



# GUÍA COMPLETA DE DEPLOYMENT EN VERCCEL - QuantPay Chain MVP

**Fecha:** 24 de Octubre de 2024

**Tiempo estimado:** 20-30 minutos para deployment básico

**Nivel:** Principiante-Intermedio



## ÍNDICE

1. Pre-requisitos
2. Configuración del Repositorio
3. Deployment en Vercel
4. Configuración de Base de Datos
5. Configuración de Variables de Entorno
6. Ejecución de Migraciones
7. Verificación y Testing
8. Configuración de Dominio Personalizado
9. Troubleshooting
10. Mantenimiento Post-Deployment

## 1. PRE-REQUISITOS

### Checklist antes de empezar:

- [ ] Cuenta de GitHub con acceso al repositorio
- [ ] Cuenta de Vercel (crear gratis en <https://vercel.com>)
- [ ] Git configurado localmente con correo correcto
- [ ] Node.js 22.x instalado (verificar con `node -v`)
- [ ] Acceso al repositorio: <https://github.com/francoMengarelli/quantpaychain-mvpro>

### Verificar estructura del proyecto:

```
# Tu proyecto debe tener esta estructura:
quantpaychain-mvpro/
├── package.json                      # Config raíz
├── vercel.json                        # Config Vercel raíz
└── quantpaychain-mvp/
    └── frontend/app/
        ├── package.json                # Aplicación Next.js
        ├── vercel.json                 # Config Vercel del frontend
        ├── next.config.js              # App Router de Next.js
        └── app/
```

## 🔍 Verificación del vercel.json raíz:

El archivo `/vercel.json` debe apuntar al directorio correcto del frontend:

```
{
  "buildCommand": "npm run build",
  "devCommand": "npm run dev",
  "installCommand": "npm install",
  "framework": "nextjs",
  "outputDirectory": ".next"
}
```

Y en `/package.json` raíz, los scripts deben apuntar al frontend:

```
{
  "scripts": {
    "dev": "cd quantpaychain-mvp/frontend/app && npm run dev",
    "build": "cd quantpaychain-mvp/frontend/app && npm run build",
    "start": "cd quantpaychain-mvp/frontend/app && npm run start",
    "postinstall": "cd quantpaychain-mvp/frontend/app && npm install"
  }
}
```

## 2. CONFIGURACIÓN DEL REPOSITORIO

### Paso 2.1: Verificar el repositorio remoto

```
cd /ruta/a/tu/proyecto/quantpaychain-mvpro
git remote -v
```

Debe mostrar:

```
origin https://github.com/francoMengarelli/quantpaychain-mvpro.git (fetch)
origin https://github.com/francoMengarelli/quantpaychain-mvpro.git (push)
```

### Paso 2.2: Asegurar que el historial de commits está correcto

Después de haber reescrito el historial con `git-filter-repo`, verificar:

```
# Ver los últimos 10 commits
git log --format="%ae | %an | %s" -10
```

Todos deben mostrar `fmengarelli@gmail.com`.

### Paso 2.3: Force push al repositorio (SOLO SI REESCRIBISTE EL HISTORIAL)

**⚠️ ADVERTENCIA:** Esto sobrescribirá el historial en GitHub. Asegúrate de tener un backup.

```
# Push forzado de todas las ramas
git push origin --force --all

# Push forzado de todos los tags (si existen)
git push origin --force --tags
```

## Paso 2.4: Verificar en GitHub

1. Ir a: <https://github.com/francoMengarelli/quantpaychain-mvpro>
  2. Verificar que los commits muestran tu correo correcto
  3. Verificar que no hay warnings de autenticación
- 

## 3. DEPLOYMENT EN VERCEL

### Método 1: Import desde GitHub (Recomendado) 🏆

#### Paso 3.1: Conectar GitHub con Vercel

1. **Ir a Vercel Dashboard:**
  - <https://vercel.com/dashboard>
2. **Add New Project:**
  - Click en “Add New...” → “Project”
3. **Import Git Repository:**
  - Click en “Import Git Repository”
  - Seleccionar GitHub
  - Si es tu primera vez, autorizar Vercel en GitHub
4. **Seleccionar el repositorio:**
  - Buscar: `francoMengarelli/quantpaychain-mvpro`
  - Click en “Import”

#### Paso 3.2: Configurar Build Settings

Vercel debería auto-detectar que es un proyecto Next.js, pero verifica:

```
Framework Preset: Next.js

Root Directory: ./
(0 especificar: quantpaychain-mvp/frontend/app si no funciona con raíz)

Build Command: npm run build
Output Directory: .next
Install Command: npm install
Development Command: npm run dev

Node.js Version: 22.x
```

**IMPORTANTE:** Si Vercel no detecta correctamente la estructura:

1. Click en “Edit” en “Root Directory”
2. Navegar a: `quantpaychain-mvp/frontend/app`
3. Guardar

## Paso 3.3: Configurar Configuración del Proyecto (antes de Deploy)

**NO hagas deploy todavía.** Primero configura:

### 1. Environment Variables:

- Click en “Environment Variables” (expand)
- Ver [Paso 5](#) para agregar las variables
- **MÍNIMO:** DATABASE\_URL , NEXTAUTH\_SECRET , NEXTAUTH\_URL

### 2. Build & Development Settings:

- Verificar que “Build Command” apunta a `npm run build`
- Verificar que “Output Directory” es `.next`

### 3. Deploy Button:

- **SOLO** después de agregar las variables críticas, click en “Deploy”

## Método 2: Vercel CLI (Alternativo)

### Paso 3.1: Instalar Vercel CLI

```
npm i -g vercel
```

### Paso 3.2: Login

```
vercel login
```

Seguir las instrucciones en el navegador.

### Paso 3.3: Link del proyecto

```
cd /ruta/a/quantpaychain-mvpro
vercel link
```

Responder:

```
? Set up and deploy "~/quantpaychain-mvpro"? [Y/n] y
? Which scope do you want to deploy to? → [Tu usuario de Vercel]
? Link to existing project? [y/N] n
? What's your project's name? quantpaychain-mvpro
? In which directory is your code located? ./
```

### Paso 3.4: Deploy

```
# Deploy a preview
vercel

# Deploy a producción (después de configurar variables)
vercel --prod
```

## 4. CONFIGURACIÓN DE BASE DE DATOS

### Opción A: Vercel Postgres (Recomendado - Más Fácil) 🏆

#### Paso 4.1: Crear Base de Datos

##### 1. En Vercel Dashboard:

- Ir a tu proyecto → Pestaña “Storage”
- Click en “Create Database”
- Seleccionar “Postgres”

##### 2. Configurar:

Database Name: quantpaychain-db  
 Region: [Seleccionar la más cercana a tus usuarios]

##### 3. Create:

- Click en “Create”
- Esperar a que se cree (toma ~1 minuto)

##### 4. Conectar al Proyecto:

- Vercel te pedirá conectar la DB al proyecto
- Seleccionar tu proyecto: quantpaychain-mvpro
- Click en “Connect”

##### 5. Variables de Entorno Auto-Agregadas:

- Vercel agregará automáticamente:

- POSTGRES\_URL
- POSTGRES\_PRISMA\_URL ← **Usar este como DATABASE\_URL**
- POSTGRES\_URL\_NON\_POOLING
- Y otros...

##### 6. Configurar DATABASE\_URL:

- Ir a Settings → Environment Variables
- Agregar nueva variable:

Key: DATABASE\_URL  
 Value: [Copiar el valor de POSTGRES\_PRISMA\_URL]

### Opción B: Supabase (Alternativa Gratis) 🏆

#### Paso 4.1: Crear Proyecto en Supabase

##### 1. Ir a: <https://supabase.com>

##### 2. Sign up / Login

##### 3. New Project:

Organization: [Tu organización]  
 Name: quantpaychain-db  
 Database Password: [Generar contraseña segura - GUARDARLA]  
 Region: [Seleccionar región cercana]

##### 4. Create new project:

- Esperar a que se cree (~2 minutos)

## Paso 4.2: Obtener Connection String

### 1. Project Settings → Database:

- Scroll down a “Connection string”
- Seleccionar “Connection pooling” (recomendado para Prisma)
- Copiar la URI:  

```
postgres://postgres.[PROJECT-REF]:[PASSWORD]@aws-0-[REGION].pooler.supabase.com:6543/
postgres?pgbouncer=true
```

### 2. Reemplazar [PASSWORD] con tu contraseña real

### 3. Agregar en Vercel:

- Settings → Environment Variables
- Key: DATABASE\_URL
- Value: [Pegar connection string]
- Environments: Production, Preview, Development
- Save

## Opción C: Neon (Alternativa Serverless) 🏆

1. **Crear cuenta:** <https://neon.tech>
2. **New Project**
3. **Copiar connection string**
4. **Agregar como DATABASE\_URL en Vercel**

## 5. CONFIGURACIÓN DE VARIABLES DE ENTORNO

### Paso 5.1: Generar NEXTAUTH\_SECRET

En tu terminal local:

```
openssl rand -base64 32
```

Copiar el output (ejemplo: X3jK9mNpQrS7tVwYz2aBcDeFgHiJkLmN )

### Paso 5.2: Agregar Variables Críticas en Vercel

1. **Ir a:** Vercel Dashboard → Tu Proyecto → Settings → Environment Variables
2. **Agregar una por una:**

#### Variable 1: DATABASE\_URL

```
Key: DATABASE_URL
Value: [Tu connection string de Postgres]
Environments:  Production  Preview  Development
```

## Variable 2: NEXTAUTH\_SECRET

Key: NEXTAUTH\_SECRET  
 Value: [Output de openssl rand -base64 32]  
 Environments:  Production  Preview  Development

## Variable 3: NEXTAUTH\_URL

Key: NEXTAUTH\_URL  
 Value: https://[tu-proyecto].vercel.app  
 Environments:  Production (dejar Preview y Dev vacíos)

**Nota:** Para NEXTAUTH\_URL , reemplazar [tu-proyecto] con el nombre real que te asignó Vercel.

## Paso 5.3: Agregar Variables Opcionales (Recomendadas)

Ver documento `VERCEL_ENV_SETUP.md` para lista completa.

**Mínimo recomendado para funcionalidad básica:**

```
# Stripe (Test Mode)
STRIPE_SECRET_KEY="sk_test_..."
NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY="pk_test_..."

# Pinata (Almacenamiento)
PINATA_JWT="eyJhbG...
NEXT_PUBLIC_PINATA_API_KEY="..."
NEXT_PUBLIC_PINATA_SECRET="..."

# OpenAI (Auditoría IA)
OPENAI_API_KEY="sk-proj-..."
AI_PROVIDER="openai"
```

## Paso 5.4: Guardar y Redeploy

Después de agregar todas las variables:

1. **Guardar cada una** (click “Add” después de cada variable)
2. **Redeploy el proyecto:**
  - Deployments → Último deployment
  - Click en “...” → “Redeploy”
  - O hacer un nuevo push a GitHub

## 6. EJECUCIÓN DE MIGRACIONES

### Importante: Prisma Migrations

Después de configurar DATABASE\_URL , necesitas ejecutar las migraciones de Prisma.

## Método 1: Desde Vercel CLI (Recomendado)

```
# En tu proyecto local
cd quantpaychain-mvp/frontend/app

# Generar cliente Prisma
npx prisma generate

# Ejecutar migraciones en producción
npx prisma migrate deploy --preview-feature
```

## Método 2: Configurar Build Command

Modificar el build command en Vercel para que ejecute migraciones automáticamente:

1. **Vercel Dashboard → Settings → General:**

2. **Build Command:**

```
bash
  npm run build
```

Cambiar a:

```
bash
  cd quantpaychain-mvp/frontend/app && npx prisma generate && npx prisma migrate deploy && npm
run build
```

1. **Save**

2. **Redeploy**

## Método 3: Usar Vercel Deploy Hook con GitHub Action

Crear un GitHub Action que ejecute migraciones:

```
# .github/workflows/deploy.yml
name: Deploy to Vercel

on:
  push:
    branches: [main]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3

      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '22'

      - name: Install dependencies
        run: |
          cd quantpaychain-mvp/frontend/app
          npm install

      - name: Generate Prisma Client
        run: |
          cd quantpaychain-mvp/frontend/app
          npx prisma generate

      - name: Run migrations
        env:
          DATABASE_URL: ${{ secrets.DATABASE_URL }}
        run: |
          cd quantpaychain-mvp/frontend/app
          npx prisma migrate deploy

      - name: Trigger Vercel Deploy
        run: curl -X POST ${{ secrets.VERCEL_DEPLOY_HOOK }}
```

## 7. VERIFICACIÓN Y TESTING

### Paso 7.1: Verificar Build Exitoso

#### 1. Vercel Dashboard → Deployments:

- Verificar que el último deployment tiene “Ready”
- Si tiene , click para ver logs de error

#### 2. Ver Build Logs:

- Click en el deployment
- Pestaña “Building”
- Buscar errores relacionados con:
  - Variables de entorno faltantes
  - Errores de Prisma
  - Errores de compilación de TypeScript

## Paso 7.2: Verificar Runtime Logs

### 1. Deployments → Functions:

- Ver logs en tiempo real de las funciones serverless
- Buscar errores de conexión a DB

## Paso 7.3: Testing Manual de la Aplicación

### 1. Abrir la URL de producción:

```
https://[tu-proyecto].vercel.app
```

### 2. Test de Health Check:

```
bash
```

```
curl https://[tu-proyecto].vercel.app/api/health
```

Debe retornar:

```
json
{"status": "ok"}
```

### 1. Test de Autenticación:

- Ir a: `https://[tu-proyecto].vercel.app/auth/signin`
- Verificar que la página carga correctamente
- Intentar crear una cuenta
- Verificar que puedes login

### 2. Test de Dashboard:

- Después de login, verificar que el dashboard carga
- Verificar que se ven las secciones principales

### 3. Test de Demo:

- Ir a: `https://[tu-proyecto].vercel.app/demo`
- Verificar que la página de demo funciona
- Intentar simular una transacción

## Paso 7.4: Verificar Conexión a Base de Datos

### En la UI:

1. Crear un usuario
2. Logout
3. Login con ese usuario
4. Si funciona, la DB está conectada 

### Desde Prisma Studio (local):

```
cd quantpaychain-mvp/frontend/app

# Conectar a la DB de producción
DATABASE_URL="tu-production-db-url" npx prisma studio
```

Verificar que hay registros en las tablas.

## 8. CONFIGURACIÓN DE DOMINIO PERSONALIZADO (OPCIONAL)

---

### Paso 8.1: Agregar Dominio en Vercel

1. Vercel Dashboard → Tu Proyecto → Settings → Domains

2. Add Domain:

Ejemplo: quantpaychain.com

3. Configurar DNS:

Vercel te dará instrucciones específicas. Generalmente:

#### Opción A: Dominio raíz (quantpaychain.com)

Tipo: A

Name: @

Value: 76.76.21.21

#### Opción B: Subdomain (app.quantpaychain.com)

Tipo: CNAME

Name: app

Value: cname.vercel-dns.com

1. Esperar propagación DNS (5-60 minutos)

2. Verificar:

bash

```
curl https://quantpaychain.com/api/health
```

### Paso 8.2: Actualizar NEXTAUTH\_URL

Después de configurar el dominio:

1. Vercel → Settings → Environment Variables

2. Editar NEXTAUTH\_URL :

De: https://[tu-proyecto].vercel.app

A: https://quantpaychain.com

3. Redeploy

---

## 9. TROUBLESHOOTING

### Error: “Build failed - Command failed with exit code 1”

#### Causas comunes:

1. Variables de entorno faltantes
2. Errores de TypeScript
3. Problemas con Prisma

#### Solución:

1. Ver logs completos en Vercel Dashboard
2. Verificar que DATABASE\_URL está configurada
3. Verificar que el build funciona localmente:

```
bash
npm run build
```

---

## Error: “Prisma Client could not connect to database”

### Solución:

1. Verificar formato de `DATABASE_URL`:

```
postgresql://user:password@host:5432/dbname?sslmode=require
```

2. Verificar que la DB está accesible públicamente

3. Verificar que las migraciones se ejecutaron:

```
bash
```

```
npx prisma migrate deploy
```

---

## Error: “NextAuth: Missing secret”

### Solución:

1. Verificar que `NEXTAUTH_SECRET` está configurada en Vercel

2. Redeploy después de agregar la variable
- 

## Error: “CORS error” al conectar wallet

### Solución:

1. Verificar que `NEXT_PUBLIC_APP_URL` está configurada

2. Agregar el dominio en la whitelist de WalletConnect (si usas WalletConnect real)
- 

## Error: “Module not found” en runtime

### Solución:

1. Verificar que todas las dependencias están en `package.json`

2. Ejecutar `npm install` y hacer push

3. Limpiar cache de Vercel:

- Settings → General → Clear Cache
- 

## Error: “Function timeout” en API routes

### Solución:

1. Vercel tiene límite de 10 segundos por función en plan gratis

2. Optimizar funciones pesadas

3. O considerar plan Pro (60 segundos timeout)
-

## 10. MANTENIMIENTO POST-DEPLOYMENT

---

### Monitoreo

#### 1. Analytics de Vercel:

- Dashboard → Analytics
- Ver métricas de uso, latencia, errores

#### 2. Function Logs:

- Dashboard → Functions
- Ver logs en tiempo real de las API routes

#### 3. Configurar Notificaciones:

- Settings → Notifications
- Activar alertas de deployment

### Actualizaciones

#### Workflow de desarrollo:

##### 1. Develop localmente:

```
bash
git checkout -b feature/nueva-funcionalidad
# Hacer cambios
npm run dev # Testear localmente
```

##### 2. Commit y Push:

```
bash
git add .
git commit -m "feat: nueva funcionalidad"
git push origin feature/nueva-funcionalidad
```

##### 3. Preview Deploy (automático):

- Vercel creará un preview deploy automáticamente
- URL: [https://\[tu-proyecto\]-git-feature-\[branch\].\[tu-usuario\].vercel.app](https://[tu-proyecto]-git-feature-[branch].[tu-usuario].vercel.app)

##### 4. Merge a Main:

```
bash
git checkout main
git merge feature/nueva-funcionalidad
git push origin main
```

##### 5. Production Deploy (automático):

- Vercel deployará automáticamente a producción

### Backups de Base de Datos

#### Si usas Vercel Postgres:

- Backups automáticos incluidos

#### Si usas Supabase:

- Backups diarios automáticos en plan gratis

#### Manual backup:

```
# Exportar datos
npx prisma db pull

# O hacer un dump completo
pg_dump $DATABASE_URL > backup.sql
```

## Rollback

Si algo sale mal después de un deploy:

1. **Vercel Dashboard → Deployments**
2. **Seleccionar un deployment anterior que funcionaba**
3. **Click en “...” → “Promote to Production”**

Esto hará rollback al deployment anterior instantáneamente.

---

## CHECKLIST FINAL

### Antes de considerar el deployment completo:

- [ ]  Build exitoso en Vercel
  - [ ]  DATABASE\_URL configurada
  - [ ]  NEXTAUTH\_SECRET configurada
  - [ ]  NEXTAUTH\_URL configurada
  - [ ]  Migraciones de Prisma ejecutadas
  - [ ]  Health check endpoint funciona
  - [ ]  Autenticación funciona (crear usuario y login)
  - [ ]  Dashboard carga correctamente
  - [ ]  Demo page funciona
  - [ ]  Variables opcionales configuradas (Stripe, Pinata, OpenAI)
  - [ ]  No hay errores en Runtime Logs
  - [ ]  Testing manual de funcionalidades principales
  - [ ]  Dominio personalizado configurado (opcional)
  - [ ]  Notificaciones de Vercel activadas
  - [ ]  Documentación del proyecto actualizada
- 



## RECURSOS ADICIONALES

### Documentación Oficial:

- **Vercel:** <https://vercel.com/docs>
- **Next.js:** <https://nextjs.org/docs>
- **Prisma:** <https://www.prisma.io/docs>
- **NextAuth.js:** <https://next-auth.js.org>

### Documentos del Proyecto:

- PROJECT\_INVENTORY.md - Inventario completo del proyecto

- `INTEGRATION_STATUS.md` - Estado de integraciones
- `VERCEL_ENV_SETUP.md` - Guía detallada de variables de entorno

## Sopporte:

- **Vercel Support:** <https://vercel.com/support>
  - **Vercel Community:** <https://github.com/vercel/vercel/discussions>
- 



## CONCLUSIÓN

¡Felicitaciones! Si seguiste todos los pasos, tu aplicación QuantPay Chain MVP debería estar funcionando en producción.

### Próximos pasos recomendados:

1. Configurar variables opcionales para features adicionales
  2. Agregar dominio personalizado
  3. Configurar monitoring y alertas
  4. Realizar testing exhaustivo de todas las funcionalidades
  5. Configurar CI/CD con GitHub Actions (opcional)
  6. Implementar tests automatizados
- 

**Documento generado el 24 de Octubre de 2024**

¡Éxito con tu deployment!