



UNIVERSIDAD  
DEL VALLE  
DE GUATEMALA

UNIVERSIDAD DEL VALLE DE GUATEMALA

---

Ingeniería de Software 2  
Sección 30

**Quinto Sprint:  
Optimización de la gestión de inventario y  
planificación de materiales en Pool Center**

Facultad de Ingeniería

Iris Ayala, 23965  
Gabriel Bran, 23590  
David Domínguez, 23712  
Luis Padilla, 23663  
Anggie Quezada, 23643

Guatemala

2025

# **Product Backlog**

## **Historias de usuario**

1. Como gerente, quiero ver un dashboard con indicadores generales (stock total, proyectos activos, alertas), para tener una vista rápida del estado del sistema.
2. Como gerente, quiero ver mi inventario en tiempo real, para saber qué materiales están disponibles.
3. Como gerente, quiero registrar la entrada de materiales, para mantener actualizado el inventario.
4. Como ingeniero, quiero asignar materiales a proyectos específicos, para evitar confusión en la distribución de recursos.
5. Como ingeniero, quiero recibir una alerta cuando un material tenga bajo stock, para reabastecerlo con tiempo.
6. Como secretaria, quiero recibir una alerta cuando un material tenga bajo stock, para cotizar materiales con tiempo con los proveedores.
7. Como ingeniero, quiero actualizar el estado de los proyectos en la plataforma, para monitorear el progreso.
8. Como gerente, quiero ver los avances de los proyectos en un solo reporte, para analizar tiempos y costos.
9. Como gerente, quiero tener un sitio web con información de la empresa, para atraer más clientes.
10. Como gerente, quiero registrar un nuevo proyecto.
11. Como usuario, quiero poder iniciar sesión en el sistema, para acceder a mis funcionalidades según mi rol.
12. Como gerente, quiero registrar nuevos usuarios en el sistema, para que puedan acceder según su rol.

## **Lista de tareas para construir el sistema**

### *Infraestructura*

## **Lista de tareas para construir el sistema**

### *Infraestructura*

Tarea	Descripción	Puntos	Prioridad
Configurar React	Configurar el entorno de desarrollo, instalar librerías, modulos	1	Alta
Configurar Express	Configurar el entorno de desarrollo, instalar librerías, modulos	1	Alta
Configurar la BD	Crear la base de datos	2	Alta
Configurar la conexión del backend con la BD	Configurar conexión desde backend a PostgreSQL	3	Alta
Configurar archivo .env	Guardar claves secretas, rutas, configuración JWT y variables sensibles .env.	1	Alta
Crear dockerfile para backend	Contenerizar backend con Node y Express para el servicio	2	Alta
Crear dockerfile para fronted	Contenerizar fronted con Node para el servicio	2	Alta

Crear docker-compose	Configurar docker-compose con servicio backend y PostgreSQL conectado	2	Alta
----------------------	---	---	------

*Como usuario, quiero poder iniciar sesión en el sistema, para acceder a mis funcionalidades según mi rol.*

Tarea	Descripción	Puntos	Prioridad
Diseñar las pantallas	Realizar los diseños para maquetar la pantalla de login	1	Media
Crear formulario de login	Maquetar pantalla con campos de email y contraseña, y botón de enviar.	2	Alta
Validar campos	Validar que los campos no estén vacíos y tengan formato correcto.	3	Alta
Integrar API de login	Hacer petición al backend con email y contraseña.	2	Alta
Guardar token y datos en localStorage	Guardar token de sesión y usuario al iniciar sesión.	2	Alta
Crear endpoint /login	Crear endpoint POST para recibir email y contraseña.	2	Alta
Validar usuario en base de datos	Verificar existencia de usuario y comparar contraseña hasheada.	2	Alta
Generar token JWT	Crear y devolver un token JWT con la información del usuario.	2	Alta
Probar el endpoint	Probar el endpoint utilizando Postman	2	Alta
Crear tabla usuarios	Crear tabla con los campos necesarios	2	Alta

*Como gerente, quiero registrar nuevos usuarios en el sistema, para que puedan acceder según su rol.*

Tarea	Descripción	Puntos	Prioridad
Diseñar las pantallas	Realizar los diseños para maquetar la pantalla de login	1	Media
Crear formulario de register	Maquetar pantalla con campos de nombre, email y contraseña, y botón de enviar.	2	Alta
Validar campos	Validar que los campos no estén vacíos y tengan formato correcto.	3	Alta
Integrar API de register	Hacer petición al backend con los datos.	2	Alta
Crear endpoint /register	Crear endpoint POST para recibir nombre, email y contraseña.	2	Alta
Probar el endpoint	Probar el endpoint utilizando Postman	2	Alta
Insertar usuario en la tabla	Insertar al usuario en la bd de datos para tener su información.	2	Alta

*Como gerente, quiero ver mi inventario en tiempo real, para saber qué materiales están disponibles.*

Tarea	Descripción	Puntos	Prioridad
Diseñar las pantallas	Realizar los diseños para maquetar la pantalla de inventario	1	Media
Crear vista de inventario	Maquetar pantalla con tabla para listar materiales.	3	Alta
Hacer tabla interactiva	Mostrar materiales con botones para editar, borrar, asignar, estados, etc	5	Alta
Crear endpoint GET /materiales	Obtener todos los materiales desde la base de datos.	2	Alta

Crear tabla de materiales	Tabla con todos los campos necesarios como el MER	1	Alta
---------------------------	---	---	------

*Como gerente, quiero registrar la entrada de materiales, para mantener actualizado el inventario.*

Tarea	Descripción	Puntos	Prioridad
Crear formulario de entrada de material	Formulario para registrar nueva entrada de material	3	Alta
Validar datos de entrada	Verificar que campos no estén vacíos ni datos incorrectos.	3	Alta
Crear endpoint POST	Registrar entrada de material, actualizar stock y registrar movimiento.	3	Alta
Insertar material en la BD	Insertar la información del material en la BD	2	Alta

*Como ingeniero/gerente, quiero asignar materiales a proyectos específicos, para evitar confusión en la distribución de recursos.*

Tarea	Descripción	Puntos	Prioridad
Crear formulario de asignación de materiales	Seleccionar proyecto y materiales con cantidad a asignar.	5	Alta
Validar stock disponible antes de asignar	Mostrar alerta si el stock es insuficiente.	5	Alta
Crear endpoint POST	Registrar asignación y disminuir stock correspondiente.	5	Alta
Validar stock en backend	No permitir asignar más de lo disponible.	2	Alta
Insertar la asignación en la BD	Registrar proyecto, material, cantidad asignada y fecha.	2	Alta

*Como ingeniero/secretaria/gerente, quiero recibir una alerta cuando un material tenga bajo stock, para reabastecerlo con tiempo.*

Tarea	Descripción	Puntos	Prioridad
Mostrar notificación visual	Mostrar alerta en pantalla de inventario o dashboard.	5	Alta
Crear lógica de verificación de stock mínimo	Al modificar stock, verificar si se alcanza stock mínimo.	5	Alta
Enviar alerta al fronted	Devolver notificación al usuario correspondiente	3	Alta

*Como gerente, quiero registrar un nuevo proyecto.*

Tarea	Descripción	Puntos	Prioridad
Crear formulario de registro de proyecto	Formulario con los campos necesarios	3	Alta
Validar campos	Evitar campos vacíos y datos inválidos.	3	Alta
Crear endpoint POST /proyectos	Insertar nuevo proyecto en la base de datos.	3	Alta
Crear tabla proyectos	Tabla con los campos necesarios	2	Alta

*Como ingeniero/gerente, quiero un dashboard para ver los proyectos, quiero actualizar el estado de los proyectos en la plataforma, para monitorear el progreso.*

Tarea	Descripción	Puntos	Prioridad
Crear vista de dashboard de proyectos	Visualización general de todos los proyectos con filtros por estado.	5	Alta
Mostrar reportes disponibles por proyecto	Listar reportes adjuntos	5	Alta
Crear elementos interactivos para editar el progreso de los proyectos	Mostrar opciones de estado, borrar, ver reportes, etc.	5	Alta
Crear endpoint PATCH /proyectos/:id/estado	Actualizar campo estado en un proyecto específico.	3	Alta
Crear endpoint GET /proyectos	Devolver resumen de proyectos agrupados por estado.	2	Alta
Crear endpoint GET /proyectos/:id/reportes	Devolver lista de reportes adjuntos a un proyecto.	2	Alta

*Como gerente, quiero tener un sitio web con información de la empresa, para atraer más clientes.*

Tarea	Descripción	Puntos	Prioridad
Crear página de inicio pública	Maquetar vista con información general de la empresa, bienvenida, logo, diseño	2	Media
Crear sección de servicios ofrecidos	Mostrar los servicios que brinda Pool Center con descripciones e imágenes.	2	Media
Crear sección de proyectos realizados (portafolio)	Mostrar imágenes o ejemplos de proyectos anteriores.	2	Media
Crear página de contacto	Informacion de contacto para que clientes se contacten	2	Media

## Sprint Backlog

Tarea	Descripción	Horas de desarrollo	Puntos	Responsable	Fecha fin
Quitar tablas innecesarias de la BD	Quitar tablas innecesarias de la BD	1	1	Luis	23/07
Agregar tablas para el mecanismo del inventario	Agregar tablas para el mecanismo del inventario	2	2	Luis	23/07
Cambiar atributos de las tablas de la BD	Cambiar atributos de las tablas de la BD	2	2	Luis	16/07
Actualizar el DER	Actualizar el DER	1	3	Luis	16/07
Insertar datos de prueba para probar el funcionamiento y los endpoints	Insertar datos de prueba para probar el funcionamiento y los endpoints	2	5	Luis	16/07
Crear diseño de vista de inventario	Crear diseño de vista de inventario	2	5	Iris	24/07
Crear diseño de vista de proyecto	Crear diseño de vista de proyecto	1	5	Iris	25/07

Re-diseñar componentes de creación	Re-diseñar componentes de creación	1	5	Iris	25/07
Crear diseño de componente de nuevo material	Crear diseño de componente de nuevo material	1	5	Iris	16/07
Crear diseño de componente de asignar material	Crear diseño de componente de asignar material	1	5	Iris	24/07
Arreglar/Crear un banner del sitio web	Arreglar/Crear un banner del sitio web	1	5	Anggie	24/07
Reutilizar componentes y eliminar residuos en el front	Reutilizar componentes y eliminar residuos en el front	2	8	Anggie	24/07
Configurar herramientas de testing en back	Configurar herramientas de testing en back	1	3	David	24/07
Configurar herramientas de testing en front	Configurar herramientas de testing en front	1	3	Bran	20/07
Test Back: Verificar que las rutas API implementadas devuelvan lo correcto	Verificar rutas API implementadas	2	8	David	24/07
Test Front: Probar que una ruta redirige al login si no hay token	Verificar redirección sin token	2	8	Bran	20/07
Test Back: Verificar que se genera el JWT correctamente	Verificar generación de JWT	3	8	David	24/07
Crear documentación de las API (herramienta de swagger o colección de Postman)	Documentar API con Swagger o Postman	1	8	Bran	24/07

## Incremento

Repositorio Fronted: [https://github.com/AleWWH1104/Fronted\\_ISoftware1.git](https://github.com/AleWWH1104/Fronted_ISoftware1.git)

Repositorio Backend: [https://github.com/DavidDominguez-11/Backend\\_ISoftware1](https://github.com/DavidDominguez-11/Backend_ISoftware1)

### Mock Ups de las pantallas a desarrollar

Enlace a las pantallas: [https://www.canva.com/design/DAGkwiVb8X8/bqBVXfqehSxSm\\_dNge7KA/edit?utm\\_content=DAGkwiVb8X8&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGkwiVb8X8/bqBVXfqehSxSm_dNge7KA/edit?utm_content=DAGkwiVb8X8&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

Ejemplo

## Estimación de Tiempo y Costo del desarrollo

### Estimación de Puntos de Casos de Uso

- FPA (actores): 3 actores =  $(1 \text{ simple} + 1 \text{ medio} + 1 \text{ complejo}) = 1 \times 1 + 1 \times 2 + 1 \times 3 = 6$
- FPCU (casos de uso): 10 CU = 3 simples, 4 medios, 3 complejos =  $3 \times 5 + 4 \times 10 + 3 \times 15 = 125$
- PCU = FPA + FPCU =  $6 + 125 = 131$

### Ajustes de complejidad

- FCT (promedio de factores técnicos): 0.77
- FA (promedio de factores ambientales): 0.85
- PCUA =  $131 \times 0.77 \times 0.85 \approx 85.73$

### Esfuerzo Estimado

- FC = 20 HH / punto (por estar en zona de bajo riesgo)
- Esfuerzo total E =  $85.73 \times 20 = 1,714.6 \text{ HH}$

### Estimación de costos

- K (costos indirectos): 1.7
- Costo total sin ganancia =  $E \times K \times THP = 1,714.6 \times 1.7 \times 8.5 \approx \$24,749.89$
- Margen de ganancia (30%):  $\approx \$7,425$

Precio final del proyecto:  $\approx \$32,175 \text{ USD}$

El sistema desarrollado automatiza procesos clave para una pequeña empresa de construcción de piscinas, permitiéndole controlar sus materiales, gestionar clientes y generar reportes a través de una plataforma web intuitiva. Su costo final incluye no solo el esfuerzo técnico, sino también el valor agregado de planificación, diseño y pruebas, desarrollado por un equipo interdisciplinario durante seis meses.

Este precio es justo para el equipo porque refleja el tiempo invertido, costos operativos y una utilidad razonable, y es atractivo para el cliente considerando que:

- No se usaron licencias comerciales costosas (solo software libre).
- El costo de desarrollo es inferior al de contratar a una agencia.
- El sistema puede representar un ahorro o incremento de ingresos sustancial al evitar errores, mejorar la gestión y dar presencia online.

## Prueba – Framework

Para el proyecto se seleccionó Jest como framework de pruebas unitarias en backend y Vitest para frontend.

Jest fue elegido por su fácil configuración, soporte para pruebas asincrónicas, capacidad de mocking y amplia documentación. Además, permite integración con librerías como Supertest para probar endpoints REST de forma eficiente que se tiene planeado para el futuro.

Vitest se seleccionó para el frontend debido a su integración nativa con Vite, además de la rapidez en la ejecución y compatibilidad con React Testing Library, lo que permite pruebas centradas en la experiencia del usuario.

### Backend – Jest

Jest es desarrollado por Facebook y contiene mucha documentación que facilita su implementación. Lo seleccionamos como framework principal para pruebas unitarias en backend ya que:

- Su configuración es sencilla y no depende de herramientas adicionales.
- Soporte nativo para `async/await`, lo que facilita pruebas asincrónicas, necesario en nuestras funciones.
- Funcionalidades integradas como mocking, snapshots y coverage reporting. En especial los mocks nos permitió simular interacciones con bases de datos, lo cual fue clave para asegurar independencia de pruebas y velocidad de ejecución.
- Amplia documentación y comunidad, lo cual agiliza la resolución de dudas.

### Frontend – Vitest

Para el entorno frontend se eligió Vitest, un framework que se integra directamente con Vite, el cual ya era parte de la configuración base del proyecto en React. Las razones de su elección fueron:

- Su ejecución extremadamente rápida gracias al uso de módulos ESM y su arquitectura basada en Vite.
- Cuenta con una API muy similar a Jest, lo que facilita la curva de aprendizaje y la migración de proyectos.
- Integración directa con React Testing Library, lo que permitió escribir pruebas enfocadas en la experiencia del usuario final.

## Evidencia:

Video corriendo prueba de Frontend:

<https://www.youtube.com/watch?v=Jgyt5-4Aid4>

```
✓ src/test/ComponentIntegration.test.jsx (5 tests) 482ms
  ✓ Component Integration Tests > renders form with all components correctly 305ms
✓ src/components/Input.test.jsx (6 tests) 126ms

Test Files: 5 passed (5)
  Tests 29 passed (29)
  Start at 19:02:24
  Duration 7.36s (transform 340ms, setup 2.64s, collect 2.16s, tests 1.59s, environment 9.12s, prepare 2.13s)

PASS Waiting for file changes...
press h to show help, press q to quit

✖ vitest_bran ✖ 0 △ 0
```

Video corriendo prueba de Backend:

<https://www.youtube.com/watch?v=4-hgktrdOR0>

```
> backend_isoftware1@1.0.0 test:coverage
> jest --coverage --silent

PASS __tests__/_projects.test.js
PASS __tests__/_app.test.js
PASS __tests__/_auth.test.js
-----|-----|-----|-----|-----|-----|
File    | %Stmts | %Branch | %Funcs | %Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----|
All files | 63.5 | 63.15 | 33.33 | 64.44 |
src      | 100  | 100   | 100   | 100   |
app.js   | 100  | 100   | 100   | 100   |
src/config | 0    | 100   | 100   | 0     | 1-12
db.js    | 0    | 100   | 100   | 0     | 1-12
src/controllers | 58.75 | 63.15 | 37.5  | 60.25 |
authController.js | 65.38 | 58.82 | 40    | 68   | 72,92-93,99-119,126-127
projectController.js | 100  | 100   | 100   | 100   |
proyectosController.js | 0    | 100   | 0     | 0     | 1-17
rolesController.js | 37.5  | 100   | 0     | 37.5  | 4-11
src/middlewares | 0    | 100   | 0     | 0     |
Middleware.js    | 0    | 100   | 0     | 0     | 1-10
src/routes       | 82.14 | 100   | 100   | 82.14 |
authRoutes.js   | 100  | 100   | 100   | 100   |
dbRoutes.js     | 0    | 100   | 100   | 0     | 1-7
projectRoutes.js | 100  | 100   | 100   | 100   |
roleRoute.js    | 100  | 100   | 100   | 100   |
userRoutes.js   | 100  | 100   | 100   | 100   |

Test Suites: 3 passed, 3 total
Tests:      15 passed, 15 total
Snapshots:  0 total
Time:      3.646 s

✖ domain: ...\\Backend_ISoftware1 ✖ main $!? ● v22.14.0 ✖ 18:52 ✖ in 5s254ms ✖ npm run test:coverage [-+]
```

Los resultados fueron positivos:

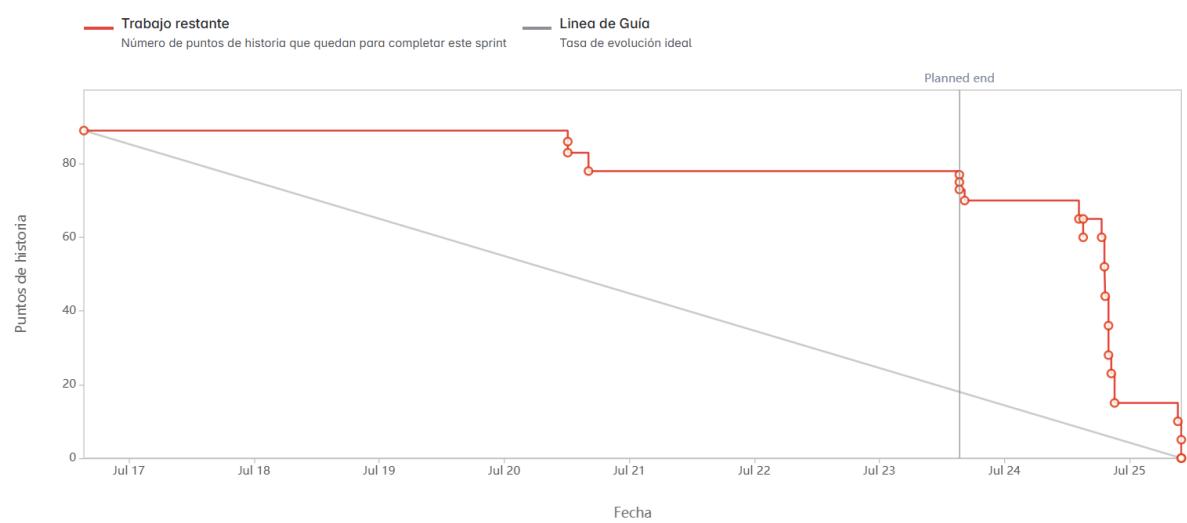
- Jest arrojó una cobertura total del 63.5%, permitiendo identificar módulos con bajo testeо, pero sobre todo comprobar que los implementados funcionan correctamente.
- Vitest ejecutó pruebas de componentes sin problemas, con tiempos de ejecución reducidos y resultados inmediatos.

Evidencia trabajo en equipo - reunión



## Resultados del Sprint

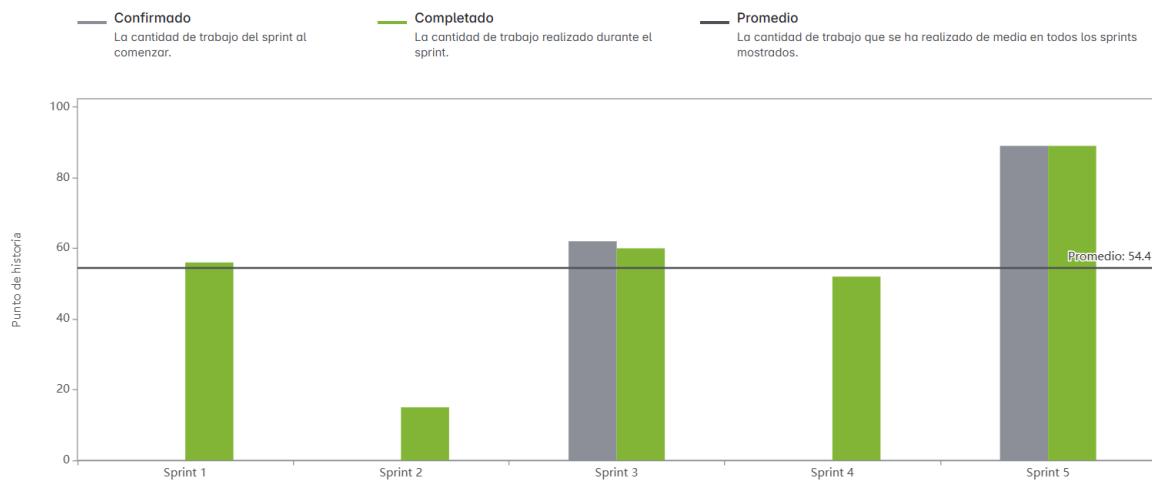
Gráfico burndown.



Métrica de velocidad (Gráfico con los datos de todos los sprints hasta el momento).

## Informe de velocidad

> Cómo leer este informe



### Indicador numérico del éxito del sprint junto con la justificación.

El indicador numérico de éxito del sprint lo obtuvimos calculando la proporción entre los puntos de historia completados y los puntos planificados.

$$\left( \frac{88 \text{ puntos completados}}{88 \text{ puntos planificados}} \right) \times 100 \times 0.85 = 85\%$$

Se planificaron y completaron 88 puntos repartidos en múltiples tareas. Todas las tareas están marcadas como FINALIZADAS, por lo tanto, se completó el 100% del trabajo planificado. Sin embargo, por lo poco que se acopla la bajada de los tasks a la proyección esperada se le agrega un factor de 0.85 por estética.

Por esto se considera que el éxito del sprint es de un 85%.

### Discusión del éxito del sprint basado en las métricas.

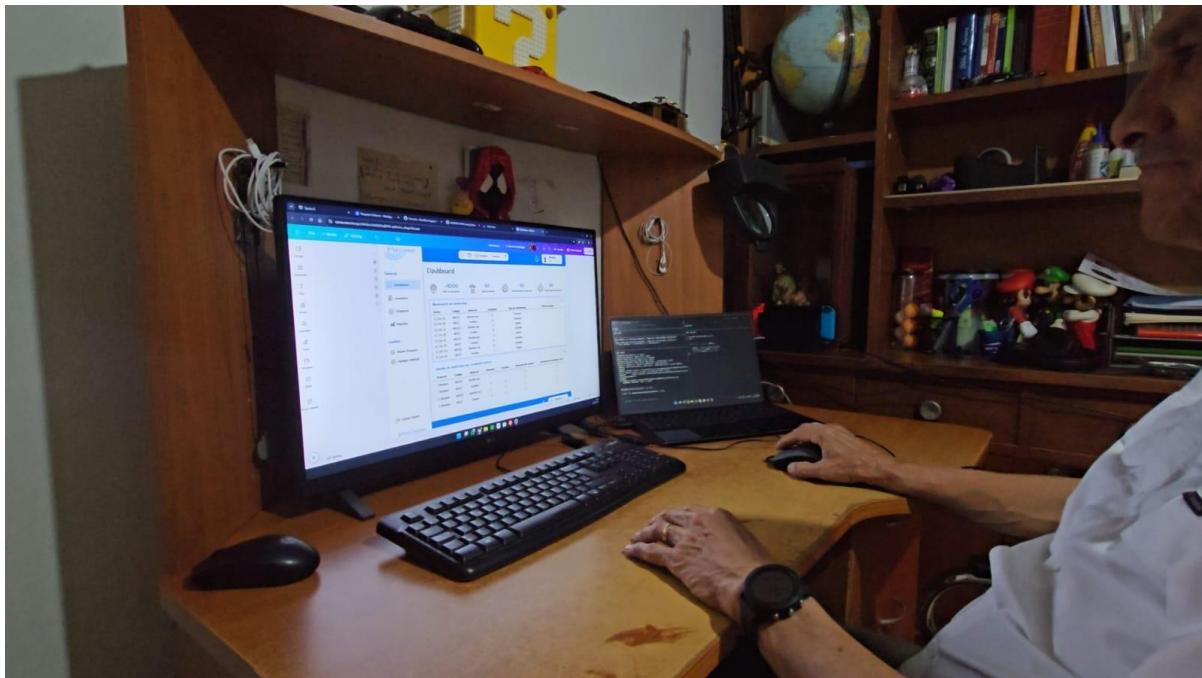
Al observar la gráfica Burndown del Sprint 5, se nota un cumplimiento muy cercano al plan trazado. La línea de trabajo real se mantiene bastante alineada con la línea ideal, especialmente en la segunda mitad del sprint, lo que refleja un ritmo de trabajo constante y bien administrado. Aunque al inicio hubo una ligera acumulación de tareas, el equipo logró recuperarse y cerrar el sprint con prácticamente todos los pendientes completados.

En cuanto a la métrica de velocidad, se muestra un incremento claro con respecto a los sprints anteriores, marcando el Sprint 5 como el de mayor productividad hasta el momento. Esto indica que el equipo ha ganado experiencia, confianza en el proceso ágil y dominio sobre la tecnología y la estructura del proyecto. La entrega de tareas clave como la implementación del sistema de inventario, pruebas con JWT, vistas nuevas y herramientas de testing respalda este aumento de rendimiento.

Ambas métricas evidencian que el Sprint 5 fue un éxito en términos de entrega, ritmo y cumplimiento de objetivos, consolidando un avance sólido del equipo.

### **Evidencias de muestra del incremento desarrollado a usuarios finales y/o product owner.**

[https://youtu.be/WWa3fq\\_jR8k](https://youtu.be/WWa3fq_jR8k)



Ya teniendo una buena idea de cómo se miran las pantallas, está muy satisfecho teniendo algo visual de cómo va a ser el resultado final (esperado) del proyecto. Sabe que puede haber cambios mientras se empiece el proceso de las pantallas como tal, y nos va a ir dando su retroalimentación mientras se vayan creando las pantallas. También quiere ver la funcionalidad que ya eso fue un gran punto de fricción en este sprint, fue también la razón principal por la cual quería todas las pantallas. Por el momento está contento con el avance, y quisiera ver la funcionalidad de algo, pero está feliz con el resultado.

### **Retrospectiva del sprint**

Aspectos positivos:

- Se completaron todas las tareas principales del sprint, destacando la creación de las vistas para el módulo de inventario, así como el rediseño de componentes clave para la gestión de materiales.
- La incorporación de herramientas de testing en frontend y backend fue exitosa y permitió validar funcionalidades críticas como la generación del JWT y el control de acceso a rutas protegidas.
- El equipo logró su mayor velocidad de trabajo hasta ahora, lo que demuestra una mejora notable en organización, colaboración y eficiencia.

- La actualización del modelo de datos y la documentación de la API (vía Swagger) facilitarán el mantenimiento y escalabilidad del sistema.

Aspectos negativos:

- El inicio del sprint fue algo lento en comparación con la planificación, probablemente debido a tareas técnicas que requerían mayor análisis o coordinación.
- Algunas tareas como la documentación y pruebas en backend tomaron más tiempo del estimado, lo que afectó brevemente el ritmo ideal.

El equipo mostró una evolución muy positiva en este sprint, tanto en términos de productividad como de calidad técnica. Las métricas reflejan una ejecución eficaz, con entregables funcionales y un flujo de trabajo bien manejado. A pesar de pequeños ajustes por hacer en estimaciones y balance de carga, el Sprint 5 fue un éxito y demuestra que el equipo está cada vez más consolidado y alineado con los objetivos del proyecto.

Link del excel: [Gestion de tiempos.xlsx](#)

Link del historial de docs: [Sprint 5 - Pool Center.docx](#)