

5 – APLICATIVO ANDROID PARA GERENCIAMENTO DE ITENS

O presente trabalho foi desenvolvido na disciplina de Programação para Dispositivos Móveis, durante o segundo semestre de 2025, sob a orientação do Professor Dr. Thiago Seti. A proposta consistiu em desenvolver um aplicativo móvel capaz de realizar cadastro, autenticação e gerenciar registros de itens com persistência local de dados, de modo a aplicar conceitos fundamentais para aplicação mobile e banco de dados embarcado.

Para a implementação da lógica do sistema, foi utilizada a linguagem Java, sendo desenvolvido na IDE oficial da plataforma Android, o Android Studio. Para armazenar as informações, adotou-se o sistema de banco de dados relacional embarcado SQLite, integrado nativamente ao sistema operacional Android.

A proposta do aplicativo constituiu na implementação das operações fundamentais de manipulação de dados, estruturadas no modelo CRUD (*Create, Read, Update, Delete*), além da criação de um módulo de autenticação de usuário para controle de acesso às funcionalidades internas.

5.1 Plataforma Android

O Android constitui um sistema operacional baseado no *kernel Linux*, destinado a dispositivos móveis. A plataforma disponibiliza um conjunto de *APIs* que permitem o acesso a recursos de *hardware*, gerenciar interfaces gráficas, comunicar com a rede e persistência de dados.

5.2 Ambiente de Desenvolvimento

O Android Studio fornece recursos para desenvolvimento, compilação, depuração e testes de aplicações móveis. A IDE integra ferramentas de gerenciamento de dependências, *layout* visual e emulador de dispositivos.

5.3 Persistência de Dados com SQLite

O SQLite caracteriza-se como um banco de dados relacional leve, embarcado e transacional. Sua integração nativa ao Android permite armazenar em ambiente local eficiente e eliminar a necessidade de servidor externo ou dedicado.

A manipulação do banco ocorre por meio da classe SQLiteOpenHelper, responsável pela criação, atualização e gerenciamento do banco de dados.

5.4 Conceito de CRUD

CRUD representa as quatro operações básicas de manipulação de dados:

- Create – Inserção de registros
- Read – Consulta de registros
- Update – Atualização de registros
- Delete – Remoção de registros

Essas operações estruturam a base funcional do sistema desenvolvido.

5.4.1 Arquitetura do Sistema

O sistema adota a arquitetura em camadas simplificada, baseada na separação de responsabilidades entre interface, lógica de persistência e banco de dados, sendo composta por:

- Camada de Interface (Activities)
- Camada de Persistência (DatabaseHelper)
- Camada de Banco de Dados (SQLite)

A separação das responsabilidades assegura organização do código e facilita a manutenção.

5.4.2 Modelagem do sistema

O modelo de dados do sistema é composto por duas entidades principais: Usuário e Item.

A entidade Usuário possui os seguintes atributos:

- id: inteiro (PK - chave primária, autoincremento)
- username: texto (único)
- password: texto

A entidade Item é composta por:

- Id: inteiro (PK - chave primária, autoincrementada)
- nome: texto

Não há relacionamento direto entre as entidades, uma vez que o cadastro de itens não depende da associação estrutural com usuários do banco de dados.

5.4.3 Interface de Autenticação

A **Figura 5.1** ilustra a interface de autenticação do aplicativo Android desenvolvido. Esta tela constitui o ponto inicial de acesso ao sistema, sendo responsável pela validação das credenciais cadastradas no banco de dados SQLite.

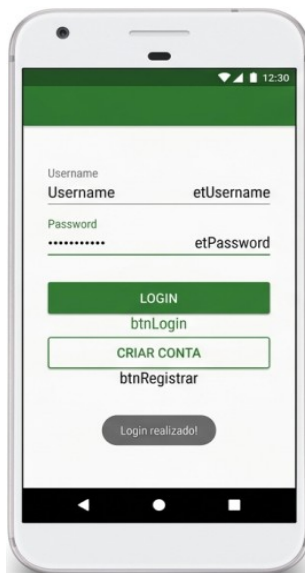
A interface é composta pelos campos *Username* e *Password*, destinados à inserção do nome de usuário e senha, respectivamente. Também estão disponíveis os botões Login e Criar Conta. O botão Login realiza a verificação das credenciais informadas e, quando válidas, permite o acesso às funcionalidades internas do sistema. Já o botão Criar Conta direciona o usuário para a tela de cadastro, possibilitando o registro de novas credenciais.

A implementação deste módulo teve como objetivo restringir o acesso às funcionalidades do aplicativo apenas aos usuários previamente cadastrados, introduzindo conceitos básicos de autenticação em aplicações móveis.

Antes da apresentação das interfaces do aplicativo, cabe destacar que as figuras apresentadas nesta seção foram reconstruídas digitalmente com auxílio de ferramenta de inteligência artificial, tomando como base a interface originalmente desenvolvida para o sistema. Tal procedimento foi adotado em razão da indisponibilidade do ambiente original de execução após a desinstalação do Android Studio dos computadores utilizados durante o desenvolvimento do projeto. As representações apresentadas preservam as

características visuais e funcionais da aplicação implementada, sendo utilizadas exclusivamente para fins de documentação acadêmica.

Figura 5.1 – Tela de autenticação do sistema



Fonte: Elaborado por ChatGPT (2026), sob orientação do autor.

5.4.4 Interface de Gerenciamento de Itens

A **Figura 5.2** é possível observar a interface principal de gerenciamento de itens do aplicativo. Esta tela concentra as operações fundamentais do modelo CRUD (Create, Read, Update e Delete), permitindo ao usuário realizar o cadastro, consulta, atualização e exclusão de registros armazenados localmente no banco de dados SQLite.

Na parte superior da interface encontra-se o campo de inserção de novos itens, acompanhado do botão de adição. A região central apresenta a listagem dos registros cadastrados, permitindo sua visualização de forma organizada. Para cada item disponível, o sistema disponibiliza botões específicos para edição e exclusão.

Ao selecionar a opção de edição, é exibida uma janela de diálogo contendo o valor atual do registro, permitindo sua alteração e posterior atualização no banco de dados. Dessa forma, a interface proporciona uma interação simples e intuitiva para o gerenciamento das informações armazenadas pela aplicação.

Figura 5.2 – Tela de gerenciamento de itens



Fonte: Elaborado por ChatGPT (2026), sob orientação do autor.

5.5 Considerações Técnicas

Do ponto de vista técnico, a escolha do SQLite mostrou-se adequada para a aplicações de pequeno porte que não exigem servidor externo. O banco de dados embarcado proporciona rapidez na leitura e gravação dos dados, além de simplicidade na implementação.

O uso do Android Studio contribuiu para organização do projeto, gerenciamento de dependências e execução de testes em múltiplos dispositivos virtuais.

Entretanto, identificam-se limitações técnicas relevantes. As senhas encontram-se armazenadas em formato de texto simples, o que compromete o nível de segurança da aplicação. Recomenda-se, como melhoria futura, a adoção de algoritmos de hash criptográficos (como SHA-256 ou BCrypt), para proteção das credenciais armazenadas.

Também se sugere a adoção de padrões arquiteturais mais robustos, como MVVM (Model-View-ViewModel), para maior escalabilidade e manutenção do código.

5.6 Resultados Obtidos

O aplicativo atendeu aos requisitos propostos na disciplina, apresentou de maneira correta as operações CRUD e o módulo de autenticação.

A aplicação demonstrou que é possível desenvolver soluções funcionais de gerenciamento de dados utilizando recursos nativos da plataforma Android, sem a necessidade de infraestrutura externa.

O projeto contribuiu para consolidação dos conhecimentos em:

- Programação Orientada a Objetos
- Modelagem de Banco de Dados
- Desenvolvimento Mobile
- Arquitetura de Software

5.7 Conclusão

O desenvolvimento deste projeto permitiu aplicar conceitos fundamentais de desenvolvimento mobile, banco de dados e programação orientada a objetos.

A utilização do Android Studio facilitou o processo de desenvolvimento, oferecendo ferramentas robustas para implementação e testes.

- Como trabalhos futuros, sugere-se:
- Implementação de criptografia de senhas.
- Integração com banco de dados remoto.
- Melhoria na interface gráfica.
- Aplicação de arquitetura MVVM.
- Implementação de validações mais robustas.

O projeto atendeu aos objetivos estabelecidos, demonstrando a viabilidade do desenvolvimento de aplicações móveis com persistência local de dados e arquitetura organizada com a utilização de Java e SQLite.