

Guía de Despliegue - QuantPayChain MVP

Tabla de Contenidos

1. [Requisitos Previos](#)
2. [Despliegue de Contratos](#)
3. [Despliegue del Frontend](#)
4. [Configuración de Variables de Entorno](#)
5. [Verificación del Despliegue](#)
6. [Troubleshooting](#)

Requisitos Previos

Software Necesario

- **Node.js**: v18.x o superior
- **npm**: v9.x o superior
- **Foundry**: Última versión estable
- **Git**: Para clonar el repositorio
- **MetaMask**: Extensión de navegador configurada

Cuentas y Servicios

- **Wallet Ethereum**: Con fondos en Sepolia testnet
- **Alchemy/Infura**: API key para RPC provider
- **Etherscan**: API key para verificación de contratos (opcional)
- **Vercel**: Cuenta para despliegue de frontend (opcional)

Obtener ETH de Testnet

```
# Faucets recomendados para Sepolia:  
# - https://sepoliafaucet.com/  
# - https://www.alchemy.com/faucets/ethereum-sepolia  
# - https://faucet.quicknode.com/ethereum/sepolia
```

Despliegue de Contratos

1. Clonar el Repositorio

```
git clone https://github.com/francoMengarelli/quantpaychain-mvp.git  
cd quantpaychain-mvp/contracts
```

2. Instalar Dependencias

```
# Instalar dependencias de Foundry
forge install

# Verificar instalación
forge --version
```

3. Configurar Variables de Entorno

Crear archivo `.env` en el directorio `contracts/` :

```
# RPC URL para Sepolia
SEPOLIA_RPC_URL=https://eth-sepolia.g.alchemy.com/v2/YOUR_API_KEY

# Private key de la wallet de despliegue (¡NUNCA compartir!)
PRIVATE_KEY=0xYOUR_PRIVATE_KEY_HERE

# Etherscan API key para verificación
ETHERSCAN_API_KEY=YOUR_ETHERSCAN_API_KEY

# Dirección del deployer (tu wallet)
DEPLOYER_ADDRESS=0xYOUR_WALLET_ADDRESS
```

4. Compilar Contratos

```
# Compilar todos los contratos
forge build

# Verificar que no hay errores
forge test
```

5. Ejecutar Despliegue

```
# Desplegar en Sepolia testnet
forge script script/Deploy.s.sol:DeployScript \
  --rpc-url $SEPOLIA_RPC_URL \
  --private-key $PRIVATE_KEY \
  --broadcast \
  --verify \
  --etherscan-api-key $ETHERSCAN_API_KEY

# El script mostrará las direcciones de los contratos desplegados
```

6. Guardar Direcciones de Contratos

El script de despliegue generará un archivo con las direcciones:

```
{
  "PaymentProcessor": "0x...",
  "TokenManager": "0x...",
  "DisputeResolver": "0x...",
  "Governance": "0x..."
}
```

⚠ **IMPORTANTE:** Guardar estas direcciones para configurar el frontend.

Despliegue del Frontend

1. Navegar al Directorio Frontend

```
cd ../frontend
```

2. Instalar Dependencias

```
npm install
```

3. Configurar Variables de Entorno

Crear archivo `.env.local` :

```
# Direcciones de contratos (del paso anterior)
NEXT_PUBLIC_PAYMENT_PROCESSOR_ADDRESS=0x...
NEXT_PUBLIC_TOKEN_MANAGER_ADDRESS=0x...
NEXT_PUBLIC_DISPUTE_RESOLVER_ADDRESS=0x...
NEXT_PUBLIC_GOVERNANCE_ADDRESS=0x...

# Configuración de red
NEXT_PUBLIC_CHAIN_ID=11155111
NEXT_PUBLIC_RPC_URL=https://eth-sepolia.g.alchemy.com/v2/YOUR_API_KEY

# Configuración de la aplicación
NEXT_PUBLIC_APP_NAME=QuantPayChain
NEXT_PUBLIC_APP_URL=https://your-domain.vercel.app
```

4. Desarrollo Local

```
# Iniciar servidor de desarrollo
npm run dev

# La aplicación estará disponible en http://localhost:3000
```

5. Build de Producción

```
# Crear build optimizado
npm run build

# Probar build localmente
npm start
```

6. Despliegue en Vercel

Opción A: Desde la CLI

```
# Instalar Vercel CLI
npm i -g vercel

# Login en Vercel
vercel login

# Desplegar
vercel --prod
```

Opción B: Desde GitHub

1. Conectar repositorio en vercel.com (<https://vercel.com>)
2. Configurar variables de entorno en el dashboard
3. Vercel desplegará automáticamente en cada push

7. Configurar Variables en Vercel Dashboard

1. Ir a Project Settings → Environment Variables
2. Agregar todas las variables de `.env.local`
3. Seleccionar entornos: Production, Preview, Development
4. Guardar y redespargar si es necesario

Configuración de Variables de Entorno

Variables Críticas





Contratos

```
# Nunca commitear estas variables
PRIVATE_KEY=          # Clave privada del deployer
SEPOLIA_RPC_URL=       # URL del proveedor RPC
ETHERSCAN_API_KEY=    # Para verificación de contratos
```

Frontend

```
# Variables públicas (prefijo NEXT_PUBLIC_)
NEXT_PUBLIC_PAYMENT_PROCESSOR_ADDRESS= # Dirección del contrato
NEXT_PUBLIC_TOKEN_MANAGER_ADDRESS=     # Dirección del contrato
NEXT_PUBLIC_DISPUTE_RESOLVER_ADDRESS=  # Dirección del contrato
NEXT_PUBLIC_GOVERNANCE_ADDRESS=        # Dirección del contrato
NEXT_PUBLIC_CHAIN_ID=                   # ID de la red (11155111 para Sepolia)
NEXT_PUBLIC_RPC_URL=                    # URL pública del RPC
```

Seguridad de Variables

-  Usar `.env.example` como plantilla
-  Agregar `.env*` a `.gitignore`
-  Rotar claves regularmente
-  Nunca commitear archivos `.env`
-  Nunca compartir claves privadas

Verificación del Despliegue

1. Verificar Contratos en Etherscan

```
# Visitar Sepolia Etherscan
https://sepolia.etherscan.io/address/YOUR_CONTRACT_ADDRESS

# Verificar que:
# - El contrato está verificado (✓ icono verde)
# - El código fuente es visible
# - Las transacciones de despliegue son exitosas
```

2. Probar Frontend

```
# Checklist de funcionalidad:
# ✓ Conexión de MetaMask funciona
# ✓ Cambio de idioma (ES/EN) funciona
# ✓ Navegación entre páginas funciona
# ✓ Interacción con contratos funciona (en modo demo)
# ✓ Responsive design en móvil/tablet/desktop
```

3. Ejecutar Tests

```
# Tests de contratos
cd contracts
forge test -vvv

# Tests de frontend (si existen)
cd ../frontend
npm test
```

4. Verificar CI/CD

```
# Revisar GitHub Actions
# - Build de contratos: ✓
# - Tests de contratos: ✓
# - Build de frontend: ✓
# - Despliegue automático: ✓
```

Troubleshooting

Problemas Comunes

Error: “Insufficient funds”

```
# Solución: Obtener más ETH de testnet
# Visitar faucets de Sepolia y solicitar fondos
```

Error: “Nonce too low”

```
# Solución: Resetear nonce en MetaMask
# Settings → Advanced → Reset Account
```

Error: “Contract verification failed”

```
# Solución: Verificar manualmente
forge verify-contract \
  --chain-id 11155111 \
  --compiler-version v0.8.20 \
  YOUR_CONTRACT_ADDRESS \
  src/YourContract.sol:YourContract \
  --etherscan-api-key $ETHERSCAN_API_KEY
```

Error: “RPC rate limit exceeded”

```
# Solución: Usar un RPC provider con mayor límite
# - Alchemy: 300 req/s en plan gratuito
# - Infura: 100,000 req/día en plan gratuito
```

Error: “Module not found” en Frontend

```
# Solución: Limpiar caché y reinstalar
rm -rf node_modules .next
npm install
npm run build
```

Logs y Debugging

Ver logs de despliegue de contratos

```
# Foundry guarda logs en:
broadcast/Deploy.s.sol/11155111/run-latest.json
```

Ver logs de Vercel

```
# Desde CLI
vercel logs

# Desde dashboard
# Project → Deployments → [Select deployment] → Logs
```

Contacto para Soporte

Si los problemas persisten:

1. Revisar [GitHub Issues](https://github.com/francoMengarelli/quantpaychain-mvp/issues) (<https://github.com/francoMengarelli/quantpaychain-mvp/issues>)
2. Crear un nuevo issue con:
 - Descripción del problema
 - Logs relevantes
 - Pasos para reproducir
 - Entorno (OS, versiones de software)

Checklist Final de Despliegue

- [] Contratos compilados sin errores
- [] Tests de contratos pasando (59/59)

- [] Contratos desplegados en Sepolia
- [] Contratos verificados en Etherscan
- [] Variables de entorno configuradas
- [] Frontend desplegado en Vercel
- [] Conexión MetaMask funcional
- [] Internacionalización funcionando
- [] Responsive design verificado
- [] CI/CD configurado y funcionando
- [] Documentación actualizada
- [] README con instrucciones claras

Última actualización: Octubre 2025

Versión: 1.0.0