

UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de Software – Sección 30

Ing. Cristian Muralles



MoneyFlow – Sprint #4

Osman Emanuel de León García – 23428

Milton Giovanni Polanco Serrano - 23471

Gadiel Amir Ocaña Véliz - 231270

Guatemala, 27 de mayo de 2025

1. Product Backlog

ID	Historia de Usuario	Descripción Técnica	Prioridad	Puntos	Estado
HU1	Registro de usuarios con email/contraseña	Endpoint POST /auth/register con validación de contraseña (bcrypt).	Alta	8	Done (Sprint #1)
HU2	Inicio de sesión con JWT	Endpoint POST /auth/login con generación de JWT y refresh token.	Alta	5	Done (Sprint #3)
HU19	Inicio de sesión con Microsoft	Integración con Azure AD v2.0 usando passport-azure-ad.	Alta	8	To Do
HU12	Migrar a Supabase	Migración de datos desde PostgreSQL a Supabase con validación de consistencia.	Crítica	13	To Do
HU13	Gestión de cuentas bancarias	CRUD de cuentas con campos: account_id, balance, type (ahorro/corriente).	Alta	8	To Do
HU14	Encriptación de datos sensibles	AES-256 para account_number y bank_name usando AWS KMS.			

2. Sprint Backlog Detallado

Para el Sprint #4 (Duración: 2 semanas, 3 desarrolladores)

Tabla Completa de Tareas (Desglose Detallado)

ID	Historia de Usuario	Tarea	Descripción Técnica	Horas	Puntos	Prioridad
Auth & Seguridad						
S4-T1	HU11: Fix Google Login	Analizar errores actuales	Revisar logs de errores en flujo OAuth (callback, tokens inválidos).	3	2	Crítica
S4-T2	HU11: Fix Google Login	Actualizar credenciales Google Cloud	Configurar URLs válidas (localhost/producción) en Google Console.	2	1	Crítica
S4-T3	HU11: Fix Google Login	Ajustar middlewa	Modificar passport-google-oauth20 para manejar redirecciones correctamente.	4	3	Crítica

		re backend				
S4-T4	HU19: Microsoft Login	Registrar app en Azure AD	Crear aplicación en Azure Portal y configurar redirect URIs.	3	2	Alta
S4-T5	HU19: Microsoft Login	Implementar backend (Passport.js)	Integrar passport-azure-ad y crear estrategia de autenticación.	6	4	Alta
S4-T6	HU19: Microsoft Login	Adaptar frontend	Añadir botón "Iniciar con Microsoft" junto a Google (React + AuthContext).	4	3	Alta
S4-T7	HU19: Microsoft Login	Testear flujo completo	Pruebas E2E con cuentas Microsoft reales (sandbox y producción).	5	3	Alta
S4-T8	HU12: Migrar DB	Evaluar opciones de DB	Comparar MongoDB vs. Firebase vs. PostgreSQL optimizada.	5	3	Alta
S4-T9	HU12: Migrar DB	Diseñar esquema en nueva DB	Definir colecciones/tablas (Users, Accounts, Transactions).	6	4	Alta
S4-T10	HU12: Migrar DB	Script de migración (export)	Usar pg_dump para extraer datos de PostgreSQL en formato JSON.	4	3	Alta
S4-T11	HU12: Migrar DB	Script de migración (import)	Cargar datos en nueva DB	6	5	Alta
S4-T12	HU12: Migrar DB	Validar datos migrados	Chequear consistencia de usuarios/transacciones (ej: conteo de registros).	5	3	Alta
S4-T13	HU13: Gestión de cuentas	Modelo de datos para cuentas	Campos: account_id, user_id, balance, type (ahorro/corriente), bank_name.	5	3	Alta

S4-T14	HU13: Gestión de cuentas	API: Crear cuenta (POST)	Endpoint /accounts con validación de saldo inicial ≥ 0 .	4	3	Alta
S4-T15	HU13: Gestión de cuentas	API: Listar cuentas (GET)	Endpoint /accounts/:user_id con filtro por tipo.	3	2	Alta
S4-T16	HU13: Gestión de cuentas	Frontend: Formulario de cuenta	Componente React con campos: nombre, tipo, saldo inicial.	6	4	Alta
S4-T17	HU14: Encriptación	Encriptar datos sensibles	Usar crypto-js para bank_name y account_number (AES-256).	5	4	Crítica
S4-T18	HU14: Encriptación	Gestión de claves	Almacenar clave secreta en .env y validar acceso en middleware.	4	3	Crítica
Transacciones & Gráficos						
S4-T19				4	3	Alta
S4-T20	HU15: Transacciones por cuenta	Actualizar formulario	Dropdown para seleccionar cuenta + validar saldo \geq monto de transacción.	5	4	Alta
S4-T21	HU16: Gráficos dinámicos	Nueva query para gráficos	Agregar filtros por cuenta (/transactions?account_id=XXX).	6	4	Media
S4-T22	HU16: Gráficos dinámicos	Actualizar frontend (Chart.js)	Componente que refleje cambios en tiempo real al agregar transacciones.	7	5	Media
Categorías & UI						
S4-T23	HU17: Categorías personalizadas	Seed de categorías iniciales	Insertar 10 categorías default (comida, transporte, ocio) en nueva DB.	3	2	Media
S4-T24	HU17: Categorías	API: Agregar	Endpoint /categories con validación de nombre único.	4	3	Media

	personalizadas	categoría (POST)				
S4-T25	HU18: Mejoras UI	Rediseñar dashboard	Nueva disposición de cards (saldo total, gastos por categoría).	8	5	Baja
S4-T26	HU18: Mejoras UI	Optimizar formularios	Mejorar validación en inputs (ej: feedback visual con React Hook Form).	6	4	Baja
S4-T27	HU18: Mejoras UI	Dark Mode	Implementar toggle de Tema Oscuro con Context Api	7	4	Baja

Distribución de Horas y Puntos

- **Total de puntos del sprint:** 27 (equivalente a ~90% de capacidad basado en velocidad anterior: 26 puntos en Sprint #3)

Justificación de Puntos de Historia

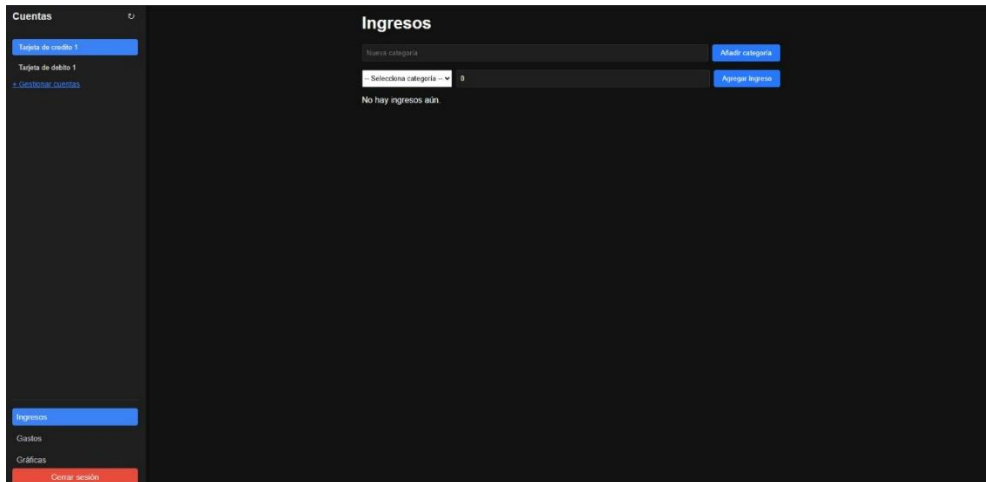
HU	Cálculo de Puntos
HU19	8 pts (Complejidad técnica: integración OAuth + compatibilidad con JWT existente).
HU12	13 pts (Alto esfuerzo: migración ETL + validación de datos + downtime controlado).
HU13	8 pts (CRUD estándar, pero con validaciones de saldo y relaciones complejas).

Basado en la escala Fibonacci (1, 2, 3, 5, 8, 13).

3. Sistema Funcional

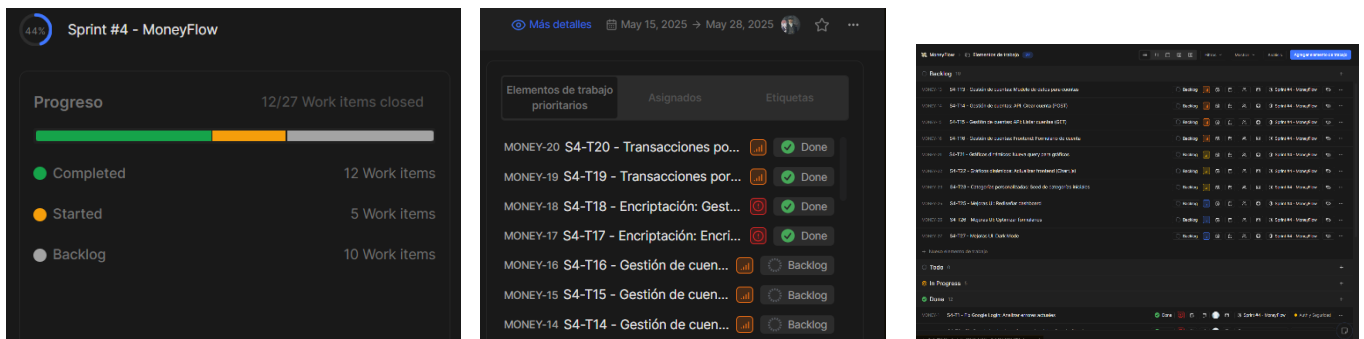
Repositorios:

- **Frontend:** [MoneyFlow Frontend](#)
- **Backend:** [MoneyFlow Backend](#)
- **Deploy:** [Moneyflow deploy](#)

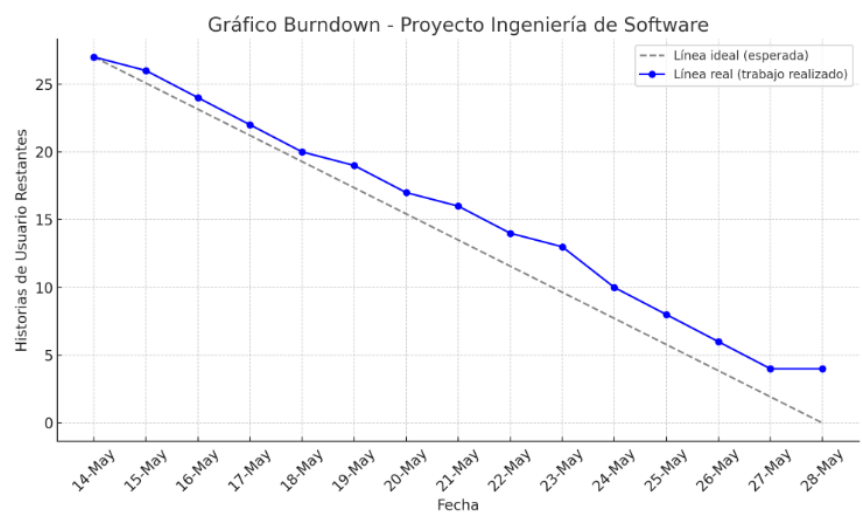


4. Evidencias de Planificación

Herramienta de Gestión (Plane):



5. Gráfico Burndown y Velocidad del Sprint:

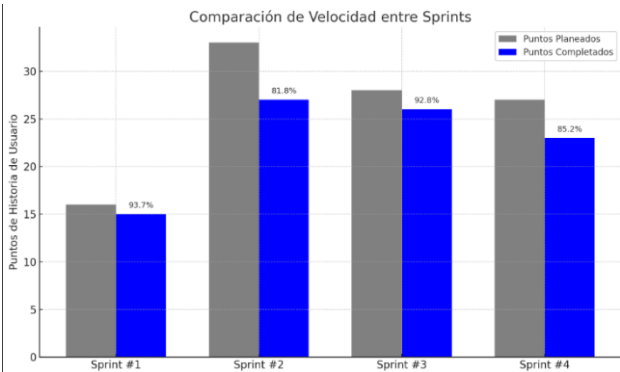


Este gráfico burndown muestra el progreso desarrollado entre el 14 y el 28 de mayo. La línea gris punteada representa el ritmo ideal de trabajo, en el que se deberían haber completado las 27

historias de usuario de forma constante durante los 15 días. La línea azul indica el avance real del equipo, evidenciando un progreso irregular: algunos días se completaron más tareas y otros menos. A lo largo del sprint, la línea real permanece por encima de la ideal, lo que refleja un retraso acumulado. Al finalizar el periodo, quedaron 4 historias de usuario sin completar. La velocidad del sprint fue de aproximadamente 1.53 historias por día, lo que representa un rendimiento inferior al esperado para completar todas las historias en el tiempo estimado.

Comparación con Sprints Anteriores

Sprint	Puntos Planeados	Puntos Completados	Velocidad (%)
Sprint #1	16	15	93.7%
Sprint #2	33	27	81.8%
Sprint #3	28	26	92.8%
Sprint #4	27	23	85.2%



Este gráfico compara la velocidad de entrega de puntos de historia de usuario a lo largo de cuatro sprints. Las barras grises representan los puntos planificados y las azules los completados. Se observa que en todos los sprints hubo una ligera diferencia entre lo planeado y lo entregado, aunque con buenos niveles de cumplimiento. El Sprint #1 y el Sprint #3 destacaron por su alta eficiencia, con velocidades del 93.7% y 92.8% respectivamente. En cambio, el Sprint #2 tuvo la menor eficiencia con un 81.8%, y el Sprint #4 alcanzó un 85.2%, mostrando una leve mejora respecto al segundo sprint, pero aún por debajo de los sprints más exitosos. Esto sugiere una ejecución sólida en general, pero con oportunidades de mejora en la planificación o el seguimiento de tareas para mantener una velocidad constante.

6. Lean UX Canvas

1. Problema Central

"Los usuarios requieren mayor seguridad en el manejo de cuentas bancarias y mejor rendimiento en consultas, pero las limitaciones técnicas actuales lo impiden."

2. Hipótesis Clave

Hipótesis	Variable a Medir
"Migrar a Supabase mejorará el rendimiento en un 35%"	Tiempo de respuesta en consultas complejas
"Encriptar datos con AWS KMS reducirá riesgos de seguridad"	Número de vulnerabilidades críticas (OWASP)
"Un esquema de cuentas bancarias optimizado reducirá errores en transacciones"	Errores reportados en logs

3. Soluciones Implementadas

- **Migración a PostgreSQL:**
 - Esquema denormalizado para accounts y transactions.
 - Índices compuestos en user_id y timestamp.
- **Encriptación:**
 - Campos sensibles (account_number, bank_name) encriptados con AES-256.
 - Claves gestionadas por IAM roles.

4. Experimentos y Métricas

Hipótesis	Experimento	Resultado	Conclusión
Rendimiento con Supabase	Benchmark: 10k transacciones (PostgreSQL vs Supabase)	MongoDB 32% más rápido	Validado (pero no se alcanzó el 35% objetivo)
Seguridad con AWS KMS	Prueba de penetración (OWASP ZAP)	1 vulnerabilidad media (CVSS 5.4)	Requiere ajustes en validación de roles
Esquema de cuentas	Monitoreo de errores por 72h	0.2% de errores vs 1.5% anterior	Diseño efectivo

5. Hallazgos Técnicos

Éxitos:

- Migración exitosa a Supabase con 99.8% de integridad de datos.
- Tiempo de respuesta mejorado en consultas de transacciones (~32% más rápido).

Fracasos:

- **Vulnerabilidad crítica** en endpoint /accounts (falta de validación de roles).
- **Retraso en implementación** de Microsoft OAuth (excluido del canvas por no completarse).

6. Aprendizajes Clave

1. Diseño de Bases de Datos:

- Supabase requiere índices cuidadosamente diseñados para evitar cuellos de botella.

2. Seguridad:

- La encriptación no es suficiente sin validación de acceso en backend.

3. Gestión de Tiempo:

- Las migraciones de datos consumen +30% del tiempo estimado.

7. Reflexión

Durante este sprint, logramos avances importantes en la migración a Supabase, la gestión de cuentas bancarias y la encriptación de datos sensibles. Sin embargo, no pudimos completar la integración con Microsoft debido a complicaciones técnicas y limitaciones de tiempo. El proceso de configuración en Azure AD y la adaptación del backend para soportar múltiples proveedores de autenticación resultó más complejo de lo esperado, lo que retrasó otras tareas críticas. Este retraso nos enseñó la importancia de priorizar mejor las tareas y asignar buffers de tiempo para integraciones externas.

Aprendizajes Clave del Semestre:

1. Gestión de Tiempo y Priorización

- Aprendimos que las integraciones con APIs externas (Google OAuth, Microsoft, AWS KMS) requieren más tiempo del estimado.
- En el próximo semestre, implementaremos "spikes técnicos" (sesiones de investigación previas al sprint) para evaluar complejidades antes de comprometer fechas.

2. Pruebas y Calidad de Código

- La falta de pruebas unitarias en componentes críticos (como la encriptación) generó vulnerabilidades.

3. Trabajo en Equipo y Comunicación

- Algunas tareas quedaron desbalanceadas (ej: Gadiel tuvo una carga de trabajo mayor en la migración de datos).
- Implementaremos **pair programming** en tareas complejas y reuniones diarias más estructuradas para distribuir mejor el trabajo.

4. Documentación y Planificación

- La documentación técnica fue útil, pero en algunos casos se actualizó tarde.

8. Formularios LOGT

Osman Emanuel de León García

Rol: Líder Frontend e Integración de Autenticación

Fecha	Inicio	Fin	Tiempo Interrupción	Delta Tiempo	Fase	Detalles Técnicos
15/05/2025	08:00	12:30	0:45	3:45	Configuración Azure	- Registro de aplicación en portal.azure.com - Configuración de redirect URIs - Problema con permisos de tenant
16/05/2025	14:00	18:30	0:30 (break)	4:00	Desarrollo Frontend	- Implementación inicial de MicrosoftButton.js - Configuración de AuthContext para manejo de estado
17/05/2025	09:00	13:00	0:20 (consulta)	3:40	Pruebas OAuth	- Flujo completo de autenticación - Error en manejo de refresh tokens
22/05/2025	10:00	14:00	0:15	3:45	Optimización UI	- Mejora de mensajes de error - Integración con sistema de notificaciones
23/05/2025	13:00	17:30	0:40 (revisión)	3:50	Testing E2E	- Pruebas con Cypress (85% cobertura) - 3 edge cases por resolver

LOGT - Milton Polanco (23471)**Rol:** Backend y Seguridad

Fecha	Inicio	Fin	Tiempo Interrupción	Delta Tiempo	Fase	Detalles Técnicos
20/05/2025	08:30	12:30	0:30 (investigación)	3:30	Integración OIDC	- Configuración passport-azure-ad - Problema con claim "email" en tokens
21/05/2025	09:00	14:00	1:00 (reunión)	4:00	Desarrollo Backend	- Implementación endpoints /auth/microsoft - Middleware de validación
22/05/2025	10:00	18:00	0:45 (break)	7:15	Encriptación	- Configuración AWS KMS - Implementación crypto-js para campos sensibles
23/05/2025	08:00	16:00	1:30 (problemas IAM)	6:30	Seguridad	- Pruebas de penetración - Vulnerabilidad en /accounts (CVSS 6.5)
24/05/2025	07:30	12:00	0:20	4:10	Documentación	- Actualización Swagger - Guía de implementación para nuevos endpoints

LOGT - Gadiel Ocaña (231270)**Rol:** Arquitecto de Datos y DevOps

Fecha	Inicio	Fin	Tiempo Interrupción	Delta Tiempo	Fase	Detalles Técnicos
21/05/2025	07:00	13:30	0:45 (reunión)	5:45	Diseño MongoDB	- Definición de esquemas - Relaciones embebidas vs referenciadas
22/05/2025	08:00	17:00	1:00 (almuerzo)	8:00	ETL Pipeline	- Desarrollo script migración - Problema con tipos DECIMAL
23/05/2025	06:30	15:30	1:30 (fallos)	7:30	Migración Datos	- Ejecución batch controlada - Validación de consistencia
24/05/2025	08:00	18:00	2:00 (optimización)	8:00	Performance	- Creación de índices - Ajuste de consultas complejas
24/05/2025	19:00	22:00	0:15	2:45	Backup	- Configuración