

Documento de Regras de Negócio - Grupo 5

Versão: 1.0 | **Data:** 2025-08-23 | **Responsável:** Analista de Negócio
Sistema: Home Broker Simulado (HTML, CSS, JavaScript) | **Base consultada:** Página de Login e fluxos relacionados

Persistência local sem backend; dados salvos em `localStorage`.

Visão Geral

O sistema permite simulação de operações de compra e venda de ativos com cotações dinâmicas, sem backend, persistindo dados no `localStorage`. Sessões são geridas por CPF. Existem contas e usuários padrão, além do fluxo de cadastro de novas contas.

Usuários, Contas e Sessão

Usuários padrão

- Conta A: CPF **111.111.111-11** / Senha **123**
- Conta B: CPF **222.222.222-22** / Senha **456**

Contas padrão

- Conta A: saldo 100.000; carteira inicial { PETR4: 300, VALE3: 200, ITUB4: 100 }.
- Conta B: saldo 10; carteira inicial { MGLU3: 100, BBAS3: 100 }.

Ativos padrão: PETR4, VALE3, ITUB4, BBDC4, ABEV3, MGLU3, BBAS3, LREN3 (com preços base).

Chaves de persistência

- `hb_usuarios`
- `hb_contas`
- `hb_session_cpf`
- `layoutPreferido` (preferência de layout)
- `hb_lembrar_cpf` (lembrar CPF no login)

Login

Acesso por CPF e senha (aceita CPF com máscara ou apenas dígitos).

- **Validações:** CPF com 11 dígitos (máscara em tempo real) e senha com mínimo de 3 caracteres.
- **UX:** Botão “Entrar” habilita apenas quando ambos válidos; tecla Enter dispara login. Mostrar/ocultar senha com alternância de tipo e ícone.
- **Lembrar-me:** Se marcado, armazena o CPF (somente dígitos) em `hb_lembrar_cpf` e pré-preenche na próxima visita.

Sucesso: Salva `hb_session_cpf` e redireciona para `dashboard.html`.

Erro: Exibe “CPF ou senha inválidos.” e reabilita o botão.

Sessão

- Em telas internas, se sessão ausente, redireciona para `login.html`.
- Se sessão inválida (usuário não encontrado), limpa sessão e redireciona para `index.html`.

Recuperação de Senha

Modal “Recuperar senha” com campos CPF e e-mail.

- **Validações:** CPF com 11 dígitos; e-mail válido.
- **Ação:** Exibe a mensagem “Se o CPF/email existirem, enviaremos instruções...” (comportamento informativo/simulador).

Cadastro de Nova Conta

Campos: Nome, CPF, WhatsApp, E-mail, Senha, Confirmar Senha.

Validações

- **Nome:** apenas letras e espaços, mínimo de 2 letras; números e símbolos são bloqueados em tempo real.
- **CPF:** 11 dígitos, com dígitos verificadores válidos; rejeita sequências (ex.: 000.../111...); **bloqueia CPF já cadastrado** (checagem em tempo real e no envio).
- **WhatsApp:** exatamente 11 dígitos numéricos.
- **E-mail:** formato válido e **único** (bloqueia e-mail já cadastrado em tempo real e no envio).

- **Senha:** mínimo de 8 caracteres; requer combinação de maiúsculas, minúsculas e número ou símbolo; bloqueia termos triviais (ex.: “Senha”, “123456”, “password”).
- **Confirmação:** senhas devem coincidir.
- **UX de erro:** mensagens coerentes por campo e **destaque visual** com borda vermelha nos campos inválidos.

Efeitos

- Cria conta com saldo inicial de 50.000 e carteira vazia `{}`.
- Persiste em `hb_contas` e `hb_usuarios`.
- Exibe mensagem de sucesso e limpa os campos.

Observação: O login aceita CPF com e sem máscara; o cadastro normaliza e armazena o CPF numericamente (sem máscara) como chave principal, prevenindo duplicidades mesmo que digitado com máscara.

Logout

Ao clicar em “Sair”, remove `hb_session_cpf` e redireciona para `index.html`.

Cotação de Ativos

- Cotações simuladas atualizadas a cada 10 segundos.
 - Variação aleatória pequena por ativo; preço arredondado para 2 casas e nunca abaixo de 0,01.
 - Histórico por até 24h, com semente inicial (~90 minutos, passo de 30s) para os gráficos.
-

Boleta de Compra e Venda

Campos: Tipo (Compra/Venda), Ativo, Quantidade, Valor por unidade.

Validações

- **Quantidade:** número inteiro > 0 e múltiplo de 100.
- **Valor:** numérico > 0 .
- **Compra:** exige saldo suficiente para quantidade \times valor.
- **Venda:** exige possuir a quantidade do ativo em carteira.

Regras de aceitação/execução

- Se `valor == cotação` : Executada imediatamente.
- Se `|valor - cotação| ≤ 5` : Aceita (pendente até reavaliação).

- Se $|\text{valor} - \text{cotação}| > 5$: Rejeitada (registrada com status Rejeitada).

Mensagens de retorno são exibidas em tela.

Atualização de Carteira e Saldo

- Compra executada: diminui saldo em quantidade \times valor e adiciona a quantidade do ativo.
 - Venda executada: aumenta saldo em quantidade \times valor e subtrai a quantidade do ativo.
 - Se a quantidade de um ativo zera, ele é removido da carteira.
 - Carteira e saldo são refletidos imediatamente na UI após execuções.
-

Livro de Ofertas (Book de Cotações)

- Tabela com os ativos e preços atuais.
 - Atualizada automaticamente a cada 10 segundos (mesmo ritmo das cotações).
-

Book de Ordens

Registra todas as ordens com: Tipo, Ativo, Quantidade, Valor, Cotação atual, Total, Status, Data/Hora.

Ação “Cancelar” disponível apenas para ordens com status “Aceita”.

Status possíveis

- **Aceita:** válida, aguardando cotação.
 - **Executada:** executada com sucesso.
 - **Rejeitada:** diferença de preço maior que R\$ 5.
 - **Cancelada:** cancelada manualmente.
-

Cancelamento de Ordem

- Apenas ordens “Aceita” podem ser canceladas.
- Ao cancelar: muda para “Cancelada”; não altera carteira nem saldo.
- Book é atualizado imediatamente.

Verificação Automática das Ordens Aceitas

- A cada 10 segundos, reavalia ordens “Aceita” com a cotação corrente.
 - **Disparo de execução:** Compra quando cotação \leq valor; Venda quando cotação \geq valor.
 - Ao executar: aplica efeitos em carteira/saldo, atualiza status e data/hora, insere no Extrato e atualiza as telas (Carteira, Book, Ordens, Extrato, Gráficos).
-

Extrato de Operações

Exibe apenas ordens Executadas.

Campos: Data/Hora, Tipo, Ativo, Quantidade, Valor Total (R\$).

Exportação: CSV com cabeçalho padrão; nome do arquivo com o nome da conta.

Alertas de Preço (Dashboard)

Configuração via checkbox “alerta de preço” e campo “preço alvo”.

- Para ativar, o preço alvo deve ser numérico > 0 .
 - Após execução de ordem, alerta quando: Compra ($\text{preçoAtual} \leq \text{preçoAlvo}$), Venda ($\text{preçoAtual} \geq \text{preçoAlvo}$).
 - Alerta exibido via `alert()` do navegador.
-

Preferências de Layout

- Opções: *default*, *light*, *dark*.
 - Persistência em `layoutPreferido` ; efeitos visuais imediatos (classe `dark-mode` e estilos).
-

Gráficos de Cotação

- **Principal:** preferência por Chart.js (linha de fechamento). Fallback Canvas 2D simples (candles e linha).
- Seleção de ativo para o gráfico (`ativoGrafico`).
- Resolução configurável (`resolucaoGrafico`) agregando OHLC por janela.

- Histórico: semente inicial (~90 minutos, passo 30s); atualização a cada 10s; responsivo ao *resize*.
-

Análise de Operações (Dashboard)

Modal exibido quando há ao menos uma operação executada.

Gráfico de barras (Chart.js) resumizando Quantidade Operada por Ativo com base no Extrato.

Alteração de Senha (Dashboard)

- Campo “Nova Senha”; validação de mínimo 3 caracteres.
 - Ao confirmar: atualiza a senha do usuário corrente em `hb_usuarios`, limpa o campo e exibe sucesso.
-

Mensageria e UX

- Estados de carregamento no login (botão “Entrar” com classe `loading` durante validação simulada).
 - Mensagens de erro/sucesso em validações de login, cadastro, boleto, cancelamento, alteração de senha e exportações.
 - Acessibilidade/atalhos: tecla Enter aciona login quando válido.
-

Regras Técnicas e Limitações

- Não há backend; todo armazenamento é `localStorage`.
 - A sessão é baseada na chave `hb_session_cpf` e na existência do usuário em `hb_usuarios`.
 - As cotações são exclusivamente simuladas e não refletem o mercado real.
 - O filtro “ordens do dia” considera o timestamp salvo na ordem.
-

Segurança

- Redirecionamento para login quando não há sessão válida.
- Limpeza de sessão se o usuário associado ao CPF não existir mais.
- Dados sensíveis (senha) em `localStorage` são apenas para fins didáticos (não recomendado em produção).

Tabelas e Campos-Chave

| Entidade | Estrutura |
|----------|--|
| Usuário | hb_usuarios[cpf] : { senha, conta, nome?, whatsapp?, email? } |
| Conta | hb_contas[id] : { nome, saldo, carteira: { [ticker]: quantidade } } |
| Ordem | { id, timestamp, dataHora, tipo, ativo, qtd, valor, cotacao, total, status } |
| Extrato | Lista de ordens Executadas com os campos acima |

Navegação

- `index.html` : landing; CTAs para login e cadastro.
- `login.html` : autenticação, “lembrar-me”, “recuperar senha”, “cadastrar”.
- `dashboard.html` : carteira, boleta, book, ordens, extrato, gráficos, análise, preferências, alertas e alteração de senha.
- “Sair”: encerra sessão e volta ao `index.html` .

Regras de Aceite de Qualidade (QA)

- **Login:** aceita CPF com e sem máscara; valida 11 dígitos e senha ≥ 3 .
- **Cadastro:** bloqueia CPF já cadastrado; exige e-mail válido e único; senhas iguais; nome somente letras/espços; WhatsApp 11 dígitos; senha forte (≥ 8 , com maiúsculas/minúsculas e número ou símbolo; sem termos triviais); erro destacado por campo.
- **Boleta:** quantidade múltipla de 100 e > 0 ; valor > 0 ; compra bloqueada sem saldo; venda bloqueada sem posição.
- **Cotações (10s):** nunca negativas; 2 casas; reavalia ordens “Aceita” e atualiza carteira/saldo/extrato.
- **Exportações:** extrato CSV com apenas Executadas; ordens do dia (JSON/XLSX) incluem todas com status.
- **Preferências/Alertas:** layout persiste; alerta requer preço alvo válido e dispara conforme tipo.

Alterações desta versão

- Validação de **Nome** (apenas letras e espaços) com sanitização em tempo real.
- Destaque visual de campos inválidos com borda vermelha e mensagens específicas por campo.
- Validação de **CPF** com dígitos verificadores e bloqueio de CPF já cadastrado (tempo real e envio).
- Validação de **WhatsApp** estrita: 11 dígitos.
- **E-mail** único: verificação de duplicidade em tempo real e no envio; correção de validação.
- **Senha** forte: ≥ 8 , combinação de maiúsculas/minúsculas e número ou símbolo; bloqueio de termos triviais; conferência de confirmação.
- Correção de função `validateEmail` duplicada.

Documento gerado para publicação em GitHub Pages.