

■■ SISTEMA STC TRADING

Plataforma Profesional de Trading Automatizado

IQ Option • Mercados OTC • Dashboard Web

■ Versión	Dashboard Pro v1.0
■■ Fecha	30 Septiembre 2025
■ Arquitectura	3 Capas (Frontend + API + Cliente)
■ Acceso Web	http://localhost:5001
■ Trading	OTC 24/7 - CALL/PUT Binario
■ Gráficos	TradingView Lightweight Charts
■ Mercados	EURUSD, GBPUSD, USDJPY, EURJPY OTC

ÍNDICE

1.	Arquitectura del Sistema	3
2.	Flujo de Inicio y Arranque	4
3.	Flujo de Credenciales	5
4.	Flujo de Velas M5	6
5.	Flujo de Órdenes de Trading	7
6.	Diccionario de Archivos	8
7.	Bondades del Sistema	9
8.	Puertos y Servicios	10
9.	Requerimientos del Sistema	11
10.	Guía de Instalación	12
11.	Configuración y Setup	13
12.	Troubleshooting	14

1. ■■ ARQUITECTURA DEL SISTEMA

Componentes Principales

El sistema STC Trading está construido con una arquitectura de 3 capas independientes que se comunican a través de APIs REST y protocolos HTTP. Esta separación garantiza escalabilidad, mantenibilidad y robustez del sistema completo.

Componente	Puerto	Tecnología	Función Principal
Dashboard Frontend	5001	Flask + HTML/JS	Interface web de usuario
API Backend	5002	Flask + Redis	Gestión de datos y órdenes
Cliente IQ Option	-	Python + IQ API	Conexión real con broker

Flujo de Comunicación

`Usuario → Dashboard (5001) → API (5002) → Cliente IQ → IQ Option Servers`

2. ■ FLUJO DE INICIO Y ARRANQUE

Secuencia de Arranque Obligatoria

El sistema debe iniciarse en un orden específico para garantizar que todos los componentes se conecten correctamente y los servicios estén disponibles cuando se necesiten.

Orden	Archivo	Puerto	Función	Tiempo Espera
1º	iq_routes_redis_patch.py	5002	API Backend + Redis Cache	5 segundos
2º	dashboard_server.py	5001	Servidor Web Dashboard	4 segundos
3º	iq_client.py	-	Cliente IQ Option	5 segundos

Script de Inicio Automático

```
start_dashboard_pro.bat ■■■ Detiene procesos previos ■■■ Inicia API Backend (puerto 5002) ■■■ Inicia Dashboard Web (puerto 5001) ■■■ Inicia Cliente IQ Option ■■■ Verifica que todos los servicios estén activos
```

3. ■ FLUJO DE CREDENCIALES

Ruta de Autenticación

Las credenciales de IQ Option viajan de forma segura a través de la arquitectura del sistema, desde la interface web hasta la conexión real con los servidores de IQ Option.

Etapas	Componente	Acción	Datos Procesados
1	dashboard_pro.html	Captura formulario	Email + Password + Balance Type
2	dashboard_server.py	Proxy HTTP POST	Redirige a puerto 5002
3	iq_routes_redis_patch.py	Procesa /api/iq/login	Valida y encola credenciales
4	iq_client.py	Conexión IQ Option	Login real + cambio de balance
5	Respuesta	Confirmación	Session ID + Estado conexión

Seguridad y Almacenamiento

- **Almacenamiento:** Memoria Redis + localStorage del navegador
- **Transmisión:** HTTP POST (local network)
- **Validación:** IQ Option servers via HTTPS
- **Sesión:** Persistente hasta logout manual

4. ■ FLUJO DE VELAS M5

Origen y Destino de Datos de Velas

Las velas de 5 minutos (M5) son la base del análisis técnico en el sistema. Estos datos viajan desde los servidores de IQ Option hasta el dashboard web para visualización en tiempo real.

```
IQ Option API → iq_client.py → Memory Redis → iq_routes_redis_patch.py
→ dashboard_pro.html → TradingView Charts
```

Etapas	Componente	Función	Datos
Obtención	iq_client.py	get_candles() cada 2s	OHLCV + timestamp
Cache	MemoryRedis	Almacén temporal	Últimas 200 velas
API	ruta /api/iq/candles	GET endpoint	JSON con velas filtradas
Frontend	ChartManager class	Fetch automático	Renderizado TradingView
Visualización	Lightweight Charts	Gráfico interactivo	Velas + línea BID

Características de las Velas

- **Timeframe:** M5 (5 minutos) - Optimizado para trading binario
- **Símbolos OTC:** EURUSD-OTC, GBPUSD-OTC, USDJPY-OTC, EURJPY-OTC
- **Disponibilidad:** 24/7 (mercados OTC siempre abiertos)
- **Actualización:** Cada 2 segundos con nuevos datos
- **Histórico:** Últimas 200 velas en memoria para análisis

5. ■ FLUJO DE ÓRDENES DE TRADING

Envío de Orden CALL/PUT

El proceso de envío de órdenes es crítico para el trading. El sistema garantiza que cada orden sea procesada de forma segura y eficiente desde el dashboard hasta IQ Option.

```
Usuario (CALL/PUT) → Dashboard → API → Cola Redis → IQ Client → IQ  
Option → Confirmación
```

Paso	Componente	Acción	Datos	Tiempo
1	dashboard_pro.html	Clic botón CALL/PUT	Symbol + Amount + Duration	Inmediato
2	JavaScript fetch()	POST /api/iq/trade	JSON con parámetros orden	< 100ms
3	dashboard_server.py	Proxy a puerto 5002	Forward request	< 50ms
4	iq_routes_redis_patch.py	Procesa y encola	Orden en cola Redis	< 200ms
5	iq_client.py	Ejecuta en IQ Option	api.buy() real	1-3 segundos
6	Confirmación	Resultado al dashboard	Order ID + Status	< 500ms

6. ■■ DICCIONARIO DE ARCHIVOS Y FUNCIONES

Backend Core

Archivo	Puerto	Líneas	Función Principal	Dependencias Clave
iq_routes_redis_patch.py	5002	~400	API REST + Cache Redis	Flask, MemoryRedis
dashboard_server.py	5001	~120	Servidor web + Proxy	Flask, requests
iq_client.py	-	~450	Cliente IQ Option	iqoptionapi, httpx

Frontend y Templates

Archivo	Tamaño	Función	Tecnologías
dashboard_pro.html	~1200 líneas	Interface moderna trading	HTML5 + CSS3 + JavaScript ES6
dashboard.html	~1200 líneas	Interface original (backup)	HTML5 + TradingView Charts

Scripts de Automatización

Script	Función	Orden de Ejecución
start_dashboard_pro.bat	Inicio sistema completo nuevo	Recomendado principal
start_system_final.bat	Inicio sistema anterior	Backup alternativo
restart_fixed_system.bat	Reinicio con correcciones	Para troubleshooting
test_trading_routes.bat	Pruebas de endpoints API	Para desarrollo

7. ■ BONDADES Y VENTAJAS DEL SISTEMA

Arquitectura Modular

- **Separación de responsabilidades:** Frontend, API y Cliente independientes
- **Escalabilidad:** Cada componente puede escalar individualmente
- **Mantenibilidad:** Modificaciones aisladas sin afectar otros módulos
- **Testabilidad:** Cada capa puede probarse por separado

Trading 24/7 OTC

- **Mercados OTC:** Disponibles las 24 horas, 7 días de la semana
- **Símbolos principales:** EURUSD-OTC, GBPUSD-OTC, USDJPY-OTC, EURJPY-OTC
- **Sin restricciones horarias:** Opera en cualquier momento
- **Latencia optimizada:** Cache en memoria para ejecución rápida

Interface Profesional

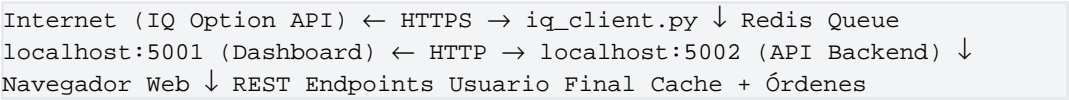
- **Dashboard moderno:** Diseño elegante inspirado en plataformas profesionales
- **Gráficos TradingView:** Velas M5 interactivas con indicadores
- **Trading integrado:** Botones CALL/PUT con un solo clic
- **Estadísticas en vivo:** Balance, operaciones y rendimiento actualizados
- **Responsive design:** Compatible con desktop y móvil

Robustez y Confiabilidad

- **Manejo inteligente de errores:** Reconexión automática y recuperación
- **Cache eficiente:** Datos persistentes en memoria para velocidad
- **Fallbacks múltiples:** CDNs de respaldo para librerías externas
- **Logs detallados:** Sistema completo de debugging y monitoreo
- **Validaciones:** Verificación de datos en cada etapa

8. ■ PUERTOS Y SERVICIOS

Mapa de Conectividad



Puerto	Servicio	Protocolo	Función	Acceso
5001	Dashboard Web	HTTP	Interface usuario	http://localhost:5001
5002	API Backend	HTTP	REST API + Cache	http://localhost:5002
-	Cliente IQ	HTTPS	Conexión externa	Proceso background
443	IQ Option	HTTPS	Broker real	Internet (external)

URLs Principales del Sistema

Endpoint	Método	Función	Respuesta
/	GET	Dashboard principal	HTML interface
/health	GET	Estado del sistema	JSON status
/api/iq/balance	GET	Saldo actual	JSON balance
/api/iq/candles	GET	Velas M5	JSON array
/api/iq/trade	POST	Enviar orden	JSON result
/api/iq/login	POST	Autenticación	JSON session

9. ■ REQUERIMIENTOS DEL SISTEMA

Sistema Operativo

OS	Versión Mínima	Recomendado	Notas
Windows	Windows 10	Windows 11	Tested y optimizado
Linux	Ubuntu 18.04+	Ubuntu 22.04	Compatible
macOS	macOS 10.14+	macOS 12+	Compatible

Hardware

Componente	Mínimo	Recomendado	Observaciones
CPU	Dual Core 2.0 GHz	Quad Core 3.0 GHz	Para múltiples procesos
RAM	4 GB	8 GB	Python + Flask + Chrome
Disco	2 GB libres	5 GB libres	Logs y cache
Red	Banda ancha	Fibra óptica	Latencia crítica trading

Software Base

Software	Versión	Propósito	Instalación
Python	3.9+	Runtime principal	python.org
Navegador Web	Chrome 90+	Interface dashboard	Pre-instalado
Terminal/CMD	Nativo	Ejecutar scripts	Sistema

10. ■ GUÍA DE INSTALACIÓN

Paso 1: Instalación de Python

1. Descargar Python 3.9+ desde **python.org**
2. Ejecutar instalador y marcar "Add to PATH"
3. Verificar: abrir CMD y escribir **python --version**
4. Debe mostrar: Python 3.9.x o superior

Paso 2: Descargar Sistema STC

1. Extraer archivos en **C:\STC_Trading_System**
2. Abrir CMD como Administrador
3. Navegar: **cd C:\STC_Trading_System**
4. Verificar archivos: **dir** (debe mostrar iq_client.py, etc.)

Paso 3: Crear Entorno Virtual

```
cd C:\STC_Trading_System python -m venv .venv311 .venv311\Scripts\activate  
python -m pip install --upgrade pip pip install -r requirements.txt
```

Paso 4: Configurar Variables

1. Abrir archivo **.env** con Notepad
2. Configurar símbolos OTC (ya preconfigurado)
3. Ajustar timeouts si es necesario
4. Guardar cambios

Paso 5: Primer Arranque

```
cd C:\STC_Trading_System start_dashboard_pro.bat
```

Verificación de Instalación

- Abrir navegador en **http://localhost:5001**
- Ver dashboard con interface moderna
- Verificar que muestra "Sistema iniciando..."
- No debe haber errores en ventanas de CMD

11. ■■ CONFIGURACIÓN Y SETUP

Paquetes Python Requeridos

Paquete	Versión	Propósito	Crítico
Flask	2.3.x	Servidor web framework	Sí
iqoptionapi	6.8.9.1	Conexión IQ Option	Sí
requests	2.31.x	HTTP client	Sí
httpx	0.24.x	Async HTTP client	Sí
python-dotenv	1.0.x	Variables entorno	Sí
reportlab	4.0.x	Generación PDF	No

Archivo requirements.txt

```
Flask==2.3.3 iqoptionapi==6.8.9.1 requests==2.31.0 httpx==0.24.1  
python-dotenv==1.0.0 reportlab==4.0.4
```

Variables de Entorno (.env)

Variable	Valor Por Defecto	Descripción
IQ_SYMBOLS	EURUSD-OTC,GBPUSD-OTC,...	Símbolos para trading
MAX_BUFFER_SIZE	2000	Tamaño cache velas
FRONTEND_CLOSED_LIMIT	300	Límite velas dashboard
ACTIVE_SOURCE	iq	Fuente de datos principal

Configuración de Red

- **Puertos locales:** 5001 y 5002 deben estar libres
- **Firewall:** Permitir conexiones locales Python
- **Antivirus:** Excluir carpeta STC_Trading_System
- **Internet:** Conexión estable para IQ Option API

12. ■ TROUBLESHOOTING Y SOLUCIÓN DE PROBLEMAS

Problemas Comunes

Problema	Causa	Solución
Puerto ocupado	Otro proceso usa 5001/5002	taskkill /f /im python.exe
Gráficos no cargan	LightweightCharts no disponible	Refrescar navegador (Ctrl+F5)
IQ no conecta	Credenciales incorrectas	Verificar email/password
Velas no actualizan	Cliente IQ desconectado	Reiniciar iq_client.py
Órdenes fallan	Mercado cerrado	Usar símbolos OTC únicamente
Dashboard no abre	Servidor no iniciado	Ejecutar start_dashboard_pro.bat

Comandos de Diagnóstico

```
# Verificar Python instalado python --version # Verificar puertos ocupados netstat -an | findstr "5001\|5002" # Verificar procesos Python tasklist | findstr python # Probar conectividad API curl http://localhost:5002/health # Probar dashboard curl http://localhost:5001/health # Reinicio completo sistema cd C:\STC_Trading_System start_dashboard_pro.bat
```

Logs y Debugging

- **Logs del sistema:** Se muestran en ventanas CMD abiertas
- **Logs del navegador:** F12 → Console para errores JavaScript
- **Archivos de log:** Carpeta logs/ (se crean automáticamente)
- **Nivel de detalle:** INFO por defecto, cambiar a DEBUG si necesario

Contacto y Soporte

- **Documentación:** Archivos README.md en el proyecto
- **Logs detallados:** Activar DEBUG en archivos Python
- **Backup dashboard:** http://localhost:5001/dashboard_old
- **Reset completo:** Eliminar .venv311 y reinstalar



Sistema STC Trading - Documentación Completa
Generado el: 30/09/2025 19:54:17
Versión: Dashboard Pro v1.0
Plataforma: Windows / Linux / macOS Compatible