

Plan de Desarrollo v2.0 - Square to QuickBooks Integration (Versión Realista)

Duración: 35 días (5 semanas)

Fecha de Inicio: 5 de agosto de 2025

Fecha de Finalización: 6 de septiembre de 2025

Mejoras Implementadas en v2.0

Timeline Realista

- **Cambio:** 2 semanas → **5 semanas** (35 días)
- **Razón:** Permite desarrollo sostenible y manejo de complejidad emergente

Testing-First Philosophy

- **Cambio:** Testing tardío → **TDD/BDD desde día 1**
- **Razón:** Mejor calidad, menos riesgos, detección temprana de problemas

Trabajo Paralelo

- **Cambio:** Dependencias secuenciales → **Contratos de API desde día 2**
- **Razón:** Frontend y backend pueden desarrollarse en paralelo

Sprints Estructurados

- **Cambio:** Fases rígidas → **5 sprints de 1 semana cada uno**
 - **Razón:** Flexibilidad, feedback continuo, entregas incrementales
-

Equipo del Proyecto

Equipo Técnico

- **Juan Manuel Carvajal** - Lead Developer & Backend Specialist
- **Ahmed Hassanain** - Frontend Developer & UI/UX Implementation
- **David Jativa** - DevOps & Database Specialist

Equipo de Diseño de Flujo

- **Santiago Arevalo** - Business Process Designer
- **Eduardo Valdés** - Data Flow Architect

Metodología

- **Framework:** Scrum con TDD/BDD
 - **Sprints:** 5 sprints de 1 semana cada uno
 - **Daily Standups:** 9:00 AM (15 minutos)
 - **Sprint Reviews:** Viernes 4:00 PM
 - **Retrospectivas:** Viernes 4:30 PM
 - **Sprint Planning:** Lunes 8:00 AM
-

SPRINT 0: SETUP Y ARQUITECTURA (Semana 1)

Objetivos del Sprint

- Establecer fundación técnica sólida
- Definir contratos de API

- Implementar TDD/BDD framework
- Crear prototipos de conexión

DÍA 1 - LUNES 5 DE AGOSTO

Tema: Kickoff y Setup del Proyecto

Morning (9:00 AM - 12:00 PM)

Sprint Planning y Setup Inicial - 8:00-9:00: Sprint Planning detallado - **9:00-9:30:** Kickoff meeting y revisión del plan v2.0 - **9:30-10:30:** Setup del repositorio Git y estructura del proyecto - **10:30-12:00:** Configuración del entorno de desarrollo con nuevas tecnologías

Tareas Específicas: - **Juan Manuel:** - Crear repositorio Git con estructura de monorepo - Configurar ESLint, Prettier, Husky - Setup inicial con Pino, Argon2, Fetch API nativa - **Estimación:** 3 horas - **David:** - Setup de Docker Compose para desarrollo - Configuración de PostgreSQL y Redis containers - Scripts de inicialización de base de datos - **Estimación:** 3 horas - **Ahmed:** - Setup del proyecto React con Vite - Configuración de Tailwind CSS y Headless UI - Estructura de componentes base - **Estimación:** 3 horas

Afternoon (1:00 PM - 6:00 PM)

- **1:00-2:00:** Almuerzo y sync del equipo
- **2:00-4:00:** Implementación de configuración base
- **4:00-5:00:** Setup de TDD/BDD framework
- **5:00-6:00:** Documentación del setup

TDD/BDD Setup: - **Juan Manuel:** Configurar Jest, Supertest, coverage reports - **Ahmed:** Configurar Playwright para E2E testing - **David:** Setup de testing database y fixtures

Deliverables del Día: - ☒ Repositorio configurado con CI/CD básico - ☒ Entorno de desarrollo funcionando con nuevas tecnologías - ☒ TDD/BDD framework operativo - ☒ Base de datos inicializada

Hito 1: ☒ Entorno de desarrollo con TDD/BDD completamente funcional

DÍA 2 - MARTES 6 DE AGOSTO

Tema: Contratos de API y Database Schema

Morning (9:00 AM - 12:00 PM)

- **9:00-9:15:** Daily standup
- **9:15-12:00:** Definición de contratos de API y schema

Tareas Específicas: - **David:** - Implementar schema completo de PostgreSQL - Configurar Prisma ORM y migraciones - Crear seeds de datos de prueba - **Estimación:** 3 horas - **Juan Manuel + Eduardo:** - **NUEVO:** Definir contratos de API detallados (OpenAPI spec) - Crear mock server para desarrollo paralelo - Documentar todos los endpoints con ejemplos - **Estimación:** 3 horas - **Santiago + Eduardo:** - Definir flujos de datos detallados - Mapear todos los campos de Square a QuickBooks - Documentar reglas de transformación - **Estimación:** 3 horas





Afternoon (1:00 PM - 6:00 PM)

- **1:00-2:00:** Almuerzo
- **2:00-5:00:** Implementación y testing
- **5:00-6:00:** Validación de contratos

Juan Manuel (Afternoon): - Implementar endpoints básicos siguiendo contratos - Escribir tests unitarios para cada endpoint (TDD) - Setup de validación con Zod - **Estimación:** 3 horas

Ahmed (Afternoon): - Crear servicios de frontend basados en contratos API - Implementar mock data para desarrollo independiente - Setup de React Query con mock endpoints - **Estimación:** 3 horas

David (Afternoon): - Optimizar queries de base de datos - Configurar índices para performance - Escribir tests de integración para DB - **Estimación:** 3 horas

Deliverables del Día: -  Contratos de API completos y documentados -  Mock server funcionando -  Schema de base de datos completo -  Frontend y backend pueden desarrollarse en paralelo

Hito 2:  Contratos de API establecidos - Desarrollo paralelo habilitado

DÍA 3 - MIÉRCOLES 7 DE AGOSTO

Tema: Prototipos de Conexión y Testing

Morning (9:00 AM - 12:00 PM)

- **9:00-9:15:** Daily standup
- **9:15-12:00:** Desarrollo de prototipos con TDD





Tareas Específicas: - **Juan Manuel:** - **TDD:** Escribir tests para Square API client - Implementar cliente de Square API con Fetch nativo - Configurar autenticación OAuth - **Estimación:** 3 horas - **David:** - **TDD:** Tests para Redis y BullMQ setup - Setup de Redis para caching - Configurar BullMQ con Dead Letter Queue - **Estimación:** 3 horas - **Ahmed:** - **BDD:** Escribir escenarios de usuario en Gherkin - Crear componentes de UI para configuración - Implementar formularios de mapeo básicos - **Estimación:** 3 horas

Afternoon (1:00 PM - 6:00 PM)

Juan Manuel: - **TDD:** Tests para QuickBooks integration - Implementar prototipo de QuickBooks client - Testing con APIs sandbox - **Estimación:** 3 horas

David: - **TDD:** Tests para job processing - Implementar worker base para procesamiento - Configurar monitoring básico con Pino - **Estimación:** 3 horas

Ahmed: - **BDD:** Implementar tests E2E con Playwright - Crear dashboard básico de monitoreo - Testing de UI components - **Estimación:** 3 horas

Deliverables del Día: -  Prototipos de conexión Square y QuickBooks -  Sistema de queues con DLQ operativo -  Suite de tests comprehensiva -  UI básica con tests E2E

Hito 3:  Prototipos funcionales con testing completo

DÍA 4 - JUEVES 8 DE AGOSTO

Tema: Integración y Validación

Morning (9:00 AM - 12:00 PM)

- **9:00-9:15:** Daily standup

- **9:15-12:00:** Integración de componentes





Tareas Específicas: - **Juan Manuel:** - Integrar Square y QuickBooks clients - **TDD:** Tests de integración end-to-end - Implementar webhook receiver básico - **Estimación:** 3 horas - **David:** - **TDD:** Tests de performance y carga - Configurar job queues para diferentes tipos - Implementar retry logic y error handling - **Estimación:** 3 horas - **Santiago & Eduardo:** - Validar flujos con prototipos - Refinar reglas de mapeo basadas en testing - Crear casos de prueba complejos - **Estimación:** 3 horas

Afternoon (1:00 PM - 6:00 PM)

Juan Manuel: - Testing de integración con datos reales (sandbox) - Implementar jobs idempotentes - Validar contratos de API con implementación real - **Estimación:** 3 horas

David: - Load testing del sistema completo - Configurar alerting básico - Optimizar configuración de containers - **Estimación:** 3 horas

Ahmed: - Integrar frontend con backend real - Testing E2E del flujo completo - Refinar UI basado en feedback - **Estimación:** 3 horas

Deliverables del Día: -  Integración completa Square-QuickBooks -  Jobs idempotentes funcionando -  Testing de performance satisfactorio -  Frontend integrado con backend

Hito 4:  Integración completa con jobs idempotentes

DÍA 5 - VIERNES 9 DE AGOSTO

Tema: Sprint Review y Refinamiento

Morning (9:00 AM - 12:00 PM)

- **9:00-9:15:** Daily standup
- **9:15-12:00:** Refinamiento y optimización

Tareas Específicas (Todo el equipo): - Refactoring basado en code review - Optimización de performance - Mejora de coverage de tests - Documentación técnica

Afternoon (1:00 PM - 5:00 PM)

- **1:00-2:00:** Almuerzo
- **2:00-4:00:** Testing comprehensivo del sprint
- **4:00-5:00:** Sprint Review y Demo

Testing del Sprint: - Ejecutar toda la suite de tests - Validar performance benchmarks
- Testing manual de flujos críticos - Preparar demo para stakeholders

Deliverables del Día: - ☒ Arquitectura sólida establecida - ☒ Prototipos funcionales -
☒ Suite de tests comprehensiva - ☒ Documentación técnica completa

Hito 5: ☒ Sprint 0 completado - Fundación técnica sólida

SPRINT 1: CORE DEVELOPMENT (Semana 2)

Objetivos del Sprint

- Implementar funcionalidad core del sistema
- Desarrollar motor de mapeo
- Crear webhook processing robusto
- Establecer monitoring avanzado

DÍA 6 - LUNES 12 DE AGOSTO

Tema: Sprint Planning y Core Development

Morning (8:00 AM - 12:00 PM)

- **8:00-9:00:** Sprint Planning detallado
- **9:00-9:15:** Daily standup
- **9:15-12:00:** Desarrollo core con TDD

Tareas Específicas: - **Juan Manuel:** - **TDD:** Tests para webhook processing - Implementar webhook receiver con validación - Desarrollar order processing pipeline - **Estimación:** 3 horas - **David:** - **TDD:** Tests para mapping engine - Implementar motor de mapeo configurable - Optimizar queries de mapeo - **Estimación:** 3 horas - **Ahmed:** - **BDD:** Scenarios para gestión de mappings - Crear UI para gestión de mappings - Implementar drag & drop para mapeo visual - **Estimación:** 3 horas

Afternoon (1:00 PM - 6:00 PM)

Desarrollo Continuo: - **Juan Manuel:** Webhook processing con Dead Letter Queue - **David:** Cache de mappings con Redis - **Ahmed:** Formularios de configuración avanzada

Deliverables del Día: - ☒ Webhook processing robusto - ☒ Motor de mapeo configurable - ☒ UI de gestión de mappings

DÍA 7 - MARTES 13 DE AGOSTO

Tema: Advanced Features y Error Handling




Morning (9:00 AM - 12:00 PM)

- **9:00-9:15:** Daily standup
- **9:15-12:00:** Features avanzadas con TDD

Tareas Específicas: - **Juan Manuel:** - **TDD:** Tests para auto-mapping inteligente - Implementar auto-creación de items QB - Desarrollar bulk operations - **Estimación:** 3 horas - **David:** - **TDD:** Tests para error handling - Implementar circuit breaker pattern - Configurar comprehensive logging con Pino - **Estimación:** 3 horas - **Ahmed:** - **BDD:** Scenarios para error handling - Crear UI para visualización de errores - Implementar notificaciones en tiempo real - **Estimación:** 3 horas

Afternoon (1:00 PM - 6:00 PM)

Continuación del desarrollo: - **Juan Manuel:** Testing de auto-mapping con datos complejos - **David:** Alerting avanzado con Prometheus - **Ahmed:** Dashboard de health del sistema

Deliverables del Día: -  Auto-mapping inteligente -  Error handling robusto - 
Monitoring avanzado

DÍA 8 - MIÉRCOLES 14 DE AGOSTO

Tema: Business Logic y Reconciliación




Morning (9:00 AM - 12:00 PM)

- **9:00-9:15:** Daily standup
- **9:15-12:00:** Lógica de negocio con TDD

Tareas Específicas: - **Juan Manuel:** - **TDD:** Tests para reconciliación financiera - Implementar fee handling automático - Desarrollar auto-ajustes fiscales - **Estimación:** 3 horas - **David:** - **TDD:** Tests para consistency checks - Implementar data consistency checker - Configurar backup automático - **Estimación:** 3 horas - **Santiago & Eduardo:** - Validar lógica de negocio implementada - Refinar reglas de reconciliación - Crear test cases para edge cases - **Estimación:** 3 horas

Afternoon (1:00 PM - 6:00 PM)

Refinamiento de business logic: - **Juan Manuel:** Testing de reconciliación con datos reales - **David:** Performance optimization - **Ahmed:** UI para reconciliación manual

Deliverables del Día: -  Reconciliación financiera automática -  Auto-ajustes fiscales -  Data consistency checks

DÍA 9 - JUEVES 15 DE AGOSTO

Tema: Integration Testing y Performance

Morning (9:00 AM - 12:00 PM)

- **9:00-9:15:** Daily standup
- **9:15-12:00:** Testing de integración

Tareas Específicas: - **Juan Manuel:** - **TDD:** Integration tests comprehensivos - Testing con volúmenes altos de datos - Validación de rate limiting - **Estimación:** 3 horas -

David: - **TDD:** Performance y load testing - Database stress testing - Infrastructure optimization - **Estimación:** 3 horas - **Ahmed:** - **BDD:** E2E testing de flujos completos - UI performance testing - Accessibility testing - **Estimación:** 3 horas

Afternoon (1:00 PM - 6:00 PM)

Testing comprehensivo: - Testing manual con escenarios complejos - Bug fixing y refinamiento - Performance tuning

Deliverables del Día: - ☒ Integration testing completo - ☒ Performance optimizado - ☒ Bug fixes implementados

DÍA 10 - VIERNES 16 DE AGOSTO

Tema: Sprint Review y Demo

Morning (9:00 AM - 12:00 PM)

- **9:00-9:15:** Daily standup
- **9:15-12:00:** Preparación de demo

Preparación: - Refinamiento final de features - Preparación de datos de demo - Testing de demo escenarios

Afternoon (1:00 PM - 5:00 PM)

- **1:00-2:00:** Almuerzo
- **2:00-4:00:** Sprint Review y Demo
- **4:00-5:00:** Retrospectiva

Deliverables del Sprint: - ☒ Core functionality completa - ☒ Motor de mapeo avanzado - ☒ Error handling robusto - ☒ Monitoring comprehensivo

Hito 6: ☒ MVP funcional con features avanzadas

SPRINT 2: UI/UX Y FEATURES AVANZADAS

(Semana 3)

Objetivos del Sprint

- Completar interfaz de usuario
- Implementar analytics y reporting
- Desarrollar features de administración
- Optimizar user experience

DÍA 11-15 - SEMANA 3

Enfoque: Frontend completo, analytics, y user experience

Tareas Principales: - **Ahmed:** UI/UX completa con testing - **Juan Manuel:** API de analytics y reporting - **David:** Optimization y monitoring avanzado

Deliverables: - ☒ UI completa y responsive - ☒ Analytics dashboard - ☒ Reporting automatizado - ☒ User management

Hito 7: ☒ Sistema completo con UI avanzada

SPRINT 3: SECURITY Y COMPLIANCE

(Semana 4)

Objetivos del Sprint

- Implementar seguridad comprehensiva
- Establecer compliance

- Crear audit trails
- Preparar para producción

DÍA 16-20 - SEMANA 4

Enfoque: Security, compliance, y production readiness

Tareas Principales: - **Juan Manuel:** Authentication, authorization, audit logging - **David:** Security hardening, compliance setup - **Ahmed:** Security UI, user management

Deliverables: - ☒ Sistema seguro con RBAC - ☒ Audit trails completos - ☒ Compliance básico - ☒ Security testing

Hito 8: ☒ Sistema seguro y compliant

SPRINT 4: DEPLOYMENT Y GO-LIVE (Semana 5)

Objetivos del Sprint

- Deployment a producción
- Testing en producción
- Training y handover
- Support post-launch

DÍA 21-25 - SEMANA 5

Enfoque: Production deployment y go-live

Tareas Principales: - **David:** Production deployment y monitoring - **Juan Manuel:** Production testing y optimization - **Ahmed:** User training y documentation

Deliverables: - ☒ Sistema en producción - ☒ Monitoring 24/7 - ☒ User training completo - ☒ Support procedures

Hito Final:  Sistema en producción operativo

Recursos Humanos - Estimación Total v2.0

Distribución de Horas por Persona (5 semanas)

Juan Manuel Carvajal (Lead Developer) - Total: 200 horas (25 días × 8 horas) - Backend development: 50% - API integrations: 25% - Testing y deployment: 25%

Ahmed Hassanain (Frontend Developer) - Total: 200 horas (25 días × 8 horas) - UI/UX development: 60% - Frontend testing: 25% - Integration support: 15%

David Jativa (DevOps & Database) - Total: 200 horas (25 días × 8 horas) - Infrastructure: 35% - Database optimization: 25% - Monitoring y deployment: 40%

Santiago Arevalo (Business Process) - Total: 60 horas (20 días × 3 horas) - Process design: 50% - Validation y testing: 50%

Eduardo Valdés (Data Flow Architect) - Total: 60 horas (20 días × 3 horas) - Data architecture: 60% - API contracts: 40%

Costo Total Estimado v2.0

- **Desarrollo técnico:** 600 horas
- **Diseño de procesos:** 120 horas
- **Total:** 720 horas de proyecto

Metodología TDD/BDD Integrada

Test-Driven Development (TDD)

```
// Ejemplo de ciclo TDD
describe('OrderProcessor', () => {
  it('should process order with modifiers correctly', async () => {
    // 1. RED: Write failing test first
    const orderData = createMockOrderWithModifiers();
    const processor = new OrderProcessor();

    const result = await processor.process(orderData);

    expect(result.qbSalesReceipt.Line).toHaveLength(3);
    expect(result.qbSalesReceipt.TotalAmt).toBe(36.37);
  });
});

// 2. GREEN: Write minimal code to pass
// 3. REFACTOR: Improve code quality
```

Behavior-Driven Development (BDD)

```
# Ejemplo de escenario BDD
Feature: Order Processing
  Scenario: Process order with modifiers
    Given a Square order with modifiers
    When the webhook is received
    Then the order should be processed
    And a QuickBooks sales receipt should be created
    And modifiers should be mapped as separate line items
```

Contratos de API (Nuevo en v2.0)

OpenAPI Specification

```
# api-contracts.yaml
openapi: 3.0.0
info:
  title: Square-QuickBooks Integration API
  version: 2.0.0
paths:
  /api/orders/{orderId}/process:
    post:
      summary: Process Square order
      parameters:
        - name: orderId
          in: path
          required: true
          schema:
            type: string
      responses:
        '200':
          description: Order processed successfully
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ProcessedOrder'
```

Hitos y Criterios de Aceptación v2.0

Hito 1: Entorno TDD/BDD (Día 1)

Criterios: - ☒ Repositorio configurado con nuevas tecnologías - ☒ TDD/BDD framework operativo - ☒ Pino, Argon2, Fetch API configurados - ☒ Tests básicos ejecutándose

Hito 2: Contratos API (Día 2)

Criterios: - ☒ OpenAPI spec completa - ☒ Mock server funcionando - ☒ Frontend y backend independientes - ☒ Documentación automática

Hito 3: Prototipos con Testing (Día 3)

Criterios: - ☒ TDD: Tests para Square/QB clients - ☒ BDD: Scenarios de usuario - ☒ Dead Letter Queue configurada - ☒ E2E tests básicos

Hito 4: Jobs Idempotentes (Día 4)

Criterios: - ☒ Jobs no duplican datos - ☒ Verificación pre-creación - ☒ Testing de reprocessing - ☒ Performance satisfactorio

Hito 5: Fundación Sólida (Día 5)

Criterios: - ☒ Arquitectura robusta - ☒ 90%+ test coverage - ☒ Documentación completa - ☒ Demo funcional

Hito 6: MVP Avanzado (Día 10)

Criterios: - ☒ Auto-mapping inteligente - ☒ Auto-ajustes fiscales - ☒ Error handling robusto - ☒ Monitoring avanzado

Hito 7: UI Completa (Día 15)

Criterios: - ☒ UI responsive y accesible - ☒ Analytics dashboard - ☒ User experience optimizada - ☒ Mobile compatibility

Hito 8: Sistema Seguro (Día 20)

Criterios: - ☒ Argon2 authentication - ☒ RBAC implementado - ☒ Audit trails completos - ☒ Security testing pasando

Hito Final: Producción (Día 25)

Criterios: - ☒ Deployment exitoso - ☒ 99.9% uptime - ☒ Zero data loss - ☒ User training completado

Riesgos y Mitigación v2.0

Riesgos Técnicos Mitigados

1. **Complejidad Emergente:** 5 semanas permiten manejo de imprevistos
2. **Integration Issues:** Contratos API desde día 2
3. **Quality Issues:** TDD/BDD desde día 1

Riesgos de Timeline Mitigados

- 1. **Scope Creep:** Sprints bien definidos con objetivos claros
- 2. **Technical Debt:** TDD previene deuda técnica
- 3. **Team Dependencies:** Trabajo paralelo habilitado

Plan de Contingencia v2.0

- **Buffer de 1 semana** para issues críticos
- **Rollback plan** detallado
- **Support 24/7** primeras 2 semanas post-deployment

Comparación v1.0 vs v2.0

Aspecto	v1.0	v2.0	Mejora
Timeline	2 semanas	5 semanas	+Realismo
Testing	Final	TDD/BDD día 1	+Calidad
Desarrollo	Secuencial	Paralelo	+Eficiencia
Sprints	Fases rígidas	5 sprints ágiles	+Flexibilidad
API Contracts	No definidos	Desde día 2	+Coordinación
Risk Buffer	2 días	1 semana	+Seguridad

Esta versión 2.0 del Plan de Desarrollo es significativamente más realista y sostenible, incorporando las mejores prácticas de desarrollo ágil y asegurando la calidad desde el primer día.