

# Aplicativo para o auxílio do ensino de Inglês

Ícaro dos A. Alvarenga<sup>1</sup>, Marcelo C. P. Santos<sup>2</sup>

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais –  
*Campus Juiz de Fora*

<sup>2</sup>Núcleo de Informática - Instituto Federal de Educação, Ciência e Tecnologia do  
Sudeste de Minas Gerais – *Campus Juiz de Fora*

icaro\_alvarenga@hotmail.com, marcelocpsantos@gmail.com

**Abstract.** In order to develop an application to assist the teaching of English language, a research of the best fitting models for mobile development was made, including the tools that facilitate that process. The outcome of this project is a fully functional application, ready to use.

**Resumo.** Com o objetivo de desenvolver um aplicativo para auxiliar o ensino da língua inglesa, foram pesquisados os modelos de desenvolvimento que mais se adequam a um desenvolvimento mobile e as ferramentas que facilitam tal desenvolvimento. O resultado de tal projeto é um aplicativo totalmente funcional e pronto para ser posto em uso.

## 1. Introdução

O aumento da popularidade de dispositivos de computação móvel gera uma grande oportunidade para a melhoria dos métodos de ensino, em particular o de línguas, tornando o aprendizado mais dinâmico e interessante para o aluno.

A motivação para o presente trabalho nasceu da observação de que outros aplicativos semelhantes possuem muitas possibilidades de melhora. O *Duolingo*® não tem suporte a perguntas com vídeo, o *Memrise*® só conta com um conteúdo pré-estabelecido e *Rosetta Stone*®, na forma gratuita, permite poucas atividades (mais diferenças são abordadas no decorrer do artigo). Além disso todos os aplicativos semelhantes têm o objetivo de ensinar a língua sem o auxílio de um professor, diferente da nossa proposta que é uma ferramenta para auxiliar o professor nessa tarefa.

Decidiu-se, portanto, pelo desenvolvimento de um aplicativo que tenha suporte a vários tipos de mídia, possibilidade de criação e edição de perguntas por parte dos professores e totalmente gratuito possibilitando assim que possa ser utilizado em várias escolas e melhorando a qualidade do ensino.

## 2. Metodologia

O desenvolvimento mobile se diferencia do desenvolvimento de *desktop* e *web* por tratar com dispositivos com limitações de hardware como tela, capacidade computacional, energia e conectividade. Tais quesitos precisam ser levados em conta na elaboração de um modelo. Jabangwe, Edison e Duc. MASAM (Jeong et al. 2008), D-Mobile (Abrahamsson et al. 2004) e Abordagem 3D (Pani e Mishra 2016) são exemplos de metodologias propostas na literatura que lidam com essas características.

Com a intenção de selecionarmos uma metodologia de projeto para o desenvolvimento do aplicativo desejado, estudamos diversas propostas que abordassem Levantamento de Requisitos, Projeto, Codificação e Testes. Foram consideradas, estudadas e avaliadas as seguintes metodologias propostas na literatura, que nem sempre abordavam todas as fases de desenvolvimento desejadas. Na Tabela 1 vemos uma relação das metodologias estudadas, com observações sobre as fases de projeto abordadas. Na Tabela 1, um “X” indica que a característica é abordada na proposta e um “O” indica que não é considerada. Na sequência, tecemos maiores comentários sobre cada uma das propostas.

Artigo	Requisito	Designer	Codificação	Teste
de Sá e Carriço (2008)	Geração de cenário e coleta de dados	Prototipagem	X	O
Abramhamson et al. (2004)	Inicialização	Produção	X	X
Binsaleh and Hassan (2011)	Exploração	X	Lançamento iterativo	X
Heredia et al. (2013)	Definição de conceito e análise de mercado	O	Ciclos iterativos e lançamento	X
Jeong et al. (2008)	X	X	X	X
Lee et al. (2015)	X	X	X	X
Losada et al. (2012)	Lista de objetivos do usuário	X	X	X
Nosseir et al. (2012)	X	X	X	X
Rahimian and Ramsin (2008)	Geração da ideia	Detalhe de análise e designer	Ciclo adaptativo	X
Škrabálek et al. (2013)	Especificações	X	X	X
Vithani and Kumar (2014)	X	X	X	X
Zeidler et al. (2008)	Geração da ideia, aspectos legais e	UX designer	X	X

	pesquisa de mercado			
Pani and Mishra (2016)	X	X	X	X
Kang et al. (2017)	X	Arquitetura, navegação e design das páginas	X	X
Mota et al. (2017)	O	Designer interativo	X	O
Kim (2018)	O	UML	O	O
Queirós et al. (2017)	O	MVP	O	O
Majchrzycka and PoniszewskaMaranda (2017)	Requisitos de segurança	Designer seguro	O	O
Zaragoza and Kim (2017)	Análise e investigação	X	X	X
Martínez et al. (2018)	X	X	X	X

**Tabela 1. Adaptada do artigo (Software engineering process models for mobile app development: A systematic literature review)**

### **2.1 - Jeong et al. (2008) - “Development Process of Mobile Application SW Based on Agile Methodology”**

Apresentado por Jeong et al. (2008), leva em conta que a maioria das empresas de desenvolvimento mobile são de pequeno porte, onde a velocidade de lançamento dos aplicativos é essencial para uma permanência no mercado. Define os recursos como papéis (um conjunto de habilidades, competências e responsabilidades de um indivíduo ou grupo), tarefa (é um objetivo de trabalho, designada a um papel específico, normalmente de poucas horas a alguns dias de trabalho efetuado somente por uma pessoa ou um pequeno grupo) e produto (é o resultado da tarefa). Seu desenvolvimento é dividido em 4 fases: preparação para o desenvolvimento (diálogo com o cliente para definir seus objetivos), incorporação (com o uso de um banco de projetos a equipe compara projetos já desenvolvidos com as necessidades do cliente, apresenta os projetos similares ao cliente e realiza as mudanças necessárias), desenvolvimento do produto (o projeto é desenvolvido e apresentado ao cliente gradualmente, se não existir um projeto obtido no fase anterior ele é totalmente desenvolvido, se existir somente as mudanças são implementadas) e comercialização (produto é testado, entra em conformidade de políticas de cada região onde será comercializado e passa por um teste de aceitação final por parte do cliente) após todos esses passos, o produto vai para distribuição.

## **2.2 - Lee et al. (2015) - “Implementation of Delivery Application using a Development Method for Mobile Applications”**

Idealizado por Lee et al. (2015). As fases de desenvolvimento adotadas são 5, análise de requisitos, designer de arquitetura (definição da arquitetura), designer de navegação (navegação no aplicativo e transmissão de dados entre páginas), designer de página (layout e funções das páginas) e implementação e teste. Duas notações principais são componente e conector. Componente representa um módulo funcional do sistema enquanto conector representa as interações entre componentes. Tais notações são usadas nos 3 diagramas produzidos após a análise e designer da aplicação, são eles: diagrama de designer da arquitetura, diagrama de designer de navegação e diagrama de designer de detalhe da página.

## **2.3 - Nosseir et al. (2012) - “Mobile Development Process Spiral”**

Desenvolvida por Nosseir et al. (2012) caracteriza-se por ser um modelo orientado a usabilidade. Utiliza como base o desenvolvimento em espiral com o aprimoramento contínuo dos requisitos seguido pelo designer e implementação. Conta com 5 iterações de 7 fases cada sendo elas: identificação dos usuários, tarefas e contextos da aplicação, definição de métricas para medir cada atributo de usabilidade, priorizar os atributos de usabilidades, especificar valores aceitáveis e ideias para as métricas, mensurar atual valor, avaliar cada atributo, comparar a avaliação com a da fase anterior. O quesito humano em uma programação torna quase impossível o desenvolvimento de um aplicativo com poucos erros de usabilidade na primeira tentativa, por isso o uso do modelo espiral para minimizar os erros no resultado final.

## **2.4 - Vithani e Kumar (2014) - “Modeling the Mobile Application Development Lifecycle”**

Uma metodologia desenvolvida por Vithani e Kumar (2014) com foco no ciclo de vida de desenvolvimento de software. Conta com 7 fases, identificação, designer, desenvolvimento, protótipo, teste e manutenção. Outra característica de dispositivos mobile levada em conta é a vida útil, funcionalidades complexas, menos interfaces físicas, mais números de telas, bateria, uso de memória, desenvolvimento híbrido e manutenção.

## **2.5 - Pani e Mishra (2016) - “A Novel Approach for Mobile Native App Development Using Ontological Design”**

É uma proposta de desenvolvimento de Pani e Mishra (2016) baseada em uma abordagem 3D levando em consideração o contexto, ontologia genérica e as fases de desenvolvimento.

Uma ontologia define os termos usados para descrever e representar um domínio, isto é, é uma descrição de conceitos e relacionamentos entre eles que pode ser usada por pessoas ou agentes de software para compartilhar informações dentro deste domínio. Por estas características, a ontologia é uma das tecnologias chaves para a implementação da Web Semântica. OWL é uma linguagem de marcação semântica para publicação e compartilhamento de ontologias, projetada para descrever classes e relacionamentos entre elas, que é atualmente recomendada pelo W3C. (Lima e Carvalho, 2005, p. 1).

As fases de desenvolvimento são 4, Análise de requisitos, Designer, Teste e Implementação. Já os contextos são, do dispositivo, do usuário, de mobilidade e social.

Tem como objetivo facilitar a troca de informações entre aplicativos do mesmo domínio (Agricultura, Medicina, etc).

## **2.6 - Martínez et al. (2018) - “Proposal for an Integrated Framework for Mobile Applications Development”**

Proposta trazida por Martínez et al. (2018), levando em conta que os processos de desenvolvimento *mobile* além de poucos não abrangem todos os aspectos particulares de um desenvolvimento para dispositivos móveis dá-se origem essa proposta. Um framework dividido em 3 camadas sendo: tipo de atividade (definição do conceito do aplicativo, utilização de técnicas como *storyboard*, entrevistas e pesquisas são usadas para identificar a arquitetura e requisitos do aplicativo), seguido por uma fase clássica de *SCRUM*, durante o *SCRUM* a última camada é aplicada. Durante o planejamento de implementação de funcionalidades os problemas chave *mobile* (conectividade, interoperabilidade, flexibilidade, energia, diferença de dispositivos, plataforma e segurança dos dados) são identificados e trabalhados a cada iteração do *SCRUM*.

## **2.7 - Critério de seleção das metodologias**

Jeong et al. (2008) é muito focado na agilidade de produção. Ele dá grande ênfase ao reaproveitamento de projetos desenvolvidos previamente pela empresa, o que não é a nossa realidade, portanto, não foi considerado adequado ao projeto.

Já o modelo de Nosseir et al. (2012) é focado na usabilidade, gastando muito tempo e recursos para refinar ao máximo esse quesito, por não ter grande necessidade de uma usabilidade tão refinada e o tempo escasso esse modelo foi descartado.

O modelo proposto por Pani e Mishra (2016) utiliza recursos focando na possibilidade de troca de informações entre aplicativos, por se tratar de um aplicativo sem a necessidade de tais trocas, foi descartado.

Chegamos então nos três últimos modelos analisados, o de Lee et al. (2015), Vithani e Kumar (2014) e Martínez et al. (2018), os três bem semelhantes, mas como Vithani e Kumar (2014) é o mais simples, voltado inclusive para o desenvolvimento por apenas uma pessoa, foi o que melhor se adaptou às nossas necessidades.

## **3. Sistema desenvolvido**

O sistema proposto foi efetivamente desenvolvido seguindo o modelo selecionado seguindo cada uma das 7 fases a seguir.

### **3.1. Fase de identificação**

A partir de uma pesquisa foram identificados alguns aplicativos que têm a mesma proposta de auxiliar o ensino de língua inglesa, entre eles estão:

Um dos mais populares é o aplicativo *Duolingo*®, que oferece recursos para aprendizado de diversos idiomas, tais como francês, inglês e espanhol através de uma proposta interativa por meio de perguntas e respostas em que o usuário contabiliza pontos, capacitando-o a um próximo nível. Embora seja gratuito, por se tratar de um software proprietário, i.e., detido por uma empresa, alguns recursos são limitados não permitindo o cadastro de perguntas além das oferecidas pela plataforma. O *Duolingo*® não oferece perguntas com vídeos.

O Memrise® também oferece conteúdo de diversos idiomas com duas versões: gratuita e paga. Possui uma proposta que visa explorar a capacidade de memorização do usuário para avançar nas lições. As lições oferecidas possuem conteúdo pré-definido e temas específicos, como os cursos voltados à preparação de exames de certificação e para o meio profissional.

O Rosetta Stone® mostrou-se o mais completo em relações às perguntas contextualizadas com mídias digitais. Há suporte para vídeos, imagens, entrada e saída de áudio, além de textos. Contudo, sua versão gratuita permite um número limitado de atividades que podem ser realizadas não deixando que o aluno avance com os estudos na plataforma.

Com base nos aplicativos identificados anteriormente foi possível ter a elaboração de um processo que dispõe das seguintes atividades: O sistema deve diferenciar um aluno de um professor a partir do login, e mostrar uma página de lições diferenciado para cada um. Deve permitir o professor cadastrar novas lições e adicionar perguntas a essas lições, perguntas do tipo verdadeiro ou falso, múltipla escolha (com 4 a 8 opções) com imagem, vídeo e áudio. Cada pergunta deve ter um campo de título, descrição, observação e as opções de resposta (somente texto).

O aluno deve ser capaz de se cadastrar, colocando o nome e a turma, e logar após o cadastro. Escolher uma das lições disponíveis e responder as questões sequencialmente da lição. Após cada resposta, é mostrada uma tela com o ranking de todos os alunos da mesma turma que estão respondendo a lição assim como a resposta dada e a resposta correta. Cada acerto correto aumenta a pontuação desse aluno. Ao final de todas as perguntas o aluno espera todos os restantes terminarem enquanto acompanha uma tela com o ranking.

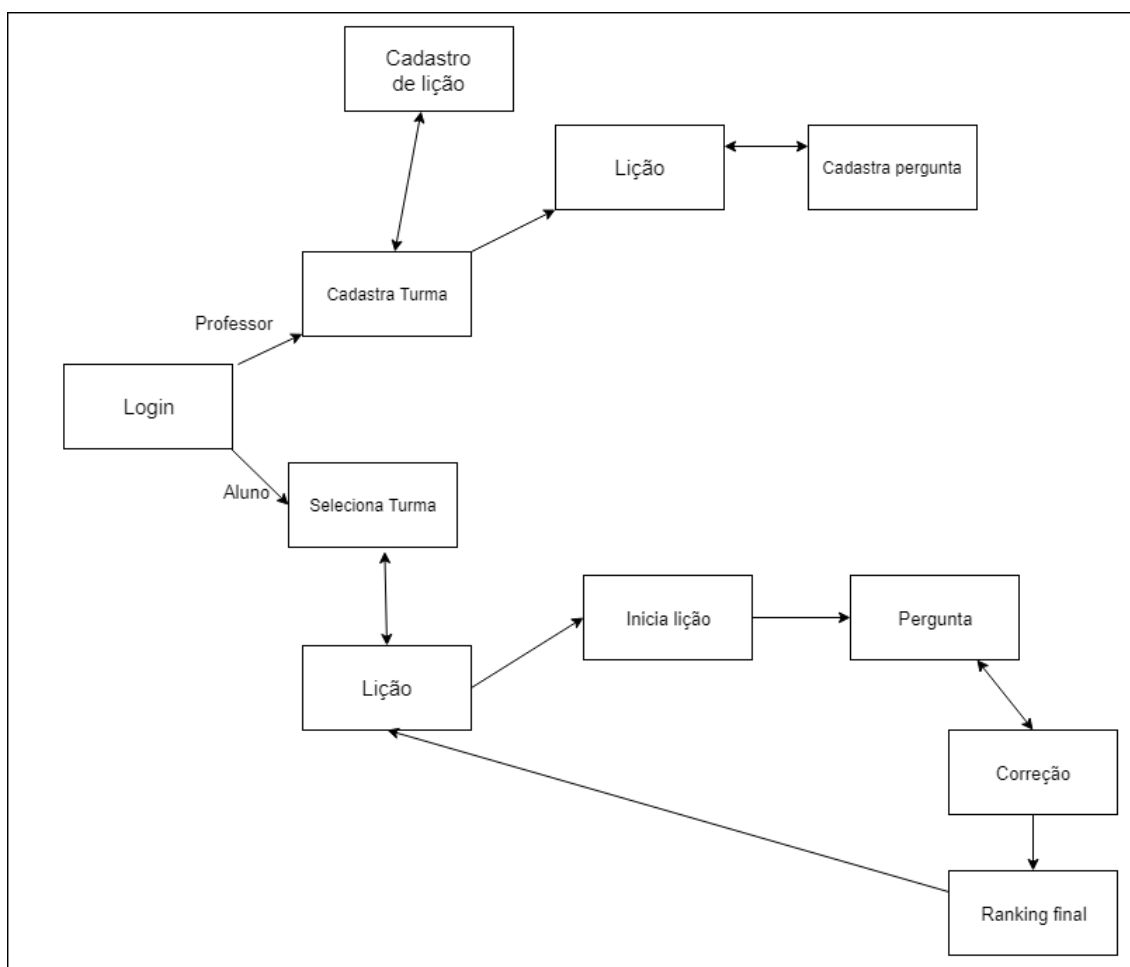
O sistema deve ser capaz de retornar a última questão respondida pelo aluno caso haja um reinício na lição, mantendo assim a pontuação já obtida até o momento.

### **3.2. Fase de designer**

A ferramenta foi definida como distribuição gratuita e desenvolvido usando a ferramenta *Visual Studio Code*, framework *Ionic 4* em conjunto com *Angular 7*, linguagem de programação *TypeScript* e banco de dados *Firebase*™.

O *Ionic 4* foi utilizado por proporcionar um desenvolvimento híbrido, ou seja, uma alternativa ao desenvolvimento nativo, sendo assim tem suporte para dispositivos *Android*™ da versão 4.4 e *iOS*® versão 8. Outra forma de descrever os recursos usados foi com o *Firebase*™ que tem como objetivo dentro do programa desenvolvido toda parte de autenticação (*Firebase Auth*™ possibilita o envio de *e-mail* de confirmação, troca de senha, possui criptografia e uma implementação extremamente simples), armazenamento (*Firebase Cloud Storage*™ conta com suporte para todos os tipos de arquivo, conta com regras de acesso, somente pessoas autenticadas pelo *Firebase Auth*™ podem ler e gravar dados) e base de dados (*Cloud Firestore*™). O banco de dados *Cloud Firestore*™ é um banco de dados não relacional o que torna a manipulação de dados mais simples, tem uma melhor escalabilidade e velocidade de acesso, além disso trabalha através de uma API orientada a objetos, facilitando assim o uso das informações, em contrapartida possui estrutura menos rígida e não é normalizado. Foi escolhido o fornecido pela *Firebase*™ por ser uma empresa da *Google LLC*, uma grande capacidade de acesso no modo gratuito e sua integração com outras ferramentas da empresa.

Com base nas atividades básicas do aplicativo foi gerado o seguinte fluxograma.



**Figura 1. Fluxograma funcional do software**

A Partir do fluxograma podemos dar início a próxima etapa.

### 3.3. Fase de Desenvolvimento

O desenvolvimento foi realizado utilizando técnicas de desenvolvimento ágil como a divisão do desenvolvimento em pequenas funções e o desenvolvimento dessas funções em um espaço de tempo de uma a duas semanas.

### 3.4. Fase de protótipo

Após o desenvolvimento de cada funcionalidade ela é testada e enviada para o feedback do orientador e então incorporada ao programa, e reiniciando o ciclo de desenvolvimento teste e feedback.

### 3.5. Fase de teste

Com o desenvolvimento completo iniciam-se os testes em emulador/simulador seguido de exames em dispositivos reais. Considerando a dificuldade de obtenção de dispositivos iOS para teste o aplicativo estará sujeito a uma nova verificação em projeto futuro. Com os testes concluídos avançamos para a próxima fase.

### 3.6. Fase de lançamento

Com o feedback final do orientador, o aplicativo é “limpo” (remoção de comentários e logs), tem o ícone do aplicativo desenvolvido e sua adequação para cada plataforma possibilitando assim o seu acesso ao público. É importante ressaltar que existe uma restrição da vinculação do aplicativo devido taxas impostas pelas lojas de aplicativos (*Apple Store e Play Store*). Apesar dessas impossibilidades de uma grande abrangência o aplicativo pode circular em um grupo menor de usuários o que torna a próxima fase importante para a vida útil do aplicativo.

### 3.7. Fase de manutenção

Essa fase é um processo contínuo onde o *feedback* dos usuários podem gerar alterações no aplicativo, correção de problemas, melhorias de desempenho e segurança. Sendo assim a fase mais importante do ciclo de vida de um aplicativo.

## 4. Apresentação do aplicativo

O aplicativo foi desenvolvido e é totalmente funcional, projeto pensando inicialmente no ensino de língua, porém pode ser utilizado para qualquer outra matéria. Conta com uma interface simples direta com elementos como botão de ajuda que dispensam um manual como na Figura 1. Durante os testes, foi identificado a necessidade de diferenciar os professores e turmas por parte do aluno dando assim origem a uma tela de escolha de professor e turma presente na Figura 2. Outro quesito importante foi a proteção na navegação entre telas, somente pessoas autorizadas podem acessar determinadas telas facilitando assim a implementação para suporte web.

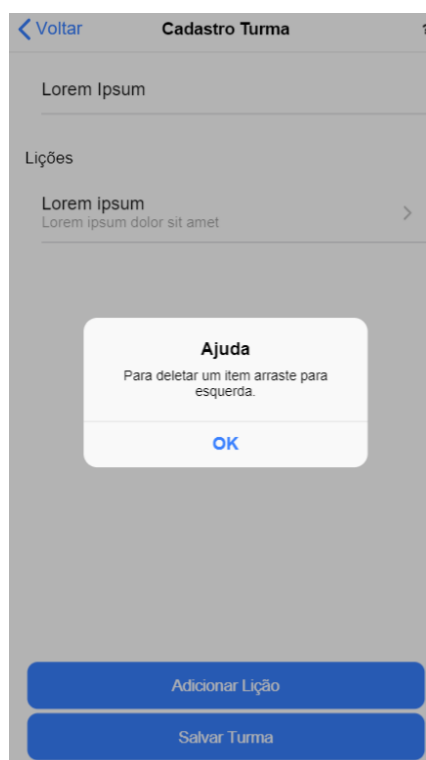


Figura 2. Tela de ajuda



☰ **Mudar de Turma**

---

SELECIONE PROFESSOR E TURMA

Professor Joaquim Castro ▾

---

Turma 1 eletrônica ▾

---

Salvar

**Figura 3. Mudança de turma**

O cadastro de exercícios por parte do professor possibilita a inserção de vídeos, imagens e áudio (Figura 4). A correção de cada resposta acontece durante a realização do exercício (Figura 5) e conta com um ranking mostrando a pontuação final de cada aluno que realizou aquela lição (Figura 6).

[← Back](#) **Multipla Escolha**

---

Titulo

---

Descrição

---

Escolher Imagem

Escolher Video

---

OPÇÕES	CORRETA
Opção 1	<input type="radio"/>
Opção 2	<input type="radio"/>
Opção 3	<input type="radio"/>
Opção 4	<input type="radio"/>

---

Salvar Pergunta

**Figura 4. Cadastro de pergunta**

[← Voltar](#) **Pergunta**

---

listas

---

qual marcador para listas ordenadas

---

CORREÇÃO

Errou 🚫

ol

ul

list

lo

---

Salvar Pergunta

**Figura 5. Correção da pergunta**



Figura 6. Ranking final da lição

## 5. Conclusão e trabalhos futuros

A proposta deste estudo foi o desenvolvimento de um aplicativo para o auxílio do ensino da língua inglesa. Tal proposta foi sugerida pela professora de línguas Patrícia Botelho do Campus Juiz de Fora do IF Sudeste MG através de um projeto de extensão. O aplicativo é totalmente funcional e está pronto para aplicações práticas. O diferencial desse aplicativo é sua praticidade, por se tratar de um aplicativo *mobile*, podendo ser usado para vários tipos de ensino como de línguas, tecnologia e etc. Além disso o seu suporte a vários tipos de mídias. Como trabalhos futuros é necessária a criação de um melhor gerenciamento das lições e perguntas por parte do professor como por exemplo o acesso às respostas dadas por cada aluno, aperfeiçoamento do *layout* para deixá-lo mais atrativo e a implementação das pequenas mudanças necessárias para uso web como o upload de arquivos.

## 6. Bibliografia

- Jabangwe, R., Edison, H. Duc, A., (2018). “Software engineering process models for mobile app development: Asystematic literature review.” *The Journal of Systems & Software* 145 (2018) pp. 98–111.
- Jeong, Y.J., Lee, J.H., Shin, G.S., (2008). “Development process of mobile application sw based on agile methodology.” 10th International Conference on Advanced Communication Technology, 2008. ICACT 2008. 1. pp. 362–366.
- Lee, J., Yang, S., Lee, R.Y., Kang, B., (2015). “Implementation of delivery application using a development method for mobile applications.” 3rd International Conference

on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI). pp. 132–136.

Martínez, D., Ferré, X., Marcillo, D., (2018). “Proposal for an integrated framework for mobile applications development.” International Conference on Information Theoretic Security. Springer, pp. 813–819.

Nosseir, A., Flood, D., Harrison, R., Ibrahim, O., (2012). “Mobile development process spiral.” Seventh International Conference on Computer Engineering Systems (ICCES). pp. 281–286.

Pani, S., Mishra, J., (2016). “A novel approach for mobile native app development using ontological design.” Int. J. Softw. Eng. Appl. 10 (9), 105–124.

Vithani, T., Kumar, A., (2014). “Modeling the mobile application development lifecycle.” Proceedings of the International MultiConference of Engineers and Computer Scientists. 1. pp. 596–600.

Lima, J., Carvalho, C., (2005). “Ontologias - OWL (Web Ontology Language).” Instituto de Informática Universidade Federal de Goiás. RT-INF\_004-05 - Relatório Técnico - 2005 - Junho