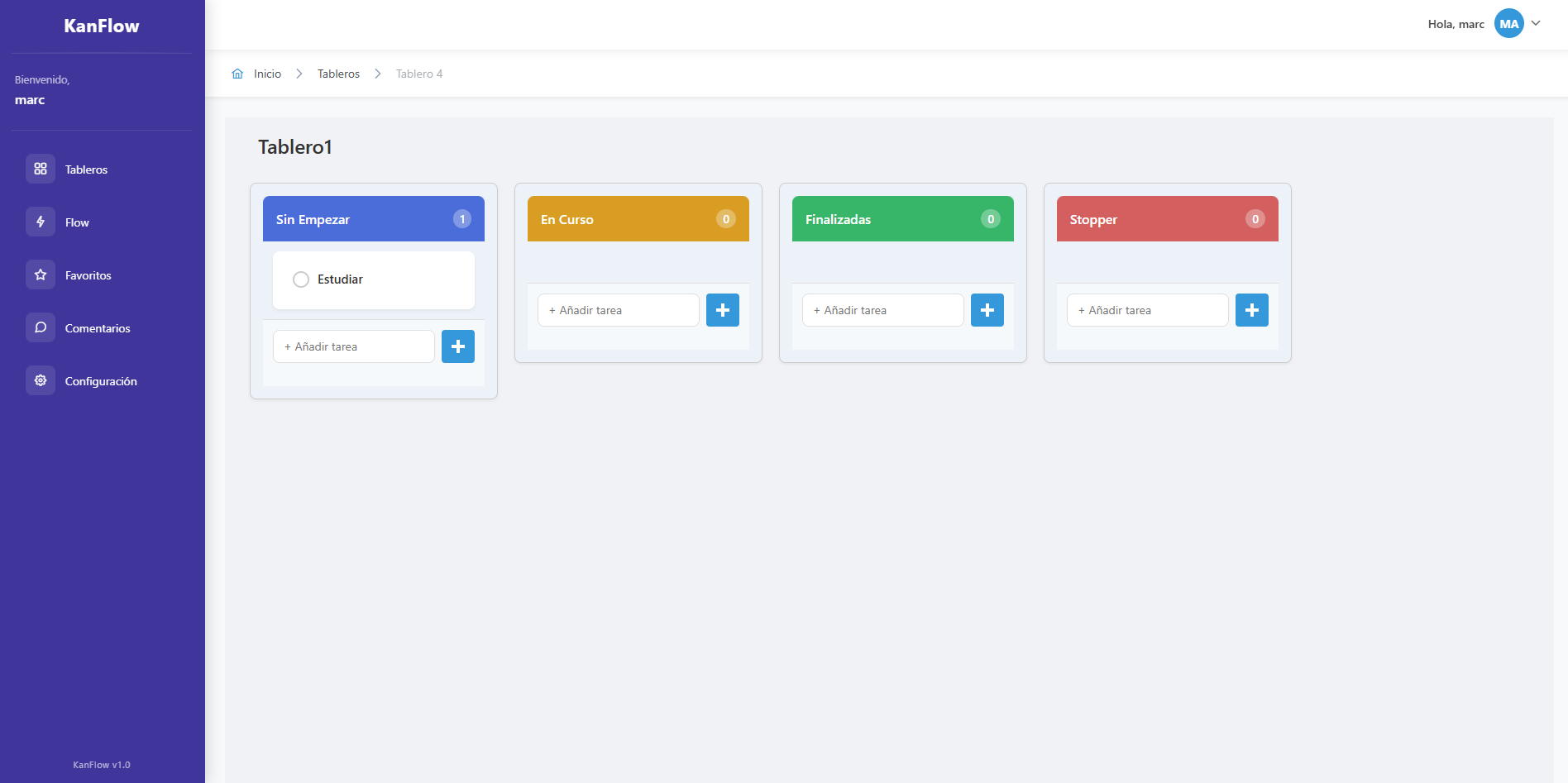


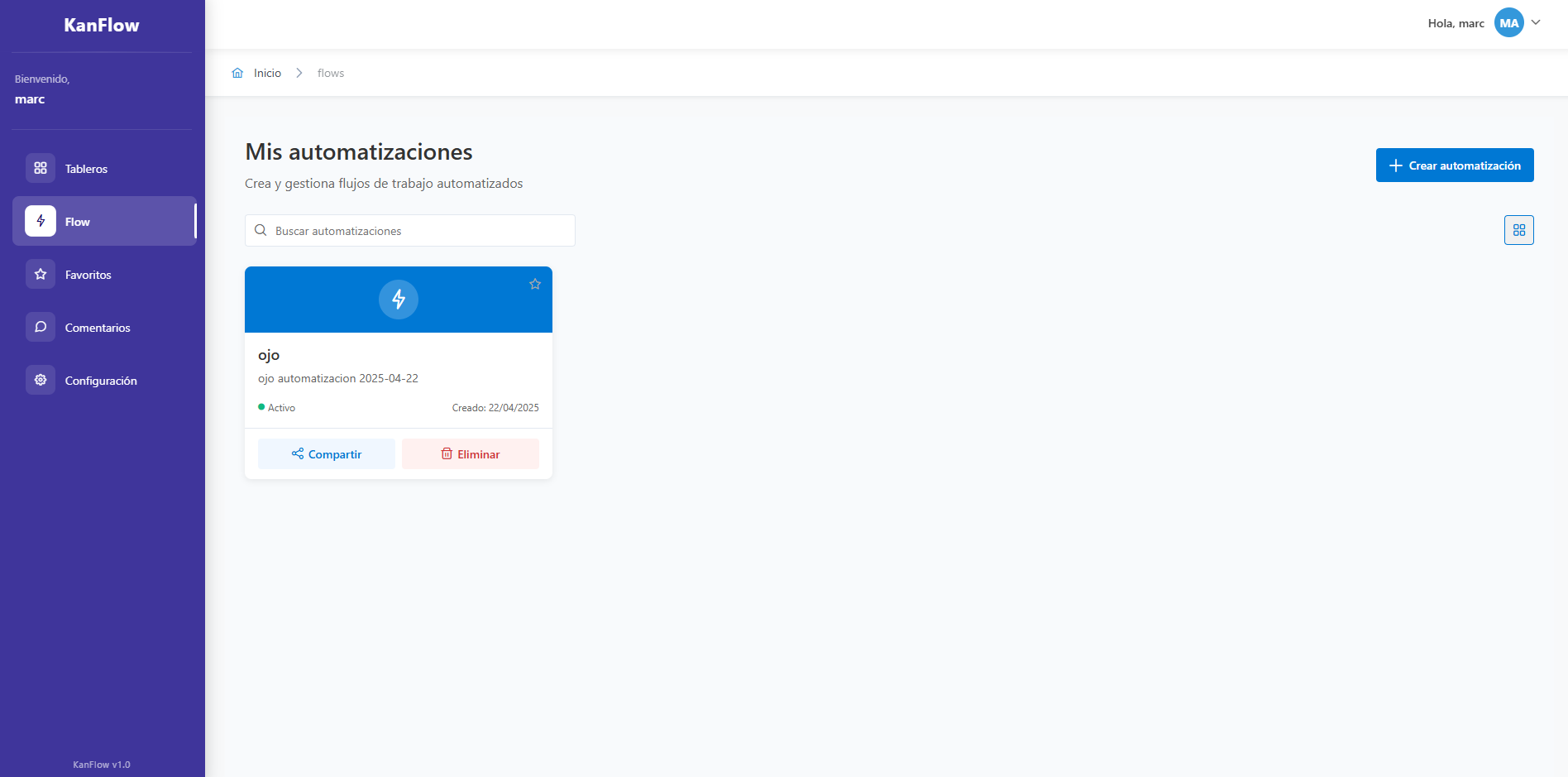


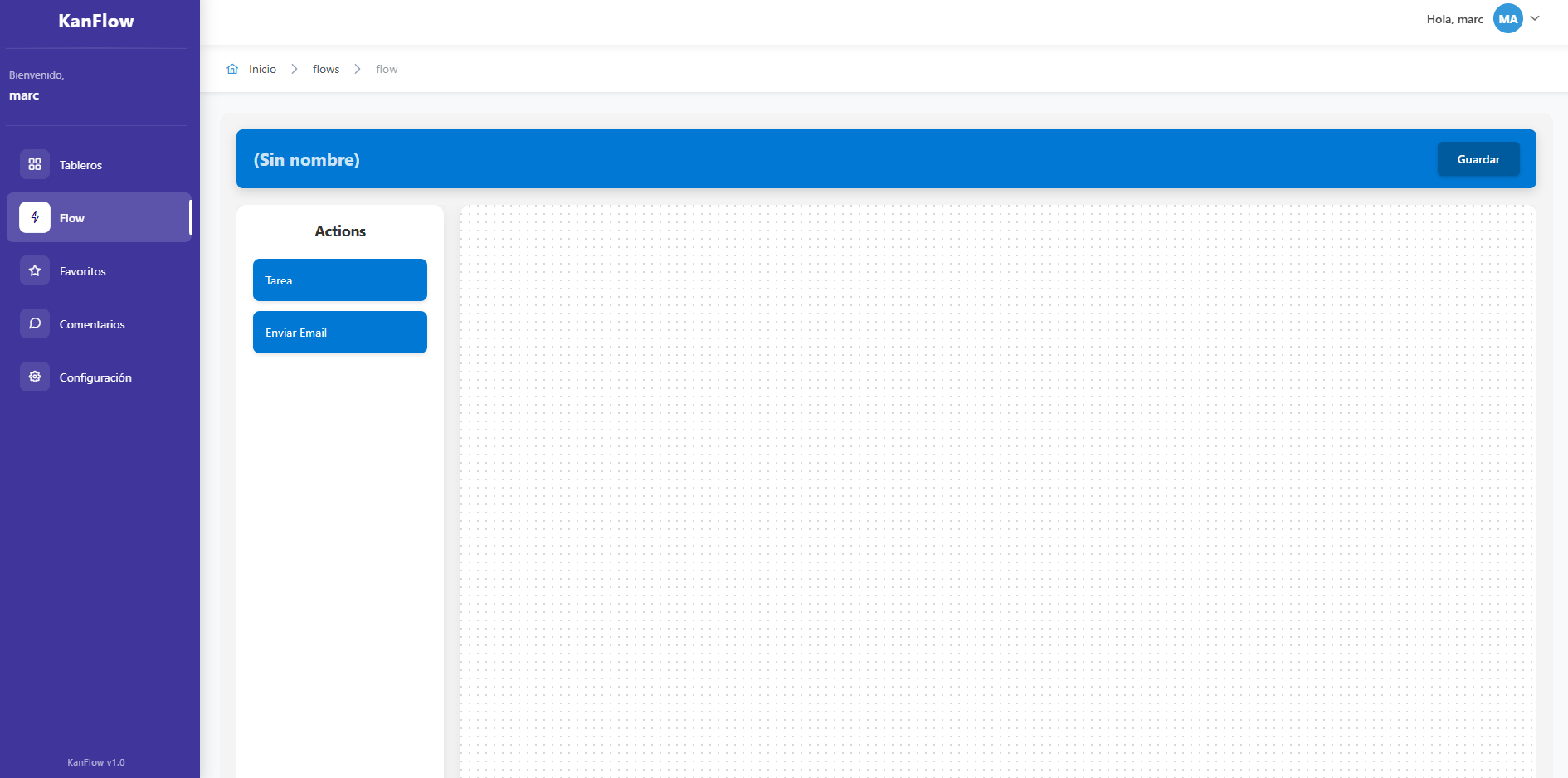


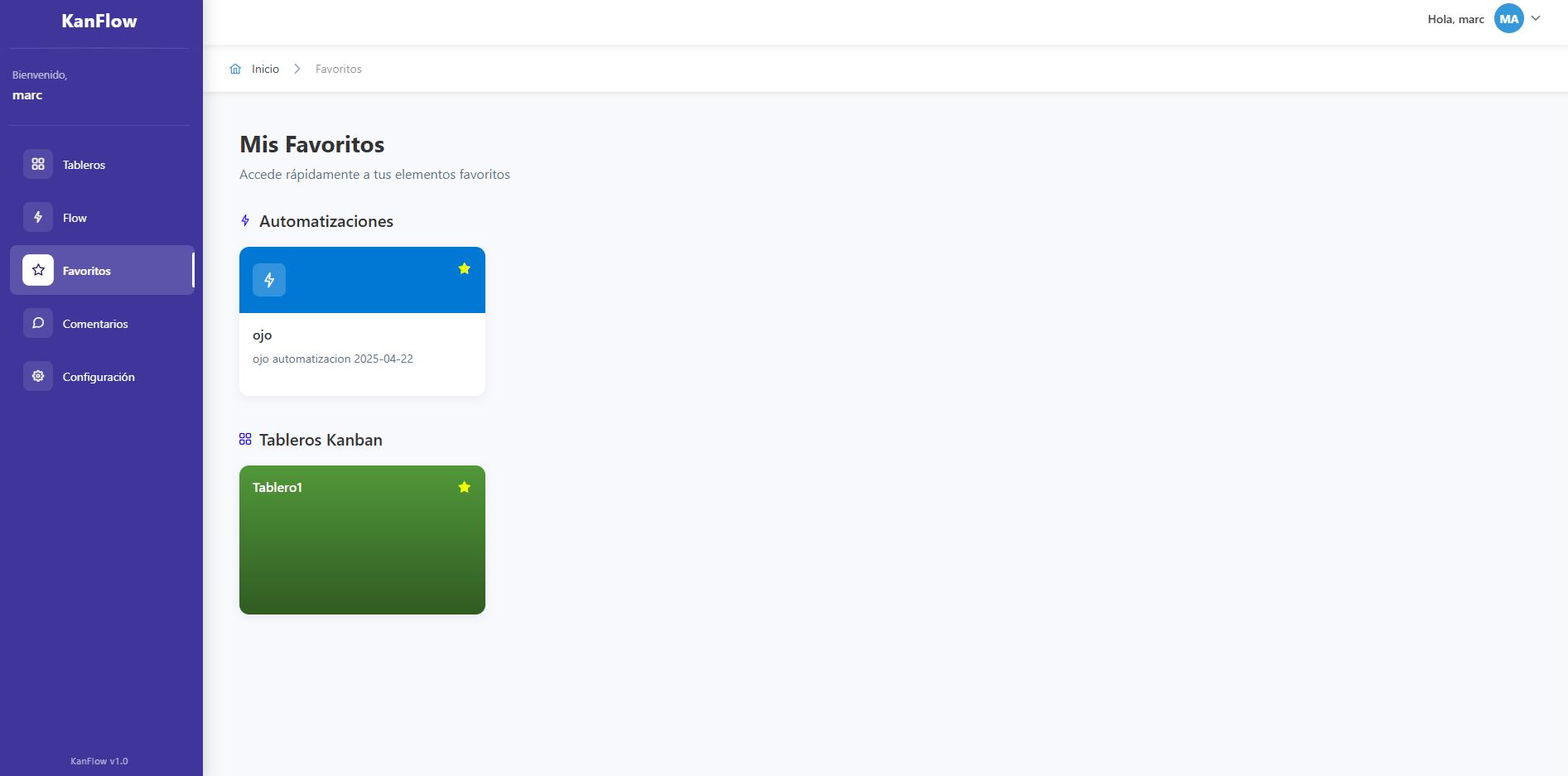


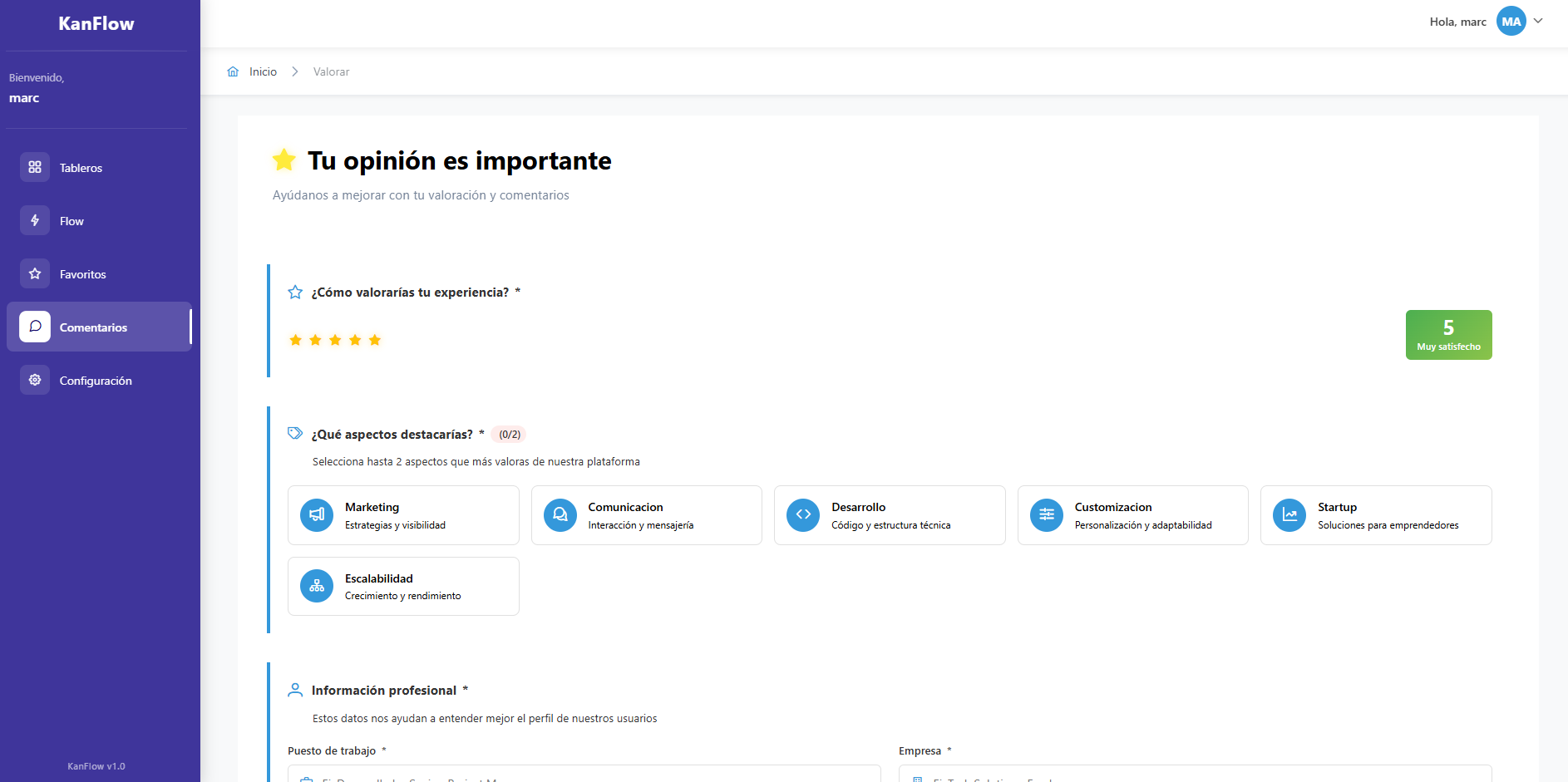


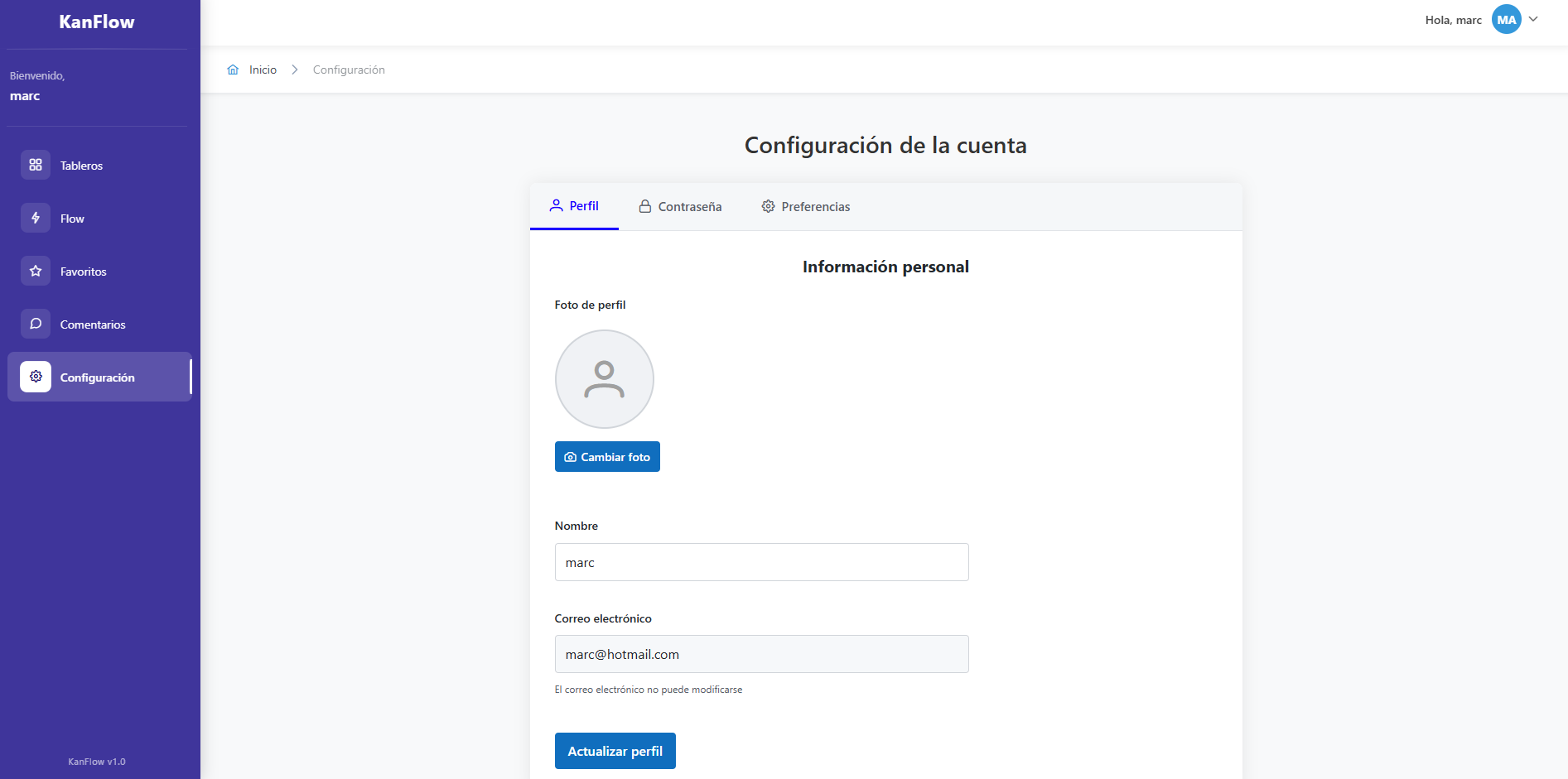


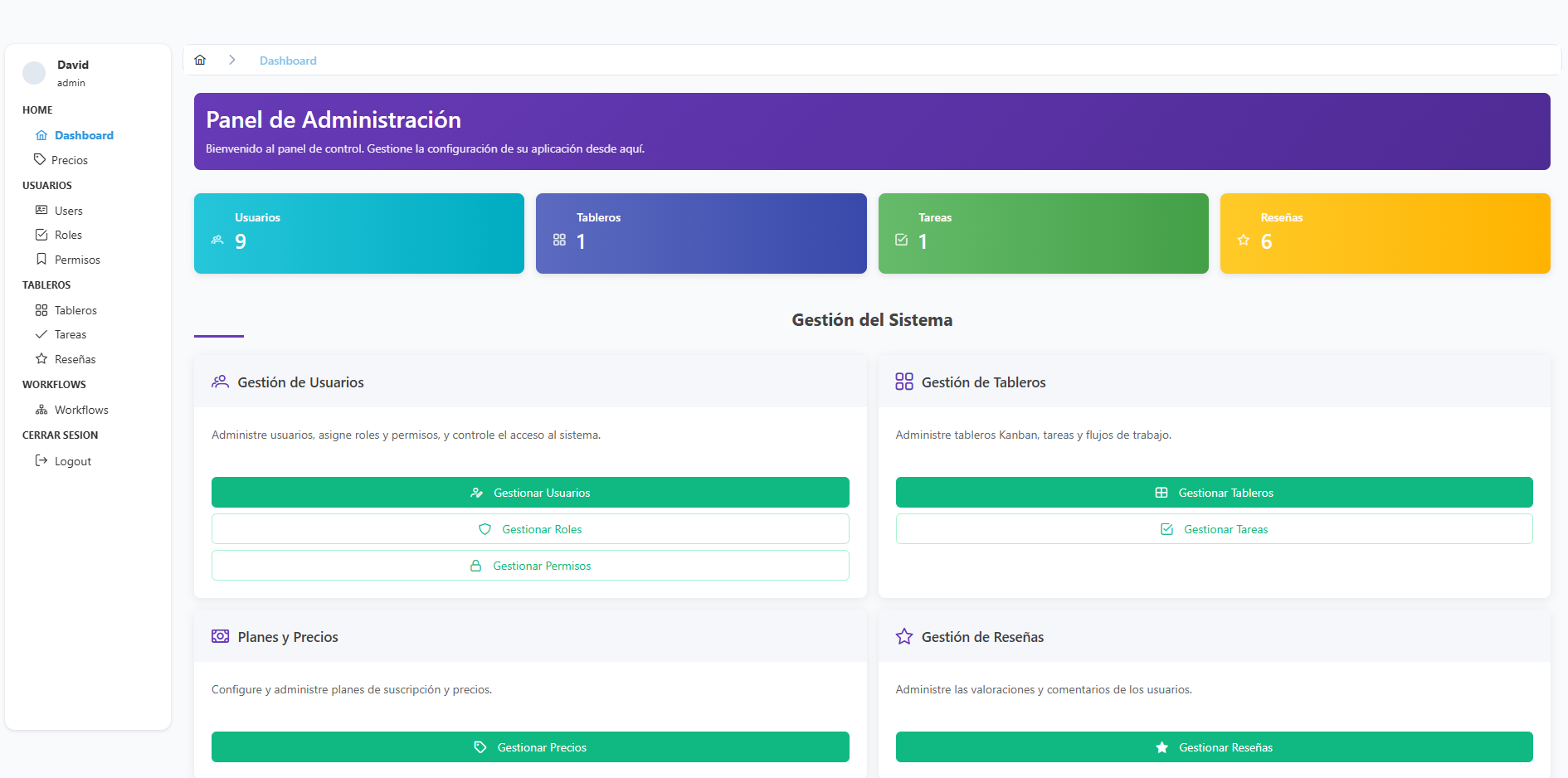


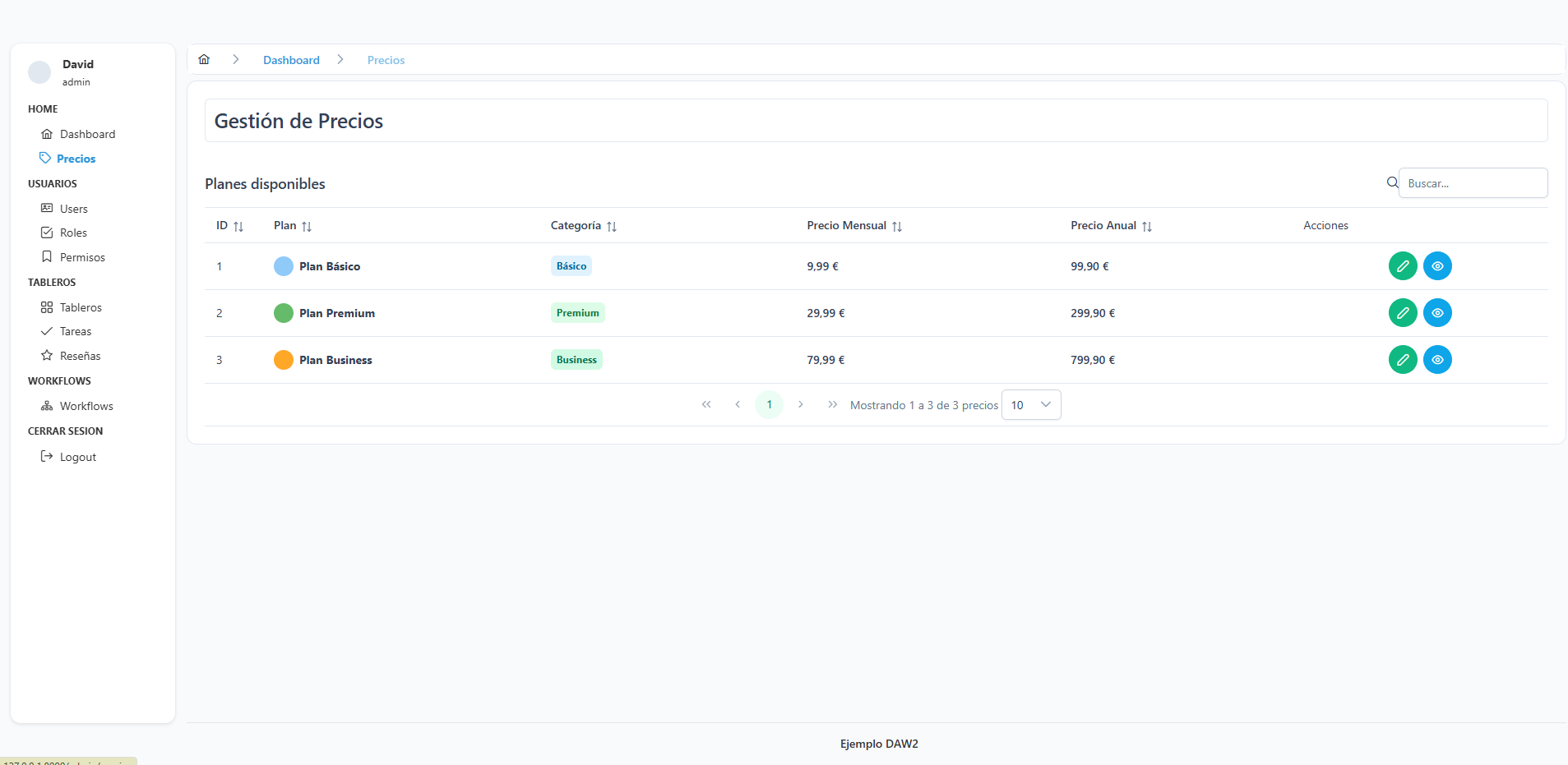


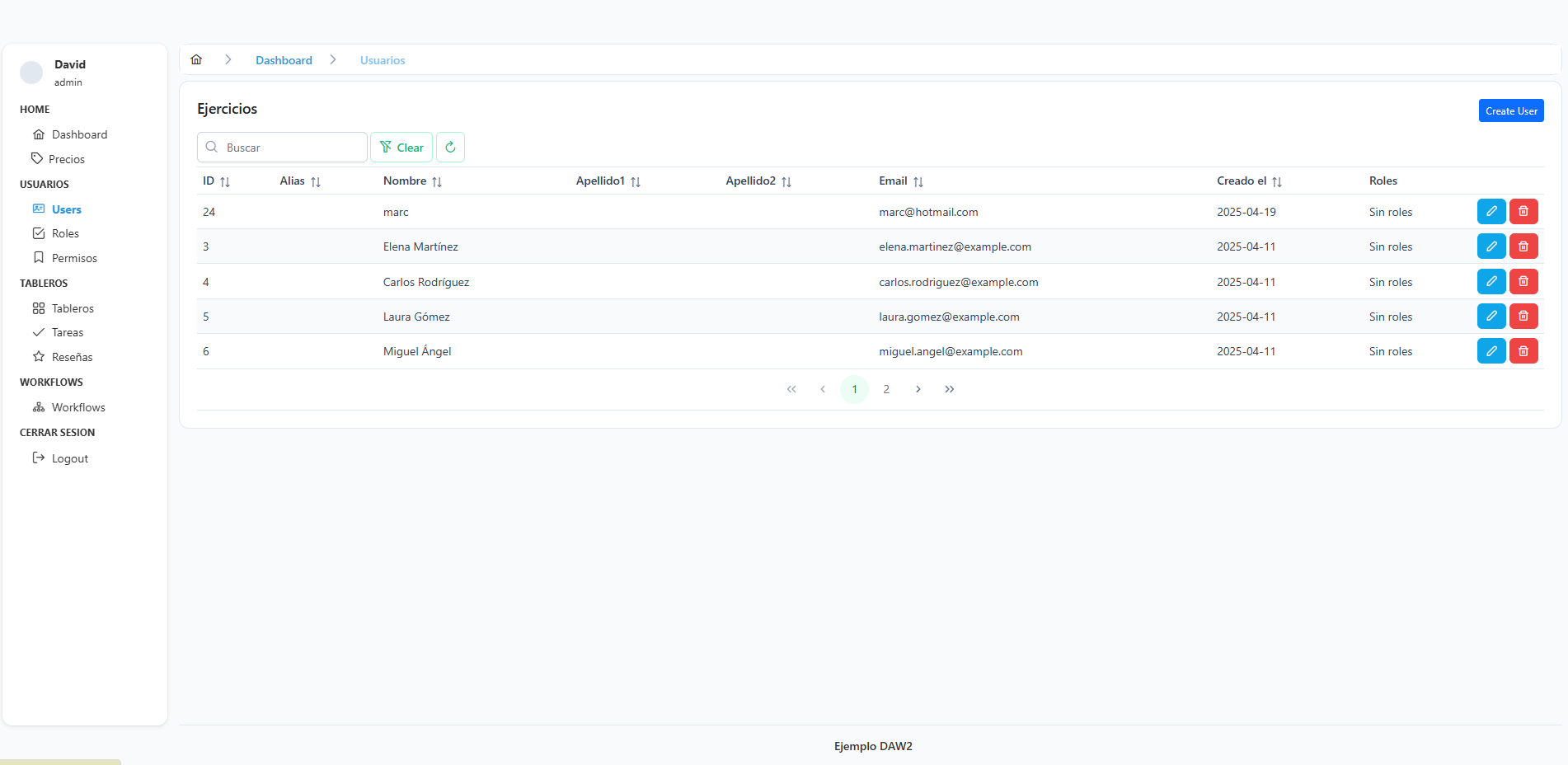


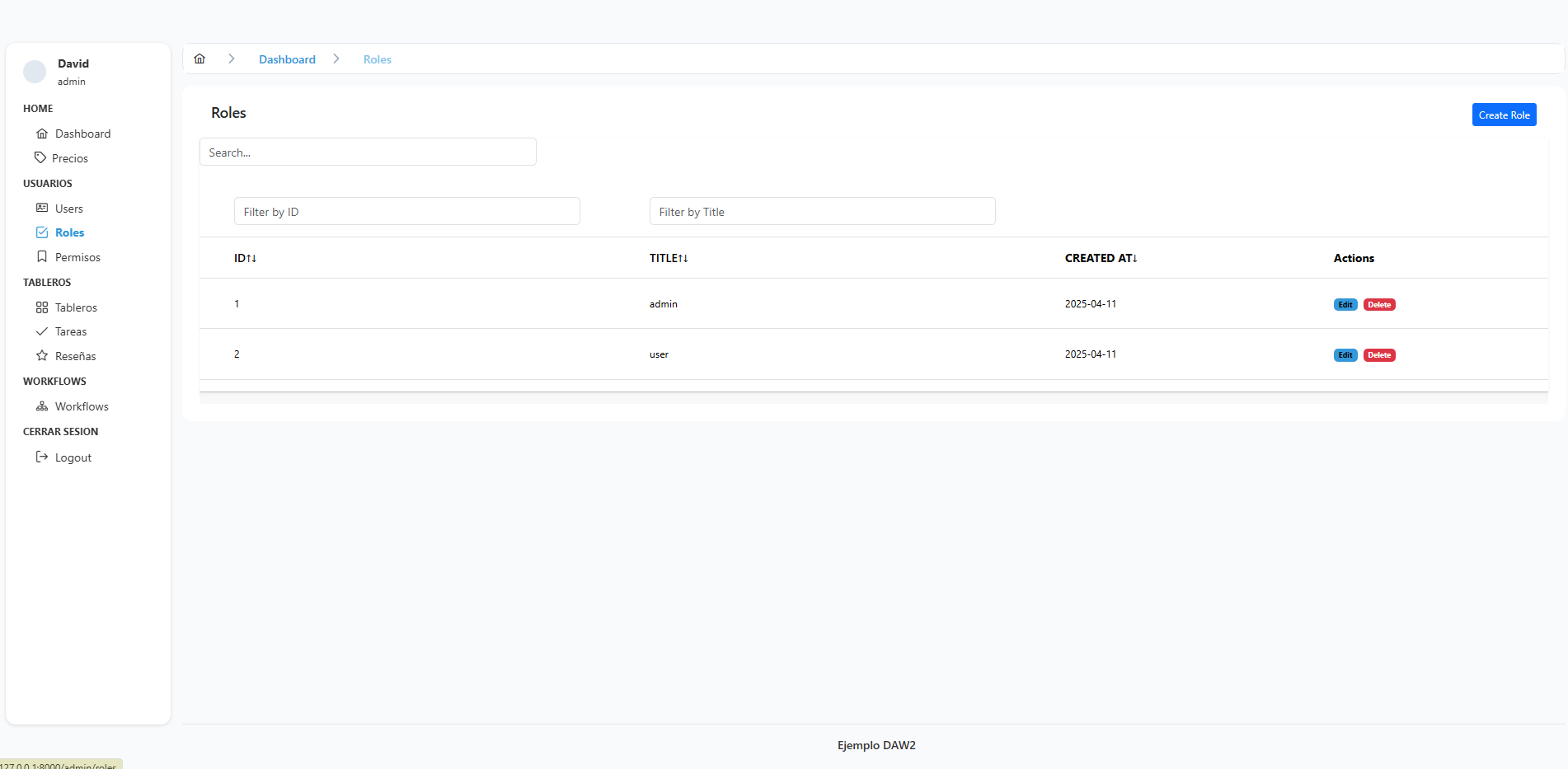


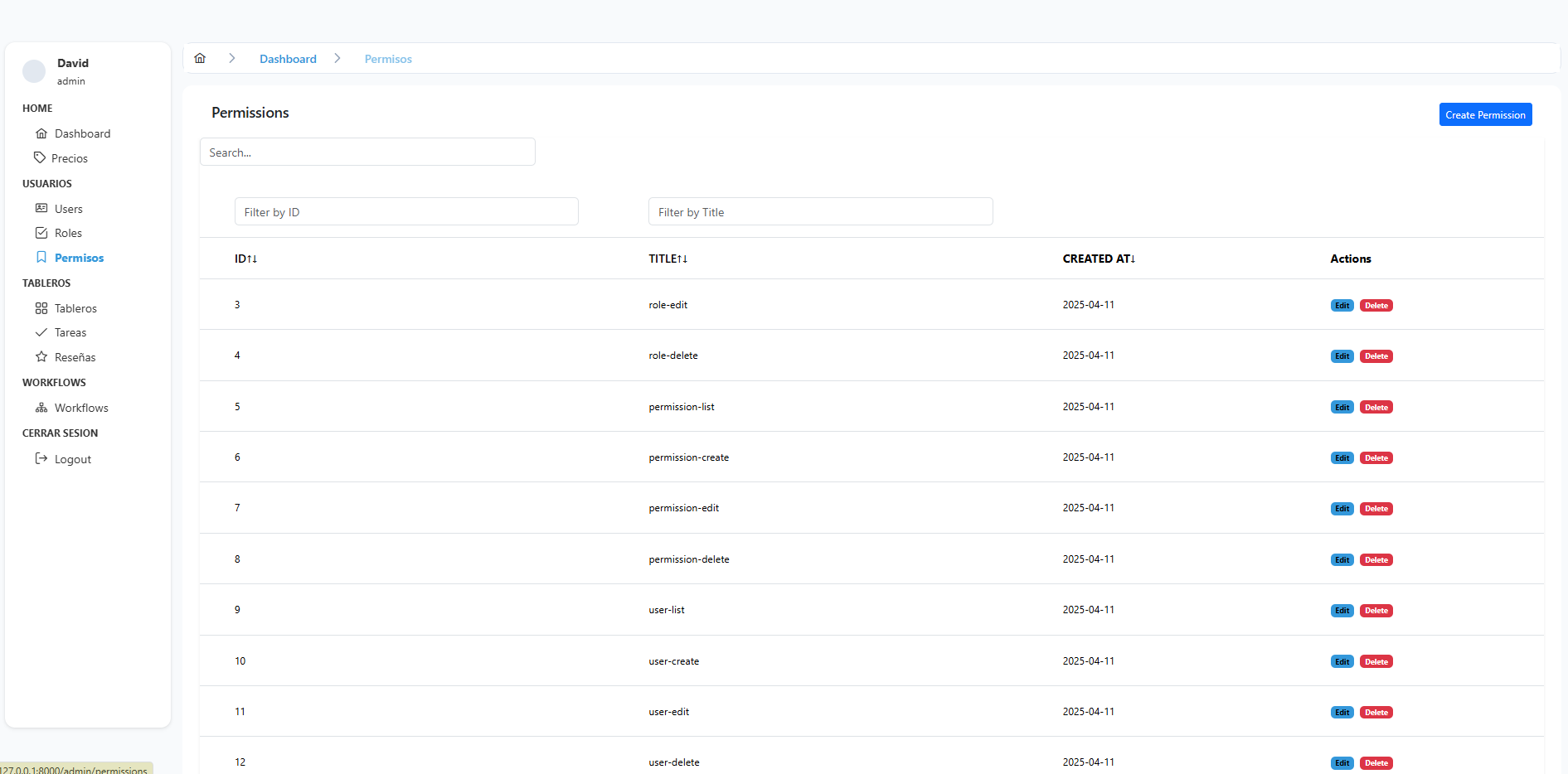


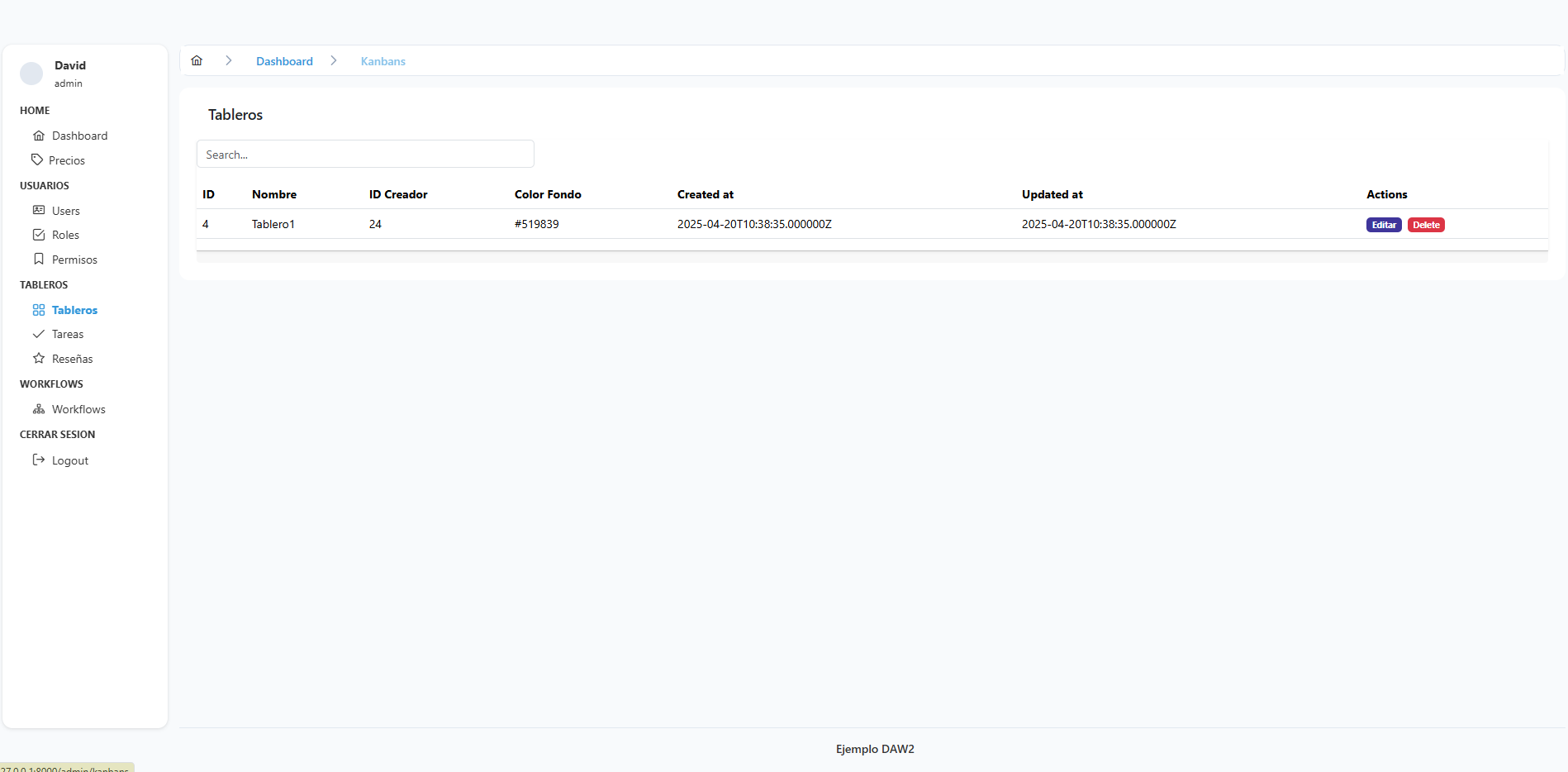


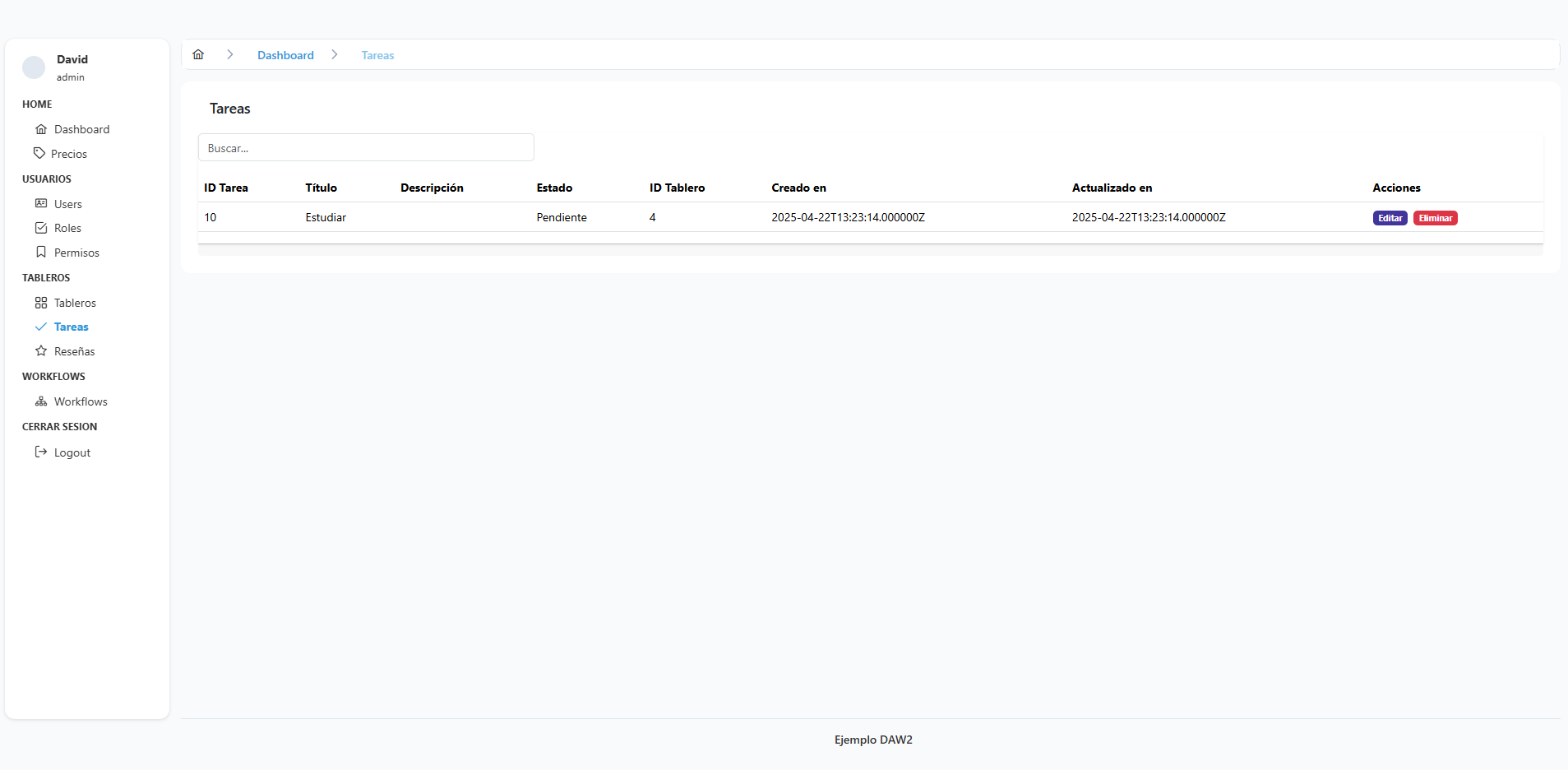


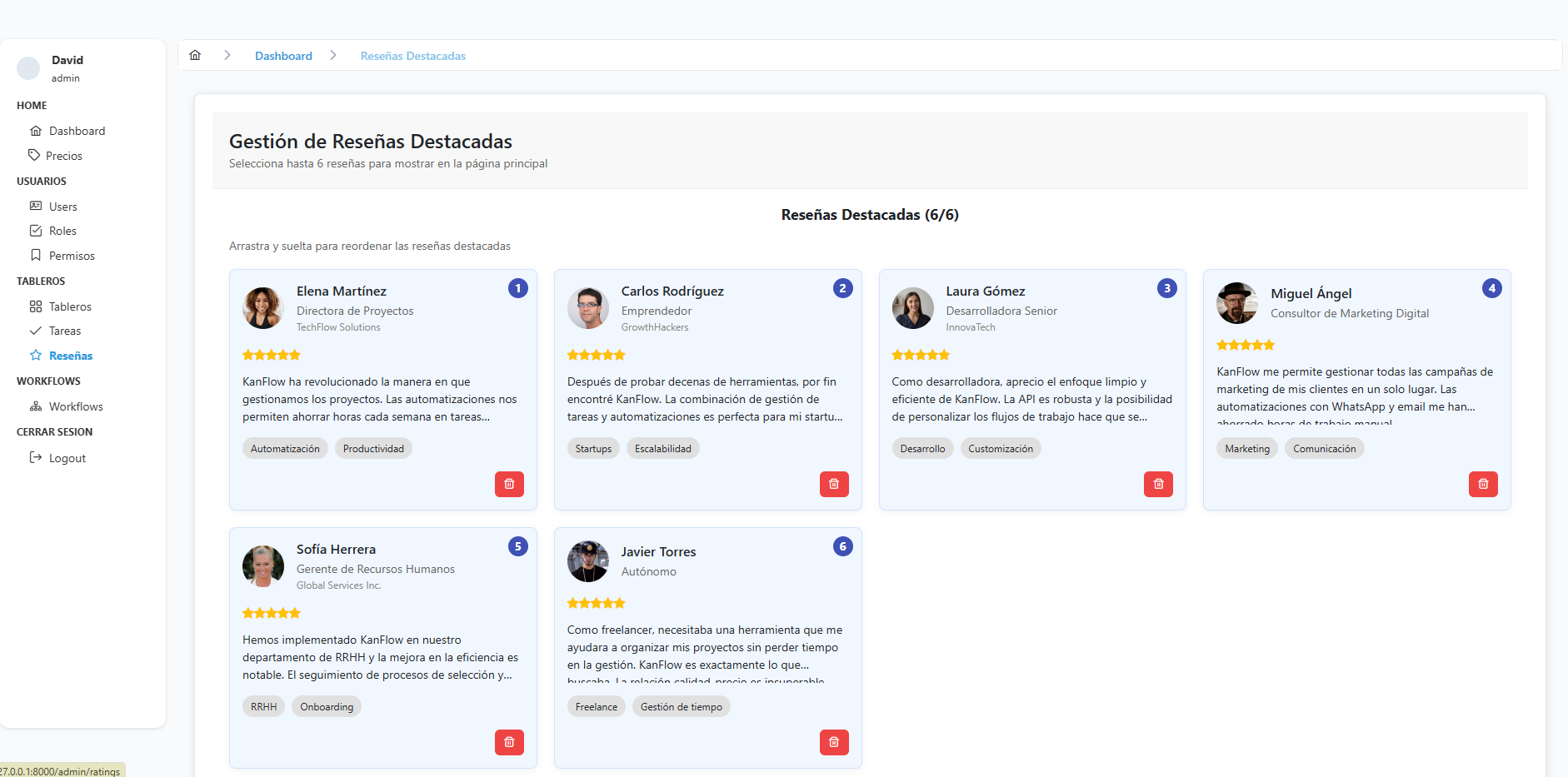


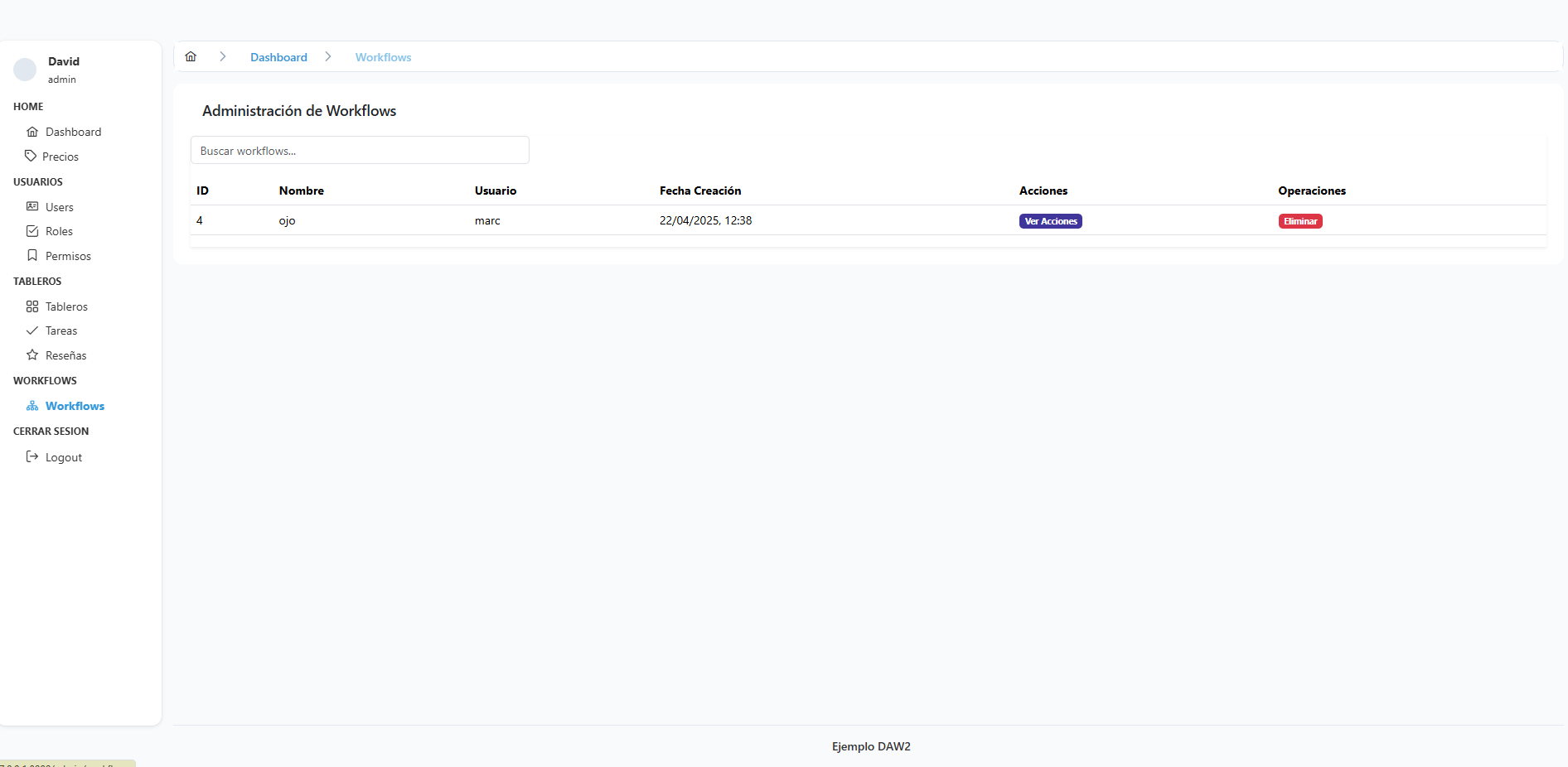












# Conclusiones

Este proyecto ha sido un buen reto y, a la vez, una gran oportunidad para aprender. Lo más complicado fue implementar el editor de bloques y hacer que todo funcionara en tiempo real. Aunque no fue fácil, aprendimos mucho en el proceso. Trabajar con Laravel y Vue nos ayudó a mejorar en desarrollo full-stack. También nos gustó poder elegir libremente la idea del proyecto y contar con mucho tiempo para desarrollarlo. Al final, fue exigente, pero lo disfrutamos bastante.

# Referencias

Claude: <https://claude.ai/>

Chat GPT: [https://chatgpt.com](http://chatgpt.com)

Asilvabe: <https://asilvabe.dev/blog/vue-draggable>

Stack Overflow: <https://stackoverflow.com/>