

DESENVOLVIMENTO DE APLICATIVO ANDROID PARA VAGAS DE EMPREGOS UTILIZANDO A METODOLOGIA ÁGIL SCRUM

Denise Xavier Fortes

Docente da Faculdade Sete de Setembro – Fasete - Mestranda em Ciência da Computação. Programa de Pós-Graduação em Ciência da Computação (PROCC) Universidade Federal de Sergipe (UFS) – São Cristóvão, SE – Brasil. Especialista em Banco de Dados pela Universidade Gama Filho (2010). Graduada em Sistemas de Informação pela Faculdade Sete de Setembro – FASETE (2010) - Paulo Afonso – BA, Brasil. denise.fortes@fasete.edu.br

Ricardo Azevedo Porto

Docente da Faculdade Sete de Setembro – FASETE. Mestre em Sistemas e Computação pela Universidade Salvador- UNIFACS. Pós-graduado em Sistemas de Informação com Ênfase em Banco de Dados pela Faculdade Ruy Barbosa. Graduado em Bacharelado em Ciência da Computação pela Universidade Tiradentes – UNIT. ric.porto@gmail.com

Zenilton de Sá Neves

Graduando em Bacharelado em Sistemas de Informação pela Faculdade Sete de Setembro – FASETE. - zeniltom.sa@gmail.com

RESUMO

Diante dos avanços tecnológicos e o crescimento exponencial de dispositivos móveis, o número de pessoas que utiliza estes dispositivos para realizarem suas tarefas aumentou de forma considerável. Sendo assim, o presente estudo visa explanar sobre conceitos, definições e tecnologias utilizadas para desenvolver um aplicativo móvel que possa auxiliar pessoas a encontrar uma nova oportunidade de trabalho. Para isso, utilizou-se um Survey para a coleta de requisitos necessários para a aplicação e a metodologia ágil para auxiliar o desenvolvimento, o Scrum Solo. Além disso, utilizou-se a IDE Android Studio para o ambiente de desenvolvimento, o Firebase Authentication para controlar o login e as sessões do app, o Firebase Realtime Database para armazenar os dados numa plataforma online.

Palavras-chave: Android. Metodologias Ágeis. Scrum. Survey.

ABSTRACT

In face of the technological advances and the exponential increase of mobile devices, the number of people that make use of these devices to carry out their tasks has significantly increased. Therefore, this paper aims to explain concepts, definitions and technologies used to develop a mobile app that could help people find a new job opportunity. For this, it was used a survey to perform the collection of necessary requirements for the application and agile methodology to assist on the development, the Scrum Solo. Besides, it was used the IDE Android Studio for the development, the Authentication Firebase To control the login and the app sessions and the Realtime Database Firebase to keep the data in an online platform.

Keywords: Android. Agile methodology. Scrum. Survey.

1 INTRODUÇÃO

Os avanços tecnológicos e o crescimento no uso de dispositivos móveis para realizar tarefas rotineiras, possibilitaram o desenvolvimento de sistemas robustos e de dispositivos portáteis inteligentes. “Estudos mostram que hoje em dia mais de 3 bilhões de pessoas possuem um aparelho celular, e isso corresponde a mais ou menos metade da população mundial” (LECHETA, 2010, p 18). De acordo com o Global Mobile Consumer Survey (2016), no Brasil, 80% dos brasileiros já utilizam smartphones. Diante deste contexto, o presente estudo visa o desenvolvimento de uma aplicação para a plataforma Android utilizando uma metodologia ágil de desenvolvimento. Onde esta aplicação exibirá as vagas de empregos disponíveis em uma determinada região.

2 FERRAMENTAS

2.1 Firebase

O *Firebase* é uma plataforma de serviços em nuvem para a construção de aplicativos mobile com o intuito de auxiliar os desenvolvedores no processo de desenvolvimento do software. (FIREBASE, 2017). Logo, pode-se afirmar que o *Firebase* é um *Backend as a Service* (BaaS), ou seja, um serviço de computação na nuvem. De acordo com Sareen (2013, p.534):

[...] também conhecido como "backend móvel como um serviço" (MBaaS), é um modelo para fornecer a desenvolvedores de aplicativos web e móveis uma maneira de vincular seus aplicativos de armazenamento em nuvem backend, além de fornecer recursos como gerenciamento de usuários, envio de notificações, e a integração com serviços de redes sociais. Estes serviços são fornecidos através do uso de kits de desenvolvimento de software (SDKs) e as interfaces de programação de aplicativos (APIs).

Dessa forma, o BaaS serve como um intermediário entre o *front-end* e o *back-end*. de acordo com Cheng (2017, p.4):

Os aplicativos móveis geralmente precisam de serviços de back-end para trabalhar com a interface de usuário front-end. Isso significa que deve haver código de back-end e servidores para trabalhar com aplicativos móveis. Firebase é um serviço em nuvem para aplicativos de back-end. Firebase pode fornecer suporte para armazenamento de dados e autenticação de usuário. Após a integração com aplicativos móveis Firebase, não precisamos escrever código de back-end ou gerenciar a infraestrutura.

2.1.1 Firebase Authentication

O *Firebase Authentication* é responsável pelo processo de autenticação. Conforme a Equipe *Firebase* (2017):

O *Firebase Authentication* fornece serviços de back-end, SDKs fáceis de usar e bibliotecas de IU prontas para autenticar usuários no seu app. Ele oferece suporte à autenticação por meio de senhas, números de telefone e provedores de identidade federados como Google, Facebook, Twitter e muito mais.

Assim, o *Firebase* permite aos desenvolvedores integrar vários métodos de autenticação nos seus aplicativos, garantindo assim, segurança. A figura 1 ilustra os métodos de autenticação disponíveis no *Firebase*, suportando a conta do *Google*, *Facebook*, *Twitter*, *GitHub* e etc. Através desta, percebe-se que é possível acessar informações básicas dos perfis das redes sociais e utilizar nos *apps* com segurança e de forma simultânea.

Figura 1. Métodos de autenticação do *Firebase*.



Fonte: FIREBASE (2017)

2.1.2 *Firebase realtime database*

O *Firebase Realtime Database* é um banco de dados hospedado na nuvem, no qual os dados armazenados são estruturados no formato JSON e sincronizados em tempo real com todos os clientes conectados. Possibilitando assim, atualizações simultâneas de dados. (FIREBASE, 2017). A seguir, a figura 2, representa o uso do *Realtime Database*, exemplifica a seguinte situação: Se um cliente altera uma informação e existem outros clientes conectados, ao mesmo tempo, aquela informação alterada é sincronizada automaticamente com todos os clientes disponíveis.

Figura 2. Sincronização em tempo real com clientes conectados



Fonte: FIREBASE (2017)

2.2 Plataformas de Software

Plataforma é um conjunto de ativos que podem ser usados para alavancar o reuso e o rápido desenvolvimento de novos produtos. Desta forma, ela define o ambiente operacional, a arquitetura em alto nível de todos os produtos desenvolvidos com base nesta plataforma e um conjunto de políticas de desenvolvimento para aperfeiçoar a plataforma e o desenvolvimento de produtos (MURER, 2012). Desta maneira, os “ativos” são as ferramentas de programação. Sendo assim, as Plataformas de *Software* surgiram para facilitar, agilizar e flexibilizar o desenvolvimento de *Softwares*, pois são responsáveis pela codificação da aplicação, disponibilizando ferramentas de programação, interpretação das linguagens de programação, execução, visualização e acompanhamento dos resultados obtidos. Para o desenvolvimento da aplicação em questão, será utilizado o *Android Studio*. De acordo com QUEIRÓS (2016), o *Android Studio* é um ambiente integrado para o desenvolvimento de aplicações para a plataforma *Android*.

3 DESENVOLVIMENTO

3.1 METODOLOGIAS ÁGEIS

Com a necessidade de obter melhores resultados no processo de desenvolvimento de *software* que proporcionem agilidade, flexibilidade, diminuição dos riscos e gerem menores custos para as empresas foram criados métodos de desenvolvimento mais ágeis, produtivos e que garantam eficiência nas mudanças que surgiriam nos projetos, nasceram as metodologias ágeis. De acordo

com SBROCCO and MACEDO (2012, p.88), são os melhores métodos para desenvolvimento de *software*, pois contém um conjunto de princípios que definem critérios para os processos de desenvolvimento ágil de sistemas. Portanto, contribuem imensamente na agilidade, qualidade e desempenho dos projetos de *software*. Dentre as quais, destaca-se a SCRUM.

3.1.1 Scrum

O Scrum é um *framework* para a gestão do desenvolvimento de *software*, composto por diversas técnicas e processos para melhorar as práticas de desenvolvimento. De acordo com (BARBOSA et. al, 2015, p.9), o Scrum “[...] vem sendo utilizado desde o início da década de 1990, em sua utilização é comum ser combinado com outra metodologia ágil”. Desta forma, produzindo melhores resultados no produto final. Barbosa et. al (2015, p.10) afirmam que o Scrum é composto por um conjunto de papéis que precisam ser respeitados, são estes:

Product Owner, o Time de Desenvolvedores e o *Scrum Master*. O *Product Owner* é o dono do produto e é a única pessoa responsável por gerenciar o *Backlog* do produto. O Time de Desenvolvimento, consiste em profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto “Pronto” ao fim de cada *Sprint*. O *Scrum Master* é o responsável por garantir que o Scrum seja entendido e aplicado pelo time.

Com esse conjunto de regras e práticas sendo aplicados corretamente é possível desenvolver qualquer tipo de produto conforme a necessidade da empresa. Segundo Barbosa et. al (2015, p.10):

O Scrum utiliza iterações com duração entre uma a quatro semanas, chamadas *Sprints*. A quantidade de *Sprint* varia de acordo com a complexidade e o tamanho da aplicação. Cada *Sprint* tem a definição do que deverá ser construído, o plano que irá guiar a construção, o trabalho e o resultado do produto. Durante uma *Sprint* não são introduzidas modificações o que permite que a equipe trabalhe em um ambiente de curto prazo, mas estável. Ao final da *Sprint* um produto funcional é entregue e os recursos pendentes são encaminhados para a próxima iteração.

São selecionadas para uma *Sprint*, as tarefas retiradas do *Product Backlog* que contém um conjunto de requisitos que definem o que precisa ser realizado. As *Sprints* resultam no artefato denominado *Sprint Backlog* e representam todas as tarefas detalhadas que deverão ser implementadas durante o tempo de iteração de uma *Sprint*, permitindo assim, a visualização do andamento dos trabalhos por todos os membros da equipe. Assim, o Scrum oferece uma melhor visão dos trabalhos a serem realizados durante o desenvolvimento, auxiliando numa melhor organização das tarefas, bem como no cumprimento dos prazos pré-estabelecidos. (SBROCCO; MACEDO, 2012, p. 168-169).

3.2 SURVEY

3.2.1 Planejamento

O objetivo geral da aplicação deste *Survey* é a coleta de requisitos necessários para o desenvolvimento do aplicativo *Android* através da aplicação de questionários na Instituição de Ensino Superior.

3.2.2 Seleção de participantes e tamanho da amostra

O processo de seleção dos participantes ocorreu na própria Instituição de Ensino Superior, a Faculdade Sete de Setembro – FASETE, cuja população selecionada foram os estudantes dos períodos mais avançados: 6º, 7º e 8º período do Curso de Bacharelado em Sistemas de Informação. O tamanho da amostra foi um total de 25 participantes.

3.2.3 Metodologia

O questionário foi desenvolvido na ferramenta *Google Forms* e disponibilizado por meio da *internet*. Foram desenvolvidas perguntas referentes ao período dos entrevistados e perguntas pré-elaboradas relacionadas a coleta de requisitos necessários para o desenvolvimento do aplicativo móvel.

3.2.4 Aplicação

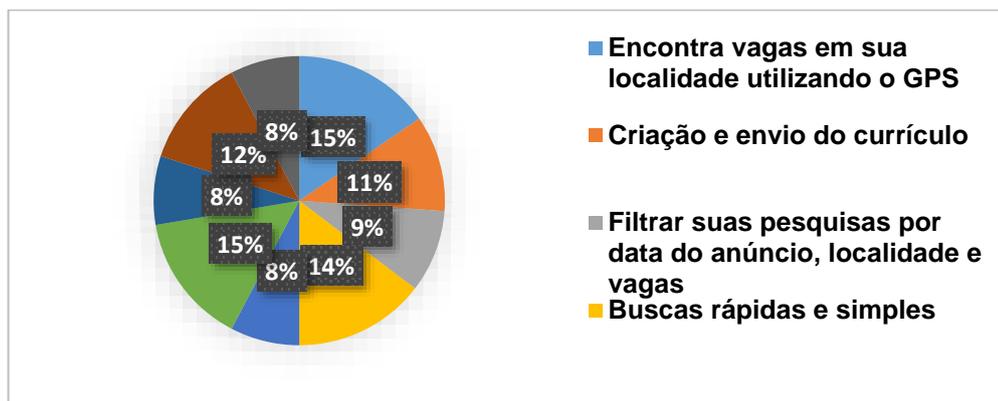
O questionário foi aplicado em sala de aula no dia 18 de novembro de 2017 para as turmas do 6º, 7º e 8º períodos do Curso Bacharelado em Sistemas de Informação. Após a aplicação, foram contabilizadas 25 respostas devidamente preenchidas, não havendo questões em branco. Com o intuito de validar os dados, averiguou-se a quantidade de participantes na sala de aula e o número de respostas, confirmando assim, o preenchimento do *Survey*.

4 RESULTADOS

4.1 Levantamento dos requisitos

Na análise sobre o levantamento de requisitos, constatou-se que (42%) dos entrevistados consideram como funcionalidades mais importantes: Encontrar vagas em sua localidade utilizando o GPS, Buscas rápidas e simples e Notificações de vagas que correspondem aos seus critérios. A funcionalidade Filtrar suas pesquisas por data do anúncio, localidade e vagas identificou-se (12%) e Criação e envio do currículo (10%). Apenas (6%) dos entrevistados consideram a Criação e uso do currículo como imprescindível. A seguir, a figura 4 ilustra o grau de importância dos requisitos adquiridos através da pesquisa.

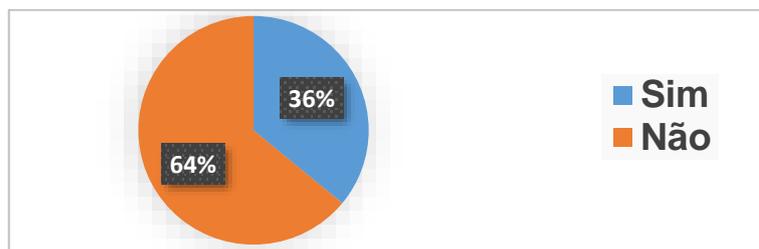
Figura 4. Funcionalidades imprescindíveis num *app* de busca de empregos



Fonte: AUTORES (2017)

Algumas funcionalidades foram sugeridas pelos entrevistados. Analisando os resultados, (64%) dos universitários não sugeriram nenhuma outra funcionalidade. Os que indicaram funcionalidades correspondem a (36%) das respostas. Os dados sobre as sugestões obtidas com o *Survey* seguem na figura 5 a seguir.

Figura 5. Sugestão de funcionalidades



Fonte: AUTORES (2017)

4.2 Análise dos resultados

A análise sobre os dados obtidos permitiu a identificação das funcionalidades mais importante para o desenvolvimento do aplicativo, sendo: Encontrar vagas em sua localidade utilizando o GPS, Buscas rápidas e simples, Notificações de vagas que correspondem aos seus critérios e Descrição completa das empresas e vagas, abrangendo (54%) dos requisitos indispensáveis. A respeito das funcionalidades sugeridas predominou o Não com (64%) das respostas.

4.3 Ameaças à validade

Analisando o processo de respostas do *Survey* pode apresentar eventuais problemas como a não compreensão de alguma questão pelos entrevistados, proveniente da não supervisão. Lidando com a ausência de experiência desses universitários sobre o tema foi escolhido uma parcela referente aos períodos do curso mais avançado, desta forma eliminando qualquer falta de conhecimento.

4.4 Aplicativo empregosal

Segundo Sbrocco e Macedo (2012, p.50) “Os requisitos funcionais representam as regras de negócio, ou seja, contêm as funcionalidades que o sistema deve atender”. A seguir, a figura 6 representa os requisitos funcionais necessários para a construção do aplicativo móvel.

Figura 6. Requisitos funcionais

| REQUISITOS FUNCIONAIS | |
|-----------------------|-------------------------|
| Identificador | Descrição |
| RF001 | Fazer <i>login</i> |
| RF002 | Manter usuário |
| RF003 | Manter empresa |
| RF004 | Manter vagas |
| RF005 | Manter currículo |
| RF006 | Gerar currículos |
| RF007 | Ofertar vagas |
| RF008 | Pesquisar vagas |
| RF009 | Candidatar à vaga |
| RF010 | Acompanhar candidaturas |
| RF011 | Analisar candidatos |

Fonte: AUTORES (2017)

A seguir, a figura 7 ilustra os requisitos não-funcionais do aplicativo. Ou seja, as restrições impostas sobre as funções oferecidas pelo sistema, ligadas ao domínio da solução (SBROCCO; MACEDO, 2012, p.50).

Figura 7. Requisitos não-funcionais

| REQUISITOS NÃO-FUNCIONAIS | |
|---------------------------|---|
| Identificador | Descrição |
| RNF001 | O aplicativo suporta no máximo 100 pessoas conectadas simultaneamente através do plano gratuito do <i>Firebase</i> . Caso precise aumentar quantidade de acessos, é necessária a obtenção de um plano pago. |
| RNF002 | Compatível com sistema operacional <i>Android</i> a partir da versão 4.0.3 |
| RNF003 | Atualização automática dos dados para todos os usuários conectados de forma online no <i>app</i> . |
| RNF004 | Design responsivo nas interfaces gráficas. |

Fonte: AUTORES (2017).

5 DIAGRAMA DE CASOS DE USO

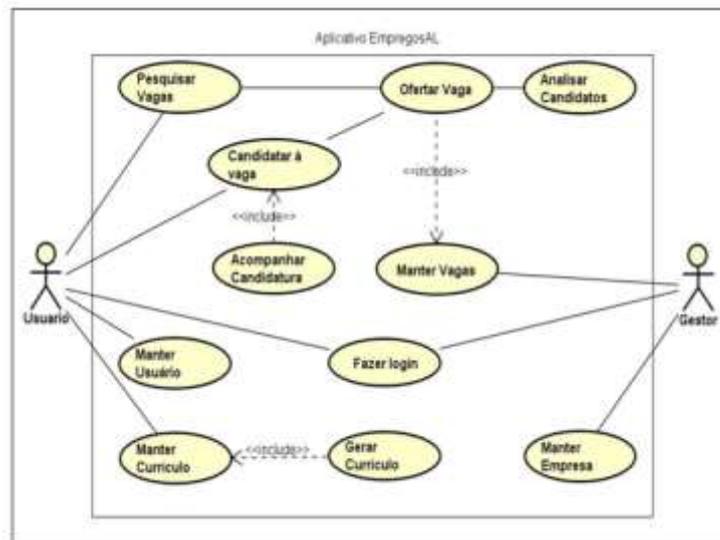
Conforme ressalta Guedes (2008, p.48), o diagrama de casos de uso tem como objetivo “apresentar uma visão eterna geral das funções e serviços que sistema deverá oferecer aos usuários[...]”. Sendo assim, são as funcionalidades do aplicativo considerando os requisitos descritos, anteriormente, no diagrama de casos de uso. Além disso, consta todas as operações que podem ser feitas pelos atores. Com a

Denise Xavier Fortes | Ricardo Azevedo Porto | Zenilton de Sá Neves

identificação dos casos de usos através das etapas anteriores, foi elaborado o diagrama de casos de uso do *app* em questão.

No diagrama abaixo, existem dois atores do sistema (Usuário e Gestor), ambos realizam *login* no sistema. O gestor tem acesso ao cadastramento de vagas e pode ofertá-las para os usuários se candidatarem, podendo também ter acesso a análise dos perfis dos candidatos. O usuário tem acesso a seu cadastro e currículo, tendo a possibilidade de pesquisar vagas, se candidatar nas vagas e acompanhar sua candidatura. A seguir, a Figura 8 apresenta o diagrama de casos de uso para o aplicativo proposto.

Figura 1. Diagrama de casos de uso do aplicativo



Fonte: AUTORES (2017).

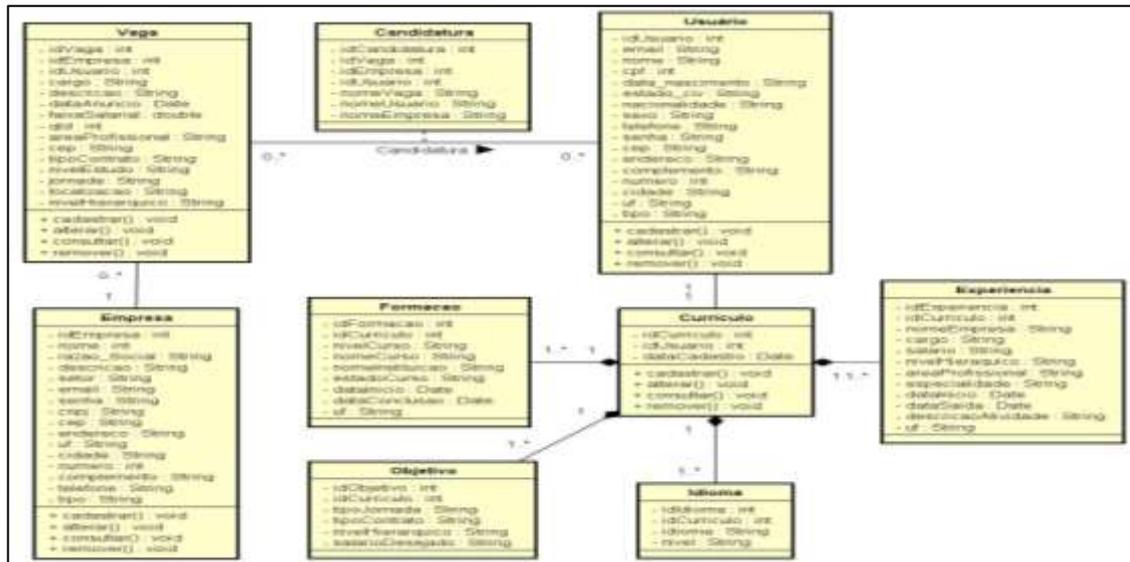
5.1 Diagrama de classes

O diagrama de classes segundo Guedes (2008, p.75) tem como objetivo:

[...] permitir a visualização das classes que comporão o sistema com seus respectivos atributos e métodos, bem como demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si.

Deste modo, o diagrama de classes permite uma visão estrutural do aplicativo móvel e ilustra um conjunto de classes, com todas as suas relações, atributos e algumas possíveis operações que podem ser aplicadas nas tabelas. A Figura 9 apresenta o diagrama de classes que foi elaborado para o aplicativo.

Figura 9. Diagrama de classes



Fonte: AUTORES (2017).

Por meio desse diagrama é possível verificar que a classe empresa é responsável pela criação de vagas, no qual a classe usuário tem acesso e gera através da relação à classe associativa candidaturas. O usuário tem permissão de administrar a classe currículo e as demais classes que o compõe.

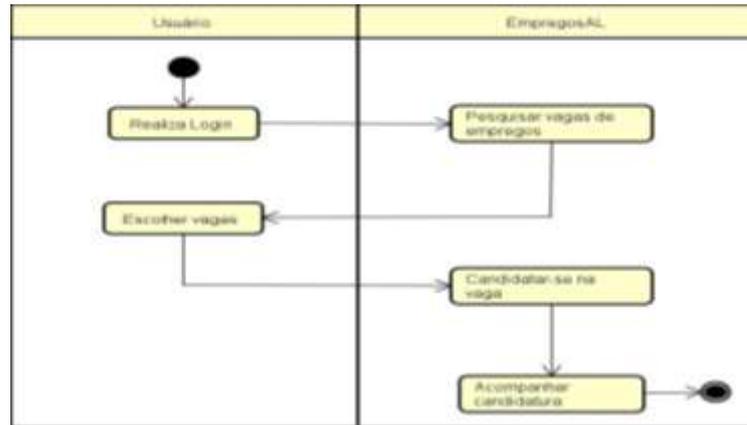
5.2 Diagrama de atividades

A respeito do diagrama de atividade, Pressman (2011, p.737) ressalta que:

O diagrama de atividade mostra o comportamento dinâmico de um sistema ou parte de um sistema através do fluxo de controle entre ações que o sistema executa. Ele é similar a um fluxograma exceto que pode mostrar fluxos concorrentes.

Portanto, o diagrama de atividades é responsável por ilustrar as ações necessárias, bem como os atores envolvidos nestas para uma determinada operação. A seguir, na figura 10 representa o fluxo de candidatura de vagas, a atividade inicia com o usuário realizando o *login* no sistema, após isso, realiza-se a pesquisa de vagas de empregos e na terceira etapa o usuário escolhe em qual vaga ele quer se candidatar. Por fim, após escolher e se candidatar a vaga, o usuário poderá acompanhar sua candidatura na vaga selecionada.

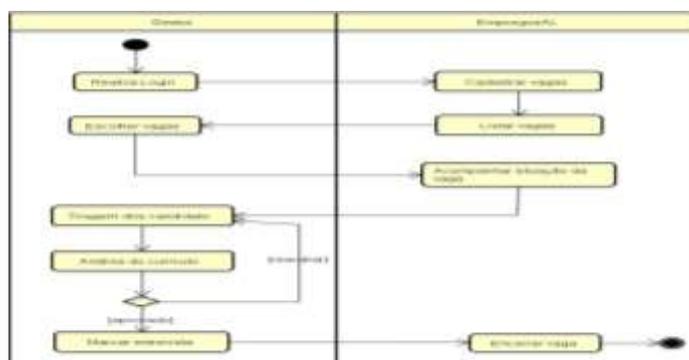
Figura 10. Diagrama de atividades 1



Fonte: AUTORES (2017).

Na Figura 11, descreve-se o fluxo de oferta de vagas e seleção dos candidatos. Através desta, percebe-se que a atividade começa com o gestor fazendo o *login* no aplicativo. Na etapa seguinte é feito o cadastramento de vagas da empresa. Na terceira etapa, o gestor lista e na quarta etapa escolhe algumas das vagas que cadastrou, em seguida, acompanha a situação. Na sexta etapa acontece a triagem dos candidatos pelo gestor que segue para etapa de análise do currículo. Nesta etapa o currículo do candidato é analisado e tomada uma decisão, caso o currículo esteja fora do perfil, é feita uma reavaliação e retorna para triagem novamente. Caso seja aprovado, segue para a próxima etapa marcar entrevista e encerra a vaga, finalizando o diagrama.

Figura 11. Diagrama de atividades 2



Fonte: AUTORES (2017).

5.3 Funcionamento do aplicativo

Nesta seção serão apresentadas as telas do aplicativo EmpregosAL juntamente com uma breve explicação de seu funcionamento. A Figura 12, ilustra a tela para o usuário com as vagas cadastradas por parte das empresas, onde é possível realizar a candidatura. Caso o usuário queira visualizar mais informações sobre a vaga, é possível clicar no *card* da vaga selecionada e uma nova tela com detalhes da vaga será aberta.

Figura 12. Lista de Vagas



Figura 13.2 Gerenciamento de currículo



Fonte: AUTORES (2017).

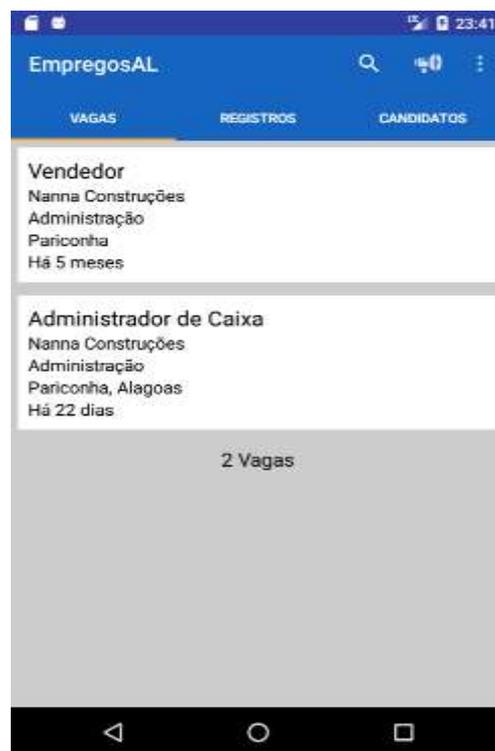
A Figura 13 representa a tela de gerenciamento do currículo do usuário, onde é possível realizar o cadastramento com detalhes de suas experiências, objetivos, formações e idiomas falados.

Na Figura 14, representa a tela de cadastramento de vagas para as organizações. Nesta tela é possível para o gestor anunciar suas vagas e expor suas exigências para determinada função.

Figura 14. Tela de cadastro de vaga



Figura 15. Lista de vagas cadastradas



Fonte: AUTORES (2017).

A Figura 15 ilustra a tela principal do *app* para organizações, nesta é possível ver as vagas cadastradas pelas empresas.

A Figura 16 ilustra a lista de candidatos para uma determinada vaga anunciada pela empresa.

Figura 16. Lista de candidatos para a vaga



Figura 17. Detalhes do candidato

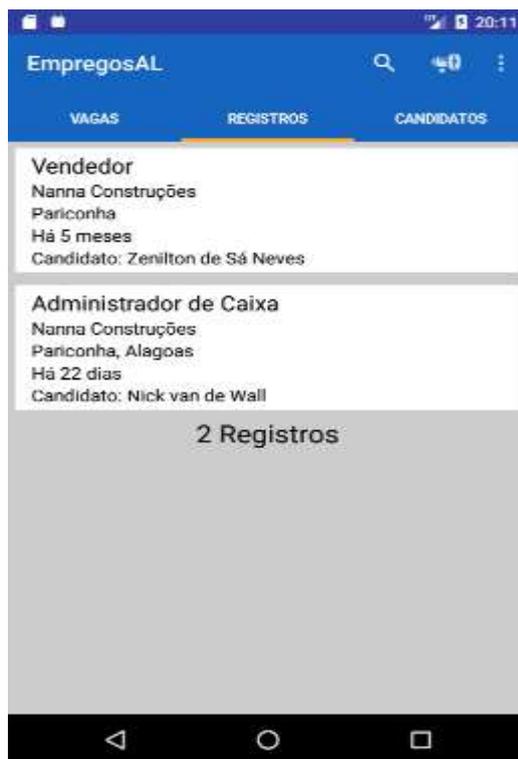


Fonte: AUTORES (2017).

A Figura 17 ilustra a tela com detalhes do candidato, esta é responsável por dispor informações do candidato e seu currículo para análise do perfil da vaga.

A Figura 18 representa a tela de registro de uma determinada vaga finalizada pela empresa. É possível clicar no *card* e ver as informações da vaga e o contato do candidato selecionado para realizar a entrevista.

Figura 18 - Registros da vaga finalizada



Fonte: AUTORES (2017).

6 CONSIDERAÇÕES FINAIS

A partir do uso destas tecnologias e metodologias ágeis de desenvolvimento de *software* neste trabalho, foi possível desenvolver um aplicativo para a plataforma *Android* capaz de auxiliar pessoas a encontrarem novas oportunidades profissionais, independente da sua localização.

Com o aplicativo instalado no dispositivo móvel, propõem-se ao usuário gerenciar suas informações pessoais, candidatar-se e acompanhar a candidatura das suas vagas de empregos, como também propôs aos gestores o cadastramento das vagas disponíveis, em suas organizações e, analisar os perfis dos candidatos admitidos na concorrência da vaga.

REFERÊNCIAS

BARBOSA, Helcio et al. **Metodologias ágeis para desenvolvimento de aplicações móveis**. Revista Pensar Tecnologia, [S.L.], v. 4, n. 2, jun. 2017. Disponível em: <<http://revistapensar.com.br/tecnologia/artigo/no=a124.pdf>>. Acesso em: 25 mai. 2017.

CHENG, Fu. **Build Mobile Apps with Ionic 2 and Firebase: Hybrid Mobile App Development**. Sandringham, Auckland, New Zealand: Apress, 2017. 247 p.

FIREBASE. **Firestore Authentication**. Disponível em: <<https://firebase.google.com/docs/auth/?hl=pt-br>>. Acesso em 18 de ago. 2017.

FIREBASE. **Firestore Realtime Database**. Disponível em: <<https://firebase.google.com/docs/database/?hl=pt-br>>. Acesso em 18 de ago. 2017.

GUEDES, Gilleanes T. A. **UML: Uma abordagem prática**. 3ª Edição. São Paulo. Novatec Editora, 2008. 336 p.

LECHETA, Ricardo R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o android SDK**. 2ª Edição. São Paulo. Novatec Editora, 2010. 607 p.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7ª Edição. AMGH Editora. Porto Alegre, 773 p.

QUEIRÓS, R. **Android: Desenvolvimento de aplicações com Android Studio**. Editora de informática LDA, 2016. 295 p.

SAREEN, Pankaj. **Cloud Computing: Types, Architecture, Applications, Concerns, Virtualization and Role of IT Governance in Cloud**. *International Journal of Advanced Research in Computer Science and Software Engineering*. Baddi University of Emerging Sciences & Technology, India. Disponível em: <http://www.academia.edu/download/35864304/virtualization_introduction.pdf>. Acesso em 12 de ago. 2017.

SBROCCO, José Henrique Teixeira Carvalho; MACEDO, Paulo Cesar. **Metodologias Ágeis: Engenharia de Software sob Medida**. 1 ed. São Paulo: Érica, 2012. 256 p.