**CICLO DE GRADO SUPERIOR EN DESARROLLO DE APLICACIONES WEB**

Proyecto de desarrollo de aplicaciones web

**TOTALTASKS**

Autor: Diego Arroyo González y Javier Rubio Gigante

Tutor: Daniel González-Calero

Año:2025

Contenido

[1.Introducción 4](#_Toc200563425)

[1.1 Planteamiento del problema 5](#_Toc200563426)

[1.2 Objetivos 6](#_Toc200563427)

[1.3 Metodologías 7](#_Toc200563428)

[2 Descripción del problema 9](#_Toc200563429)

[3 Proyecto Web 11](#_Toc200563430)

[3.1 Lenguajes utilizados 11](#_Toc200563431)

[3.2 Interfaz de usuario 12](#_Toc200563432)

[4 Solución del problema 14](#_Toc200563433)

[4.1 Pantalla de inicio de sesión 14](#_Toc200563434)

[4.2 Pantalla de registro 18](#_Toc200563435)

[4.2.1 Registro 18](#_Toc200563436)

[4.2.2 Validación y seguridad 19](#_Toc200563437)

[4.3 Pantalla de inicio de la aplicación 20](#_Toc200563438)

[4.4 Dashboard 23](#_Toc200563439)

[5 Evaluación 24](#_Toc200563440)

[6 Conclusión 27](#_Toc200563441)

[7 Referencias 28](#_Toc200563442)

[Figura 1 - Login 14](#_Toc200048709)

[Figura 2 - Inicio de sesión Google 15](#_Toc200048710)

[Figura 3 - Inicio sesión Github 16](#_Toc200048711)

[Figura 4 - Modal error login 17](#_Toc200048712)

[Figura 5 - Registro 18](#_Toc200048713)

[Figura 6 - Modal de error en el registro 19](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048714)

[Figura 7 - Página principal 20](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048715)

[Figura 8 - Sección de funcionalidades 21](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048716)

[Figura 9 - Sección de sectores 21](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048717)

[Figura 10 - Historias de productividad 22](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048718)

[Figura 11 - Preguntas frecuentes 22](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048719)

[Figura 12 - Vista del dashboard de proyectos en TotalTasks. 23](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048720)

[Figura 13 - Opciones de inicio de sesión. 24](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048721)

[Figura 14 - Vista del calendario con tareas sincronizadas. 24](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048722)

[Figura 15 - Chat integrado en un proyecto. 25](file:///C:\Users\Javier\Documents\Obsidian\DAW\TotalTasks\Documentation\Memoria%20Proyecto%20Fin%20De%20Grado%20Diego%20Arroyo%20González%20Javier%20Rubio%20Gigante.docx#_Toc200048723)

**RESUMEN**

Tras realizar un estudio acerca de las plataformas actuales de gestión de proyectos, se pudo apreciar que la gran mayoría son de **pago** o presentan **muchas limitaciones** en sus versiones gratuitas. Esto motivó el desarrollo de una herramienta completamente **funcional**, **intuitiva** y **accesible** que permitiera a cualquier equipo, sin importar su tamaño o presupuesto, organizar y gestionar sus proyectos de manera profesional y flexible.

Basado en las metodologías de trabajo más utilizadas en la industria, como *Scrum* y *Kanban*, y utilizando tecnologías actuales, se ha decidido desarrollar esta plataforma desde cero. Se trata de una aplicación web desarrollada con *Spring Boot*, *Visual Studio Code* y una base de datos *MySQL*, con control de versiones a través de *GitHub*, pensada específicamente para entornos colaborativos y equipos de trabajo.

Se ha dedicado especial atención al sistema de gestión de proyectos, verdadero núcleo de esta aplicación. Cada proyecto se adapta dinámicamente a su metodología. En los proyectos *Scrum*, es posible crear historias de usuario, dividirlas en tareas, agruparlas por *Sprints* y gestionar el *sprint backlog*, todo desde una interfaz clara y directa.

En los proyectos *Kanban*, el usuario dispone de **tableros configurables** que permiten arrastrar tareas, organizarlas por columnas personalizadas, y adaptar el flujo de trabajo a las necesidades concretas del equipo. Se ha puesto especial cuidado en que cada metodología se represente de forma clara y útil, permitiendo a los equipos trabajar cómodamente sin barreras técnicas.

Para mantener la atención del usuario y fomentar el trabajo colaborativo, se han integrado **múltiples herramientas complementarias**, como un sistema de notificaciones inteligente que avisa de tareas asignadas o próximas a vencer, un chat interno por proyecto para mejorar la comunicación entre miembros, y la posibilidad de sincronizar tareas con *Google Calendar*, haciendo más cómodo su seguimiento.

Otro aspecto importante ha sido la integración con *GitHub*, que permite importar repositorios, asociar tareas a colaboradores y mantener la coherencia entre la planificación del proyecto y su código fuente.

También se ha trabajado en el sistema de autenticación, permitiendo acceder mediante correo y contraseña o utilizando cuentas de *Google* o *GitHub*, facilitando así el acceso y vinculación con herramientas externas.

Por último, se ha buscado que la interfaz sea **limpia**, **rápida** e **intuitiva**. Desde la página de inicio hasta el dashboard de proyectos, cada apartado ha sido pensado para que el usuario se sienta cómodo, pueda navegar con fluidez y configure su entorno de trabajo sin complicaciones.

**1.Introducción**

El proyecto nace como respuesta a la creciente necesidad de contar con herramientas de gestión de **proyectos accesibles, funcionales y adaptados** a los nuevos entornos de trabajo colaborativos.

Tras analizar las plataformas más populares del mercado, se detectó una clara barrera: la mayoría requieren de **suscripciones de pago** para acceder a funcionalidades realmente útiles, lo que limita su uso a empresas con mayores recursos.

Con esta premisa, se ha desarrollado una aplicación web pensada para equipos de trabajo y pequeñas empresas, que permite gestionar sus proyectos de forma profesional, aprovechando las tecnologías actuales y **ofreciendo una experiencia fluida, moderna y personalizable.**

A través de esta plataforma, los usuarios pueden organizar sus tareas, colaborar en tiempo real, y adaptar su flujo de trabajo ya sea con **metodologías ágiles** como *Scrum* o visuales como *Kanban*.

Uno de los principales objetivos ha sido evitar que la gestión del proyecto se convierta en un proceso monótono o encorsetado. Para ello, se han implementado múltiples funcionalidades que aportan **dinamismo** a la herramienta, como **tableros totalmente personalizables**, integración con *GitHub*, sincronización con *Google Calendar*, un sistema de notificaciones inteligente y un chat por proyecto, todo enfocado a mejorar la experiencia del usuario y mantener su atención y eficiencia.

A continuación, se desarrollarán de forma detallada todos los aspectos del proyecto, desde la idea inicial hasta la implementación completa de todas sus funcionalidades.

## Planteamiento del problema

La idea de desarrollar esta plataforma de gestión de proyectos surge tras analizar las herramientas más utilizadas en entornos colaborativos y descubrir que, en su mayoría, son **soluciones limitadas** por planes de pago o con **funcionalidades restringidas** que no siempre se ajustan a las verdaderas necesidades de los equipos.

Este proyecto nace con el propósito de ir un paso más allá, ofreciendo una alternativa **gratuita y flexible** que permita a cualquier equipo organizar su trabajo de forma eficiente sin barreras económicas.

Uno de los principales retos abordados ha sido construir una solución que no solo sea funcional, sino también **intuitiva y adaptable a distintos estilos de trabajo.**

En muchas plataformas actuales se detecta una cierta rigidez a la hora de implementar metodologías como *Scrum* o *Kanban*, lo que frustra la experiencia del usuario y limita su capacidad para gestionar los proyectos a su manera. Aquí, en cambio, se ha apostado por una arquitectura que permita **moldear la aplicación al flujo real de trabajo de cada equipo.**

La aplicación está pensada para todo tipo de usuarios, desde estudiantes hasta equipos profesionales, gracias a una interfaz limpia, opciones de personalización, y un enfoque centrado en la colaboración.

El objetivo no es solo gestionar tareas, sino fomentar una **dinámica de trabajo ágil**, transparente y continua, donde los usuarios puedan **comunicarse**, planificar *sprints*, asignar tareas y ver el progreso de un proyecto de forma **clara y visual.**

Con todo esto, se espera alcanzar un amplio rango de usuarios, desde jóvenes que comienzan a trabajar en equipo en sus primeros proyectos hasta empresas que buscan una solución ligera, potente y sin costes ocultos. Se trata, en definitiva, de una herramienta sin barreras, con potencial de crecimiento continuo y sin límites definidos en cuanto a su evolución y adopción.

## Objetivos

Los objetivos establecidos a la hora de desarrollar esta plataforma de gestión de proyectos estuvieron claros desde el principio, ya que el equipo conocía bien las carencias de las herramientas actuales y tenía una visión clara de lo que un usuario necesita al enfrentarse al trabajo colaborativo. Desde el inicio se ha buscado construir una solución que cumpla los siguientes objetivos:

* Crear una plataforma de gestión de proyectos flexible que permita trabajar con metodologías ágiles como *Scrum* o proyectos *Kanban*, sin imponer limitaciones y adaptándose a la estructura de cada equipo.
* Ofrecer una interfaz **moderna e intuitiva** que combine una apariencia atractiva con una experiencia de usuario fluida, accesible tanto para principiantes como para profesionales.
* Dotar al sistema de una fuerte **integración con servicios externos** como *GitHub* y *Google Calendar*, permitiendo así automatizar parte del flujo de trabajo y conectar el proyecto con herramientas del día a día.
* Fomentar la **colaboración y la comunicación** dentro de los proyectos, integrando funciones como chat por proyecto, notificaciones inteligentes y asignación personalizada de tareas.
* Diseñar una aplicación **escalable**, que pueda crecer en funcionalidades y volumen de usuarios sin perder **rendimiento**, y que permita futuras integraciones en forma de módulos o funcionalidades añadidas.
* Desarrollar una herramienta basada en tecnologías modernas como *Spring Boot*, *MySQL* y servicios *OAuth* para garantizar seguridad, velocidad y una base sólida para futuras actualizaciones y expansiones.

## Metodologías

Todo lo relacionado con el control del desarrollo de esta plataforma se ha llevado a cabo de una manera sencilla pero efectiva, buscando en todo momento mantener la calidad del código y la coherencia en el proceso de trabajo.

Desde un primer momento se utilizó *GitHub* como sistema de control de versiones, lo que ha permitido que todos los integrantes del equipo pudieran trabajar simultáneamente en las distintas funcionalidades, sin poner en riesgo la estabilidad del proyecto.

Además, gracias al uso de ramas y *pull requests*, se ha podido mantener una **trazabilidad clara de cada avance.**

Durante las fases iniciales del desarrollo, se utilizó la herramienta *Taiga* para organizar y dividir las tareas según el enfoque ágil. De esta forma, se comenzaron los primeros *sprints* siguiendo una metodología tipo *Scrum*, con historias de usuario y tareas asignadas a cada miembro del equipo.

A medida que nuestra propia plataforma fue avanzando, migramos progresivamente la gestión de tareas y sprints a nuestro propio entorno, utilizando nuestros tableros *Kanban* internos para experimentar de forma directa con las funcionalidades que íbamos implementando.

Esto no solo nos permitió validar nuestras herramientas, sino también identificar mejoras desde el punto de vista del usuario final.

El trabajo se ha estructurado en evolutivos, priorizando siempre aquellas funcionalidades que servían de base para otras dependientes. Por ejemplo, si una funcionalidad como la autenticación era imprescindible para otras como la gestión de proyectos, se abordaba primero, garantizando así fluidez en el desarrollo. No se subía código sin probar, y cada funcionalidad era testada localmente antes de integrarse en el repositorio principal.

Las decisiones importantes se tomaban de forma consensuada entre los miembros del equipo, y las tareas se repartían equitativamente, respetando en todo momento las preferencias y fortalezas de cada uno.

En cuanto a la parte estética y funcional, se optó por un enfoque escalonado:

* Primero se desarrollaba la funcionalidad completa, asegurando que no hubiera errores o conflictos, y una vez validada, se procedía a integrar los estilos visuales y detalles de interfaz. Esto permitió una mayor estabilidad durante el desarrollo y una integración más limpia de la experiencia de usuario.
* Por último, tras cada revisión o tutoría, se incorporaban de inmediato los cambios o sugerencias indicadas, permitiendo así mantener un ritmo constante y minimizar bloqueos.

A lo largo del desarrollo también surgieron nuevas ideas y funcionalidades que se valoraban en función del tiempo y valor añadido;

Si resultaban viables, se incorporaban progresivamente al proyecto, haciendo que la plataforma evolucionara de forma orgánica y coherente con los objetivos iniciales.

# Descripción del problema

En el panorama actual de herramientas de gestión de proyectos, la mayoría de las soluciones disponibles presentan limitaciones importantes: planes de pago que **restringen funcionalidades clave**, **interfaces poco intuitivas**, y una escasa adaptabilidad a los distintos estilos de trabajo. Esto representa un obstáculo, especialmente para estudiantes, equipos pequeños o iniciativas que buscan colaboración real sin incurrir en costes.

Además, muchas de estas plataformas no contemplan una experiencia de usuario moderna, integrada y personalizada que fomente la colaboración ágil. Esto deja un vacío entre la necesidad de herramientas eficientes y accesibles, y lo que realmente ofrecen las opciones comerciales.

TotalTasks nace para dar respuesta a esas carencias, ofreciendo una plataforma **gratuita, versátil y accesible** para cualquier tipo de usuario, sin importar su experiencia técnica o el tamaño de su equipo. Se plantea como una solución funcional e intuitiva, adaptada tanto a metodologías ágiles (como Scrum) como a sistemas más visuales y flexibles (como Kanban), con un alto grado de personalización.

A nivel funcional, la aplicación permite:

* Autenticación cifrada.
* Dashboards diferenciados según el método de autenticación.
* Gestión de proyectos personalizados.
* Personalización visual de la plataforma a través de herramientas integradas.
* Calendario integrado.
* Sistema de notificaciones.
* Chat interno dentro de cada proyecto, que fomenta la colaboración directa y ágil entre los miembros.
* Perfil de usuario configurable.
* Validaciones de formularios.

Entre los escenarios de uso y manejo de errores, se contemplan los siguientes:

* Acceso a URLs no existentes: se muestra una página de error 404 personalizada.
* Intento de creación o unión a un proyecto con nombre ya existente: se genera un mensaje de error.
* Intento de acceso a una pantalla protegida sin autenticación previa: redirección al login.
* Comportamientos alternativos según el estado del usuario (por ejemplo, si ya está logueado, se evita mostrar nuevamente el login).
* Detección y control de proyectos con nombres duplicados en dashboards individuales.
* Gestión dinámica del flujo de tareas dentro de los sprints y su retorno al backlog si no se completan a tiempo.

Esta solución está pensada para crecer junto con las necesidades de los usuarios, permitiendo una evolución continua sin costes ni restricciones, y ofreciendo una experiencia robusta tanto para nuevos usuarios como para equipos experimentados que buscan eficiencia, claridad y colaboración real.

# Proyecto Web

## Lenguajes utilizados

Para el desarrollo de esta aplicación web se han utilizado los siguientes lenguajes de programación y marcado.

A continuación, se describe el uso principal de cada uno de estos lenguajes dentro del proyecto:

* Java: es el lenguaje principal utilizado para la lógica del servidor. A través del framework Spring Boot se ha estructurado la arquitectura interna de la aplicación, controlando aspectos como la gestión de peticiones, servicios, controladores y conexión con la base de datos. Spring ha permitido crear una aplicación robusta, escalable y de fácil mantenimiento.
* HTML y CSS: se han utilizado para estructurar y dar estilo a la interfaz de usuario. Gracias a su uso conjunto se ha podido diseñar un entorno visual claro, limpio y coherente que favorece la experiencia del usuario, manteniendo la simplicidad que caracteriza a esta herramienta.
* JavaScript: utilizado para dotar de interactividad a la aplicación en el lado del cliente. Junto con librerías como jQuery, permite realizar acciones sin necesidad de recargar la página completa, mejorando la fluidez de la navegación. Además, se ha hecho uso de SortableJS para implementar funcionalidades de drag and drop dentro de los tableros Kanban, y de FullCalendar para la visualización dinámica y manejable de calendarios de tareas, eventos y planificación.
* XML: presente en la configuración del proyecto a través del archivo pom.xml, que gestiona las dependencias de Maven en el entorno de desarrollo de Spring Boot.

El uso coordinado de estos lenguajes ha permitido construir una aplicación moderna, funcional y adaptable, sentando unas bases técnicas sólidas para posibles ampliaciones futuras. Cada uno de los lenguajes ha sido escogido en función de su idoneidad para las distintas capas de la aplicación, asegurando un desarrollo limpio y modular.

## Interfaz de usuario

En cuanto a la interfaz de usuario, se ha optado por un diseño **limpio y moderno**, desarrollado íntegramente a medida sin utilizar *frameworks* como *TailwindCSS* o *Bootstrap*. En su lugar, se ha creado un archivo general.css común que define una paleta de colores **coherente**, tipografías, tamaños y estilos reutilizables para todo el proyecto, lo cual permite mantener la estética unificada en todas las vistas.

Se ha dado prioridad a la claridad visual, la **coherencia** y la **accesibilidad**. Para ello, se ha utilizado la fuente *Poppins*, animaciones personalizadas y estructuras claras que guían al usuario sin sobrecargar la pantalla. El objetivo ha sido garantizar que tanto usuarios nuevos como experimentados puedan orientarse fácilmente dentro de la plataforma y realizar sus tareas sin fricción.

Aunque la mayoría de los componentes visuales se han diseñado desde cero, en ciertos casos puntuales se han reutilizado botones estilizados de la plataforma [uiverse.io](https://uiverse.io/), adaptándolos para que se integren con el resto del diseño. Estos botones se han personalizado con la paleta de colores del sistema y están presentes en elementos interactivos clave como formularios, modales o paneles de acción.

El calendario del proyecto es una excepción en cuanto a diseño, ya que se ha construido sobre la base de **FullCalendar**, una librería especializada que ofrece gran funcionalidad. A pesar de usar una base externa, el diseño visual se ha modificado mediante *CSS* para mantener la coherencia estética con el resto del sistema.

A nivel estructural, se han implementado fragmentos reutilizables mediante *Thymeleaf* para **optimizar** y **centralizar** elementos comunes como el *header*, el *footer* y el *chatbot* de ayuda. Estos fragmentos permiten mantener el código *HTML*, *CSS* y *JS* organizado y fácil de actualizar, además de garantizar que estos elementos estén presentes de forma consistente en todas las páginas de la aplicación.

Durante el desarrollo, también se han aplicado efectos visuales interactivos (*hover*, animaciones de aparición, cambios de color en botones, etc.) que aportan una respuesta clara al usuario cada vez que interactúa con la interfaz. Estos detalles ayudan a mejorar la experiencia de uso al hacer la interfaz más **dinámica** y **comprensible**.

La navegación entre vistas, como el panel de inicio, la creación de proyectos, el área de tareas en formato *Kanban*, el *backlog* de *sprints* o el calendario, mantiene una estructura común que refuerza la familiaridad con el sistema y evita que el usuario se sienta desorientado.

En definitiva, se ha logrado una interfaz funcional, visualmente agradable y centrada en la experiencia del usuario, diseñada a medida para cubrir las necesidades de gestión de proyectos de manera clara y eficaz.

# Solución del problema

TotalTasks es una plataforma web desarrollada para cubrir las necesidades actuales de gestión de proyectos y tareas en entornos colaborativos. Su diseño combina accesibilidad, seguridad y flexibilidad, y se estructura en torno a funcionalidades clave que abordan directamente los problemas detectados.

## Pantalla de inicio de sesión

La pantalla de inicio de sesión permite a los usuarios autenticarse en *TotalTasks* utilizando su cuenta registrada o mediante proveedores externos como **Google** o **GitHub**. El diseño de la vista es prácticamente idéntico al de registro, manteniendo una experiencia visual coherente.

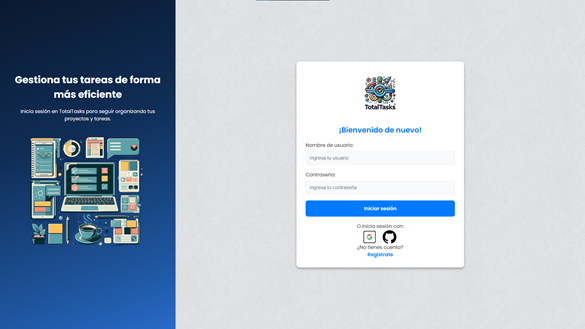


Figura 1 - Login

**Estructura de la interfaz**

La vista está dividida en dos columnas:

* **Izquierda**: igual que en la pantalla de registro, con un fondo en degradado azul corporativo, una imagen decorativa y un mensaje de bienvenida a la plataforma.
* **Derecha**: contiene el formulario de inicio de sesión sobre un fondo texturizado gris claro, dentro de un contenedor limpio y moderno.

**Formulario de inicio de sesión manual**

El formulario solicita dos datos básicos:

* Nombre de usuario
* Contraseña

Al hacer clic en “**Iniciar sesión**”, se valida que ambos campos estén completos y luego se realiza una petición *POST* al *endpoint* /usuarios/comprobarLogin, enviando los datos como *JSON*. Si la respuesta es "Encontrado", el usuario es redirigido al **dashboard** de la plataforma. En caso de error (usuario incorrecto, contraseña inválida, etc.), se muestra un **modal de error** con el mensaje correspondiente, utilizando la misma interfaz y animación suave que en la pantalla de registro.

**Inicio de sesión con Google**

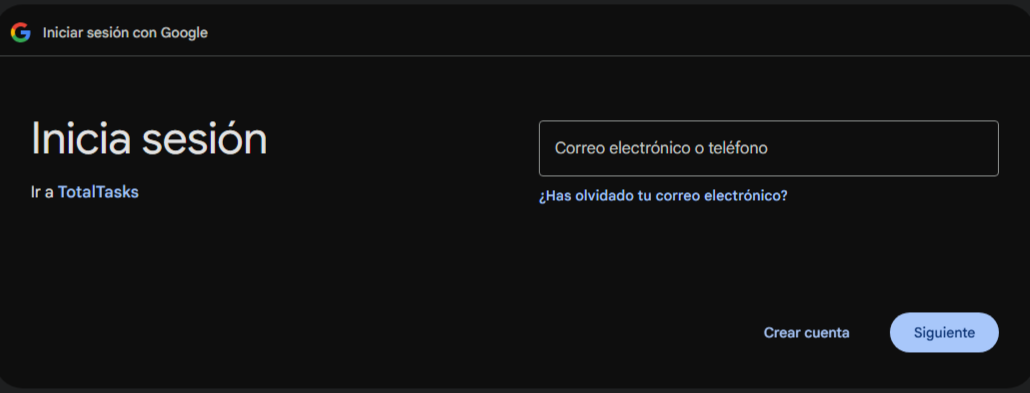
****

Figura 2 - Inicio de sesión Google

La plataforma integra *Google Sign-In* utilizando el cliente oficial *de Google Identity*. Al iniciar sesión con Google:

1. Se decodifica el JWT recibido para extraer nombre, correo e imagen de perfil.
2. Se construye un objeto con la información y se envía una petición POST al backend vía /usuarios/registrarGoogle.
3. Si el usuario ya está registrado o se registra automáticamente, es redirigido al **dashboard**.

Este flujo permite que nuevos usuarios creen su cuenta sin necesidad de rellenar manualmente un formulario, y que usuarios existentes se autentiquen rápidamente.

**Inicio de sesión con GitHub**

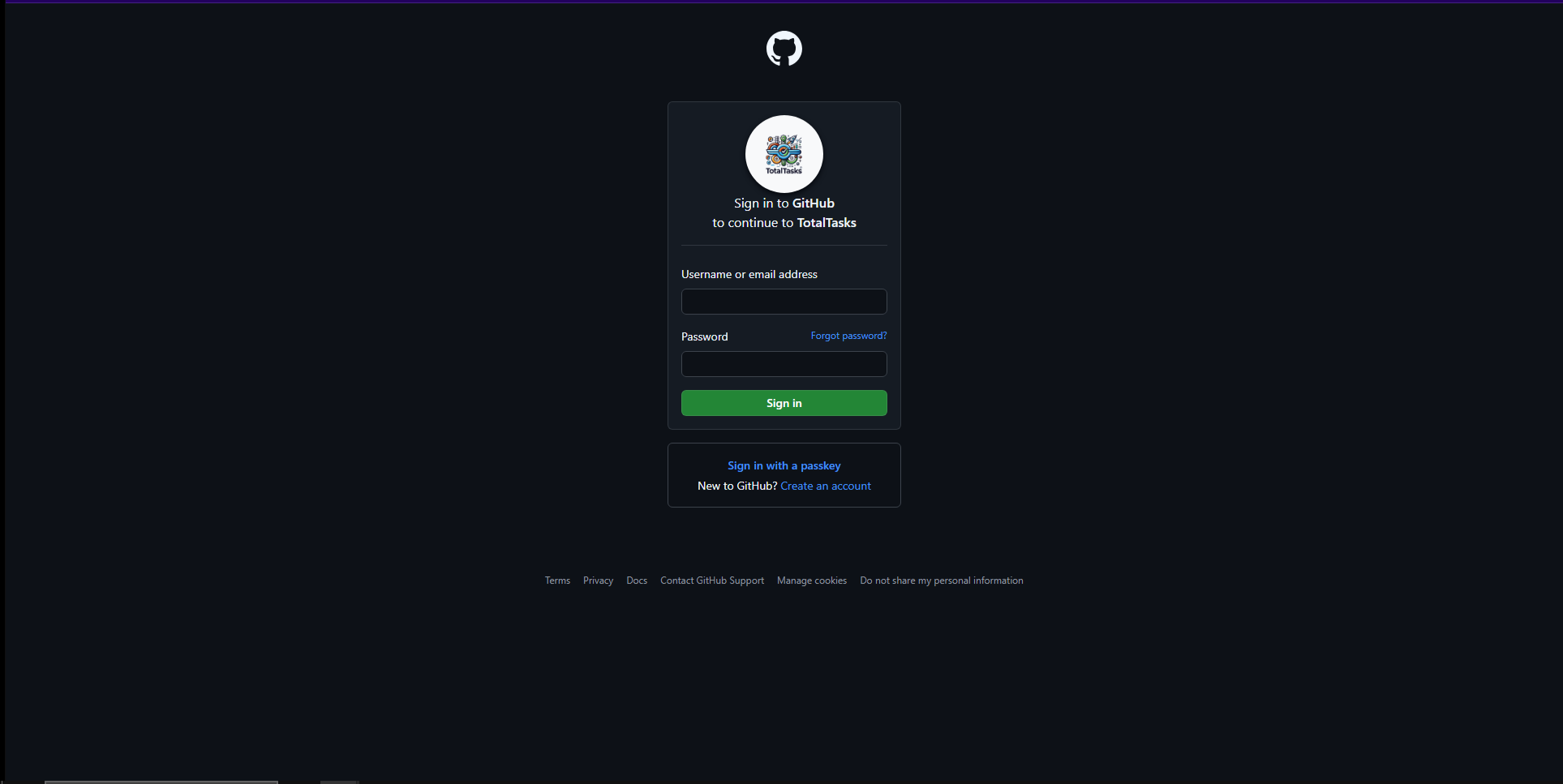
****

Figura 3 - Inicio sesión Github

El botón de *login* con *GitHub* redirige al usuario al flujo de autenticación OAuth de *GitHub*, solicitando permisos extendidos que incluyen:

* Lectura del perfil de usuario
* Acceso al correo electrónico
* Gestión de repositorios (incluyendo *hooks* y eliminación)
* Acceso a *gists*

Una vez completado el proceso, *GitHub* redirige al usuario a /usuarios/githubCallback, donde se gestiona la autenticación e incorporación del usuario a la plataforma.

**El inicio de sesión mediante GitHub aporta una serie de ventajas adicionales que enriquecen la experiencia del usuario.** Al autenticarte con esta plataforma, TotalTasks accede a los repositorios del usuario, lo que permite ofrecer un *dashboard* personalizado desde el que se pueden crear proyectos directamente a partir de estos repositorios.

Además, dentro de cada proyecto asociado a GitHub, se habilita un menú específico con funcionalidades avanzadas como:

* Enlace directo al repositorio original.
* Acceso al despliegue operativo si el proyecto cuenta con GitHub Pages u otra solución de hosting conectada.
* Visualización de estadísticas relevantes sobre el repositorio, como poder visualizar número de *commits* por autor, *lenguajes de programación* utilizadoso las diferentes ramas del proyecto.

Estas integraciones convierten a TotalTasks en una solución especialmente potente para desarrolladores y equipos técnicos que ya utilizan GitHub como entorno principal de trabajo.

**Seguridad y experiencia de usuario**

****

Figura 4 - Modal error login

* Se utilizan los mismos modales de error y éxito que en la pantalla de registro para mantener coherencia visual.
* Las credenciales se gestionan de forma segura en el backend, y las contraseñas están encriptadas usando Bcrypt.
* Las integraciones con Google y GitHub simplifican el proceso de autenticación y reducen fricción para el usuario.

## Pantalla de registro

### Registro

La pantalla de **registro manual de usuarios** permite a nuevos visitantes crear una cuenta en TotalTasks completando un formulario sencillo y seguro. Esta vista se divide visualmente en **dos secciones principales**:

* **Sección izquierda:** presenta un fondo con degradado en tonos azules corporativos, una imagen decorativa y un mensaje motivador:

*“Gestiona tus tareas de forma más eficiente. Únete a TotalTasks, la herramienta en la nube para llevar el control de tus proyectos donde quiera que estés.”*

* **Sección derecha:** contiene el formulario de registro sobre un fondo texturizado gris claro. Este formulario está compuesto por los siguientes campos:
  + Nombre (validado para evitar números)
  + Nombre de usuario
  + Correo electrónico
  + Contraseña (con validaciones de seguridad)
  + Casilla de aceptación de Términos y Condiciones

El diseño del formulario es limpio y funcional, adaptado para una experiencia fluida tanto en escritorio como en dispositivos móviles.

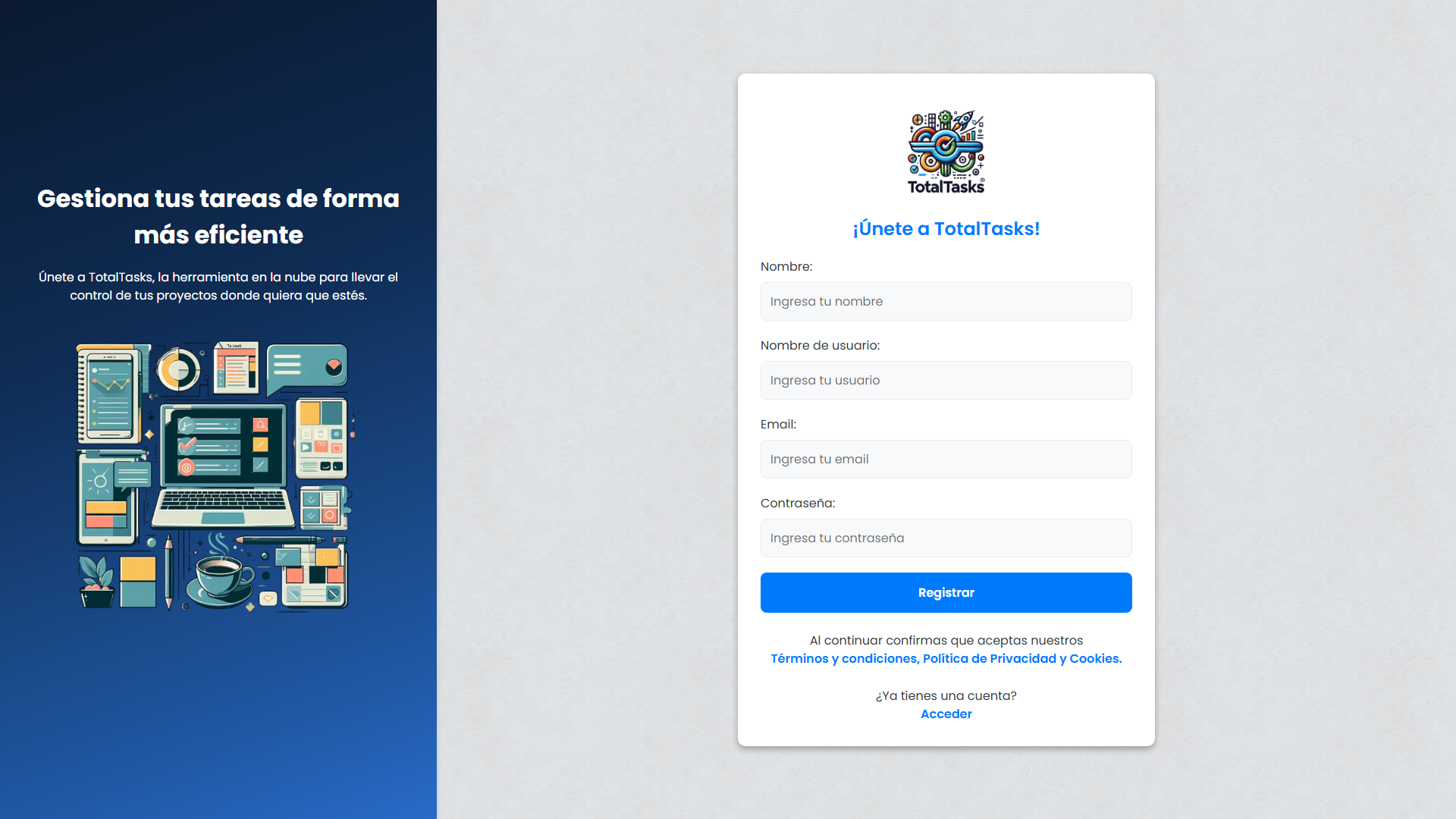


Figura 5 - Registro

### Validación y seguridad

La validación se realiza del lado del cliente mediante JavaScript, antes de enviar los datos al servidor:

* El **nombre** no debe contener números.
* La **contraseña** debe cumplir con los siguientes requisitos mínimos:
  + Al menos 8 caracteres
  + Una letra mayúscula
  + Un número

Al **detectar errores**, se activa un modal emergente con animación suave, desenfocando el fondo para resaltar el mensaje y facilitar la corrección por parte del usuario.

Una vez validado, se envía una petición *AJAX* con los datos al *backend* mediante *POST* a la ruta /usuarios/registrar. Si el servidor confirma la creación del usuario con el mensaje "Cuenta creada con Éxito.", se muestra un modal de éxito y se redirige automáticamente al login tras 3 segundos.

Las contraseñas se **encriptan en base de datos** usando *Bcrypt*, asegurando la confidencialidad de las credenciales.

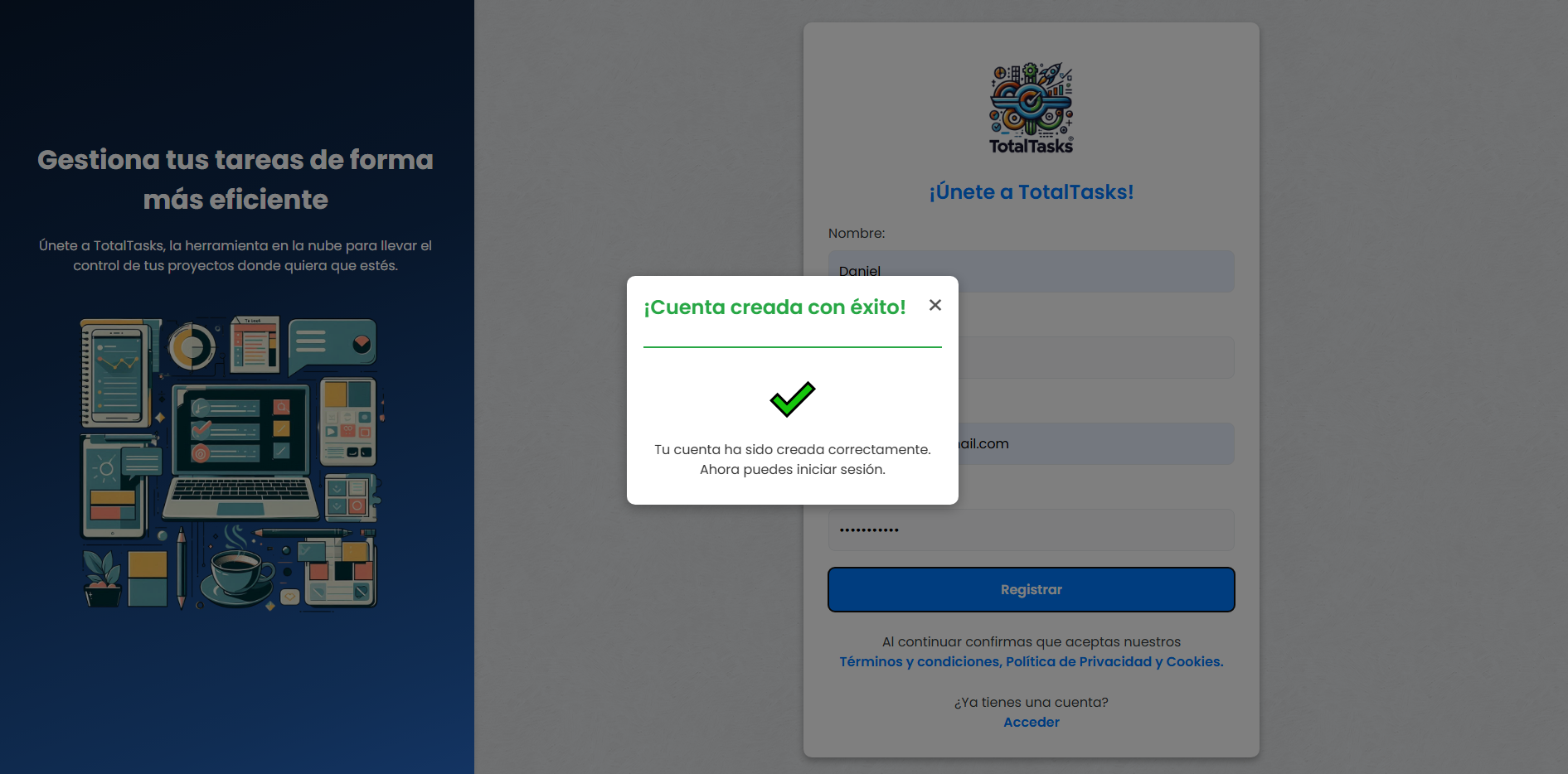


Figura 6 - Modal de éxito en el registro

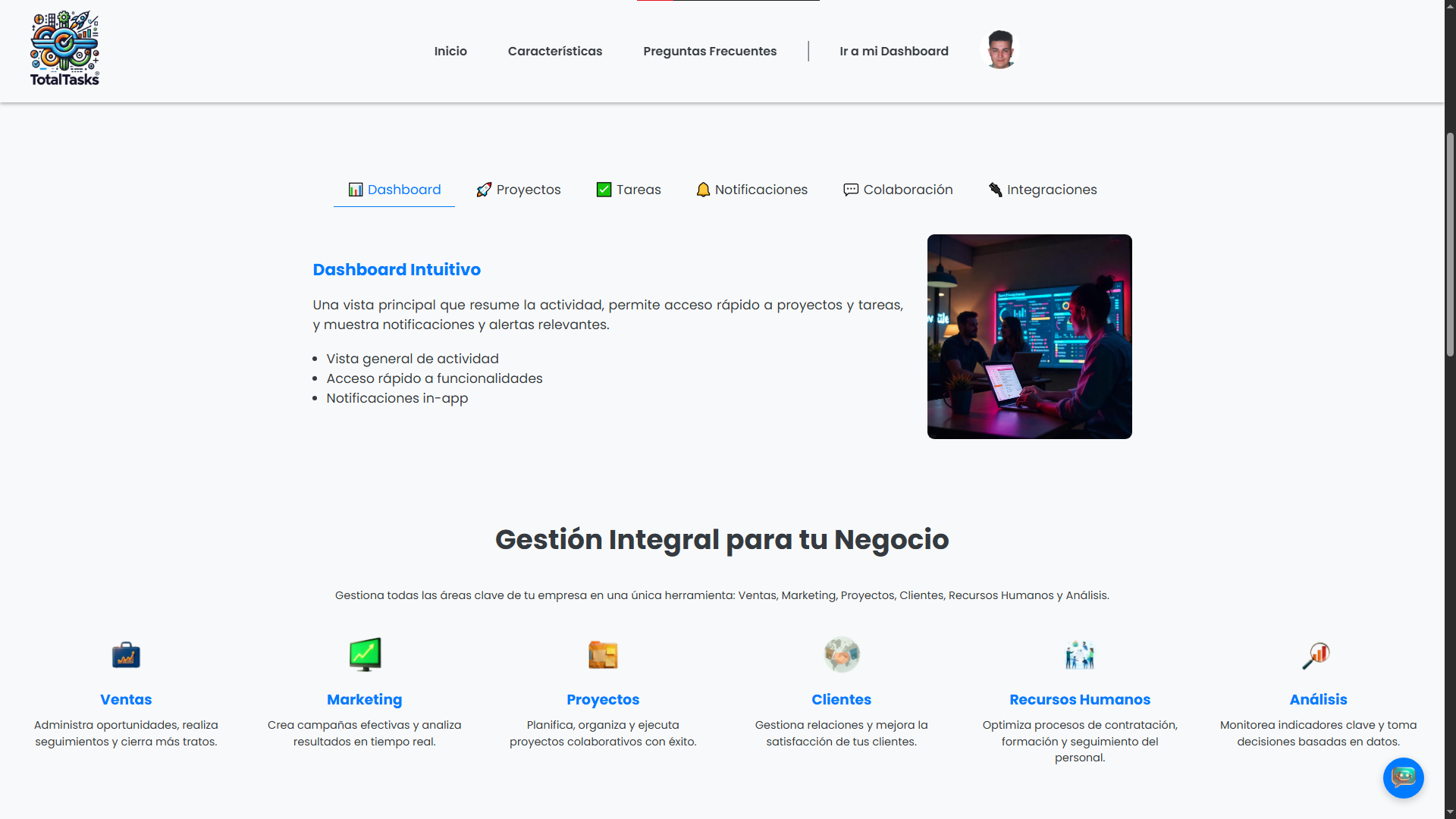
## Pantalla de inicio de la aplicación

En la pantalla principal de la aplicación se presenta una página orientada a captar el interés del usuario, mostrando de forma atractiva las principales funcionalidades y ventajas del sistema.

Esta página sirve como puerta de entrada a la plataforma, ayudando al usuario a comprender rápidamente qué ofrece la aplicación y animándole a registrarse o probarla sin compromiso.

Nada más entrar, el usuario puede ver un encabezado superior con acceso directo a las secciones más relevantes: **Inicio**, **Características**, **Preguntas** **frecuentes**, **Iniciar sesión** y un botón destacado de **Pruébala gratis**. En la esquina inferior derecha se encuentra también un botón flotante que despliega el **chatbot de ayuda**, disponible en todo momento para resolver dudas.

Figura 7 - Página de inicio

Más abajo se encuentran diversos bloques con iconos representativos que agrupan ventajas según áreas como **ventas, marketing, proyectos, clientes, recursos humanos y análisis,** reforzando la versatilidad de la herramienta en distintos sectores profesionales.

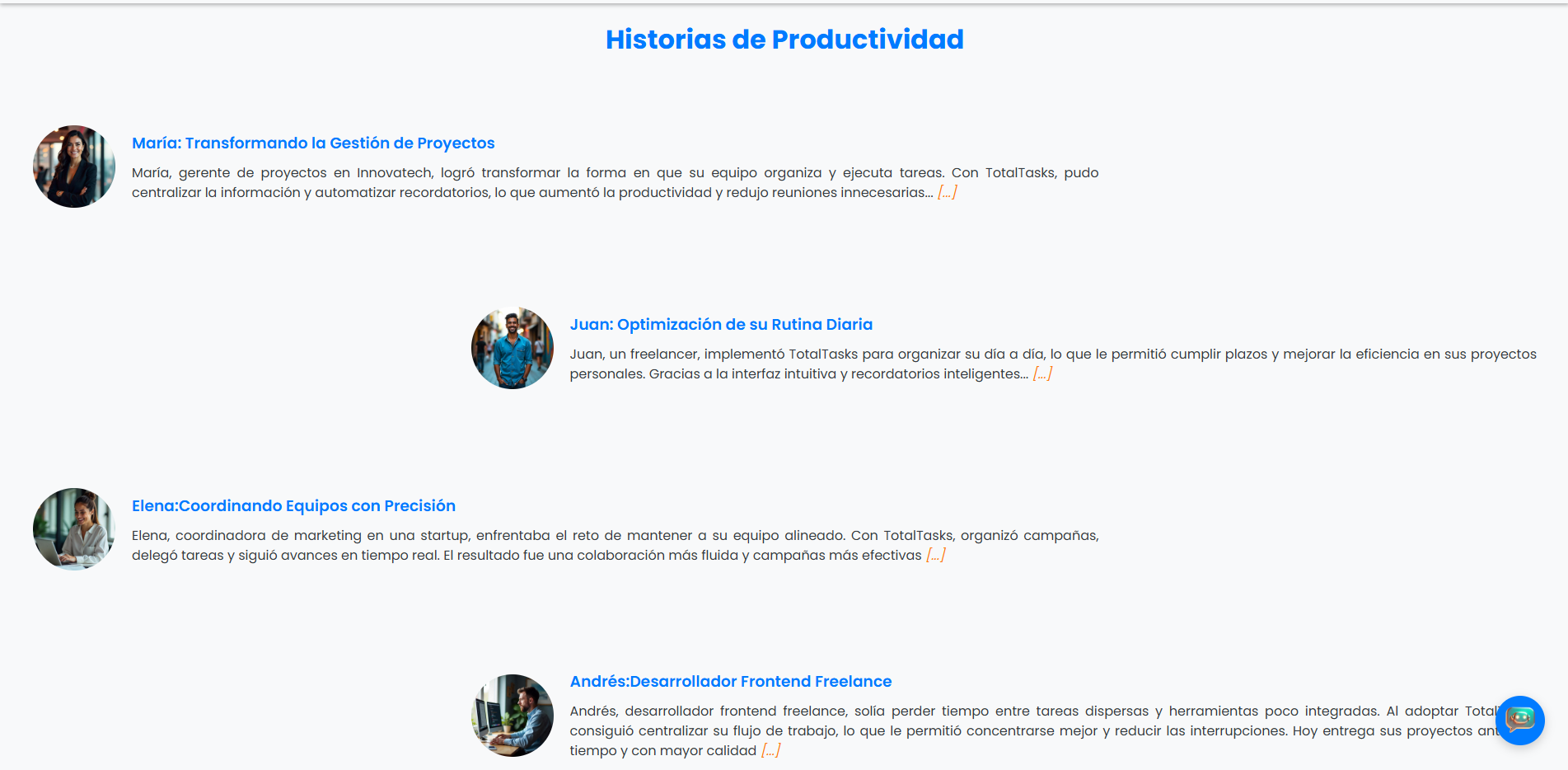
La parte inferior de la página presenta una sección titulada **Historias de productividad**, donde diferentes usuarios comparten con foto de perfil su experiencia utilizando la aplicación. Por último, se incluye un bloque de **preguntas frecuentes** para resolver posibles dudas comunes antes del registro, y el *footer* general que aparece en todas las páginas del sistema.

Figura 8 - Historias de productividad

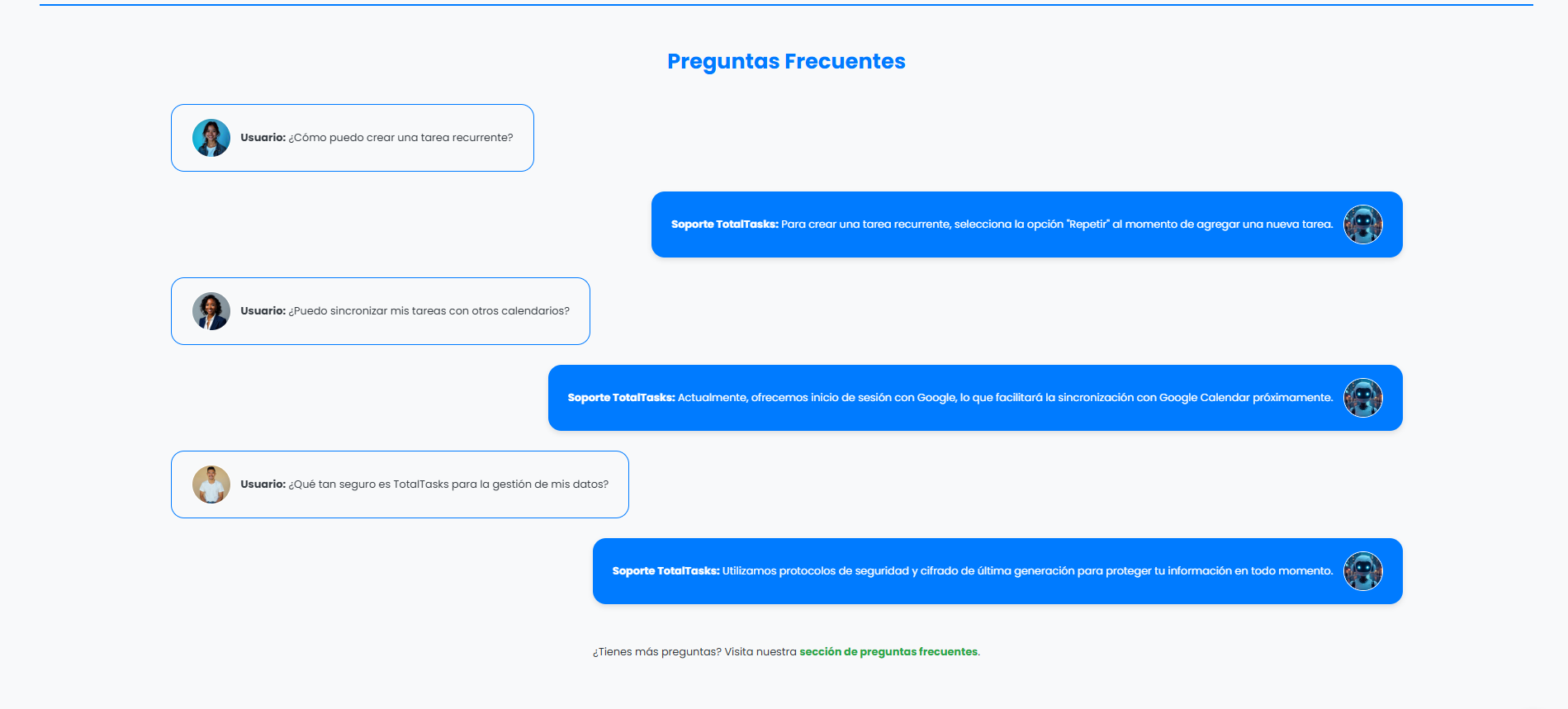


Figura 9 - Preguntas frecuentes

## Dashboard

La aplicación cuenta con **dos versiones del dashboard**, diferenciadas según el método de autenticación del usuario:

### Dashboard Genérico

Este dashboard se presenta a los usuarios que acceden mediante Google o el sistema de autenticación interno. Incluye:

* **Estructura general** con encabezado (*header*), pie de página (*footer*) y una sección principal donde se gestionan los proyectos.
* **Dos botones funcionales**:
  + **Crear Proyecto**: Abre un formulario modal donde el usuario puede ingresar el nombre del proyecto, una breve descripción y seleccionar la metodología de trabajo (*Scrum* o *Kanban*).
  + **Unirse a un Proyecto**: Permite unirse a proyectos existentes introduciendo un código generado por el sistema.

### Dashboard GitHub

Este dashboard está diseñado exclusivamente para los usuarios que inician sesión con GitHub, e incluye funcionalidades adicionales orientadas a la integración con repositorios:

* **Diseño dividido**: La pantalla se reparte en dos secciones:
  + Un contenedor lateral tipo *aside* a la izquierda que muestra los repositorios del usuario obtenidos desde GitHub.
  + La sección principal, similar al dashboard genérico, donde se gestionan los proyectos.
* **Autocompletado en la creación de proyectos**: Al crear un nuevo proyecto, los campos del formulario pueden autocompletarse utilizando datos del repositorio GitHub seleccionado (nombre, descripción, etc.).

### Gestión Visual de Proyectos

Ambas versiones del dashboard incluyen una sección visual donde se listan los proyectos creados o en los que el usuario participa. Cada proyecto aparece como un contenedor que muestra:

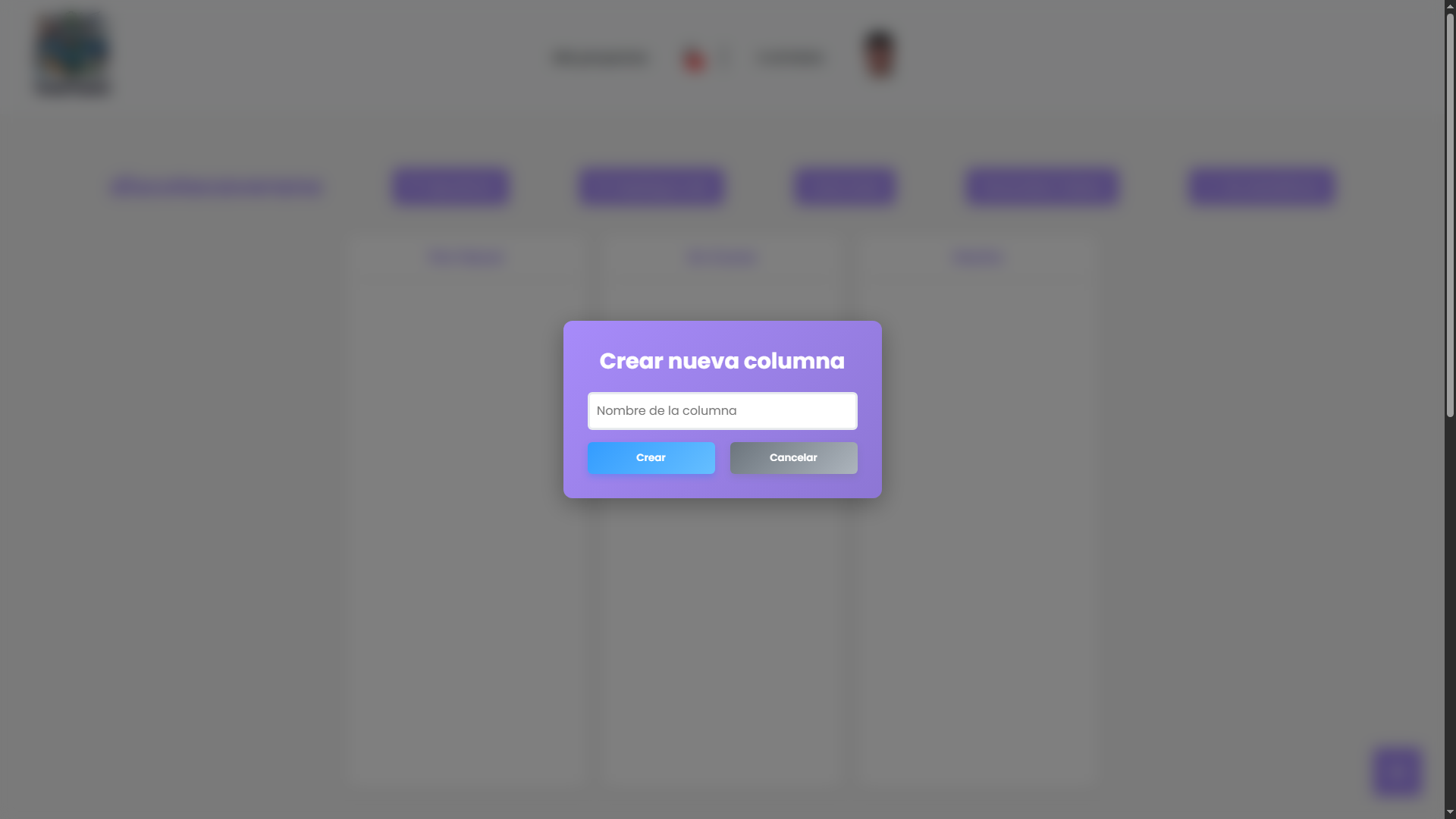
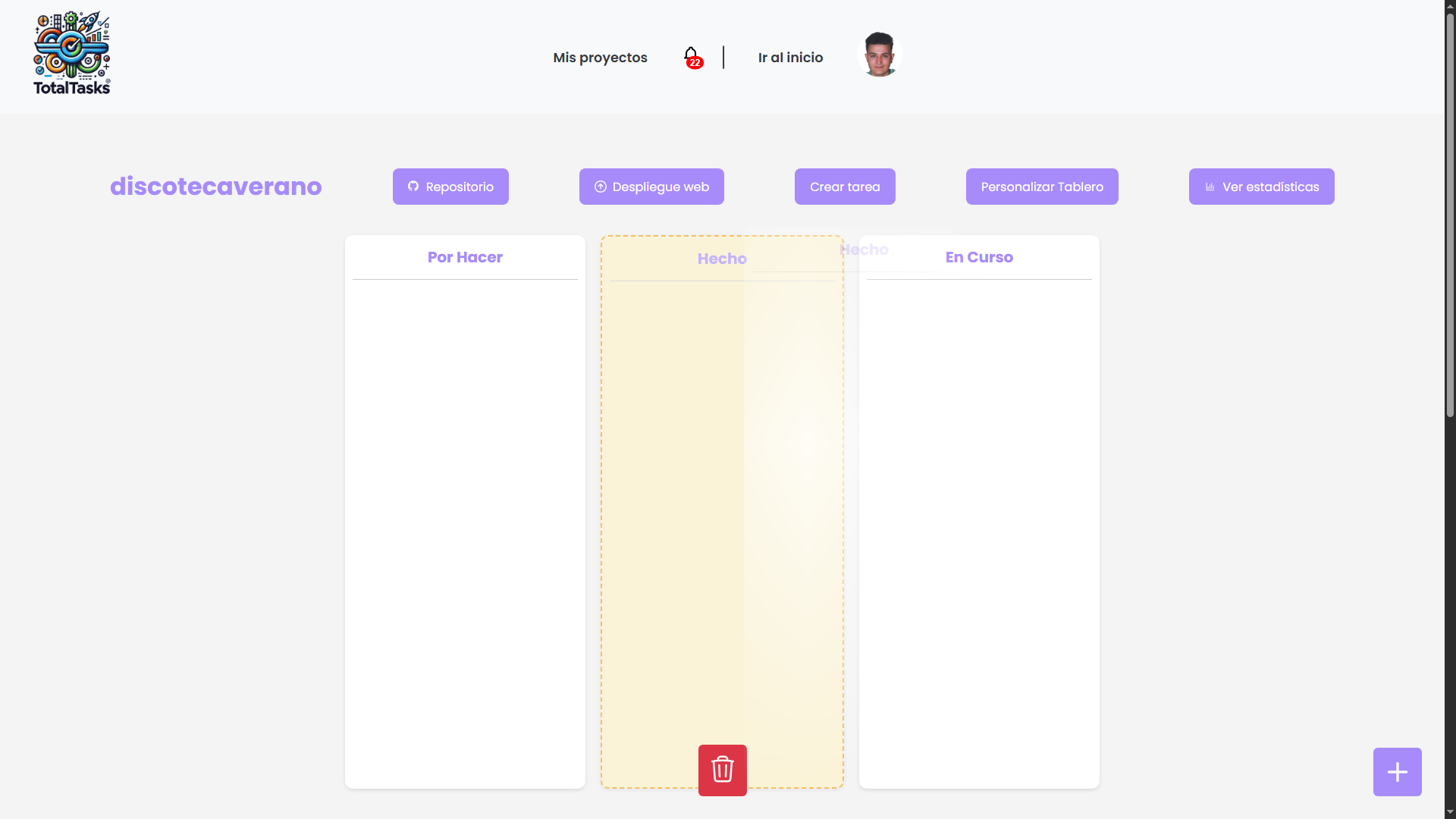
* El **nombre, descripción, metodología y color identificativo** del proyecto.
* Un botón para **eliminar el proyecto** (visible solo si el usuario es el administrador y creador).
* Un botón para **salir del proyecto** si el usuario solo es miembro.
* Un botón para **copiar al portapapeles** el código de unión al proyecto, facilitando la colaboración entre usuarios.

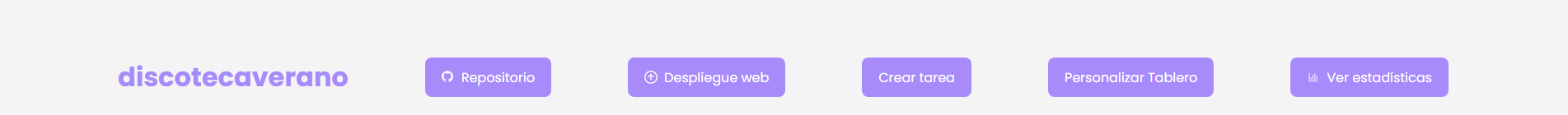
## Kanban

**Pantalla de Tablero Kanban**

La vista de tablero Kanban está diseñada para facilitar la organización visual de tareas mediante **tablones personalizables** y **tareas dinámicas**. Es altamente interactiva y permite una gestión completa del flujo de trabajo de cada proyecto.

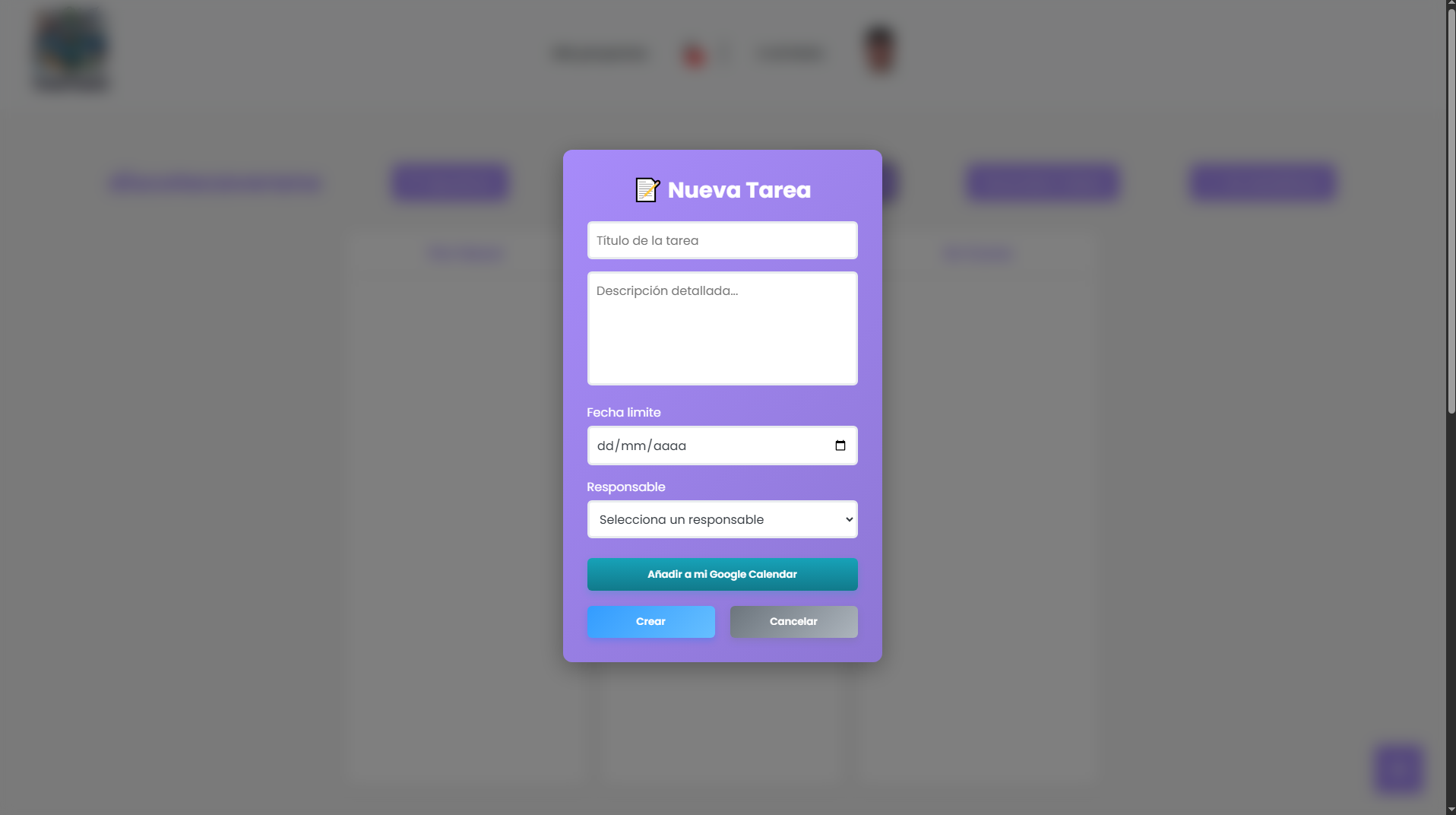
**Creación y Gestión de Tablones**

* ****En la parte superior derecha se encuentra un **botón para crear un nuevo tablón**. Al pulsarlo, se despliega un formulario que solicita el **nombre de la columna**.
* Los tablones pueden **reordenarse mediante arrastrar y soltar** (*drag and drop*) según las preferencias del usuario.
* El **nombre de cada tablón es editable** haciendo clic directamente sobre él, lo que permite cambiar su texto en línea.

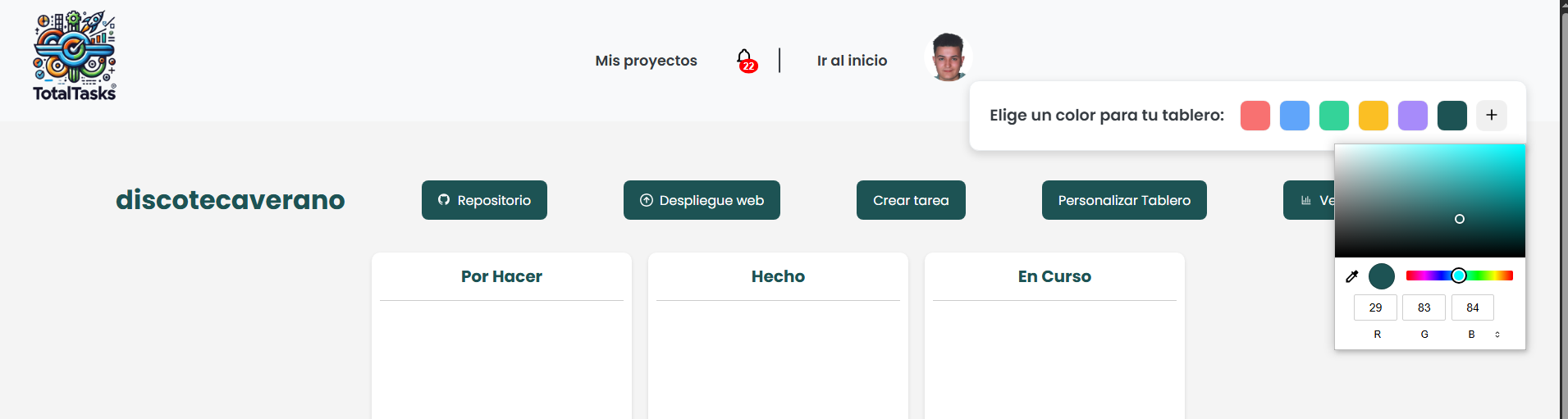
**Menú Superior del Proyecto**

En la parte superior de la pantalla se muestra el **nombre del proyecto**. Si el proyecto fue creado a partir de un repositorio de GitHub, se habilitan funciones adicionales:

* **Ver repositorio**: Redirige al repositorio original en GitHub.
* **Ver despliegue web**: Accede a la URL del despliegue (si está configurado).
* **Generar estadísticas**:
  + Número de *commits*.
  + Lenguajes de programación usados.
  + Actividad por ramas.

**Tareas**

* Las tareas se crean mediante el botón **“Crear tarea”**, disponible en el menú superior.
* Al crear una tarea se solicitan:
  + **Nombre** de la tarea.
  + **Descripción**.
  + **Fecha límite**.
  + **Responsable asignado**.
  + **Permite la sincronización con Google Calendar**.
* Las tareas se pueden:
  + **Mover entre tablones** libremente (*drag and drop*).
  + **Editar** pulsando sobre ellas para abrir su modal correspondiente.
  + **Eliminar**: tanto tareas como tablones se eliminan arrastrándolos hacia un **botón flotante de eliminación** que aparece en la parte inferior. Al eliminar un tablón, **se eliminan todas sus tareas internas**.

**Personalización del Tablero**

La interfaz incluye un botón **“Personalizar tablero”**, el cual abre un **selector de color (color picker)** con dos opciones:

* Elegir entre una serie de **colores predefinidos**.
* Seleccionar y guardar un **color personalizado**, que se aplicará al proyecto y modificará la **paleta de colores global de la aplicación**.

**Vinculación con el Calendario**

La primera vez que se vincula una tarea con una fecha, se abre una **vista de calendario** que permite:

* Visualizar las tareas próximas.
* **Modificar fechas** de tareas mediante *drag and drop*, actualizándose automáticamente en el sistema.

**Chat en Tiempo Real**

En la parte inferior de la pantalla se encuentra una sección dedicada al **chat en tiempo real**, implementado mediante **WebSockets**. Este sistema permite la comunicación instantánea entre los miembros del proyecto sin necesidad de recargar la página.

# Evaluación

Desde el inicio del desarrollo de este proyecto, el objetivo principal fue claro: crear una aplicación web que permitiera a los usuarios **gestionar tareas de manera sencilla, organizada y adaptada** a metodologías como *Kanban* o *Scrum*.

Esta herramienta debía facilitar tanto la **planificación personal** como el **trabajo colaborativo**, especialmente en entornos de equipos de desarrollo o gestión de proyectos. Además, la aplicación debía ser **accesible** desde cualquier lugar y ofrecer una **experiencia visual clara, moderna y útil.**

Al comienzo del diseño técnico, valoramos distintas posibilidades para la arquitectura y las tecnologías. Consideramos el uso de soluciones como *Trello* o *Jira* como referencia de funcionalidades, aunque su grado de complejidad y coste no eran viables para una solución de código propio y totalmente adaptable.

Finalmente, optamos por una solución con backend en *Spring Boot* y *frontend* en *HTML*, *CSS* y *JavaScript*, usando *Thymeleaf* como motor de plantillas y complementos como *FullCalendar* para la **visualización de tareas en calendario.**

Figura 10 - Vista del dashboard de proyectos en TotalTasks.

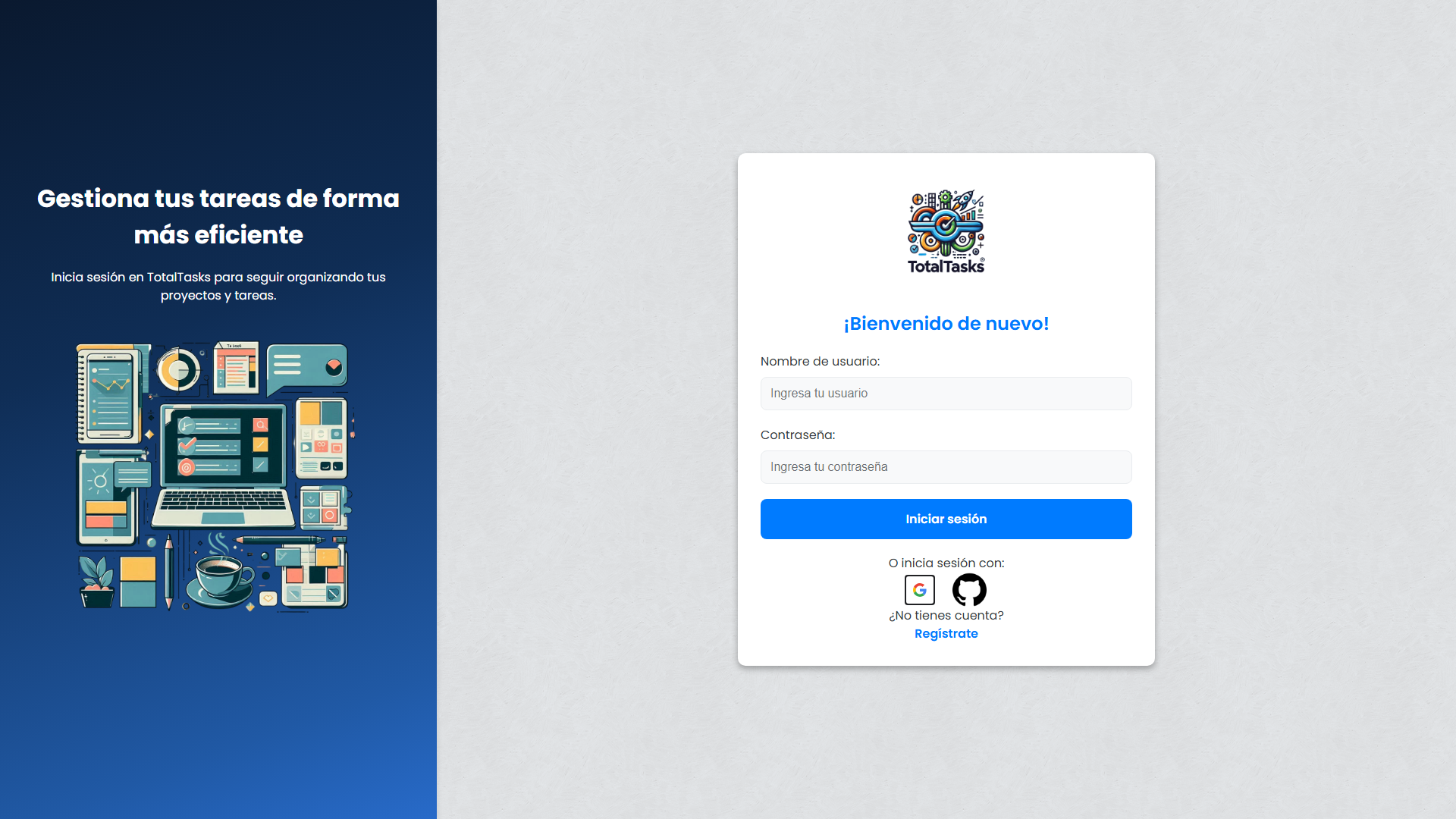
Durante el proceso también se consideraron diferentes alternativas para funcionalidades concretas como la autenticación. Si bien en un principio se pensó en usar solo registro clásico con email y contraseña, se decidió incorporar login con Google y GitHub para facilitar el acceso y ofrecer un entorno más moderno y profesional, similar al que encontraría un usuario en herramientas reales de mercado, además el inicio de sesión con estos entornos nos permite desarrollar funcionalidades más destacables en comparación con otras marcas.

Figura 11 - Opciones de inicio de sesión.

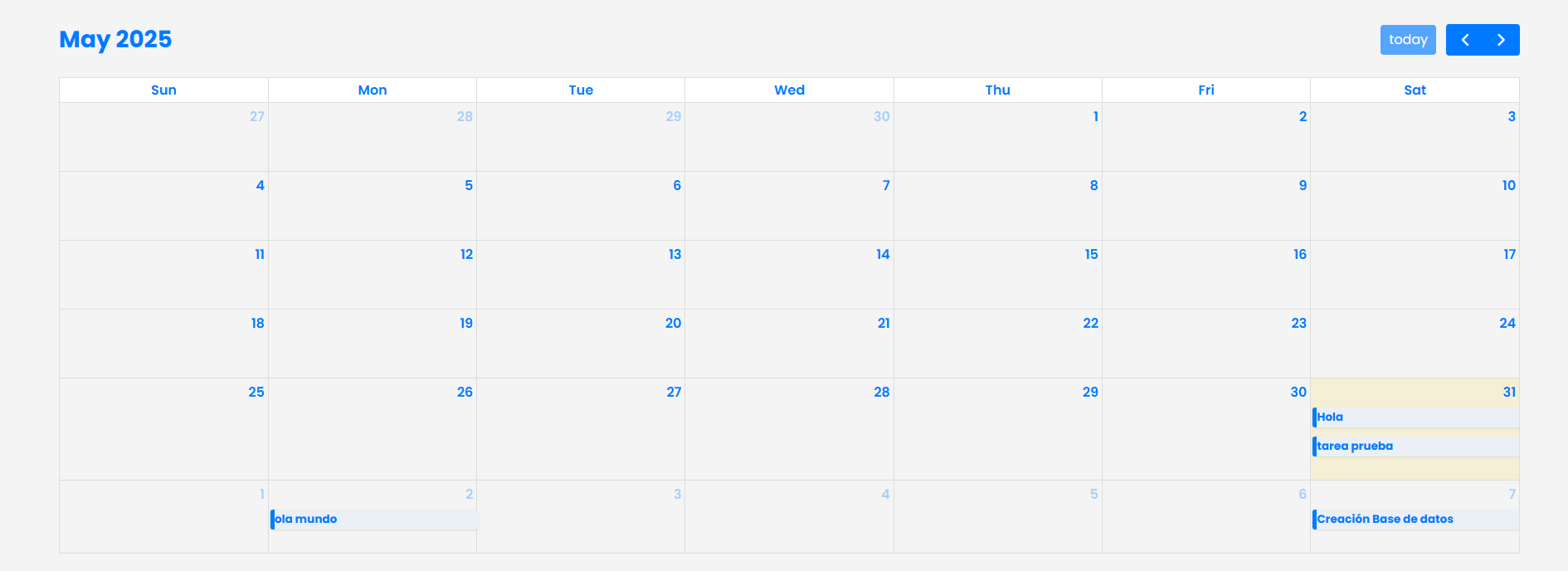
Una de las decisiones clave fue cómo gestionar las tareas y cómo distribuirlas visualmente. Se evaluó si usar tableros estáticos o dinámicos, y finalmente se optó por permitir la creación de tablones personalizados, donde las tareas pueden moverse entre estados mediante drag-and-drop, siguiendo una lógica similar a Kanban. También se implementó integración con Google Calendar para dar más valor a las tareas con fecha límite, permitiendo que los usuarios las añadan directamente a su calendario personal.

Figura 12 - Vista del calendario con tareas sincronizadas.

El mayor reto técnico fue la integración de funcionalidades en tiempo real como el chat interno y la gestión dinámica de tareas. Implementar *Websockets* en *Spring Boot* junto a *JavaScript* no fue trivial, pero permitió lograr una comunicación fluida dentro de los proyectos. Del mismo modo, otra dificultad fue manejar correctamente el control de acceso a proyectos, asegurando que ningún usuario pueda ver o modificar proyectos ajenos.

En lo relativo a los aspectos visuales, se cuidó mucho la experiencia de usuario. Se aplicaron patrones de diseño claros, una paleta de colores suave y elementos visuales personalizados como patrones de fondo, y se trabajó en detalles como las notificaciones y las validaciones en formularios.

Figura 13 - Chat integrado en un proyecto.

# Conclusión

Este proyecto ha resultado especialmente enriquecedor por tres razones principales:

* La oportunidad de desarrollar una aplicación desde cero.
* La posibilidad de aplicar de forma práctica los conocimientos adquiridos durante el grado.
* La satisfacción de haber creado una herramienta real, funcional y útil que podría utilizarse en un entorno profesional o académico.

Durante el desarrollo hemos podido trabajar de manera transversal con contenidos de prácticamente todas las asignaturas del ciclo: acceso a bases de datos, lógica de negocio, servicios web, interfaz gráfica, control de sesiones… integración de *APIs* externas como *Google Calendar* o autenticación *OAuth2* con *Google* y *GitHub*, entre otros. Ha sido un proyecto completo y desafiante, que nos ha permitido consolidar conocimientos y adquirir nuevos.

El objetivo principal era crear una aplicación de **gestión de proyectos** y **tareas colaborativa** se ha cumplido satisfactoriamente. Además, se han logrado metas secundarias relevantes y la implementación de funcionalidades avanzadas como el sistema de notificaciones, los **tablones Kanban personalizables** o la **sincronización con calendarios externos**.

A pesar de ello, hay aspectos que podrían mejorarse o extenderse en el futuro. Sería interesante optimizar el rendimiento en dispositivos móviles, ampliar las opciones de personalización visual por parte del usuario, o incluso explorar posibilidades de monetización mediante suscripciones o versiones premium.

|  |  |
| --- | --- |
| **Objetivo** | **Resultado** |
| **Primario:** Aplicación web de gestión de tareas y proyectos | ✅ CONSEGUIDO |
| **Secundario:** Gestión colaborativa en tiempo real con notificaciones | ✅ CONSEGUIDO |
| **Secundario:** Integración con Google Calendar | ✅ CONSEGUIDO |
| **Secundario:** Control de acceso y roles por proyecto | ✅ CONSEGUIDO |
| **Secundario:** Diseño adaptativo y experiencia de usuario | ✅ CONSEGUIDO |
| **Secundario:** Despliegue online y accesibilidad desde navegador | ⏳ PENDIENTE |
| **Secundario:** Monetización (suscripciones, publicidad, etc.) | ⏳ PENDIENTE |

# Referencias

<https://www.youtube.com/watch?v=xAaGxhDiGg8>

<https://www.youtube.com/watch?v=6TT5kQrS858>

<https://www.youtube.com/watch?v=RI3UpqUZq50>

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

<https://docs.github.com/en/apps/oauth-apps/building-oauth-apps/scopes-for-oauth-apps>

<https://codepen.io/leedeeya/pen/YPzEwdQ>

<https://codepen.io/RicardoYare/pen/LEYzxqY>

<https://codepen.io/shivani0805/pen/NWQPBQN>

<https://codepen.io/tommyho/pen/LYwYQXz>

<https://codepen.io/nilimadas/pen/gOVpqaL>

<https://codepen.io/jerora98/pen/MWNaYgb>

<https://codepen.io/Ranjith-R-the-builder/pen/WNVbEOO>

<https://www.baeldung.com/websockets-spring>

<https://developer.mozilla.org/es/docs/Web/API/WebSocket>

<https://serverjs.io/tutorials/chat/>

<https://sortablejs.github.io/Sortable/>

<https://uiverse.io/vinodjangid07/thin-duck-22>

<https://uiverse.io/svqantonio/great-moose-35>

<https://chatgpt.com/canvas/shared/6807d812307481918895fd2b81f863d6>

<https://codepen.io/SabAsan/pen/LYNgBoB>

<https://www.atlassian.com/es/software/jira/templates/scrum>