

# GERADOR DE CÓDIGO PARA APLICATIVO ANDROID

## CODE GENERATOR FOR ANDROID APP

<sup>1</sup>VIEIRA, F. C.; <sup>2</sup>DELFINO, S. R.

<sup>1 e 2</sup>Bacharelado em Sistemas de Informação - Faculdades Integradas de Ourinhos - FIO/FEMM

<sup>2</sup>Centro de Estudos em Games Aplicados - Faculdade de Tecnologia de Ourinhos - FATEC

### RESUMO

No desenvolvimento de aplicações que envolvem banco de dados são comuns operações de inserção, seleção, exclusão e edição. A codificação de tais operadores é repetitiva e toma tempo significativo do projeto, impossibilitando maior dedicação a etapas fundamentais para garantir a satisfação dos clientes, que abrangem o planejamento e implementação de funcionalidades complexas. Desta forma, o presente trabalho se propõe a desenvolver uma ferramenta para geração de código fonte a partir dos metadados do modelo de dados da aplicação, o qual deverá codificar automaticamente todas as operações básicas relacionadas ao banco de dados. Considerando a ascendência das tecnologias móveis no mercado mundial, pretende-se aperfeiçoar o desenvolvimento de aplicativos para a plataforma Android, bem como melhorar a qualidade do código gerado, seguindo os padrões dos projetos Android.

**Palavras-chave:** Aplicativos Android. Banco de Dados. Gerador de Código.

### ABSTRACT

In developing applications that involve database operations are common insertion, selection, deleting, and editing. The encoding of such operators is repetitive and takes significant time project, preventing greater dedication to critical steps to ensure customer satisfaction, covering the planning and implementation of complex functionality. Thus, this study aims to develop a tool for generating source code from the metadata of the data model of the application, which will automatically encode all basic operations related to the database. Given the ascendancy of mobile technologies in the world market, we intend to enhance the development of applications for the Android platform as well as improve the quality of generated code, following the standards of Android projects.

**Keywords:** Android Applications. Database. Code Generator.

### INTRODUÇÃO

Na economia moderna é frequentemente difícil prever como um software irá evoluir com o tempo. As condições de mercado mudam rapidamente, as necessidades dos usuários finais se alteram e novas ameaças competitivas emergem sem aviso. É preciso ser ágil o suficiente para dar uma resposta ao ambiente de negócios. (PRESSMAN, 2011, 82).

Segundo Lecheta (2010, p. 19), mais da metade da população mundial possui aparelho celular, além do que mesmo os usuários comuns procuram por aparelhos com recursos diversos, ficando clara a importância de se pensar e desenvolver funcionalidades inovadoras e que chamem a atenção do consumidor.

Não obstante, o ambiente empresarial, afirma Lecheta (2010, p. 19), visando lucratividade e inovação, busca invariavelmente implantar ao seu cotidiano aplicações móveis a fim de agilizar seus processos, já que é possível com as tecnologias atuais integrar o sistema da empresa às aplicações utilizadas nos tablets ou smartphones.

Para manter-se em alta no mercado e seguir essa evolução, Lecheta (2010, p. 20) diz que as empresas, desenvolvedores, fabricantes e operadoras de celulares competem na corrida por tecnologias móveis.

Ferreira (2011, p. 3), diz que há, hoje no mercado, diversos sistemas operacionais para celulares e smartphones, o que causa falta de padronização e um enorme esforço ao tentar portar aplicações e utilizar recursos entre estes diferentes modelos e marcas. O desenvolvimento e expansão da plataforma Android teve como finalidade justamente solucionar esse problema. E maneiras de facilitar e melhorar a produção de aplicações são sempre requisitadas.

O gerador de código apresenta-se como a base para prosseguir no desenvolvimento de uma aplicação, já que é ferramenta de software desenvolvida a fim de gerar código fonte. Geradores de código, conforme Herrington (2003, p. 3, tradução nossa) são programas que escrevem programas, sendo capazes de construir códigos de várias linguagens e realizar a geração de uma só vez.

Dentre os geradores de código ativo há gama enorme de ferramentas que vão desde as mais simples até as mais complexas, sendo possível encontrar geradores diferentes quanto à funcionalidade, usabilidade, complexidade e tipos de saída (HERRINGTON, 2003, tradução nossa).

Herrington (2003, p. 29, tradução nossa), explica que os modelos de geradores de código tendem a ter arquiteturas vastamente diferentes e são usados para diferentes problemas e de várias maneiras. Herrington (2003), classifica seis geradores de código ativo.

Primeiramente, o código munger, conforme Herrington (2003, p. 29, tradução nossa) onde se entra com arquivos do código fonte, usando expressões regulares ou simples análise sintática de código e, em seguida usa parte dos mesmos ou templates para construir arquivos de saída.

O expansor incline-code é muito utilizado para integrar código Structured Query Language (SQL) no arquivo de código fonte. Este tipo de gerador recebe código fonte como entrada e gera a produção de código como saída.

Já o gerador mixed-code procura especialmente por comentários formatados, e quando os encontra, preenche a área de comentário com algum código fonte novo exigido pela produção. Os comentários no código especificam o mapeamento entre elementos e controle de dados, e o gerador mixed-code adiciona uma implementação que casa a especificação com o comentário (HERRINGTON, 2003, p. 30-31, tradução nossa).

Na geração partial-class é realizada a leitura de arquivo resumido que possui informação suficiente para definir as classes básicas. Na sequência, são construídas bibliotecas de saída de classe através de templates. Essas mesmas classes são posteriormente compiladas com classes desenvolvidas pelos programadores. (HERRINGTON, 2003, p. 31, tradução nossa).

Na camada ou gerador de camada, o gerador tem a responsabilidade de construir camada completa de sistema de n-camadas. Um exemplo deste tipo de gerador de camada é geração modeldriven, onde uma aplicação para Unified Modeling Language (UML) é usada em conjunto, com gerador e um arquivo de definição de entrada (frequentemente em Extensible Markup Language (XML)) para uma saída ou mais camadas de um sistema. (HERRINGTON, 2003, p. 32, tradução nossa).

E por fim, a linguagem full-domain, que de acordo com Herrington (2003, p. 33, tradução nossa), é uma linguagem Turing completamente projetada para permitir que desenvolvedores representem os conceitos no domínio mais facilmente. Uma linguagem Turing completa é linguagem de computador de uso geral que dá suporte a todas as gerências variáveis, lógica, ramificação, funcional e habilidades de decomposição de objeto, incluídas com linguagens de programação.

O presente projeto se justifica pela necessidade de uma ferramenta para auxiliar o desenvolvimento de aplicações Android a partir da geração do código base, o qual demanda muito tempo de programação, possibilitando assim que o programador possa se dedicar mais ao planejamento da aplicação e implementação de funcionalidades específicas, garantindo projeto eficiente e resultado satisfatório.

O projeto tem como objetivo, a criação de ferramenta para geração de código das operações básicas de criação, seleção, exclusão e atualização dos dados de aplicativos Android, através da obtenção dos metadados do modelo de dados da aplicação. Pretende-se gerar a base para o desenvolvimento de aplicativo Android, automatizando a criação das telas e das classes, que gerenciam cada tela, para as

operações básicas da aplicação, bem como dos pacotes a que pertencem e demais classes fundamentais.

## MATERIAL E MÉTODOS

O software de geração de código funcionará em modo desktop. A linguagem Java foi escolhida para o desenvolvimento do projeto que será desenvolvido, utilizando o NetBeans IDE.

Para incorporar a aplicação gerada serão utilizados princípios e características da estrutura de arquivos e diretórios do próprio projeto de aplicativos Android. Segundo Lecheta (2013, p. 69), há arquivos que são criados automaticamente no projeto e que representam a estrutura básica de aplicação Android.

A persistência dos dados nos aplicativos gerados a partir da ferramenta desenvolvida, se realizará por meio do SQLite, sendo necessária a obtenção dos metadados contidos no modelo de dados do sistema.

A extração dos metadados será por meio das classes `DataBaseMetadata`<sup>1</sup> e `ResultSetMetadata`<sup>2</sup>, ambas pertencentes à API JDBC. O desenvolvimento do gerador seguirá estrutura previamente prototipada, tendo modificações, conforme necessidade.

O gerador conterà uma tela, previamente prototipada conforme Figura 1, onde será possível realizar as configurações referentes à aplicação a ser gerada. Como já foi dito, a ferramenta seguirá a estrutura de arquivos de aplicativos Android, gerando na pasta `res` as `activitys`, bem como as classes referentes ao banco de dados, organizadas em quatro pacotes: `modelo`, `dao`, `util` e `activitys`.

No pacote `modelo` ficarão os objetos do negócio, responsáveis por fornecer acesso aos dados. O `dao` conterà as classes referentes às operações CRUD do projeto gerado. No pacote `util` deverão estar as classes referentes à criação do banco de dados. E por fim, no pacote `activitys`, estarão justamente as `activitys` que são as classes que interagem com o XML. O processo de geração que se pretende seguir é demonstrado na Figura 2.

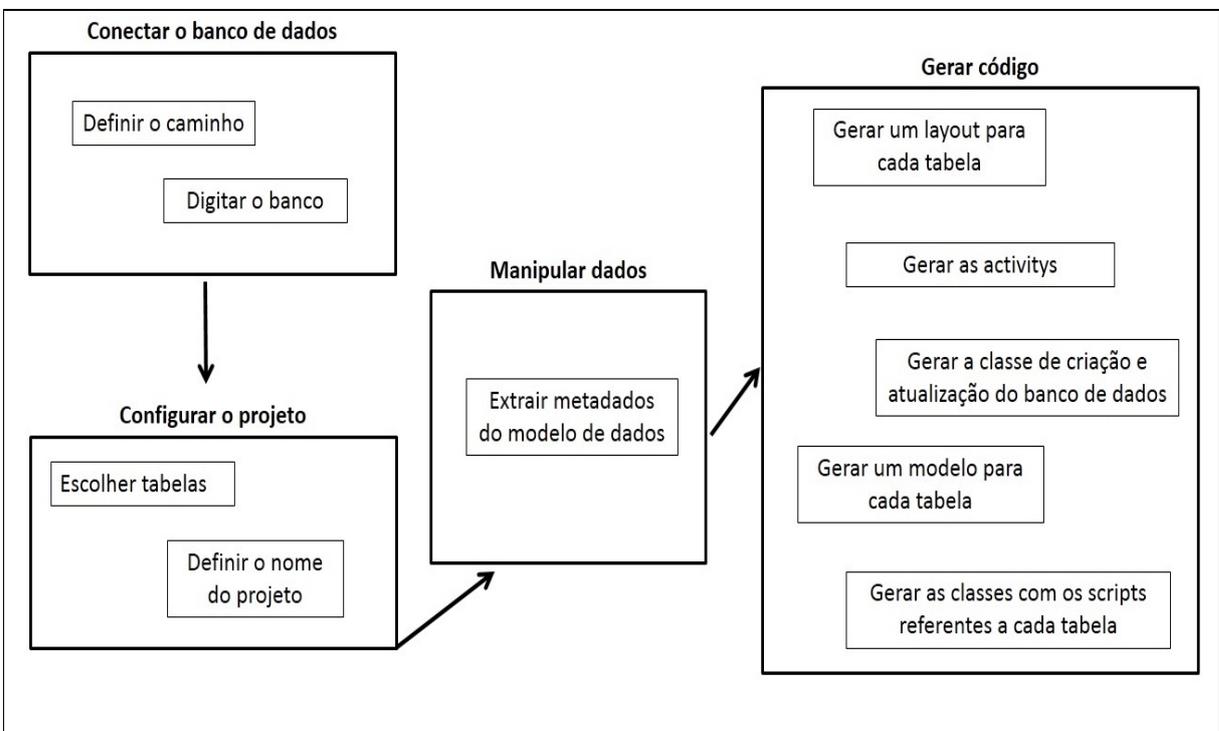
---

<sup>1</sup> `java.sql.DatabaseMetaData`: atua sobre o banco de dados como um todo: os nomes das tabelas, índices de tabela, nome de produto de banco de dados e versão, e as ações que o banco de dados suporta (JAVA2S, 2009, tradução nossa).

<sup>2</sup> `java.sql.ResultSetMetaData`: atua sobre os tipos e as propriedades das colunas em um objeto `ResultSet` (JAVA2S, 2009, tradução nossa).

**Figura 1** – Protótipo da tela de configuração do sistema de geração de código

**Figura 2** – Processo de geração



A validação do software ocorrerá por meio da geração de código de modelo de dados de aplicativo Android, previamente implementado manualmente no ADT. A

pasta gerada será importada para ambiente de desenvolvimento Android sendo realizadas as alterações necessárias e, conseguinte a emulação do projeto. Desta maneira será possível avaliar os resultados obtidos através da geração.

## **RESULTADOS E DISCUSSÃO**

Os resultados pretendidos se referem ao bom funcionamento da ferramenta considerando o objetivo que permeia este projeto, ou seja, a possibilidade de agilidade e melhoramento do processo de desenvolvimento de aplicativos Android através da geração automática das operações básicas.

Espera-se que a ferramenta seja capaz de gerar código fonte, usando os metadados obtidos do modelo de dados, possibilitando ao utilizador realizar as configurações necessárias.

Por fim, a expectativa é a geração dos pacotes básicos da aplicação Android, podendo o arquivo gerado ser importado para um ambiente de desenvolvimento integrado que suporte a plataforma Android, para que as funcionalidades complexas sejam codificadas pelo programador, podendo finalizar o projeto e executar a aplicação.

## **CONCLUSÃO**

A plataforma Android oferece grande diversidade de componentes para projetar e desenvolver a interface com o usuário, além de apoiar-se na linguagem Java para implementar suas funcionalidades. Apresenta ainda reflexo significativo no mercado de aparelhos móveis atualmente, sendo muito procurado pelos desenvolvedores.

Solucionar a necessidade de redução no tempo de produção e melhoramento na qualidade do código gerado é fundamental neste cenário. Desta forma, o gerador que se pretende desenvolver neste estudo é uma resposta a essas necessidades.

Destaca-se a importância da ferramenta para o desenvolvimento de tais aplicações, da qual se espera extrair um aperfeiçoamento do processo.

## REFERÊNCIAS

FERREIRA, R. L. **Mini Curso Android**. 2011. Disponível em: <<http://www.dniel.com.br/dniel/cursoAndroid.pdf>>. Acesso em: 25 abr. 2013.

HERRINGTON, J. **Code generation in action**. Greenwich: Manning Publications Co. 372 p. 2003.

LECHETA, R. **Google Android: aprenda a criar aplicações com dispositivos móveis com o Android SDK**. 2 ed. São Paulo: Novatec, 2010.

LECHETA, R. **Google Android: aprenda a criar aplicações com dispositivos móveis com o Android SDK**. 3 ed. São Paulo: Novatec, 2013.

PRESSMAN, R. S. **Engenharia de Software: uma abordagem profissional**. 7 ed. Porto alegre: AMGH, 2011.