

Universidad del Valle de Guatemala

Facultad de Ingeniería

Departamento de Ciencia de la Computación

Ingeniería de Software II

1966

UNIVERSIDAD

GUATEMALA



## DESIGN BETTER

PLATAFORMA DE DISEÑO Y PERSONALIZACIÓN DE MODA

Pablo José Méndez Alvarado – 23975

Luis Fernando Palacios López – 239333

Roberto Samuel Nájera Marroquín – 23781

André Emilio Pivaral López – 23574

DEL VALLE DE

*Excelencia que trasciende*

**Catedrático:** Erick Francisco Marroquín Rodríguez

**Sección:** 20

Nueva Guatemala de la Asunción, 8 de agosto de 2025

## Índice

|   |    |
|---|----|
| Índice.....   | 1  |
| Product Backlog.....  | 2  |
| Épica 1: Gestión de Usuarios .....                          | 2  |
| Épica 2: Personalización de Prendas .....                   | 2  |
| Épica 3: Visualización Realista (3D) .....                  | 2  |
| Épica 4: Seguimiento de Pedidos .....                       | 3  |
| Épica 5: Comunicación y Colaboración .....                  | 4  |
| Épica 6: Administración y Gestión de Plataforma .....       | 4  |
| Épica 7: Infraestructura Técnica .....                      | 4  |
| Épica 8: Gestión de Pagos .....                             | 5  |
| Épica 9: Despliegue y Publicación del Sistema .....         | 5  |
| Épica 10: Mantenimiento y Escalabilidad .....               | 5  |
| Sprint Backlog .....  | 6  |
| Pila del Sprint .....                                       | 6  |
| Calendario .....  | 9  |
| Resultados del Sprint .....                                 | 11 |
| Software Funcionando.....                                   | 11 |
| Tareas Concluidas .....                                     | 12 |
| Repositorios con el código desarrollado: .....              | 12 |
| Métricas del Sprint.....                                    | 12 |
| Gráfico de Sprint Burndown de puntos de historia .....      | 12 |
| Gráfico de Sprint Burndown de Recuento de Incidencias ..... | 13 |
| Reporte de Velocidad .....                                  | 13 |
| Interpretacion del gráfico BurnDown: .....                  | 14 |
| Plan Maestro .....  | 14 |
| Evidencia de pruebas:.....                                  | 16 |
| Presupuesto .....   | 17 |
| Evidencia de Pruebas con Usuarios finales .....             | 18 |
| Reflexión del Desempeño del Equipo en el Sprint VI.....     | 19 |
| Aspectos a Observar para el Futuro .....                    | 20 |
| Informe general de gestión para el Sprint 6: .....          | 21 |

## Product Backlog

Las tareas subrayadas en rojo son las que fueron completadas durante este Sprint y anteriores.

### Épica 1: Gestión de Usuarios

Prioridad: Alta

Historia de usuario: Como usuario (diseñador o cliente), quiero poder registrarme e iniciar sesión para acceder a funcionalidades específicas de la plataforma.

Tareas:

- Crear formulario de registro de usuarios.
- Implementar autenticación vía correo electrónico (con confirmación).
- Integrar inicio de sesión con redes sociales (Google, Facebook).
- Crear sistema para recuperación de contraseña.
- Desarrollar interfaces diferenciadas para cliente, diseñador y administrador.

### Épica 2: Personalización de Prendas

Prioridad: Alta

Historia de usuario: Como cliente, quiero personalizar prendas según mis preferencias y medidas corporales para recibir productos únicos.

Tareas:

- Crear base de datos de plantillas y moldes estándar personalizables.
- Implementar interfaz interactiva para elección de tipos de prendas, telas, colores y estilos.
- Desarrollar sistema para ingresar medidas personalizadas y almacenarlas.
- Crear el layout básico de la vista de personalización (frontend).
- Implementar selección interactiva de tipos de prendas (menú desplegable, botones).
- Desarrollar selección de colores y materiales usando paletas visuales dinámicas.
- Desarrollar sistema de recomendación inteligente (IA) de estilos según medidas.
- Implementar persistencia temporal de selecciones del usuario usando almacenamiento local (localStorage).
- Conectar frontend con el backend mediante llamadas REST API para almacenar personalizaciones permanentemente.

### Épica 3: Visualización Realista (3D)

Prioridad: Alta

Historia de usuario: Como usuario, quiero previsualizar digitalmente mis prendas personalizadas antes de su confección.

Tareas:

- Investigar e integrar biblioteca para visualizaciones en 3D (por ejemplo, Three.js, Babylon.js).

- Investigar documentación técnica y tutoriales de biblioteca elegida (Three.js o Babylon.js).
- Configurar entorno inicial básico para renderizado 3D en frontend.
- Crear modelo básico estático para prueba inicial de visualización.
- Implementar carga dinámica de modelos 3D desde backend.
- Programar interacción en tiempo real con controles básicos (zoom, rotación, desplazamiento).
- Desarrollar integración con personalizaciones elegidas por usuario (color, material, tamaño).
- Crear función para actualizar visualización inmediatamente al cambiar opciones del usuario.
- Optimizar rendimiento de carga de modelos (carga progresiva, caching).
- Realizar pruebas de rendimiento y usabilidad de la visualización 3D.
- Desarrollar función para visualización dinámica en tiempo real (actualización inmediata tras cambios).
- Implementar creación de avatares personalizados según medidas del cliente.

#### **Épica 4: Seguimiento de Pedidos**

**Prioridad: Media-alta**

**Historia de usuario: Como cliente, deseo conocer en tiempo real el estado de mi pedido desde su creación hasta la entrega.**

#### **Tareas:**

- Crear un sistema visual de tracking del estado de pedidos.
- Desarrollar notificaciones automáticas sobre cambios de estado.
- Implementar historial detallado de modificaciones en cada pedido.
- Definir estados específicos del pedido (e.g., Recibido, En diseño, En producción, Enviado, Entregado).
- Diseñar visualmente el componente gráfico para tracking en el frontend (barra de progreso visual).
- Implementar la lógica en backend para almacenar y modificar estados del pedido.
- Crear API REST para consultar y actualizar estado de pedidos desde frontend.
- Desarrollar frontend para reflejar en tiempo real cambios en los estados del pedido.

- Configurar sistema automático de notificaciones (email y/o push) al usuario al cambiar estados.
- Desarrollar módulo de historial visual para mostrar la trazabilidad completa del pedido.

## **Épica 5: Comunicación y Colaboración**

Prioridad: Alta

Historia de usuario: Como usuario, quiero una comunicación efectiva y directa entre cliente y diseñador.

Tareas:

- Desarrollar un módulo interno para mensajería y envío de archivos adjuntos.
- Implementar chatbot inteligente para responder preguntas frecuentes y derivar a un asesor cuando sea necesario.
- Investigar soluciones tecnológicas existentes para chatbots (por ejemplo, DialogFlow).
- Crear cuenta y configurar entorno básico del chatbot en la plataforma seleccionada.
- Programar flujo básico de conversación con respuestas predefinidas (FAQ inicial).
- Implementar lógica del chatbot para derivar consultas complejas a un asesor humano.
- Integrar chatbot en el frontend del sitio (widget flotante).
- Crear módulo básico de mensajería interna entre usuarios (cliente-diseñador).
- Desarrollar backend para almacenar mensajes y archivos adjuntos en base de datos.
- Crear interfaz gráfica en frontend para intercambio de mensajes en tiempo real (tipo chat).
- Programar notificaciones instantáneas al recibir nuevos mensajes.
- Integrar sección FAQ para consultas rápidas.

## **Épica 6: Administración y Gestión de Plataforma**

Prioridad: Media

Historia de usuario: Como administrador, deseo gestionar usuarios, contenido y realizar análisis de datos sobre el uso de la plataforma.

Tareas:

- Desarrollar panel de administración con gestión de usuarios.
- Implementar herramienta para moderación de diseños publicados.
- Crear módulo de análisis estadístico de uso (dashboard de métricas).

## **Épica 7: Infraestructura Técnica**

Prioridad: Alta

Historia de usuario: Como desarrollador, quiero un entorno de desarrollo replicable usando contenedores para facilitar el despliegue y mantenimiento del sistema.

**Tareas:**

- Instalar y configurar Docker o Podman.
- Crear Dockerfile para el servicio web y base de datos.
- Preparar archivo docker-compose para orquestación de servicios.
- Configurar variables de entorno necesarias para ambos servicios.
- Probar construcción local de imágenes Docker (docker build).
- Escribir documentación técnica (README.md) con pasos de ejecución de contenedores.

## **Épica 8: Gestión de Pagos**

Prioridad: Media

Historia de usuario: Como cliente, quiero pagar de forma segura y rápida por los pedidos realizados.

**Tareas:**

- Investigar plataformas de pago compatibles (PayPal, Stripe, MercadoPago).
- Implementar módulo de pagos integrando la plataforma elegida.
- Configurar entorno backend para comunicación segura con la API de pagos.
- Desarrollar módulo backend para crear transacciones y manejar callbacks/respuestas de pagos.
- Implementar validación de datos y cifrado para garantizar seguridad en transacciones.
- Asegurar cumplimiento legal (facturación SAT Guatemala).
- Documentar claramente los procesos técnicos para futuras auditorías.

## **Épica 9: Despliegue y Publicación del Sistema**

Prioridad: Media-alta

Historia de usuario: Como usuario final, deseo acceder fácilmente a la plataforma desde cualquier dispositivo.

**Tareas:**

- Seleccionar proveedor de hosting en la nube.
- Desplegar aplicación usando servicios en la nube (AWS, Heroku, DigitalOcean).
- Realizar pruebas en entorno de producción.

## **Épica 10: Mantenimiento y Escalabilidad**

Prioridad: Baja-media

Historia de usuario: Como administrador del sistema, quiero garantizar que el sistema sea mantenible y escalable a largo plazo.

Tareas:

- Establecer procesos de backup automático de datos.
- Documentar claramente el código y procedimientos técnicos.
- Implementar monitoreo automático de rendimiento y alertas de caídas.

## **Sprint Backlog**

### **Pila del Sprint**

A continuación, se listan las tareas bajo cada historia de usuario. Se indica: nombre, descripción, horas estimadas, puntos de historia, responsable y fecha probable de finalización.

Estimación total: 8 puntos (11h 55m)

[SCRUMSITO-105] Componentes visuales del frontend para mensajería

Descripción: Creación e integración de elementos visuales para la interfaz de mensajería.

Horas estimadas: 1 h 50 m

Puntos de historia: 1

Responsable: LUIS FERNANDO PALACIOS LOPEZ

Fecha fin: 07/08/2025

[SCRUMSITO-104] APIs de conexión para mensajería

Descripción: Desarrollo de endpoints para conectar el frontend con el módulo de mensajería.

Horas estimadas: 2 h 20 m

Puntos de historia: 1

Responsable: ANDRE EMILIO PIVARAL LOPEZ

Fecha fin: 08/08/2025

[SCRUMSITO-103] Modelos internos del backend para módulo de mensajería

Descripción: Implementación de modelos de datos internos para manejar conversaciones y ensajes.

Horas estimadas: 0 h 45 m

Puntos de historia: 1

Responsable: PABLO JOSE MENDEZ ALVARADO

Fecha fin: 01/08/2025

[SCRUMSITO-102] Conectar APIs de pedidos desde el frontend

Descripción: Integración de llamadas a la API para actualizar y modificar pedidos en tiempo real.

Horas estimadas: 1 h 30 m

Puntos de historia: 1

Responsable: ROBERTO SAMUEL NAJERA MARROQUIN

Fecha fin: 07/08/2025

[SCRUMSITO-101] Organizar componentes del frontend para seguimiento de pedidos

Descripción: Reestructuración de componentes para mejorar la experiencia de seguimiento de pedidos.

Horas estimadas: 2 h 10 m

Puntos de historia: 1

Responsable: ROBERTO SAMUEL NAJERA MARROQUIN

Fecha fin: 08/08/2025

[SCRUMSITO-32] Sección de FAQ para consultas rápidas

Descripción: Creación de una página de preguntas frecuentes para resolver dudas comunes.



Horas estimadas: 1 h 5 m

Puntos de historia: 1

Responsable: ANDRE EMILIO PIVARAL LOPEZ

Fecha fin: 08/08/2025

[SCRUMSITO-31] Chatbot inteligente para FAQs y derivación a asesor

Descripción: Implementación de un chatbot que responde preguntas y canaliza a un agente cuando es necesario.

Horas estimadas: 1 h 25 m

Puntos de historia: 1

Responsable: PABLO JOSE MENDEZ ALVARADO

Fecha fin: 06/08/2025

[SCRUMSITO-30] Módulo interno de mensajería con archivos adjuntos

Descripción: Desarrollo de mensajería interna con soporte para envío y recepción de archivos.

Horas estimadas: 0 h 50 m

Puntos de historia: 1

Responsable: LUIS FERNANDO PALACIOS LOPEZ

Fecha fin: 07/08/2025

User Story Mapping

|                     |  |                            |  |                                      |                                   |                                |                                   |                           |
|---------------------|--|----------------------------|--|--------------------------------------|-----------------------------------|--------------------------------|-----------------------------------|---------------------------|
| Usuario Actividades | Actividades                                |                            |  |                                      |                                   |                                |                                   |                           |
|                     | Diseñar una prenda                         | Diseñador Independiente    | Personalizar una prenda                  | Cliente final                        | Gestionar la Plataforma           | Administrador                  | Capacitar Usuarios                | Diseñador Novato          |
| Usuario Tareas      | Tareas                                     |                            |  |                                      |                                   |                                |                                   |                           |
|                     | Crear un Diseño Base                       | Simular en 3D              | Elegir características                   | Validar diseño                       | Mantener Seguridad                | Analizar rendimiento           | Aprender Herramientas             | Resolver Problemas        |
| Prioridad 1         | Subtareas                                  |                            |  |                                      |                                   |                                |                                   |                           |
|                     | Seleccionar plantilla según tipo de cuerpo | Cargar modelo en 3D        | Seleccionar telas                        | Visualizar en 3D con medidas propias | Actualizar permisos de usuarios   | Monitorear tiempos de carga    | Completar tutorial interactivo    | Contactar Soporte Técnico |
| Prioridad 2         | Crear un nuevo documento                   | Aplicar texturas y colores | Añadir detalles                          |                                      | Revisar alertas de contenido      |                                |                                   | Reportar errores críticos |
|                     | Añadir detalles estéticos                  | Ajustar Proporciones       | Comparar opciones con diseños anteriores | Enviar comentarios al diseñador      | Realizar copias de seguridad      | Identificar errores frecuentes | Practicar con funciones básicas   | Revisar FAQ actualizadas  |
| Prioridad 3         |  |                            | Guardar Preferencias                     |                                      |                                   |                                | Explorar funciones avanzadas      | Restablecer contraseña    |
|                     | Exportar a diseño editable                 | Enviar al Cliente          | Consultar costos adicionales             | Descargar imagen de referencia       | Documentar políticas de seguridad | Implementar actualizaciones    | Acceder a bibliotecas de recursos | Sugerir mejoras           |

Calendario

Q Buscar en ...

Persona asignada

Tipo

Estado

Más filtros

Hoy

< ago 2025 >

| Lun  | Mar | Mié | Jue   | Vie   |
|--|-----|-----|---|-------|
| 28   | 29  | 30  | 31  | ago 1 |
|  |     |     | Tablero Sprint 6                                |       |
|  |     |     | SCRUMSITO-102 Cone...<br>SCRUMSITO-104 Organ... |       |
| 4  | 5   | 6   | 7   | 8     |
| Tablero Sprint 6                               |     |     |   |       |
| SCRUMSITO-105 Comp...<br>SCRUMSITO-30 modul... |     |     |   |       |
|  |     |     |   |       |
| 11   | 12  | 13  | 14  | 15    |
|  |     |     |   |       |
| 18   | 19  | 20  | 21  | 22    |

Quickstart

Evidencia del uso de las técnicas de scrum.

Atlassian uses cookies to improve your browsing experience, perform analytics and research, and conduct advertising. Accept all cookies to indicate that you agree to our use of cookies on your device. [Atlassian cookies and tracking notice](#)

Preferences Only necessary ✓ Accept all

Jira

Proyectos

design-better

Resumen Cronograma Backlog Tablero Calendario Informes Lista Formularios Metas More 6 +

Buscar e... Epic Tipo

SCRUMSITO-30 modulo interno de mensajería con archivos adjuntos TAREAS POR HACER

SCRUMSITO-32 seccion de FAQ para consultas rápidas TAREAS POR HACER

SCRUMSITO-31 chatbot inteligente para preguntas frecuentes y derivar a asesor cuando ... TAREAS POR HACER

SCRUMSITO-103 modelos internos del backend para modulo de mensajería TAREAS POR HACER

SCRUMSITO-49 Como usuario, quiero una comunicación efectiva y directa entre cliente y... TAREAS POR HACER

SCRUMSITO-104 APIs de conexión para mensajería TAREAS POR HACER

SCRUMSITO-105 Componentes visuales del frontend para mensajería TAREAS POR HACER

+ Crear

9 activ Quickstart

design-better.atlassian.net/jira/software/projects/SCRUMSITO/list

Jira

Proyectos

design-better

Resumen Cronograma Backlog Tablero Calendario Informes Lista Formularios Metas

Buscar en la lista Filtro

|   | Tipo | Clave       | Resumen  | Estado     | Comentarios                       |
|---|------|-------------|--|------------|-----------------------------------|
| # | +    | SCRUMSIT... | Documentacion  | FINALIZADA | Añadir comentario Tablero Spri... |
|   |      | SCRUMSIT... | Métricas Sprint  | FINALIZADA | Añadir comentario Tablero Spri... |
|   |      | SCRUMSIT... | Resultados Sprint  | FINALIZADA | Añadir comentario Tablero Spri... |
|   |      | SCRUMSIT... | Solucionar coordinacion de arranque inicial de contenedor... | FINALIZADA | Añadir comentario Tablero Spri... |
|   |      | SCRUMSIT... | Calendario   | FINALIZADA | Añadir comentario Tablero Spri... |
|   |      | SCRUMSIT... | Implementar diseño nuevo y más agradable                     | FINALIZADA | Añadir comentario Tablero Spri... |
|   |      | SCRUMSIT... | Modularizar código actual en el frontend                     | FINALIZADA | Añadir comentario Tablero Spri... |
|   |      | SCRUMSIT... | Resolver problemas con interfaces difereneciadas             | FINALIZADA | Añadir comentario Tablero Spri... |

+ Crear

Quickstart

Enlace de Jira: <https://design-better.atlassian.net/jira/software/projects/SCRUMSITO/boards/1?atlOrigin=eyJpIjoiNjY0OGUzYjk5NzU0NGQ5YmFhMzk5NTZiYjZhMDVhMTA1LCJwIjoiajI9>

Carpetas de grabaciones: [Evidencias Scrum](#)

## Resultados del Sprint

### Software Funcionando

DesignBetter

[Inicio](#) [Mis Pedidos](#) [Chatbot](#)

Cerrar sesión

## Hola, usuario

Revisa el estado de tus pedidos y diseños en proceso

### Resumen de Pedidos

Pendientes  
0

En proceso  
0

Completados  
0

Acciones rápidas  
[Nuevo pedido](#)

Asistente de Moda  
Especialista en Diseño

¡Hola! Soy tu asistente virtual especializado en moda. ¿En qué puedo ayudarte hoy?

Escribe tu pregunta sobre moda, tallas,

Presiona Enter para enviar • Shift+Enter para nueva línea

DesignBetter

[Inicio](#) [Mis Pedidos](#) [Chatbot](#)

Cerrar sesión

## Tus Pedidos

Revisa el estado y historial de tus pedidos

Filtrar por

Todos los pedidos

Pendientes

En proceso

Completados

-

8 de agosto de 2025

Diseñador: Sin asignar

Pendiente

+  
Nuevo pedido

DesignBetter

[Inicio](#) [Mis Pedidos](#) [Chatbot](#)

Cerrar sesión

DesignBetter

[Inicio](#) [Mis Pedidos](#) [Chatbot](#)

Cerrar sesión

Tareas Concluidas

|                          |                                     |             |   |            |                   |                 |
|--------------------------|-------------------------------------|-------------|---|------------|-------------------|-----------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Cambiar formato de json en el ingreso del patrón a un mo...     | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Arreglar la barra de navegación del cliente                     | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Reparar selección de opciones en pedido para que no se r...     | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Corregir contraste de colores de los textos en el modo os...    | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Arreglar presentación de opciones en solicitud de pedido        | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Añadir una vista de cargado cuando se hacen solicitudes ...     | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | En el formulario de registro de patrón, incluir entrada de g... | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Arreglar el desglose de la vista del dashboard por rol para...  | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Añadir pantalla donde avisa el correo de confirmación al r...   | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Añadir un menú de opciones, algunas sin implementar tod...      | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Implementar APIs para los formularios en el frontend            | FINALIZADA | Añadir comentario | Tablero Spri... |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | SCRUMSIT... | Corregir ingreso de medidas                                     | FINALIZADA | Añadir comentario | Tablero Spri... |

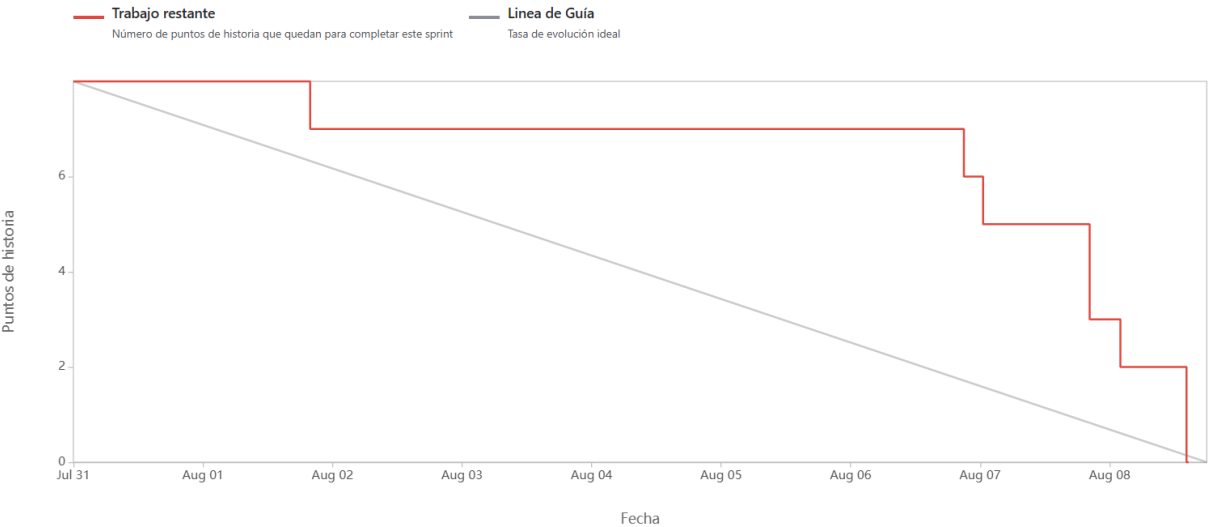
No hubieron tareas planificadas y no concluidas para este Sprint

Repositorios con el código desarrollado:  
<https://github.com/Ultimate-Truth-Seeker/DesignBetterFrontend>  
<https://github.com/Ultimate-Truth-Seeker/DesignBetterBackend>

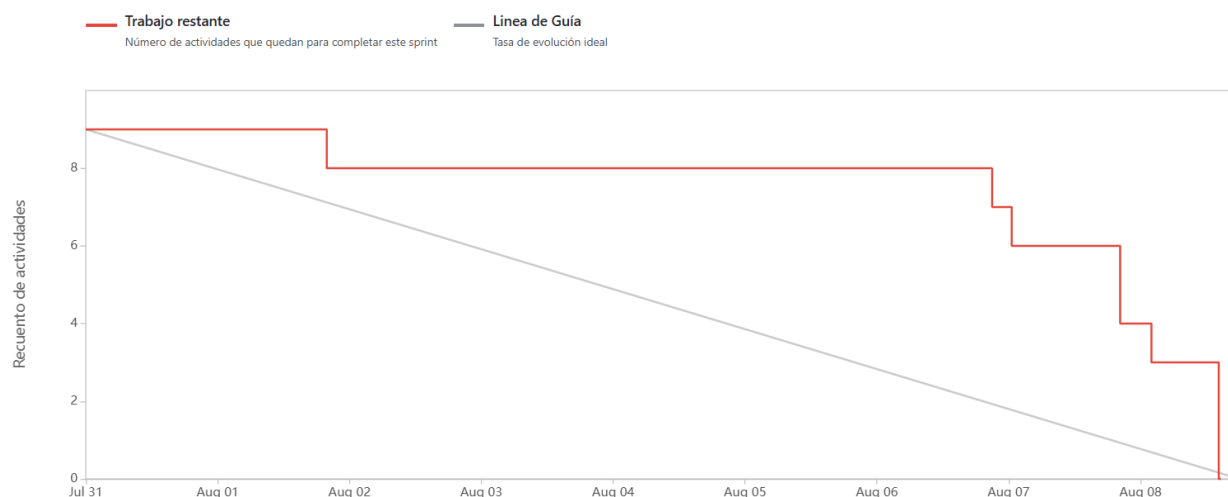
Métricas del Sprint

Gráfico de Sprint Burndown de puntos de historia

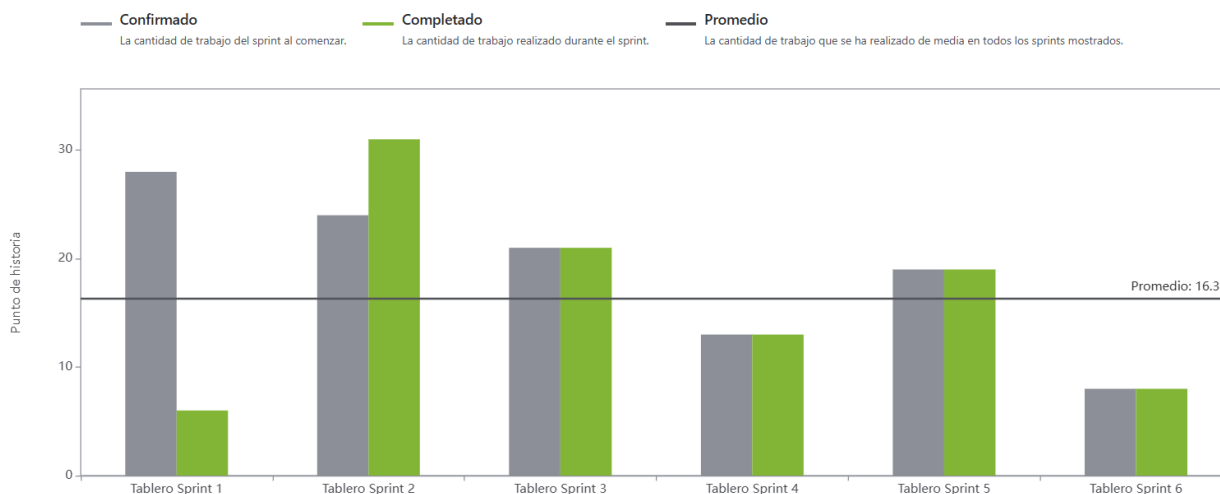
Fecha - 31 de julio de 2025 - 8 de agosto de 2025



## Gráfico de Sprint Burndown de Recuento de Incidencias



## Reporte de Velocidad



## Éxito del Sprint

Para evaluar de manera objetiva el rendimiento del equipo durante el Sprint, se utilizó un indicador de éxito compuesto por dos factores: el cumplimiento de todas las tareas (60%) y la entrega puntual de cada una (40%).

### Ecuación

$$\text{Indicador de Éxito} = (\text{Tareas completadas} \times 0.6) + (\text{Promedio de puntualidad} \times 0.4)$$

$$= (100 \times 0.6) + (67.5 \times 0.4) = 95.67\%$$

- Tareas comprometidas al inicio del sprint: 8
- Tareas completadas: 8
- Puntaje promedio por cumplimiento de fechas: 67.5/100

El Sprint 6 se completó con el 100 % de las tareas comprometidas entregadas (8/8), alcanzando un Indicador de Éxito del 87.5 %. Este resultado refleja que, aunque la totalidad del trabajo fue finalizada, la puntualidad promedio (67.5 %) redujo la calificación global. La planificación inicial se vio afectada por un inicio lento, seguido de un ritmo acelerado en la segunda mitad del sprint, lo que permitió recuperar el avance y cerrar con éxito.

Este comportamiento evidencia una acumulación de tareas hacia el final (efecto “ola final”), lo que si bien permitió cumplir con el alcance, genera riesgos como sobrecarga de trabajo, disminución de calidad por presión temporal o menor capacidad de reacción ante bloqueos de último minuto.

### Interpretacion del gráfico BurnDown:

El gráfico Burndown muestra un avance horizontal seguida de una caída fuerte en el avance de las tareas, lo cual implica impuntualidad y poco avance temprano en el desarrollo del sprint. Esto significa que se deben de mejorar las prácticas del progreso de las tareas para evitar este tipo de resultados en las gráficas, y se concluye que hubo algunos fallos en la ejecución de la consistencia de las entregas de tareas del sprint.

### Plan Maestro

| TC ID# | Escenario/Condición   | Carga de Trabajo | Variables específicas   | Resultado Esperado   |
|--------|---|------------------|---|--|
| TC001  | Registro de usuario con datos válidos   | 1 usuario        | <a href="#">correo_electronico=cliente1@example.com</a> ;<br>nombre=Cliente Uno;<br>password=Segura123!;<br>rol=cliente | HTTP 201 Created; se envía 1 correo de activación; usuario creado inactivo en BD |
| TC002  | Intentar login antes de activar la cuenta                                     | 1 usuario        | <a href="#">correo=cliente1@example.com</a> ; password=Segura123!   | HTTP 401 Unauthorized; mensaje “Cuenta no activada” o similar                    |
| TC003  | Registro con correo ya existente  | 1 usuario        | <a href="#">correo_electronico=cliente1@example.com</a> (ya en BD)  | HTTP 400 Bad Request; mensaje “Correo ya registrado”                             |
| TC004  | Registro con contraseña que no cumple políticas (muy corta o sin complejidad) | 1 usuario        | password=abc  | HTTP 400 Bad Request; mensaje “Contraseña demasiado débil”                       |

|       |  |           |   |  |
|-------|--|-----------|---|--|
| TC005 | El diseñador genera y guarda un nuevo patrón con todos los campos válidos                | 1 usuario | diseñador autenticado;<br>payload: nombre='Vestido A',<br>geometría=datos válidos,<br>observaciones='...' | HTTP 201 Created;<br>PlantillaPrenda creada en<br>BD con esos valores y<br>asociada al diseñador       |
| TC006 | Intentar guardar un patrón faltando un campo obligatorio (p.ej. geometría o tipo_cuerpo) | 1 usuario | payload sin 'tipo_cuerpo'   | HTTP 400 Bad Request;<br>mensaje "tipo_cuerpo es<br>obligatorio"                                       |
| TC007 | Cliente personaliza plantilla existente con valores válidos                              | 1 usuario | cliente autenticado;<br>plantilla=1; color='Rojo';<br>medidas_usuario={busto:90,ci<br>ntura:70,cadera:95} | HTTP 200 OK; respuesta<br>con vista previa JSON de la<br>personalización y cálculo de<br>medidas       |
| TC008 | Cliente crea un nuevo pedido sobre plantilla personalizada                               | 1 usuario | cliente autenticado;<br>plantilla=1; disenador=2;<br>ajustes='Sin ajuste'; notas='...'                    | HTTP 201 Created;<br>data.estado='pendiente';<br>data.pago_realizado=false;<br>data.usuario=cliente.id |
| TC009 | Cliente intenta modificar el estado de su propio pedido                                  | 1 usuario | cliente autenticado;<br>pedido_id=10; payload<br>estado='diseno'  | HTTP 403 Forbidden o 404   |
| TC010 | Diseñador cambia estado de pedido de 'pendiente' a 'diseno' y se registra en historial   | 1 usuario | disenador autenticado;<br>pedido_id=11; payload<br>estado='diseno'  | HTTP 200 OK; en BD<br>Pedido.estado='diseno';<br>registro en<br>PedidoEstadoHistoria                   |
| TC011 | Diseñador marca pago de pedido como realizado  | 1 usuario | disenador autenticado;<br>pedido_id=12; payload<br>pago_realizado=True                                    | HTTP 200 OK;<br>Pedido.pago_realizado=True<br>e en BD  |
| TC012 | Cliente lista solo sus pedidos (listado y detalle)                                       | 1 usuario | cliente1 autenticado; cliente2<br>con pedidos distintos   | GET /listar-pedidos<br>devuelve solo pedidos de<br>cliente1; GET /detalle-<br>pedido/{otro} -> 403     |
| TC013 | Cliente y diseñador con un pedido en común inician conversación en módulo de mensajería  | 1 usuario | cliente1, disenador1,<br>pedido_id=13   | HTTP 200 OK al POST<br>/mensajes; mensaje<br>aparece en GET<br>/conversacion/13                        |
| TC014 | Usuario sin pedido en común intenta enviar mensaje                                       | 1 usuario | otro_cliente autenticado;<br>mismo payload que TC013  | HTTP 403 Forbidden   |



|       |  |           |  |  |
|-------|--|-----------|--|--|
| TC015 | Generar previsualización 3D válida con medidas correctas                             | 1 usuario | cliente autenticado; pedido_id=14; medidas={alto:180, ancho:50...} | HTTP 200 OK; respuesta con URL o data del modelo 3D; canvas 3D sin errores |
| TC016 | Intentar generar previsualización 3D sin proporcionar todas las medidas obligatorias | 1 usuario | payload medidas incompletas  | HTTP 400 Bad Request; mensaje 'Faltan datos de medida: cintura, cadera...' |
| TC017 | Listar preguntas frecuentes (FAQ)  | 1 usuario | —  | HTTP 200 OK; array de FAQs con campos {pregunta, respuesta}                |
| TC018 | Buscar en FAQs usando palabra clave  | 1 usuario | query='envío'  | HTTP 200 OK; solo FAQs que contengan 'envío'                               |

### Evidencia de pruebas:

A continuación se presentan la implementación de los tests de funcionalidad de los primeros casos de uso relacionados a la autenticación y registro de usuarios. Se presentan los resultados obtenidos tanto en frontend como en backend:

The screenshot shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with folders like 'app', 'dashboard', 'admin', 'chatbot', 'cliente', 'configuracion', 'mensajes', 'pagos', 'pedidos', 'layout.tsx', 'page.tsx', 'disenadores', and 'diseñador'. The file 'authFlow.test.tsx' is open in the editor, showing a Jest test for a registration flow. The test is a 'describe' block with an 'it' block that simulates a user registration process using Cypress commands like 'type', 'click', and 'getByRole'. The terminal at the bottom shows the output of running the tests, indicating that all tests passed.

```

describe('Flujo de registro', () => {
  it('permite registrarse, activar cuenta y entrar al dashboard', async () => {
    const user = useEvent.setup();
    render(
      <MemoryRouter initialEntries={['/']}>
        <GoogleAuthProvider clientId="test-client-id">
          <RegisterForm />
        </GoogleAuthProvider>
      </MemoryRouter>,
    );

    /* Completar formulario de registro */
    await user.type(screen.getByLabelText(/correo electrónico/i), 'cliente@example.com');
    await user.type(screen.getByLabelText(/contraseñas/i), 'Segura123!');
    await user.type(screen.getByLabelText(/confirmar contraseña/i), 'Segura123!');
    await user.click(screen.getByRole('checkbox', { name: /acepto/i }));
    await user.click(screen.getByRole('button', { name: /registrarse/i }));
  });
});

```

```

found 0 vulnerabilities
root@44422a36f051:/app# npm run test

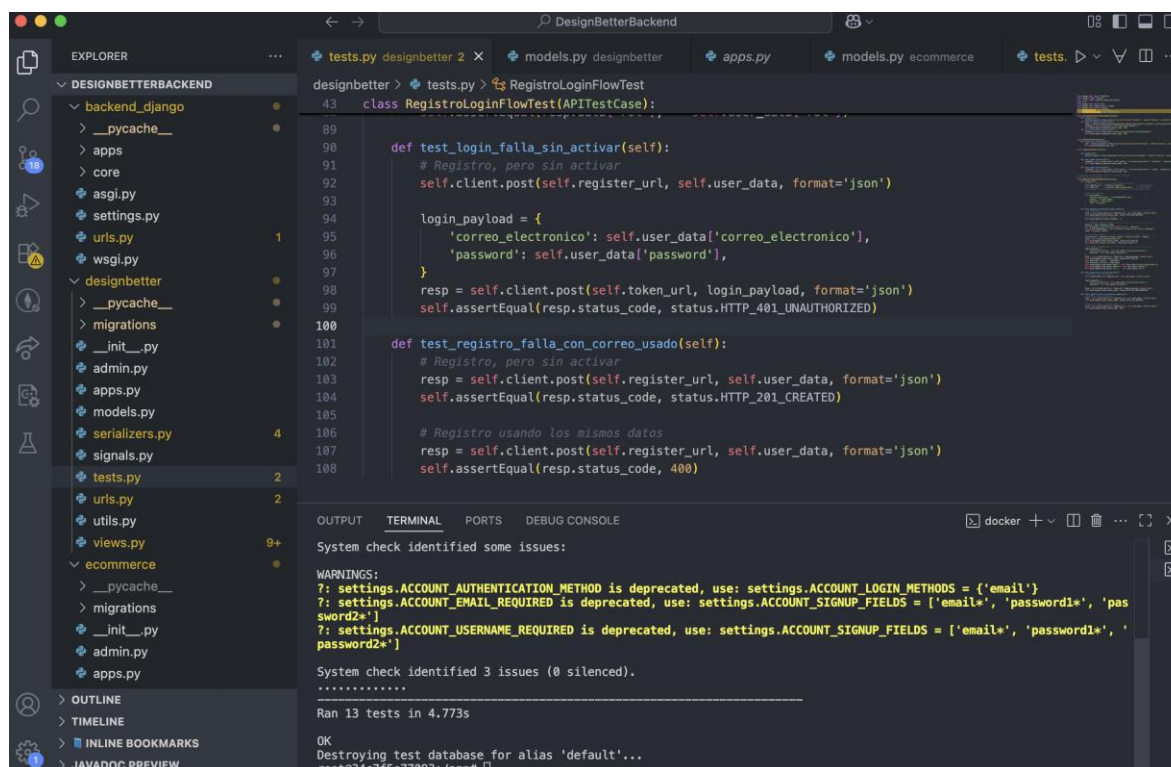
> frontend@0.1.0 test
> vitest

DEV v3.2.4 /app
✓ __tests__/project.test.tsx (3 tests) 28ms
✓ __tests__/authFlow.test.tsx (3 tests) 412ms

Test Files  2 passed (2)
Tests       6 passed (6)
Start at    05:49:36
Duration   981ms (transform 173ms, setup 39ms, collect 370ms, tests 440ms, environment 392ms, prepare 73ms)

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

```



## Presupuesto

En el presente presupuesto, se detalla la estimación de costos asociados al desarrollo del proyecto Design Better, abarcando tanto el componente de Backend como el de Frontend, con base en la tarifa por hora, la cantidad de horas laboradas y el tiempo transcurrido de ejecución.

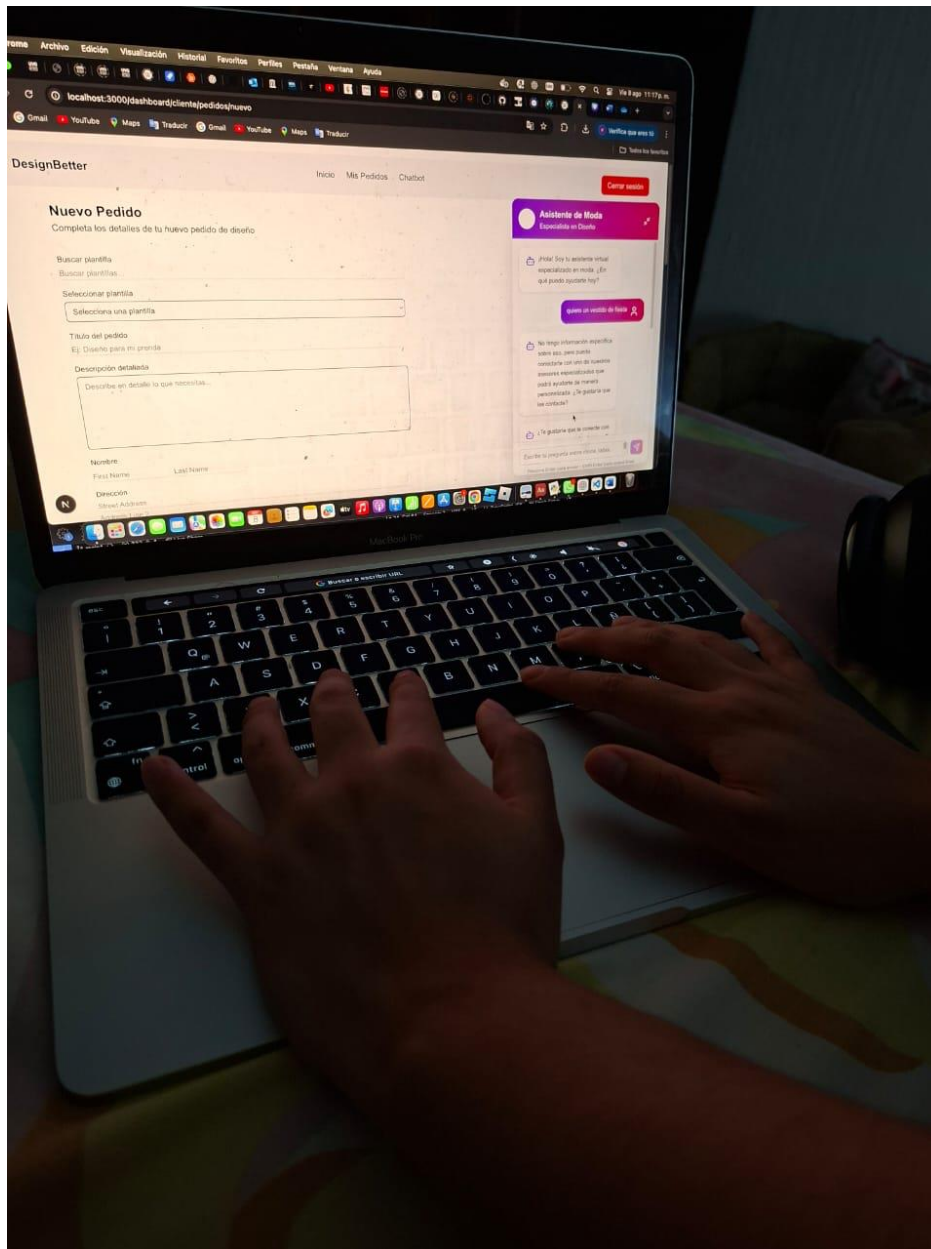
En primera instancia, se establece la tarifa unitaria por hora de trabajo. El valor acordado es de 15 USD por hora, que, considerando un tipo de cambio de 7.8 quetzales por dólar estadounidense, se traduce a Q117 por hora. Este valor servirá como referencia estándar para el cálculo de los costos mensuales y acumulados del equipo.

En cuanto a la carga laboral mensual, se proyecta que cada desarrollador destine 3 horas diarias, 5 días a la semana, durante un total de 4 semanas, lo que representa 60 horas de trabajo por mes por persona. El equipo de desarrollo está conformado por cuatro integrantes, generando así un total de 240 horas mensuales.

El costo mensual se determina multiplicando las 240 horas por la tarifa de Q117/hora, obteniéndose un monto de Q28,080. Considerando un periodo de ejecución de 7.5 meses a la fecha, el gasto acumulado asciende a Q210,600, resultado de multiplicar el costo mensual por el número de meses transcurridos.

Adicionalmente, se incorpora un indicador porcentual de ejecución presupuestaria. Este se calcula dividiendo el gasto acumulado entre el presupuesto total asignado al proyecto y multiplicando el resultado por 100. De manera complementaria, se analiza el tiempo transcurrido en función de la duración total planificada de 10 meses, evidenciándose que, con 7.5 meses ejecutados, se ha consumido un 75% del tiempo disponible.

## Evidencia de Pruebas con Usuarios finales



Usuario de prueba: diseñadora Gaby Nájera

Fotografía usada con su permiso

Fecha: 8 de agosto a las 23:00

Resumen de comentarios:

- Menciono que se podían añadir algunas medidas adicionales al formulario haciendo referencia a diferentes talles que se podían dar

- Le pareció correcto la idea de tener solamente la información del pago y no realizar directamente la compra en línea por facilidad y seguridad
- Le agradó bastante la idea del chatbot que se implementó de ayuda y le pareció divertido e interesante
- Menciona que en relación a la solicitud de pedidos, algunas prendas pueden requerir ciertas reglas o atributos especiales y sería importante considerar esta lógica en la entrega final
- Menciona que el precio puede variar dependiendo de las opciones específicas de cada prenda, por lo que al ejecutar el pedido se debe de considerar este aspecto

## **Reflexión del Desempeño del Equipo en el Sprint VI**

Durante el Sprint VI, el equipo tuvo un inicio más lento en la ejecución, con pocos puntos completados en los primeros días. Un período intermedio con baja actividad, que podría estar asociado a dependencias técnicas, análisis prolongados o tareas de mayor complejidad que retrasaron la entrega de resultados visibles. Un cierre acelerado, con la mayor parte del avance concentrado en la última parte del sprint. La no ejecución desde el primer día redujo la ventana de trabajo efectiva, obligando a un mayor esfuerzo en el tramo final. Algunas actividades críticas pudieron haberse subestimado en tiempo o dificultad, retrasando entregas intermedias.

Aspectos Clave Para Mejorar para el Próximo Sprint:

### **1. Inicio efectivo desde el Día 1**

El sprint no comenzó con un ritmo alto de trabajo desde el primer día, reduciendo la ventana real de ejecución. Esto obligó a comprimir actividades hacia el final, aumentando la presión y el riesgo de errores. Por lo que se sugiere establecer una reunión inicial de arranque para asignar y aclarar todas las tareas desde el minuto cero.

### **2. Mejor gestión del flujo de trabajo**

La curva de trabajo muestra picos y valles de actividad, con largos períodos sin entregas visibles y un cierre muy cargado. Un flujo irregular dificulta la detección temprana de retrasos y aumenta la probabilidad de acumulación de problemas hacia el final. Por lo que proponemos implementar mini-revisiones intermedias (cada 3-4 días) para asegurar que haya entregas parciales de valor constante.

### **3. Desglose más fino de tareas complejas**

Algunas tareas probablemente fueron más complejas de lo previsto y se mantuvieron abiertas varios días sin avances visibles. Las tareas grandes esconden el progreso real, impiden entregas parciales y dificultan reasignaciones rápidas si hay bloqueos. Por lo que sería mejor si se asignaran pares técnicos para las tareas críticas, de forma que se pueda avanzar aunque un miembro tenga problemas o no esté disponible.

#### 4. Refuerzo en puntualidad

Aunque todas las tareas se completaron, solo se obtuvo un 67.5 % en cumplimiento de fechas previstas. Esto indica que las entregas se retrasaron respecto a los plazos internos definidos. La baja puntualidad reduce la confiabilidad del equipo ante compromisos externos y puede condicionar el desempeño en sprints con dependencias fuertes. Por lo que una acción sugerida es analizar después de cada retraso la causa raíz (subestimación, dependencia externa, bloqueo técnico, prioridad cambiada). Y monitorear diariamente el semáforo de puntualidad (verde: en fecha, amarillo: en riesgo, rojo: retrasado).

### Aspectos a Observar para el Futuro

#### 1. Arranques tardíos del sprint

El sprint no comenzó con productividad desde el primer día. Aunque este retraso pudo haber sido útil para preparación, documentación o análisis, es importante cuantificar cuánto valor se pierde o retrasa por no activar el trabajo técnico desde el inicio. Esto puede optimizarse sin perder planificación.

Recomendación: aprovechar el día uno para tareas de baja carga o rápida ejecución, lo que genera inercia positiva desde el inicio.

#### 2. Sustentabilidad del ritmo en el próximo ciclo

Habiendo alcanzado un rendimiento alto justo antes del corte, es importante asegurar continuidad después del mes de descanso. Hay riesgo de “desacondicionamiento” si no se retoman rápidamente los hábitos y rituales ágiles.

Recomendación: abrir el siguiente sprint con una breve recapitulación, revisión de backlog y objetivos compartidos para reactivar el compromiso colectivo.

El Sprint 6 reafirma que el equipo tiene la capacidad técnica y colaborativa para cumplir todo lo planificado, incluso en condiciones de arranque tardío. Sin embargo, la puntualidad y la distribución del trabajo son áreas críticas para optimizar. Mantener el ritmo constante desde el

inicio y asegurar entregas intermedias permitirá elevar el Indicador de Éxito por encima del 90 %, reduciendo riesgos y mejorando la previsibilidad de resultados.

### **Informe general de gestión para el Sprint 6:**

Fecha de Inicio: jueves 31 de julio de 2025

Fecha de finalización: viernes 8 de agosto de 2025

Descripción de tareas propuestas y completadas: desglosadas en el sprint

Los formularios LOGT para el mismo se añadieron dentro de la carpeta específica dentro del repositorio de entrega:

<https://github.com/Ultimate-Truth-Seeker/ProyectoIS>

Los vínculos de los documentos de entregas anteriores se pueden consultar viendo el historial de commits para el README.md

Historial de versiones del documento:

[Sprint VI.docx](#)