⚠ DEPENDENCIAS TÉCNICAS Y PUNTOS CRÍTICOS - TIENDAS TRESMAS

🎯 RESUMEN EJECUTIVO DE RIESGOS

📊 Matriz de Criticidad

```
NIVEL DE CRITICIDAD vs IMPACTO EN NEGOCIO

🔴 CRÍTICO (Impacto Alto - Urgencia Alta)
├── 💾 Migración de Base de Datos        [Impacto: 9/10 | Urgencia: 8/10]
├── 🔒 Implementación de Seguridad       [Impacto: 10/10 | Urgencia: 9/10]
└── 🚀 Despliegue en Producción          [Impacto: 8/10 | Urgencia: 7/10]

🟡 IMPORTANTE (Impacto Medio - Urgencia Media)
├── 📊 Sistema de Monitoreo              [Impacto: 7/10 | Urgencia: 6/10]
├── ⚡ Optimización de Performance       [Impacto: 6/10 | Urgencia: 5/10]
└── 🔄 Sistema de Backup                 [Impacto: 8/10 | Urgencia: 4/10]

🟢 OPCIONAL (Impacto Bajo - Urgencia Baja)
├── 📱 Aplicación Móvil                  [Impacto: 5/10 | Urgencia: 2/10]
├── 🔗 Integraciones Adicionales         [Impacto: 4/10 | Urgencia: 2/10]
└── 🎨 Mejoras de UI/UX                  [Impacto: 3/10 | Urgencia: 1/10]
```

🔴 PUNTOS CRÍTICOS DETALLADOS

💾 MIGRACIÓN DE BASE DE DATOS

🎯 Criticidad: 🔴 **ALTA** | ⏰ **Timeline:** 2 semanas

📋 Situación Actual

```
ESTADO ACTUAL DE LA BASE DE DATOS
├── Tecnología: JSON file-based storage
├── Ubicación: In-memory + file persistence
├── Capacidad: 2,071 productos cargados
├── Performance: <10ms query time
├── Limitaciones:
│   ├── No transacciones ACID
│   ├── No concurrencia real
│   ├── Escalabilidad limitada
│   ├── Backup manual
│   └── No índices optimizados
```

🎯 Estado Objetivo

```
ESTADO OBJETIVO - POSTGRESQL
├── Tecnología: PostgreSQL 14+
├── Ubicación: Servidor dedicado
├── Capacidad: 100,000+ productos
├── Performance: <5ms query time
├── Beneficios:
│   ├── Transacciones ACID completas
│   ├── Concurrencia real (1000+ usuarios)
│   ├── Escalabilidad horizontal
│   ├── Backup automático
│   └── Índices optimizados
```

⚠️ Riesgos Identificados

1. 🔴 **Pérdida de Datos durante Migración**
2. Probabilidad: 15%
3. Impacto: Crítico
4. Mitigación: Backup completo + testing exhaustivo
5. 🟡 **Downtime Extendido**
6. Probabilidad: 30%
7. Impacto: Alto
8. Mitigación: Migración en horario no laboral
9. 🟡 **Incompatibilidades de Esquema**
10. Probabilidad: 25%
11. Impacto: Medio

12. Mitigación: Scripts de migración probados

📋 Plan de Migración

```
PLAN DE MIGRACIÓN DETALLADO

🎯 FASE 1: PREPARACIÓN (3 días)
├── Día 1: Setup PostgreSQL server
├── Día 2: Crear esquemas y tablas
├── Día 3: Scripts de migración

🎯 FASE 2: TESTING (2 días)
├── Día 4: Migración en ambiente test
├── Día 5: Validación de datos

🎯 FASE 3: PRODUCCIÓN (2 días)
├── Día 6: Backup completo
├── Día 7: Migración producción + validación
```

🔒 IMPLEMENTACIÓN DE SEGURIDAD

🎯 Criticidad: 🔴 CRÍTICA | **⏰ Timeline:** 1 semana

📋 Vulnerabilidades Actuales

```
ANÁLISIS DE SEGURIDAD ACTUAL
├── 🔴 HTTP sin encriptación
│   ├── Riesgo: Interceptación de datos
│   ├── Impacto: Credenciales expuestas
│   └── CVSS Score: 8.5/10
│
├── 🟡 JWT sin rotación
│   ├── Riesgo: Tokens de larga duración
│   ├── Impacto: Sesiones comprometidas
│   └── CVSS Score: 6.2/10
│
├── 🟡 Validación de entrada básica
│   ├── Riesgo: Inyección de código
│   ├── Impacto: Compromiso del sistema
│   └── CVSS Score: 7.1/10
│
└── 🟢 CORS configurado correctamente
    ├── Estado: Implementado
    ├── Riesgo: Bajo
    └── CVSS Score: 2.1/10
```

🎯 Implementaciones Requeridas

```
PLAN DE SEGURIDAD INTEGRAL

🔒 NIVEL 1: TRANSPORTE (Día 1-2)
├── SSL/TLS Certificate (Let's Encrypt)
├── HTTPS enforcement (301 redirects)
├── HSTS headers implementation
└── Secure cookie flags

🔒 NIVEL 2: AUTENTICACIÓN (Día 3-4)
├── JWT token rotation (15 min expiry)
├── Refresh token mechanism
├── Multi-factor authentication (opcional)
└── Password strength enforcement

🔒 NIVEL 3: APLICACIÓN (Día 5)
├── Input sanitization enhancement
├── SQL injection prevention
├── XSS protection headers
└── Rate limiting implementation
```

🛡️ Matriz de Amenazas

```
THREAT MATRIX ANALYSIS

🔴 ALTA PROBABILIDAD - ALTO IMPACTO
├── Man-in-the-middle attacks        [P: 80% | I: 9/10]
├── Session hijacking                [P: 60% | I: 8/10]
└── Data interception                [P: 70% | I: 9/10]

🟡 MEDIA PROBABILIDAD - MEDIO IMPACTO
├── Brute force attacks              [P: 40% | I: 6/10]
├── Cross-site scripting             [P: 30% | I: 5/10]
└── SQL injection                    [P: 20% | I: 8/10]

🟢 BAJA PROBABILIDAD - BAJO IMPACTO
├── DDoS attacks                     [P: 10% | I: 4/10]
├── Social engineering               [P: 15% | I: 3/10]
└── Physical access                  [P: 5% | I: 2/10]
```

🚀 DESPLIEGUE EN PRODUCCIÓN

🎯 Criticidad: 🔴 ALTA | **⏰ Timeline:** 1 semana

📋 Requisitos de Infraestructura

```
INFRAESTRUCTURA DE PRODUCCIÓN

🖥️ SERVIDOR PRINCIPAL
├── CPU: 4 cores mínimo (8 recomendado)
├── RAM: 8GB mínimo (16GB recomendado)
├── Storage: 100GB SSD mínimo
├── Network: 1Gbps connection
└── OS: Ubuntu 22.04 LTS

🌐 SERVICIOS EXTERNOS
├── Domain: tiendas-tresmas.com
├── SSL: Let's Encrypt certificate
├── CDN: CloudFlare (opcional)
├── Email: SMTP service
└── Backup: Cloud storage (AWS S3)

📊 MONITOREO
├── Uptime monitoring: UptimeRobot
├── Performance: New Relic / DataDog
├── Logs: ELK Stack
├── Alerts: PagerDuty / Slack
└── Analytics: Google Analytics
```

⚠️ Riesgos de Despliegue

```
DEPLOYMENT RISK ASSESSMENT

🔴 RIESGOS CRÍTICOS
├── 🔴 Configuración incorrecta del servidor
│   ├── Probabilidad: 25%
│   ├── Impacto: Sistema no funcional
│   └── Mitigación: Checklist de deployment
│
├── 🔴 Problemas de DNS/Domain
│   ├── Probabilidad: 20%
│   ├── Impacto: Sitio inaccesible
│   └── Mitigación: Testing previo
│
└── 🔴 Fallas en migración de datos
    ├── Probabilidad: 15%
    ├── Impacto: Pérdida de información
    └── Mitigación: Backup + rollback plan

🟡 RIESGOS MEDIOS
├── 🟡 Performance degradation
├── 🟡 SSL certificate issues
├── 🟡 Email delivery problems
└── 🟡 Monitoring setup delays

🟢 RIESGOS BAJOS
├── 🟢 Minor UI/UX issues
├── 🟢 Documentation gaps
└── 🟢 Training requirements
```

🔗 DEPENDENCIAS TÉCNICAS CRÍTICAS

📊 Mapa de Dependencias

```
DEPENDENCY MAPPING

🎯 DEPENDENCIAS CRÍTICAS (Bloquean el proyecto)
├── 🐍 Python 3.11+                    [🟢 Disponible]
│   ├── Impacto: Backend no funciona
│   ├── Alternativas: Python 3.9+
│   └── Riesgo: Bajo
│
├── 🌶️ Flask Framework                 [🟢 Estable]
│   ├── Impacto: API no disponible
│   ├── Alternativas: FastAPI, Django
│   └── Riesgo: Muy bajo
│
├── 💾 PostgreSQL 14+                  [⏳ Por instalar]
│   ├── Impacto: Escalabilidad limitada
│   ├── Alternativas: MySQL, MongoDB
│   └── Riesgo: Medio
│
└── 🔒 SSL Certificate                  [⏳ Por obtener]
    ├── Impacto: Seguridad comprometida
    ├── Alternativas: Self-signed (dev only)
    └── Riesgo: Alto

🎯 DEPENDENCIAS IMPORTANTES (Afectan funcionalidad)
├── 📊 openpyxl (Excel processing)      [🟢 Instalado]
├── 🖼️ Pillow (Image processing)        [🟢 Instalado]
├── 🔑 PyJWT (Authentication)           [🟢 Instalado]
└── 📡 requests (HTTP client)           [🟢 Instalado]

🎯 DEPENDENCIAS OPCIONALES (Mejoran experiencia)
├── 📈 Analytics libraries              [⏳ Futuro]
├── 📱 Mobile frameworks                [⏳ Futuro]
├── 🔗 Third-party APIs                 [⏳ Futuro]
└── 🎨 Advanced UI libraries            [⏳ Futuro]
```

🔄 Cadena de Dependencias

```
DEPENDENCY CHAIN ANALYSIS

📱 FRONTEND CHAIN
User Browser → HTML/CSS/JS → API Calls → Backend
├── Falla Browser: 5% usuarios afectados
├── Falla HTML/CSS: 100% usuarios afectados
├── Falla JS: 90% funcionalidad perdida
└── Falla API: 100% funcionalidad perdida

⚙ BACKEND CHAIN
API Request → Flask → Business Logic → Database → Response
├── Falla Flask: 100% API no disponible
├── Falla Logic: Funcionalidad específica afectada
├── Falla Database: 100% datos no disponibles
└── Falla Response: Timeout/Error 500

💾 DATABASE CHAIN
Query → JSON Parser → File System → Data Return
├── Falla Parser: 100% queries fallan
├── Falla File System: 100% datos perdidos
├── Falla Data Return: Inconsistencias
└── Falla Query: Funcionalidad específica afectada
```

📋 PLAN DE CONTINGENCIA

🚨 Escenarios de Crisis

```
CRISIS SCENARIOS & RESPONSE PLANS

🔴 ESCENARIO 1: PÉRDIDA TOTAL DE DATOS
├── Probabilidad: 5%
├── Impacto: Crítico (10/10)
├── Detección: Monitoring alerts + user reports
├── Respuesta Inmediata:
│   ├── 1. Activar backup más reciente (RTO: 30 min)
│   ├── 2. Notificar a usuarios (ETA: 15 min)
│   ├── 3. Investigar causa raíz
│   └── 4. Implementar fix permanente
├── Prevención:
│   ├── Backup automático cada 6 horas
│   ├── Replicación en tiempo real
│   └── Testing de restore mensual

🔴 ESCENARIO 2: COMPROMISO DE SEGURIDAD
├── Probabilidad: 15%
├── Impacto: Crítico (9/10)
├── Detección: Security monitoring + anomaly detection
├── Respuesta Inmediata:
│   ├── 1. Aislar sistema comprometido (RTO: 5 min)
│   ├── 2. Cambiar todas las credenciales
│   ├── 3. Auditoría de seguridad completa
│   └── 4. Notificación a usuarios afectados
├── Prevención:
│   ├── Penetration testing trimestral
│   ├── Security patches automáticos
│   └── Monitoring 24/7

🟡 ESCENARIO 3: DEGRADACIÓN DE PERFORMANCE
├── Probabilidad: 30%
├── Impacto: Medio (6/10)
├── Detección: Performance monitoring
├── Respuesta Inmediata:
│   ├── 1. Identificar bottleneck (RTO: 10 min)
│   ├── 2. Aplicar fix temporal
│   ├── 3. Escalar recursos si necesario
│   └── 4. Optimización permanente
├── Prevención:
│   ├── Load testing regular
│   ├── Capacity planning
│   └── Auto-scaling configurado
```

🔄 Procedimientos de Rollback

```
ROLLBACK PROCEDURES

🎯 ROLLBACK DE APLICACIÓN
├── Tiempo estimado: 15 minutos
├── Pasos:
│   ├── 1. Detener servicios actuales
│   ├── 2. Restaurar versión anterior
│   ├── 3. Verificar funcionalidad
│   └── 4. Notificar resolución
├── Triggers:
│   ├── Error rate > 5%
│   ├── Response time > 5s
│   └── User complaints > 10

🎯 ROLLBACK DE BASE DE DATOS
├── Tiempo estimado: 30 minutos
├── Pasos:
│   ├── 1. Backup de estado actual
│   ├── 2. Restaurar backup anterior
│   ├── 3. Verificar integridad
│   └── 4. Sincronizar aplicación
├── Triggers:
│   ├── Data corruption detected
│   ├── Query performance < 50%
│   └── Migration failures

🎯 ROLLBACK DE INFRAESTRUCTURA
├── Tiempo estimado: 60 minutos
├── Pasos:
│   ├── 1. Activar servidor backup
│   ├── 2. Redirigir tráfico
│   ├── 3. Sincronizar datos
│   └── 4. Verificar operación
├── Triggers:
│   ├── Server downtime > 5 min
│   ├── Network issues
│   └── Hardware failures
```

📊 MÉTRICAS DE MONITOREO CRÍTICO

🎯 KPIs Técnicos

```
TECHNICAL KPIs MONITORING

⚡ PERFORMANCE METRICS
├── Response Time API: <500ms          [🎯 Target | 🚨 Alert >2s]
├── Page Load Time: <2s                [🎯 Target | 🚨 Alert >5s]
├── Database Query: <10ms              [🎯 Target | 🚨 Alert >100ms]
├── Memory Usage: <80%                 [🎯 Target | 🚨 Alert >90%]
└── CPU Usage: <70%                    [🎯 Target | 🚨 Alert >85%]

🔒 SECURITY METRICS
├── Failed Login Attempts: <10/hour    [🎯 Target | 🚨 Alert >50/hour]
├── Suspicious IPs: 0                  [🎯 Target | 🚨 Alert >5]
├── SSL Certificate: Valid             [🎯 Target | 🚨 Alert <30 days]
├── Security Scan Score: A+            [🎯 Target | 🚨 Alert <B]
└── Vulnerability Count: 0             [🎯 Target | 🚨 Alert >0]

📊 AVAILABILITY METRICS
├── Uptime: >99.5%                     [🎯 Target | 🚨 Alert <99%]
├── Error Rate: <0.5%                  [🎯 Target | 🚨 Alert >2%]
├── Successful Requests: >99%          [🎯 Target | 🚨 Alert <95%]
├── Backup Success: 100%               [🎯 Target | 🚨 Alert <100%]
└── Recovery Time: <15min              [🎯 Target | 🚨 Alert >30min]
```

📈 Alertas Configuradas

```
ALERT CONFIGURATION

🔴 CRITICAL ALERTS (Immediate Response)
├── System Down                        [SMS + Call + Email]
├── Database Unavailable               [SMS + Call + Email]
├── Security Breach                    [SMS + Call + Email]
├── Data Loss Detected                 [SMS + Call + Email]
└── SSL Certificate Expired            [SMS + Email]

🟡 WARNING ALERTS (Response within 1 hour)
├── High Response Time                 [Email + Slack]
├── High Error Rate                    [Email + Slack]
├── Low Disk Space                     [Email + Slack]
├── High CPU/Memory Usage              [Email + Slack]
└── Backup Failure                     [Email + Slack]

🟢 INFO ALERTS (Response within 24 hours)
├── Unusual Traffic Patterns           [Email]
├── Performance Degradation            [Email]
├── New User Registrations             [Email]
├── Feature Usage Statistics           [Email]
└── System Updates Available           [Email]
```

🎯 RECOMENDACIONES ESTRATÉGICAS

📋 Prioridades Inmediatas (Próximas 2 semanas)

1. 🔒 **Implementar HTTPS** - Crítico para seguridad
2. 💾 **Migrar a PostgreSQL** - Esencial para escalabilidad
3. 📊 **Configurar Monitoreo** - Necesario para operaciones
4. 🔄 **Sistema de Backup** - Protección de datos

📋 Prioridades Mediano Plazo (1-3 meses)

1. ⚡ **Optimización de Performance** - Mejorar experiencia usuario
2. 🔗 **APIs Adicionales** - Expandir funcionalidades
3. 📱 **Aplicación Móvil** - Alcance de mercado
4. 🌍 **Internacionalización** - Expansión geográfica

📋 Prioridades Largo Plazo (3-12 meses)

1. 🤖 **Inteligencia Artificial** - Recomendaciones automáticas
2. 📊 **Analytics Avanzados** - Business Intelligence
3. 🔗 **Integraciones ERP** - Sistemas empresariales
4. ☁️ **Multi-cloud Strategy** - Redundancia y disponibilidad

---

🎯 **CONCLUSIÓN:** El proyecto TIENDAS TRESMAS presenta una **arquitectura sólida** con **riesgos controlables**. Las dependencias críticas están **identificadas y gestionadas**. El **plan de contingencia** está preparado para los escenarios más probables. La **migración a producción** requiere atención especial en **seguridad** y **base de datos**, pero el sistema está **técnicamente listo** para el despliegue exitoso.