Universidade Positivo

Campus Ecoville

Curso Ciência da Computação

Mateus Bassi, Vinicius Ito e William Santos

Aplicativo Biblioteca: Projeto de Persistência de Dados Utilizando Room no Android

Curitiba, PR

2024

**1.0 INTRODUÇÃO**

O desenvolvimento de aplicativos móveis requer uma arquitetura bem estruturada, que facilite a manipulação e a persistência de dados de maneira eficiente. No contexto deste projeto, foi desenvolvido um aplicativo de biblioteca em Kotlin que utiliza o Room, uma biblioteca de persistência de dados local para Android, com o objetivo de fornecer uma solução robusta e escalável para armazenar e manipular informações de livros lidos pelos usuários.

A escolha do Room como solução de persistência de dados se deve à sua integração direta com o ecossistema Android e ao suporte nativo para operações com bancos de dados SQLite, o que facilita a criação de aplicativos com armazenamento local confiável. Dessa forma, o aplicativo é projetado para fornecer aos usuários uma experiência de uso intuitiva e funcional, onde é possível cadastrar, listar, atualizar e excluir registros de livros, preservando as informações mesmo quando o aplicativo é reiniciado ou fechado.

Neste relatório, a estrutura do app é detalhada, destacando as principais etapas de implementação da persistência de dados, as entidades utilizadas, os Data Access Objects (DAOs), e a configuração da classe de banco de dados. Além disso, descrevemos os fluxos operacionais para inserção, consulta e exclusão de dados, bem como as soluções criativas propostas para aprimorar a experiência do usuário e o desempenho da aplicação.

Esse projeto contribui para a construção de um modelo de armazenamento eficiente para aplicativos que necessitam de persistência de dados, reforçando as boas práticas de desenvolvimento em Android. A sistematização do conhecimento adquirido durante o desenvolvimento do aplicativo pode ser utilizada como base para futuros projetos que necessitem de integração com o Room, facilitando a aplicação em diferentes contextos e escalas.

Assim, ao longo deste relatório, espera-se que a descrição detalhada do desenvolvimento e das decisões técnicas forneça uma visão completa dos métodos empregados e dos desafios enfrentados, contribuindo para o aprimoramento de práticas de persistência de dados em projetos Android e a criação de aplicativos mais eficazes e intuitivos para os usuários finais.

**2.0 DESENVOLVIMENTO**

Neste capítulo, detalham-se as estruturas de dados e o fluxo de operações desenvolvidos para o gerenciamento dos registros de livros no aplicativo, desde o cadastro até as operações de consulta e exclusão. O sistema foi projetado para oferecer uma interface amigável e funcionalidade robusta, facilitando a organização e o controle dos livros que o usuário deseja registrar.

**2.1 Estrutura de Dados**

A estrutura de dados do aplicativo foi organizada de forma a refletir cada livro registrado como uma entidade no banco de dados. Abaixo, descrevem-se os componentes da estrutura de dados desenvolvida.

**2.1.1 Entidade Livro**

A entidade **Livro** representa a tabela principal do banco de dados, onde cada registro equivale a um livro cadastrado pelo usuário. Os campos desta entidade foram definidos para contemplar os dados essenciais de cada registro:

* **id**: Campo identificador único, gerado automaticamente pelo sistema, que permite que cada livro possua uma identificação exclusiva no banco de dados.
* **titulo**: Campo de texto que armazena o título do livro, facilitando a identificação e a consulta pelo usuário.
* **autor**: Campo de texto para o nome do autor do livro, importante para organizar e classificar os registros de maneira eficiente.
* **dataConclusao**: Campo de data que armazena a data em que o usuário concluiu a leitura do livro.
* **avaliacao**: Campo numérico que armazena a nota dada pelo usuário ao livro, variando de 1 a 5, permitindo que o usuário registre sua avaliação pessoal.

**2.1.2 DAO (Data Access Object)**

Para gerenciar o acesso e as operações com os dados da entidade **Livro**, foi implementado o **DAO LivroDao**, que define métodos para inserção, listagem, atualização e exclusão dos registros de livros. Abaixo estão descritos os principais métodos:

* **inserir(livro: Livro)**: Método que insere um novo livro na base de dados. Após a inserção, o registro é imediatamente adicionado à lista exibida na interface.
* **listarTodos()**: Método que retorna todos os livros cadastrados, ordenados pela data de conclusão. Esta função visa facilitar a visualização sequencial dos livros conforme a ordem de leitura.
* **obterPorId(id: Int)**: Método que realiza uma busca específica de um livro utilizando seu identificador único. Ele é especialmente útil para operações de atualização e exclusão de registros.
* **atualizar(livro: Livro)**: Método que permite a atualização das informações de um livro já existente, possibilitando que o usuário edite dados como título, autor e avaliação.
* **deletar(livro: Livro)**: Método que remove um livro do banco de dados. Este processo garante que o usuário possa manter sua lista de livros atualizada.

**2.2 Banco de Dados**

O banco de dados do aplicativo foi implementado com a tecnologia **Room**, que permite uma estrutura de armazenamento relacional robusta e eficiente. A classe **BibliotecaDatabase** estende a **RoomDatabase** e define as configurações da base de dados. Abaixo estão descritas as principais características da implementação:

* **Entidade**: A entidade principal utilizada é a **Livro**.
* **DAO**: O acesso ao banco de dados é gerido pelo **LivroDao**, que facilita a interação do sistema com os registros de livros.
* **Versão do banco**: O banco de dados está configurado inicialmente na versão 1, o que permite futuras migrações e ajustes sem comprometer a compatibilidade dos dados.
* **Singleton**: Foi implementado um padrão de instância Singleton para o banco de dados, garantindo que todos os componentes do aplicativo acessem a mesma instância, assegurando a consistência e o desempenho das operações de dados.

**2.3 Fluxos de Operações**

Os fluxos de operações definem como o usuário interage com os dados registrados no aplicativo, desde o momento de inserção até a exclusão de registros. Abaixo são descritos os principais fluxos de interação com o sistema.

**2.3.1 Fluxo de Inserção**

O fluxo de inserção é iniciado pelo usuário quando ele deseja cadastrar um novo livro na base de dados. O processo ocorre da seguinte forma:

1. O usuário preenche os campos **Título**, **Autor**, **Data de Conclusão** e **Avaliação** no formulário de cadastro.
2. Após o preenchimento, o usuário aciona o botão "Cadastrar Livro".
3. O aplicativo realiza uma validação dos dados para garantir que todos os campos foram preenchidos e que a avaliação está no intervalo permitido de 1 a 5.
4. Em caso de dados válidos, o livro é inserido no banco de dados, e a lista de livros é imediatamente atualizada na interface para refletir o novo registro.

**2.3.2 Fluxo de Consulta**

Para facilitar o acesso às informações de livros já cadastrados, o aplicativo realiza automaticamente uma consulta e exibição dos livros ao iniciar. Esse processo ocorre conforme descrito a seguir:

* Na inicialização, o aplicativo carrega todos os livros cadastrados no banco de dados e exibe-os em uma lista, ordenada pela data de conclusão.
* Cada item na lista apresenta o título, autor, data de conclusão e avaliação do livro, proporcionando ao usuário uma visualização clara e rápida dos livros lidos.

**2.3.3 Fluxo de Exclusão**

Embora o aplicativo não tenha implementado uma opção de exclusão direta na interface, o fluxo de exclusão foi projetado para permitir que, em futuras versões, o usuário possa selecionar um livro específico e removê-lo da base de dados de forma segura, evitando exclusões acidentais.

**2.4 Soluções Criativas**

Para aprimorar a usabilidade e a funcionalidade do aplicativo, foram propostas algumas melhorias no sistema. Abaixo estão listadas as principais sugestões de evolução:

* **Validação de dados**: A validação dos dados foi implementada no botão "Cadastrar Livro", garantindo que todos os campos estejam preenchidos e que a avaliação seja um valor numérico entre 1 e 5.
* **Ordenação e Filtros**: A inclusão de opções de ordenação, permitindo que o usuário organize a lista por título ou autor, além de um sistema de busca para encontrar livros específicos, traria mais flexibilidade e personalização para o uso do aplicativo.
* **Sistema de Marcação**: A adição de um campo de status que identifique se o livro foi lido ou ainda está pendente facilitaria a organização dos livros entre concluídos e não lidos, tornando o aplicativo ainda mais útil para o usuário.

Com essas melhorias, o aplicativo atenderia melhor às necessidades de organização e gestão dos registros de leitura, oferecendo ao usuário uma experiência aprimorada e um controle mais preciso de suas leituras.

**3.0 CONCLUSÃO**

O desenvolvimento deste projeto proporcionou uma compreensão aprofundada da estrutura e funcionamento da biblioteca Room, destacando a importância da definição adequada de entidades, DAOs e da configuração do banco de dados para a persistência de dados em um aplicativo. A implementação do padrão Singleton para a criação da instância do banco de dados foi crucial para garantir a eficiência e a segurança do aplicativo, permitindo que diferentes componentes acessassem a mesma instância de forma controlada.

A validação dos dados inseridos pelo usuário e as propostas de melhoria identificadas ao longo do desenvolvimento contribuíram significativamente para aprimorar a experiência do usuário e o desempenho geral do aplicativo. A inclusão de funcionalidades como ordenação, filtragem e um sistema de marcação para controle de leitura demonstrou a preocupação em tornar a interface mais amigável e funcional, atendendo assim às necessidades dos usuários.

Entre os principais desafios enfrentados, destacam-se a criação de uma interface intuitiva, capaz de facilitar a interação do usuário com o sistema, e a implementação correta dos métodos de manipulação de dados de maneira assíncrona, o que é essencial para manter a fluidez do aplicativo e evitar travamentos durante as operações.

Em suma, o projeto não apenas consolidou conhecimentos técnicos sobre o uso da biblioteca Room e gerenciamento de dados em Android, mas também evidenciou a importância de uma abordagem centrada no usuário para o desenvolvimento de aplicativos. As lições aprendidas e as soluções implementadas servem como base para futuros desenvolvimentos, oferecendo um caminho claro para a evolução e expansão das funcionalidades do aplicativo.