# Contents

# □ QUANTPAYCHAIN - Reporte de Estado del Proyecto

**Fecha de Generación:** Diciembre 2024
**Versión del Proyecto:** 2.0.0
**Autor del Proyecto:** Franco Mengarelli

---

## □ Resumen Ejecutivo

**QuantPayChain** es una plataforma de tokenización de activos del mundo real (RWA - Real World Assets) con seguridad post-cuántica. El proyecto combina tecnología blockchain con criptografía resistente a computación cuántica y cumplimiento ISO 20022 para mensajería financiera.

**Estado General: □ EN DESARROLLO ACTIVO**

| Componente | Estado | Desplegado |
|---|---|---|
| Frontend Next.js | □ Funcional | Vercel |
| Backend FastAPI | □ Funcional | Render |
| Base de Datos | □ Conectado | MongoDB Atlas |
| QPC-Core Library | □ Desarrollado | □ No desplegado |
| QPC-Service (Microservicio) | □ Desarrollado | □ No desplegado |

---

## □ Arquitectura del Proyecto

```
quantpaychain/
  /app/
     backend/                          #  API Backend (FastAPI)
        server.py                     # Servidor principal
        models_earnings.py            # Modelos de ganancias
        services_earnings.py          # Servicios de dividendos
        requirements.txt              # Dependencias Python

     frontend/                         #  ABANDONADO (React CRA antiguo)

     quantpaychain-clean/              #  MONOREPO PRINCIPAL
        apps/
           web/                       #  Frontend Next.js (ACTIVO)
              app/                    # Páginas App Router
              components/             # Componentes React
              providers/              # Context Providers
              public/                 # Assets estáticos
```

```
    qpc-service/              #   Microservicio QPC (Node.js)
        src/server.ts         # API Express

packages/
    qpc-core/                 #   Librería Core TypeScript
        core/
            pqc-layer/        # Criptografía Post-Cuántica
            iso20022-gateway/ # Gateway ISO 20022
            ai-kyc-aml/       # Motor KYC/AML con IA
```

---

## ✅ Frontend (Next.js) - ACTIVO

**Ubicación:** `/app/quantpaychain-clean/apps/web`

**Tecnologías:**

- **Framework:** Next.js 14.1.0 (App Router)
- **UI:** Tailwind CSS + Radix UI + Shadcn/ui
- **Web3:** RainbowKit 2.2.9 + wagmi 2.18.2 + viem 2.40.0
- **Auth:** Supabase Auth Helpers
- **Estado:** Zustand + React Query

**Páginas Implementadas:**

| Ruta | Archivo | Estado | Descripción |
|---|---|---|---|
| `/` | `(with-web3)/page.tsx` | ✅ Funcional | Landing page con estadísticas |
| `/marketplace` | `(with-web3)/marketplace/page.tsx` | ✅ Funcional | Listado de tokens RWA |
| `/dashboard` | `dashboard/page.tsx` | ✅ Funcional | Panel de usuario |
| `/portfolio` | `portfolio/page.tsx` | ✅ Funcional | Portafolio de inversiones |
| `/earnings` | `earnings/page.tsx` | ✅ Funcional | Ganancias y dividendos |
| `/reports` | `reports/page.tsx` | ✅ Funcional | Reportes ISO 20022 |
| `/create-asset` | `create-asset/page.tsx` | ✅ Funcional | Crear nuevo activo |
| `/create-asset-v2` | `create-asset-v2/page.tsx` | ✅ Funcional | Versión alternativa |
| `/token/[id]` | `token/[id]/page.tsx` | ✅ Funcional | Detalle de token |
| `/login` | `login/page.tsx` | ✅ Funcional | Inicio de sesión |
| `/register` | `register/page.tsx` | ✅ Funcional | Registro |
| `/docs` | `docs/page.tsx` | ✅ Funcional | Documentación |
| `/docs/whitepaper` | `docs/whitepaper/page.tsx` | ✅ Funcional | Whitepaper |
| `/docs/technical-guide` | `docs/technical-guide/page.tsx` | ✅ Funcional | Guía técnica |
| `/demo` | `demo/page.tsx` | ✅ Funcional | Demostración |
| `/services` | `services/page.tsx` | ✅ Funcional | Servicios |

**Componentes Principales:**

| Componente | Archivo | Descripción |
|---|---|---|
| Navbar | `components/navbar.tsx` | Navegación principal |
| AI Advisor Panel | `components/ai-advisor-panel.tsx` | Asesor legal con IA |
| Wallet Button | `components/wallet-button.tsx` | Conexión de wallet |
| Page Layout | `components/page-layout.tsx` | Layout base |
| UI Components | `components/ui/*` | Shadcn/ui components |

**Dependencias Principales:**

```
{
  "next": "14.1.0",
  "react": "^18.2.0",
  "@rainbow-me/rainbowkit": "2.2.9",
  "wagmi": "2.18.2",
  "viem": "2.40.0",
  "@supabase/auth-helpers-nextjs": "^0.9.0",
  "@tanstack/react-query": "5.90.10",
  "tailwindcss": "^3.4.0"
}
```

⚠ **Deuda Técnica Frontend:**

1. `@supabase/auth-helpers-nextjs` - DEPRECADO, migrar a `@supabase/ssr`
2. `eslint@8.57.1` - Actualizar a v9
3. **Warnings de** `@metamask/sdk` - Deprecation de `Buffer`

---

## **Backend (FastAPI) - ACTIVO**

**Ubicación:** `/app/backend`

**Despliegue: Render (`quantpaychain-api2.onrender.com`)**

**Tecnologías:**

- **Framework:** FastAPI 0.110.1
- **Base de Datos:** MongoDB (Motor 3.3.1)
- **Pagos:** Stripe
- **IA:** OpenAI via Emergent Integrations

**Endpoints Implementados:**

**Autenticación**

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/api/auth/session` | Crear sesión desde OAuth |
| GET | `/api/auth/me` | Obtener usuario actual |
| POST | `/api/auth/logout` | Cerrar sesión |

### Activos RWA

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/api/assets` | Crear activo |
| GET | `/api/assets` | Listar activos |
| GET | `/api/assets/{id}` | Obtener activo |

### Tokens

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/api/tokens` | Crear token |
| GET | `/api/tokens` | Listar tokens |
| GET | `/api/tokens/{id}` | Obtener token |

### Blockchains

| Método | Endpoint | Descripción |
|---|---|---|
| GET | `/api/blockchains` | Listar redes soportadas |

### Pagos (Stripe)

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/api/payments/checkout` | Crear sesión de checkout |
| GET | `/api/payments/status/{session_id}` | Estado del pago |
| POST | `/api/webhook/stripe` | Webhook de Stripe |

### Transacciones

| Método | Endpoint | Descripción |
|---|---|---|
| GET | `/api/transactions` | Listar transacciones |
| POST | `/api/transactions/complete-purchase` | Completar compra |

### 🤖 IA y Reportes

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/api/ai/analyze-asset` | Analizar activo con IA |
| POST | `/api/reports/generate` | Generar reporte ISO 20022 |
| GET | `/api/reports` | Listar reportes |

### 💰 Ganancias y Dividendos

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/api/earnings/revenue` | Registrar ingreso |
| POST | `/api/earnings/distribute-dividends/{asset_id}` | Distribuir dividendos |
| GET | `/api/earnings/asset/{id}/performance` | Rendimiento de activo |
| GET | `/api/earnings/portfolio` | Portafolio del usuario |
| GET | `/api/earnings/dividends` | Historial de dividendos |
| GET | `/api/earnings/platform-stats` | Estadísticas (admin) |

### 📊 Dashboard

| Método | Endpoint | Descripción |
|---|---|---|
| GET | `/api/dashboard/stats` | Estadísticas del usuario |

**Modelos de Datos (MongoDB):**

```
# Colecciones
- users              # Usuarios registrados
- user_sessions      # Sesiones activas
- rwa_assets         # Activos del mundo real
- tokens             # Tokens emitidos
- transactions       # Transacciones
- payment_transactions # Pagos de Stripe
- iso_reports        # Reportes generados
- dividend_distributions # Distribución de dividendos
- portfolio_holdings # Holdings de usuarios
- asset_revenue      # Ingresos por activo
```

**Dependencias Principales:**

```
fastapi==0.110.1
motor==3.3.1
pymongo==4.5.0
pydantic==2.12.4
stripe==14.0.0
```

```
openai==1.99.9
emergentintegrations==0.1.0
uvicorn==0.25.0
```

---

## □ QPC-Core Library - DESARROLLADO, NO CONECTADO

**Ubicación:** `/app/quantpaychain-clean/packages/qpc-core`

Esta es la librería core que contiene toda la lógica de negocio avanzada para:

### 1. PQC Layer (Criptografía Post-Cuántica)

**Ubicación:** `core/pqc-layer/`

| Archivo | Descripción |
|---|---|
| `key-generator.ts` | Generación de keypairs resistentes a cuántica |
| `key-manager.ts` | Gestión de claves |
| `crypto-operations.ts` | Firma y verificación |
| `contract-manager.ts` | Gestión de contratos |
| `types.ts` | Tipos TypeScript |
| `errors.ts` | Errores personalizados |

**Algoritmos Soportados:** - CRYSTALS-Dilithium (firmas) - SPHINCS+ (firmas) - CRYSTALS-Kyber (encriptación)

### 2. ISO 20022 Gateway

**Ubicación:** `core/iso20022-gateway/`

| Archivo | Descripción |
|---|---|
| `parser.ts` | Parseo de mensajes XML |
| `validator.ts` | Validación de esquemas |
| `transformer.ts` | Transformación de formatos |
| `types.ts` | Tipos de mensajes |
| `errors.ts` | Errores de validación |

**Mensajes Soportados:** - `pain.001.001.08` - Payment Initiation - `pain.002.001.10` - Payment Status - `camt.053.001.08` - Bank Statement - `camt.054.001.08` - Debit/Credit Notification

### 3. AI KYC/AML Engine

**Ubicación:** `core/ai-kyc-aml/`

| Archivo | Descripción |
|---|---|
| `risk-scorer.ts` | Puntuación de riesgo |
| `sanctions-checker.ts` | Verificación de sanciones |
| `document-verifier.ts` | Verificación de documentos |
| `pattern-detector.ts` | Detección de patrones sospechosos |
| `rules-engine.ts` | Motor de reglas |
| `compliance-reporter.ts` | Generación de reportes |

**Dependencias:**

```
{
  "libsodium-wrappers": "^0.7.11",
  "fast-xml-parser": "^4.3.2",
  "ajv": "^8.12.0",
  "uuid": "^9.0.1",
  "winston": "^3.11.0"
}
```

---

### □ QPC-Service (Microservicio) - DESARROLLADO, NO DESPLEGADO

**Ubicación:** `/app/quantpaychain-clean/apps/qpc-service`

Microservicio Express.js que expone la funcionalidad de `qpc-core` via HTTP.

**Endpoints:**

**PQC (Criptografía Post-Cuántica)**

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/pqc/generate-keypair` | Generar par de claves |
| POST | `/pqc/sign` | Firmar mensaje |
| POST | `/pqc/verify` | Verificar firma |
| POST | `/pqc/encrypt` | Encriptar datos |

**ISO 20022**

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/iso20022/parse` | Parsear XML |
| POST | `/iso20022/validate` | Validar mensaje |

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/iso20022/to-internal` | Convertir a formato interno |
| POST | `/iso20022/to-iso` | Generar XML ISO |
| POST | `/iso20022/process` | Procesar mensaje completo |

### KYC/AML

| Método | Endpoint | Descripción |
|---|---|---|
| POST | `/kyc-aml/compliance-check` | Verificación de compliance |
| POST | `/kyc-aml/verify-document` | Verificar documento |
| POST | `/kyc-aml/generate-report` | Generar reporte |
| GET | `/kyc-aml/summary` | Resumen de compliance |

⚠ **Estado:**

- **Código:** ✅ Completo
- **Build:** ⚠ Requiere compilación
- **Despliegue:** ❌ No desplegado
- **Integración:** ❌ No conectado al backend principal

---

## 🗄 Base de Datos

### MongoDB Atlas

- **Provider:** MongoDB Atlas (M0 Free Tier)
- **Base de datos:** `quantpaychain_db`
- **Conexión:** Via `MONGO_URL` en variables de entorno

### Colecciones Activas:

| Colección | Descripción | Índices |
|---|---|---|
| `users` | Usuarios | `email`, `id` |
| `user_sessions` | Sesiones | `session_token` |
| `rwa_assets` | Activos RWA | `id`, `owner_id` |
| `tokens` | Tokens | `id`, `asset_id` |
| `transactions` | Transacciones | `buyer_id`, `seller_id` |
| `payment_transactions` | Pagos Stripe | `session_id` |
| `iso_reports` | Reportes | `user_id` |
| `dividend_distributions` | Dividendos | `user_id`, `asset_id` |
| `portfolio_holdings` | Holdings | `user_id`, `token_id` |
| `asset_revenue` | Ingresos | `asset_id` |

---

## □ Variables de Entorno

### Backend (`/app/backend/.env`)

```
MONGO_URL=mongodb+srv://...@cluster.mongodb.net/quantpaychain_db
DB_NAME=quantpaychain_db
SUPABASE_URL=https://xxx.supabase.co
SUPABASE_KEY=eyJ...
CORS_ORIGINS=https://www.quantpaychain.com,https://quantpaychain.com
EMERGENT_LLM_KEY=sk-emergent-xxx
STRIPE_API_KEY=sk_test_xxx
```

### Frontend (`/app/quantpaychain-clean/apps/web/.env.local`)

```
NEXT_PUBLIC_SUPABASE_URL=https://xxx.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJ...
NEXT_PUBLIC_API_URL=https://quantpaychain-api2.onrender.com
```

---

## □ Despliegues

### Frontend (Vercel)

- **URL:** `https://www.quantpaychain.com`
- **Repositorio:** Conectado a GitHub
- **Build:** `yarn build`
- **Framework:** Next.js (Auto-detectado)

### Backend (Render)

- **URL:** `https://quantpaychain-api2.onrender.com`
- **Tipo:** Web Service
- **Build Command:** `pip install -r requirements.txt`
- **Start Command:** `uvicorn server:app --host 0.0.0.0 --port $PORT`

---

## □ Lo que FALTA por Desarrollar/Conectar

### Prioridad Alta □

1. **Desplegar QPC-Service**
   - Crear servicio en Render
   - Configurar build de TypeScript
   - Conectar con backend principal
   - Estimar: 2-4 horas
2. **Integrar QPC-Core con Backend**
   - El backend debe llamar a qpc-service para:
     - Firmas post-cuánticas en transacciones

    - Validación ISO 20022 real
    - Verificación KYC/AML
  - Estimar: 4-8 horas
3. **Migrar Supabase Auth**
  - De `@supabase/auth-helpers-nextjs` a `@supabase/ssr`
  - Actualizar middleware
  - Estimar: 2-3 horas

**Prioridad Media** 

4. **Sistema DID (Identidad Digital Descentralizada)**
  - Documento técnico existe
  - Requiere diseño e implementación
  - Estimar: 40+ horas
5. **Contratos Inteligentes Reales**
  - Actualmente simulados
  - Desplegar en testnets
  - Estimar: 20+ horas
6. **Integración Web3 Completa**
  - Conexión real con wallets
  - Transacciones on-chain
  - Estimar: 16+ horas

**Prioridad Baja** 

7. **Actualizar ESLint**
  - De v8 a v9
  - Actualizar configuración
  - Estimar: 1-2 horas
8. **Tests Automatizados**
  - Unit tests para qpc-core
  - Integration tests para API
  - E2E tests para frontend
  - Estimar: 16+ horas

---

## Integraciones de Terceros

| Servicio | Estado | Uso |
|---|---|---|
| **Supabase** |  Conectado | Autenticación OAuth |
| **MongoDB Atlas** |  Conectado | Base de datos |
| **Stripe** |  Conectado | Procesamiento de pagos |
| **OpenAI** (via Emergent) |  Conectado | AI Advisor, Análisis |
| **RainbowKit** |  Configurado | Conexión de wallets |
| **Vercel** |  Desplegado | Hosting frontend |

| Servicio | Estado | Uso |
|---|---|---|
| **Render** |  Desplegado | Hosting backend |

---

## Archivos Importantes de Referencia

```
/app/backend/server.py              # API principal
/app/backend/services_earnings.py   # Lógica de dividendos
/app/backend/models_earnings.py     # Modelos Pydantic

/app/quantpaychain-clean/apps/web/
   app/layout.tsx                   # Layout raíz
   app/(with-web3)/layout.tsx       # Layout con Web3
   middleware.ts                    # Middleware de auth
   next.config.js                   # Configuración Next
   package.json                     # Dependencias

/app/quantpaychain-clean/packages/qpc-core/
   core/index.ts                    # Exportaciones
   core/pqc-layer/                  # Criptografía PQ
   core/iso20022-gateway/           # Gateway financiero
   core/ai-kyc-aml/                 # Motor compliance

/app/quantpaychain-clean/apps/qpc-service/
   src/server.ts                    # Microservicio
```

---

## Problemas Conocidos

1. **Caché de Vercel** - Requiere “Clear build cache” al desplegar
2. **CORS en producción** - Debe configurarse `CORS_ORIGINS` exactamente
3. **PWA Service Worker** - Puede cachear versiones antiguas
4. **Web3Provider** - Solo en rutas `(with-web3)` para evitar conflictos

---

## Contacto y Recursos

- **Dominio:** www.quantpaychain.com
- **API Docs:** https://quantpaychain-api2.onrender.com/docs
- **Vercel Dashboard:** dashboard.vercel.com
- **Render Dashboard:** dashboard.render.com
- **MongoDB Atlas:** cloud.mongodb.com

---

## 🚨 Notas para Nuevos Desarrolladores

1. **NO usar** `/app/frontend` - Es el proyecto React antiguo, ABANDONADO
2. **El proyecto principal es** `/app/quantpaychain-clean`
3. **Siempre usar yarn**, nunca npm
4. **Hot reload activo** - No reiniciar servicios manualmente
5. **Variables de entorno** - Nunca hardcodear, usar `.env`
6. **Prefijo** `/api` - Todas las rutas del backend deben tenerlo
7. **MongoDB** `_id` - Siempre excluir con `{"_id": 0}` en queries

---

*Documento generado automáticamente - Diciembre 2024*