

UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de software II

Sección 20

Ing. Pablo Koch



Corte Sprint 5: Fluentis

Diego Patzán - 23525

Diego Lopez - 23242

June Herrera - 231038

Ihan Marroquin – 23108

GUATEMALA, 25 de julio 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 @azure/msal-react) 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.

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	Task	DEVOPS: CON...	DONE	JW	2
FLUEN-313	Tarea 59: Implementar hooks de pre-commit	Task	DEVOPS: CON...	DONE	JW	2
FLUEN-333	Tarea 61: Setupear las diferentes tecnologías a los repositorio...	Task	DEVOPS: CON...	DONE	D	2
FLUEN-314	Tarea 60: Configurar entornos de desarrollo con Docker Com...	Task	DEVOPS: CON...	DONE	DL	3
FLUEN-312	Tarea 58: Configurar linters y formateadores para el código	Task	DEVOPS: CON...	DONE	D	3
FLUEN-315	Tarea 62: Configurar variables de entorno	Task	DEVOPS: CON...	DONE	IM	2
FLUEN-316	Tarea 63: Configurar CI/CD para el backend	Task	DEVOPS: CON...	DONE	DL	5
FLUEN-330	Tarea 77: Inicializar proyecto de React con Vite para el frontend.	Task	DEVOPS: CON...	DONE	D	5
FLUEN-329	Tarea 76: Inicializar proyecto de ASP .NET CORE para el backe...	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	Task	DEVOPS: CON...	DONE	D	5
FLUEN-325	Tarea 72: Aplicar migraciones iniciales a la DB.	Task	DEVOPS: CON...	DONE	JW	5
FLUEN-331	Tarea 78: Creación de modelos y estructuras para la DB en el ...	Task	DEVOPS: CON...	DONE	JW	6
FLUEN-334	Tarea 80: Creación de mock up data para el backend e inserci...	Task	DEVOPS: CON...	DONE	IM	5
FLUEN-234	Tarea 6: Creación, lectura, actualización y eliminación de usar...	Task	ÉPICA 1: AUTE...	DONE	JW	3
FLUEN-335	Tarea 81: Creación operaciones crud de rol y departamento d...	Task	ÉPICA 1: AUTE...	DONE	JW	3
FLUEN-336	Tarea 82: Crear Azure Entra Id with user mocks	Task	ÉPICA 1: AUTE...	DONE	DL	6
FLUEN-228	Tarea 1: Configurar Azure AD para inicio sesion en el backend...	Task	ÉPICA 1: AUTE...	DONE	DL	5
FLUEN-229	Tarea 2: Implementar middleware de autenticación en ASP.NE...	Task	ÉPICA 1: AUTE...	DONE	DL	3
FLUEN-233	Tarea 5: Modificar la tabla Usuario para incluir oid	Task	ÉPICA 1: AUTE...	DONE	IM	2
FLUEN-231	Tarea 4: Desarrollar página de landigpage	Task	ÉPICA 1: AUTE...	DONE	D	2
FLUEN-338	Tarea 84: Crear react router en el fronted con rutas protegidas	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

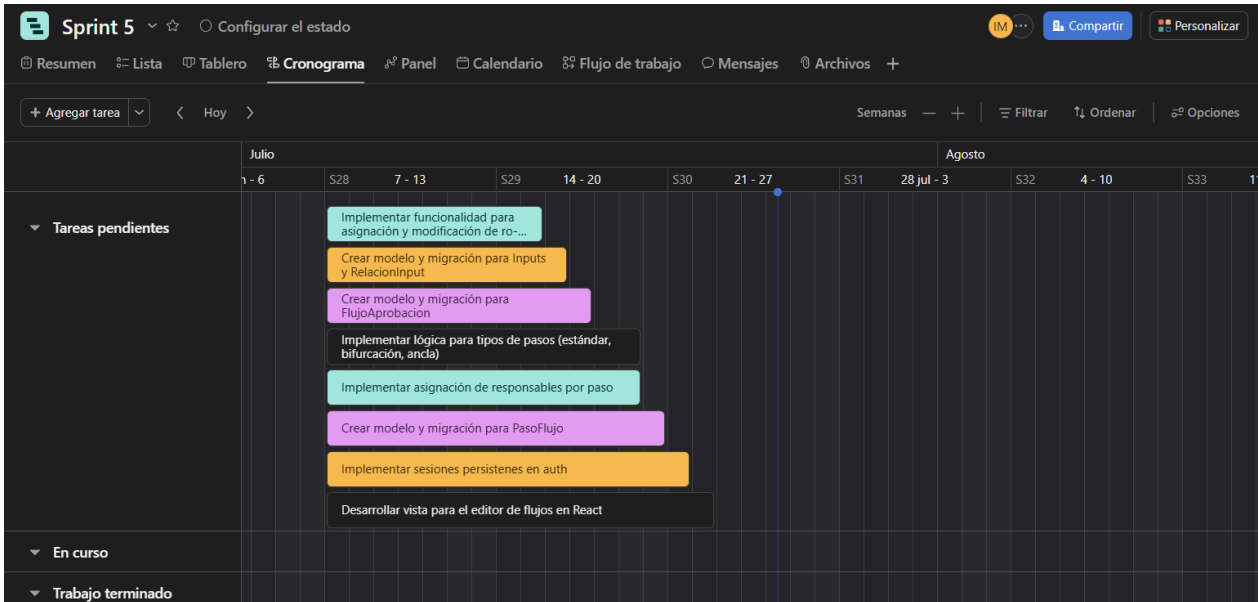
Pila del Sprint

Historias de usuario en el Sprint:

- Implementar funcionalidad para asignación y modificación de roles, usuarios, departamentos, y grupos:
 - Descripción: Implementación de las asignaciones de roles, departamentos y grupos para los usuarios
 - Horas estimadas de desarrollo: 15 horas
 - Puntos de historia asignados a la tarea: 10
 - Responsable de desarrollarla: Diego Patzan y June Herrera
 - Fecha probable de terminación de la tarea: 15 de julio
- Implementar sesiones persistentes en auth:
 - Descripción: Lograr mantener mayor tiempo el token de sesion iniciada, ademas

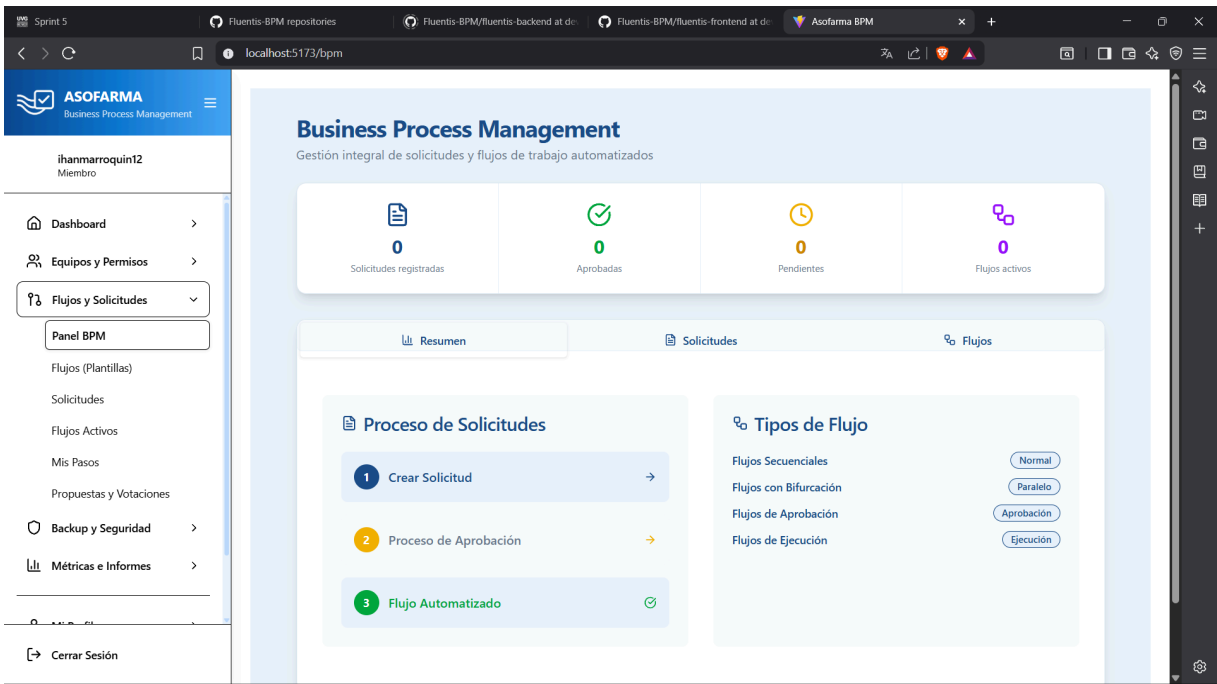
- de guardarlo en local storage para el refresco de la pagina
 - Horas estimadas de desarrollo: 2 horas
 - Puntos de historia asignados a la tarea: 5
 - Responsable de desarrollarla: June Herrera
 - Fecha probable de terminación de la tarea: 16 de julio
- Crear modelo y migración para FlujoAprobacion
 - Descripción: Definir el modelo en el backend y generar la migración para la tabla FlujoAprobacion.
 - Horas estimadas de desarrollo: 3 horas
 - Puntos de historia asignados a la tarea: 2
 - Responsable de desarrollarla: June Herrera
 - Fecha probable de terminación de la tarea: 17 de julio
- Desarrollar vista para el editor de flujos en React
 - Descripción: Crear una interfaz gráfica de drag & drop para diseñar flujos con pasos secuenciales y paralelos.
 - Horas estimadas de desarrollo: 20 horas
 - Puntos de historia asignados a la tarea: 8
 - Responsable de desarrollarla: Diego Patzan E ihan Marroquin
 - Fecha probable de terminación de la tarea: 22 de julio
- Implementar lógica para tipos de pasos (estándar, bifurcación, ancla)
 - Descripción: Añadir funcionalidad en el backend para manejar los diferentes tipos de pasos (normal, bifurcacion, union).
 - Horas estimadas de desarrollo: 6 horas
 - Puntos de historia asignados a la tarea: 5
 - Responsable de desarrollarla: Diego Lopez
 - Fecha probable de terminación de la tarea: 19 de julio
- Implementar asignación de responsables por paso
 - Descripción: Desarrollar la lógica en el backend para asignar responsables por nombre, rol, relación o grupo.
 - Horas estimadas de desarrollo: 6 horas
 - Puntos de historia asignados a la tarea: 5
 - Responsable de desarrollarla: Diego Lopez
 - Fecha probable de terminación de la tarea: 19 de julio
- Crear modelo y migración para Inputs y RelacionInput
 - Descripción: Definir modelos y migraciones para manejar campos dinámicos en formularios.
 - Horas estimadas de desarrollo: 4 horas
 - Puntos de historia asignados a la tarea: 3
 - Responsable de desarrollarla: June Herrera e Ihan Marroquin
 - Fecha probable de terminación de la tarea: 21 de julio
- Crear modelo y migración para PasoFlujo
 - Descripción: Definir el modelo y migración para la tabla PasoFlujo.
 - Horas estimadas de desarrollo: 3 horas
 - Puntos de historia asignados a la tarea: 2
 - Responsable de desarrollarla: June Herrera y Diego Lopez
 - Fecha probable de terminación de la tarea: 20 de julio

Calendario de planificación del Sprint:



Resultados del Sprint

Software funcionando

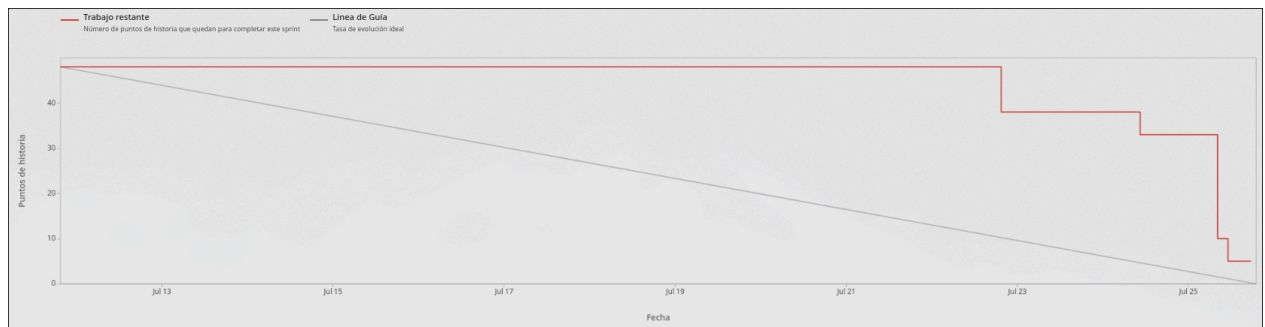


Lista de tareas:

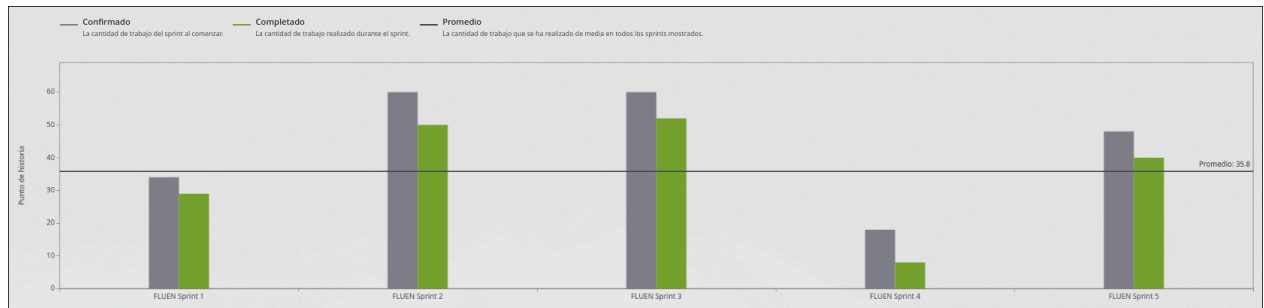
- Lista de tareas concluidas:
 - Implementar funcionalidad para asignación y modificación de roles, usuarios, departamentos, y grupos:
 - Implementar sesiones persistentes en auth
 - Crear modelo y migración para FlujoAprobacion
 - Desarrollar vista para el editor de flujos en React
 - Implementar lógica para tipos de pasos (estándar, bifurcación, ancla)
 - Implementar asignación de responsables por paso
 - Crear modelo y migración para Inputs y RelacionInput
 - Crear modelo y migración para PasoFlujo
- Lista de tareas que se planificaron y no se pudieron concluir:
 - Desarrollar vista para definir campos dinámicos

Métricas del Sprint

Gráfica Burndown



Métrica de velocidad



Conclusión del Sprint:

Consideramos que merecemos una calificación de 8 en este Sprint, ya que aunque tuvimos un progreso muy considerable, no logramos completarlo completo, dejando únicamente una tarea pendiente y realizando exitosamente todas las demás y aunque completamos varias, tuvimos inconvenientes con lograr completarlo ya que comenzamos algo tarde. La manera mas facil y sensata de mejorar es empezar el Sprint desde el primer dia que se asigna para poder satisfacer la demanda que necesitamos cumplir, teniendo siempre en cuenta la comunicación entre nosotros.

Código desarrollado

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

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

Cálculo de Costo y Tiempo del proyecto

Costo total:

1. PCU

$$PCU = FPA + FPCU = 3 + (15 \times 10) = 3 + 150 = 153$$

2. Factores

- $FCT=1.01$ $FCT = 1.01$
- $FA=0.65$ $FA = 0.65$

3. Puntos ajustados (PCUA)

$$PCUA=153 \times 1.01 \times 0.65 \approx 100.44$$

4. Esfuerzo total (horas-hombre)

$$E=100.44 \times 24 \approx 2\,410.7 \text{ h}$$

5. Costo total

$$C=E \times K \times THP = 2\,410.7 \times 2 \times 12.5 = 2\,410.7 \times 25 \approx \$60\,267$$

Distribución por desarrollador (equipo de 4)

- **Horas por persona:** $2\,410.74 \approx 602.7 \text{ h}$
- **Duración aproximada:**
 - Dedicando 40 h/semana $\Rightarrow 602.7 \text{ h} \div 40 \text{ h/sem} \approx 15.1 \text{ semanas}$
 - Dedicando 20 h/semana $\Rightarrow 602.7 \text{ h} \div 20 \text{ h/sem} \approx 30.1 \text{ semanas}$

> **Costo total estimado:** \$60 267 USD

Pruebas

Selección de herramientas para pruebas unitarias:



The infographic is titled "SELECCIÓN DE HERRAMIENTAS" and is divided into two columns. The left column is for the backend, featuring the .NET Foundation logo and the title "BACKEND - XUNIT". It describes xUnit as a modern, extensible, and parallel execution framework suitable for projects that need to stay updated and efficient in testing. It lists reasons: a clean API, active community support, and the ability to handle complex tests. The right column is for the frontend, featuring a Jest logo and the title "FRONTEND - JEST CON REACT TESTING LIBRARY". It describes Jest as a recommended tool by the React community, easy to configure (especially with Create React App), and promoting good testing practices and user experience. It lists reasons: simplicity, official support, and abundant resources. The background of the infographic shows a modern office with large windows and people working at desks.

SELECCIÓN DE HERRAMIENTAS



BACKEND - XUNIT

Se seleccionó xUnit por su enfoque moderno, extensibilidad y soporte para ejecución paralela, adecuado para un proyecto en desarrollo que busca mantenerse actualizado y eficiente en pruebas.

Razones: API limpia, soporte comunitario activo, y capacidad para manejar pruebas complejas.



FRONTEND - JEST CON REACT TESTING LIBRARY

Se seleccionó Jest con React Testing Library, ya que es recomendado por la comunidad React, fácil de configurar (especialmente con Create React App), y promueve buenas prácticas de prueba, enfocándose en la experiencia del usuario.

Razones: simplicidad, soporte oficial, y recursos abundantes.

Evidencia de ejecución de pruebas unitarias:

<https://youtu.be/Vy8guf-CiEw>

Interacción con el usuario

Nuestro Product Owner es Manuel López, encargado de la farmacéutica a nivel Centroamericano y Caribe

Documentación

Retrospectiva del sprint:

Durante el desarrollo del sprint se logró completar nuevas funcionalidades claves como lo es el inicio del desarrollo del módulo de solicitudes y flujos. Esto dio gran valor a la aplicación, mostrando el potencial de la misma; sin embargo, durante el sprint quedaron pendientes 5 puntos de historia. Esto se debe a que nos encontramos con fenómeno que sufrimos todos los integrantes. Debido a las vacaciones de medio año, hubo un gran espacio donde no se siguió desarrollando el proyecto y se desconectó del mismo, debido a esto retomar de nuevo el proyecto fue un proceso lento que retraso el desarrollo. No obstante, a pesar de esto, durante los últimos días del sprint se terminaron varias tareas, logrando con el objetivo de cumplir el flujo de creación de solicitudes y migración al flujo al ser aprobado. Por lo tanto, en definitiva se pudo ser más eficiente durante este sprint; sin embargo, esto es debido a que como grupo buscamos ser mejores en cada iteración, y esto se ve reflejado en la velocidad de puntos de este sprint, que ha sido mayor que al promedio durante todo el desarrollo.

LOGT

Diego Patzan – 23525

Fecha	Inicio	Fin	interru pción	Delta tiempo	Fase	Comentarios
22-07-2025	12:00	15:00	0	3:00	Diseño	Inicio con el diseño de la vista bpm
24-07-2025	15:00	17:00	0	2:00	Union	Iniciar la union del diseño con el código actual
24-07-2025	14:00	15:00		01:00	revisión	Solucionar problemas de diseño, luego de la implementación

25-07-2025	15:00	15:30		00:30	Logt	Hacer logt
------------	-------	-------	--	-------	------	------------

Ihan Marroquin - 23108

Fecha	Inicio	Fin	interrupción	Delta tiempo	Fase	Comentarios
24-07-2025	15:00	17:00	0	3:00	Diseño	Iniciar la union del diseño con el código actual
25-07-2025	12:00	16:00	0	4:00	Diseño	Arreglo completo de la UI para mejorar la estética general
25-07-2025	17:00	17:30	0	00:30	Logt	Hacer logt

LOGT - Diego López - 23242

Fecha	Inicio	Fin	Interrupción	Delta Tiempo	Fase	Comentarios
18-07-2025	9:00	11:30	0	2:30	Implementación	Implementación de lógica para tipos de pasos (estándar, bifurcación, ancla)
18-07-2025	14:00	17:00	0	3:00	Implementación	Implementacion de asignación de responsables por paso

19-07-2025	13:00	16:00	0	3:00	Creación	Creación de modelo y migración para FlujoAprobacion
25-07-2025	17:00	17:30	0	00:30	Logt	Hacer logt

LOGT - June Herrera - 231038

Fecha	Inicio	Fin	Interrupción	Delta Tiempo	Fase	Comentarios
19-07-2025	13:00	16:00	0	3:00	Creación	Creación de modelo y migración para FlujoAprobacion
19-07-2025	17:00	18:30	0	1:30	Implementación	Implementación de sesiones persistenes en auth
20-07-2025	10:00	14:00	0	4:00	Creación	Creación de modelo y migración para PasoFlujo
25-07-2025	17:00	17:30	0	00:30	Logt	Hacer logt