# Cambios Implementados en Simos - Rama manusUpdate

## Resumen de Mejoras

Este documento describe todos los cambios implementados en el sistema Simos para mejorar el entrenamiento de modelos de IA y actualizar la API a la versión 3.

## Archivos Modificados

### Frontend (React)

#### 1. Nuevo Componente de Entrenamiento

* **TrainingPage/TrainingPage.jsx**: Componente principal para el entrenamiento de modelos de IA
* **TrainingPage/TrainingPage.css**: Estilos para el componente de entrenamiento
* **TrainingPage/TrainingVisualization.jsx**: Componente para visualización en tiempo real del entrenamiento
* **TrainingPage/TrainingVisualization.css**: Estilos para la visualización

#### 2. Actualizaciones de Navegación

* **App.jsx**: Agregada nueva ruta /entrenamiento para acceder al sistema de entrenamiento
* **Sidebar.jsx**: Agregado botón "Entrenamiento IA" en el menú de navegación

#### 3. Mejoras en Configuración Data AI

* **ConfigDataPage.jsx**: Actualizada para usar API v3 y mostrar datos gráficamente

### Backend (Python)

#### 1. Nuevos Manejadores

* **training\_handler.py**: Manejador completo para operaciones de entrenamiento del modelo
* **simulation\_handler.py**: Manejador para simulaciones del modelo entrenado
* **socket\_optimizer.py**: Optimizador de comunicación por WebSocket

#### 2. API v3

* **api\_v3\_routes.py**: Nuevas rutas de API v3 para entrenamiento y simulación

#### 3. Actualizaciones del Modelo de IA

* **ai\_model.py**: Mejorado para incluir todas las transacciones, fees y comisiones
* **main\_v3.py**: Integración de todos los nuevos componentes

## Funcionalidades Implementadas

### 1. Sistema de Entrenamiento Dividido por Secciones

#### Creación de CSV de Datos

* Formulario para configurar parámetros de entrenamiento
* Validación de fecha (anterior a la actual)
* Cálculo automático de operaciones posibles
* Selección de símbolos (cantidad numérica o lista específica)
* Configuración de intervalos de tiempo
* Envío de datos a Simo por endpoint con respuesta de status

#### Entrenamiento del Modelo

* Carga y conversión de CSV a diccionario para el modelo
* Inclusión de todas las transacciones, fees y comisiones
* Visualización gráfica en tiempo real del estado del modelo vía WebSocket
* Gráfico final visible al completar el entrenamiento

#### Sistema de Pruebas

* Carga de CSV diferente al de entrenamiento
* Ejecución de pruebas del modelo
* Visualización de resultados de precisión, recall y F1-score

#### Simulación

* Configuración de parámetros de simulación
* Ejecución de simulaciones del modelo entrenado
* Visualización de resultados y evolución del balance

### 2. Actualización a API v3

#### Eliminación de V2

* Removidas referencias a la versión V2 de la UI
* Actualización completa a la nueva API v3

#### Nuevos Endpoints

* /api/v3/create-training-csv: Crear CSV de entrenamiento
* /api/v3/start-training: Iniciar entrenamiento del modelo
* /api/v3/run-tests: Ejecutar pruebas del modelo
* /api/v3/start-simulation: Iniciar simulación
* /api/v3/data-ai: Obtener datos del modelo (una sola vez)
* /api/sebo/symbols: Obtener símbolos de Sebo

#### Configuración Data AI

* Datos del modelo mostrados gráficamente
* Envío de datos una sola vez por /v3
* Actualización solo cuando se solicita explícitamente

### 3. Optimización de Comunicación por Socket

#### Top 20

* Envío automático cada 5 segundos
* Optimización para reducir carga del servidor

#### Balance

* Actualización solo al cargar la página
* Actualización al finalizar una operación
* Cache de balance para mejorar rendimiento

## Estructura de Archivos

archivos\_modificados/

├── README\_CAMBIOS.md # Este archivo

├── TrainingPage/ # Componente de entrenamiento

│ ├── TrainingPage.jsx

│ ├── TrainingPage.css

│ ├── TrainingVisualization.jsx

│ └── TrainingVisualization.css

├── App.jsx # Rutas actualizadas

├── Sidebar.jsx # Navegación actualizada

├── ConfigDataPage.jsx # Configuración Data AI mejorada

├── training\_handler.py # Manejador de entrenamiento

├── simulation\_handler.py # Manejador de simulación

├── api\_v3\_routes.py # Rutas API v3

├── socket\_optimizer.py # Optimizador de WebSocket

├── main\_v3.py # Aplicación principal actualizada

└── ai\_model.py # Modelo de IA mejorado

## Instalación y Uso

### 1. Frontend

1. Copiar los archivos del frontend a sus ubicaciones correspondientes en UI/clients/src/
2. Instalar dependencias si es necesario: npm install
3. Reiniciar el servidor de desarrollo

### 2. Backend

1. Copiar los archivos del backend a sus ubicaciones correspondientes en V3/
2. Instalar dependencias de Python si es necesario
3. Reiniciar el servidor V3

### 3. Acceso

* Navegar a la nueva sección "Entrenamiento IA" desde el menú lateral
* Usar la pestaña "Datos AI" en "Config Data" para ver información del modelo

## Características Técnicas

### Validaciones Implementadas

* Fecha anterior a la actual para datos de entrenamiento
* Formato CSV válido para archivos de prueba
* Validación de parámetros de simulación
* Verificación de modelo entrenado antes de pruebas/simulación

### Optimizaciones

* Cache de balance para reducir llamadas a API
* Envío optimizado de Top 20 cada 5 segundos
* Carga de datos AI solo cuando se solicita
* Visualización en tiempo real sin bloquear la UI

### Compatibilidad

* Compatible con la estructura existente de Simos
* No rompe funcionalidades existentes
* Integración transparente con V3

## Notas de Desarrollo

* Todos los archivos mantienen la estructura y convenciones del proyecto original
* Se agregaron logs detallados para facilitar el debugging
* Manejo de errores robusto en todas las funcionalidades
* Interfaz responsive para diferentes tamaños de pantalla

## Próximos Pasos

1. Probar todas las funcionalidades en el entorno de desarrollo
2. Verificar la integración con Sebo para obtener símbolos reales
3. Ajustar parámetros de optimización según el rendimiento observado
4. Documentar cualquier configuración adicional necesaria