

Finanza

Sistema Completo

Documentação Técnica Completa

Fluxogramas, Diagramas e Arquitetura

Componentes:

- Aplicativo Mobile (Android)
- Servidor (Java + MySQL)
- Desktop Admin (Java Swing)
- Banco de Dados (MySQL/SQLite)

Índice

1. Arquitetura do Sistema

Visão geral da arquitetura e componentes

2. Fluxo do Aplicativo Mobile

Todas as telas e navegação do app Android

3. Protocolo do Servidor

Comandos e processamento do servidor

4. Aplicativo Desktop Admin

Interface de administração desktop

5. Esquema do Banco de Dados

Tabelas, relacionamentos e views

6. Sequência: Login

Fluxo detalhado do processo de login

7. Sequência: Transação

Fluxo de adição de transação

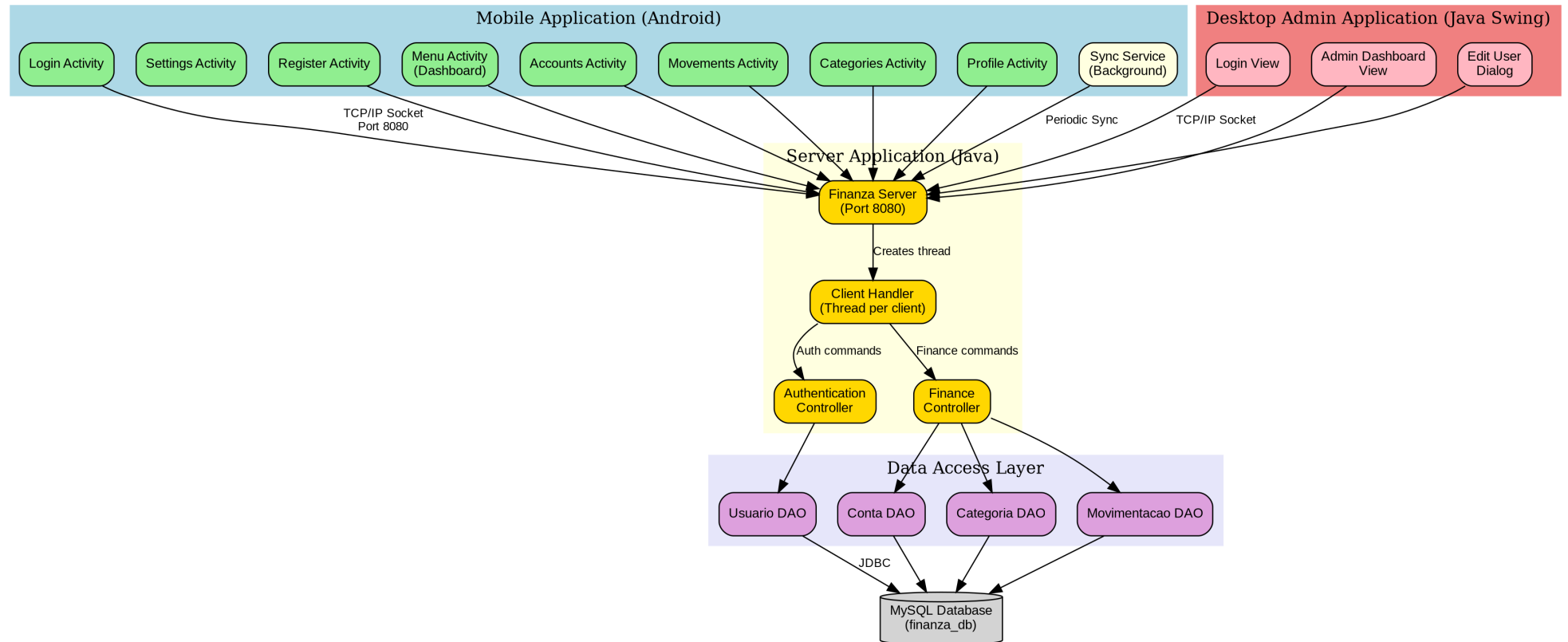
8. Fluxo de Sincronização

Sincronização entre dispositivos

1. Arquitetura do Sistema

Este diagrama mostra a arquitetura completa do sistema Finanza, incluindo:

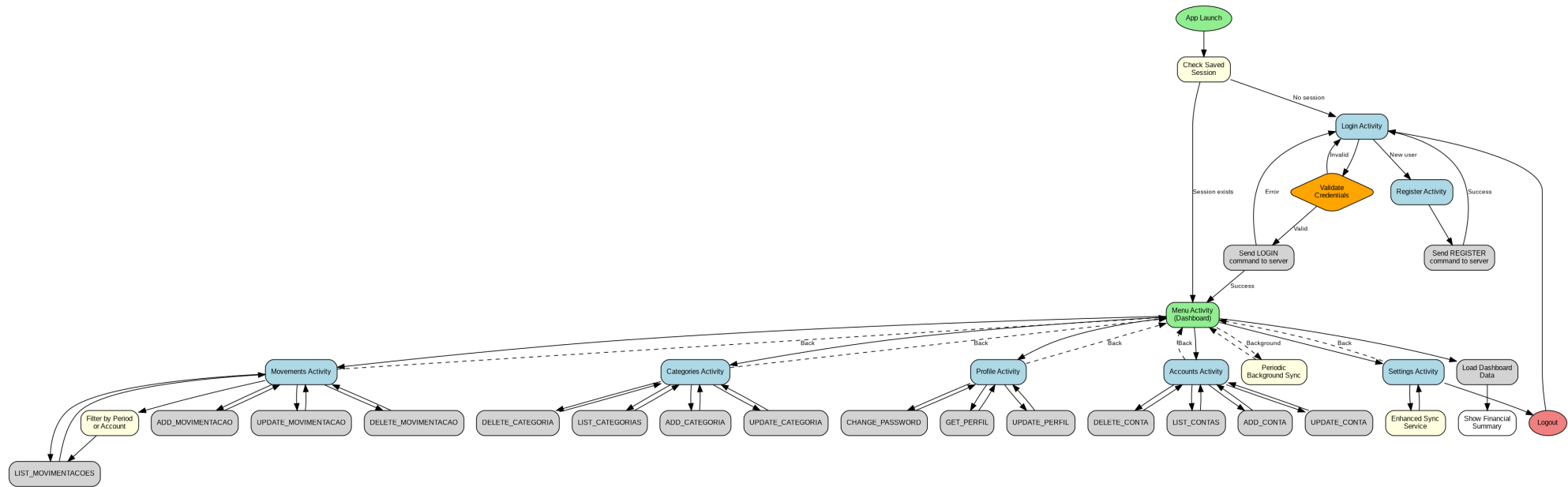
- Aplicativo Mobile Android com todas as activities
- Desktop Admin para gerenciamento de usuários
- Servidor Java com ClientHandler multi-thread
- Camada DAO para acesso ao banco
- Banco de dados MySQL
- Comunicação via Socket TCP/IP na porta 8080



2. Fluxo do Aplicativo Mobile

Fluxograma completo do aplicativo Android, mostrando:

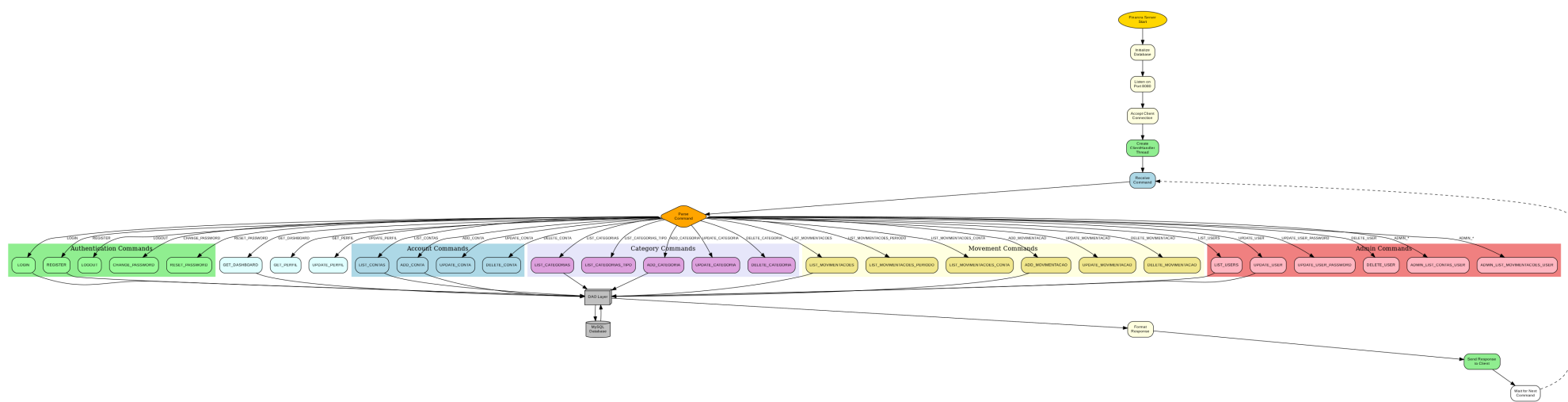
- Processo de autenticação (login/registro)
- Dashboard com resumo financeiro
- Gestão de contas (adicionar, editar, excluir)
- Gestão de movimentações (receitas e despesas)
- Gestão de categorias personalizadas
- Perfil do usuário e configurações
- Serviço de sincronização em background



3. Protocolo do Servidor

Diagrama detalhado do protocolo de comunicação servidor, incluindo:

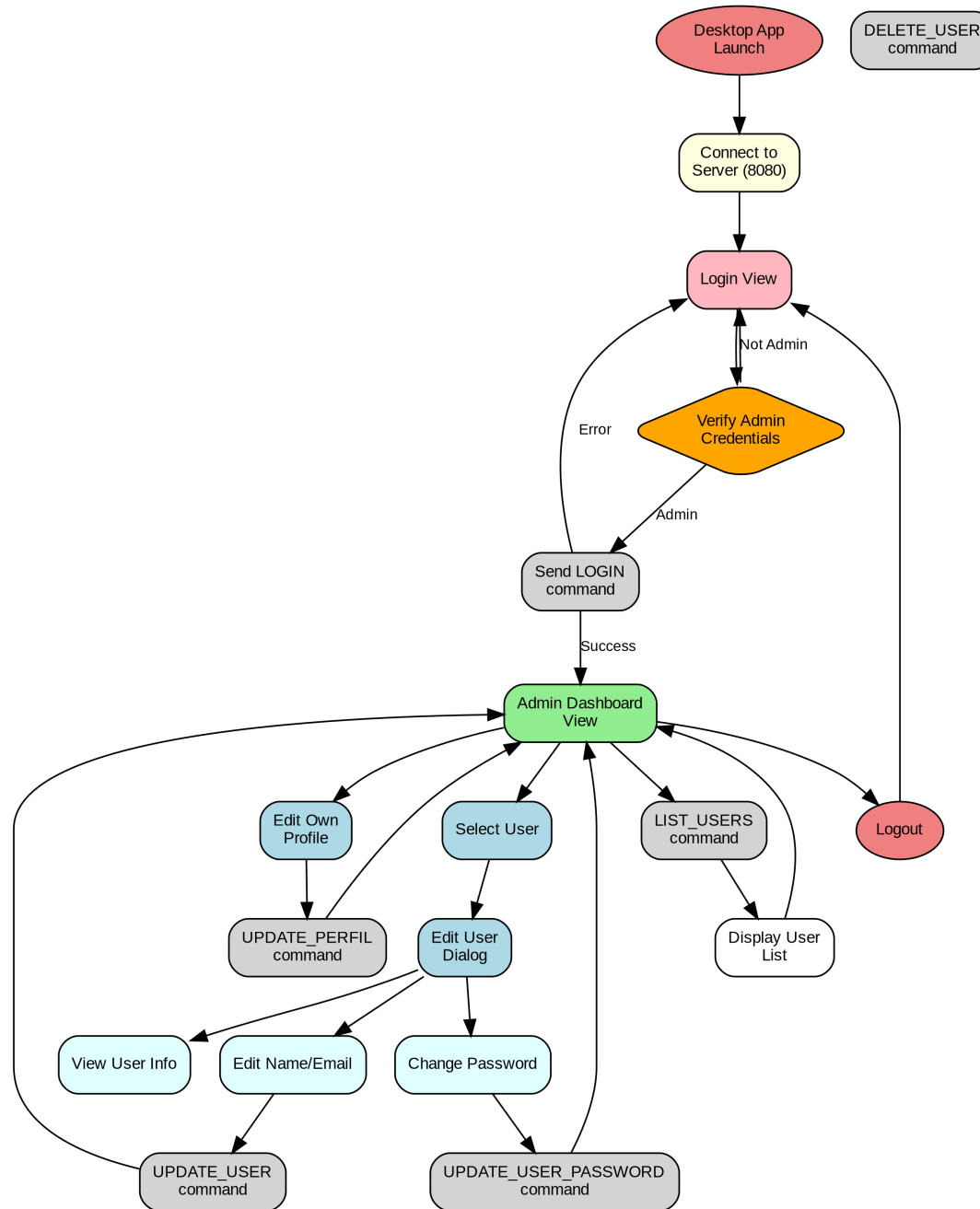
- Mais de 40 comandos suportados
- Comandos de autenticação (LOGIN, REGISTER, LOGOUT)
- Comandos de contas (LIST, ADD, UPDATE, DELETE)
- Comandos de categorias e movimentações
- Comandos administrativos (gestão de usuários)
- Camada DAO e acesso ao banco de dados
- Formato de requisições e respostas



4. Aplicativo Desktop Admin

Fluxo da aplicação desktop administrativa, mostrando:

- Login exclusivo para administradores
- Dashboard com lista de todos os usuários
- Edição de dados de usuários (nome, email)
- Alteração de senhas de usuários
- Visualização de informações
- Edição do próprio perfil do admin
- Comandos do protocolo utilizados



5. Esquema do Banco de Dados

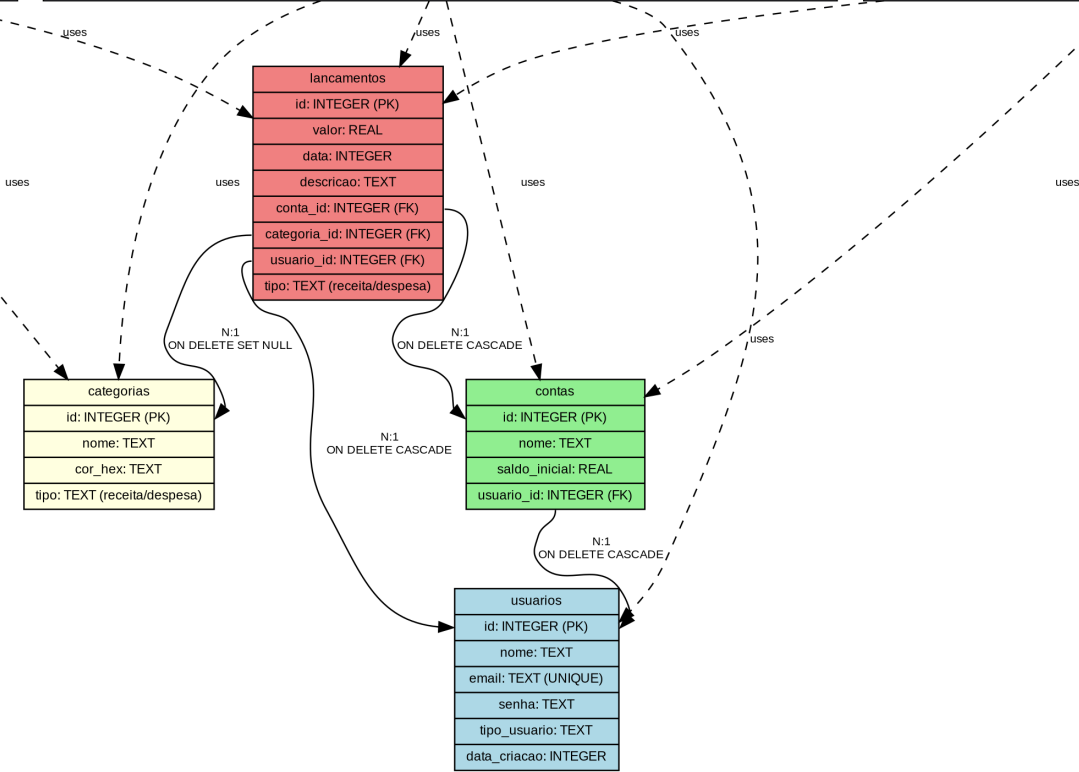
Diagrama ER (Entidade-Relacionamento) completo:

- Tabela usuarios (id, nome, email, senha, tipo_usuario)
- Tabela contas (id, nome, saldo_inicial, usuario_id)
- Tabela categorias (id, nome, cor_hex, tipo)
- Tabela lancamentos (id, valor, data, descricao, ...)
- Views: v_saldo_contas, v_resumo_categorias, v_lancamentos_detalhados
- Relacionamentos e chaves estrangeiras
- Triggers de validação

{<table>v_resumo_categorias (VIEW)|id|nome|cor_hex|tipo|total_lancamentos|total_valor}

{<table>v_lancamentos_detalhados (VIEW)|id|valor|data|descricao|tipo|conta_nome|categoria_nome|categoria_cor|usuario_nome}

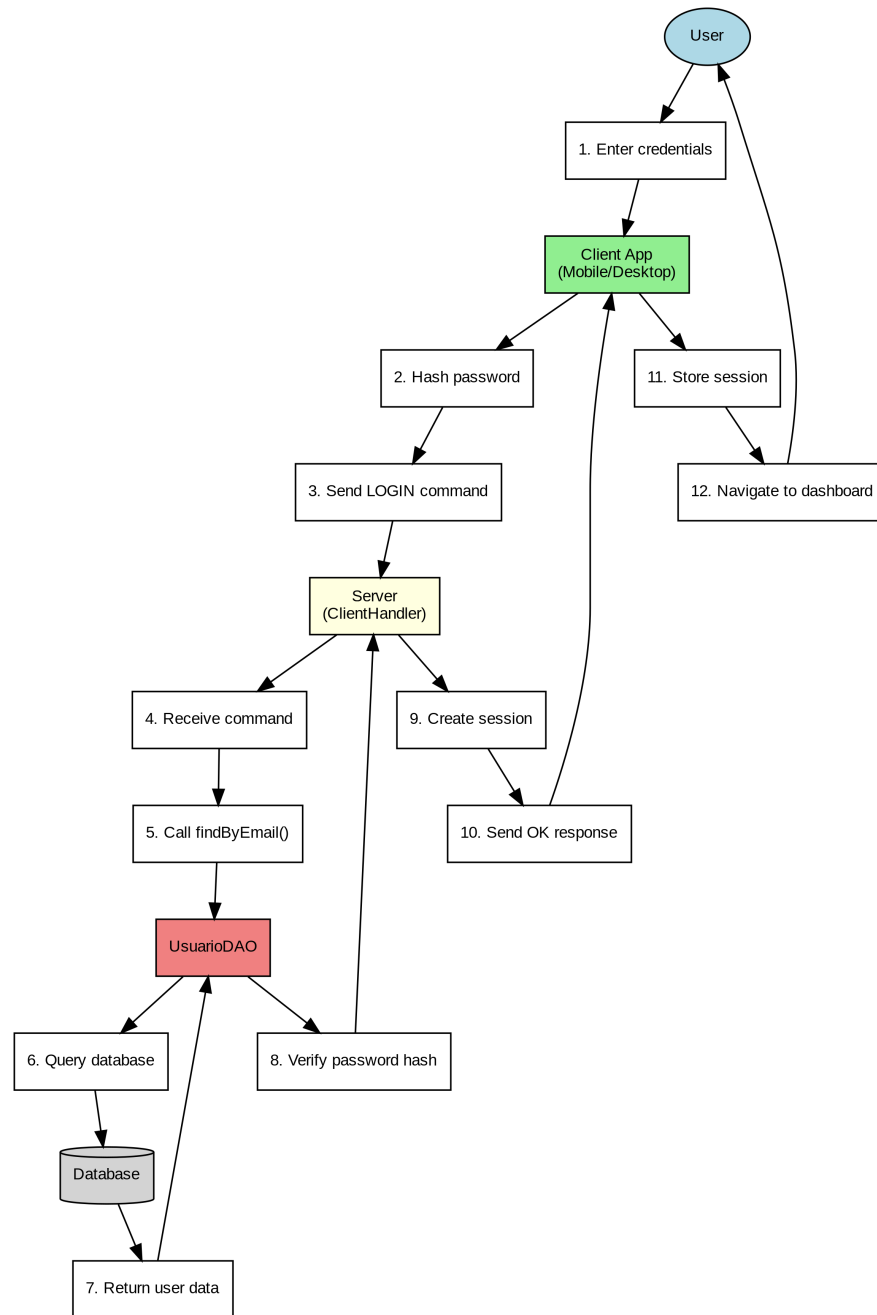
{<table>v_saldo_contas (VIEW)|id|nome|saldo_inicial|usuario_id|movimentacao|saldo_atual}



6. Sequência: Processo de Login

Diagrama de sequência detalhado do processo de login:

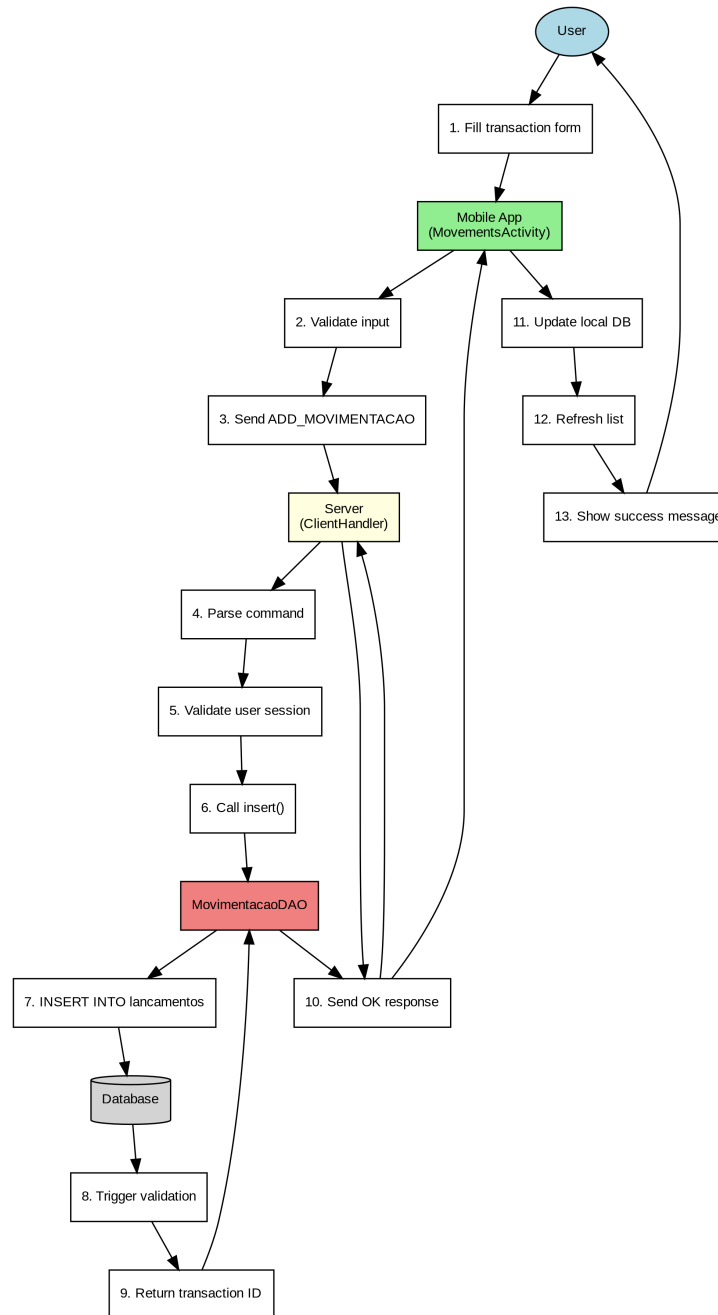
1. Usuário insere credenciais na interface
2. Cliente faz hash da senha
3. Envia comando LOGIN para o servidor
4. Servidor valida via UsuarioDAO
5. Consulta banco de dados
6. Verifica hash da senha
7. Cria sessão do usuário
8. Retorna resposta OK com dados do usuário
9. Cliente armazena sessão
10. Navega para dashboard



7. Sequência: Adicionar Transação

Diagrama de sequência para adicionar uma transação:

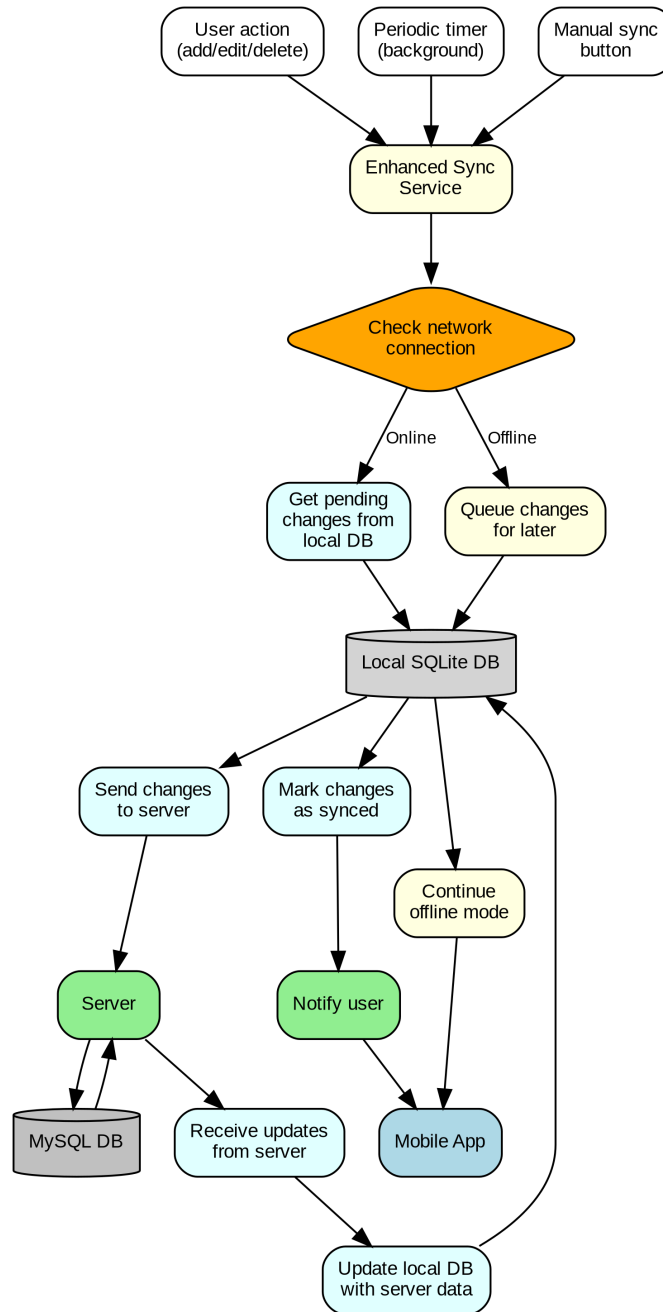
1. Usuário preenche formulário de transação
2. App valida entrada localmente
3. Envia ADD_MOVIMENTACAO para servidor
4. Servidor parseia comando
5. Valida sessão do usuário
6. MovimentacaoDAO insere no banco
7. Trigger valida dados (valor não zero, tipo correto)
8. Retorna ID da transação
9. Servidor envia OK para cliente
10. Cliente atualiza banco local
11. Atualiza lista de transações
12. Exibe mensagem de sucesso



8. Fluxo de Sincronização

Diagrama do processo de sincronização de dados:

- Gatilhos: ação do usuário, timer periódico, sync manual
- Verifica conexão de rede
- Modo online: sincroniza mudanças pendentes
- Modo offline: enfileira mudanças para depois
- Envia alterações locais para servidor
- Recebe atualizações do servidor
- Atualiza banco de dados local
- Marca mudanças como sincronizadas
- Notifica usuário do resultado



Resumo Técnico

Tecnologias Utilizadas:

- Java (JDK 17+) - Servidor e Desktop
- Android (Java + XML) - App Mobile
- MySQL - Banco de dados principal
- SQLite - Banco de dados local (mobile)
- Sockets TCP/IP - Comunicação cliente-servidor
 - Java Swing - Interface desktop

Funcionalidades Principais:

- Autenticação segura com criptografia
- Gestão completa de finanças pessoais
 - Sincronização em tempo real
 - Suporte offline (mobile)
- Administração centralizada (desktop)
 - Relatórios financeiros

Multiusuário com isolamento de dados

© 2024 - Finanza System - Todos os direitos reservados

Arquitetura:

- Cliente-Servidor TCP/IP