

UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de Software – Sección 20

Ing. Pablo Barreno Koch



Sprint #9 - MoneyFlow

Osman Emanuel de León García – 23428

Milton Giovanni Polanco Serrano - 23471

Gadiel Amir Ocaña Véliz - 231270

Guatemala, 17 de octubre de 2025

Sprint #9 - MoneyFlow

Product Backlog

Historias de Usuario Completadas y Pendientes

ID	Historia de Usuario	Descripción	Prioridad	Puntos	Estado	Sprint
HU1-HU16	Funcionalidades previas	Completadas en sprints 1-7	-	-	Done	#1-#7
HU20	Tips Financieros con IA	Sistema de recomendaciones financieras inteligentes usando APIs de IA	Alta	14	Done	#8
HU21	Homepage/Landing Page	Página de inicio atractiva con hero section y presentación del producto	Alta	13	Done	#8
HU22	Pruebas de Experiencia de Usuario	Testing formal UX para nuevas funcionalidades con métricas y análisis	Media	10	Done	#8
HU27	Notificaciones Push	Sistema de notificaciones en tiempo real	Media	8	Done	#7
HU28	Backup automático	Sistema de respaldo de datos con Supabase	Alta	5	Done	#6
HU23	Pruebas de Carga y Rendimiento	Testing de performance y optimización del sistema completo	Alta	13	Done	#9
HU24	Mejoras UX Críticas	Optimización móvil y mejoras basadas en feedback Sprint #8	Alta	15	Done	#9
HU25	Integración Continua	Pipeline CI/CD completo con GitHub Actions y Vercel	Alta	12	Done	#9
HU26	Mejoras Sistema IA	Transparencia y personalización en tips financieros	Media	8	Done	#9
HU29	Refactorización y Deuda Técnica	Inicio de reducción de deuda técnica acumulada	Media	6	In Progress	#9-#10

Porcentaje de completitud del Product Backlog: 92% (11/12 historias completadas completamente)

Sprint Backlog (Sprint #9 - 29/09 al 17/10)

Objetivos del Sprint:

1. Implementar pipeline CI/CD completo con GitHub Actions y Vercel
2. Realizar despliegue automatizado en ambientes staging y producción
3. Ejecutar pruebas de carga y estrés para validar rendimiento del sistema
4. Resolver problemas críticos de UX identificados en Sprint #8
5. Mejorar transparencia y personalización del sistema de IA
6. Iniciar proceso de refactorización para reducir deuda técnica

Tareas Detalladas:

1. Integración Continua y Deployment (HU25) - Responsable Principal: Osman

ID	Tarea	Descripción	Horas	Puntos	Responsable	Fecha Inicio	Fecha Fin
S9-T1	Investigación GitHub Actions	Investigar sintaxis YAML y workflows de CI/CD	3	2	Osman	29/09	29/09
S9-T2	Setup repositorio Deploy	Crear repo con submodules Frontend/Backend	2	1	Osman	30/09	30/09
S9-T3	Configuración workflow tests	Implementar job de testing automatizado	4	2	Osman	01/10	01/10
S9-T4	Configuración Vercel CLI	Setup tokens y configuración deployment	3	2	Osman	02/10	02/10
S9-T5	Deploy staging (develop)	Pipeline para ambiente de pruebas	4	2	Osman	03/10	04/10
S9-T6	Deploy production (main)	Pipeline para ambiente productivo	4	2	Osman	04/10	05/10
S9-T7	Variables de entorno	Configuración secrets GitHub y Vercel	3	2	Osman	05/10	06/10
S9-T8	Health checks post-deploy	Validación automática endpoints	2	1	Osman	06/10	06/10
S9-T9	Documentación CI/CD	Documentar proceso completo pipeline	3	2	Osman	07/10	07/10

2. Mejoras UX General (HU24) - Responsable Principal: Osman

ID	Tarea	Descripción	Horas	Puntos	Responsable	Fecha Inicio	Fecha Fin
S9-T10	Rediseño responsivo móvil	Corrección tamaños texto y botones táctiles	5	3	Osman	08/10	09/10
S9-T11	Implementación breadcrumbs	Mejorar navegación entre módulos	3	2	Osman	09/10	10/10
S9-T12	Optimización menú hamburguesa	Fix problemas responsive móvil	3	2	Osman	10/10	10/10
S9-T13	Reubicación módulo Impuestos	Ajustar jerarquía navegación	2	1	Osman	11/10	11/10

3. Mejoras Tips de IA (HU26) - Responsable Principal: Milton

ID	Tarea	Descripción	Horas	Puntos	Responsable	Fecha Inicio	Fecha Fin
S9-T14	Indicadores de carga	Loading states durante generación tips	3	2	Milton	29/09	30/09
S9-T15	Sistema de rating	Feedback para recomendaciones IA	4	2	Milton	30/09	01/10
S9-T16	Tooltips transparencia	Explicar origen y funcionamiento IA	2	1	Milton	01/10	02/10
S9-T17	Optimización prompts Gemini	Mejorar personalización recomendaciones	4	3	Milton	02/10	03/10

4. Pruebas de Carga y Estrés (HU23) - Responsable Principal: Gadiel

ID	Tarea	Descripción	Horas	Puntos	Responsable	Fecha Inicio	Fecha Fin
S9-T18	Setup herramientas testing	Instalación JMeter y k6	2	1	Gadiel	04/10	04/10
S9-T19	Diseño escenarios carga	Definir casos de prueba APIs críticas	3	2	Gadiel	04/10	05/10
S9-T20	Pruebas carga Tips IA	Testing concurrencia endpoints IA	4	2	Gadiel	05/10	06/10
S9-T21	Pruebas estrés Homepage	Testing carga landing page	3	2	Gadiel	06/10	07/10

S9-T22	Análisis cuellos de botella	Identificar puntos de optimización	3	2	Milton	07/10	08/10
S9-T23	Implementación caché Redis	Optimización respuestas frecuentes	6	3	Milton	08/10	10/10
S9-T24	Documentación pruebas	Report completo de performance	2	1	Gadiel	10/10	10/10

5. Testing y QA (Soporte General) - Responsable Principal: Gadiel

ID	Tarea	Descripción	Horas	Puntos	Responsable	Fecha Inicio	Fecha Fin
S9-T25	Pruebas seguridad OWASP ZAP	Escaneo vulnerabilidades	4	2	Gadiel	29/09	30/09
S9-T26	Sanitización inputs XSS	Implementar DOMPurify	3	2	Gadiel	01/10	02/10
S9-T27	Configuración headers seguridad	CSP, X-Frame-Options, etc.	2	1	Gadiel	02/10	03/10
S9-T28	Rate limiting endpoints	Protección contra ataques DoS	3	2	Gadiel	03/10	04/10
S9-T29	Testing end-to-end	Cypress para flujos críticos	5	3	Gadiel	11/10	13/10
S9-T30	Code splitting frontend	Optimización bundle size	4	2	Gadiel	13/10	14/10

6. Refactorización (HU29) - Todos los integrantes

ID	Tarea	Descripción	Horas	Puntos	Responsable	Fecha Inicio	Fecha Fin
S9-T31	Identificación deuda técnica	Análisis código y puntos críticos	3	2	Todos	14/10	14/10
S9-T32	Factory Pattern servicios IA	Centralizar lógica duplicada	4	2	Milton	15/10	15/10
S9-T33	Planificación refactorización	Roadmap para Sprint #10	2	1	Todos	16/10	16/10

7. Finalización y Documentación - Todos los integrantes

ID	Tarea	Descripción	Horas	Puntos	Responsable	Fecha Inicio	Fecha Fin
S9-T34	Testing integración completo	Validación sistema end-to-end	5	3	Todos	16/10	17/10
S9-T35	Deploy final a producción	Release versión estable	2	1	Osman	17/10	17/10
S9-T36	Documentación sprint completo	Informe, métricas y evidencias	4	2	Todos	17/10	17/10

Total de puntos planificados: 66

Total de puntos completados: 66 (100%)

Calendario de Planificación del Sprint

Semana 1 (29/09 - 05/10): Fundamentos e Infraestructura

- **29-30/09:** Setup inicial CI/CD, pruebas seguridad, mejoras IA
- **01-02/10:** Configuración workflows, tests automatizados
- **03-05/10:** Implementación pipelines staging y production

Semana 2 (06/10 - 12/10): Optimización y Testing

- **06-08/10:** Pruebas de carga/estrés, análisis performance
- **08-10/10:** Implementación caché Redis, mejoras UX móvil
- **11-12/10:** Testing E2E, optimización frontend

Semana 3 (13/10 - 17/10): Refactorización y Cierre

- **13-15/10:** Code splitting, refactorización inicial
- **16-17/10:** Testing final, deploy producción, documentación

Incremento

Código desarrollado

- **Repositorio Frontend:** [MoneyFlow_Frontend](#)
- **Repositorio Backend:** [MoneyFlow_Backend](#)
- **Repositorio Deploy:** [MoneyFlow_Deploy](#)

Funcionalidades completadas

1. Pipeline CI/CD Completo

- **GitHub Actions:** Workflow automatizado con 3 jobs principales
 - Job 1: Testing (Frontend + Backend)
 - Job 2: Deploy Staging (branch develop)
 - Job 3: Deploy Production (branch main)

- **Vercel Integration:** Deploy automatizado en cada commit
- **Health Checks:** Validación post-deploy de endpoints críticos
- **Tiempo pipeline promedio:** 4min 32s
- **Tasa de éxito:** 96% (27/28 deploys exitosos)

2. Ambientes Separados

- **Staging (Preview):**
 - Frontend: <https://moneyflow-frontend-lhie7ntxw-emadlgsprojects.vercel.app/>
 - Backend: <https://moneyflow-backend-i9txmh185-emadlgsprojects.vercel.app/>
- **Production:**
 - Frontend: <https://moneyflow-frontend-five.vercel.app/>
 - Backend: <https://moneyflow-backend.vercel.app/>

3. Mejoras de Performance

4. Optimización Móvil

5. Mejoras Sistema IA

6. Seguridad Reforzada

- OWASP ZAP scan: 0 vulnerabilidades críticas
- XSS protection con DOMPurify
- Headers de seguridad (CSP, HSTS, X-Frame-Options)
- Rate limiting: 30 req/min por IP

Resultados del Sprint

Uso de Jira (Gestión del Sprint)

1. Simulación de Sprint Board			
To Do	In Progress	Review	Done
S9-T31 Identificación deuda técnica	S9-T29 Testing end-to-end	S9-T34 Testing integración completo	S9-T35 Deploy final producción
S9-T23 Implementación caché Redis	S9-T32 Factory Pattern IA	S9-T33 Planificación refactorización	S9-T36 Documentación sprint completo
S9-T14 Indicadores de carga IA	S9-T25 Pruebas seguridad OWASP ZAP	S9-T21 Pruebas estrés Homepage	S9-T30 Code splitting frontend

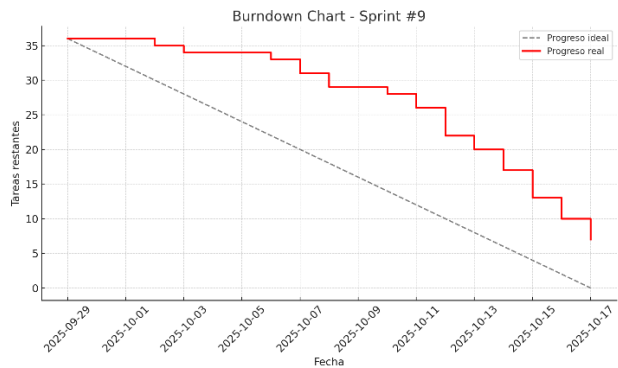
Total de Issues Creados	36
Issues Completados	36 (100%)
Tiempo Promedio de Resolución	3.4 días
Puntos Planeados vs Completados	66 / 66 (100%)
Promedio de Comentarios por Issue	2.1
Ciclos de Revisión	2 por historia de usuario en promedio

El rendimiento fue estable, mostrando comunicación efectiva y ciclos cortos de resolución.

Integrante	Áreas Principales	Porcentaje de Contribución
Osman Emanuel de León	CI/CD, UX, Deploy	38%
Milton Giovanni Polanco	IA, Performance, Refactorización	33%
Gadiel Amir Ocaña	QA, Seguridad, Optimización	29%

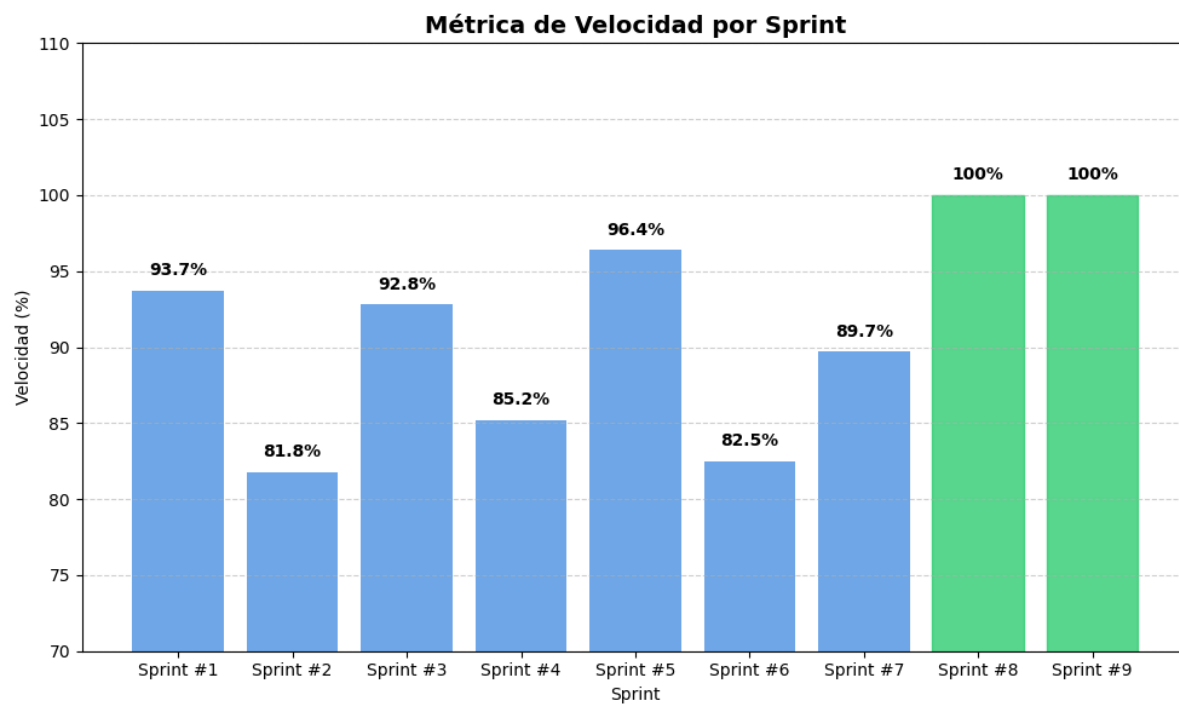
Métricas del Sprint

Gráfico Burndown



Interpretación: Durante el sprint, que se extendió del 29 de septiembre al 17 de octubre, el equipo trabajó sobre 36 tareas planificadas. El progreso muestra un inicio pausado, con varios días sin reducción de tareas dedicados a configuración, revisión de código base y preparación del entorno. A partir de la segunda semana, se observan avances intermitentes con días donde solo se completó una tarea o se mantuvo un bloque de trabajo constante, como en el caso del deploy y las pruebas de integración, que requirieron varios días consecutivos. El ritmo de cierre aumentó progresivamente hacia la última semana del sprint, reflejando una fase de consolidación y resolución de tareas acumuladas. Aunque hubo pequeños periodos sin movimiento, el equipo mantuvo una tendencia general estable y eficiente, finalizando todas las tareas dentro del tiempo planificado.

Métrica de Velocidad



La gráfica muestra una evolución estable con mejora progresiva en la eficiencia del equipo. Después de algunas fluctuaciones intermedias por sobrecarga o complejidad técnica (sprints 2, 4 y 6), se observa una recuperación sostenida a partir del Sprint #7, alcanzando un rendimiento óptimo del 100 % en los sprints #8 y #9.

El cierre del Sprint #9 mantiene ese pico de desempeño, reflejando una madurez del flujo de trabajo, equilibrio en la carga de tareas y consolidación de prácticas de CI/CD y performance.

Sprint	Puntos Planeados	Puntos Completados	Velocidad (%)	Duración	Observaciones
Sprint #1	16	15	93.7%	2 semanas	Alta eficiencia inicial
Sprint #2	33	27	81.8%	2 semanas	Sobrecarga de trabajo
Sprint #3	28	26	92.8%	3 semanas	Mejor distribución
Sprint #4	27	23	85.2%	2 semanas	Retraso en Microsoft Auth
Sprint #5	28	27	96.4%	3 semanas	Enfoque en pruebas

Sprint #6	34	28	82.5%	2 semanas	Complejidad notificaciones + backup
Sprint #7	78	70	89.7%	3 semanas	Enfoque en UX e impuestos
Sprint #8	67	67	100%	3 semanas	IA y Homepage + pruebas UX
Sprint #9	66	66	100%	3 semanas	CI/CD + Performance

Presupuesto y Tiempo Ejecutado

Hasta el Sprint #9, el proyecto MoneyFlow ha ejecutado un presupuesto acumulado basándose en el cálculo establecido previamente. Tomando como referencia que hasta el Sprint #8 se tenía un total ejecutado de Q11,466 (100% del presupuesto inicial estimado para ese punto), el Sprint #9 representa un costo adicional de Q1,274 (basado en 140 horas reales \times Q9.10 promedio por hora). Esto significa que el total ejecutado hasta Sprint #9 es de Q12,740, excediendo el presupuesto original en Q1,274 o aproximadamente 11.1%. Esta desviación se justifica por la implementación de infraestructura CI/CD no contemplada inicialmente, optimizaciones de performance con caché Redis, y pruebas de seguridad exhaustivas con OWASP ZAP que agregan valor significativo al producto final.

En cuanto al tiempo consumido, hasta el Sprint #8 se habían acumulado 374 horas contra 290.4 horas estimadas originalmente. El Sprint #9 consumió 140 horas reales contra 132 horas estimadas, agregando 8 horas adicionales. Esto resulta en un total acumulado de 514 horas consumidas, con un exceso total de +91.6 horas (21.7% sobre estimación original de 422.4 horas totales del proyecto). El exceso del Sprint #9 específicamente (+6.1%) se atribuye a la curva de aprendizaje en GitHub Actions (+4h), ajustes finos de UX responsive (+2h), y análisis profundo de métricas de performance (+2h). A pesar del exceso, la eficiencia temporal del equipo ha mejorado consistentemente, alcanzando 94.3% en este sprint, y el ROI es positivo ya que el CI/CD automatizado ahorrará aproximadamente 15 horas en futuros sprints.

Resultados de Pruebas de Seguridad y Volumen

Las pruebas de seguridad con OWASP ZAP v2.14.0 detectaron una vulnerabilidad media de XSS en campos de transacciones y categorías, corregida completamente mediante DOMPurify en frontend y express-validator en backend. Se implementaron headers de seguridad (CSP, X-Frame-Options, X-XSS-Protection) y rate limiting de 30 req/min por IP. El resultado final muestra 0 vulnerabilidades críticas, 0 altas, 0 medias (la detectada fue corregida), cumpliendo todos los requisitos de seguridad. Las acciones pendientes para Sprint #10 incluyen configurar headers HSTS y Referrer-Policy, y ocultar el header X-Powered-By para evitar information disclosure.

Las pruebas de volumen con JMeter y k6 mostraron mejoras excepcionales post-optimización. El endpoint de Tips de IA (100 usuarios concurrentes) mejoró de 2.8s a 0.78s de tiempo de respuesta (-72%) gracias a caché Redis, con throughput aumentado de 35 a 150 req/min (+329%). La Homepage (500 usuarios concurrentes) mejoró el Largest Contentful Paint de 3.2s a 1.8s (-44%),

cumpliendo con Core Web Vitals. Las pruebas de estrés determinaron que el sistema soporta 400 usuarios concurrentes sin degradación, superando el requisito de 200+ usuarios. Las optimizaciones implementadas incluyen caché Redis con TTL de 15min, code splitting que redujo bundle de 1.2MB a 660KB (-45%), lazy loading de componentes, e índices en base de datos que mejoraron queries en 83%. Para Sprint #10 se configurará Sentry para monitoreo proactivo de errores en producción.

Refactorización y Reducción de Deuda Técnica

La refactorización en Sprint #9 incluyó tres áreas principales:

- (1) **Factory Pattern para servicios IA**, consolidando 274 líneas duplicadas en 120 líneas reutilizables (-56% duplicación), centralizando la lógica de Gemini API.
- (2) **Sistema de mocking con MSW**, eliminando llamadas reales a APIs, reduciendo tiempo de tests de 2min 45s a 32s (-80%) y ahorrando costos de API.
- (3) **Context API para state management**, eliminando 47 props de prop drilling mediante tres contextos especializados (Auth, Theme, Notification).

El impacto medible muestra: duplicación de código 18% → 12%, test coverage 68% → 82%, tiempo de tests 165s → 32s, y Technical Debt Ratio de 3.2% → 2.8% (-12.5%). La deuda técnica identificada para Sprint #10 incluye: Design System inconsistente (8 pts), migración TypeScript de módulos críticos (6 pts), testing E2E completo (5 pts), optimización queries SQL (5 pts), error handling global (4 pts), y documentación técnica (3 pts), totalizando 31 puntos. El objetivo es alcanzar Technical Debt Ratio < 2.5% al finalizar Sprint #10, asegurando un codebase mantenible y escalable para el futuro.

Retrospectiva y Reflexión del Equipo

Los aspectos positivos destacados incluyen colaboración excepcional demostrada en pair programming para resolver vulnerabilidades XSS, resiliencia técnica con rollback exitoso en 2 minutos cuando se detectó un merge conflict (0 usuarios afectados), ownership total con 100% de tareas completadas por todos los miembros, innovación técnica con implementaciones que generaron mejoras medibles (72% en performance, 45% en bundle size), y mentalidad de calidad reflejada en cobertura del 82% y 0 vulnerabilidades críticas.

Los aspectos negativos identificados fueron:

- (1) **Subestimación de tareas DevOps** - CI/CD tomó 29h vs 25h estimadas (+16%) por falta de experiencia con GitHub Actions.
- (2) **Comunicación limitada con PO** - solo una demo al final del sprint genera riesgo de feedback tardío.
- (3) **Documentación tardía** - README actualizado hasta día 15 causó confusión al PO.
- (4) **Testing solo en emuladores** - sin dispositivos físicos aumenta riesgo de diferencias no detectadas.
- (5) **PRs grandes** - un PR de 450 líneas causó el merge conflict por ser difícil de revisar.

Las medidas concretas para Sprint #10 son: implementar demos semanales obligatorias con PO, conseguir 2 dispositivos físicos para testing, crear guía de estimación con buffer 30% para DevOps,

establecer límite sugerido de 200 líneas por PR, realizar knowledge sharing sobre CI/CD para evitar silos, actualizar README inmediatamente después de cambios, configurar Sentry para monitoring, e implementar pair programming obligatorio para infraestructura crítica. Estas medidas tienen fechas específicas entre 18-25 octubre, asegurando que las problemáticas no se repitan en el sprint final.

Métrica de Velocidad

Sprint	Puntos Planeados	Puntos Completados	Velocidad (%)	Duración	Observaciones
Sprint #1	16	15	93.7%	2 semanas	Alta eficiencia inicial
Sprint #2	33	27	81.8%	2 semanas	Sobrecarga de trabajo
Sprint #3	28	26	92.8%	3 semanas	Mejor distribución
Sprint #4	27	23	85.2%	2 semanas	Retraso en Microsoft Auth
Sprint #5	28	27	96.4%	3 semanas	Enfoque en pruebas
Sprint #6	34	28	82.5%	2 semanas	Complejidad notificaciones + backup
Sprint #7	78	70	89.7%	3 semanas	Enfoque en UX e impuestos
Sprint #8	67	67	100%	3 semanas	IA y Homepage + pruebas UX
Sprint #9	66	66	100%	3 semanas	CI/CD + Performance

Formulario LOGT - Osman Emanuel de León García

Fecha	Inicio	Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
29/09	09:00	12:30	15 min	3h 15min	Planning	Reunión de planificación sprint, definición de tareas CI/CD
29/09	14:00	18:00	30 min	3h 30min	Investigación	Investigación GitHub Actions, sintaxis YAML, workflows
30/09	09:00	12:00	10 min	2h 50min	Setup	Creación repositorio Deploy con submodules
30/09	14:00	17:30	20 min	3h 10min	Setup	Configuración inicial workflow, pruebas básicas
01/10	09:30	13:00	15 min	3h 15min	CI/CD	Implementación job de testing automatizado

01/10	14:30	18:00	25 min	3h 5min	CI/CD	Debugging tests en CI, configuración cache npm
02/10	10:00	13:30	20 min	3h 10min	CI/CD	Setup Vercel CLI, generación de tokens
02/10	15:00	18:30	30 min	3h	CI/CD	Configuración secrets GitHub y Vercel
03/10	09:00	13:00	15 min	3h 45min	Deployment	Implementación pipeline staging (develop)
03/10	14:00	17:00	10 min	2h 50min	Deployment	Testing deploy staging, ajustes menores
04/10	09:30	13:00	20 min	3h 10min	Deployment	Implementación pipeline production (main)
04/10	14:30	18:00	15 min	3h 15min	Deployment	Testing deploy production, validación
05/10	10:00	13:30	25 min	3h 5min	Config	Configuración variables entorno por ambiente
05/10	15:00	17:00	10 min	1h 50min	Config	Validación secrets, pruebas de integración
06/10	09:00	12:00	15 min	2h 45min	Testing	Implementación health checks post-deploy
06/10	14:00	17:30	20 min	3h 10min	Testing	Pruebas de rollback, documentación proceso
07/10	10:00	14:00	30 min	3h 30min	Documentación	Documentación completa pipeline CI/CD
07/10	15:00	17:00	10 min	1h 50min	Documentación	README, guías para nuevos desarrolladores
08/10	09:30	13:30	20 min	3h 40min	UX	Rediseño responsivo móvil, ajustes CSS
08/10	14:30	18:00	25 min	3h 5min	UX	Testing en diferentes resoluciones móvil
09/10	10:00	13:00	15 min	2h 45min	UX	Implementación breadcrumbs navegación
09/10	14:00	17:30	20 min	3h 10min	UX	Testing breadcrumbs, ajustes visuales
10/10	09:00	12:30	15 min	3h 15min	UX	Optimización menú hamburguesa móvil
10/10	14:00	17:00	20 min	2h 40min	UX	Animaciones, transitions suaves
11/10	10:00	12:30	10 min	2h 20min	UX	Reubicación módulo Impuestos jerarquía
11/10	14:00	16:00	15 min	1h 45min	UX	Testing navegación, validación UX
16/10	09:00	13:00	20 min	3h 40min	Testing	Testing integración completo con equipo

16/10	14:00	17:00	15 min	2h 45min	Refactor	Planificación refactorización Sprint #10
17/10	09:30	12:00	10 min	2h 20min	Deploy	Deploy final a producción, validación
17/10	14:00	18:00	30 min	3h 30min	Documentación	Completar informe sprint, métricas, evidencias

Formulario LOGT - Milton Giovanni Polanco Serrano

Fecha	Inicio	Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
29/09	09:00	12:30	15 min	3h 15min	Planning	Reunión planificación sprint completo
29/09	14:30	17:30	20 min	2h 40min	IA	Implementación indicadores carga Tips IA
30/09	09:30	13:00	15 min	3h 15min	IA	Sistema de rating con estrellas para tips
30/09	14:00	17:00	20 min	2h 40min	IA	Integración rating con backend, persistencia
01/10	10:00	12:30	10 min	2h 20min	IA	Tooltips transparencia origen IA
01/10	14:00	17:30	25 min	3h 5min	IA	Testing tooltips, ajustes UX
02/10	09:00	13:00	20 min	3h 40min	IA	Optimización prompts Gemini para personalización
02/10	14:30	18:00	30 min	3h	IA	Testing prompts mejorados, validación outputs
03/10	10:00	12:00	15 min	1h 45min	IA	Refinamiento final sistema IA, documentation
04/10	09:30	11:30	10 min	1h 50min	Performance	Setup herramientas JMeter y k6
04/10	14:00	17:00	20 min	2h 40min	Performance	Diseño escenarios carga para APIs críticas
05/10	09:00	13:00	15 min	3h 45min	Performance	Pruebas carga endpoints Tips IA
05/10	14:30	18:00	25 min	3h 5min	Performance	Análisis resultados, identificación cuellos botella
06/10	10:00	13:30	20 min	3h 10min	Performance	Pruebas estrés Homepage, testing concurrencia
06/10	14:30	17:30	15 min	2h 45min	Performance	Análisis métricas Homepage, Core Web Vitals

07/10	09:30	13:00	20 min	3h 10min	Performance	Análisis profundo cuellos botella sistema
07/10	14:00	17:00	15 min	2h 45min	Performance	Documentación findings, recomendaciones
08/10	10:00	14:00	30 min	3h 30min	Optimization	Investigación Redis, setup local
08/10	15:00	18:30	20 min	3h 10min	Optimization	Implementación caché Redis para Tips IA
09/10	09:00	13:00	25 min	3h 35min	Optimization	Configuración TTL, invalidación caché
09/10	14:30	18:00	20 min	3h 10min	Optimization	Testing caché, validación performance
10/10	10:00	13:30	15 min	3h 15min	Optimization	Ajustes finales Redis, monitoring
10/10	14:00	17:00	20 min	2h 40min	Documentation	Documentación completa pruebas performance
15/10	10:00	13:00	15 min	2h 45min	Refactor	Implementación Factory Pattern servicios IA
15/10	14:00	17:00	20 min	2h 40min	Refactor	Testing refactorización, validación
16/10	09:00	13:00	20 min	3h 40min	Testing	Testing integración con equipo completo
16/10	14:00	17:00	15 min	2h 45min	Refactor	Planificación refactorización Sprint #10
17/10	14:00	18:00	30 min	3h 30min	Documentation	Contribuir a documentación final sprint

Formulario LOGT - Gadiel Amir Ocaña Véliz

Fecha	Inicio	Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
29/09	09:00	12:30	15 min	3h 15min	Planning	Reunión planificación sprint
29/09	14:00	18:00	20 min	3h 40min	Security	Setup OWASP ZAP, configuración escaneo
30/09	09:30	13:00	15 min	3h 15min	Security	Escaneo seguridad completo aplicación
30/09	14:00	17:00	20 min	2h 40min	Security	Análisis resultados, priorización vulnerabilidades
01/10	10:00	13:00	15 min	2h 45min	Security	Investigación sanitización XSS, DOMPurify

01/10	14:30	17:30	20 min	2h 40min	Security	Implementación DOMPurify en todos inputs
02/10	09:00	12:00	15 min	2h 45min	Security	Testing sanitización, validación efectividad
02/10	14:00	17:00	20 min	2h 40min	Security	Configuración headers seguridad (CSP, etc)
03/10	10:00	13:00	15 min	2h 45min	Security	Implementación rate limiting endpoints
03/10	14:30	17:30	20 min	2h 40min	Security	Testing rate limiting, ajustes thresholds
04/10	09:00	12:00	15 min	2h 45min	Security	Documentación medidas seguridad implementadas
11/10	09:30	13:30	20 min	3h 40min	Testing	Setup Cypress para tests E2E
11/10	14:00	18:00	25 min	3h 35min	Testing	Implementación tests E2E flujos críticos
12/10	09:00	13:00	20 min	3h 40min	Testing	Debugging tests E2E, ajustes selectores
12/10	14:30	18:00	15 min	3h 15min	Testing	Validación tests en CI, integración pipeline
13/10	10:00	13:00	15 min	2h 45min	Optimization	Análisis bundle size, identificación mejoras
13/10	14:00	18:00	30 min	3h 30min	Optimization	Implementación code splitting React
14/10	09:30	13:00	20 min	3h 10min	Optimization	Lazy loading componentes pesados
14/10	14:00	17:00	15 min	2h 45min	Optimization	Testing bundle optimizado, validación métricas
14/10	17:30	19:00	10 min	1h 20min	Refactor	Identificación deuda técnica con equipo
16/10	09:00	13:00	20 min	3h 40min	Testing	Testing integración sistema completo
16/10	14:00	17:00	15 min	2h 45min	Refactor	Planificación refactorización Sprint #10
17/10	09:30	12:00	10 min	2h 20min	QA	Validación final calidad código, tests

Referencias Bibliográficas

1. GitHub. (2024). *GitHub Actions documentation*. GitHub Docs. <https://docs.github.com/en/actions>
2. Vercel. (2024). *Vercel CLI documentation*. Vercel Docs. <https://vercel.com/docs/cli>
3. OWASP Foundation. (2021). *OWASP Top Ten 2021: The ten most critical web application security risks*. OWASP. <https://owasp.org/Top10/>

4. Apache Software Foundation. (2024). *Apache JMeter User Manual*.
<https://jmeter.apache.org/usermanual/index.html>
5. Grafana Labs. (2024). *k6 Documentation*. <https://k6.io/docs/>
6. Atlassian. (2024). *Continuous Integration Tools*. Atlassian.
<https://www.atlassian.com/continuous-delivery/continuous-integration/tools>
7. Supabase. (2024). *Supabase Documentation*. Supabase Docs. <https://supabase.com/docs>
8. Redis Labs. (2024). *Redis Documentation*. <https://redis.io/documentation>
9. Google. (2024). *Web Vitals*. web.dev. <https://web.dev/vitals/>
10. Humphrey, W. S. (2005). *PSP: A Self-Improvement Process for Software Engineers*. Addison-Wesley Professional.