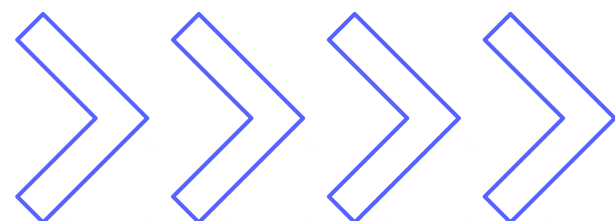


TASKY RD

**Aplicación Web y movil para crear,
editar, completar y eliminar tareas**





Descripción

Tasky RD será una aplicación web que permitirá a equipos u organizaciones crear, asignar, editar, completar y eliminar tareas dentro de un entorno visual intuitivo. Contará con una interfaz gráfica tipo tablero (estilo Kanban) donde los usuarios podrán visualizar el estado de cada tarea, arrastrarla entre columnas, añadir comentarios, adjuntar archivos y marcar su progreso.

El sistema integrará un backend para el manejo de usuarios, autenticación, almacenamiento de tareas y permisos, mientras que el frontend se encargará de ofrecer una experiencia visual dinámica mediante componentes GUI interactivos.

La aplicación será accesible desde navegadores web y tendrá diseño responsivo para uso en dispositivos móviles y de escritorio.



Objetivos



Colaboración

Gestionar tareas colaborativamente mediante un tablero Kanban que permita crear, asignar y completar tareas.



Usabilidad

Ofrecer una experiencia intuitiva y responsiva accesible desde dispositivos móviles y de escritorio.



Seguridad

Proteger la información del usuario mediante autenticación, roles y almacenamiento seguro de datos.



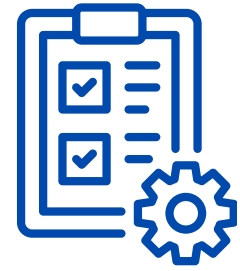
Entorno de desarrollo

VS Code

Node.js

ReactJS

PostgreSQL



Requerimientos Funcionales del GUI

◆ Pantalla de Inicio de Sesión

- **Formulario con correo y contraseña.**
- **Botones: Iniciar sesión, Registrarse, Recuperar contraseña.**

◆ Tablero Kanban

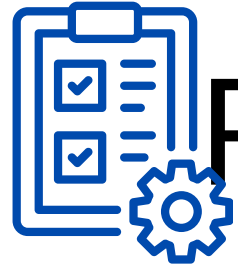
- **Columnas: Pendiente, En Progreso, Completada.**
- **Tarjetas mostrando título, descripción, vencimiento, responsable y prioridad.**
- **Función drag & drop.**

◆ Gestión de Tareas

- **Crear, editar, eliminar tareas.**
- **Adjuntar archivos y agregar comentarios.**
- **Formulario con campos completos.**

◆ Responsividad y Notificaciones

- **Diseño adaptable a móvil, tablet y PC.**
- **Alertas visuales no intrusivas.**



Requerimientos No Funcionales y Casos de Uso

◆ Requerimientos No Funcionales

- **Usabilidad:** Interfaz simple, coherente e intuitiva.
- **Rendimiento:** Carga menor a 2 segundos.
- **Accesibilidad:** Cumplir WCAG AA.
- **Seguridad:** Sanitización de formularios, no mostrar datos sensibles.

◆ Casos de Uso Clave

- **CU01:** Crear tarea.
- **CU02:** Mover tarea en el tablero.
- **CU03:** Comentar tarea.
- **CU04:** Editar tarea.

◆ Requerimientos Técnicos

- **ReactJS** para la GUI.
- **API REST** con Node.js.
- **Datos estructurados** para PostgreSQL.



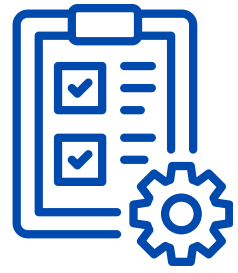
Análisis de requerimientos técnicos de plataforma para diseño GUI

Plataforma de ejecución de la GUI

- **Tipo de aplicación:** Aplicación web.
- **Navegadores objetivo:**
 - **Soporte moderno:** Chrome, Edge, Firefox y Safari en sus versiones recientes.
- **Dispositivos objetivo:**
 - Prioridad desktop-first (pantallas de laptop y PC), asegurando también adaptación correcta a tablets y móviles.

Requerimientos de software para la GUI

- **Framework de interfaz:**
 - ReactJS como framework principal para la GUI, aprovechando componentes reutilizables y un diseño modular.
- **Lenguajes y tecnologías:**
 - HTML5 y CSS3 (con posible uso de framework de estilos como TailwindCSS o CSS Modules).
 - JavaScript/TypeScript para la lógica de la interfaz.
- **Backend y API:**
 - Node.js con API REST que expone servicios para autenticación, tareas, comentarios, archivos, etc. La GUI consume estos endpoints vía HTTP/JSON.
- **Base de datos:**
 - PostgreSQL, que almacena usuarios, tareas, estados, comentarios y metadatos de archivos. La GUI debe estar alineada con el modelo de datos (campos disponibles, restricciones, tipos).
- **Herramientas de desarrollo:**
 - VS Code como IDE principal para diseño y codificación de la interfaz, integrando plugins para React, linting, formato de código y control de versiones.



Requerimientos técnicos específicos del GUI

A partir de los requerimientos funcionales y no funcionales del documento:

- **Pantallas clave:**

- Inicio de sesión (login/registro/recuperar contraseña).
- Tablero Kanban con columnas Pendiente, En Progreso, Completada.
- Formularios de creación/edición de tareas.
- Sección de comentarios y adjuntos.

- **Interacción y componentes gráficos:**

- Soporte para drag & drop en las tarjetas del tablero.
- Tarjetas con información: título, descripción, fecha de vencimiento, responsable y prioridad.
- Notificaciones visuales no intrusivas (toasts, banners, badges).

- **Requerimientos no funcionales relacionados a la GUI:**

- Usabilidad: interfaz simple, coherente e intuitiva para usuarios de nivel tecnológico principiante–intermedio.
- Rendimiento: tiempo de carga del frontend menor a 2 segundos en condiciones normales.
- Accesibilidad: cumplir al menos con WCAG AA en contraste de colores, tamaños de fuente y navegación por teclado.
- Seguridad en la GUI:
 - Sanitización de entradas en formularios.
 - No mostrar datos sensibles en la interfaz (por ejemplo, contraseñas o tokens).

- **Requerimientos de red e infraestructura:**

- Conexión estable a internet para comunicación con la API REST.
- Servidor de aplicaciones capaz de servir la SPA de React y la API Node.js con tiempos de respuesta bajos.
- Sistema de almacenamiento para archivos adjuntos (puede ser disco del servidor o servicio cloud).



Análisis de requerimientos de personal para diseño GUI

PERFILES PRINCIPALES

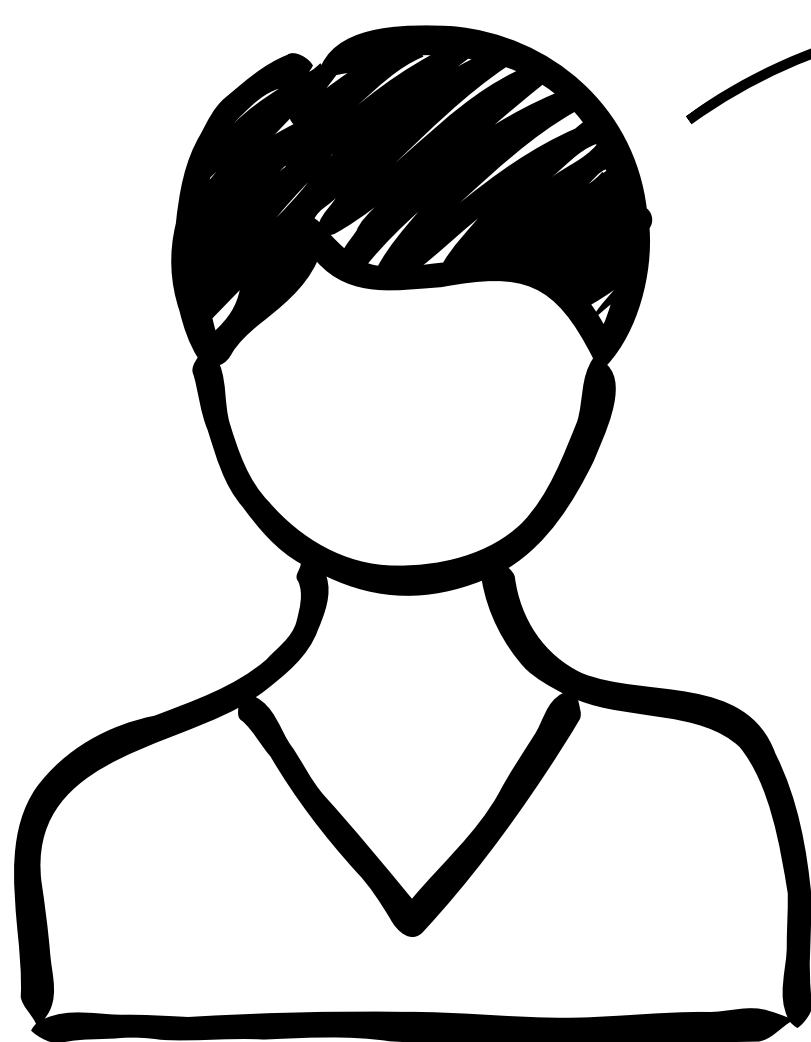
1. Analista de requerimientos / Product Owner
2. UX Researcher / Analista de usabilidad
3. Diseñador UI (Diseño visual de la GUI)
4. Desarrollador Frontend (ReactJS)
 - Implementa la GUI usando ReactJS, integrando:
 - Consume la API REST del backend Node.js.
 - Se asegura de que el diseño sea responsivo (mobile/desktop).
5. Desarrollador Backend / Integración
6. QA / Tester de usabilidad y funcional
 - Diseña casos de prueba para
 - Ejecuta pruebas con usuarios reales para validar que la GUI sea entendible para el perfil de Tasky RD.
7. DevOps / Administrador de plataforma (opcional según el tamaño del equipo)
 - Automatiza despliegues de frontend y backend.
 - Configura entornos (desarrollo, pruebas, producción).
 - Supervisa rendimiento y disponibilidad de la aplicación.

RELACIÓN ENTRE ROLES Y ENTREGABLES DE GUI

- Analista + UX → historias de usuario, flujos de navegación, escenarios de uso.
- UX + UI → wireframes, prototipos de alta fidelidad, guía de estilos.
- UI + Frontend → implementación de componentes React fieles al diseño.
- Frontend + Backend → integración funcional, manejo de errores y mensajes claros en la interfaz.
- QA → reporte de bugs, problemas de usabilidad y propuestas de mejora.

Análisis de usabilidad del software para el diseño GUI de Tasky RD

Perfil de usuario

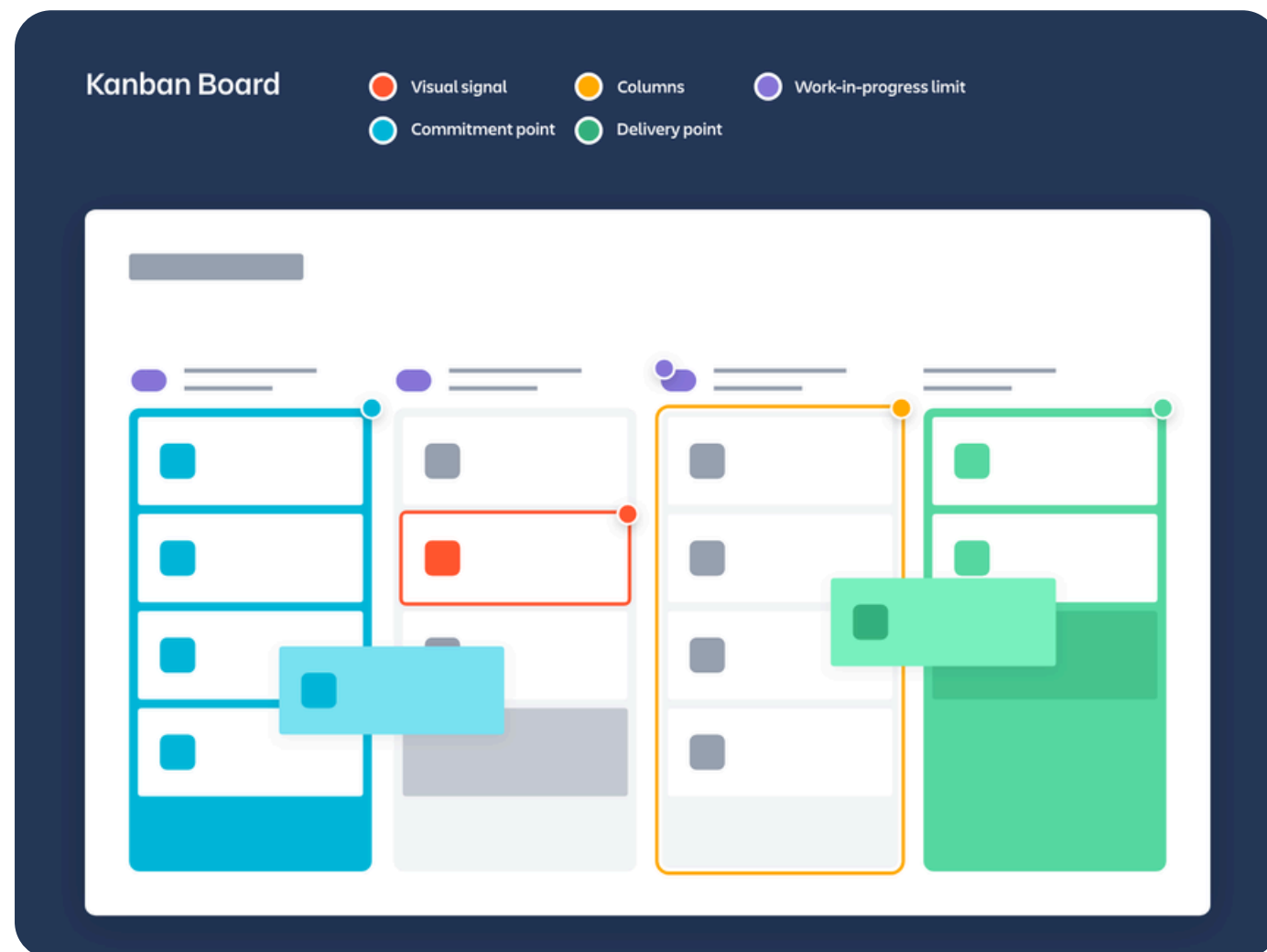


Tasky RD está dirigido a personas entre 20 y 50 años, con un nivel tecnológico de **principiante** a **intermedio**.

Estudiante, profesional, ejecutivo



Contexto de uso



La aplicación se utilizará principalmente en **computadoras**, por lo que se prioriza un diseño **desktop-first** que aproveche el espacio en pantalla.



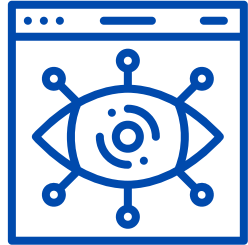
Objetivo principal del usuario

Consultar
tareas

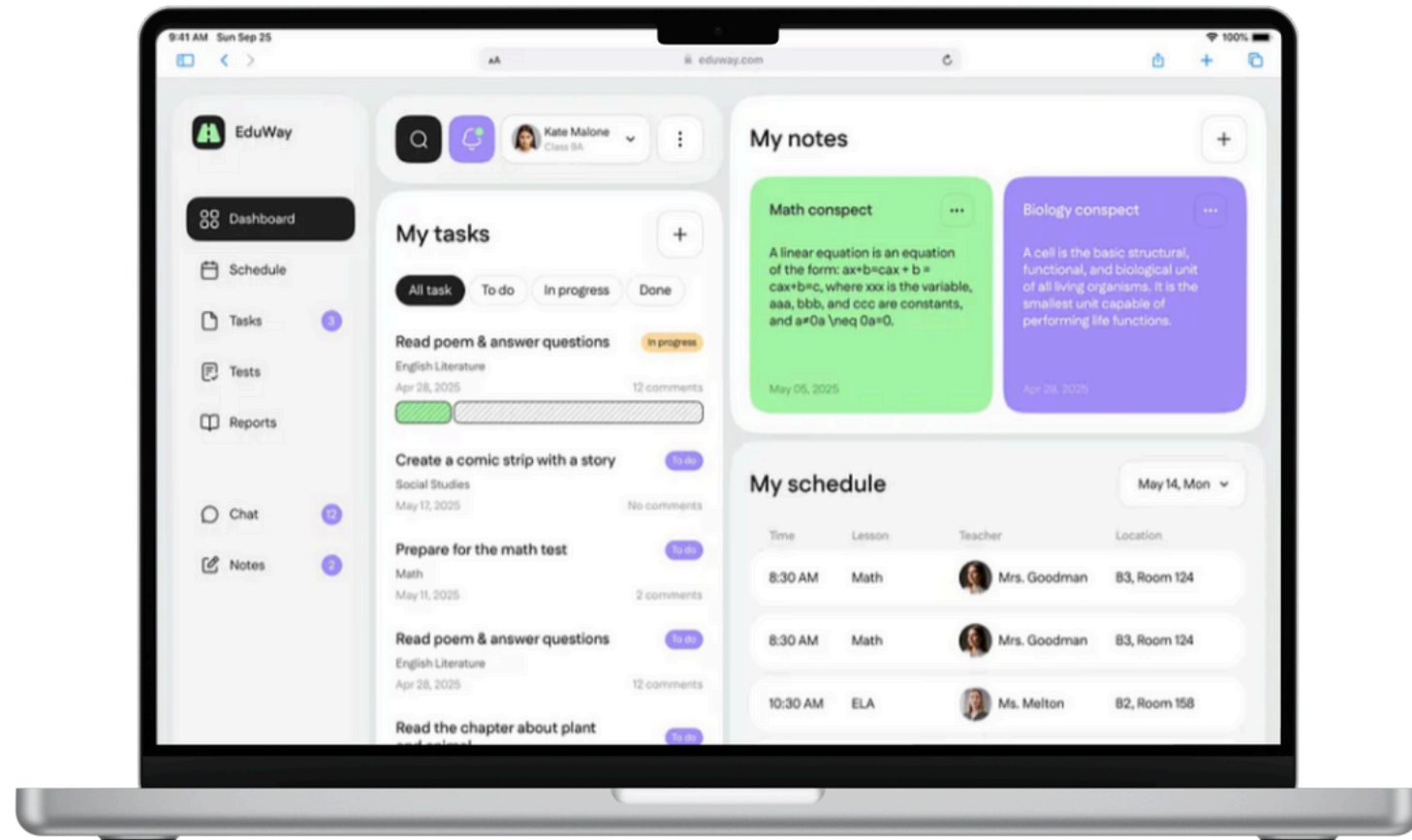
Identificar
prioridades

Ver fechas

Distinguir
tareas



Estilo visual



Tasky RD adoptará un estilo **moderno**, **profesional** y **corporativo**, utilizando colores pasteles.

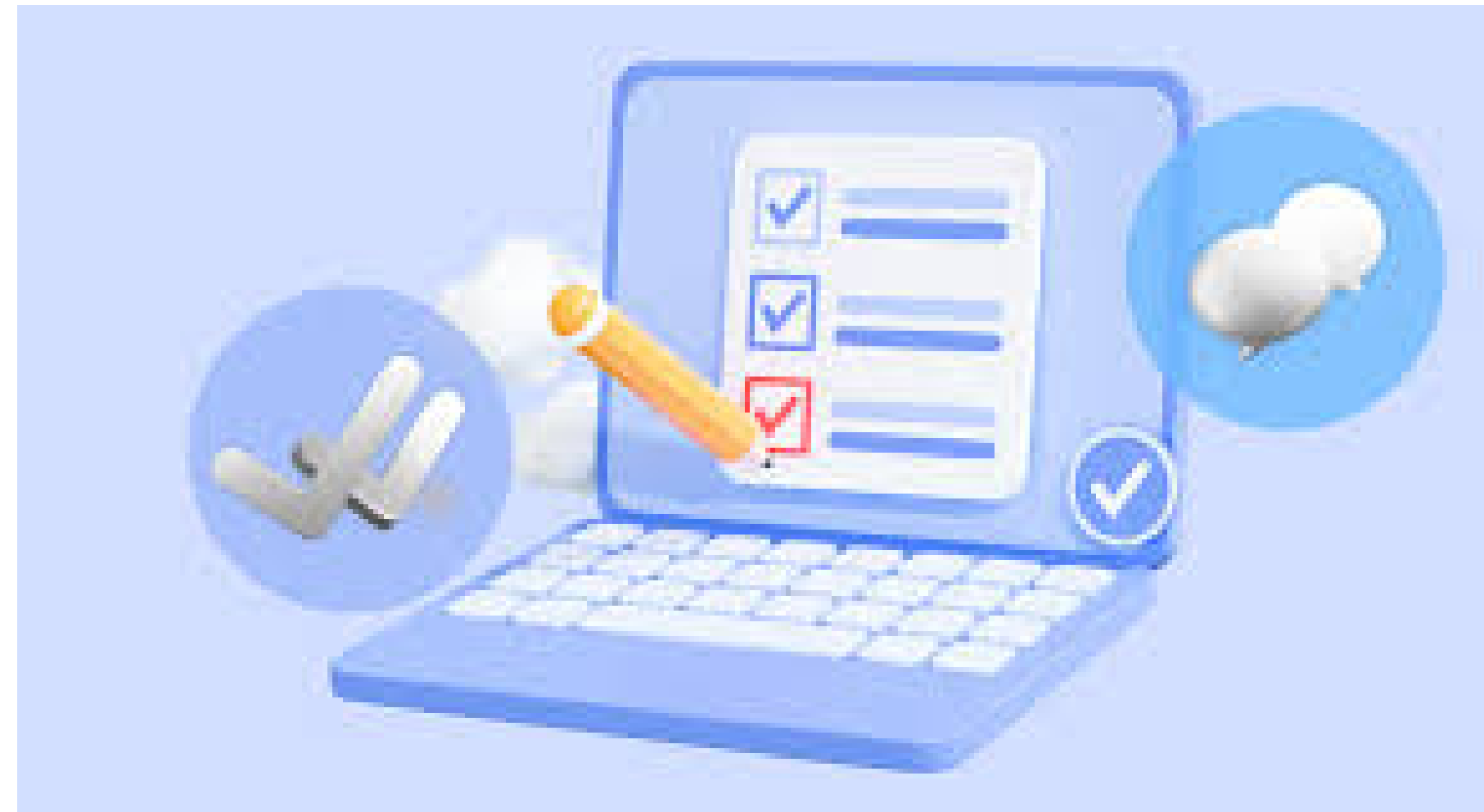


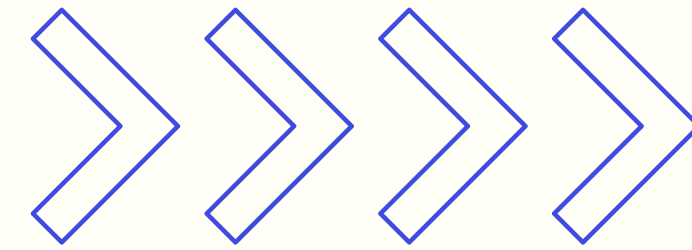
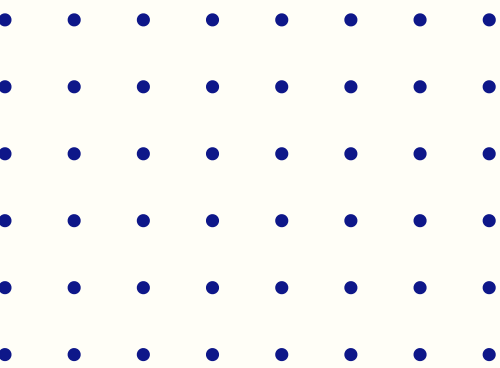
Análisis de Soportabilidad del Software para el Diseño GUI – Tasky RD

◆ Soportabilidad basada en la arquitectura y tecnologías utilizadas

ReactJS (Frontend) – Alta soportabilidad

- Claridad en la separación Frontend–Backend
- Escalabilidad y organización del almacenamiento
- Diseño GUI modular y reutilizable
- Soportabilidad en el diseño responsivo
- Soportabilidad en seguridad y manejo de permisos
- Uso de VS Code como entorno de desarrollo





MUCHAS

Gracias

