

# Sistema de Investimento em Moedas

## 1. Introdução

### 1.1. Visão Geral do Projeto

O projeto consiste no desenvolvimento de uma aplicação web para substituir e automatizar o processo de análise de mercado atualmente realizado em uma planilha Excel. O sistema permitirá o login de usuários, o cadastro de informações de base (como moedas e feriados) e a entrada de dados ("pontos") em horários predefinidos.

O núcleo do sistema será um motor de cálculo que processa essas entradas para gerar um conjunto de sinais técnicos (ex: "Ciclo Concluído", "Contração", "Acúmulo"). Esses sinais serão exibidos em um dashboard visual, codificado por cores (Força/Fraqueza), que espelha a planilha, servindo como a principal ferramenta de análise do usuário.

### 1.2. Objetivos

- **Centralizar a Entrada de Dados:** Criar um formulário para registrar dados de mercado (pontos) de forma estruturada.
- **Automatizar Cálculos:** Implementar a lógica de cálculo para gerar todos os indicadores e sinais.
- **Visualizar Sinais:** Apresentar os dados calculados em um dashboard web para análise rápida.
- **Garantir Segurança:** Implementar um sistema de login seguro com controle de sessão.

---

## 2. Requisitos Funcionais (RF)

Esta seção detalha as funcionalidades que o sistema deve executar.

### RF 1.0: Gestão de Usuários e Segurança

- **RF 1.1:** O sistema deve ter uma tela de **login** com campos para "usuário" e "senha".
- **RF 1.2:** As senhas dos usuários devem ser armazenadas no banco de dados de forma **criptografada**.
- **RF 1.3:** O sistema deve fornecer uma opção para "Registrar Usuário".
- **RF 1.4:** Um usuário só pode ter **uma sessão ativa** por vez (não permitir logins simultâneos).
- **RF 1.5:** O sistema deve **"logar" (registrar) o endereço IP** de cada tentativa de login bem-sucedida.
- **RF 1.6:** O sistema deve implementar um **timeout de sessão** (deslogar o usuário após um período de inatividade).

### RF 2.0: Cadastros e Configurações (Admin)

- **RF 2.1:** O sistema deve permitir o **CRUD (Create, Read, Update, Delete)** para os itens de cadastro. A exclusão deve ser **lógica** (inativação) e não física.
- **RF 2.2:** Deve haver um módulo para **Cadastro de Moedas**.

- Campos obrigatórios: Abreviação (3 caracteres), Descrição, Região (Mercado), Hora do Mercado.
- **RF 2.3:** Deve haver um módulo para **Cadastro de Feriados**, pois podem impactar os dias de operação.

### RF 3.0: Entrada de Dados

- **RF 3.1:** O sistema deve ter um módulo para "Dar Entrada nos Pontos por Moeda por Dia".
- **RF 3.2:** O sistema deve diferenciar entradas **"Parcial" (até 21:00)** e **"Final" (após 21:00)**.
- **RF 3.3:** O módulo de entrada deve suportar o **timeframe H4**, com 6 entradas diárias nos horários: 00:00, 04:00, 08:00, 12:00, 16:00, 20:00.
- **RF 3.4:** O módulo de entrada deve suportar o **timeframe H1**, com 24 entradas diárias, de 00:00 até 23:00.
- **RF 3.5:** Os dados de entrada devem ser persistidos com data, hora e moeda correspondentes.

### RF 4.0: Motor de Cálculo

- **RF 4.1:** O sistema deve calcular e determinar o status dos seguintes indicadores:
  - a) Ciclo Concluído
  - b) Qual Ciclo Deve
  - c) Qual Contração
  - d) Tem Acúmulo
  - e) Teto/Piso de Acúmulo
  - f) Flutuante
  - g) Flutuante no Sentido do Ciclo que Deve
  - h) Ponto de Parada
  - i) Quebra de Score (definido como "Maior Peso")
  - j) Permissão
  - k) Diferença de Pontos do Dia Anterior
  - l) Benefícios (calculado "pelos pesos")

### RF 5.0: Dashboard de Visualização

- **RF 5.1:** O sistema deve exibir o dashboard principal em formato de **tabela (grade)**, replicando a estrutura da planilha.
- **RF 5.2:** O dashboard deve exibir os dados por **Data, Mercado, Moeda e Time Frame** (Mensal, W1, D1, H4, H1).
- **RF 5.3:** As colunas da tabela devem exibir o resultado de todos os cálculos do **RF 4.0** (ex: "Ciclo Concluído", "Contração", "Acúmulo", "Ponto de Parada 21h", "Permissão", "Quebra de Score", "Benefit").
- **RF 5.4:** O status de cada indicador (coluna) deve ser **codificado por cores**:
  - **Verde:** Indica "Força" (ou estado positivo).
  - **Vermelho:** Indica "Fraqueza" (ou estado negativo).
  - **Amarelo:** Indica "Não Operar" ou atenção.

- **RF 5.5:** Colunas de dados numéricos (Pts, Dif Dia Ant.) e textuais (Região) também devem ser exibidas conforme o layout.

---

### 3. Requisitos Não Funcionais (RNF)

- **RNF 1. (Segurança):** Além da criptografia de senha (RF 1.2), o sistema deve ser protegido contra ataques comuns (como SQL Injection e XSS).
- **RNF 2. (Desempenho):** O dashboard principal deve carregar rapidamente, mesmo com um histórico de dados crescente. Os cálculos podem precisar ser processados em segundo plano (batch) para não atrasar a interface do usuário.
- **RNF 3. (Usabilidade):** A interface de entrada de dados (RF 3.0) deve ser intuitiva e rápida. O dashboard (RF 5.0) deve ser claro, legível e de fácil interpretação visual.
- **RNF 4. (Manutenibilidade):** A lógica de cálculo (RF 4.0) deve ser modularizada para permitir ajustes e calibrações futuras nos "pesos" e regras.

---

### 4. Principais Entidades de Dados

- **Usuário:** (ID, Nome, Email, SenhaCriptografada, DataRegistro)
- **Moeda:** (ID, Abrev, Descricao, Regiao, HoraMercado, Ativo)
- **Feriado:** (ID, Data, Descricao)
- **EntradaH1:** (ID, MoedaID, DataHora, Pontos, UsuarioID)
- **EntradaH4:** (ID, MoedaID, DataHora, Pontos, UsuarioID)
- **SinalCalculado:** (ID, MoedaID, Timeframe, Data, [Campos para cada indicador: CicloConcluido, DonoCiclo, Contracao, Acumulo, ...])
- **LogLogin:** (ID, UsuarioID, IP, DataHora)