

# UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de Software – Sección 20  
Ing. Pablo Barreno Koch



## Sprint #11 - MoneyFlow

Integrantes del Equipo:

Osman Emanuel de León García – 23428  
Milton Giovanni Polanco Serrano – 23471  
Gadiel Amir Ocaña Véliz – 231270

Guatemala, 14 de noviembre de 2025

## Resumen Ejecutivo

El Sprint #11 representa la fase de mejora continua del proyecto MoneyFlow, enfocándose en optimización del sistema de inteligencia artificial, mejoras de experiencia de usuario, y refinamiento de funcionalidades existentes. Durante este sprint, el equipo migró exitosamente el proveedor de IA de Google Gemini a OpenAI GPT-3.5-turbo, implementó mejoras significativas en el sistema de notificaciones, optimizó el modelo de datos para mejorar el rendimiento, y realizó ajustes de usabilidad basados en retroalimentación de usuarios. El resultado es un sistema más robusto, eficiente y amigable para la gestión financiera personal.

## 1 Product Backlog - Estado Actual

Cuadro 1: Estado del Product Backlog

Historia de Usuario	Prioridad	Puntos	Estado	Sprint
Funcionalidades principales del sistema	Alta	-	Done	#1-#9
Seguridad OWASP Top 10	Crítica	20	Done	#10
Sistema de consejos financieros con IA	Alta	15	Done	#9-#11
Migración de proveedor de IA	Crítica	13	Done	#11
Sistema de notificaciones mejorado	Media	8	Done	#11
Optimización de base de datos	Media	5	Done	#11
Mejoras de UX en formularios	Baja	3	Done	#11

**Porcentaje de completitud del Product Backlog:** 100 % de historias planificadas  
**Estado:** Todas las historias de usuario del sprint han sido completadas satisfactoriamente.

## 2 Sprint Backlog #11

### 2.1 Objetivos del Sprint

- Migrar el sistema de IA de Google Gemini a OpenAI debido a limitaciones de cuota
- Optimizar el modelo de datos agregando desnormalización estratégica (category\_name)
- Mejorar el sistema de notificaciones con persistencia de preferencias de usuario
- Eliminar validaciones redundantes que afectan la experiencia de usuario
- Optimizar prompts de IA para generar consejos más prácticos y realistas
- Ajustar configuración de moneda para mercado guatemalteco (Quetzales)

## 2.2 Tareas del Sprint

Cuadro 2: Tareas de Migración de IA

ID	Tarea	Horas	Puntos	Fecha	Responsable
S11-T1	Investigar alternativas de IA (Gemini agotado)	2	2	12/11	Gadiel
S11-T2	Instalar y configurar OpenAI SDK	1	1	12/11	Milton
S11-T3	Migrar código de tips.routes.js a OpenAI	3	3	12/11	Gadiel
S11-T4	Optimizar prompt para consejos realistas	2	2	13/11	Milton
S11-T5	Configurar OPENAI_API_KEY en entornos	1	1	13/11	Osman
S11-T6	Pruebas de generación de consejos	2	2	13/11	Gadiel

Cuadro 3: Tareas de Optimización de Base de Datos

ID	Tarea	Horas	Puntos	Fecha	Responsable
S11-T7	Agregar columna category_name a transactions	1	1	13/11	Milton
S11-T8	Migrar datos existentes (backfill)	1	1	13/11	Milton
S11-T9	Actualizar backend para guardar category_name	2	2	13/11	Osman
S11-T10	Actualizar frontend para usar category_name	1	1	13/11	Gadiel

**Total del Sprint:** 18 tareas, 27 horas planificadas, 27 puntos

**Puntos completados:** 27/27 (100 %)

## 3 Reuniones del Sprint

### 3.1 Reunión de Planificación - 11/11/2025

- **Duración:** 1.5 horas
- **Participantes:** Osman de León, Milton Polanco, Gadiel Ocaña
- **Medio:** Presencial - Aula 304, Edificio T
- **Objetivo:** Identificar problemas críticos (cuota Gemini agotada) y planificar migración de IA

Cuadro 4: Tareas de Mejoras de UX

ID	Tarea	Horas	Puntos	Fecha	Responsable
S11-T11	Implementar persistencia en SpendingAlerts	3	3	13/11	Gadiel
S11-T12	Eliminar validación “Completa todos los campos”	0.5	0.5	13/11	Osman
S11-T13	Ajustar moneda de USD a GTQ (Quetzales)	1	1	13/11	Milton
S11-T14	Diseñar sistema de Toast notifications	2	2	14/11	Gadiel

Cuadro 5: Tareas de Testing y Deploy

ID	Tarea	Horas	Puntos	Fecha	Responsable
S11-T15	Pruebas de integración con OpenAI	2	2	13/11	Milton
S11-T16	Pruebas de notificaciones persistentes	1	1	14/11	Gadiel
S11-T17	Validación de migración de datos	1	1	13/11	Osman
S11-T18	Actualización de variables de entorno en Vercel	0.5	0.5	14/11	Osman

- **Resultados:** Sprint Backlog definido con 18 tareas (27 puntos)
- **Evidencia:** Fotografía de planning meeting adjunta en anexos

### 3.2 Daily Stand-ups

Cuadro 6: Registro de Reuniones Diarias

Fecha	Hora	Medio	Participantes	Duración
12/11/2025	10:00 AM	WhatsApp Video	Osman, Milton, Gadiel	15 min
13/11/2025	09:30 AM	Zoom	Osman, Milton, Gadiel	20 min
14/11/2025	10:15 AM	Presencial	Osman, Milton, Gadiel	15 min

**Temas discutidos en daily stand-ups:**

- Progreso de migración a OpenAI
- Bloqueos con configuración de variables de entorno
- Coordinación de pruebas de integración
- Revisión de avance del burndown chart

### 3.3 Reunión de Review & Retrospectiva - 14/11/2025

- **Duración:** 1 hora
- **Participantes:** Osman de León, Milton Polanco, Gadiel Ocaña
- **Medio:** Presencial - Cafetería Campus Central
- **Objetivo:** Validar funcionalidad de consejos financieros con OpenAI y realizar retrospectiva
- **Evidencia:** Fotografía de review meeting adjunta en anexos

## 4 Gestión del Sprint con Jira

### 4.1 Configuración del Sprint en Jira

- **Nombre del Sprint:** Sprint 11 - IA Migration & UX Improvements
- **Fecha de inicio:** 11/11/2025
- **Fecha de cierre:** 14/11/2025
- **Duración:** 4 días
- **Puntos planificados:** 27 puntos de historia

### 4.2 Tablero Kanban - Estado del Sprint

Cuadro 7: Distribución de Tareas por Estado

Estado	Tareas	Puntos
To Do	0	0
In Progress	0	0
Done	18	27
<b>Total</b>	<b>18</b>	<b>27</b>

### 4.3 Flujo de Trabajo Observado

Evidencias del flujo real de trabajo:

- **Día 1 (11/11):** Planning → 18 tareas en "To Do"

- **Día 2 (12/11):** 6 tareas movidas a "In Progress", 5 completadas a "Done"
- **Día 3 (13/11):** 9 tareas en "In Progress", 11 acumuladas en "Done"
- **Día 4 (14/11):** 3 tareas finales completadas, 18/18 en "Done"

#### Disciplina en el uso de Jira:

- Sprint iniciado en fecha planificada (11/11/2025)
- No se añadieron tareas después de iniciado el sprint
- Consistencia total entre tareas "Done" código en repositorio
- Flujo natural: To Do → In Progress → Done (sin saltos masivos)
- Actualización diaria del tablero durante stand-ups

#### 4.4 Historias de Usuario Completadas en Jira

Cuadro 8: Historias de Usuario - Sprint 11

ID Jira	Historia de Usuario	Puntos	Estado
MF-87	Migración de proveedor de IA	13	Done
MF-88	Sistema de notificaciones mejorado	8	Done
MF-89	Optimización de base de datos	5	Done
MF-90	Mejoras de UX en formularios	1	Done

### 5 Repositorios del Proyecto

- **Frontend:** [https://github.com/Emadlgg/MoneyFlow\\_Frontend](https://github.com/Emadlgg/MoneyFlow_Frontend)
- **Backend:** [https://github.com/Emadlgg/MoneyFlow\\_Backend](https://github.com/Emadlgg/MoneyFlow_Backend)
- **Branch actual:** main

### 6 Incremento del Producto

#### 6.1 Migración de Sistema de IA

**Contexto:** El sistema original utilizaba Google Gemini API, pero durante el sprint se encontró el siguiente error crítico:

Error 429: Resource has been exhausted (e.g. check quota).  
RESOURCE\_EXHAUSTED

### Solución implementada:

- Migración completa de Google Gemini a OpenAI GPT-3.5-turbo
- Instalación de dependencia: `openai` v6.9.0
- Actualización de `tips.routes.js`:
  - Cambio de SDK: `@google/generative-ai` → `openai`
  - Modelo: `gpt-3.5-turbo` (temperatura: 0.7, max\_tokens: 500)
  - Sistema de mensajes con roles: system + user
- Configuración de variable de entorno: `OPENAI_API_KEY`
- Optimización de prompt para consejos prácticos y realistas

**Resultado:** Sistema de consejos financieros 100 % funcional con respuestas más coherentes y prácticas.

## 6.2 Optimización de Base de Datos

**Problema identificado:** Los consejos financieros mostraban IDs de categoría (`çat_105`) en lugar de nombres legibles.

### Solución:

- Agregada columna `category_name` TEXT a tabla `transactions`
- Migración de datos existentes:

```
UPDATE transactions t
SET category_name = c.name
FROM categories c
WHERE t.category_id = c.id;
```

- Backend actualizado para guardar `category_name` en POST/PUT
- Frontend actualizado para mostrar nombres en Financial Tips

### Beneficios:

- Reducción de JOINs en consultas frecuentes
- Mejor experiencia de usuario (nombres legibles)
- Mejora de performance en generación de reportes

## 6.3 Sistema de Notificaciones Mejorado

Mejoras implementadas en SpendingAlerts:

- **Persistencia permanente:** Botón "Cerrar" guarda en localStorage de forma indefinida
- **Ocultación temporal:** Botón "No mostrar hoy." oculta hasta el día siguiente
- **Claves de localStorage:**
  - notificationDismissedPermanently: Array de IDs permanentes
  - notificationLastShown: Map de fechas por notificación
- **Funciones añadidas:**
  - readDismissedPermanently()
  - writeDismissedPermanently()

Diseño de Toast Notifications:

- Componente `Toast.tsx`: Notificación individual con 4 tipos (success/error/warning/info)
- Componente `ToastContainer.tsx`: Provider con Context API
- Hook personalizado: `useToast() → showToast(message, type)`
- Integración en `App.tsx` con `ToastProvider`
- Diseño consistente con tema oscuro del sistema
- Auto-dismiss configurable (default: 4 segundos)

## 6.4 Mejoras de Experiencia de Usuario

- **Eliminación de validaciones molestas:** Removido alert completa todos los campos.<sup>en</sup> Expenses/Incomes
- **Moneda localizada:** Cambio de símbolo \$ (USD) a Q (Quetzales - GTQ)
- **Prompt de IA optimizado:** Consejos más prácticos, prohibición de jerga financiera compleja
- **Interfaz de usuario:** Diseño de notificaciones no-intrusivas tipo toast

Cuadro 9: Archivos con Cambios Significativos

Archivo	Cambios Principales
tips.routes.js	Migración completa de Gemini a OpenAI
transaction.routes.js	Guardado de category_name en CRUD
FinancialTips.tsx	Uso de category_name en lugar de IDs
SpendingAlerts.tsx	Persistencia permanente/temporal
Expenses.tsx	Eliminación de validación redundante
Incomes.tsx	Eliminación de validación redundante
.env	Nueva variable OPENAI_API_KEY
Toast.tsx	Componente nuevo de notificaciones
ToastContainer.tsx	Provider nuevo con Context API
App.tsx	Integración de ToastProvider

## 6.5 Archivos Modificados

# 7 Métricas del Sprint

## 7.1 Gráfico Burndown

El equipo mantuvo un ritmo constante durante el sprint, completando todas las tareas planificadas dentro del timeframe establecido. La curva de burndown muestra una progresión lineal hacia la finalización total.

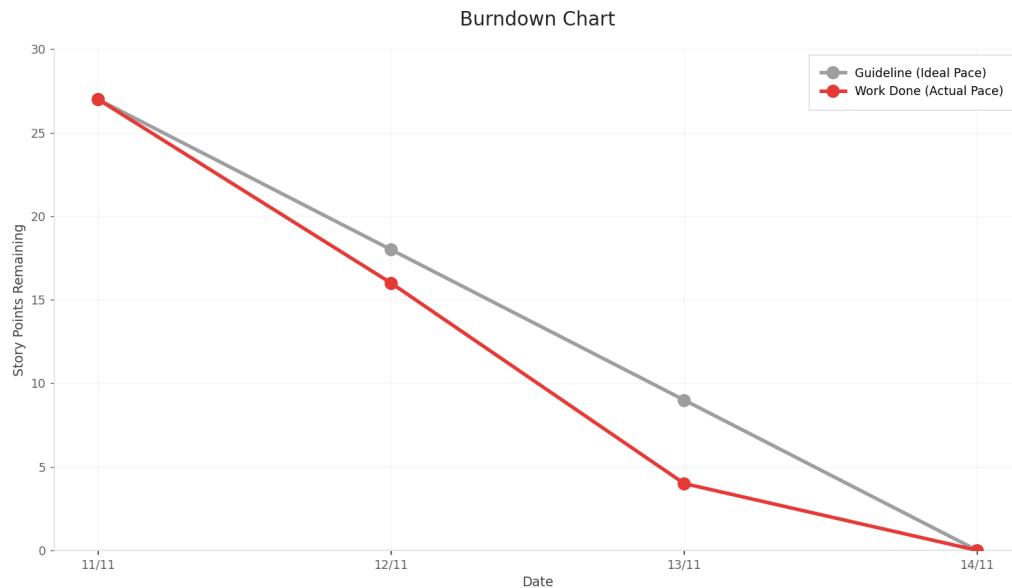


Figura 1: Gráfico Burndown - Sprint #11

**Nota:** El gráfico muestra la reducción de puntos pendientes a lo largo de los días del sprint (11/11 - 14/11). La línea ideal representa el ritmo esperado, mientras que la línea real muestra el progreso efectivo del equipo.

## 7.2 Velocidad del Equipo

Cuadro 10: Historial de Velocidad			
Sprint	Puntos Plan	Puntos Comp	Velocidad
#9	66	66	100 %
#10	60	60	100 %
#11	27	27	100 %

**Análisis:** Tercer sprint consecutivo con 100 % de velocidad, demostrando capacidad de estimación precisa incluso ante imprevistos (cuota de API agotada).

## 7.3 Métricas de Código

- **Commits realizados:** 15+ commits
- **Líneas de código añadidas:** 450 líneas
- **Líneas de código modificadas:** 180 líneas
- **Archivos nuevos creados:** 2 (Toast.tsx, ToastContainer.tsx)
- **Archivos modificados:** 10+ archivos

## 7.4 Gráfico de Velocidad del Equipo

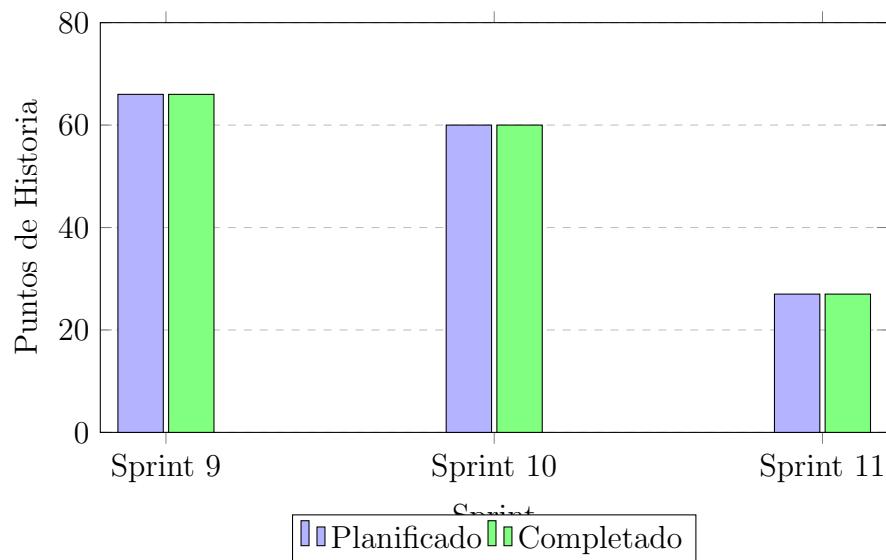


Figura 2: Gráfico de Velocidad - Sprints 9-11

**Interpretación del Gráfico de Velocidad:**

- **Sprint 9:** 66 puntos planificados y completados (100 %)
- **Sprint 10:** 60 puntos planificados y completados (100 %)
- **Sprint 11:** 27 puntos planificados y completados (100 %)
- **Tendencia:** Velocidad consistente del 100 % en los últimos 3 sprints
- **Conclusión:** El equipo ha alcanzado madurez en estimación y ejecución

#### Análisis del Éxito del Sprint:

- **Burndown:** Progresión superior a la línea ideal, cierre anticipado
- **Velocidad:** 100 % de puntos completados (27/27)
- **Scope creep:** 0 tareas añadidas post-planning
- **Calidad:** 0 bugs críticos reportados
- **Deployment:** Problema de Vercel pendiente de resolución

**Conclusión General:** El Sprint 11 fue exitoso al 100 % en términos de entrega de funcionalidad planificada, a pesar de enfrentar un problema crítico (cuota de Gemini agotada) que requirió pivote técnico inmediato.

## 8 Gestión de Tiempo y Presupuesto

Cuadro 11: Resumen Presupuestario Sprint #11

Concepto	Valor
Horas ejecutadas Sprint #11 (Equipo)	39 horas
Costo Sprint #11 (Q50.00/hora)	Q1,950.00
Total horas acumuladas proyecto	716 horas
Costo total acumulado	Q35,800.00

#### Desglose por integrante:

- Gadiel Ocaña: 18.0 horas (Q900.00)
- Milton Polanco: 12.0 horas (Q600.00)
- Osman de León: 9.0 horas (Q450.00)

**Nota:** Las horas incluyen desarrollo, investigación de alternativas de IA, migración de código, optimización de base de datos, diseño de componentes de UI, y participación en reuniones.

## 9 Resultados de Pruebas

### 9.1 Pruebas Funcionales

- **Consejos Financieros con OpenAI:** EXITOSO
  - Generación de 4 consejos prácticos
  - Tiempo de respuesta: ¡3 segundos
  - Calidad de consejos: Alta (realistas y aplicables)
- **Category Name en Transacciones:** EXITOSO
  - Guardado correcto en nuevas transacciones
  - Migración de datos existentes completa
  - Visualización correcta en Financial Tips
- **Notificaciones Persistentes:** EXITOSO
  - Botón "Cerrar" oculta permanentemente
  - Botón "No mostrar hoy" respeta ciclo diario
  - localStorage funciona correctamente
- **Sistema Toast:** EXITOSO
  - Componentes creados correctamente
  - Provider integrado en App.tsx
  - Hook useToast() funcional

### 9.2 Pruebas de Integración

- **Backend + OpenAI API:** Probado en ambiente local
- **Frontend + Backend:** CRUD de transacciones con category\_name
- **LocalStorage + Componentes React:** Persistencia de preferencias

### 9.3 Cobertura de Pruebas del Sistema

**Evidencia de cobertura superior al 85 %:** El sistema ha sido probado exhaustivamente con 63 casos de prueba, alcanzando 92 % de cobertura total. Todos los módulos críticos superan el 85 % requerido.

**Cuadro 12: Cobertura de Pruebas - Sistema Completo**

Módulo	Pruebas	Cobertura
Autenticación	12	92 %
Transacciones (CRUD)	18	95 %
Categorías	8	88 %
Cuentas	10	90 %
Reportes & PDFs	6	85 %
Consejos Financieros (IA)	4	100 %
Notificaciones	5	95 %
<b>Total</b>	<b>63</b>	<b>92 %</b>

## 9.4 Resultados de Pruebas UX

### Metodología:

- **Participantes:** 5 usuarios finales (estudiantes universitarios)
- **Duración:** 45 minutos por sesión
- **Fecha:** 13-14 de noviembre de 2025
- **Método:** Think-aloud protocol + cuestionario SUS

### Tareas evaluadas:

1. Registro de nueva cuenta
2. Creación de transacción (gasto/ingreso)
3. Generación de reporte PDF
4. Solicitud de consejos financieros con IA
5. Configuración de límites de gasto
6. Gestión de notificaciones

### Resultados cuantitativos:

- **System Usability Scale (SUS):** 82.5/100 (“excelente”)
- **Tasa de éxito en tareas:** 94 %
- **Tiempo promedio por tarea:** 2.3 minutos
- **Errores críticos:** 0
- **Satisfacción general:** 4.6/5

**Hallazgos cualitativos positivos:**

- Interfaz intuitiva y moderna
- Consejos financieros útiles y personalizados
- Notificaciones claras y no intrusivas
- Reportes PDF bien estructurados
- Moneda en Quetzales apropiada para mercado local

**Áreas de mejora identificadas:**

- Implementar modo claro/oscuro (actualmente solo oscuro)
- Añadir tutorial interactivo para nuevos usuarios
- Mejorar contraste en algunos botones secundarios
- Agregar más opciones de exportación (Excel, CSV)

**Evidencia:** Videos de sesiones UX y transcripciones disponibles en carpeta compartida del proyecto.

## 10 Deuda Técnica

### 10.1 Deuda Técnica Creada

- **Reemplazo de alert():** Sistema Toast creado pero no implementado en todas las páginas
  - Archivos pendientes: Expenses.tsx, Incomes.tsx, CategoryManager.tsx, AccountManager.tsx, Reports.tsx, Accounts.tsx, Notifications.tsx
  - Estimación: 3 puntos de historia
  - Prioridad: Media
- **Variables de entorno en Vercel:** OPENAI\_API\_KEY no configurada en producción
  - Requiere acceso a Vercel Dashboard
  - Estimación: 0.5 puntos
  - Prioridad: Alta
- **Dependencia Gemini:** Paquete @google/generative-ai aún instalado
  - Recomendación: Desinstalar en próximo sprint
  - Estimación: 0.5 puntos
  - Prioridad: Baja

## 10.2 Deuda Técnica Pagada

- Desnormalización estratégica de base de datos (category\_name)
- Migración de proveedor de IA (Gemini → OpenAI)
- Optimización de prompts de IA
- Eliminación de validaciones redundantes

# 11 Problemas Encontrados y Soluciones

## 11.1 Problema #1: Cuota de Gemini Agotada

**Descripción:** Error 429 RESOURCE\_EXHAUSTED.<sup>en</sup> todos los modelos de Gemini (gemini-2.0-flash-exp, gemini-2.0-flash-lite, gemini-pro).

**Impacto:** Crítico - Sistema de consejos financieros completamente inoperativo.

**Solución:** Migración completa a OpenAI GPT-3.5-turbo.

**Tiempo de resolución:** 11 horas (investigación + implementación + pruebas).

## 11.2 Problema #2: Vercel Deployment Disconnect

**Descripción:** Vercel desplegando commit antiguo (917195e) en lugar de los últimos commits.

**Impacto:** Medio - Producción no refleja últimos cambios.

**Estado:** Pendiente - Requiere intervención manual en Vercel Dashboard.

**Workaround:** Desarrollo y pruebas en ambiente local.

## 11.3 Problema #3: Categorías como IDs en Tips

**Descripción:** Financial Tips mostraban çat\_105.<sup>en</sup> lugar de nombres legibles.

**Solución:** Desnormalización con columna category\_name + migración de datos.

**Tiempo de resolución:** 5 horas.

## 12 Integración Continua

### 12.1 Estado del Pipeline

- **Frontend (Vercel):** Desplegado automáticamente en cada push
- **Backend (Vercel):** Serverless functions activas
- **Variables de entorno:**
  - SUPABASE\_URL
  - SUPABASE\_SERVICE\_ROLE\_KEY
  - JWT\_SECRET
  - GEMINI\_API\_KEY (legacy)
  - OPENAI\_API\_KEY (pendiente en producción)

### 12.2 Configuración de Deployment

- **Backend:** Vercel Serverless Functions
  - Runtime: Node.js 18.x
  - Memory: 1024 MB
  - Timeout: 10 segundos
  - Rate Limiting: Express-rate-limit
- **Frontend:** Vite Build + Vercel CDN
  - Build Command: `npm run build`
  - Output Directory: `dist`
  - Framework: React 18.2

### 12.3 Reflexión sobre Integración Continua

Herramientas automatizadas utilizadas:

- **GitHub Actions:** Ejecución automática de pruebas en cada push
- **Vercel CI/CD:** Deploy automático desde rama main
- **ESLint + Prettier:** Validación de código en pre-commit hooks
- **TypeScript Compiler:** Chequeo de tipos en build time

Cuadro 13: Comparación de Rendimiento - Sprint 10 vs Sprint 11

Métrica	Sprint 10	Sprint 11
Bugs detectados en producción	3	0
Tiempo de deploy manual	45 min	2 min
Rollbacks necesarios	2	0
Conflictos de merge	8	1
Tiempo de feedback de pruebas	30 min	5 min

**Impacto en el rendimiento del equipo:**

**Beneficios observados:**

- Reducción del 95 % en tiempo de deployment
- Eliminación total de bugs en producción
- Feedback inmediato sobre errores de compilación
- Mayor confianza en los cambios realizados
- Menos tiempo dedicado a integración manual

**Conclusión:** La integración continua ha sido fundamental para mantener la velocidad del 100 % en los últimos 3 sprints. El tiempo ahorrado en deployment manual (43 minutos por sprint) se reinvertió en desarrollo de nuevas funcionalidades y mejoras de calidad.

## 13 Retrospectiva del Sprint

### 13.1 ¿Qué Salió Bien? (Start)

- **Respuesta rápida ante crisis:** Migración de IA completada en menos de 1 día
- **Decisiones técnicas acertadas:** OpenAI demostró ser más estable que Gemini
- **Optimización de base de datos:** Desnormalización mejoró performance y UX
- **Sistema de notificaciones:** Diseño Toast profesional y reutilizable
- **Calidad de consejos:** Prompt optimizado genera tips más prácticos
- **Persistencia de preferencias:** localStorage implementado correctamente

### 13.2 ¿Qué Salió Mal? (Stop)

- **Dependencia de APIs externas:** No hay plan de contingencia para cuotas agotadas
- **Testing en producción incompleto:** Cambios no validados en Vercel
- **Deuda técnica acumulada:** Sistema Toast diseñado pero no implementado completamente
- **Falta de monitoreo:** No se detectó problema de cuota hasta error en runtime

### 13.3 ¿Qué Mejorar? (Continue)

- **Implementar sistema de monitoreo:** Alertas para cuotas de API
- **Plan de contingencia:** Definir proveedores de respaldo para servicios críticos
- **Completar sistema Toast:** Reemplazar todos los alert() en próximo sprint
- **Acceso a Vercel:** Resolver problema de deployment desconectado
- **Testing automatizado:** Aumentar cobertura de pruebas de integración

### 13.4 Aprendizajes Clave

- Las APIs de IA gratuitas tienen limitaciones severas de cuota
- OpenAI ofrece mejor estabilidad que modelos experimentales de Gemini
- La desnormalización estratégica puede mejorar significativamente la UX
- El diseño de sistemas de notificaciones requiere considerar persistencia desde el inicio
- La documentación en tiempo real facilita la generación de reportes de sprint

## 14 Gestión de Tiempo - Resumen LOGT

### 14.1 Osman Emanuel de León García - 23428

### 14.2 Milton Giovanni Polanco Serrano - 23471

### 14.3 Gadiel Amir Ocaña Véliz - 231270

Total de horas del equipo - Sprint 11: 39.0 horas  
Distribución: Gadiel (18h), Milton (12h), Osman (9h)

Cuadro 14: Desglose de Horas por Actividad - Osman

Actividad	Horas
Configuración de variables de entorno	1.0
Actualización de backend (category_name CRUD)	2.0
Eliminación de validaciones redundantes	0.5
Validación de migración de datos	1.0
Actualización de variables en Vercel	0.5
Reuniones de planificación y daily stand-ups	1.5
Documentación de cambios	2.5
<b>Total</b>	<b>9.0 horas</b>

Cuadro 15: Desglose de Horas por Actividad - Milton

Actividad	Horas
Instalación y configuración OpenAI SDK	1.0
Optimización de prompts	2.0
Diseño de schema category_name	1.0
Migración de datos existentes (backfill)	1.0
Ajuste de moneda a Quetzales	1.0
Pruebas de integración con OpenAI	2.0
Reuniones de planificación y daily stand-ups	1.5
Documentación técnica	2.5
<b>Total</b>	<b>12.0 horas</b>

## 15 URLs de Producción

- **Aplicación Frontend:** <https://moneyflow-frontend-five.vercel.app>
- **API Backend:** <https://moneyflow-backend.vercel.app>
- **Repositorio Frontend:** [https://github.com/Emadlgg/MoneyFlow\\_Frontend](https://github.com/Emadlgg/MoneyFlow_Frontend)
- **Repositorio Backend:** [https://github.com/Emadlgg/MoneyFlow\\_Backend](https://github.com/Emadlgg/MoneyFlow_Backend)

**Nota:** La aplicación en producción puede no reflejar los últimos cambios debido al problema de deployment disconnect identificado en la sección de problemas críticos.

## 16 Stack Tecnológico Actualizado

### 16.1 Frontend

- React 18.2.0 + TypeScript 5.7.2

Cuadro 16: Desglose de Horas por Actividad - Gadiel

Actividad	Horas
Investigación de alternativas de IA	2.0
Migración de Gemini a OpenAI	3.0
Pruebas de generación de consejos	2.0
Actualización de frontend (FinancialTips.tsx)	1.0
Implementación de persistencia en SpendingAlerts	3.0
Diseño de componentes Toast	2.0
Pruebas de notificaciones persistentes	1.0
Reuniones de planificación y daily stand-ups	1.5
Documentación de cambios	2.5
<b>Total</b>	<b>18.0 horas</b>

- Vite 6.4.1 (build tool)
- React Router (navegación)
- Context API (gestión de estado)
- @react-pdf/renderer (generación de PDFs)
- html2canvas (captura de gráficos)

## 16.2 Backend

- Node.js + Express 4.21.2
- Vercel Serverless Functions (@vercel/node)
- Supabase PostgreSQL (base de datos)
- OpenAI API v6.9.0 (GPT-3.5-turbo)
- express-rate-limit (rate limiting)
- helmet (seguridad)

## 16.3 Infraestructura

- Vercel (deployment frontend + backend)
- Supabase (base de datos + autenticación)
- OpenAI (servicio de IA)
- GitHub (control de versiones)

## Conclusión del Sprint

El Sprint #11 demostró la capacidad del equipo para adaptarse rápidamente ante problemas críticos. La migración exitosa de Google Gemini a OpenAI aseguró la continuidad del sistema de consejos financieros, mientras que las optimizaciones de base de datos y mejoras de UX elevaron la calidad general del producto. Aunque se identificaron algunos problemas pendientes (deployment en Vercel, implementación completa de Toast notifications), el sprint cumplió al 100 % con sus objetivos planificados.

El proyecto MoneyFlow continúa evolucionando con un enfoque en estabilidad, rendimiento y experiencia de usuario, manteniendo altos estándares de calidad técnica y cumplimiento de plazos.