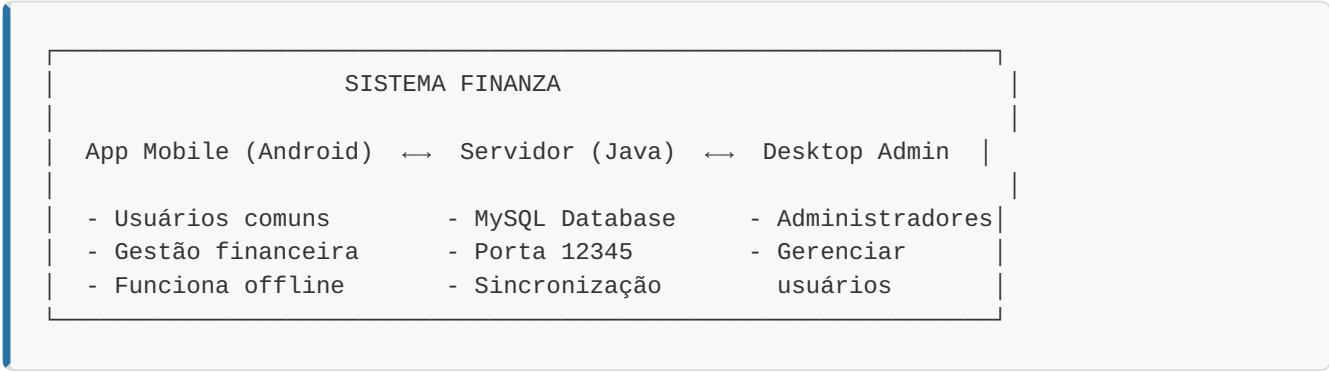
📊 FLUXOGRAMA SIMPLIFICADO

Sistema de Controle Financeiro Finanza

IFSUL - Campus Venâncio Aires

📊 FLUXOGRAMA SIMPLIFICADO - Sistema Finanza

🎯 Visão Geral do Sistema

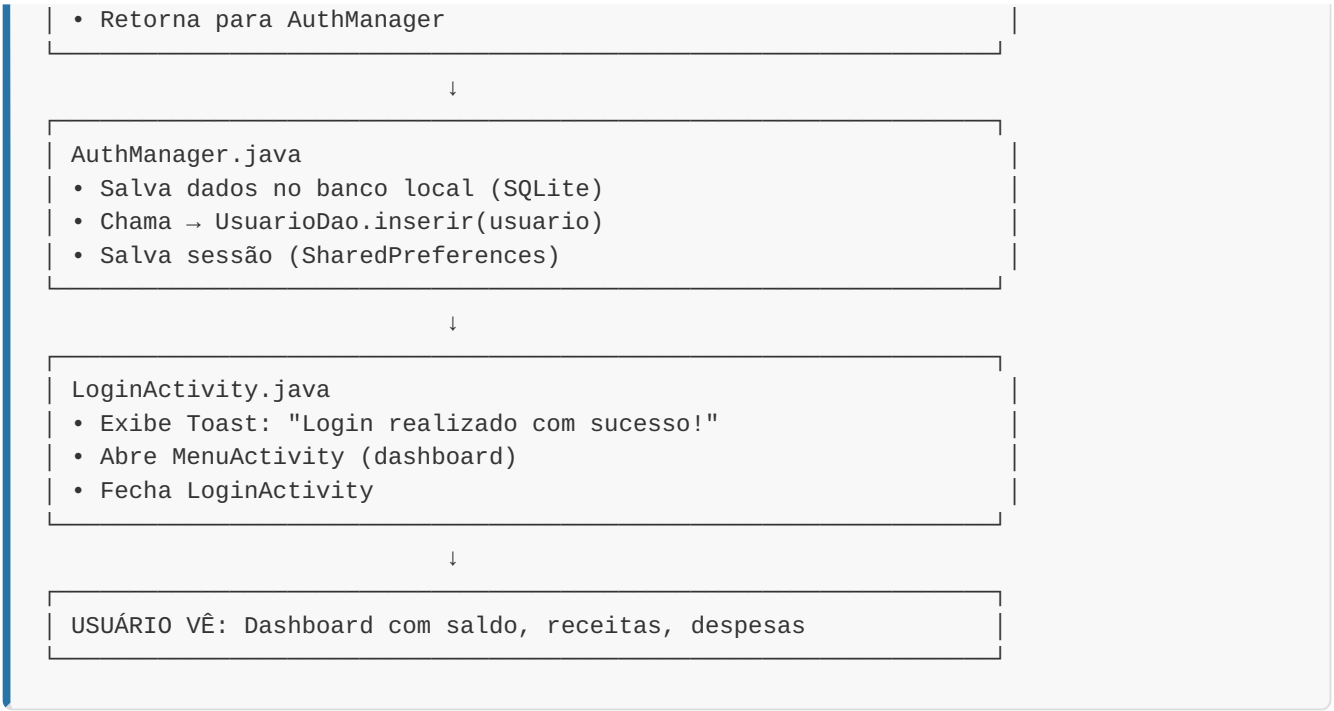


📱 MOBILE (Android)

Estrutura de Arquivos

```
app/src/main/java/com/example/finanza/
│
├── ui/                    # TELAS (8 arquivos)
│   ├── LoginActivity         → Tela de login
│   ├── RegisterActivity      → Cadastro de usuário
│   ├── MenuActivity          → Dashboard principal
│   ├── AccountsActivity      → Gerenciar contas
│   ├── CategoriaActivity     → Gerenciar categorias
│   ├── MovementsActivity     → Lançamentos financeiros
│   ├── ProfileActivity       → Perfil do usuário
│   └── SettingsActivity      → Configurações
│
├── model/                 # DADOS (4 arquivos)
│   ├── Usuario               → id, nome, email, senha
│   ├── Conta                 → id, nome, tipo, saldo
│   ├── Categoria             → id, nome, tipo (receita/despesa)
│   └── Lancamento            → id, valor, data, descrição
│
├── db/                    # BANCO LOCAL SQLite (5 arquivos)
│   ├── AppDatabase           → Configuração do banco
│   ├── UsuarioDao            → Buscar/Salvar usuários
│   ├── ContaDao              → Buscar/Salvar contas
│   ├── CategoriaDao          → Buscar/Salvar categorias
│   └── LancamentoDao         → Buscar/Salvar lançamentos
│
└── network/               # COMUNICAÇÃO (6 arquivos)
    ├── ServerClient          → Conecta ao servidor (porta 12345)
    ├── Protocol              → Define comandos (LOGIN, ADD_CONTA, etc)
    ├── AuthManager           → Gerencia login/sessão
    ├── SyncService           → Sincroniza dados
    ├── EnhancedSyncService   → Sincronização avançada
    └── ConflictResolutionManager → Resolve conflitos
```

🔄 FLUXO 1: Login no Mobile



Tempo total: 1-2 segundos

Arquivos envolvidos: - Mobile: LoginActivity → AuthManager → ServerClient → UsuarioDao → MenuActivity - Servidor: ClientHandler → UsuarioDAO → MySQL

🔄 FLUXO 2: Adicionar uma Despesa

Tempo total: 1-3 segundos

Banco de dados modificado: - SQLite (Mobile): 2 tabelas (lancamento, conta) - MySQL (Servidor): 2 tabelas (movimentacao, conta)

🔄 FLUXO 3: Dashboard (MenuActivity)

```
│ │ [Contas] [Categorias] [Lançamentos]│                          │
│ │ [Perfil] [Configurações]           │                          │
│ └────────────────────────────────────┘                          │
└─────────────────────────────────────────────────────────────────┘
```

Queries executadas: 1. SELECT * FROM conta WHERE usuarioId = 1 2. SELECT SUM(valor) FROM lancamento WHERE tipo='receita' AND usuarioId=1 3. SELECT SUM(valor) FROM lancamento WHERE tipo='despesa' AND usuarioId=1

🖥️ SERVIDOR (Java)

Estrutura de Arquivos

```
DESKTOP VERSION/ServidorFinanza/src/
│
├── MainServidor.java         # Inicia o servidor
│
├── server/
│   ├── FinanzaServer.java    # Servidor TCP (porta 12345)
│   ├── ClientHandler.java    # Processa cada cliente
│   └── Protocol.java         # Define comandos
│
├── dao/                      # Acesso ao MySQL
│   ├── UsuarioDAO.java
│   ├── ContaDAO.java
│   ├── CategoriaDAO.java
│   └── MovimentacaoDAO.java
│
├── model/
│   ├── Usuario.java
│   ├── Conta.java
│   ├── Categoria.java
│   └── Movimentacao.java
│
└── util/
    ├── DatabaseUtil.java     # Conexão MySQL
    └── SecurityUtil.java     # Criptografia SHA-256
```

Fluxo do Servidor

```
│ • Formata resposta                                                  │
│ • Retorna: "OK|dados" ou "ERROR|mensagem"                           │
└─────────────────────────────────────────────────────────────────────┘
```

💻 DESKTOP ADMIN (Java Swing)

Estrutura de Arquivos

```
DESKTOP VERSION/ClienteFinanza/src/
│
├── MainCliente.java          # Inicia a aplicação
│
├── view/                     # Interface Swing
│   ├── LoginView.java        # Tela de login
│   ├── AdminDashboardView.java  # Tabela de usuários
│   └── EditarUsuarioDialog.java # Editar usuário
│
├── controller/
│   ├── AuthController.java   # Controle de login
│   └── FinanceController.java # Controle de operações
│
├── model/
│   ├── Usuario.java
│   ├── Conta.java
│   ├── Categoria.java
│   └── Movimentacao.java
│
└── util/
    └── NetworkClient.java    # Cliente TCP para servidor
```

Fluxo Desktop Admin

```
┌──────────────────────────────────────────────────────────────┐
│ AdminDashboardView.carregarUsuarios()                        │
│ • Chama: FinanceController.listarUsuarios()                  │
│ • Recebe lista de usuários do servidor                       │
│ • Preenche JTable                                            │
└──────────────────────────────────────────────────────────────┘
                              ↓
┌──────────────────────────────────────────────────────────────┐
│ ADMIN VÊ: Dashboard                                          │
│ ┌──────────────────────────────────────────────────┐         │
│ │ 👥 Gerenciamento de Usuários                     │         │
│ ├──────────────────────────────────────────────────┤         │
│ │ ID │ Nome      │ Email              │ Tipo       │         │
│ ├────┼───────────┼────────────────────┼────────────┤         │
│ │ 1  │ Admin     │ admin@finanza.com  │ Administrador │      │
│ │ 2  │ João      │ joao@gmail.com     │ Usuário    │         │
│ │ 3  │ Maria     │ maria@gmail.com    │ Usuário    │         │
│ └────┴───────────┴────────────────────┴────────────┘         │
│ [Editar] [Excluir] [Novo]                                    │
└──────────────────────────────────────────────────────────────┘
```

📡 PROTOCOLO DE COMUNICAÇÃO

Comandos Principais

Comando	Formato	Resposta	Onde Processa
LOGIN	LOGIN|email|senha_hash	OK|dados_usuario	ClientHandler.processarLogin()
REGISTER	REGISTER|nome|email|senha|uuid	OK|id	ClientHandler.processarRegistro()
ADD_CONTA	ADD_CONTA|nome|tipo|saldo|userId	OK|id_conta	ClientHandler.processarAddConta()
LIST_CONTAS	LIST_CONTAS|userId	OK|lista	ClientHandler.processarListContas()
ADD_MOVIMENTACAO	ADD_MOVIMENTACAO|valor|data|desc|...	OK|id	ClientHandler.processarAddMovimentacao()
LIST_MOVIMENTACOES	LIST_MOVIMENTACOES|userId	OK|lista	ClientHandler.processarListMovimentacoes()

Formato de Mensagens

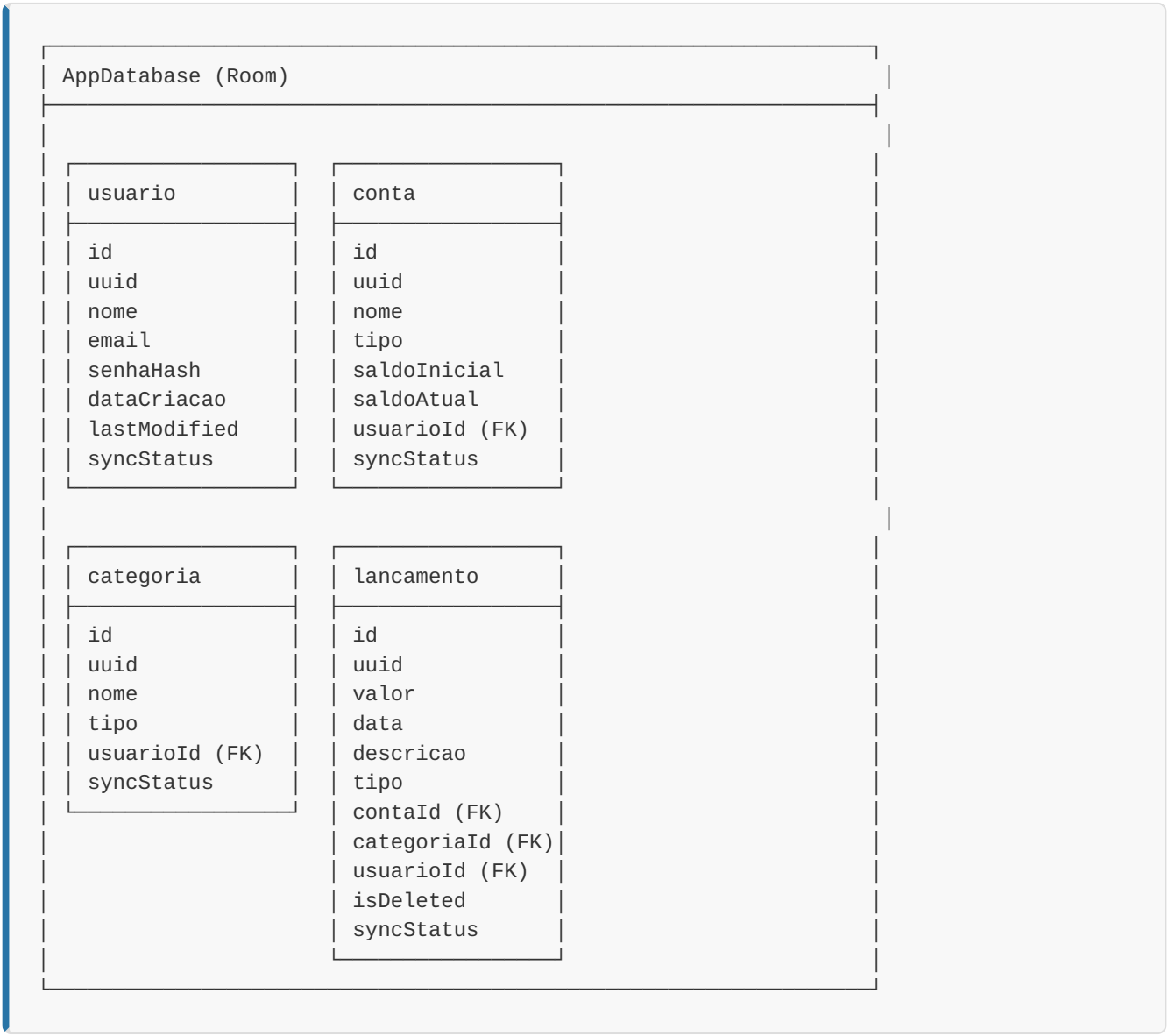
```
ENVIO:    COMANDO|PARAM1|PARAM2|PARAM3
RESPOSTA: OK|dados ou ERROR|mensagem

Exemplo:
Envio:    LOGIN|joao@gmail.com|abc123hash
Resposta: OK|{"id":2,"nome":"João","email":"joao@gmail.com"}

Envio:    ADD_CONTA|Nubank|corrente|500.00|2
Resposta: OK|15
```

🗄️ BANCO DE DADOS

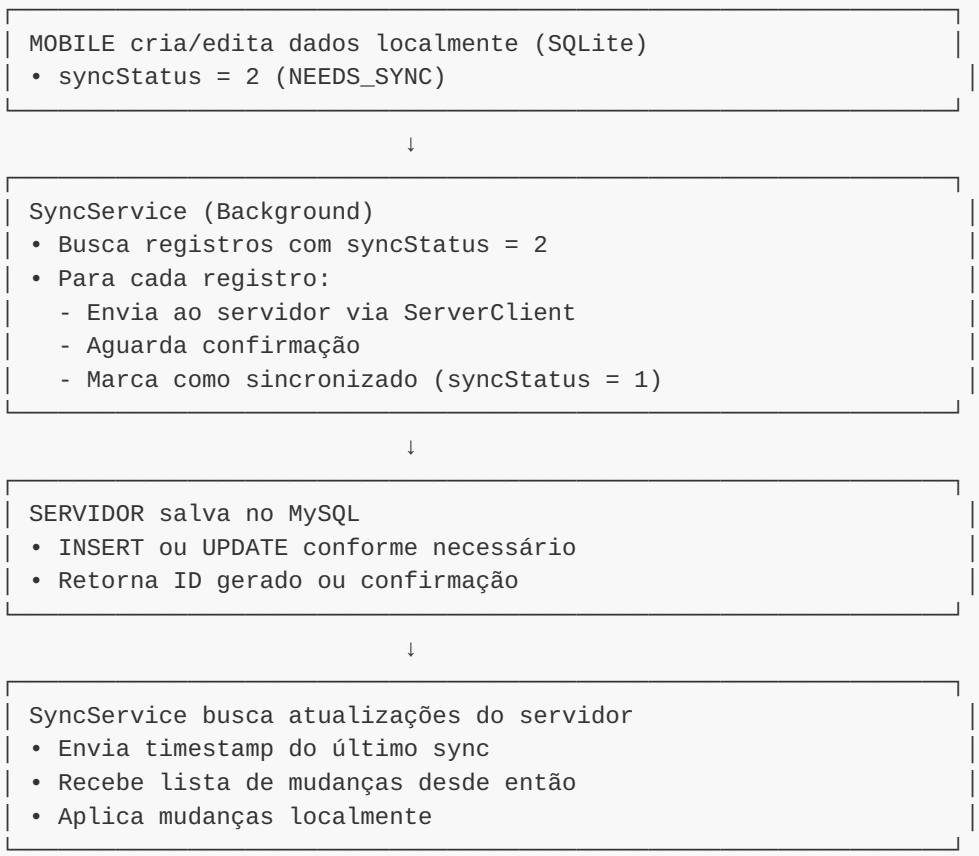
Mobile (SQLite)



Servidor (MySQL)

🔄 SINCRONIZAÇÃO

Como Funciona



Estados de Sincronização

syncStatus	Significado	Cor
0	LOCAL_ONLY - Apenas local	⚪ Branco
1	SYNCED - Sincronizado	✅ Verde
2	NEEDS_SYNC - Precisa sincronizar	🔄 Amarelo
3	CONFLICT - Conflito detectado	⚠️ Laranja
4	DELETED - Deletado	🗑️ Cinza

📊 RESUMO DE ARQUIVOS POR FUNÇÃO

Quando o usuário FAZ LOGIN:

Mobile: 1. LoginActivity.java → Exibe formulário 2. AuthManager.java → Criptografa e valida 3. ServerClient.java → Envia ao servidor 4. UsuarioDao.java → Salva localmente após sucesso 5. MenuActivity.java → Abre dashboard

Servidor: 1. ClientHandler.java → Recebe comando LOGIN 2. UsuarioDAO.java → Busca no MySQL 3. SecurityUtil.java → Valida senha 4. Retorna dados do usuário

---

Quando o usuário ADICIONA DESPESA:

Mobile: 1. MovementsActivity.java → Exibe formulário 2. DataIntegrityValidator.java → Valida dados 3. LancamentoDao.java → Insere no SQLite 4. ContaDao.java → Atualiza saldo 5. EnhancedSyncService.java → Sincroniza com servidor 6. ServerClient.java → Envia dados

Servidor: 1. ClientHandler.java → Recebe comando ADD_MOVIMENTACAO 2. MovimentacaoDAO.java → Insere no MySQL 3. ContaDAO.java → Atualiza saldo no servidor 4. Retorna ID gerado

---

Quando ADMIN EDITA USUÁRIO:

Desktop: 1. AdminDashboardView.java → Lista usuários 2. EditarUsuarioDialog.java → Formulário de edição 3. FinanceController.java → Processa alterações 4. NetworkClient.java → Envia ao servidor

Servidor: 1. ClientHandler.java → Recebe comando UPDATE_USER 2. UsuarioDAO.java → UPDATE no MySQL 3. Retorna confirmação

---

🎯 CONCLUSÃO

Principais Arquivos por Camada

INTERFACE (o que o usuário vê): - Mobile: LoginActivity, MenuActivity, MovementsActivity, AccountsActivity - Desktop: LoginView, AdminDashboardView, EditarUsuarioDialog

LÓGICA (processamento): - Mobile: AuthManager, SyncService, EnhancedSyncService, DataIntegrityValidator - Desktop: AuthController, FinanceController - Servidor: ClientHandler, Protocol

DADOS (banco de dados): - Mobile: UsuarioDao, ContaDao, CategoriaDao, LancamentoDao (SQLite) - Servidor: UsuarioDAO, ContaDAO, CategoriaDAO, MovimentacaoDAO (MySQL)

COMUNICAÇÃO (rede): - Mobile: ServerClient, Protocol - Desktop: NetworkClient - Servidor: FinanzaServer, ClientHandler

Fluxo Geral de Dados

```
USUÁRIO INTERAGE
       ↓
  ACTIVITY (UI)
       ↓
  MANAGER/VALIDATOR (Lógica)
       ↓
  DAO LOCAL (SQLite)
       ↓
  SYNC SERVICE (Sincronização)
       ↓
  SERVER CLIENT (Rede)
       ↓
  INTERNET
       ↓
  CLIENT HANDLER (Servidor)
       ↓
  DAO SERVIDOR (MySQL)
       ↓
  RESPOSTA VOLTA
```

Documento criado para: Sistema Finanza
Objetivo: Entender rapidamente o fluxo do software
Autor: Kalleby Schultz - IFSUL Campus Venâncio Aires
Data: 2024