

# Sistema Web de Aplicação de Provas através de Dispositivos Móveis

Vinicius Bisognin Immich<sup>1</sup>, Bruno Boniati<sup>2</sup>

<sup>1</sup> Especialização em Projeto e Desenvolvimento de Aplicações para Dispositivos Móveis  
– Universidade Regional Integrada do Alto Uruguai e das Missões (URI)  
Rua Assis Brasil, 709 – Itapagé – Frederico Westphalen – RS – Brasil

<sup>2</sup> Instituto Federal Farroupilha (IFFar) Caixa Postal 169 – 98.400-000 – Fred.  
Westphalen/RS

vinibiso@gmail.com, bruno.boniati@iffarroupilha.edu.br

**Resumo.** *O presente trabalho tem como objetivo descrever o processo de desenvolvimento de um sistema web pelo qual um professor consegue criar executar uma avaliação objetiva através de dispositivos móveis. Existem várias soluções para integrar a tecnologia da informação na sala de aula, porém nenhuma de código aberto que use os modelos tradicionais de comunicação cliente servidor em conjunto com um aplicativo de aplicação de provas e que corrige a prova automaticamente ao receber um gabarito. Foram utilizadas tecnologias como Python/Django para o sistema web e Javascript/Cordova para o aplicativo de execução de provas.*

**Abstract.** *This paper describes the process of developing a web system by which a teacher can create a objective test and execute it through mobile devices. There are several solutions to integrate information technology in the classroom, but none open source that use traditional models of client-server communication in conjunction with an test runner app and is capable of automatically proofing a test when the answers are received. Technologies like Python/Django for the web system and Javascript/Cordova for the test running app were used.*

## 1. Introdução

É notável na sociedade moderna que os computadores, de todas as formas, se tornaram parte integral da vida de todos. As tecnologias permitem que algo seja encontrado instantaneamente, seja um documento, um assunto ou até mesmo um táxi. O progresso feito nos últimos 60 anos no que convém a agilizar processos e facilitar vidas pessoais bem como o dia-a-dia é indiscutível, porém o mesmo não pode ser dito quando se considera a utilização das mesmas tecnologias integradas nos ambientes de sala de aula e espaços escolares.

Nota-se que, em geral, métodos tradicionais de ensino encontram dificuldades em reter a atenção de boa parcela dos alunos e observa-se, até mesmo, que parte destes não se satisfaz com a forma como recursos tecnológicos vêm sendo utilizados dentro da sala de aula. O presente trabalho tem como objetivo descrever o processo de planejamento e implementação de um sistema baseado em tecnologias web para a criação e aplicação de avaliações através de dispositivos móveis usando um *framework* de criação de aplicativos híbridos, que usam tecnologias web para distribuir o mesmo aplicativo para diversas plataformas.

## 2. Estado da Arte

Existem várias formas de visualizar a utilização de recursos computacionais colaborando com o processo educacional, como por exemplo: a facilitação da pesquisa e estudo científico através de ferramentas de busca, a utilização de jogos ou outras atividades on-line que envolvem os alunos de forma mais dinâmica e colaborativa, além do hoje já bastante difundido estilo de Educação a Distância (EaD), provendo oportunidades iguais para pessoas que talvez não tenham acesso a uma universidade presencial em sua cidade ou região.

Estudos mostram que o uso de *tablets* em sala de aula não só ajudam no engajamento e motivação dos alunos, bem como no ensino de forma mais prática e ágil, mas também pode ajudar o aluno a tirar notas mais altas. Um destes estudos é conduzido na cidade de Auburn, Maine, nos Estados Unidos, onde 16 classes do jardim de infância em metade dos distritos da cidade receberam iPads para usarem por nove semanas. Ao total, 266 estudantes participaram, sendo 129 com iPads e 137 sem, sendo que todos os alunos foram testados antes e depois do uso dos mesmos. De acordo com os testes de alfabetização, ao total, as classes usando *tablets* superaram as que não usaram em todas as medidas de alfabetização testadas [Dalrymple, J., 2012]. Outro estudo conduzido na o MIT (Instituto de Tecnologia de Massachusetts), também nos Estados Unidos, tinha como objetivo o uso de um sistema *tablet-pc* para aumentar o nível de interações bem como o de aprendizagem dos alunos. O estudo foi conduzido durante uma classe de ciências da computação inicial e, comparando com a prova inicial, onde ainda não havia sido introduzido o sistema, apenas 37.7% dos alunos ficaram no top 10%. Ao final da classe, o aluno número foi para 44.4%, sendo que, desde a primeira prova, a nova metodologia e sistema foram inseridos [Koile. K and Singer, D., 2006].

## 3. Solução Proposta

Na tentativa de tentar explorar recursos tecnológicos aplicáveis diretamente na área educacional, propõe-se o desenvolvimento de uma aplicação para habilitar o professor a realizar avaliações (provas, exercícios, etc.), bem como sua correção, no ambiente de sala de aula, utilizando-se de dispositivos móveis. Para tal, serão utilizadas tecnologias web e plataformas *mobile*. A ideia é desenvolver um sistema web através do qual o professor consiga, de forma fácil, antes ou em sala de aula, criar e distribuir uma prova para *tablets* ou *smartphones* conectados na mesma rede que o dispositivo usado como servidor para o sistema (de forma independente do acesso à Internet). O sistema tem como requisito a capacidade do professor de criar e alterar provas, questões e alternativas, bem como integrar e comunicar para o aplicativo instalado nos dispositivos móveis a prova que está sendo executada. Além disso, ao receber o gabarito digital do aluno, o sistema deve calcular uma média ponderada do mesmo, baseada no gabarito criado pelo professor, e possibilitar aos estudantes a consulta imediata do resultado.

### 3.1. Arquitetura

Na Web é muito comum que um computador tido como “cliente” se comunique através de uma rede de dados, física ou não, com outro computador, o “servidor”. O servidor em questão pode executar vários programas, ao mesmo tempo que geralmente tem o objetivo de compartilhar recursos ou conteúdo com o cliente. Para que uma comunicação seja estabelecida, é preciso que o programa cliente faça a primeira requisição ao servidor, porém é importante lembrar que o servidor não manda requisições, apenas responde as do cliente.

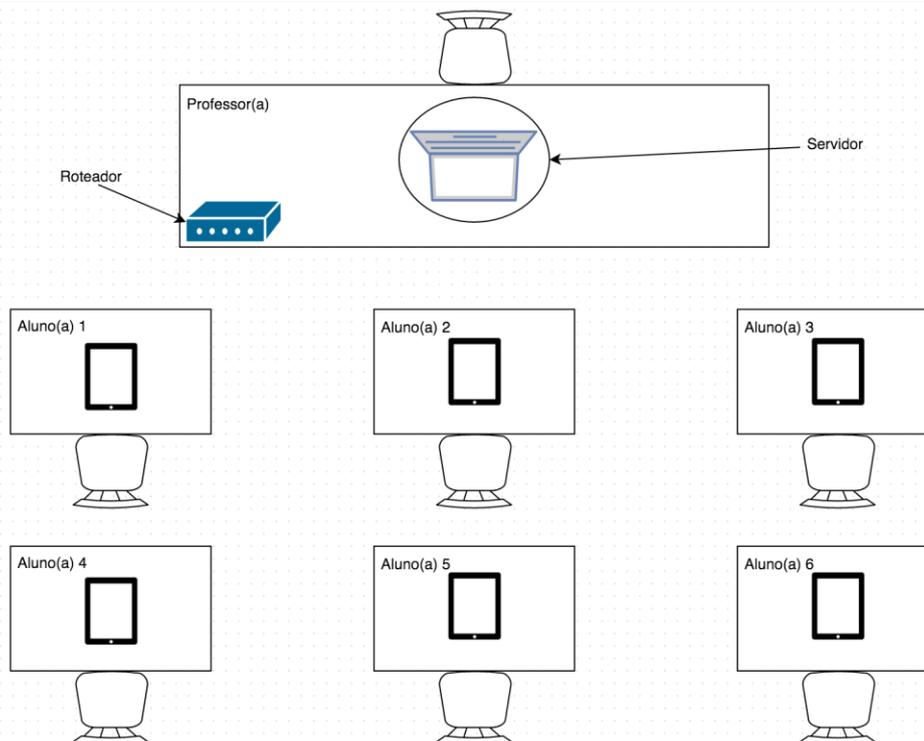


Figura 1. Exemplificação da arquitetura usada em sala de aula

Para a implantação do sistema proposto, utilizando-se de uma arquitetura cliente-servidor, será necessário que o sistema web de criação e distribuição de provas esteja hospedado em um servidor que esteja disponível aos clientes na mesma rede de dados dos dispositivos utilizados pelos alunos. A figura 1 permite imaginar um típico cenário de utilização da solução proposta. Pode-se observar que o computador do professor faz o papel de servidor e que está conectado ao mesmo roteador que disponibiliza o acesso aos clientes (dispositivos utilizados pelos alunos).

Além disso, são também necessárias N plataformas clientes (dispositivos móveis com sistema operacional Android,) nas quais o aplicativo de aplicação de prova deverá estar instalado. Neste caso, N é o número de alunos que irão realizar a avaliação. Cabe destacar que se faz necessário o uso de um dispositivo intermediário como ponto de acesso, o qual não precisa estar conectado à internet e será utilizado como ponto de ligação de acesso à rede sem fio, a qual pode inclusive ser específica daquele espaço escolar (ex. sala de aula).

#### 4. Protótipo

Essa seção descreve quais tecnologias foram usadas no desenvolvimento do protótipo, bem como, em detalhes, o processo de planejamento e implementação, tanto do sistema web de criação e manipulação de provas, quanto do aplicativo responsável pela execução da instância de uma prova.

##### 4.1. Modelo Entidade-Relacionamento

A modelagem de entidade e relacionamento serve para mostrar de forma conceitual a forma como os dados serão armazenados, bem como suas relações. A tarefa de modelar corretamente um sistema é necessária para que consiga atender às exigências dos requisitos. É importante lembrar que o diagrama a seguir representa como os dados são

armazenados, e não como estão explicitamente criados dentro da estrutura do framework Django, que é gratuito e de código aberto mantido pela organização DSF (Django Software Foundation), que mantém-se através de doações de empresas e indivíduos, visto que a mesma é independente e sem fins lucrativos [The Web Framework for perfectionist with deadlines.]. Tal representação é indiferente ao banco de dados relacional usado, no caso SQLite, pois existe uma camada de abstração na criação de modelos.

Como visto na figura 2, toda a estrutura depende de um usuário (User), cujos dados para essa implementação não são relevantes, pois são usados apenas como requisitos de segurança e para que uma mesma instância do sistema não possa ser usada por vários usuários diferentes. Um usuário pode realizar várias provas (Exam). Cada prova em si tem um nome (name) e uma data/hora a partir da qual poderá ser acessada. Além disso, existem dois campos de controle de status, ativo(active) e fechado (closed). Tais campos servem para representar ao sistema quais ações podem ser tomadas pelo usuário. Assim sendo, quando ativo e fechado são falsos, isso representa que a avaliação foi apenas criada. Quando apenas ativo ou fechado são verdadeiros, isso representa para o sistema que determinada prova está respectivamente ativa ou fechada. Quando uma prova está ativa significa que ela está sendo executada no momento, e quanto está fechada significa que já foi executada e as notas já foram computadas.

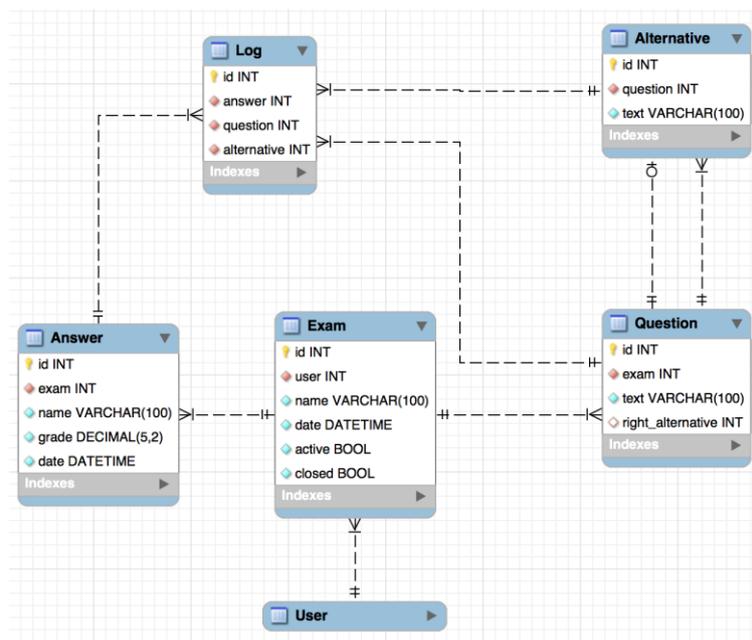


Figura 2. Diagrama de Modelagem ER que representa a estrutura criada pelos modelos

O modelo que representa uma questão (Question), tem como primeiro campo a prova (exam), uma chave estrangeira para a entidade de prova, o que representa que uma prova pode ter várias questões, ao qual pertence o texto da questão (text) e a alternativa correta (right\_alternative) para determinada questão, também um chave estrangeira, porém de relacionamento único como alternativa, de forma que não é necessário controlar a alternativa correta na entidade de mesmo nome, visto que a própria questão contém essa informação. Vale destacar que este último relacionamento

pode ser nulo, visto que, no processo de criação de uma prova, a questão é criada antes da alternativa.

A entidade de gabarito (Answer) pertence a uma prova (exam) e contém dados do usuário que realizou a mesma, bem como a nota(grade) calculada pelo sistema. Também se faz o registro da data/hora (date) do momento em que a informação foi recebida e inserida como um registro na base de dados. Além disso, para fins de programação, no framework Django uma relação de muitos para muitos, como é o caso entre gabarito e questão, deve ser escrita em um dos modelos ao qual pertence. Nesse caso, tal campo foi criado dentro do gabarito e explicitado que tal relacionamento deveria passar por outro modelo criado com o nome de Log, que por si mesmo é uma entidade com 3 chaves estrangeiras diferentes, sendo duas delas gabarito(answer) e questão(question), o que forma a relação de muitos para muitos. Porém, existe uma terceira, com o nome de alternativa(alternative), que armazena a alternativa respondida para tal questão. Dessa maneira, é guardado o gabarito por completo, caso o Usuário(Professor) deseje ver as mesmas.

## 4.2. Implementação

Baseado nos modelos requisitos descritos, optou-se por começar o processo de implementação pelo sistema web, visto que é este que apresenta todas as opções de manipulação de prova para o usuário. Nesse sistema é possível criar, excluir e alterar provas, questões e alternativas, bem como ativar, fechar e monitorar o andamento de provas. Ainda dentro do sistema web, desenvolvido usando a linguagem de programação Python juntamente com o framework web Django, foi necessária a implementação de uma Application Programming Interface (API), que seria responsável pela integração do sistema web com o aplicativo de aplicação dos testes/avaliações.

Por fim, foi desenvolvido o aplicativo usando como linguagem de programação principal o Javascript em conjunto com o framework Cordova, que torna possível que tais códigos sejam executados em múltiplas plataformas. O aplicativo desenvolvido tem como objetivo básico a captação da prova, previamente ativada no sistema, a fim de executá-la de forma que no final seja emitido para a API um objeto no qual consta o nome do aluno e o gabarito da prova respondida.

### 4.2.1. Web

Para a estrutura do projeto web foram criados dois módulos. O primeiro módulo tem o nome de administrator que, como o próprio nome sugere, contém todas as aplicações necessárias para que um usuário consiga criar, ativar e controlar uma prova. Dentro desse módulo existem dois aplicativos. O primeiro, com o nome de main(principal), controla todas as funções de autenticação, recuperação e troca de senha. Devido ao uso do framework Django, o desenvolvimento desse aplicativo é facilitado, visto que todas as views e models de autenticação são disponibilizadas nativamente pelo framework. É necessário apenas a criação das urls e templates para que tal sistema esteja funcionando. Além disso, o aplicativo também serve para ligar as urls de todos os outros aplicativos dentro do módulo, de forma a manter uma url saudável e consistente, seguindo o modelo “domínio.com.br/módulo/aplicativo”.

O segundo aplicativo tem nome de exams (provas). É neste aplicativo que são implementados os modelos previamente discutidos na seção 4.2. Para assegurar que um usuário X não consiga acessar a prova criada por um usuário Y, foi criado um mixin1 que implementa tal segurança, comparando o usuário que criou a prova que está sendo

acessada e o usuário que está logado. Caso estes sejam diferentes, o usuário é redirecionado para a tela de login inicial, onde há uma mensagem avisando que o usuário não tem permissão para acessar determinada url. Todas as views que controlam a aplicação são baseadas em classes. Sendo assim, é necessário apenas estender esse mixin em sua criação e o desenvolvimento pode prosseguir sem preocupação quanto à autorização e sem duplicidade de código.

A estrutura do sistema de criação em si é simples ao entrar no sistema, o usuário consegue visualizar uma lista de todas as provas feitas por ele. Nesta mesma tela, é também possível ver um link que redireciona o usuário para a criação de uma nova prova. Ao criar uma prova, o usuário é redirecionado para os detalhes da mesma, onde as seguintes ações podem ser feitas enquanto a prova não estiver ativa: remover e editar a mesma; criar, alterar ou remover questões; e ativar a prova. Porém, para que isso possa ser feito, é necessário que exista pelo menos uma questão válida.

Ao criar uma questão, o usuário é redirecionado para os detalhes da mesma, onde é possível editá-la, bem como criar ou editar alternativas. É importante lembrar que, para que uma questão seja considerada válida pelo sistema, a mesma precisa conter pelos menos uma alternativa correta. Ao ativar uma prova, o usuário não consegue mais editá-la, e é redirecionado para o relatório. A tela de relatório mostra ao usuário todas as provas feitas e suas notas, uma vez que as respostas do aluno forem recebidas. Não é necessário que o usuário atualize a tela, visto que o conteúdo é automaticamente atualizado a cada 10 segundos assincronamente. Na mesma tela, o usuário consegue fechar a prova, fazendo com que a mesma não possa mais ser enviada ou recebida pela api. Porém, o usuário ainda consegue ter acesso aos dados de tal prova. Vale lembrar que é a ativação da prova que mostra para o módulo de api que a mesma pode ser enviada, executada e recebida.

O segundo módulo do sistema web consiste de Application Programming Interface, que tem como objetivo integrar o aplicativo mobile e o sistema de criação de provas, e leva o nome de API. Esse módulo utiliza-se dos modelos previamente descritos e possui três views principais. A primeira consiste de um ping que avalia se a plataforma de integração está ou não disponível. A segunda, `SendExamView`, é a view que verifica se existe uma pesquisa ativa no banco de dados. Se existir, retorna uma mensagem mostrando que uma nova prova está disponível, juntamente com um objeto com os dados necessário para que a prova seja executada. É importante lembrar que as questões e alternativas são ordenadas aleatoriamente no momento da serialização das provas, fazendo com que as mesmas sejam diferentes umas das outras em estrutura, mas ainda com o mesmo conteúdo.

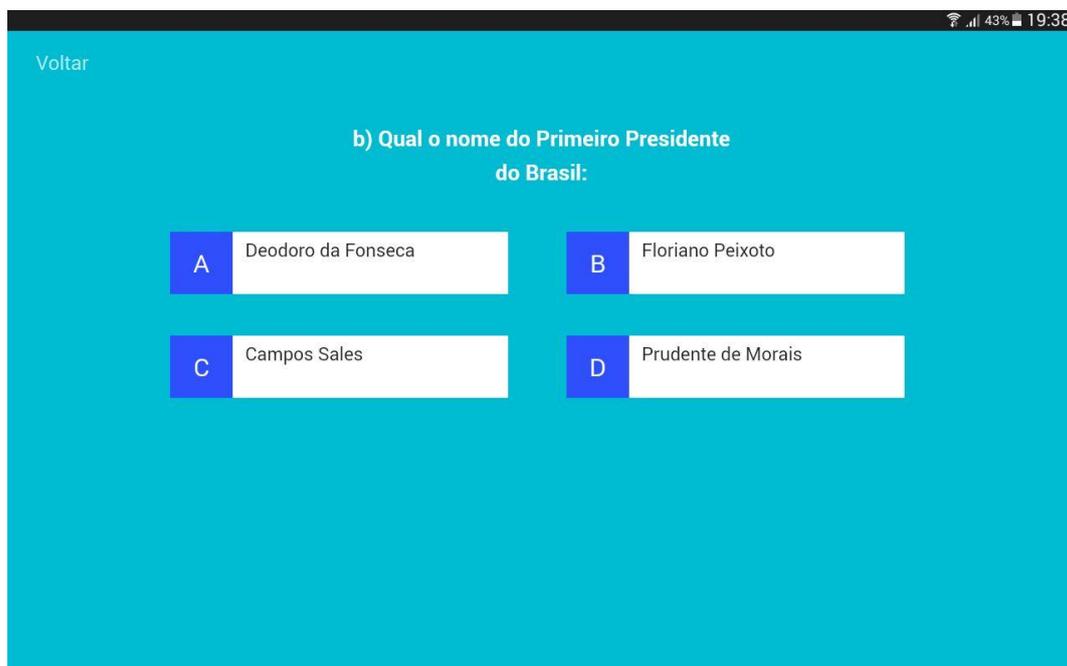
A terceira e última view, com o nome de `ReceiveVoteView`, é responsável por receber, desserializar e validar um objeto de respostas estruturado e, se válido, inseri-lo na base de dados juntamente com a média ponderada que é calculada no momento em que é inserido o registro. Essa view tem três possíveis respostas: “OK”, que mostra para o aplicativo que o gabarito foi registrado; “EXAM\_CLOSED”, que representa que a prova foi fechada e não pode mais ser registrado um gabarito; e “ERROR”, que significa que um erro desconhecido ocorreu e que o gabarito não foi registrado.

#### 4.2.2. Mobile

O aplicativo pode ser dividido em duas grandes partes. A primeira consiste do processo de inicialização do aplicativo até o momento em que uma prova é detectada para ser executada. Na rota inicial e principal de inicialização, a primeira ação pedida é que seja

digitado o endereço de IP do servidor que roda o sistema web. Com esse dado, o usuário é redirecionado para a rota de espera. Essa rota é controlada pela WaitView, que é responsável por, a cada 30 segundos, redirecionar o usuário para a rota de sincronização.

A rota de sincronização executa apenas uma view de contextualização do usuário, onde é apenas mostrado que o aplicativo está sincronizado. Nessa mesma rota, é iniciado um procedimento que requisita do servidor web se existe alguma prova para ser executada. Esta requisição é feita através de uma função wrapper, que sempre executa um ping para a API antes que a requisição seja enviada, de forma a diminuir a possibilidade de erros de comunicação. Caso não exista uma prova ativa, o usuário é redirecionado novamente para a rota de espera e esse ciclo é repetido até o momento em que um prova é recebida. Quando o objeto representativo da prova é salvo em memória, o usuário é redirecionado para a segunda parte do aplicativo.



**Figura 3. Uma questão renderizada pelo aplicativo**

A segunda parte do sistema consiste de uma rota e view principal e duas de apoio. A principal é a rota de execução de prova, onde é pedido que o usuário digite o seu nome para que a prova possa ser executada. O fluxo de execução é mantido através de um contador que inicia-se com zero e vai até o número de questões da prova. Quando uma alternativa da questão é selecionada pelo usuário na interface, como mostrada na figura 3, a resposta é registrada, o contador é incrementado e a função de renderização é chamada. Em todas as questões que não são a primeira é possível executar a ação de voltar, onde o contador é decrementado e a função de renderização é novamente executada.

Quando o contador for maior que o número de questões na prova, é mostrado para o usuário um gabarito de todas as questões que ele respondeu, bem como as alternativas selecionadas. O usuário pode então voltar e responder novamente alguma questão ou finalizar a prova. Ao finalizar, é enviado para a API o objeto que representa o gabarito respondido do usuário. É então mostrado ao usuário que suas respostas estão sendo sincronizadas e, quando é recebida a resposta do sistema web, é então mostrada uma mensagem de que seu gabarito foi registrado com sucesso. Quando o processo de

envio de um gabarito é iniciado, uma *flag* é lançada para avisar o sistema de que aquela instância do aplicativo não pode mais executar nenhuma prova.

## 5. Conclusão

No decorrer do trabalho foram abordados assuntos como o que a tecnologia da informação trouxe e ainda pode trazer para o meio da educação. Mais especificamente, como os dispositivos móveis, *tablets* e *smartphones*, podem ajudar o ambiente de sala de aula, sendo objetos de atração e engajamento ao invés de distração. Como alguns estudos abordados no trabalho mostram, existem ligações entre o uso de *tablets* no âmbito escolar e a melhora do desempenho do aluno em testes e, conseqüentemente, em suas notas no geral. Sendo assim, foi proposta uma solução que se utiliza das tecnologias web para facilitar a criação, execução e processo de classificação de exercícios ou provas executados dentro de sala de aula através de dispositivos móveis, sem o uso de papel.

Para o protótipo da solução, foram utilizadas tecnologias como a linguagem de programação Python, em conjunto com o *framework* web Django para o sistema web que controla e gerencia testes, bem como avalia os mesmos, baseado no gabarito criado, e o Javascript, juntamente com o *framework* de desenvolvimento de aplicativos híbridos Cordova, para a implementação do aplicativo que é responsável por captar, executar e enviar os resultados do teste para o sistema web.

## Referências

- Crockford, D. (2001) “JavaScript: The World's Most Misunderstood Programming Language”, <http://www.crockford.com/javascript/javascript.html>, Maio, 2016.
- “Django: The Web Framework for Perfectionists with Deadlines”, <https://www.djangoproject.com/>, Maio, 2016.
- Dalrymple, J. (2012), The Loop, “iPad improves Kindergartners literacy scores”, <http://www.loopinsight.com/2012/02/17/ipad-improves-kindergartners-literacy-scores/> Maio, 2016.
- Koile. K and Singer, D. (2006), “Development of a Tablet-PC-based System to Increase Instructor-Student Classroom Interactions and Student Learning”, WIPTE 2006 (Workshop on the Impact of Pen-based Technology on Education), Purdue University, April, 2006.