Demo Usage Guide - QuantPayChain MVP

⚠️ **AUTOMATIC TRANSLATION — REVIEW REQUIRED**

This document has been automatically translated from Spanish. Technical review is recommended for accuracy.

Introduction

This guide will help you explore all the features of the QuantPayChain MVP demo. The demo is designed to showcase the system's capabilities without needing to connect a real wallet or perform blockchain transactions.

Demo Access

Production URL

```
https://quantpaychain-mvp.vercel.app
```

Local Development

```
cd frontend
npm install
npm run dev
# Visit http://localhost:3000
```

Demo Features

Simulated Mode

- ✅ No MetaMask required
- ✅ No testnet funds required
- ✅ Instant simulated transactions
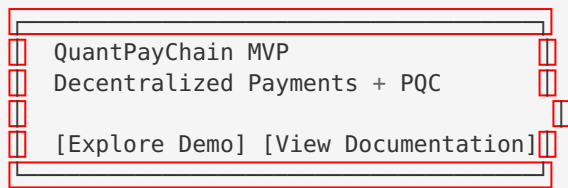- ✅ Preloaded sample data
- ✅ Fully functional for demonstration

Main Navigation

1. Home Page

Main Sections

- **Hero Section**: Project introduction
- **Features**: Cards with key functionalities
- **Statistics**: System metrics
- **Call to Action**: Buttons to explore

Interactive Elements

```
QuantPayChain MVP
Decentralized Payments + PQC


[Explore Demo] [View Documentation]
```

2. Dashboard

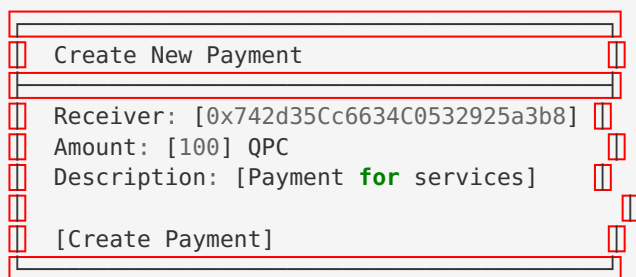### Overview

Dashboard shows user activity summary:

```
Dashboard

Balance: 1000 QPC
Active Payments: 3
Disputes: 1

[Create Payment] [View History]
```

Available Metrics

- **Total Balance**: Available QPC tokens
- **Pending Payments**: Payments in escrow
- **Completed Payments**: Transaction history
- **Active Disputes**: Cases in resolution

3. Create Payment

Payment Form

```
Create New Payment

Receiver: [0x742d35Cc6634C0532925a3b8]
Amount: [100] QPC
Description: [Payment for services]

[Create Payment]
```

Steps to Create Payment

1. **Enter Receiver Address**
   - Format: Valid Ethereum address (0x...)
   - Automatic format validation

2. **Specify Amount**
   - Minimum: 1 QPC
   - Maximum: Available balance
   - Real-time validation

3. **Add Description**
   - Optional but recommended
   - Maximum 200 characters
   - Helps with tracking

4. **Confirm Transaction**
   - Review details
   - Click "Create Payment"
   - Instant confirmation in demo mode

Result

```
✅ Payment created successfully
ID: #12345
Status: PENDING
Funds locked in escrow
```

4. Payment Management

Payment List

```
┌──────────────────────────────────────────────┐
│  My Payments                                 │
├──────────────────────────────────────────────┤
│ #12345 | 100 QPC | PENDING   | [Complete] [Dispute]│
│ #12344 | 50 QPC  | COMPLETED | [View Details]│
│ #12343 | 75 QPC  | DISPUTED  | [View Dispute]│
└──────────────────────────────────────────────┘
```

Available Actions

Complete Payment

- **Who**: Only payer
- **When**: PENDING status
- **Effect**: Releases funds to receiver
- **Simulation**: Instant

Request Refund

- **Who**: Only receiver
- **When**: PENDING status
- **Effect**: Returns funds to payer
- **Simulation**: Instant

Initiate Dispute

- **Who**: Payer or receiver
- **When**: PENDING status
- **Effect**: Locks payment and opens case
- **Simulation**: Creates simulated dispute

5. Dispute System

Dispute View

```
┌──────────────────────────────────────┐
│  Dispute #001                         │
├──────────────────────────────────────┤
│  Payment: #12345                      │
│  Initiator: 0x742d... (Payer)         │
│  Reason: Service not delivered        │
│  Status: VOTING                       │
│                                       │
│  Votes for payer: 3                   │
│  Votes for receiver: 1                │
│                                       │
│  [Submit Evidence] [View History]     │
└──────────────────────────────────────┘
```

Resolution Process

1. Dispute Opening

```
Payer/Receiver → Initiate Dispute
             ↓
        Provide Reason
             ↓
        Status: OPEN
```

2. Evidence Submission

```
Involved Parties → Submit Evidence
              ↓
         Documents/Links
              ↓
         Visible to Arbitrators
```

3. Voting

```
Arbitrators → Review Evidence
          ↓
       Cast Vote
          ↓
       Status: VOTING
```

4. Resolution

```
System → Count Votes
        ↓
   Determine Winner
        ↓
   Release/Refund Funds
        ↓
   Status: RESOLVED
```

6. Governance

Create Proposal

```
New Proposal

Title: [Reduce transaction fee]
Description:
[I propose reducing the fee from 2% to 1.5%
 to incentivize more transactions...]

Actions:
- Contract: PaymentProcessor
- Function: setFeePercentage
- Parameters: 150 (1.5%)

[Create Proposal]
```

Vote on Proposals

```
Proposal #005

Reduce transaction fee
Proposed by: 0x742d...
Status: ACTIVE

Votes for: 1,250,000 QPC (62%)
Votes against: 750,000 QPC (38%)

Your voting power: 10,000 QPC

[Vote For] [Vote Against]
```

Internationalization

Language Switching

Language Selector

```
🌐 ES ▼

✓ Español
  English
```

Available Languages

- **Spanish (ES)**: Default language
- **English (EN)**: Complete translation

Persistence

- Preference saved in localStorage

- Maintained between sessions
- Instant change without reload

Advanced Features

1. Search and Filters

Filter Payments

```
┌─────────────────────────────────────┐
│ ┌─────────────────────────────────┐ │
│ │  Filters                        │ │
│ ├─────────────────────────────────┤ │
│ │  Status: [All ▼]                │ │
│ │  Date: [Last month ▼]           │ │
│ │  Amount: [Any ▼]                │ │
│ │                                 │ │
│ │  [Apply] [Clear]                │ │
│ └─────────────────────────────────┘ │
└─────────────────────────────────────┘
```

Search

```
┌─────────────────────────────────────┐
│ 🔍 Search by ID, address or description │
└─────────────────────────────────────┘
```

2. Notifications

Notification Types

- ✅ **Success**: Operation completed
- ℹ️ **Info**: General information
- ⚠️ **Warning**: Action requires attention
- ❌ **Error**: Operation failed

Example

```
┌─────────────────────────────────────┐
│ ┌─────────────────────────────────┐ │
│ │ ✅ Payment completed successfully │ │
│ │ ID: #12345 | Amount: 100 QPC    │ │
│ │ [View Details] [×]              │ │
│ └─────────────────────────────────┘ │
└─────────────────────────────────────┘
```

3. Transaction History

History View

```
┌──────────────────────────────────────┐
│  Transaction History                 │
├──────────────────────────────────────┤
│  Date        | Type      | Amount | Status    │
├──────────────────────────────────────┤
│  10/09/2025  | Payment   | 100    | Completed │
│  10/08/2025  | Refund    | 50     | Completed │
│  10/07/2025  | Payment   | 75     | Disputed  │
└──────────────────────────────────────┘
```

Export Data

```
[Export CSV] [Export PDF]
```

Responsive Design

Views by Device

Desktop (>1024px)

- 3-column layout
- Navigation sidebar
- Expanded charts

Tablet (768px - 1024px)

- 2-column layout
- Hamburger menu
- Adapted charts

Mobile (<768px)

- 1-column layout
- Bottom bar navigation
- Simplified charts

Sample Data

Simulated Users

```javascript
const demoUsers = [
  {
    address: "0x742d35Cc6634C0532925a3b844BC454e4438f44e",
    name: "Alice",
    balance: 1000
  },
  {
    address: "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
    name: "Bob",
    balance: 500
  }
];
```

Preloaded Payments

```javascript
const demoPayments = [
  {
    id: 12345,
    payer: "0x742d...",
    payee: "0x5B38...",
    amount: 100,
    status: "PENDING",
    description: "Payment for services"
  }
];
```

Troubleshooting

Common Issues

Demo doesn't load

```
Solution:
1. Verify internet connection
2. Clear browser cache
3. Try incognito mode
4. Verify JavaScript is enabled
```

Language switching doesn't work

```
Solution:
1. Verify browser localStorage
2. Clear cookies and cache
3. Reload page
```

Buttons don't respond

```
Solution:
1. Check browser console (F12)
2. Report error with screenshot
3. Try another browser
```

Best Practices

For Demonstrations

1. **Preparation**
   - Familiarize with all functions
   - Have use case scenarios prepared
   - Verify everything works before presenting

2. **During Demo**
   - Explain context of each action
   - Show different user flows
   - Highlight security features

3. **Frequently Asked Questions**
   - How does it integrate with real blockchain?
   - What about post-quantum cryptography?
   - When will it be in production?

Next Steps

After Demo

1. **Explore Documentation**
   - README.md (./README.md)
   - CONTRACTS.md (./CONTRACTS.md)
   - SECURITY-PQC.md (../SECURITY-PQC.md)

2. **Test with Testnet**
   - Connect MetaMask
   - Get Sepolia ETH
   - Perform real transactions

3. **Contribute**
   - Report bugs
   - Suggest improvements
   - Contribute code

Contact

For demo questions:
- **GitHub Issues**: Report issue (https://github.com/francoMengarelli/quantpaychain-mvp/issues)
- **Documentation**: See `/docs` folder

Last updated: October 2025
Demo version: 1.0.0