# Informe Inicial - Avance 1

## Capítulo 1: Problema y Propuesta de Solución

En muchas empresas y organizaciones, la gestión de tareas diarias se realiza de forma descentralizada, mediante hojas de cálculo, correos electrónicos o canales de mensajería dispersos. Esta fragmentación dificulta el control del avance, la trazabilidad de las acciones y la coordinación entre las diferentes áreas de trabajo. Supervisores y trabajadores suelen carecer de una herramienta centralizada que les permita planificar, asignar y dar seguimiento a las actividades de manera visual y estructurada.

Propuesta:  
Desarrollar una aplicación web tipo Kanban para la gestión de tareas empresariales, organizada por áreas, supervisores y trabajadores. El sistema permitirá crear tableros de tareas por área, asignar responsables, registrar avances y mantener un historial de cambios y acciones inspirado en las buenas prácticas de Git (como versiones o estados de tareas). De este modo, se busca mejorar la organización interna, la comunicación y la transparencia en el seguimiento de las actividades cotidianas dentro de la empresa.

## Capítulo 2: Objetivos SMART

Objetivo General

Gestionar mediante una aplicación web tipo Kanban la administración centralizada de tareas en empresas, organizada por áreas, supervisores y trabajadores, que permita asignar, registrar y dar seguimiento a actividades cotidianas con trazabilidad de acciones inspirada en Git.

Objetivos Específicos

|  |  |
| --- | --- |
| S (Específico) | Crear una app web Kanban empresarial con jerarquía de usuarios. |
| M (Medible) | Entregar del sistema funcional y documentado. |
| A (Alcanzable) | Programar con Node.js, Express y MySQL2. |
| R (Relevante) | Mejorar la organización, comunicación y control interno. |
| T (Tiempo) | Entregar un plazo de 12 semanas. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | S | M | A | R | T |
| Específico 1 | Levantar y documentar los requerimientos funcionales y no funcionales del sistema, definiendo roles y flujos de trabajo. | | | | |
| Definir requerimientos y estructura de roles. | Documento de requerimientos validado. | Factible mediante entrevistas y análisis interno. | Es la base para el diseño del sistema. | Durante las 2 primeras semanas. |
| Específico 2 | Diseñar la arquitectura del sistema y el modelo de base de datos relacional. | | | | |
| Elaborar diagramas UML/DFD y modelo ER. | Entregables técnicos revisados. | Realizable con herramientas de modelado. | Permite una implementación estructurada. | En la semana 3. |
| Específico 3 | Implementar una API REST en Node.js con Express y MySQL2 para la gestión de usuarios, áreas y tareas. | | | | |
| Desarrollar endpoints funcionales. | Verificación mediante pruebas de API. | Totalmente alcanzable con tu stack actual. | Proporciona la lógica principal del sistema. | En un periodo de 4 semanas. |
| Específico 4 | Desarrollar la interfaz web responsiva con tablero Kanban integrado al backend. | | | | |
| Construir la UI de interacción con tareas. | Funcionalidad validada mediante pruebas de usuario. | Realizable con frameworks front-end o HTML/CSS. | Es la interfaz visual principal del sistema. | En 3 semanas. |
| Específico 5 | Integrar el registro de acciones e historial de cambios inspirado en Git. | | | | |
| Implementar módulo de trazabilidad. | Validación mediante logs y visualización de historial. | Viable mediante registro de acciones en BD. | Aporta innovación y valor diferencial. | En 2 semanas. |
| Específico 6 | Realizar pruebas funcionales, documentación técnica y despliegue básico del sistema. | | | | |
| Probar y documentar todo el proyecto. | Entrega de informe y sistema operativo. | Alcanzable con herramientas de testing y documentación. | Asegura la calidad y cierre del proyecto. | En la última semana (semana 12). |

## Capítulo 3: Elementos de Innovación

El proyecto propone una innovación funcional al trasladar principios de control de versiones y trazabilidad, propios de entornos de desarrollo de software (como Git), al ámbito de la gestión de tareas empresariales. En lugar de centrarse en el control de código fuente, la aplicación aplica el concepto de historial de cambios y registro de acciones para las tareas cotidianas de los trabajadores. Esto permite que cada tarea cuente con un seguimiento estructurado y transparente, facilitando la comunicación entre supervisores y empleados.  
  
Además, el sistema se distingue por su estructura jerárquica basada en áreas, supervisores y trabajadores, lo que permite adaptar la plataforma a diferentes tipos de organizaciones. Esta combinación de un tablero Kanban con trazabilidad inspirada en Git representa una propuesta innovadora dentro del ámbito de la gestión de procesos internos.

## Capítulo 4: Reflexión - La Catedral y el Bazar

La reflexión 'La Catedral y el Bazar' plantea dos modelos de desarrollo de software: la 'Catedral', caracterizada por la planificación centralizada y el control jerárquico, y el 'Bazar', basado en la colaboración abierta y la mejora continua por parte de múltiples participantes.  
  
El enfoque de este proyecto combina elementos de ambos modelos. Desde la perspectiva de la 'Catedral', se sigue una planificación estructurada y una jerarquía de roles (áreas, supervisores, trabajadores), que asegura control y claridad en las responsabilidades. Sin embargo, también incorpora la filosofía del 'Bazar', ya que fomenta la colaboración, la comunicación abierta y la mejora continua mediante el uso de tableros Kanban y el registro transparente de acciones.  
  
En síntesis, el proyecto busca equilibrar la estructura organizacional (Catedral) con la flexibilidad y participación activa (Bazar), logrando así un entorno digital más colaborativo dentro de la empresa.

## Capítulo 5: Licencia del Proyecto GPL

El proyecto se distribuirá bajo la Licencia Pública General de GNU (GPL v3), una de las licencias más reconocidas del software libre. Esta licencia garantiza que el código fuente del programa permanezca abierto y disponible para todos los usuarios, quienes podrán usar, estudiar, modificar y redistribuir el software, siempre y cuando mantengan los mismos términos de libertad en las versiones derivadas.  
  
La elección de la licencia GPL se justifica porque promueve la colaboración, la transparencia y el desarrollo comunitario, asegurando que cualquier mejora o adaptación del sistema también se comparta bajo la misma licencia.

De esta manera, el proyecto se alinea con los valores del software libre y con la filosofía de apertura y mejora continua que inspiran su desarrollo.

## Capítulo 6: Diagrama de Procesos / DFD

El siguiente diagrama representa de manera general el flujo de procesos del sistema Kanban empresarial. En esta primera versión, el modelo describe cómo las tareas son creadas, asignadas y actualizadas dentro del flujo de trabajo jerárquico:  
  
1. El \*\*Supervisor de Área\*\* crea y organiza tableros de tareas para su equipo.  
2. Cada \*\*Trabajador\*\* visualiza las tareas asignadas y actualiza su estado (To Do, In Progress, Done).  
3. El sistema registra cada cambio en una base de datos y actualiza el historial de acciones.  
4. Los \*\*Administradores o Supervisores\*\* pueden visualizar reportes de avance y trazabilidad.  
  
Este flujo podrá ser posteriormente modelado formalmente mediante un diagrama BPMN o DFD detallado con herramientas como Draw.io, Lucidchart o Bizagi.