

Taskboard – Visión general de la aplicación

Taskboard es una aplicación web para organizar el trabajo de un equipo en **workspaces, proyectos y tareas**. La idea es tener una vista sencilla pero potente para ver qué hay que hacer, quién está con qué y qué ha pasado recientemente en el proyecto.

¿Qué problema resuelve?

- Permite **centralizar las tareas** de un proyecto en un solo lugar.
 - Da **contexto rápido**: responsable, prioridad, esfuerzo, fechas, subtareas y comentarios.
 - Facilita hacer **seguimiento de la actividad** (historial) y de las **menciones** entre miembros.
 - Incluye un panel de usuario para ajustar perfil, seguridad y preferencias de notificaciones.
-

Funcionalidades principales

1. Workspaces y proyectos

- Puedes trabajar con uno o varios **workspaces** (espacios de trabajo).
- Dentro de cada workspace hay **proyectos**.
- Desde el panel principal eliges el workspace y el proyecto activos; a partir de ahí todo el contenido se filtra a ese contexto.

2. Gestión de tareas

La parte central de la app es la gestión de tareas del proyecto:

- **Lista de tareas / panel principal** (`TasksManagementPanel`, `TaskList`):
 - Muestra las tareas del proyecto actual.
 - Si no hay workspace o proyecto seleccionado, se explican los pasos para empezar (seleccionar workspace, crear proyecto, etc.).
- **Vista tablón** (`TaskSectionsBoard`):
 - Muestra las tareas agrupadas en secciones (por fechas o por epic/grupo).
 - Cada tarjeta de tarea enseña título, descripción corta, responsable, fecha límite y actividad reciente.
 - Permite marcar una tarea como completada desde el propio tablón.
 - Botón rápido “+ Agregar tarea” para crear nuevas tareas en el contexto actual.
- **Panel de detalle de tarea** (`TaskDetailPanel`):
 - Se abre al seleccionar una tarea concreta.
 - Muestra:
 - Estado (pendiente / completada).
 - Prioridad (alta, media, baja) y esfuerzo (S/M/L).
 - Responsable actual y opción para cambiarlo.
 - Epic/grupo de la tarea.

- Etiquetas (tags) con alta/baja rápida.
- Cronología: creada, fecha límite, completada.
- Incluye **subtareas**:
 - Crear, marcar como completadas, eliminar y refrescar subtareas de una tarea.
- Muestra **actividad reciente** y **último comentario** asociados a la tarea.
- Integra el componente de **comentarios** ([TaskComments](#)) para conversar alrededor de la tarea.
- Indica si hay **otros usuarios viendo la misma tarea en tiempo real** (presencia vía Supabase).

3. Comentarios, menciones y actividad

- **Comentarios de tarea** ([TaskComments](#)):
 - Permiten discutir detalles de la tarea dentro de la propia app.
 - Soportan menciones a otros miembros del proyecto.
- **Resumen de menciones** ([MentionDigest](#)):
 - Muestra notificaciones donde se ha mencionado a usuarios en comentarios.
 - Agrupa las menciones por usuario y enseña el comentario más reciente, la tarea y un pequeño extracto.
- **Historial de actividad** ([ActivityLog](#)):
 - Lista las acciones recientes de un proyecto (tareas creadas, completadas, reabiertas, cambios de asignación, comentarios, etc.).
 - Se puede filtrar por:
 - Rango de fechas (presets tipo “últimos 7/14/30 días” o fechas personalizadas).
 - Usuario actor.
 - Tipo de acción.
 - Se actualiza en tiempo casi real usando canales de Supabase.

4. Panel de usuario y cuenta ([UserPanel](#))

- Muestra datos básicos del usuario: nombre, email, rol, organización, avatar, etc.
- Permite:
 - **Editar perfil** (nombre, compañía, bio y avatar con subida a Supabase Storage).
 - Ajustar **seguridad y contraseñas** (incluye flujo de cambio de contraseña y reautenticación).
 - Configurar **notificaciones** (resúmenes, noticias del producto, etc.).
- También enseña estadísticas de uso (proyectos, tareas, completadas, colaboradores), actividad reciente y novedades del producto.
- Incluye el botón de **cerrar sesión**.

5. Layout y navegación ([AppLayout](#))

- Componente de layout general que aplica el theme, el fondo y la estructura de la página.
- Incluye:
 - Barra lateral con navegación principal y acciones rápidas.
 - Breadcrumbs (migas de pan) para ubicarse dentro de la app.
 - Tarjetas de estado/resumen en la parte superior.
 - Contenedor principal para el contenido de cada pantalla.

Flujo de uso típico

1. **Iniciar sesión** con tu cuenta (gestionado a través de Supabase Auth).
 2. **Seleccionar workspace y proyecto** desde la navegación principal.
 3. **Ver las tareas** del proyecto:
 - En modo lista o tablón según la configuración.
 4. **Crear o editar tareas**:
 - Añadir título, descripción, prioridad, esfuerzo, responsable, epic y etiquetas.
 - Dividir trabajo en subtareas cuando haga falta.
 5. **Comunicarse con el equipo**:
 - Escribir comentarios en las tareas y mencionar a otros miembros.
 - Revisar el resumen de menciones ([MentionDigest](#)).
 6. **Seguir el progreso**:
 - Revisar el historial de actividad ([ActivityLog](#)).
 - Ver estadísticas personales en el panel de usuario.
-

Tecnologías y stack

- **Front-end**:
 - [React](#) (Create React App) como framework de UI.
 - Componentes de UI con [Flowbite React](#) y [Flowbite](#).
 - Estilos con [Tailwind CSS](#).
 - **Backend / BaaS**:
 - [Supabase](#) para:
 - Base de datos y tablas (tareas, activity_log, notifications, etc.).
 - Autenticación de usuarios.
 - Almacenamiento de archivos (avatars).
 - Canales en tiempo real para actividad y presencia.
 - **Tooling**:
 - Create React App con [react-scripts](#).
 - ESLint integrado en CRA.
 - Testing con [@testing-library/react](#) y utilidades relacionadas.
-

Scripts y entorno

En [package.json](#) tienes los scripts estándar de CRA:

- `npm start` – Arranca la app en modo desarrollo.
- `npm run build` – Genera el build de producción.
- `npm test` – Ejecuta los tests.

Variables de entorno importantes (prefijo `REACT_APP_`):

- `REACT_APP_SUPABASE_URL` – URL del proyecto Supabase.
- `REACT_APP_SUPABASE_ANON_KEY` – Clave pública (anon) de Supabase.

Estas se leen en `src/supabaseClient.js` para crear el cliente compartido de Supabase.

Notas finales

Taskboard está pensada para ser **claramente visual**, con un diseño oscuro moderno y componentes accesibles. La documentación de este archivo es un resumen general: el código fuente (sobre todo en `src/components/`) es la mejor referencia para ver el detalle de cada funcionalidad.