

GERAÇÃO DE CÓDIGO PARA APLICAÇÕES CRUD MÓVEIS VISANDO MÚLTIPLAS PLATAFORMAS DE NUVEM

KELLERSON KURTZ¹; LISANE BRISOLARA²

¹Universidade Federal de Pelotas – kkurtz@inf.ufpel.edu.br

²Universidade Federal de Pelotas – lisane@inf.ufpel.edu.br

1. INTRODUÇÃO

Os desenvolvedores de aplicativos para dispositivos móveis devem lidar com dificuldades relacionadas a limitações de capacidade de processamento e de armazenamento bem como restrições de consumo energético inerentes a estas plataformas. Estas restrições combinadas com particularidade das diferentes plataformas de desenvolvimento disponíveis representam desafios aos desenvolvedores.

Abordagens de desenvolvimento orientadas a modelos vêm sendo empregadas para facilitar e agilizar o processo de desenvolvimento de software (PAPOTTI, 2013). Nestas abordagens, modelos guiam o desenvolvimento e ferramentas de suporte recebem estes modelos como entrada e geram código, de modo a reduzir o período de implementação. Um exemplo é a ferramenta GenCode (PARADA et al., 2015) que usa os diagramas de classe e de sequência para gerar código para aplicativos Android e Windows Phone.

A computação na nuvem é empregada quando uma aplicação móvel requer o armazenamento de grande volume de dados, ou requer maior poder de processamento, ou menor consumo da bateria (DINH et al., 2013). Desta forma, a aplicação não fica limitada aos recursos disponibilizados pelo dispositivo, dados ou poder de processamento externo são acessados via Internet. Contudo, o emprego da nuvem traz desafios adicionais aos desenvolvedores.

Considerando aplicativos que armazenam dados na nuvem, as operações mais frequentes são: criar, ler, atualizar e excluir dados – chamadas operações CRUD. Em Ribeiro (2016) foi proposto um padrão para representação das operações CRUD em aplicativos Android. O padrão proposto emprega diagramas de classe e de sequência e abstrai detalhes das diferentes plataformas de nuvem, visando suportar a geração de código multi-plataforma. No entanto, este trabalho anterior foca na modelagem da comunicação do aplicativo com a nuvem, não preocupando-se com a automatização da geração de código em si.

Este trabalho propõe o aperfeiçoamento e a implementação do padrão proposto em Ribeiro (2016), visando a geração automática de código de aplicações CRUD móveis que empregam infraestruturas de nuvem. O emprego do padrão permitirá abstrair detalhes de comunicação com uma dada infraestrutura de nuvem, facilitando o projeto dos aplicativos e a migração dos mesmos para diferentes nuvens. Além disso, a ferramenta GenCode está sendo estendida para incluir o suporte às plataformas de nuvem Bluemix (IBM, 2016) e Google App Engine (GOOGLE, 2016).

2. METODOLOGIA

Inicialmente, uma revisão da implementação da GenCode foi realizada, visando entender como a ferramenta interpreta o modelo, representado por um arquivo XML, de modo a capturar elementos dos dois diagramas empregados.

O padrão proposto por Ribeiro (2016) para a modelagem de aplicações CRUD, foi revisado e refinado, visando suportar sua implementação na GenCode. Além disso, o estudo de caso existente envolvendo as plataformas Bluemix e Google App Engine foi revisado, identificando algumas modificações requeridas no padrão a fim de que a transparência da infraestrutura da nuvem fosse completa, propiciando que a partir de um mesmo modelo fosse possível gerar código para as duas plataformas alvo.

Posteriormente, um estudo das plataformas Bluemix e Google App Engine foi realizado, visando compreender a implementação das operações CRUD em cada uma das plataformas. Este estudo serviu de base para planejar a extensão da GenCode, incluindo o suporte à interpretação do novo padrão e a geração automática de código para as operações CRUD nas referidas plataformas.

3. RESULTADOS E DISCUSSÃO

O padrão definido propõe um conjunto de regras que permitem ao desenvolvedor representar através de modelos, de forma simples e objetiva, quais tipos de objetos serão hospedados na nuvem, quais operações CRUD serão permitidas para cada tipo de objeto e qual classe possuirá os métodos responsáveis por estas operações. Todas estas definições devem ser feitas independentemente da plataforma de nuvem a ser empregada.

No diagrama de classes, o ponto inicial é a criação de uma superclasse chamada "CLOUD", da qual, todas as demais classes que terão suas instâncias armazenadas na nuvem deverão ser derivadas. Todo objeto a ser hospedado na nuvem, deve possuir o atributo "key" do tipo *String*, que o diferenciará dos demais objetos. Este atributo será empregado nas buscas, atualizações e exclusões.

Após, a classe responsável pela interação com a nuvem deve conter um método correspondente para cada operação CRUD a ser permitida àquele tipo de objeto. A nomenclatura dos métodos referentes às operações CRUD deve obedecer a seguinte regra "<operação><tipo_objeto>", na qual "<operação>" deve ser substituído por "create", "read", "update" ou "delete"; e "<tipo_objeto>" deve ser substituído pelo tipo do objeto a ser armazenado na nuvem. O diagrama de sequência é empregado para estabelecer a comunicação das instâncias destas classes com os demais componentes da aplicação. Por fim, o modelo UML é capturado pela Gencode que realiza a identificação dos componentes da aplicação e posteriormente, a geração do código correspondente.

A nova versão da GenCode deve capturar as informações do modelo UML e, caso trate-se de uma aplicação envolvendo a nuvem, a ferramenta identificará toda a comunicação representada nos diagramas e gerará adequadamente os métodos responsáveis pelas operações CRUD. Primeiramente, métodos independentes de plataforma são gerados. Quando uma plataforma for selecionada, métodos que efetivamente, fazem a comunicação com os servidores serão gerados. Estes métodos recebem o nome "<operação><tipo_objeto>Cloud" e são totalmente dependentes da plataforma selecionada pelo desenvolvedor.

Uma aplicação Android de lista de compras é usada como estudo de caso para demonstrar a abordagem proposta para modelagem e geração de código. Figura 1 ilustra as classes da aplicação, onde a classe "Item", representa os objetos armazenados na nuvem. A classe "MainActivity" é responsável pela manipulação

das operações CRUD para instâncias de "Item" e a classe "InterfaceGrafica" é responsável por invocar os métodos CRUD de "MainActivity".

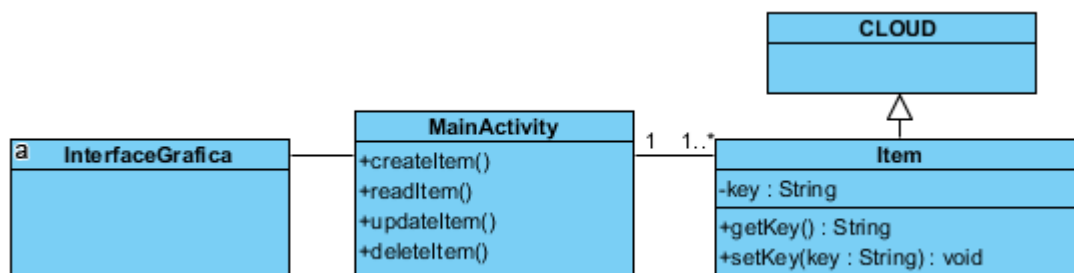


Figura 1. Diagrama de classe da aplicação Lista de Compras

Na Figura 2, o diagrama de sequência representa a modelagem independente de plataforma da inserção de um novo item na lista. Para a implementação em uma plataforma específica, este comportamento deverá ser transformado. As Figuras 3 e 4 representam as modificações no comportamento do método "createItem" que a GenCode processa para a inserção de um novo item na lista quando as plataformas Bluemix e Google App Engine são selecionadas, respectivamente. É possível observar que a interação com as infraestruturas de nuvem se difere apenas pela invocação do método específico para salvamento dos dados na nuvem. No caso do uso da Bluemix, Item por ser uma especialização de uma classe provida pela infraestrutura, já oferece este método. Porém, no caso do emprego da Goggle App Engine, o método deve ser invocado para o componente Firebase, provido pela plataforma.

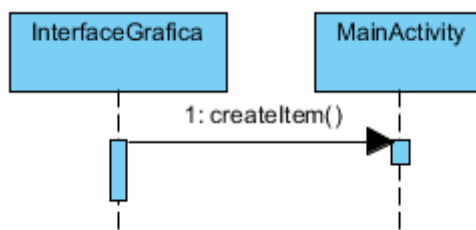


Figura 2. Diagrama de sequência para inserção de um novo item na lista

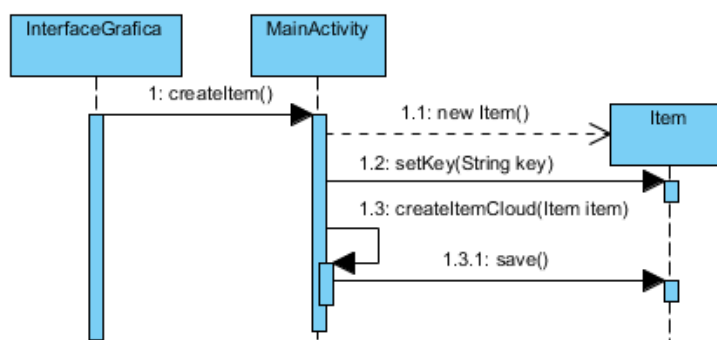


Figura 3. Diagrama de sequência do código gerado para Bluemix

Por limitações de espaço, a abordagem proposta foi discutida usando apenas a operação de inserção do CRUD. Além disso, foram discutidos aqui apenas os modelos que refletem os códigos a serem gerados para cada plataforma, omitindo trechos de código gerados a partir dos mesmos.

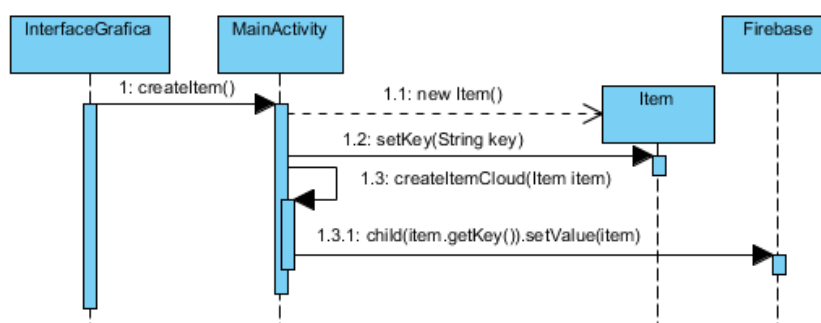


Figura 4. Diagrama de sequência do código gerado para Google App Engine

4. CONCLUSÕES

Este trabalho apresenta um padrão simples de modelagem de aplicações CRUD móveis empregando infraestruturas de nuvem. A modelagem empregada baseia-se em diagramas de classe e de sequência, representando estrutura e comportamento da aplicação. O suporte ao padrão proposto está sendo implementado na ferramenta GenCode para automatizar a geração de código para aplicativos CRUD de computação móvel na nuvem empregando Bluemix e Google App Engine. Desta forma, desenvolvedores não necessitam de conhecimentos específicos sobre a programação com plataformas de nuvem para desenvolver aplicativos CRUD na nuvem. Além disso, a migração de aplicativos locais para uma infraestrutura de nuvem, ou entre infraestruturas, é facilitada uma vez que códigos são gerados automaticamente a partir de modelo UML independente de plataforma de nuvem. Como trabalhos futuros, pretendemos concluir a extensão da GenCode discutida aqui e realizar as devidas validações. Posteriormente, planejamos oferecer também suporte a outras importantes plataformas de nuvem, como Microsoft Azure e Amazon Web Services.

5. REFERÊNCIAS BIBLIOGRÁFICAS

RIBEIRO, B. G. **Uma abordagem para modelagem de aplicações de computação móvel na nuvem**. 2016. Dissertação de Mestrado em Computação – PPGC, UFPel.

PARADA, A., MARQUES, M., BRISOLARA, L. Automating mobile application development: UML-based code generation for Android and Windows Phone. **Revista de Informática Teórica e Aplicada**, vol. 22, n. 2, pp. 31-50, 2015.

PAPOTTI, P. **Um processo dirigido a modelos para geração de código**. 2013. Dissertação de Pós-Graduação em Ciência da Computação – PPGCC, UFSCar.

DINH, H. T. et al. A survey of mobile cloud computing: architecture, APplications, and APProaches. **Communications and Mobile Computing**, vol. 13, n. 18, pp. 1587-1611.

IBM. **IBM Bluemix**. Acessado em 27 jul. 2016. Online. Disponível em: <http://www.ibm.com/cloud-computing/bluemix/br-pt/>

GOOGLE. **Google App Engine**. Acessado em 27 jul. 2016. Online. Disponível em: <https://cloud.google.com/appengine/>