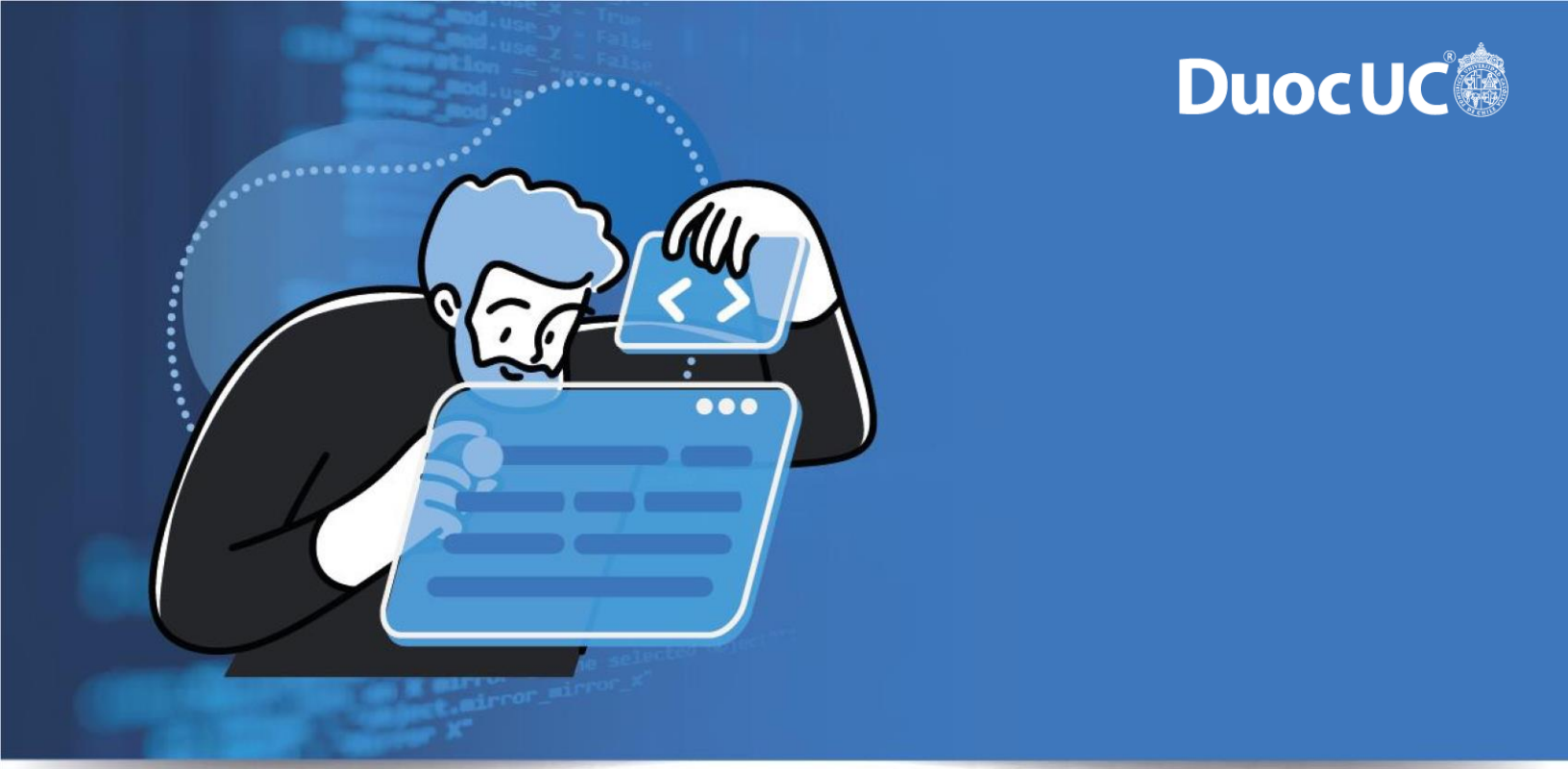


Carrera

Analista Programador Computacional

Ingeniería de Software (PRY3211)

Formato de respuesta

Nombre estudiante: Iván Felipe Valdivia Ríos	
Asignatura: Ingeniería en Software	**Carrera: Analista Programador**
Profesor: Jorge Canales	**Fecha: 12-10-2025**

Documento Proyect Burndown (versión final)

KaiTasks

Descripción de la metodología de trabajo (Scrum)

Versión 2.0

Historial de Revisiones

Fecha	Versión	Descripción	Autores
05/08/2025	0.1	Borrador inicial con la definición preliminar de la Visión del Proyecto, objetivos estratégicos y estructura de apartados a desarrollar.	Equipo KaiTasks
12/08/2025	0.5	Ampliación de requerimientos: identificación de necesidades funcionales (gestión de tareas, validación, notificaciones) y no funcionales (seguridad, rendimiento).	Equipo de Análisis
20/08/2025	0.8	Definición de arquitectura técnica preliminar, selección de tecnologías (Angular, Node.js, PostgreSQL, AWS). Inclusión de herramientas de gestión (Trello, Figma).	Equipo de Arquitectura
28/08/2025	0.9	Revisión con stakeholders: ajustes en alcance, mejoras en requisitos de filtros y reportes, incorporación de sugerencias para el panel de control.	Product Owner + Stakeholders
01/09/2025	1.0	Versión consolidada: documento formalizado con Visión del Proyecto, Product Backlog inicial, ampliación a Sprint Backlog, esquema visual de Roadmap y profundización de Tecnologías y Herramientas.	Equipo KaiTasks
12/10/2025	2.0	Versión final con aplicación funcional y documentación completa.	Equipo KaiTasks

Tabla de Contenidos

Contenido

1. Introducción 11
1.1.1 Propósito de este documento 11
1.1.2 Problemática que resolver 11
1.1.3 Objetivo del Proyecto 12
1.1.4 Alcances 12
2. Descripción General de la Metodología ágil a adoptar. 14
2.1. Fundamentación 14
2.2. Valores de trabajo 14
Personas y roles del proyecto. 15
3. Product Backlog. Lista de Componentes y artefactos para Construir. 16
3.1. Épicas e historias de usuarios. 16
4. Definición tecnologías de Desarrollo a utilizar 7
4.1. Criterios generales de “Done”: 7
5. Definición de tecnologías de implementación. 8
6. Roadmap 10
7. Producto BackLog Final 12
8. Resumen de Sprints 13
9. Burndown Chart 7
10. Retrospectiva Final 7

1. Introducción

Este documento describe la implementación de la metodología de trabajo Scrum para el desarrollo del proyecto **KaiTasks**, una aplicación orientada a la gestión organizacional de tareas bajo un enfoque ágil.

El sistema busca mejorar la coordinación entre diferentes áreas de una empresa (jefaturas, finanzas, logística, operaciones), permitiendo que los colaboradores creen, asignen, validen y den seguimiento a tareas en tiempo real.

La propuesta integra buenas prácticas de Kaizen (mejora continua), Lean Management (eficiencia en procesos) y Scrum (planificación iterativa e incremental).

1.1.1 Propósito de este documento

El propósito de este documento es poder facilitar la información de referencia necesaria a las personas involucradas en el desarrollo del desarrollo de la app, definiendo visión, roles, backlog y plan de trabajo en modalidad ágil.

Esto para establecer la visión inicial del producto y definir un marco de referencia para su desarrollo:

- Describir la problemática a resolver.
- Identificar roles y responsabilidades.
- Definir las épicas e historias de usuario iniciales.
- Presentar una planificación ágil con roadmap y backlog.
- Establecer las tecnologías y herramientas de soporte.

1.1.2 Problemática que resolver

En muchas organizaciones, la gestión de tareas se realiza en correos electrónicos, planillas dispersas o reuniones presenciales, lo que provoca:

- Retrasos en la entrega de actividades.
- Falta de visibilidad de qué tareas están en progreso o atrasadas.
- Ausencia de validación estructurada por parte de las jefaturas.
- Dificultad para coordinar trabajo entre áreas como finanzas, logística y operaciones.

La consecuencia es una baja productividad y pérdida de eficiencia organizacional. **KaiTasks** busca centralizar y agilizar la gestión, entregando transparencia, trazabilidad y notificaciones automáticas.

1.1.3 Objetivo del Proyecto

Desarrollar e implementar un sistema de gestión de tareas organizacionales que permita a todos los miembros de la institución:

- Crear y asignar tareas según rol o área.
- Validar avances y finalización de actividades.
- Visualizar tareas en distintos estados (pendientes, en progreso, atrasadas, completadas).
- Aplicar filtros avanzados para mejorar la toma de decisiones.
- Recibir notificaciones automáticas sobre plazos y validaciones.

1.1.4 Alcances

- Creación y gestión de tareas individuales y de equipo.
- Tablero de tareas con filtros (por área, estado, responsable, prioridad).

- Notificaciones en tiempo real para asignaciones y vencimientos.
- Validación de tareas por jefatura antes de darlas por finalizadas.
- No contempla en la primera versión integración con ERP o CRM corporativos.
- No incluye análisis financiero avanzado ni reportes BI.

2. Descripción General de la Metodología ágil a adoptar.

2.1. Fundamentación

Se adoptará Scrum como marco de trabajo, con sprints de 2 semanas y entregables incrementales. La naturaleza adaptable de Scrum permite responder rápidamente a cambios de prioridades en una organización.

El ciclo de vida del producto será iterativo e incremental:

- Iterativo porque se revisan y mejoran funcionalidades en cada sprint.
- Incremental porque el sistema crecerá en módulos: creación de tareas → validación → notificaciones → reportes

2.2. Valores de trabajo

El equipo Scrum trabajará bajo los valores de:

- **Compromiso**: cumplir los acuerdos de sprint.
- **Transparencia**: todos pueden ver el estado del trabajo.
- **Colaboración**: áreas diferentes trabajando sobre un mismo sistema.
- **Mejora continua (Kaizen)**: cada sprint es una oportunidad de ajustar y optimizar.

Personas y roles del proyecto.

Persona	Rol	Función
Director del proyecto	Product Owner	Define visión, prioriza backlog y valida entregables.
Coordinador TI	Scrum Master	Guía la aplicación de Scrum, elimina impedimentos.
Jefaturas de área y colaboradores	Stakeholders	Validan requerimientos, revisan tareas de sus equipos.
Developer Frontend	Developer 1	Consturye la interfaz angular/ionic
Developer Backend	Developer 2	Implementa API en Node.js/Django
Develor QA/BD	Developer 3	Diseña pruebas de calidad y administra la base de datos.

3.Product Backlog. Lista de Componentes y artefactos para Construir.

3.1. Épicas e historias de usuarios.

Épica	Historia de Usuario	Prioridad	Estado	Puntos de Historia	Esfuerzo (horas)	Duración (días)	Sprint
Gestión de Tareas	Como jefe quiero asignar tareas a colaboradores para asegurar ejecución del área	Alta	Pendiente	8	24 H	3	1
Validación	Como jefe quiero aprobar tareas antes de que se marquen como completadas.	Alta	Pendiente	5	15 H	2	1
Notificaciones	Como colaborador quiero recibir alertas del vencimiento de tareas o cuando se me asignen tareas.	Media	Pendiente	8	24 H	3	2
Filtros	Como usuario o jefatura, quiero filtrar tareas por área, estado o responsable	Media	Pendiente	5	15 H	2	2
Dashboard	Como gerente, quiero ver un panel de indicadores que muestre información del estado de las tareas (por ejemplo. Finalizadas, atrasadas, etc)	Alta	Pendiente	13	40 H	5	3

4.Definición tecnologías de Desarrollo a utilizar

En **KaiTasks**, una historia o incremento se considerará *Done* (completado) solo cuando cumpla con criterios claros y verificables. Estos criterios garantizan calidad y coherencia en las entregas.

4.1. Criterios generales de “Done”:

1. **Implementación completa** de la funcionalidad descrita en la historia de usuario.
2. **Pruebas unitarias aprobadas** y sin errores críticos en la ejecución.
3. **Validación funcional**: revisada y aprobada por el Product Owner en la Sprint Review.
4. **Documentación mínima** incluida (descripción técnica y manual de uso básico).
5. **Integración estable**: debe funcionar en conjunto con las demás funcionalidades desarrolladas.
6. **Aceptación de usuarios clave** (en pruebas internas por jefaturas o equipo de QA).

Con lo anterior, se entiende que una tarea de “notificaciones” estará *Done* solo si:

- Se generan alertas al asignar tareas y al vencer plazos.
- El colaborador recibe notificación en su panel y/o correo.
- Las notificaciones se registran en la base de datos.
- Ha pasado pruebas de QA y revisión del Product Owner.

5.Definición de tecnologías de implementación.

- **Frontend (Interfaz de Usuario)**
  - **Angular + Ionic**:
    - Permite construir una app híbrida (web y móvil).
    - Ionic aporta componentes UI optimizados para dispositivos móviles.
    - Angular asegura escalabilidad y modularidad del código.
- **Backend (Lógica y Servicios)**
  - **Node.js con Express** (alternativa: Django REST Framework en Python):
    - Ideal para APIs RESTful.
    - Alto rendimiento para manejar múltiples peticiones concurrentes.
    - Ecosistema amplio de librerías.
- **Base de Datos**
  - **PostgreSQL**:
    - Relacional, robusta y segura.
    - Maneja transacciones y reportes con eficiencia.
    - Ideal para consultas avanzadas (filtrado por estado, área, responsable).
- **Infraestructura / Deployment · AWS (Amazon Web Services)**:
    - EC2 para servidores.
    - RDS para base de datos.
    - S3 para almacenamiento de archivos.
  - Alternativa: **Azure** o **Heroku** para despliegue inicial.
- **Valor Justificativo**

Estas tecnologías permiten construir una solución escalable, multiplataforma, con integración ágil y costo accesible.
El proyecto **KaiTasks** usará un conjunto de herramientas alineadas a la filosofía ágil:

- **Figma:**
  - Diseño de prototipos interactivos (pantallas: inicio, tareas, notificaciones, dashboard).
  - Permite validar con usuarios antes de programar.
- **GitHub:**
  - Control de versiones del código.
  - Ramas separadas por funcionalidad (feature/bugfix).
  - Pull Requests para revisión de código.
- **Microsoft Teams:**
  - Comunicación entre equipo.
  - Grabación de reuniones de avance.
  - Herramienta para presentar el video final.
- **Burndown Chart (generado en Trello o Excel):**
  - Seguimiento gráfico de historias completadas vs. tiempo.
  - Permite verificar si el sprint avanza al ritmo esperado.

6.Roadmap

- **Sprint 1 (Semana 1–2): Funcionalidades Básicas** • Diseño de maquetas en Figma (pantallas principales).
    - Implementación de creación y asignación de tareas.
    - Configuración inicial de Trello y GitHub.
    - Primer burndown chart.
- **Sprint 2 (Semana 3–4): Colaboración y Notificaciones**
    - Implementación de validación de tareas por jefaturas.
    - Desarrollo del sistema de notificaciones (alertas de asignación y vencimiento).
    - Filtros básicos (estado: pendiente, en progreso, completada).

- Revisión de sprint con demo funcional.

- **Sprint 3 (Semana 5–6): Reportes y Dashboard**
  - Desarrollo de dashboard de indicadores con: ○ El total de tareas creadas.
    - Tareas completadas vs. atrasadas.
    - Estadísticas por área.
  - Integración de filtros avanzados (área, responsable, prioridad).
  - Documentación técnica.

- **Semana 7: Cierre del Proyecto**
  - Pruebas integrales.
  - Ajustes finales.
  - Grabación del video (máx. 7 min) en Teams.

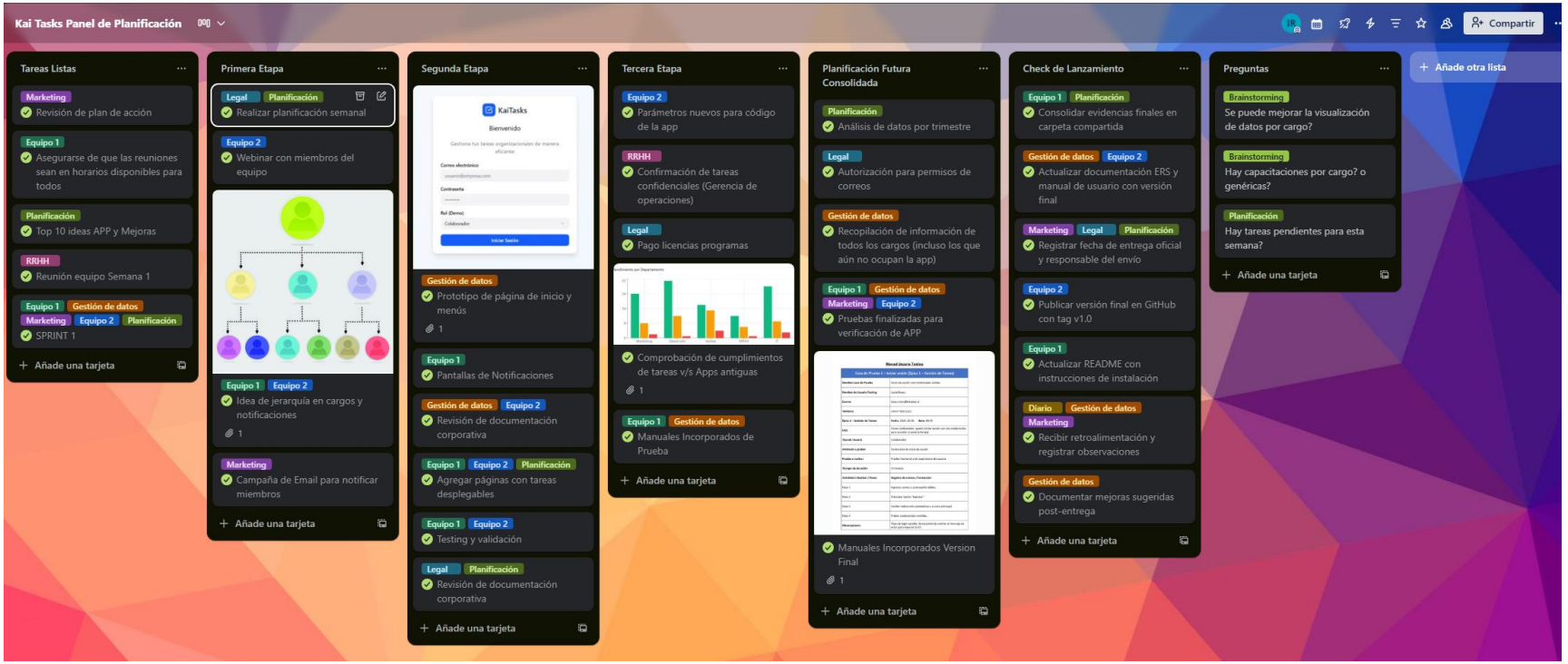
7.Producto BackLog Final

Épica	Historia de Usuario	Prioridad	Estado	Sprint
Gestión de Tareas	Como jefe quiero asignar tareas.	Alta	Completado	1
Validación	Como jefe quiero aprobar tareas antes de cerrarlas.	Alta	Completado	2
Notificaciones	Como colaborador quiero recibir alertas.	Media	Completado	2
Filtros	Como usuario quiero filtrar tareas por área o estado.	Media	Completado	3
Dashboard	Como gerente quiero visualizar KPIs.	Alta	Completado	3
Seguridad	Como usuario quiero iniciar sesión según rol.	Alta	Completado	4

8.Resumen de Sprints

Sprint	Objetivo	Resultados	Avance
1	Definir backlog y estructura base	Login funcional	90%
2	Implementar validaciones y notificaciones	Flujo funcional	100%
3	Desarrollar dashboard y filtros	Métricas y KPIs	95%
4	Pruebas y documentación final	Versión final operativa	100%

9.Burndown Chart



10. Retrospectiva Final

El equipo de desarrollo mantuvo una comunicación constante, adaptándose a cambios y cumpliendo los objetivos de cada sprint. El uso de Scrum permitió identificar oportunidades de mejora y garantizar entregas funcionales.

Duoc UC