

UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de software II

Sección 20

Ing. Pablo Koch



Sprint 6: Fluentis

Diego Patzán - 23525

Diego Lopez - 23242

June Herrera - 231038

Ihan Marroquin – 23108

GUATEMALA, 8 de agosto de 2025

Product Backlog

Backlog Completo del Proyecto BPM de ASOFARMA

Tipo	Resumen	Descripción
Epic	Épica 1: Autenticación y Gestión de Usuarios	Gestión de autenticación y usuarios
Story	HU1: Inicio de sesión con Azure AD	Inicio de sesión con Azure AD
Task	Tarea 1: Configurar Azure AD para SSO en el backend (ASP.NET Core)	Registrar la aplicación en Azure AD, obtener credenciales (Client ID, Tenant ID) y configurar la autenticación.
Task	Tarea 2: Implementar middleware de autenticación en ASP.NET Core	Configurar el middleware para validar tokens JWT de Azure AD y extraer `oid`, `name`, y `email`.
Task	Tarea 3: Implementar inicio de sesión en el frontend (React)	Configurar la redirección a Azure AD para autenticación y manejar el token recibido.
Task	Tarea 4: Desarrollar página de login	Crear una página en React que redirija a Azure AD para iniciar sesión.
Story	HU2: Registro automático de usuarios	Registro automático de usuarios basado en Azure AD

Task	Tarea 5: Modificar la tabla Usuario para incluir oid	Agregar un campo oid (varchar(255)) a la tabla Usuario en SQL Server y crear un índice único.
Task	Tarea 6: Crear servicio para buscar o crear usuarios basado en oid	Desarrollar un servicio en el backend que busque el usuario por oid y lo cree si no existe, usando datos del token.
Story	HU3: Actualización de información básica	Actualización de información básica del usuario.
Task	Tarea 7: Implementar lógica para actualizar información del usuario	Asegurar que los datos del usuario (nombre, email) se actualicen en la base de datos si cambian en Azure AD.
Story	HU4: Asignación de roles internos	Asignación de roles internos a los usuarios.
Task	Tarea 8: Crear vista para gestionar roles internos	Desarrollar una interfaz en React para que los administradores asignen roles a los usuarios.
Task	Tarea 9: Implementar API para asignar roles	Crear endpoints en el backend para actualizar el rol_id de los usuarios.
Story	HU5: Gestión de departamentos	Gestión de departamentos para usuarios.
Task	Tarea 10: Crear vista para gestionar departamentos	Desarrollar una interfaz en React para asignar usuarios a departamentos.
Task	Tarea 11: Implementar API para asignar departamentos	Crear endpoints en el backend para actualizar el departamento_id de los usuarios.
Epic	Épica 2: Gestión de Flujos de Aprobación	Gestión de flujos de aprobación

Story	HU6: Creación de plantillas de flujos	Creación de plantillas de flujos de aprobación
Task	Tarea 12: Crear modelo y migración para FlujoAprobacion	Definir el modelo en el backend y generar la migración para la tabla FlujoAprobacion.
Task	Tarea 13: Implementar API para crear plantillas de flujos	Desarrollar endpoints para crear y listar plantillas de flujos.
Task	Tarea 14: Desarrollar vista para el editor de flujos en React	Crear una interfaz gráfica de drag & drop para diseñar flujos con pasos secuenciales y paralelos.
Story	HU7: Definición de tipos de pasos	Definición de tipos de pasos en el flujo
Task	Tarea 15: Implementar lógica para tipos de pasos (estándar, bifurcación, ancla)	Añadir funcionalidad en el backend para manejar los diferentes tipos de pasos (normal, bifurcacion, union).
Story	HU8: Asignación de responsables por paso	Asignación de responsables para cada paso
Task	Tarea 16: Crear modelo y migración para PasoFlujo	Definir el modelo y migración para la tabla PasoFlujo.
Task	Tarea 17: Implementar asignación de responsables por paso	Desarrollar la lógica en el backend para asignar responsables por nombre, rol, relación o grupo.
Story	HU9: Configuración de reglas de aprobación	Configuración de reglas de aprobación para flujos

Task	Tarea 18: Desarrollar vista para configurar reglas de aprobación	Añadir opciones en la interfaz de React para definir reglas de aprobación (unánime, individual, ancla).
Task	Tarea 19: Implementar API para guardar reglas de aprobación	Crear endpoints para almacenar las reglas de aprobación en la base de datos.
Story	HU10: Creación de formularios dinámicos	Creación de formularios dinámicos para solicitudes
Task	Tarea 20: Crear modelo y migración para Inputs y RelacionInput	Definir modelos y migraciones para manejar campos dinámicos en formularios.
Task	Tarea 21: Desarrollar vista para definir campos dinámicos	Implementar en React la capacidad de agregar campos personalizados a la solicitud y pasos.
Story	HU11: Gestión de grupos de aprobación	Gestión de grupos de aprobación en flujos
Task	Tarea 22: Implementar API para gestionar grupos de aprobación	Crear endpoints para crear, listar y asignar grupos de aprobación.
Task	Tarea 23: Desarrollar vista para gestionar grupos de aprobación	Crear una interfaz en React para que los administradores gestionen grupos.
Epic	Épica 3: Gestión de Solicitudes	Gestión de solicitudes
Story	HU12: Inicio de una solicitud	Inicio de una solicitud basada en plantillas

Task	Tarea 24: Crear modelo y migración para Solicitud	Definir el modelo y migración para la tabla Solicitud.
Task	Tarea 25: Implementar API para iniciar una solicitud	Desarrollar endpoints para que los usuarios inicien solicitudes basadas en plantillas.
Task	Tarea 26: Desarrollar vista para iniciar una solicitud	Crear un formulario en React para seleccionar una plantilla y completar los campos iniciales.
Task	Tarea 27: Implementar lógica para generar FlujoActivo al enviar la solicitud	Añadir funcionalidad en el backend para crear un flujo activo al enviar una solicitud.
Story	HU13: Visualización del estado de la solicitud	Visualización del estado y progreso de una solicitud
Task	Tarea 28: Crear vista para visualizar el estado de la solicitud	Desarrollar una interfaz en React que muestre el estado actual y los pasos completados de una solicitud.
Story	HU14: Ejecución de tareas en pasos	Ejecución de tareas en cada paso de una solicitud
Task	Tarea 29: Implementar API para ejecutar tareas en pasos	Crear endpoints para que los ejecutores completen tareas y actualicen el estado del paso.
Task	Tarea 30: Desarrollar vista para ejecutar tareas	Implementar en React una lista de tareas pendientes y formularios para completarlas.
Story	HU15: Aprobación o rechazo de pasos	Aprobación o rechazo de tareas en cada paso

Task	Tarea 31: Implementar API para aprobar o rechazar pasos	Desarrollar endpoints para que los aprobadores tomen decisiones y registren comentarios.
Task	Tarea 32: Desarrollar vista para aprobar o rechazar pasos	Crear una interfaz en React con botones de acción rápida (aprobar, rechazar, regresar).
Story	HU16: Delegación de responsabilidades	Delegación de tareas a otros usuarios
Task	Tarea 33: Implementar lógica de delegación	Añadir funcionalidad en el backend para delegar responsabilidades a otros usuarios.
Task	Tarea 34: Desarrollar vista para delegar tareas	Implementar en React la opción de delegar con selección de usuario y fechas.
Story	HU17: Gestión de excepciones	Gestión y registro de excepciones en flujos
Task	Tarea 35: Implementar gestión de excepciones	Crear endpoints y lógica para registrar y manejar excepciones en los pasos.
Epic	Épica 4: Monitoreo y Reportes	Monitoreo y generación de reportes
Story	HU18: Visualización de flujos activos	Visualización de flujos activos
Task	Tarea 36: Desarrollar vista para visualizar flujos activos	Crear una interfaz en React que liste los flujos activos y su estado.
Story	HU19: Generación de informes	Generación de informes personalizados

Task	Tarea 37: Implementar API para generar informes	Crear endpoints para generar y descargar informes personalizados.
Task	Tarea 38: Desarrollar vista para generar informes	Implementar en React una interfaz para seleccionar parámetros y generar informes.
Story	HU20: Alertas y notificaciones	Sistema de alertas y notificaciones
Task	Tarea 39: Configurar sistema de notificaciones	Implementar un servicio en el backend para enviar notificaciones a los usuarios.
Task	Tarea 40: Desarrollar vista para notificaciones	Crear una sección en React donde los usuarios vean sus notificaciones.
Story	HU21: Dashboard de métricas	Dashboard con KPIs y métricas en tiempo real
Task	Tarea 41: Crear modelo y migración para Metrica	Definir el modelo y migración para almacenar KPIs.
Task	Tarea 42: Implementar cálculo de métricas en tiempo real	Desarrollar la lógica en el backend para calcular métricas como tiempos de aprobación.
Task	Tarea 43: Desarrollar dashboard de métricas en React	Crear una vista con gráficos y tablas para mostrar KPIs.
Epic	Épica 5: Mejora Continua	Propuestas y mejoras continuas
Story	HU22: Propuesta de mejoras	Propuestas de mejora en el proyecto
Task	Tarea 44: Crear modelo y migración para Propuesta	Definir el modelo y migración para almacenar propuestas de mejora.

Task	Tarea 45: Implementar API para crear y listar propuestas	Desarrollar endpoints para que los usuarios propongan mejoras.
Task	Tarea 46: Desarrollar vista para proponer mejoras	Crear un formulario en React para que los usuarios envíen propuestas.
Story	HU23: Votación sobre propuestas	Sistema de votación sobre propuestas de mejora
Task	Tarea 47: Implementar sistema de votación	Añadir funcionalidad en el backend para que los usuarios voten sobre propuestas.
Task	Tarea 48: Desarrollar vista para votar propuestas	Implementar en React una lista de propuestas con opciones de voto.
Story	HU24: Implementación de mejoras	Implementación de mejoras aprobadas
Task	Tarea 49: Implementar lógica para implementar mejoras aprobadas	Desarrollar la capacidad de actualizar plantillas de flujos basadas en propuestas aprobadas.
Epic	Épica 6: Integraciones y Mantenimiento	Integraciones y mantenimiento del sistema
Story	HU25: Integración con Azure AD	Integración con Azure AD para SSO en contenedores
Task	Tarea 50: Configurar integración con Azure AD en Docker	Asegurar que la configuración de Azure AD funcione dentro del contenedor.
Story	HU26: Gestión de backups	Automatización y gestión de backups

Task	Tarea 51: Crear script para backups automáticos	Desarrollar un script en el backend para programar backups de la base de datos.
Story	HU27: Reporte de incidentes	Reporte de incidentes y problemas técnicos
Task	Tarea 52: Implementar API para reportar incidentes	Crear endpoints para que los usuarios reporten problemas técnicos.
Task	Tarea 53: Desarrollar vista para reportar incidentes	Implementar un formulario en React para reportar incidentes.
Story	HU28: Auditoría y cumplimiento	Auditoría y cumplimiento de normativas
Task	Tarea 54: Implementar registro de auditoría	Añadir logging en el backend para registrar acciones clave (creación, aprobación, etc.).
Task	Tarea 55: Desarrollar vista para exportar datos de auditoría	Crear una interfaz en React para que los auditores exporten registros.
Epic	DevOps: Configuración y Despliegue	Tareas de configuración inicial y despliegue
Task	Tarea 56: Crear repositorios separados para frontend y backend	Configurar dos repositorios en GitHub: uno para frontend y otro para backend.
Task	Tarea 57: Configurar ramas y flujo de trabajo Git	Establecer ramas (main, develop), políticas de branching y protección de ramas.

Task	Tarea 58: Configurar linters y formateadores para el código	Instalar ESLint y Prettier para frontend, y StyleCop para backend.
Task	Tarea 59: Implementar hooks de pre-commit	Usar Husky para ejecutar linters y pruebas antes de cada commit.
Task	Tarea 60: Configurar entornos de desarrollo con Docker Compose	Crear docker-compose.yml para levantar frontend, backend y base de datos.
Task	Tarea 61: Configurar variables de entorno	Definir variables para desarrollo, staging y producción con archivos .env.
Task	Tarea 62: Configurar CI/CD para el backend	Implementar pipeline en Azure DevOps o GitHub Actions para compilar, probar y desplegar el backend.
Task	Tarea 63: Configurar CI/CD para el frontend	Crear pipeline para compilar y desplegar el frontend a un servidor estático.
Task	Tarea 64: Configurar despliegue en Azure App Service (backend)	Preparar el backend para desplegarse en Azure App Service con slots de staging y producción.
Task	Tarea 65: Configurar despliegue del frontend en Azure Static Web Apps	Desplegar el frontend en Azure Static Web Apps con integración continua.
Task	Tarea 66: Configurar la base de datos en Azure SQL Database	Migrar la base de datos a Azure SQL Database y configurar la conexión segura.
Task	Tarea 67: Implementar estrategia de backups automáticos	Configurar backups automáticos de la base de datos en Azure.

Task	Tarea 68: Configurar SSL y seguridad en despliegue	Asegurar que frontend y backend usen HTTPS con certificados SSL.
Task	Tarea 69: Documentar el proceso de despliegue	Crear un documento con instrucciones para despliegues, rollbacks y mantenimiento.
Task	Tarea 70: Configurar el entorno de desarrollo con Docker	Instalar y configurar Docker en las máquinas de desarrollo, incluyendo la imagen de SQL Server.
Task	Tarea 71: Crear base de datos y aplicar migraciones iniciales	Configurar la conexión a SQL Server y ejecutar las migraciones para crear las tablas.
Task	Tarea 72: Implementar autenticación en el frontend	Configurar la librería de autenticación en React (como <code>@azure/msal-react</code>) para manejar el flujo de SSO.
Task	Tarea 73: Configurar CORS y seguridad en el backend	Asegurar que el backend acepte solicitudes del frontend y maneje correctamente los tokens.
Task	Tarea 74: Implementar manejo de errores y logging	Añadir manejo de excepciones y logging en el backend para facilitar el debugging.

USER STORY MAPPING:

https://lucid.app/lucidspark/b138eab5-c160-4dfa-b45d-38d14c6a41ef/edit?viewport_loc=-2088%2C-1820%2C7680%2C3780%2C0_0&invitationId=inv_66ae4aac-d27f-4e77-905a-d89641737a07

Tareas Finalizadas por sprint

Sprint 1

Key	Summary	Work type	Epic	Status	Assignee	Story points
FLUEN-311 ↗	Tarea 57: Configurar ramas y flujo de trabajo Git	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	JW	2
FLUEN-313 ↗	Tarea 59: Implementar hooks de pre-commit	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	JW	2
FLUEN-333 ↗	Tarea 61: Setupear las diferentes tecnologías a los repositorio...	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	D	2
FLUEN-314 ↗	Tarea 60: Configurar entornos de desarrollo con Docker Com...	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	DL	3
FLUEN-312 ↗	Tarea 58: Configurar linters y formateadores para el código	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	D	3
FLUEN-315 ↗	Tarea 62: Configurar variables de entorno	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	IM	2
FLUEN-316 ↗	Tarea 63: Configurar CI/CD para el backend	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	DL	5
FLUEN-330 ↗	Tarea 77: Inicializar proyecto de React con Vite para el frontend.	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	D	5
FLUEN-329 ↗	Tarea 76: Inicializar proyecto de ASP .NET CORE para el backe...	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	JW	5

Sprint 2

Key	Summary	Work type	Epic	Status	Assignee	Story points
FLUEN-317 ↗	Tarea 64: Configurar CI/CD para el frontend	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	D	5
FLUEN-325 ↗	Tarea 72: Aplicar migraciones iniciales a la DB.	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	JW	5
FLUEN-331 ↗	Tarea 78: Creación de modelos y estructuras para la DB en el ...	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	JW	6
FLUEN-334 ↗	Tarea 80: Creación de mock up data para el backend e inserci...	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	IM	5
FLUEN-234 ↗	Tarea 6: Creación, lectura, actualización y eliminación de usu...	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	JW	3
FLUEN-335 ↗	Tarea 81: Creación operaciones crud de rol y departamento d...	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	JW	3
FLUEN-336 ↗	Tarea 82: Crear Azure Entra Id with user mocks	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	6
FLUEN-228 ↗	Tarea 1: Configurar Azure AD para inicio sesion en el backend...	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	5
FLUEN-229 ↗	Tarea 2: Implementar middleware de autenticación en ASP.NE...	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	3
FLUEN-233 ↗	Tarea 5: Modificar la tabla Usuario para incluir oid	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	IM	2
FLUEN-231 ↗	Tarea 4: Desarrollar página de landingpage	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	D	2
FLUEN-338 ↗	Tarea 84: Crear react router en el fronted con rutas protegidas	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	5

Sprint 3

Key	Summary	Work type	Epic	Status	Assignee	Story points
FLUEN-337 ↗	Tarea 83: Crear navbar y layout reactivo segun auth	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	D	3
FLUEN-339 ↗	Tarea 85: Crear interceptor de peticiones para agregar token	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	3
FLUEN-230 ↗	Tarea 3: Implementar inicio de sesión en el frontend (React)	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	D	6
FLUEN-326 ↗	Tarea 73: Implementar autenticación en el frontend	<input checked="" type="checkbox"/> Task	DEVOPS: CON...	DONE	DL	5
FLUEN-340 ↗	Tarea 80: Agregar jobtitle y superiores en la db para manejar j...	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	JW	2
FLUEN-239 ↗	Tarea 9: Implementar API para asignar roles	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	IM	2
FLUEN-242 ↗	Tarea 11: Implementar API para asignar departamentos	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	JW	2
FLUEN-341 ↗	Tarea 8: Definir si los roles se van a manejar de forma general ...	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	8
FLUEN-342 ↗	Tarea 90: Definir si los roles serán aplicados globalmente o po...	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	6
FLUEN-343 ↗	Tarea 91: Controlador para Rol definido.	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	JW	4
FLUEN-344 ↗	Tarea 92: Migrar usuario de Azure a la base de datos.	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	4
FLUEN-345 ↗	Tarea 93: Manejar relaciones jerárquicas en el backend.	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	JW	3
FLUEN-346 ↗	Tarea 94: Crear vista para gestión de Usuarios.	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	D	4

Sprint 4

Completed work items						View in All work navigator
Key	Summary	Work type	Epic	Status	Assignee	Story points
FLUEN-236 ↗	Tarea 7: Implementar vista de perfil de usuario.	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	IM	2
FLUEN-238 ↗	Tarea 8: Crear vista para gestionar roles internos	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	D	3
FLUEN-241 ↗	Tarea 10: Crear vista para gestionar departamentos	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUTE...	DONE	DL	3

Sprint 5

Key	Summary	Work type	Epic	Status	Assignee	Story points
FLUEN-347 ↗	Tarea 100: Implementar funcionalidad para asignación y...	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUT...	DONE	D	10
FLUEN-348 ↗	Tarea 101: Implementar sesiones persistentes en auth	<input checked="" type="checkbox"/> Task	ÉPICA 1: AUT...	DONE		5
FLUEN-250 ↗	Tarea 16: Crear modelo y migración para PasoFlujo	<input checked="" type="checkbox"/> Task	ÉPICA 2: GES...	DONE		2
FLUEN-244 ↗	Tarea 12: Crear modelo y migración para FlujoAprobacion	<input checked="" type="checkbox"/> Task	ÉPICA 2: GES...	DONE		2
FLUEN-246 ↗	Tarea 14: Desarrollar vista para el editor de flujos en React	<input checked="" type="checkbox"/> Task	ÉPICA 2: GES...	DONE		8
FLUEN-248 ↗	Tarea 15: Implementar lógica para tipos de pasos (están...	<input checked="" type="checkbox"/> Task	ÉPICA 2: GES...	DONE		5
FLUEN-251 ↗	Tarea 17: Implementar asignación de responsables por p...	<input checked="" type="checkbox"/> Task	ÉPICA 2: GES...	DONE		5
FLUEN-256 ↗	Tarea 20: Crear modelo y migración para Inputs y Relaci...	<input checked="" type="checkbox"/> Task	ÉPICA 2: GES...	DONE		3

Pila del Sprint

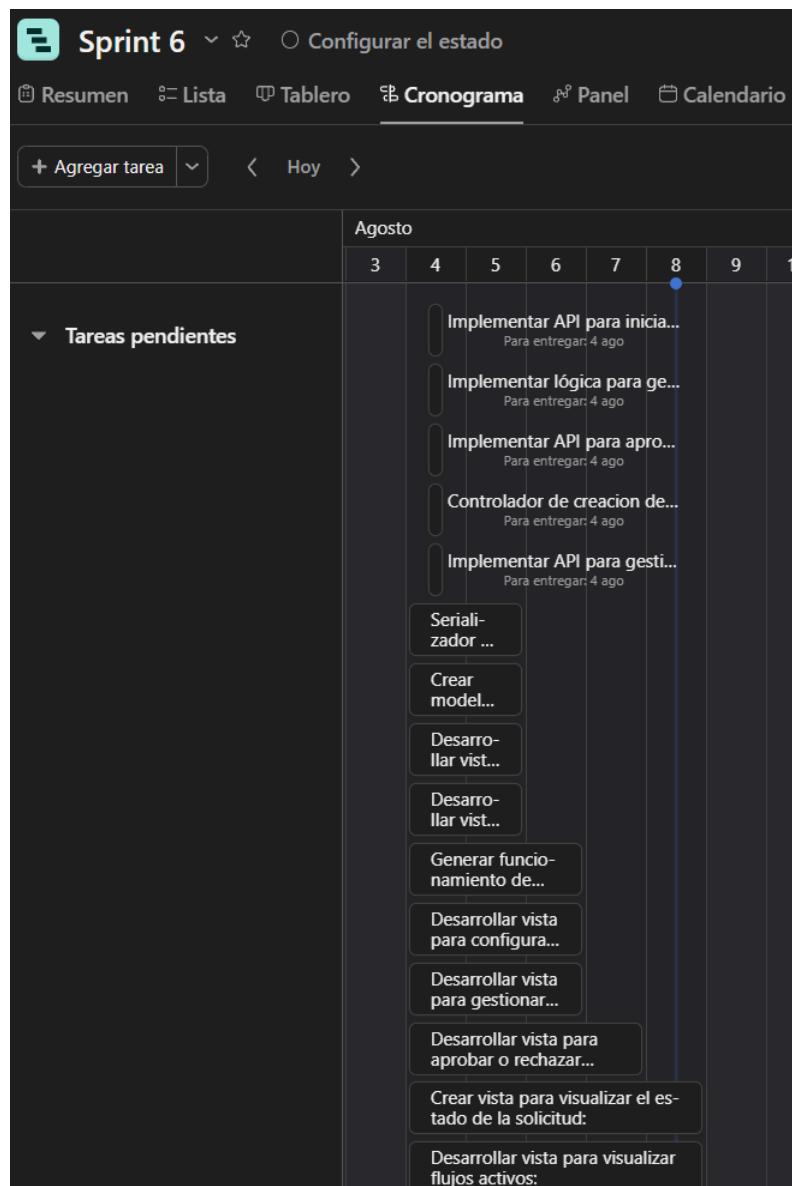
Historias de usuario en el Sprint:

- Implementar API para iniciar una solicitud:
 - Descripción: Desarrollar endpoints para que los usuarios inicien solicitudes basadas en plantillas.
 - Horas estimadas de desarrollo: 5 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Diego Lopez
 - Fecha probable de terminación de la tarea: 04 de agosto
- Implementar lógica para generar FlujoActivo al enviar la solicitud:
 - Descripción: Añadir funcionalidad en el backend para crear un flujo activo al enviar una solicitud.
 - Horas estimadas de desarrollo: 3 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Diego Lopez
 - Fecha probable de terminación de la tarea: 04 de agosto
- Implementar API para aprobar o rechazar pasos:
 - Descripción: Desarrollar endpoints para que los aprobadores tomen decisiones y registren comentarios.
 - Horas estimadas de desarrollo: 3 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Diego Lopez
 - Fecha probable de terminación de la tarea: 04 de agosto
- Controlador de creacion de inputs:
 - Descripción: Implementacion de creacion de inputs funcionales para el front
 - Horas estimadas de desarrollo: 2 horas
 - Puntos de historia asignados a la tarea: 2
 - Responsable de desarrollarla: Diego Lopez
 - Fecha probable de terminación de la tarea: 04 de agosto
- Implementar API para gestionar grupos de aprobación:
 - Descripción: Crear endpoints para crear, listar y asignar grupos de aprobación.
 - Horas estimadas de desarrollo: 5 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Diego Lopez
 - Fecha probable de terminación de la tarea: 04 de agosto
- Serializador y deserializador de campos json input
 - Descripción: Implementacion de json input
 - Horas estimadas de desarrollo: 8 horas
 - Puntos de historia asignados a la tarea: 5
 - Responsable de desarrollarla: June Herrera
 - Fecha probable de terminación de la tarea: 05 de agosto
- Crear modelo y migración para Inputs:

- Descripción: Definir el modelo y migración para la tabla Solicitud.
 - Horas estimadas de desarrollo: 3 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: June Herrera
 - Fecha probable de terminación de la tarea: 05 de agosto
- Desarrollar vista para iniciar una solicitud
 - Descripción: Crear un formulario en React para seleccionar una plantilla y completar los campos iniciales.
 - Horas estimadas de desarrollo: 7 horas
 - Puntos de historia asignados a la tarea: 5
 - Responsable de desarrollarla: Ihan Marroquin
 - Fecha probable de terminación de la tarea: 05 de agosto
- Desarrollar vista para definir campos dinámicos
 - Descripción: Implementar en React la capacidad de agregar campos personalizados a la solicitud y pasos.
 - Horas estimadas de desarrollo: 6 horas
 - Puntos de historia asignados a la tarea: 5
 - Responsable de desarrollarla: June Herrera
 - Fecha probable de terminación de la tarea: 05 de agosto
- Generar funcionamiento de vista conexión entre pasos
 - Descripción: Implementar la visibilidad entre pasos de flujo
 - Horas estimadas de desarrollo: 3 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Diego Lopez
 - Fecha probable de terminación de la tarea: 06 de agosto
- Desarrollar vista para configurar reglas de aprobación
 - Descripción: Añadir opciones en la interfaz de React para definir reglas de aprobación (unánime, individual, ancla).
 - Horas estimadas de desarrollo: 4 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Ihan Marroquin
 - Fecha probable de terminación de la tarea: 06 de agosto
- Desarrollar vista para gestionar grupos de aprobación
 - Descripción: Crear una interfaz en React para que los administradores gestionen grupos.
 - Horas estimadas de desarrollo: 3 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Diego Patzán
 - Fecha probable de terminación de la tarea: 06 de agosto
- Desarrollar vista para aprobar o rechazar pasos
 - Descripción: Crear una interfaz en React con botones de acción rápida (aprobar, rechazar, regresar).
 - Horas estimadas de desarrollo: 3 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Ihan Marroquin
 - Fecha probable de terminación de la tarea: 07 de agosto
- Crear vista para visualizar el estado de la solicitud:
 - Descripción: Desarrollar una interfaz en React que muestre el estado actual y los

- pasos completados de una solicitud.
- Horas estimadas de desarrollo: 7 horas
 - Puntos de historia asignados a la tarea: 5
 - Responsable de desarrollarla: Diego Patzan
 - Fecha probable de terminación de la tarea: 08 de agosto
- Desarrollar vista para visualizar flujos activos:
 - Descripción: Crear una interfaz en React que liste los flujos activos y su estado.
 - Horas estimadas de desarrollo: 2 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: Ihan Marroquin
 - Fecha probable de terminación de la tarea: 08 de agosto

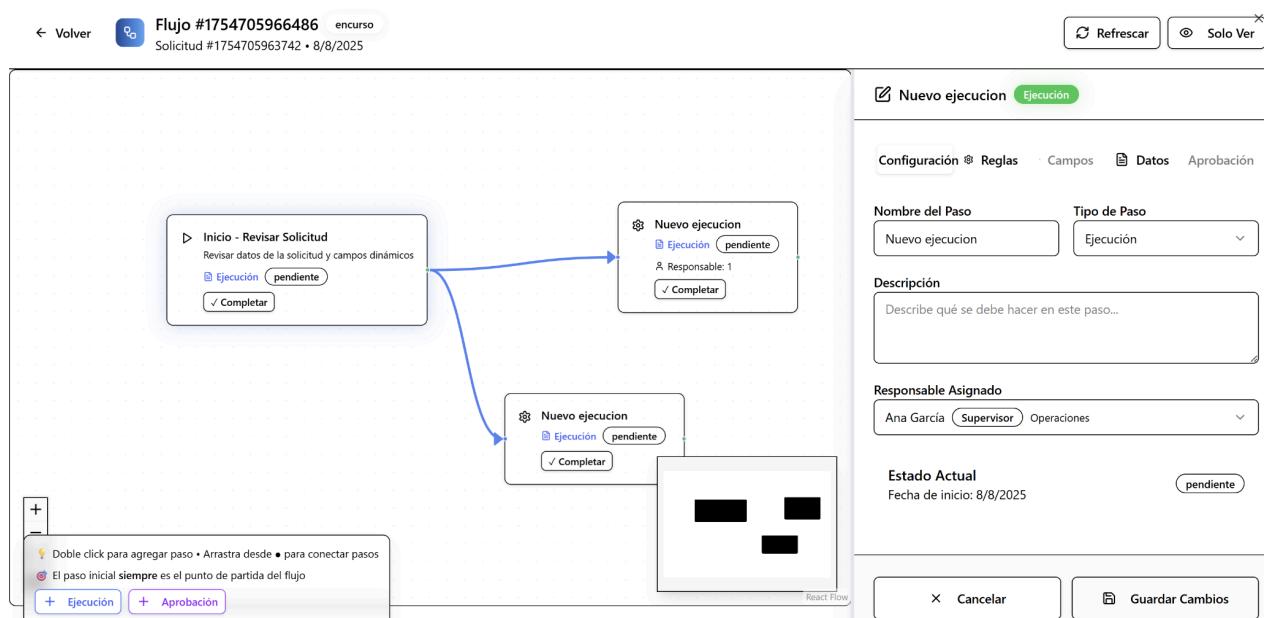
Calendario de planificación del Sprint:



Resultados del Sprint

Software funcionando

The screenshot shows the Asofarma BPM application interface. On the left, there is a sidebar with the user information "ihanmarroquin12 Miembro" and navigation links such as Dashboard, Equipos y Permisos, Flujos y Solicitudes (selected), Panel BPM, and Cerrar Sesión. The main content area is titled "Business Process Management" and "Gestión integral de solicitudes y flujos de trabajo automatizados". It features four summary cards: "Solicitudes registradas" (0), "Aprobadas" (0), "Pendientes" (0), and "Flujos activos" (0). Below these are two tabs: "Resumen" (selected) and "Solicitudes". The "Resumen" tab contains three cards: "Proceso de Solicitud" (with steps: Crear Solicitud, Proceso de Aprobación, and Flujo Automatizado), "Tipos de Flujo" (listing Normal, Paralelo, Aprobación, and Ejecución), and a section for "Flujos Secuenciales", "Flujos con Bifurcación", "Flujos de Aprobación", and "Flujos de Ejecución".

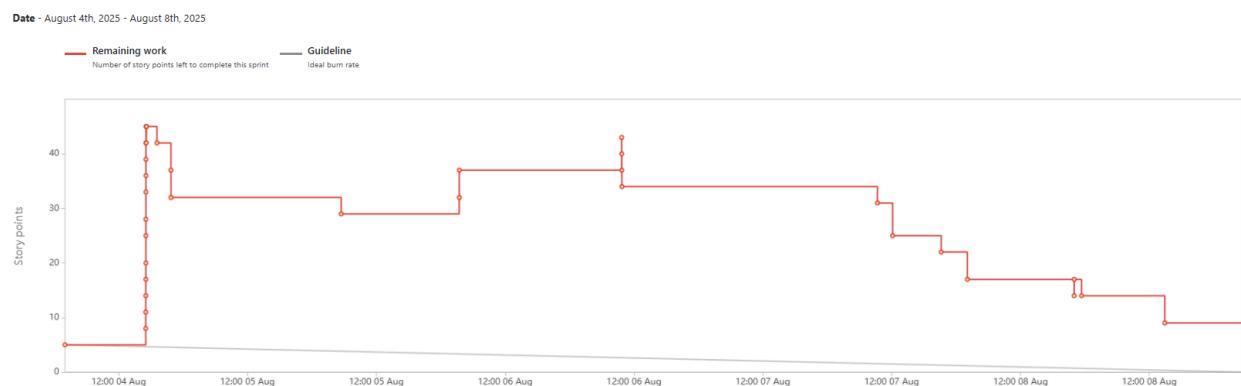


Lista de tareas:

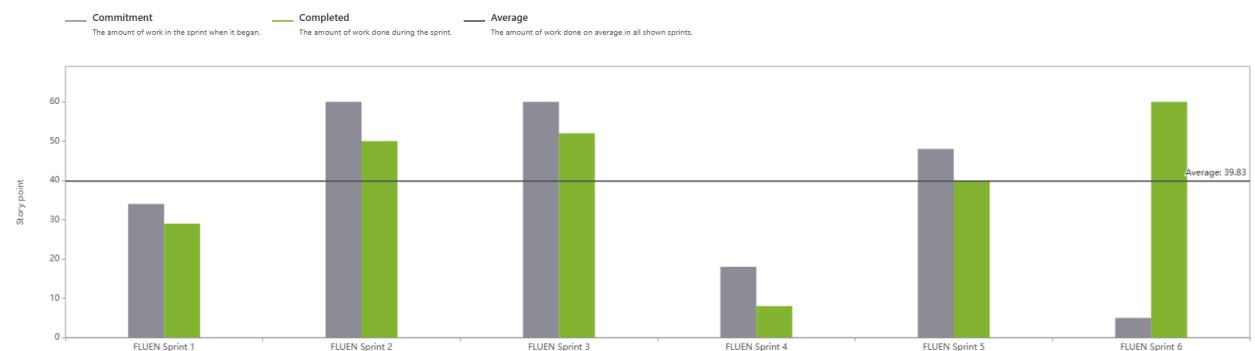
- Lista de tareas concluidas:
 - FLUEN-259 – Tarea 22: Implementar API para gestionar grupos de aprobación (3 SP)
 - FLUEN-263 – Tarea 25: Implementar API para iniciar una solicitud (3 SP)
 - FLUEN-265 – Tarea 27: Implementar lógica para generar FlujoActivo al enviar la solicitud (3 SP)
 - FLUEN-272 – Tarea 31: Implementar API para aprobar o rechazar pasos (3 SP)
 - FLUEN-349 – Tarea 76: Generar funcionamiento de vista conexión entre pasos (3 SP)
 - FLUEN-353 – Tarea 80: Controlador de creación de inputs (2 SP)
 - FLUEN-357 – Tarea 84: Desarrollar API para creación y manipulación de pasos (8 SP)
 - FLUEN-260 – Tarea 23: Desarrollar vista para gestionar grupos de aprobación (3 SP)
 - FLUEN-267 – Tarea 28: Crear vista para visualizar el estado de la solicitud (5 SP)
 - FLUEN-253 – Tarea 18: Desarrollar vista para configurar reglas de aprobación (3 SP)
 - FLUEN-264 – Tarea 26: Desarrollar vista para iniciar una solicitud (5 SP)
 - FLUEN-273 – Tarea 32: Desarrollar vista para aprobar o rechazar pasos (3 SP)
 - FLUEN-280 – Tarea 36: Desarrollar vista para visualizar flujos activos (3 SP)
 - FLUEN-257 – Tarea 21: Desarrollar vista para definir campos dinámicos (5 SP)
 - FLUEN-262 – Tarea 24: Crear modelo y migración para Solicitud (3 SP)
 - FLUEN-354 – Tarea 81: Serializador y deserializador de campos JSON input (5 SP)
- Lista de tareas inconclusas:
 - N/A

Métricas del Sprint

Gráfica Burndown



Métrica de velocidad



Conclusión del Sprint:

Consideramos que merecemos una calificación de 9 en este Sprint, pues tuvimos bastante progreso para el tiempo que duró el sprint. Asimismo, se lograron completar todas las tareas asignadas para el sprint. Se tuvo una velocidad más elevada a diferencia de las anteriores y mejor comunicación con las reuniones diarias. Finalmente, consideramos que siempre hay un rango de mejora en relación con ser más precavidos con las horas de entrega, para así poder entregar antes de fecha para anticipar errores y llegar a presentar más preparados.

Código desarrollado

Front-End: <https://github.com/Fluentis-BPM/fluentis-frontend/tree/develop>

Back-End: <https://github.com/Fluentis-BPM/fluentis-backend/tree/develop>

Pruebas Siguiendo Plan Maestro

Plan de Pruebas Maestro - Proyecto Fluentis

Preparado por:

Nombre
Diego Patzan
Iban Marroquin
Diego López
June Herrera

Introducción

Campo	Descripción
Alcance de las Pruebas	Este plan cubre autenticación (Azure AD, recuperación, cambio de contraseña), gestión de usuarios (roles, departamentos, perfiles), flujos de aprobación (creación, edición, validación, publicación), solicitudes (inicio, estado, ejecución, aprobación, delegación, excepciones), grupos de aprobación (públicos), formularios dinámicos (inputs), visualización (flujos activos, KPIs, métricas), notificaciones, propuestas, votaciones, backups, incidentes, auditoría, integraciones externas (ERP, APIs, LDAP), seguridad (SSL, firewall, permisos), reportes, y despliegue (DevOps). Incluye pruebas funcionales, de integración, regresión, carga, seguridad y E2E.
Riesgos y Suposiciones	Suponemos que Azure AD, APIs y Docker están configurados. Riesgos: retrasos, errores de

Campo	Descripción
	integración, fallos de seguridad, y problemas de despliegue.

Recursos

Tester	% de Participación
Equipo de Desarrollo	80% (32 horas/semana)

Alcance

Descripción
Pruebas de todas las funcionalidades (autenticación, flujos, solicitudes, grupos, KPIs, integraciones, seguridad) y regresión para módulos críticos (aprobaciones, métricas, backups). Incluye CU-01 a CU-136 para cubrir el ciclo completo de usuario en ASOFARMA.

Fuera del Alcance

Descripción
Pruebas de rendimiento extremo (>5000 usuarios) y traducciones (delegadas a externos).

Características a probar

Tipo de Prueba	Detalle
Rango de Fechas	04/08/2025 - 22/08/2025
Pruebas de Funcionalidad	
ID Caso Prueba	Escenario / Variables / Resultado Esperado
F-01: Iniciar Sesión (CU-01)	Escenario: Ingresar credenciales. Variables: Usuario="admin", Contraseña="123". Resultado Esperado: Acceso concedido.
F-03: Gestionar Usuarios (CU-03)	Escenario: Crear usuario. Variables: Nombre="Juan", Rol="Gerente". Resultado Esperado: Usuario creado.
F-04: Ver Métricas Globales (CU-04)	Escenario: Consultar dashboard. Variables: Período="01/08". Resultado Esperado: Indicadores visibles.

Tipo de Prueba	Detalle
F-05: Crear nueva solicitud (CU-05)	Escenario: Diseñar solicitud y paso 0. Variables: Inputs= []. Resultado Esperado: flujo guardado.
F-06: Editar Definición de flujo (CU-06)	Escenario: Modificar flujo. Variables: flujo=1, Cambio="Tiempo". Resultado Esperado: flujo actualizado.
F-07: Eliminar Definición de flujo (CU-07)	Escenario: Eliminar flujo. Variables: flujo=1. Resultado Esperado: flujo eliminado.
F-08: Visualizar Lista de flujos (CU-08)	Escenario: Filtrar flujos. Variables: Estado="activo". Resultado Esperado: Lista filtrada.
F-09: Iniciar Instancia de flujo (CU-09)	Escenario: Iniciar flujo. Variables: flujo=1. Resultado Esperado: Instancia creada.
F-10: Ver Tareas Asignadas (CU-10)	Escenario: Listar tareas. Variables: Usuario=2. Resultado Esperado: Tareas visibles.
F-11: Completar Tarea (CU-11)	Escenario: Completar tarea. Variables: Tarea=1, Datos="Aprobado". Resultado Esperado: Tarea completada.
F-12: Reasignar Tarea (CU-12)	Escenario: Derivar tarea. Variables: Tarea=1, NuevoUsuario=3. Resultado Esperado: Tarea reasignada.
F-13: Diseñar Flujo de Trabajo (CU-13)	Escenario: Conectar actividades. Variables: Flujo=1, Paso="Aprobación". Resultado Esperado: Flujo diseñado.
F-14: Validar Flujo (CU-14)	Escenario: Simular flujo. Variables: Flujo=1. Resultado Esperado: Simulación exitosa.
F-15: Publicar Flujo (CU-15)	Escenario: Poner en línea. Variables: Flujo=1. Resultado Esperado: Flujo disponible.
F-16: Monitorear Instancias (CU-16)	Escenario: Ver estado. Variables: Instancia=1. Resultado Esperado: Estado visible.
F-18: Revisar Historial de Auditoría (CU-18)	Escenario: Consultar trazabilidad. Variables: Usuario=2. Resultado Esperado: Historial visible.
F-19: Descargar Bitácora (CU-19)	Escenario: Exportar auditoría. Variables: Formato="CSV". Resultado Esperado: Archivo descargado.

Tipo de Prueba	Detalle
F-20: Consumir API para Crear Instancia (CU-20)	Escenario: Iniciar vía API. Variables: flujo=1. Resultado Esperado: Instancia creada.
F-21: Consultar Estado vía API (CU-21)	Escenario: Obtener estado. Variables: Instancia=1. Resultado Esperado: Estado devuelto.
F-22: Configurar Roles y Permisos (CU-22)	Escenario: Definir rol. Variables: Rol="Gerente", Permiso="Editar". Resultado Esperado: Rol configurado.
F-23: Parametrizar Notificaciones (CU-23)	Escenario: Crear plantilla. Variables: Evento="Aprobación". Resultado Esperado: Plantilla guardada.
F-24: Adjuntar Documentos a Tarea (CU-24)	Escenario: Subir archivo. Variables: Tarea=1, Archivo="doc.pdf". Resultado Esperado: Archivo adjuntado.
F-25: Versionar Definición de flujo (CU-25)	Escenario: Guardar versión. Variables: flujo=1, Versión="2.0". Resultado Esperado: Versión registrada.
F-26: Filtrar Instancias (CU-26)	Escenario: Filtrar por fecha. Variables: Fecha="01/08". Resultado Esperado: Instancias filtradas.
F-27: Cambiar Contraseña (CU-27)	Escenario: Modificar contraseña. Variables: Nueva="newpass". Resultado Esperado: Contraseña actualizada.
F-28: Actualizar Perfil (CU-28)	Escenario: Editar perfil. Variables: Nombre="Juan". Resultado Esperado: Perfil actualizado.
F-29: Reinicio de Servicios (CU-29)	Escenario: Reiniciar servicio. Variables: Servicio="API". Resultado Esperado: Servicio activo.
F-30: Consultar Logs de Error (CU-30)	Escenario: Ver logs. Variables: Fecha="01/08". Resultado Esperado: Logs visibles.
F-31: Duplicar Flujo (CU-31)	Escenario: Clonar flujo. Variables: Flujo=1. Resultado Esperado: Flujo duplicado.
F-32: Analizar Métricas de flujos (CU-32)	Escenario: Ver KPIs. Variables: flujo=1. Resultado Esperado: Métricas mostradas.

Tipo de Prueba	Detalle
F-33: Ver Landing Page (CU-33)	Escenario: Acceder página. Variables: URL=". Resultado Esperado: Página pública visible.
F-34: Exportar Configuración (CU-34)	Escenario: Descargar configuración. Variables: Formato="JSON". Resultado Esperado: Archivo descargado.
F-35: Cerrar Sesión (CU-35)	Escenario: Terminar sesión. Variables: Usuario=2. Resultado Esperado: Sesión cerrada.
F-36: Gestionar Entornos (CU-36)	Escenario: Crear entorno. Variables: Nombre="Prueba". Resultado Esperado: Entorno configurado.
F-39: Registrar Evento por API (CU-39)	Escenario: Enviar evento. Variables: Evento="stock bajo". Resultado Esperado: Evento registrado.
F-40: Consumir Endpoint de Usuarios (CU-40)	Escenario: Obtener usuarios. Variables: Página=1. Resultado Esperado: Lista con 10 usuarios.
F-41: Autenticación OAuth (CU-41)	Escenario: Obtener token. Variables: ClientID="123". Resultado Esperado: Token JWT válido.
F-42: Refrescar Token (CU-42)	Escenario: Renovar token. Variables: RefreshToken="abc". Resultado Esperado: Nuevo JWT.
F-43: Validar Permisos de API (CU-43)	Escenario: Verificar acceso. Variables: Scope="read". Resultado Esperado: Acceso concedido.
F-44: Generar Reporte Diario (CU-44)	Escenario: Crear informe. Variables: Fecha="01/08/2025". Resultado Esperado: Informe generado.
F-45: Programar Envío de Reportes (CU-45)	Escenario: Configurar envío. Variables: Hora="9:00". Resultado Esperado: Envío programado.
F-46: Filtrar Reportes por Fechas (CU-46)	Escenario: Filtrar rango. Variables: Desde="01/08", Hasta="31/08". Resultado Esperado: Datos filtrados.

Tipo de Prueba	Detalle
F-47: Exportar Reporte a Excel (CU-47)	Escenario: Descargar .xlsx. Variables: Reporte=1. Resultado Esperado: Archivo descargado.
F-48: Ejecutar Pruebas de Estrés (CU-48)	Escenario: Simular carga. Variables: Usuarios=500. Resultado Esperado: Sistema estable.
F-49: Ejecutar Pruebas de Seguridad (CU-49)	Escenario: Escanear vulnerabilidades. Variables: Tipo="SQLi". Resultado Esperado: Sin fallos.
F-50: Gestionar Certificados SSL (CU-50)	Escenario: Subir certificado. Variables: Archivo="cert.pem". Resultado Esperado: HTTPS activo.
F-51: Configurar Firewall (CU-51)	Escenario: Definir reglas. Variables: Puerto=443. Resultado Esperado: Acceso restringido.
F-52: Auditar Accesos Fallidos (CU-52)	Escenario: Listar intentos. Variables: Día="01/08". Resultado Esperado: Lista visible.
F-55: Clonar Definición de flujo (CU-55)	Escenario: Copiar flujo. Variables: Flujo=1. Resultado Esperado: Flujo clonado.
F-56: Agregar Condiciones Avanzadas (CU-56)	Escenario: Definir condición. Variables: Expresión="monto>10000". Resultado Esperado: Condición aplicada.
F-57: Configurar Roles Dinámicos (CU-57)	Escenario: Asignar rol dinámico. Variables: Atributo="Gerente". Resultado Esperado: Rol asignado.
F-58: Alertar Instancias en Error (CU-58)	Escenario: Detectar fallo. Variables: Instancia=1. Resultado Esperado: Notificación enviada.
F-59: Reintentar Tareas Fallidas (CU-59)	Escenario: Reintentar tarea. Variables: Tarea=1. Resultado Esperado: Tarea reejecutada.
F-60: Cancelar Instancia vía API (CU-60)	Escenario: Cancelar flujo. Variables: Instancia=1. Resultado Esperado: Estado "cancelado".
F-61: Pausar Instancia vía API (CU-61)	Escenario: Pausar flujo. Variables: Instancia=1. Resultado Esperado: Estado "pausado".
F-63: Dashboard de Cuellos de Botella (CU-63)	Escenario: Ver cuellos. Variables: Período="01/08". Resultado Esperado: Tareas retrasadas visibles.

Tipo de Prueba	Detalle
F-65: Comparar Versiones de flujo (CU-65)	Escenario: Comparar versiones. Variables: V1=1, V2=2. Resultado Esperado: Diferencias mostradas.
F-68: Rechazar Tarea con Comentario (CU-68)	Escenario: Rechazar tarea. Variables: Tarea=1, Comentario="Falta dato". Resultado Esperado: Tarea rechazada.
F-69: Comentar Tarea (CU-69)	Escenario: Agregar comentario. Variables: Tarea=1, Nota="Revisar". Resultado Esperado: Comentario guardado.
F-70: Suscribirse a Notificaciones (CU-70)	Escenario: Activar notificaciones. Variables: Usuario=2. Resultado Esperado: Alertas recibidas.
F-71: Ver Estado de Notificaciones (CU-71)	Escenario: Consultar historial. Variables: Usuario=2. Resultado Esperado: Historial visible.
F-72: Respaldar Base de Datos (CU-72)	Escenario: Ejecutar backup. Variables: Tipo="completo". Resultado Esperado: Backup generado.
F-73: Restaurar Base de Datos (CU-73)	Escenario: Cargar backup. Variables: Archivo="backup.bak". Resultado Esperado: Datos restaurados.
F-74: Optimizar Índices (CU-74)	Escenario: Reconstruir índices. Variables: Tabla="Usuario". Resultado Esperado: Rendimiento mejorado.
F-75: Archivar Datos Antiguos (CU-75)	Escenario: Archivar datos. Variables: Fecha="01/08/2024". Resultado Esperado: Datos archivados.
F-76: Configurar Copias de Seguridad (CU-76)	Escenario: Definir política. Variables: RPO="1h". Resultado Esperado: Política aplicada.
F-77: Mostrar Tutorial Interactivo (CU-77)	Escenario: Acceder tutorial. Variables: Usuario=1. Resultado Esperado: Tutorial visible.
F-78: Acceder a Sección de Ayuda (CU-78)	Escenario: Buscar ayuda. Variables: Tema="Login". Resultado Esperado: Manual mostrado.

Tipo de Prueba	Detalle
F-79: Abrir Ticket de Soporte (CU-79)	Escenario: Crear ticket. Variables: Descripción="Error". Resultado Esperado: Ticket registrado.
F-80: Gestionar Tickets (CU-80)	Escenario: Cambiar estado. Variables: Ticket=1, Estado="Cerrado". Resultado Esperado: Estado actualizado.
F-81: Documentar Solución (CU-81)	Escenario: Adjuntar solución. Variables: Ticket=1, Solución="Reiniciar". Resultado Esperado: Documento guardado.
F-82: Ajustar Tiempos de Expiración (CU-82)	Escenario: Configurar inactividad. Variables: Tiempo="30min". Resultado Esperado: Sesión cerrada.
F-83: Forzar Cierre de Sesión (CU-83)	Escenario: Expulsar usuario. Variables: Usuario=2. Resultado Esperado: Sesión cerrada.
F-84: Revisar Sesiones Activas (CU-84)	Escenario: Ver dispositivos. Variables: Usuario=2. Resultado Esperado: Lista visible.
F-85: Aprobar Instancia (CU-85)	Escenario: Dar OK final. Variables: Instancia=1. Resultado Esperado: Estado "aprobado".
F-86: Rechazar Instancia (CU-86)	Escenario: Detener ejecución. Variables: Instancia=1. Resultado Esperado: Estado "rechazado".
F-89: Subir Evidencia Adicional (CU-89)	Escenario: Subir archivo. Variables: Solicitud=1, Archivo="evidencia.jpg". Resultado Esperado: Archivo adjuntado.
F-92: Enviar Informe de Pruebas (CU-92)	Escenario: Subir informe. Variables: Archivo="pruebas.pdf". Resultado Esperado: Informe recibido.
F-95: Comparar Métricas entre Períodos (CU-95)	Escenario: Comparar meses. Variables: Mes1="Jul", Mes2="Ago". Resultado Esperado: Diferencias mostradas.
F-96: Escalar Instancias por Tiempo (CU-96)	Escenario: Exceder SLA. Variables: Instancia=1. Resultado Esperado: Notificación enviada.
F-97: Habilitar Modo de Simulación (CU-97)	Escenario: Simular flujo. Variables: Flujo=1. Resultado Esperado: Datos no impactados.

Tipo de Prueba	Detalle
F-98: Generar Flujo desde Plantilla (CU-98)	Escenario: Crear flujo. Variables: Plantilla=1. Resultado Esperado: Flujo generado.
F-99: Bloquear Endpoints (CU-99)	Escenario: Deshabilitar ruta. Variables: Endpoint="/login". Resultado Esperado: Acceso denegado.
F-100: Ejecutar Pruebas E2E (CU-100)	Escenario: Correr Playwright. Variables: Flujo=1. Resultado Esperado: Pruebas exitosas.
F-101: Validar Tiempos de Respuesta (CU-101)	Escenario: Medir latencia. Variables: Endpoint="/api". Resultado Esperado: <500ms.
F-102: Pruebas Cross-Browser (CU-102)	Escenario: Verificar navegadores. Variables: Navegadores="Chrome, Firefox". Resultado Esperado: Funcionalidad intacta.
F-103: Pruebas de Accesibilidad (CU-103)	Escenario: Validar WCAG. Variables: Componente="Formulario". Resultado Esperado: Cumplimiento.
F-104: Pruebas de Internacionalización (CU-104)	Escenario: Cambiar idioma. Variables: Idioma="es". Resultado Esperado: Textos adaptados.
F-105: Pruebas de Corrección de Errores (CU-105)	Escenario: Verificar bug. Variables: Bug=1. Resultado Esperado: Corrección sin regresión.
F-106: Revisar Logs de IP (CU-106)	Escenario: Analizar accesos. Variables: Día="01/08". Resultado Esperado: Historial visible.
F-108: Rotar Claves de API (CU-108)	Escenario: Cambiar clave. Variables: API="123". Resultado Esperado: Nueva clave activa.
F-109: Gestionar Whitelist de IP (CU-109)	Escenario: Permitir rango. Variables: IP="192.168.1.0/24". Resultado Esperado: Acceso restringido.
F-110: Restaurar Servicio (CU-110)	Escenario: Recuperar sistema. Variables: Backup="backup.bak". Resultado Esperado: Servicio activo.
F-111: Verificar Integridad de Despliegue (CU-111)	Escenario: Comparar artefactos. Variables: Versión="1.0". Resultado Esperado: Consistencia confirmada.

Tipo de Prueba	Detalle
F-112: Descargar Acta de Finalización (CU-112)	Escenario: Obtener acta. Variables: Instancia=1. Resultado Esperado: Archivo descargado.
F-113: Validar Firma Digital (CU-113)	Escenario: Revisar sello. Variables: Documento="acta.pdf". Resultado Esperado: Firma válida.
F-114: Auditar Permisos de Roles (CU-114)	Escenario: Verificar accesos. Variables: Rol="Gerente". Resultado Esperado: Permisos correctos.
F-115: Gestionar Tokens SSO (CU-115)	Escenario: Configurar SSO. Variables: Proveedor="AzureAD". Resultado Esperado: SSO activo.
F-118: Validar Expresiones Regulares (CU-118)	Escenario: Probar regex. Variables: Expresión="^\d{3}\$". Resultado Esperado: Validación exitosa.
F-119: Forzar Migración de Datos (CU-119)	Escenario: Ejecutar migración. Variables: Script="migrate.sql". Resultado Esperado: Esquema actualizado.
F-120: Versionar Elementos de Flujo (CU-120)	Escenario: Registrar cambio. Variables: Flujo=1. Resultado Esperado: Versión guardada.
F-121: Generar Heatmap de Actividad (CU-121)	Escenario: Visualizar carga. Variables: Período="01/08". Resultado Esperado: Mapa de calor mostrado.
F-122: Exportar Métricas a BI (CU-122)	Escenario: Publicar datos. Variables: Herramienta="PowerBI". Resultado Esperado: Datos exportados.
F-123: Filtrar Vistas por Favoritos (CU-123)	Escenario: Marcar favorito. Variables: Vista="KPIs". Resultado Esperado: Vista guardada.
F-124: Compartir Vista con Compañero (CU-124)	Escenario: Enviar enlace. Variables: Vista=1, Usuario=2. Resultado Esperado: Enlace recibido.
F-126: Validar Payload JSON Schema (CU-126)	Escenario: Enviar payload. Variables: JSON="malformed". Resultado Esperado: Rechazo.

Tipo de Prueba	Detalle
F-127: Monitorizar Colas de Mensajes (CU-127)	Escenario: Ver estado. Variables: Cola="notificaciones". Resultado Esperado: Estado visible.
F-128: Limpiar Colas Obsoletas (CU-128)	Escenario: Eliminar mensajes. Variables: Cola="vieja". Resultado Esperado: Cola limpia.
F-129: Definir Reglas de Escalamiento (CU-129)	Escenario: Configurar escalamiento. Variables: Condición="SLA>24h". Resultado Esperado: Regla aplicada.
F-130: Importar Flujo desde BPMN (CU-130)	Escenario: Subir archivo. Variables: Archivo="flujo.bpmn". Resultado Esperado: Flujo mapeado.
F-131: Ejecutar Pruebas de Contrato API (CU-131)	Escenario: Validar Swagger. Variables: Endpoint="/api". Resultado Esperado: Cumplimiento.
F-132: Medir Cobertura Unitaria (CU-132)	Escenario: Generar reporte. Variables: Proyecto="Fluentis". Resultado Esperado: >80% cobertura.
F-133: Pruebas de Rendimiento E2E (CU-133)	Escenario: Simular usuarios. Variables: Usuarios=100. Resultado Esperado: Sistema estable.
Pruebas de Carga	
Identificador	Parte / Condición / Resultado Esperado / Método
C-01	API de Aprobación / 500 solicitudes / <2 seg / JMeter
C-02	Dashboard KPIs / 200 usuarios / <1 seg / JMeter
C-03	API Batch (CU-125) / 1000 requests / <5 seg / JMeter
Pruebas de Seguridad	
Identificador	Condición / Elemento / Resultado Esperado
S-01	Acceso no autorizado / Endpoint Login / Código 403
S-02	Inyección SQL / Campo Título / Error bloqueado

Tipo de Prueba	Detalle
S-03	Pruebas de Seguridad (CU-49) / Escaneo XSS / Sin vulnerabilidades

Criterios de Aceptación o Fallo

Id Criterio	Descripción	Aprobación	Fallo
C1	Cobertura de Pruebas	90% de casos exitosos	Reejecutar y ajustar código
C2	Mensajes de Validación	Errores con mensajes claros	Corregir y retestear
C3	Tiempo de Respuesta	<2 seg en APIs críticas	Optimizar y retestear
C4	Cobertura Unitaria	>80% (CU-132)	Aumentar pruebas unitarias

Criterios de Suspensión y Reanudación

Criterio de Suspensión	Criterio de Reanudación
>20% de casos fallidos	Corrección de defectos
Fallo crítico en API	Nueva versión backend disponible
Errores de seguridad	Parches aplicados y retest
Fallo en despliegue (CU-111)	Artefactos corregidos

Infraestructura

Elemento	Detalle
Entorno	Desarrollo local (Windows), Azure (despliegue)
Lenguajes	JavaScript (React), C# (.NET Core)
Frameworks	React, .NET Core
Herramientas	Cypress, NUnit/SpecFlow, JMeter, Playwright

Suposiciones

Descripción
Azure AD, APIs y Docker están configurados. Usuarios de prueba están capacitados. Integraciones externas (ERP, LDAP) tienen acceso preliminar.

Riesgos

No	Riesgos	Probabilidad (1-5)	Impacto (1-5)	Severidad	Plan de Mitigación
1	Retrasos en desarrollo	2	5	10	Revisión semanal
2	Errores en integración	3	4	12	Pruebas exhaustivas
3	Fallos en seguridad	2	5	10	Auditorías regulares
4	Problemas con Docker/Despliegue	2	4	8	Pruebas previas locales
5	Fallos en integraciones externas (CU-37)	2	4	8	Validación previa con terceros

Pruebas automatizadas:

https://www.youtube.com/watch?v=jyMLgQtIXc&ab_channel=IhanMarroquin

Cálculo de Costo Consumido hasta la fecha

1. Tiempo Consumido

Fecha de inicio: 21 de enero de 2025

Fecha actual: 08 de agosto de 2025

Fecha de finalización planeada: 21 de noviembre de 2025

Cálculo del tiempo transcurrido:

Desde el 21 de enero hasta el 08 de agosto de 2025:

Enero: 10 días (21-31)

Febrero: 28 días

Marzo: 31 días

Abril: 30 días

Mayo: 31 días

Junio: 30 días

Julio: 31 días

Agosto: 8 días (hasta el 08)

Total de días: $10 + 28 + 31 + 30 + 31 + 30 + 31 + 8 = 199$ días

Semanas transcurridas: $199 \text{ días} \div 7 \approx 28.43$ semanas

Esfuerzo acumulado por persona:

Horas por persona por semana: 8 horas

Horas totales por persona hasta ahora: $8 \text{ horas/semana} \times 28.43 \text{ semanas} \approx 227.44$ horas

Horas totales del equipo: $227.44 \text{ horas} \times 4 \text{ personas} \approx 909.76$ horas

Tiempo total estimado con 8 horas/semana por persona:

Horas totales por persona: 602.7 horas

Semanas por persona: $602.7 \div 8 \approx 75.34$ semanas

Semanas totales del equipo: $75.34 \div 4 \approx 18.84$ semanas

Días totales: $18.84 \text{ semanas} \times 7 \approx 131.88$ días

Porcentaje de tiempo consumido:

$199 \text{ días} / 131.88 \text{ días} \approx 150.9\%$.

2. Porcentaje Gastado

Costo total estimado: \$60,267 USD.

Esfuerzo total estimado: 2,410.7 horas.

Esfuerzo consumido: 909.76 horas.

Proporción de esfuerzo consumido: $909.76 / 2,410.7 \approx 37.7\%$.

Costo acumulado: $37.7\% \times \$60,267 \approx \$22,720.66$ USD.

Documentación

Retrospectiva del sprint:

Durante este sprint logramos completar con éxito todas las tareas planificadas, incluyendo el desarrollo e integración de APIs y vistas clave para la gestión de flujos de aprobación, solicitudes y monitoreo. A pesar de los ajustes en estimaciones y la incorporación de nuevas tareas a lo largo del ciclo, el equipo mantuvo un ritmo constante y una comunicación efectiva que permitió cumplir con los objetivos sin comprometer la calidad. Este avance consolidó la base funcional del sistema, habilitando procesos completos de creación, gestión y visualización de flujos y solicitudes. El trabajo coordinado y la rápida adaptación a los cambios fueron factores decisivos para el cierre exitoso del sprint.

LOGT

Diego Patzan – 23525

Fecha	Inicio	Fin	interr u pción	Delta tiempo	Fase	Comentarios
28-07-2025	12:00	15:00	0	3:00	Diseño	Diseño de UI para grupos de aprobacion
29-07-2025	15:00	17:00	0	2:00	Fix	FIx en la rutas del proyecto
01-07-2025	14:00	15:00		01:00	revisión	Solucionar problemas de diseño, luego de la implementación
03-07-2025	15:00	15:30		00:30	Logt	Hacer logt

Ihan Marroquin - 23108

Fecha	Inicio	Fin	interr u pción	Delta tiempo	Fase	Comentarios
05-08-2025	10:00	15:00	0	5:00	Diseño	Desarrollar vista para iniciar una solicitud
06-08-2025	14:00	16:00	0	2:00	Diseño	Desarrollar vista para configurar reglas de aprobación
07-08-2025	17:00	20:00	0	3:00	Diseño	Desarrollar vista para aprobar o rechazar pasos
08-08-2025	12:00	14:00	0	2:00	Diseño	Desarrollar vista para visualizar flujos activos

LOGT - Diego López - 23242

Fecha	Inicio	Fin	Interru pción	Delta Tiempo	Fase	Comentarios
18-07-2025	9:00	11:30	0	2:30	Implementación	Tarea 22: Implementar API para gestionar grupos de aprobación
18-07-2025	14:00	17:00	0	3:00	Implementación	Tarea 25: Implementar API para iniciar una solicitud

19-07-2025	13:00	16:00	0	3:00	Creación	Tarea 27: Implementar lógica para generar FlujoActivo al enviar la solicitud
25-07-2025	17:00	17:30	0	00:30	Logt	Tarea 31: Implementar API para aprobar o rechazar pasos

LOGT - June Herrera - 231038

Fecha	Inicio	Fin	Interrupción	Delta Tiempo	Fase	Comentarios
03-08-2025	13:00	16:00	0	3:00	Arreglos	Arreglos UI y flujos
04-08-2025	17:00	18:30	0	1:30	Refactorizar	Refactorizar controladores y DTOs de Input.
06-08-2025	10:00	14:00	0	4:00	Creación	Inputs dinámicos, fullscreen de editor de flujos y mejorar estilos.
08-08-2025	17:00	17:30	0	00:30	Logt	Hacer logt