

BogoTrash: Informe 1

Integrantes del Equipo:

- Daniel Santiago Avila Medina
- Santiago Avila Barbudo
- David Santiago Piñeros Rodriguez
- Ivan Alejandro Pardo Montenegro

1. Introducción

Este informe documenta el proceso seguido para la gestión de nuestro proyecto, especificando las acciones realizadas, la forma en que se implementaron y la metodología utilizada para asegurar un enfoque eficiente. Finalmente, se incluyen evidencias visuales como capturas del repositorio, seguimiento en el tablero Kanban y gráficos del proyecto, brindando una visión clara y organizada del avance logrado.

2. Organización del repositorio

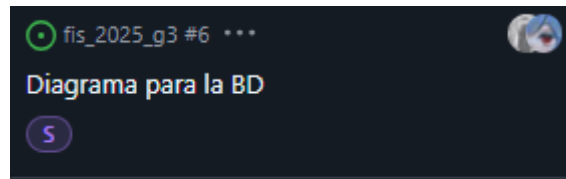
Para la organización y estructura de las carpetas del proyecto se siguió lo establecido en el boilerplate que se nos indicó de utilizar.

📁 .github	Estructura boilerplate del proyecto	5 days ago
📁 conf	Estructura boilerplate del proyecto	5 days ago
📁 docs	visualización de la app en Android estudio (#18)	1 hour ago
📁 jupyter	Fixes #15: Configure mi entrono de desarrollo en Android St...	yesterday
📁 scripts	Fixes #15: Configure mi entrono de desarrollo en Android St...	yesterday
📁 src	Add Logo de la app como icono y en layout #26	1 hour ago
📁 temp	Fixes #15: Configure mi entrono de desarrollo en Android St...	yesterday
📄 .gitignore	Estructura boilerplate del proyecto	5 days ago
📄 CHANGELOG.md	Estructura boilerplate del proyecto	5 days ago
📄 CONTRIBUTING.md	Estructura boilerplate del proyecto	5 days ago
📄 Dockerfile	Estructura boilerplate del proyecto	5 days ago
📄 LICENSE	Estructura boilerplate del proyecto	5 days ago
📄 Makefile	Estructura boilerplate del proyecto	5 days ago
📄 README.md	fix: ✂ resuelve conflictos en README y unifica contenido (#...	24 minutes ago
📄 docker-compose.yml	Estructura boilerplate del proyecto	5 days ago

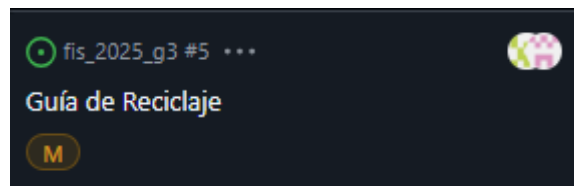
3. Tamaño de las tareas

Se usaron etiquetas para definir la complejidad de las tareas:

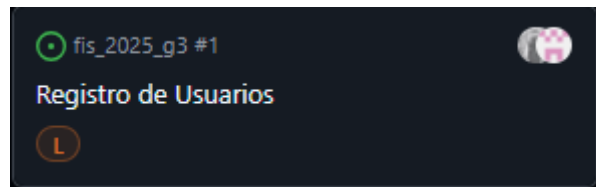
- S: Menos de un día.



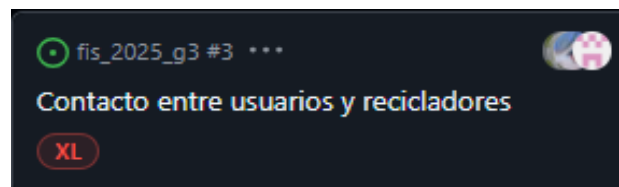
- M: Entre 1-3 días.



- L: Más de 3 días, dividiéndose en subtareas.



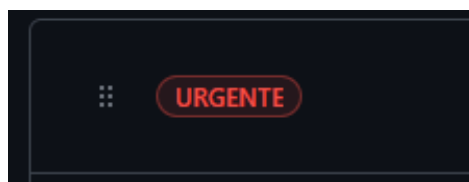
- XL: Más de 8 días, dividiéndose en subtareas.



4. Prioridad de las tareas

Para clasificar la prioridad de las tareas tenemos:

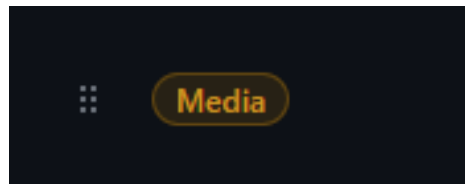
- Tareas que deben ser realizadas de manera urgente y de la manera más rápida posible



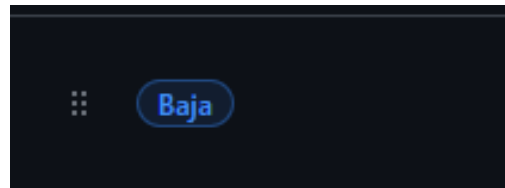
- Tareas que tienen una prioridad alta y deben priorizarse sobre las demás



- Tareas que tienen una prioridad media y deben priorizarse solo sobre las de prioridad baja



- Tareas que no tienen mucho impacto en el desarrollo del proyecto para ese punto y poseen la prioridad más baja




5. Issue de ejemplo


Para observar todo lo anterior de manera aplicada, usaremos el siguiente issue de ejemplo.



En una vista más detallada, podemos observar cómo se tienen asignados unos responsables, una prioridad baja, un milestone, un tamaño S, y un estimado de 2, el cual fue determinado por todos los miembros del equipo mediante poker scrum.

Assignees

 Dimich1tri

 dsavilam


Labels

Diseño

Type

Task

Projects

 KAMBAN_FIS_2025_G3

Status Done


Priority Baja

Size S

Estimate 2

Sprints Sprint 1 • Feb 18 - Mar 3

Milestone

 Configuración Inicial del proyecto

Closed 3 hours ago, 100% complete

Adicionalmente se muestran otros detalles, como el proyecto al que está asociado al igual que el sprint en el que se desarrolló dentro del mismo, y otros como el tipo y labels asignados dentro de los establecidos dentro del proyecto.

6. Procesos Desarrollados

1. Análisis de un Milestone: Configuración Inicial del Proyecto



Objetivos del Milestone

Este milestone se diseñó con los siguientes objetivos:

- Establecer una estructura base del proyecto en el lenguaje de programación seleccionado (**Kotlin**).
- Configurar el entorno de desarrollo para todos los miembros del equipo.
- Definir la organización del repositorio mediante una estructura de carpetas adecuada.
- Implementar una metodología de gestión de tareas basada en **Kanban**.
- Diseñar los elementos visuales iniciales, como el logo de la aplicación.

2. Análisis de las Tareas

El milestone incluyó un total de **5 issues**, todos ellos finalizados exitosamente. A continuación, se presenta un análisis detallado de cada uno:

2.1 Crear la estructura base del proyecto en Kotlin

- **Etiqueta:** *Task*
- **Responsable:** Dimich1tri
- **Estado:** Cerrado hace 3 horas
- **Descripción:** Se creó la estructura mínima del proyecto utilizando **Kotlin**, permitiendo que los demás desarrolladores puedan empezar a contribuir con funcionalidades adicionales.

2.2 Configurar el entorno de desarrollo

- **Etiqueta:** *Task*
- **Responsable:** dsvilam

- **Estado:** Cerrado ayer
- **Descripción:** Se estableció un entorno de desarrollo estandarizado para el equipo, incluyendo configuraciones de IDEs, dependencias y herramientas necesarias.

2.3 Diseño del logo de la aplicación

- **Etiqueta:** *Diseño*
- **Responsable:** Dimich1tri
- **Estado:** Cerrado hace 5 días
- **Descripción:** Se realizó el diseño del logo oficial de la aplicación, asegurando coherencia visual con la identidad del proyecto.

2.4 Configurar el tablero KANBAN

- **Etiqueta:** *Task*
- **Responsable:** Dimich1tri
- **Estado:** Cerrado hace 5 días
- **Descripción:** Se estableció un tablero Kanban para la gestión de tareas, facilitando la organización y seguimiento del trabajo en equipo.

2.5 Crear la estructura de carpetas en el repo

- **Etiqueta:** *Task*
- **Responsable:** Dimich1tri
- **Estado:** Cerrado hace 5 días
- **Descripción:** Se definió una estructura de carpetas para el repositorio, permitiendo la organización eficiente del código y otros recursos del proyecto.

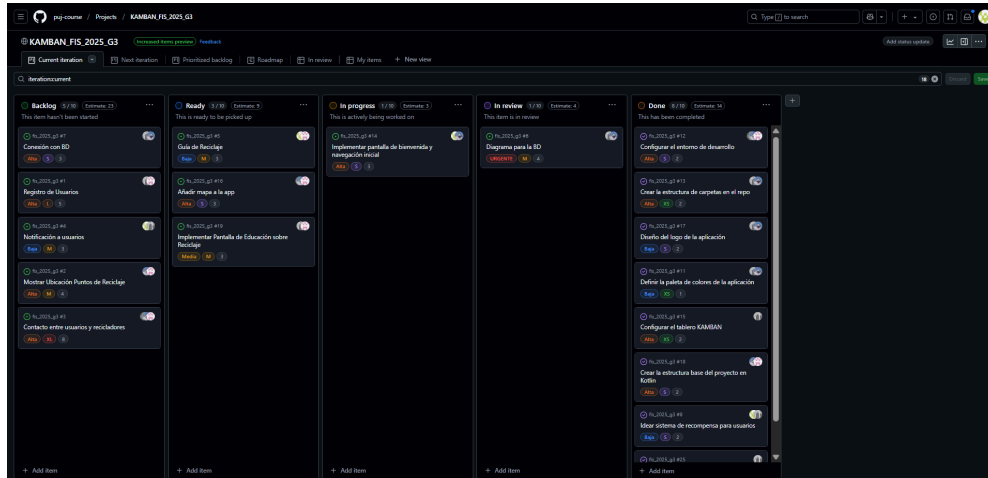
7. Evaluación y Resultados

La culminación exitosa de este milestone garantiza que el proyecto cuenta con:

- Un punto de partida bien estructurado para el desarrollo.
- Un entorno de trabajo homogéneo para el equipo.
- Una gestión de tareas eficiente mediante Kanban.
- Una organización clara del código fuente y recursos.
- Elementos visuales iniciales alineados con la identidad del proyecto.

El milestone "**Configuración Inicial del Proyecto**" fue esencial para sentar las bases del desarrollo, eliminando posibles bloqueos iniciales y permitiendo un avance progresivo en la construcción de funcionalidades más avanzadas. Su correcto desarrollo optimizará el rendimiento del equipo y mejorará la gestión del tiempo y recursos a lo largo del ciclo de vida del proyecto.

8. Tablero Kanban:



Backlog (Pendientes)

- Contiene tareas que aún no han sido iniciadas.
- Ejemplo: "Conexión con BD", "Registro de Usuarios", "Notificación a usuarios".
- Se etiquetan con prioridad (Alta, Baja, Media) y tamaño de la tarea (S, M, L, XL).

Ready (Listas para iniciar)

- Son tareas que han sido priorizadas y están listas para que un desarrollador las tome.
- Ejemplo: "Guía de Reciclaje", "Añadir mapa a la app".
- Aún no están en progreso, pero son candidatas para la siguiente fase.

In Progress (En progreso)

- Tareas en las que actualmente está trabajando un miembro del equipo.
- Ejemplo: "Implementar pantalla de bienvenida y navegación inicial".
- Indica que la tarea ha sido asignada y está en desarrollo.

In Review (En revisión)

- Tareas que ya fueron implementadas y están en proceso de validación.
- Ejemplo: "Diagrama para la BD".
- Puede implicar revisión de código, pruebas o aprobación por parte del equipo.

Done (Completadas)

- Son tareas finalizadas y aprobadas.
- Ejemplo: "Configurar el entorno de desarrollo", "Crear la estructura de carpetas en el repo", "Diseño del logo de la aplicación".
- Estas tareas ya no requieren más trabajo y han sido entregadas.

9. Análisis de las Gráficas

Tipo de Gráfica: Burndown Chart

Un Burndown Chart es una herramienta común en la gestión ágil de proyectos que muestra el trabajo restante a lo largo del tiempo en un proyecto dividido en sprints.

Elementos Clave de la Gráfica



1. Ejes:

- Eje X: Representa los diferentes sprints del proyecto (Sprint 1, Sprint 2, Sprint 3, Sprint 4).
- Eje Y: Representa la cantidad de trabajo pendiente o en proceso (probablemente en número de tareas o puntos de historia).

2. Curvas de Colores:

- Azul (Backlog): Representa las tareas aún en la lista de pendientes.
- Verde (Ready): Representa las tareas que están listas para trabajarse.
- Naranja (In Progress): Representa las tareas actualmente en desarrollo.
- Rosa (In Review): Representa tareas que están en revisión.
- Amarillo (Done): Representa las tareas completadas.

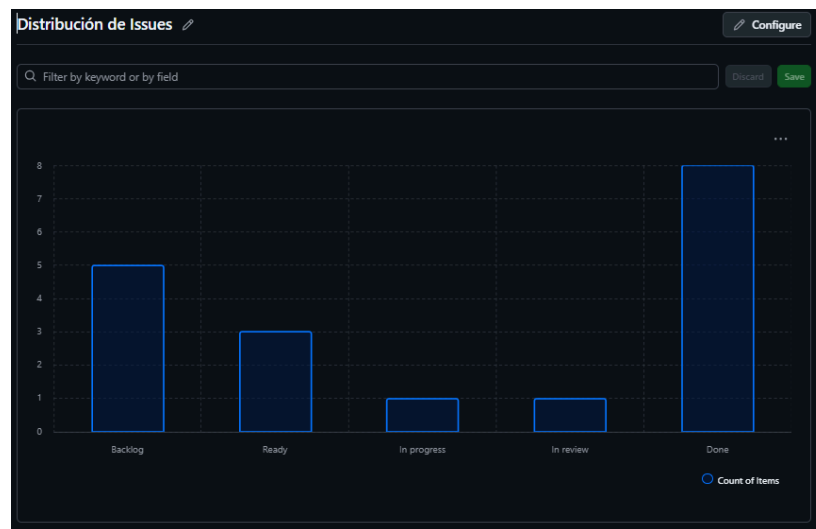
3. Patrón Observado:

- Inicialmente, en el Sprint 1, hay un crecimiento en la cantidad de tareas en progreso y en revisión.
- En el Sprint 2, se alcanza un pico en las tareas en progreso y revisión.
- A partir del Sprint 3, las tareas comienzan a disminuir rápidamente, lo que indica que el equipo está completando el trabajo.
- Para el Sprint 4, las tareas pendientes casi desaparecen, excepto por un leve aumento en el backlog (azul), lo que sugiere que se han agregado nuevas tareas o que algunas no se completaron.

Tipo de Gráfica: Distribución de Issues

Descripción General:

Esta gráfica de barras muestra la distribución de issues en diferentes estados dentro de un sistema de gestión de tareas. Se presentan cinco categorías: Backlog, Ready, In Progress, In Review y Done, con el número de issues en cada una de ellas.



Observaciones por Categoría:

1. **Backlog (5 issues):** Representa las tareas acumuladas que aún no han sido abordadas. Es el segundo grupo más grande, lo que indica que aún hay trabajo pendiente por planificar.
2. **Ready (3 issues):** Son tareas listas para ser ejecutadas, pero que aún no han sido iniciadas.
3. **In Progress (1 issue):** Solo hay una tarea en desarrollo, lo que puede sugerir que el equipo está abordando las tareas de forma controlada o que hay capacidad disponible.
4. **In Review (1 issue):** Una tarea está en proceso de revisión antes de su finalización.
5. **Done (8 issues):** Es el grupo más grande, lo que refleja un buen nivel de finalización de tareas.

Tipo de Gráfica: Tareas por Prioridad

Descripción General:

Esta gráfica de barras representa la cantidad de tareas clasificadas según su nivel de prioridad: Urgente, Alta, Media y Baja. Cada barra indica cuántas tareas pertenecen a cada categoría.



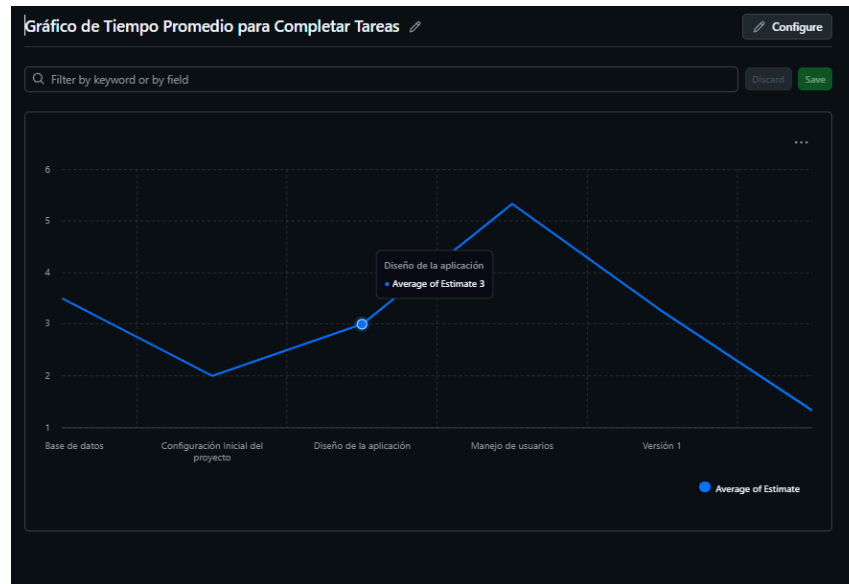
Observaciones por Prioridad:

- 1. Urgente (~1 tarea):** Hay muy pocas tareas con esta clasificación, lo que sugiere que el sistema tiene pocos problemas críticos que requieren atención inmediata.
- 2. Alta (11 tareas):** Es la categoría con mayor cantidad de tareas, lo que indica que la mayoría de los elementos en la lista tienen una prioridad significativa.
- 3. Media (~1 tarea):** Casi no hay tareas con prioridad media, lo que podría sugerir que las tareas tienden a clasificarse como alta o baja sin un término medio.
- 4. Baja (5 tareas):** Hay una cantidad moderada de tareas de baja prioridad, lo que sugiere que hay tareas que pueden postergarse sin afectar significativamente el flujo de trabajo.

Tipo de Gráfica: Tiempo Promedio para Completar Tareas

Descripción General:

Esta gráfica de línea muestra el tiempo promedio estimado para completar diferentes tareas dentro de un proyecto. El eje **X** representa las distintas fases o componentes del proyecto, mientras que el eje **Y** muestra el tiempo promedio en unidades no especificadas (posiblemente días u horas).



Observaciones Clave:

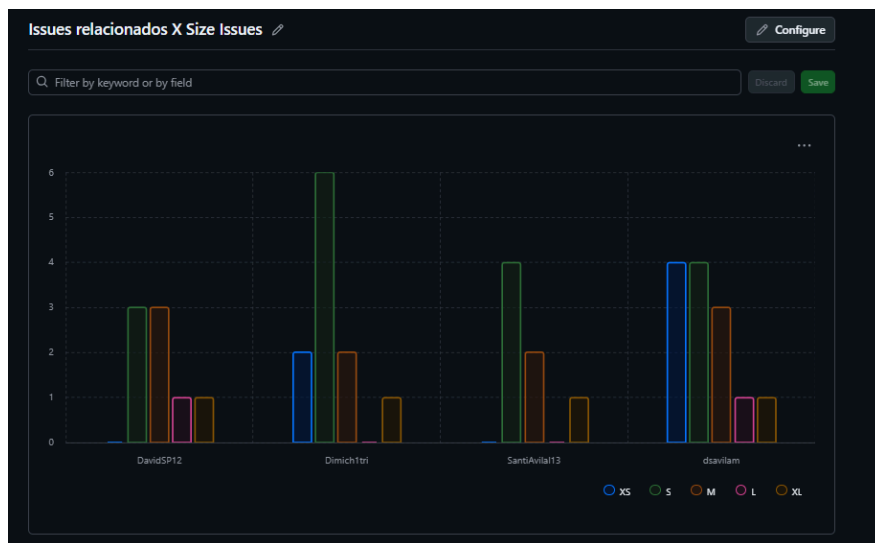
1. **Base de datos (~3.5 unidades de tiempo):** La configuración inicial de la base de datos tomó un tiempo moderado, aunque no es la tarea más demandante.
2. **Configuración inicial del proyecto (~2 unidades de tiempo):** Es la fase que menos tiempo ha requerido, posiblemente porque se trata de ajustes básicos antes del desarrollo real.
3. **Diseño de la aplicación (~3 unidades de tiempo):** Se observa un leve incremento en el tiempo necesario para esta fase en comparación con la configuración inicial.
4. **Manejo de usuarios (~5.5 unidades de tiempo):** Es la tarea con el mayor tiempo promedio de finalización, lo que sugiere que tiene una alta complejidad o requiere más trabajo que las demás.

5. **Versión 1 (~1.5 unidades de tiempo):** Se observa una fuerte disminución en el tiempo promedio, indicando que esta fase final tomó menos tiempo, probablemente porque consolidó el trabajo realizado previamente.

Tipo de Gráfica: Issues Relacionados X Size Issues

Descripción General:

Este gráfico de barras muestra la cantidad de problemas ("issues") relacionados con diferentes tamaños de producto o elemento (XS, S, M, L, XL) para varios usuarios o colaboradores (DavidSP12, Dimich1tri, SantiAvila13, dsavilam).



Observaciones Clave:

1. **DavidSP12:**

- Los problemas más comunes están en los tamaños **S y M**.
- También hay algunos problemas en **L y XS**, aunque en menor cantidad.

2. **Dimich1tri:**

- Presenta la mayor cantidad de problemas en el tamaño **S**.
- También enfrenta problemas en **XS, M y L**, aunque en menor cantidad.

3. **SantiAvila13:**

- Los problemas predominan en el tamaño **S**.
- Hay incidencias en **M y L**, pero son menos frecuentes.

4. dsavilam:

- Es el único con problemas significativos en **XS y S**.
- También tiene incidencias en **M y L**, aunque en menor número.

Tipo de Gráfica: Tamaño de Issues

Descripción General:

Este gráfico de barras horizontales muestra la cantidad de problemas asociados a cada tamaño (XS, S, M, L, XL).



Observaciones Clave:

1. El tamaño S presenta la mayor cantidad de issues, con un total de 6.
2. El tamaño M también tiene un alto número de issues, con 5 reportes.
3. El tamaño XS tiene 4 issues, lo que indica que también es problemático.
4. Los tamaños L y XL tienen significativamente menos issues, con 1 reporte cada uno.

10. Explicación y Justificación

La implementación que realizamos de nuestro flujo de trabajo ágil mediante GitHub Projects demuestra una estructura bien pensada que optimiza la gestión del proyecto que vamos a realizar de BogoTrash en varios aspectos clave, tales como:

- I. Organización del Tablero: Esta distribución permite visualizar fácilmente el progreso de las tareas y detectar cuellos de botella, como se evidencia en las capturas de pantalla del informe.
- II. Gestión de Sprints y Milestones: La creación del Milestone "Versión 1" con fechas límite específicas y la división en sprints de dos semanas nos permite una planificación correcta del proyecto. Esta estructura temporal permite:
 - Mantener un ritmo constante en el desarrollo
 - Facilitar la entrega incremental de funcionalidades
 - Mejorar la previsibilidad del proyectoSistema de Etiquetado: El uso de etiquetas es particularmente efectivo:
 - Facilitan la identificación rápida del tipo de trabajo
 - Permiten filtrar tareas por área de especialización
 - Mejoran la asignación de recursos según experiencia del desarrollador a cargoEtiquetas de Tamaño (S, M, L, XL):
 - Proporcionan una estimación del esfuerzo requerido según el tamaño del trabajo
 - Ayudan a mantener una distribución equilibrada de trabajo
 - Facilitan la identificación de tareas que necesitan ser divididasSeguimiento y Métricas: Las métricas y tablas creadas nos ayudan a identificar cómo funciona el diseño del proyecto realizado en GitHub projects
 - El flujo de trabajo a través de las diferentes etapas
 - La distribución de tareas entre los miembros del equipo
 - El progreso general del sprint

Esta estructura nos está ayudando a realizar una gestión efectiva del proyecto, gracias a herramientas como el tablero Kamban, los gráficos, la creación de milestones, la gestión de issues y todo lo que nos permite hacer Github, estos elementos nos permiten un seguimiento detallado del progreso y facilita la identificación de posibles problemas o retrasos con rapidez.