**Universidad del Valle de Guatemala**

Facultad de Ingeniería

Departamento de Ciencia de la Computación

Ingeniería de Software II

**DESIGN BETTER**

PLATAFORMA DE DISEÑO Y PERSONALIZACIÓN DE MODA

Pablo José Méndez Alvarado – 23975

Luis Fernando Palacios López – 23933

Roberto Samuel Nájera Marroquín – 23781

André Emilio Pivaral López – 23574

**Catedrático:** Erick Francisco Marroquín Rodríguez

**Sección:** 20

Nueva Guatemala de la Asunción, 26 de septiembre de 2025

**Índice**

[Índice 1](#_Toc553612547)

[Product Backlog 2](#_Toc65281971)

[Épica 1: Gestión de Usuarios 3](#_Toc150861901)

[Épica 2: Personalización de Prendas 3](#_Toc1672189090)

[Épica 3: Visualización Realista (3D) 3](#_Toc181717923)

[Épica 5: Comunicación y Colaboración 5](#_Toc939246828)

[Épica 6: Administración y Gestión de Plataforma 5](#_Toc2106525092)

[Épica 7: Infraestructura Técnica 5](#_Toc794809052)

[Épica 8: Gestión de Pagos 6](#_Toc1618921009)

[Épica 9: Despliegue y Publicación del Sistema 6](#_Toc684506902)

[Épica 10: Mantenimiento y Escalabilidad 6](#_Toc390482520)

[Sprint Backlog 7](#_Toc1472679503)

[Pila del Sprint 7](#_Toc886439062)

[Calendario 10](#_Toc1291673311)

[Carpeta de grabaciones de reuniones diarias: 10](#_Toc2060513647)

[Enlace de Jira: 10](#_Toc1842071314)

[Resultados del Sprint 10](#_Toc947995750)

[Software Funcionando 11](#_Toc1509190801)

[Tareas Concluidas 11](#_Toc1024286568)

[Repositorios con el código desarrollado: 11](#_Toc727739752)

[Métricas del Sprint 12](#_Toc952743172)

[Gráfico de Sprint Burndown de puntos de historia 12](#_Toc2078638312)

[Gráfico de Sprint Burndown de Recuento de Incidencias 12](#_Toc1613277262)

[Reporte de Velocidad 12](#_Toc1133777065)

[Interpretacion del gráfico BurnDown: 13](#_Toc878978796)

[Pruebas de UX y resultados 13](#_Toc475701685)

[Calendario y responsable Asignado 13](#_Toc1308489176)

[Documentación de métodos para realizar las pruebas 13](#_Toc1619503897)

[Pilestone Color Vision Simulator 13](#_Toc1777394554)

[Pruebas Realizadas 13](#_Toc2007443200)

[Resultados, Hallazgos y Conclusiones 13](#_Toc1800049569)

[Pruebas de Carga, Estres y Seguridad 13](#_Toc2132894266)

[Pruebas de Carga y Estres 13](#_Toc159857202)

[Pruebas de Seguridad 14](#_Toc1032941648)

[Refactorización 16](#_Toc1759064101)

[Presupuesto 16](#_Toc1129150409)

[Evidencia de Pruebas con Usuarios finales 17](#_Toc1482449233)

[Reflexión del Desempeño del Equipo en el Sprint VIII 18](#_Toc1718659161)

[Informe general de gestión para el Sprint VIII: 19](#_Toc1870753729)

# 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.
* 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: 12 puntos (16h 45m)

[SCRUMSITO-118] Implementar nuevas variantes de tests para las funcionalidades   
Descripción: Generación de pruebas adicionales para ampliar la cobertura y robustez del sistema.   
Horas estimadas: 1 h 35 m   
Puntos de historia: 1   
Responsable: ROBERTO SAMUEL NAJERA MARROQUIN   
Fecha fin: 22/08/2025

[SCRUMSITO-117] Crear mockups de componentes reutilizables   
Descripción: Diseño de mockups de componentes visuales que servirán de base para futuros refactorizados.   
Horas estimadas: 1 h 50 m   
Puntos de historia: 1   
Responsable: LUIS FERNANDO PALACIOS LOPEZ   
Fecha fin: 23/08/2025

[SCRUMSITO-116] Crear componentes básicos generales con props en base a los mockups   
Descripción: Desarrollo de componentes genéricos y parametrizables siguiendo los mockups establecidos.   
Horas estimadas: 1 h 25 m   
Puntos de historia: 1   
Responsable: LUIS FERNANDO PALACIOS LOPEZ   
Fecha fin: 22/08/2025

[SCRUMSITO-115] Añadir pantallas no implementadas de visualización 3D y creación de patrones   
Descripción: Implementación de vistas adicionales para soporte de modelado 3D y patrones textiles.   
Horas estimadas: 2 h 00 m   
Puntos de historia: 1   
Responsable: PABLO JOSE MENDEZ ALVARADO   
Fecha fin: 28/08/2025

[SCRUMSITO-114] Crear componentes para actualización de tracking y estado de pago   
Descripción: Desarrollo de componentes que permiten al cliente y diseñador gestionar tracking y pagos.   
Horas estimadas: 1 h 20 m   
Puntos de historia: 1   
Responsable: PABLO JOSE MENDEZ ALVARADO   
Fecha fin: 25/08/2025

[SCRUMSITO-113] Refactorizar pantallas de diseñador   
Descripción: Reestructuración y optimización de las pantallas de diseñador para mejorar usabilidad y consistencia.   
Horas estimadas: 1 h 40 m   
Puntos de historia: 1   
Responsable: LUIS FERNANDO PALACIOS LOPEZ   
Fecha fin: 27/08/2025

[SCRUMSITO-112] Refactorizar pantalla de cliente   
Descripción: Mejora y limpieza de código en la pantalla de cliente para asegurar una experiencia más fluida.   
Horas estimadas: 1 h 15 m   
Puntos de historia: 1   
Responsable: ROBERTO SAMUEL NAJERA MARROQUIN   
Fecha fin: 25/08/2025

[SCRUMSITO-111] Refactorizar pantalla de inicio   
Descripción: Ajuste de la pantalla principal para optimizar navegación y desempeño visual.   
Horas estimadas: 1 h 45 m   
Puntos de historia: 1   
Responsable: ANDRE EMILIO PIVARAL LOPEZ   
Fecha fin: 24/08/2025

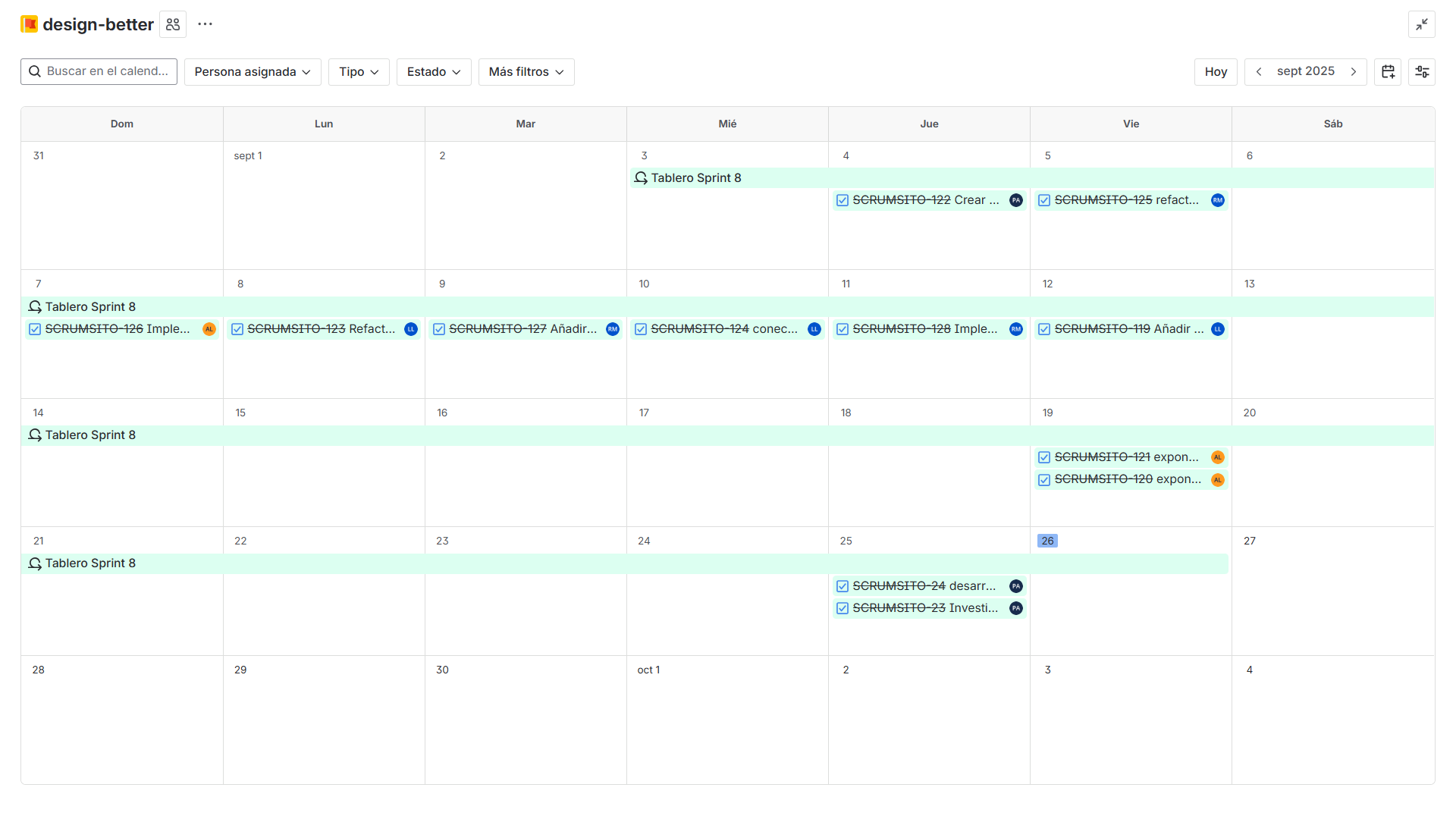
[SCRUMSITO-110] Refactorizar pantallas de autenticación   
Descripción: Reestructuración de las pantallas de login, registro y recuperación de credenciales.   
Horas estimadas: 1 h 30 m   
Puntos de historia: 1   
Responsable: ANDRE EMILIO PIVARAL LOPEZ   
Fecha fin: 24/08/2025

[SCRUMSITO-109] Cambiar almacenamiento de credenciales en local storage a cookies   
Descripción: Migración del sistema de credenciales hacia un esquema más seguro usando cookies.   
Horas estimadas: 1 h 20 m   
Puntos de historia: 1   
Responsable: ANDRE EMILIO PIVARAL LOPEZ   
Fecha fin: 29/08/2025

[SCRUMSITO-108] Crear mockups básicos de las pantallas principales   
Descripción: Diseño de prototipos iniciales para definir la estructura general de las pantallas clave.   
Horas estimadas: 1 h 10 m   
Puntos de historia: 1   
Responsable: ROBERTO SAMUEL NAJERA MARROQUIN   
Fecha fin: 23/08/2025

[SCRUMSITO-107] Crear componentes de creación de plantillas para diseñador   
Descripción: Construcción de componentes que permitan a los diseñadores elaborar y guardar plantillas.   
Horas estimadas: 1 h 55 m   
Puntos de historia: 1   
Responsable: PABLO JOSE MENDEZ ALVARADO   
Fecha fin: 26/08/2025

# Calendario



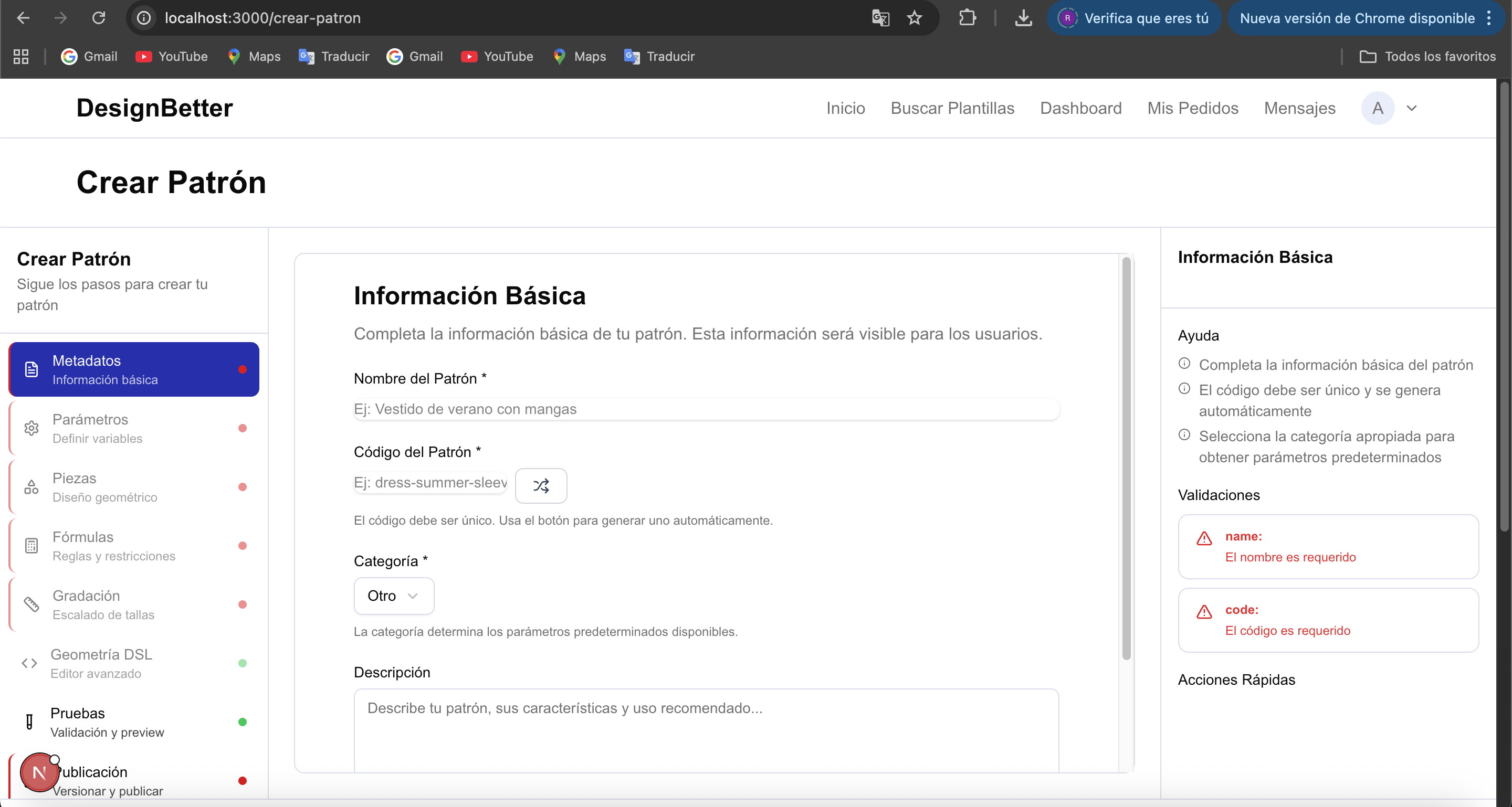
Evidencia del uso de las técnicas de scrum.

Carpeta de grabaciones de reuniones diarias: [Evidencias Scrum](https://uvggt-my.sharepoint.com/:f:/g/personal/naj23781_uvg_edu_gt/ElEUnQVfnAdPqVaViehEFnYB21Q28foWiOdg0mQry8OG7w)

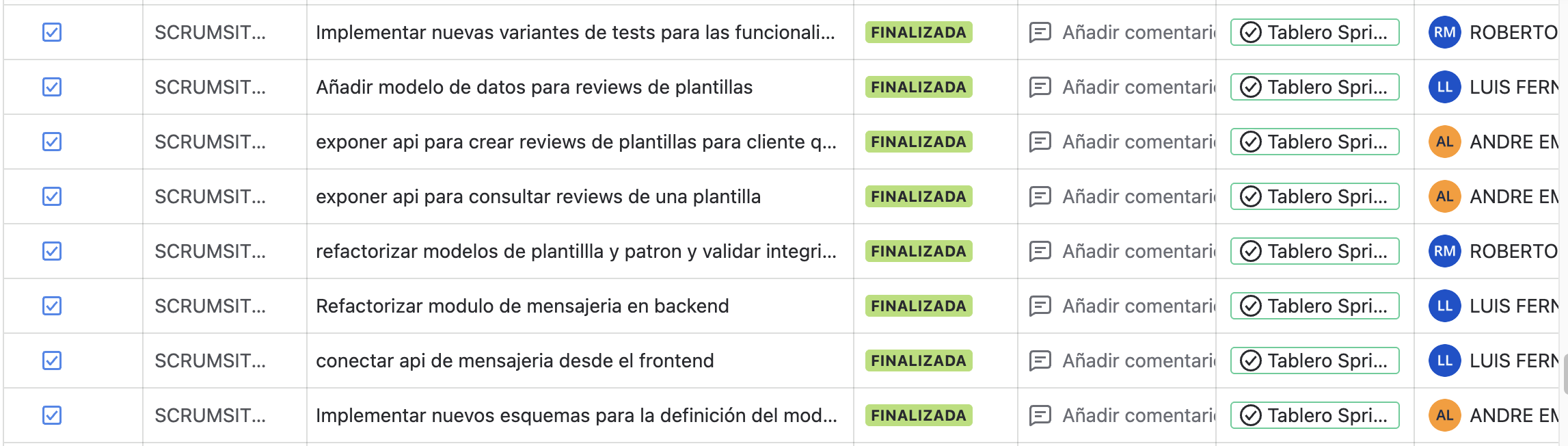
Enlace de Jira: <https://design-better.atlassian.net/jira/software/projects/SCRUMSITO/boards/1?atlOrigin=eyJpIjoiNjY0OGUzYjk5NzU0NGQ5YmFhMzk5NTZiYjZhMDVkMTAiLCJwIjoiaiJ9>

# Resultados del Sprint

## Software Funcionando



## Tareas Concluidas



No hubo tareas planificadas y no concluidas para este Sprint

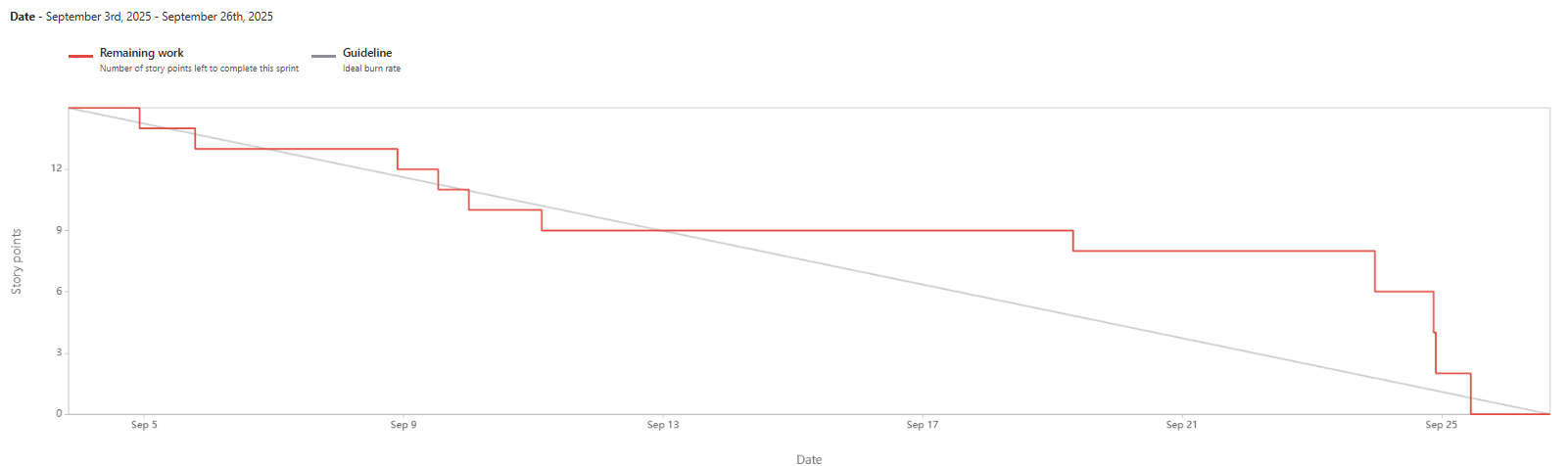
## Repositorios con el código desarrollado:

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

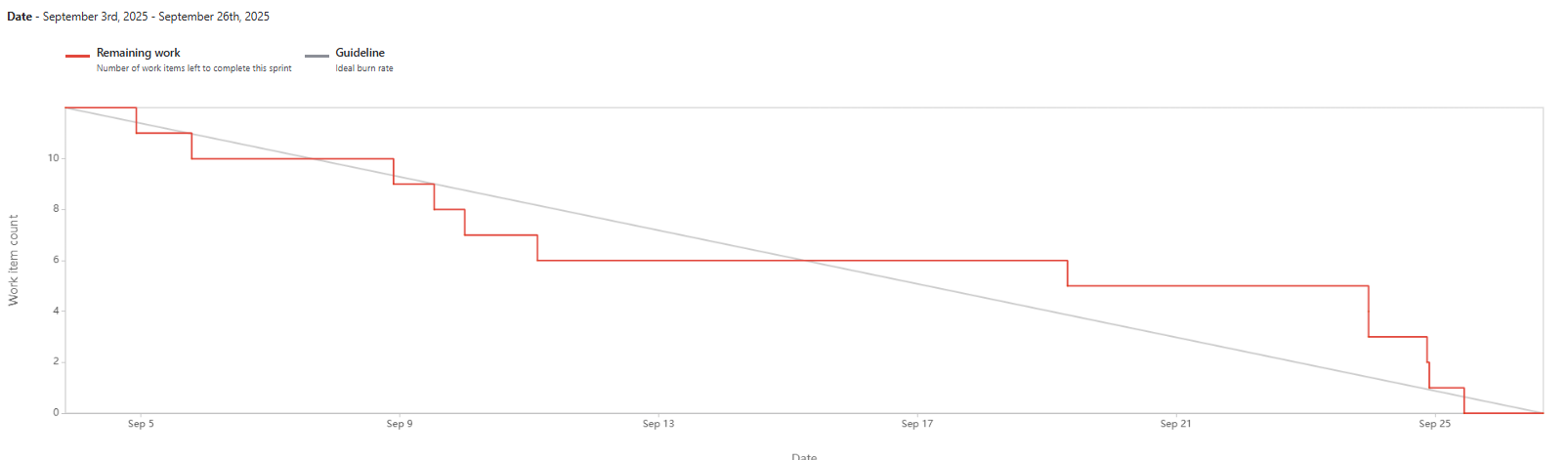
<https://github.com/Ultimate-Truth-Seeker/DesignBetterBackend>

# Métricas del Sprint

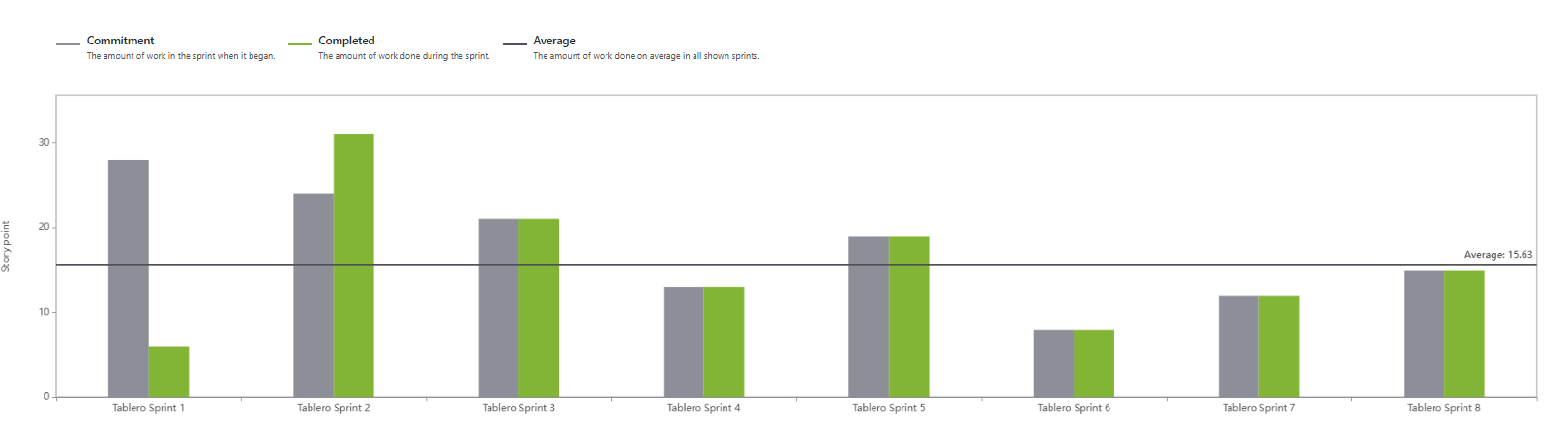
## Gráfico de Sprint Burndown de puntos de historia



## 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**

Indicador de Éxito = (Tareas completadas × 0.6) + (Promedio de puntualidad × 0.4)  
 = (100 × 0.6) + (85.0 × 0.4) = 94.0%

• Tareas comprometidas al inicio del sprint: 12  
 • Tareas completadas: 12  
 • Puntaje promedio por cumplimiento de fechas: 70/100

El Sprint VIII se completó con el 100 % de las tareas comprometidas entregadas (12/12), alcanzando un Indicador de Éxito del 94.0 %. Este resultado refleja que, aunque la totalidad del trabajo fue finalizada, la puntualidad promedio (70.0 %) redujo la calificación global. La semana de descanso también hubo actividad, y al completar todo al tiempo luego de esa semana indica que pudo haber subido la moral del equipo; pero antes de esta hubo un pequeño estancamiento.

## 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. Tambien se concluye que hubo un ritm constante, lo cual es tambien una buena medida del desempeño y la entrega continua necesaria para un buen proyecto.

# **Pruebas de UX y resultados**

Para las pruebas de UX se añadieron los elementos TC019 hasta TC023 al plan maestro. Las especficaciones son las siguientes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC018 | Buscar en Chatbot FAQs usando palabra clave | 1 usuario | query='envío' | HTTP 200 OK; solo FAQs que contengan 'envío' |
| TC019 | Test de contraste de colores en la interfaz | 2 usuario | Pantalla de personalización (colores, telas) | Contraste mínimo de 4.5:1 (WCAG AA) para texto; mensaje de aprobación si cumple |
| TC020 | Verificación de accesibilidad para daltonismo (deuteranopia) | 1 usuario | Pantalla de detalle de plantilla, con selección de colores y previsualización 3D | Colores distinguibles bajo simulación de deuteranopia; mensaje de aprobación si distinguibles |
| TC021 | Verificación de accesibilidad para daltonismo (protanopia) | 1 usuario | Pantalla de detalle de plantilla, con selección de colores y previsualización 3D | Colores distinguibles bajo simulación de protanopia; mensaje de aprobación si distinguibles |
| TC022 | Verificación de accesibilidad para daltonismo (tritanopia) | 1 usuario | Pantalla de detalle de plantilla, con selección de colores y previsualización 3D | Colores distinguibles bajo simulación de tritanopia; mensaje de aprobación si distinguibles |
| TC023 | Prueba de navegación con lector de pantalla | 1 usuario | Toda la plataforma (registro, personalización, tracking) | Navegación completa sin errores; lector de pantalla (e.g., NVDA) lee correctamente todos los elementos |

# **Calendario y responsable Asignado**

|  |  |  |  |
| --- | --- | --- | --- |
| Responsable | Pruebas asignadas | Número de usuarios | Fecha de realización |
| André Pivaral | TC020, TC021, TC022 | 1 | 12/9/2025 |
| Pablo Méndez | TC018 | 1 | 15/9/2025 |
| Luis Palacios | TC019 | 2 | 7/9/2025 y 12/09/2025 |
| Roberto Nájera | TC023 | 1 | 11/9/2025 |

# **Documentación de métodos para realizar las pruebas**

Para las pruebas de Daltonismo, se convertirá las pantallas de nuestro Frontend utilizando herramientas en línea para generar una apariencia como las vería una persona con estas condiciones. Estas se les mostraran a personas de visión normal para que puedan confirmar que el contraste es adecuado. La prueba con lector de pantalla se estará realizando con la utilidad de VoiceOver para MacOS. El resto de las pruebas se realizarán en persona o virtualmente dependiendo de la disponibilidad y se les mostrarán los avances, y el resultado que se obtenga de estas pruebas nos indicara si se cumplen o no.

A continuación, se presenta la herramienta a utilizar para este propósito:

## **Pilestone Color Vision Simulator**

Consideramos adecuado utilizar como única herramienta para nuestras pruebas de interfaz de usuario a Pilestone, ya que permite subir imágenes; como capturas de pantalla del sistema, mockups o imágenes de componentes específicos, y ver su apariencia bajo los tres grandes grupos de deficiencia cromática con dos niveles por condición: Deuteranopia, Protanopia y Tritanopia. Esto nos posibilita evaluar de forma sistemática tanto casos moderados (anomalías tricromáticas) como severos (dicromatías), alineado con la clasificación clínica vigente y con los modelos usados en investigación.

Además, el simulador no almacena las imágenes y funciona en el navegador, lo que protege la confidencialidad de artefactos de diseño y de capturas de entornos no públicos; los resultados se pueden descargar y adjuntar como evidencia de las pruebas de UX.

Integrar Pilestone en el plan de pruebas nos permite verificar legibilidad, mapas de color en gráficas y accesibilidad de iconografía directamente sobre los componentes que usamos en el ciclo de desarrollo, con una trazabilidad clara y sin requerir herramientas adicionales.

# **Pruebas Realizadas**

En el siguiente enlace se encontrarán las evidencias y resultados de las pruebas realizadas de UX conforme se vayan añadiendo. Puede consultar las fotos, videos y rúbricas de evidencia en esta carpeta compartida:  
[Resultados y evidencias pruebas UX](https://uvggt-my.sharepoint.com/:f:/g/personal/naj23781_uvg_edu_gt/Ejegjkv-vTFMoSKE26SNioABnG-B84N0eu8v_Jgju1UY4w)

# **Resultados, Hallazgos y Conclusiones**

A continuación, se detallan los resultados y la discusión respectiva para cada una de las pruebas de UX realizadas.

Para TC018, El chatbot destaca en precisión y usabilidad práctica, logrando un 90% que lo hace viable para producción. La alta cobertura funcional (60 puntos de 60) compensa la debilidad aislada en UX, y el contexto de moda asegura que no parezca desconectado del negocio. En términos del stack técnico (Next.js/Django/Docker), no se reportaron issues, lo que implica una integración estable. El único fallo en el tono de marca (10% de peso) es el "talón de Aquiles", ya que podría impactar la percepción emocional en un sector como la moda, donde la narrativa es clave. Si no se corrige, podría bajar la satisfacción del usuario en pruebas futuras o en uso real. Con 90%, el chatbot pasa la prueba, pero está en el límite inferior del umbral. Esto sugiere que es funcional para un lanzamiento inicial, pero no exento de pulido. Con ajustes menores en el tono, podría elevarse a un 100%, mejorando la experiencia diferenciadora del proyecto.

Para TC019, la evaluación con dos usuarios alcanzó un desempeño global de 60% (6/10 en ambos casos), mostrando que ambos identificaron una preferencia clara por una de las versiones, pero con limitaciones importantes en la facilidad de lectura y la claridad de los elementos, ya que en ambas dimensiones solo se obtuvo 1/2 puntos, lo que refleja que al menos una versión no logra una legibilidad adecuada. En comodidad visual se evidenciaron diferencias: un usuario no reportó cansancio mientras que el otro sí, lo que indica que el contraste actual no se adapta de forma uniforme a distintos perfiles visuales. Finalmente, aunque se dieron algunas sugerencias de mejora, estas fueron generales y no totalmente aplicables, por lo que se recomienda ajustar los contrastes de texto y fondo, así como incorporar apoyos no cromáticos (bordes, sombreados o iconografía diferenciada) que refuercen la legibilidad y garanticen una experiencia más consistente para todos los usuarios.

Para TC020, TC021 y TC022, la evaluación con simulación de deuteranopía, protanopía y tritanopía muestra un desempeño global del 81%: la pantalla inicial alcanzó 84%, el formulario de nuevo pedido 80% y el dashboard principal 78%; por condición, el desempeño agregado fue 82% en deuteranopía, 82% en protanopía y 78% en tritanopía, indicando que el sistema es mayormente usable bajo las tres simulaciones pero requiere ajustes de contraste y señales no cromáticas; tales como iconos, etiquetas, bordes, patrones y focos visibles; especialmente en el dashboard para cerrar la brecha observada y consolidar la aprobación en todas las pantallas.

Para TC023, los resultados obtenidos según la rúbrica adjunta nos permiten inferir información importante. En general las pruebas de navegación con lector de voz fluyeron bastante bien, pudiéndose navegar con éxito a través de las diferentes ventanas de la aplicación y entendiéndose el flujo de la interfaz. El único detalle que hizo falta para cumplir con la prueba al 100% fue que se anunciaran alertas que surgieran. Cómo la aplicación no genera muchas alertas no es un problema grave, pero se puede mejorar este aspecto. Esos fueron los hallazgos principales de esta prueba, y nos permiten identificar esa carencia que se buscará resolver tratando de que las alertas e información emergente se indique cómo es debido. De esta manera lograremos incluir los hallazgos encontrados de manera significativa. Potencialmente, se estará resolviendo en el penúltimo o último sprint cuando se esté limpiando y revisando el código por última vez antes de pasarse a producción.

En conclusión final general, estas pruebas nos permitieron evaluar la efectividad y diseño de nuestra interfaz gráfica para la experiencia de usuario, especialmente en el caso de usuario que tienen discapacidades visuales, ya que estas pruebas nos permitieron identificar las fortalezas y debilidades presentes en nuestra aplicación. Dados estos hallazgos expuestos en los párrafos anteriores, se espera poder incluirlos como guía para nuestro desarrollo y se espera también poder solventar la mayoría de estos en los últimos dos esprints, por medio de revisiones técnicas y ajustes específicos, al momento de auditar de manera general y previa a producción el desarrollo obtenido de la aplicación.

# Pruebas de Carga, Estres y Seguridad

## Pruebas de Carga y Estres

Las pruebas de estrés aún no han sido realizadas, sin embargo ya están planificadas para cuando el proyecto sea puesto en producción.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Caso | Herramienta | Descripción | Configuración |
| STR001 | Creación masiva de usuarios | Locust + k6 | Simular 1000–5000 registros concurrentes de usuarios. | **Locust:** 500 usuarios simultáneos, spawn rate 50, duración 10m. **k6:** 0→1000 VUs en 5m, sostenido 10m. |
| STR002 | Picos de login | Locust + k6 | Ataque de ráfagas de login (burst) para probar autenticación. | **Locust:** 200 usuarios, spawn rate 200, ciclos de 2m. **k6:** stages con 1m a 200 RPS → 1m pausa → repetir 3 veces. |
| STR003 | Creación de pedidos concurrentes | Locust + k6 | Simular múltiples usuarios creando órdenes al mismo tiempo. | **Locust:** 300 usuarios, spawn rate 30, duración 15m. **k6:** 0→300 VUs en 3m, sostenido 15m. |
| STR004 | Búsquedas intensivas | Locust + k6 | Consultas GET concurrentes con filtros. | **Locust:** 400 usuarios, spawn rate 40. **k6:** 100 RPS durante 20m. |
| STR005 | Subida de archivos | Locust + k6 | Carga concurrente de archivos pequeños (1–5MB). | **Locust:** 100 usuarios, spawn rate 10. **k6:** 50 VUs constantes durante 10m. |
| STR006 | Soak test (resistencia) | Locust + k6 | Carga sostenida para medir fugas de memoria. | **Locust:** 50 usuarios, duración 3h. **k6:** 100 VUs constantes por 3h. |

## Pruebas de Seguridad

Estas fueron las pruebas de seguridad que se añadieron al plan maestro

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SEC ID** | **OWASP 2021** | **Caso / Objetivo** | **Parámetros / Entrada** | **Pasos** | **Resultado Esperado** |
| SEC001 | A01: Broken Access Control | Prevenir IDOR: un cliente intenta ver el detalle de un pedido que pertenece a otro usuario | GET /orders/pedidos /{pedido\_otro\_id} con JWT de cliente1 | 1) Autenticar cliente1; 2) Intentar acceder al pedido de cliente2 | HTTP 403/404; no se filtra información sensible; logs registran intento |
| SEC002 | A01: Broken Access Control | Evitar escalamiento horizontal/vertical: cliente intenta cambiar estado del pedido | PATCH /orders/pedidos/{id}/estado/ body={estado:'produccion'} con JWT de cliente | 1) Autenticar cliente; 2) Enviar PATCH al endpoint restringido a diseñadores | HTTP 403; no se crea registro en PedidoEstadoHistoria |
| SEC003 | A03: Injection | Probar inyección SQL en búsquedas/filtros de plantillas | GET /plantillas?search=' OR 1=1 -- / %27; DROP TABLE pedidos;-- (payloads) | 1) Enviar payloads maliciosos; 2) Revisar respuesta y estado de BD | HTTP 400/200 seguro; sin trazas de error SQL; integridad de BD intacta |
| SEC004 | A05: Security Misconfiguration | Validar CORS restringido al dominio del frontend | Preflight y fetch desde Origin atacante <http://evil.local> | 1) Enviar OPTIONS/GET con Origin malicioso; 2) Observar cabeceras | No Access-Control-Allow-Origin para orígenes no permitidos; bloqueo |
| SEC005 | A02: Cryptographic Failures | Asegurar transporte y cookies/headers: HTTPS, HSTS, flags de seguridad | Endpoints sensibles (auth, pedidos) | 1) Forzar HTTP→HTTPS; 2) Verificar HSTS; 3) Revisar Set-Cookie (Secure, HttpOnly, SameSite); 4) Sin secretos en respuestas | HTTPS forzado; HSTS; cookies seguras; sin exposición de secretos |
| SEC006 | A07: Identification and Authentication Failures | JWT inválido/expirado/manipulado y rate limiting de login | Authorization: Bearer <token expirado/tampered>; múltiples intentos a /auth/login/ | 1) Llamar API con JWT inválido/expirado; 2) 20 intentos fallidos de login | HTTP 401 genérico; 429 tras N intentos; sin razón específica expuesta |

Los resultados y ejecuciones de las pruebas se encuentran en el siguiente enlace

<https://uvggt-my.sharepoint.com/:f:/g/personal/pal23933_uvg_edu_gt/EmwOMavy6FZPrTLQdb4PjfIBarKBF2AS11iq8z8jZqVzRQ?e=9XTNru>

En general todo salió bien de las pruebas de seguridad, excepto la SEC005, que necesitar de completar tareas de seguridad en las conexiones en los siguientes sprints. Es importante controlar el rate limiting de peticiones al backend poniendole un ajuste claro de hastan cuando saca un tipo diferente de error http que bloquee el abuso. Todas estas son las tareas de mitigación para los fallo de seguridad encontrados.

# **Refactorización**

La deuda técnica de nuestro proyecto de software consiste en los siguientes puntos:

* Componentes no usados de frontend o deprecados con versiones nuevas
* Apis no conectadas o simuladas de manera temporal para probar funcionalidades
* Utilidades del backend para formatos de patronaje no implementadas en el resultado final
* Conexiones https y cookies de credenciales pospuestas
* Carpetas de elementos sin implementar por diferentes razones, por ejemplo mala comunicación de tareas por hacer y avances no planificados
* Tests obsoletos por modelos que se refactorizaron

Durante los últimos sprints que quedan hasta antes de subir el proyecto a producción, estaremos refactorizando y limpiando los componentes que mencionamos que forma parte de la deuda técnica, así como las funcionalidades y características adicionales mencionadas. El método principal para el procesos de refactorización será el Extract Method/Composing Method, para funciones que se ejecutan de manera indpendiente. Este proceso ya se lleva usando varias veces en algunas partes del frontend y se continuará aprovechando para solventar la deuda técnica acumulada.

# **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.

Se establece la tarifa unitaria por hora de trabajo en 15 USD; considerando un tipo de cambio de 7.8 quetzales por dólar estadounidense, equivale a Q. 117.00 por hora. Este valor sirve como referencia estándar para el cálculo de los costos mensuales y acumulados del equipo.

La carga laboral mensual prevista es de 3 horas diarias por desarrollador, 5 días a la semana durante 4 semanas, lo que representa 60 horas de trabajo por mes por persona. El equipo de desarrollo está conformado por cuatro integrantes, generando un total de 240 horas mensuales.

El costo mensual se determina multiplicando las 240 horas por la tarifa de Q. 117.00 por hora, obteniendo Q. 28,080.00. Considerando un período de ejecución de 8 meses a la fecha, el gasto acumulado asciende a Q. 224,640.00.

Adicionalmente, se incorpora un indicador porcentual de ejecución presupuestaria, calculado al dividir el gasto acumulado entre el presupuesto total asignado al proyecto y multiplicar el resultado por 100. De manera complementaria, se analiza el tiempo transcurrido en función de la duración total planificada de 10 meses; con 8 meses ejecutados, se ha consumido el 80% del tiempo disponible.

# Evidencia de Pruebas con Usuarios finales



Usuario de prueba: diseñadora Gaby Nájera

Fotografía usada con su permiso

Fecha: 26 de septiembre a las 22:50

Resumen de comentarios:

* Le gustó la idea de la funcionalidad de construcción de patrones, pero consider que esta no es prioritaria ya que ya existen muchos programas que pueden generar patrones de esta manera, pero sigue siendo útil dentro de la aplicación.
* Reitera que la necesidad principal y diferenciador de nuestro producto consiste en la gradación de los patrones de vestuario de acuerdo con las medidas del cliente, cosa que es prioridad y no debe de depender necesariamente de las tallas predefinidas sino de las fórmulas que se requieran para cada situación
* Mencionó que las reglas de precio a implementarse no deben de ser demasiado complejas para no confundir a los clientes o desanimarlos, pero puede ser útil calcular el costo real de manera independiente.

# Reflexión del Desempeño del Equipo en el Sprint VIII

Durante el Sprint VIII, en general, este sprint demuestra madurez en el equipo: el descanso no impidió el cumplimiento, y la racha de completitud a tiempo sugiere procesos optimizados. Sin embargo, los plateaus destacan cómo interrupciones externas pueden alterar el flujo, aunque en este caso no fue crítico. A pesar del descanso, el resultado final es positivo, se alcanzó el objetivo principal (12/12 completados), aunque no se llegó al tope de la línea roja (15), lo que podría indicar que algunos puntos extras quedaron pendientes. La puntuación de 94/100 es sólida, destacando excelencia en actividades, pero con margen en puntualidad.

Aspectos Clave Para Mejorar para el Próximo Sprint:

1. Gestión de plateaus durante interrupciones

La semana de descanso causó estancamientos visibles; planea "tareas de bajo impacto" o preparación remota para minimizar el impacto en el flujo. Ajusta la planificación inicial para contabilizar descansos, redistribuyendo puntos en periodos activos.

2. Alineación con la línea ideal

La desviación inicial y escalonada sugiere avances en bursts en lugar de constantes. Apunta a un ritmo más distribuido mediante desglose de tareas en sub-tareas diarias, reduciendo el riesgo de cuellos de botella.

3. Refuerzo en puntualidad

Si bien todas las tareas fueron concluidas, el cumplimiento de fechas alcanzó únicamente un 70.0%. Esto evidencia que las entregas no se alinearon con los plazos internos planificados, lo que debilita la percepción de confiabilidad y puede impactar en futuros sprints con alta interdependencia. Para mejorar este indicador, se propone establecer un margen de seguridad en la planificación (buffers de tiempo realistas) y reforzar la estimación colaborativa de esfuerzo, de modo que los plazos definidos reflejen mejor la capacidad real del equipo.

Aspectos a Observar para el Futuro

Impacto de descansos en la motivación y productividad: Observar si pausas recurrentes afectan la moral o generan "deuda técnica" acumulada. Realizar retrospectives post-sprint para medir esto cualitativamente.

Sostenibilidad de la racha: Con varios sprints completados a tiempo, vigilar signos de burnout; la mejora en tiempos está bastante bien, pero asegurarse de que no sea a costa de calidad (revisando métricas como defectos post-entrega).

Tendencias en gráficas futuras: Comparar con sprints previos; si los plateaus persisten sin descansos, podría indicar ineficiencias internas.

En conclusión, el Sprint 8 refleja un desempeño sólido con la finalización exitosa de los 12 puntos de historia, alcanzando una puntuación de 88/100, impulsada por un 100% en actividades y un 70% en puntualidad. La semana de descanso explica los plateaus en la gráfica, pero el equipo demostró resiliencia al recuperar el ritmo y mantener su racha de completitud a tiempo. Para el futuro, mejorar la puntualidad y distribuir mejor el esfuerzo durante interrupciones puede elevar los resultados, mientras se vigila la sostenibilidad y la precisión en estimaciones ante posibles incrementos de complejidad.

# Informe general de gestión para el Sprint VIII:

Fecha de Inicio: 3 de septiembre de 2025

Fecha de finalización: viernes 26 de septiembre 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 VIII.docx](https://uvggt-my.sharepoint.com/:w:/g/personal/piv23574_uvg_edu_gt/EezIKyDQ97RHlCHZnj0HA68BPRHOIsWc-t3UD42kOv1cWQ?e=OQdcHe)