



ISSN: 1696-8352 - BRASIL – ENERO 2016

DESENVOLVIMENTO DE APLICAÇÕES MOBILE CROSS-PLATFORM UTILIZANDO PHONEGAP

Bezerra, Peterson Tubini.

Digital Specialist, Talk Agency. Australia

peterston@talkagency.com.au.

Schimiguel, Juliano.

Doutorado e Mestrado em Ciência da Computação pelo Instituto de Computação da UNICAMP.

jschimiguel@anchieta.br.

RESUMO

Num cenário de TI realista, as aplicações tendem à nascer e morrer rapidamente, o desenvolvimento de aplicações mobile cross-platform podem ser a melhor solução. O objetivo deste trabalho é apresentar a tecnologia PhoneGap desenvolvendo um protótipo de aplicação mobile para Android e Windows Phone, dois dos Sistemas Operacionais mais utilizados pelo mundo todo. Na qual demonstra o exemplo de um projeto real de curto prazo, com pouco recursos econômicos, pensando no desafio enfrentado por desenvolvedores web em diversas empresas.

Palavras-Chave: desenvolvimento ágil de software, processo de software, desenvolvimento mobile cross-platform.

ABSTRACT

On a realistic IT scenario where applications are born and die quickly, the development of mobile cross-plataform applications should be the best option. The aim of this project is to introduce the PhoneGap tool by developing a prototype of mobile application for Android and Windows Phone, two of the most used mobile operating systems in the world. This will demonstrate an example of a real world project of short deadline, with few economic

resources, with thought based on the challenge of web developers in worldwide companies.

Keywords: agile software development, software process, mobile cross-platform software development.

1.INTRODUÇÃO

Atualmente mais da metade dos usuários brasileiros utilizam a plataforma mobile, os 3 sistemas operacionais mais utilizados são: Android, ios e Windows Phone.

Num cenário de TI realista, as aplicações tendem à nascer e morrer rapidamente, e na hora do desenvolvimento de uma aplicação mobile multi-plataforma a escolha certa da melhor tecnologia pode fazer a diferença num projeto onde o tempo de desenvolvimento precisa ser curto, e de baixo custo, porém atingindo à todos. Este projeto, visa apresentar a tecnologia PhoneGap desenvolvendo um protótipo de aplicação mobile para Android e Windows Phone, dois dos sistemas operacionais mais utilizados pelo mundo todo. Estima-se que cerca de 80% dos usuários mobile estejam utilizando Android e os outros 20% se dividem em iOS, Windows Phone e outros.

O PhoneGap é um framework criado pela Adobe com o objetivo de que apenas um código-fonte, criar aplicações mobile multi-plataforma nativas utilizando HTML, CSS, JavaScript. Sua grande vantagem na utilização para desenvolvimento é a velocidade e facilidade, para que os desenvolvedores não precisem estar limitados ao desenvolvimento com linguagens específicas para cada tipo de plataforma.

Atualmente, há 5.590 desenvolvedores registrados utilizando o PhoneGap como framework para as aplicações mobile, com cerca de 545 apenas no Brasil. Além do mais, a Apple e a Microsoft recentemente citaram que o futuro é HTML5. Com estes novos campos no mundo de frameworks e aplicações mobile, soluções como esta, devem não só aumentar o mercado de mobile, como gerar uma nova revolução trazendo as aplicações web de volta à prioridade do mundo atual. Além de utilização de HTML5, CSS e JavaScript, possibilita a criação de aplicações mais robustas como a utilização de backservices REST em outra linguagem como PHP ou ASP.NET para aplicações que realizem CRUD ou então criar soluções corporativas completas

utilizando um pequeno time, ou até mesmo um único desenvolvedor para trabalhar em projeto de desenvolvimento mobile.

O protótipo deste projeto visa a construção de uma aplicação para Android e Windows Phone utilizando a ferramenta PhoneGap, utilizando suas APIs, e com a criação de um webservice REST em PHP, na qual demonstrará o exemplo de um projeto real de curto prazo, com pouco recursos econômicos, pensando no desafio enfrentado por desenvolvedores web em diversas empresas. Como cenário o protótipo demonstrará uma aplicação simples que permitirá realizar o cadastro de usuários, permissões destes usuários, realização de login, e a demonstração da utilização de recursos do dispositivo como câmera e gravação de áudio.

2.APLICAÇÕES MÓVEIS HÍBRIDAS X APLICAÇÕES MÓVEIS NATIVAS

A discussão entre aplicações móveis híbridas e aplicações móveis nativas começou em meados de 2010, e cresce nos dias de hoje entre os desenvolvedores. Há vários tópicos contidos nesta discussão, e neste campo de pesquisas, como: avaliação de performance, capacidades, interfaces, custo de desenvolvimento,

(CHARLAND; LEROUX, 2011), define que a competição entre código nativo e código web, em resumo é, uma guerra entre Javascript contra as linguagens compiladas. E também uma guerra de motores Javascript, pois cada browser utiliza um motor diferente para a execução de Javascript, como por exemplo: Chakra (Windows Phone), Google V8 (Android), Javascriptcore (iOS), que neste caso acaba sendo uma vantagem para as aplicações móveis híbridas, já que quanto melhor o desempenho dos browsers e de seus motores Javascript, maior será o desempenho de uma aplicação móvel híbrida.

O desenvolvimento de aplicações móveis nativas varia de acordo com a plataforma escolhida, que possui sua própria ferramenta de desenvolvimento, e uma linguagem escolhida, que na maioria dos casos como será demonstrado neste

trabalho, são linguagens e ferramentas diferentes para cada plataforma. Uma breve comparação entre aplicações móveis híbridas e aplicações móveis nativas, ocorre na compilação do projeto: Ao invés da compilação ocorrer uma vez por plataforma, no desenvolvimento híbrido, a compilação ocorre uma única vez, para todas as plataformas.

O custo de uma aplicação móvel híbrida segundo (CHARLAND; LEROUX, 2011), pode ser muito alta para aplicações de pequeno à médio porte. Por exemplo, uma startup na qual deseja atingir diversos públicos, não haveria como manter diversos programadores, em diversas linguagens diferentes para apenas poder atingir os sistemas operacionais mais populares, como mostra a tabela abaixo:

Figura 1 – Habilidades requeridas para 9 sistemas operacionais móveis

Required skill sets for nine mobile OSs.	
Mobile OS Type	Skill Set Required
Apple iOS	C, Objective C
Google Android	Java (Harmony flavored, Dalvik VM)
RIM BlackBerry	Java (J2ME flavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows Mobile	.NET
Window 7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
MeeGo	C, C++, HTML/CSS/JS
Samsung bada	C++

Fonte: CHARLAND; LEROUX, 2011, p.3

Uma das vantagens principais da utilização do Phonegap para desenvolvimento, é baixo custo de recursos: O desenvolvedor pode utilizar como IDE um editor de texto,

e realizar testes pelo próprio browser, exigindo assim, o mínimo de recursos para o desenvolvedor.

Para o desenvolvimento de aplicativos para iOS, é necessário uma licença de desenvolvedor Apple, que custa em torno de 100 dólares anuais, e possuir um dispositivo com sistema operacional Mac OSX para a compilação do projeto. Utiliza-se o Xcode (IDE de desenvolvimento da Apple), ou seja, indiretamente o desenvolvedor necessita de um MacBook ou até mesmo um computador Mac para o desenvolvimento da aplicação. Porém ao utilizar a ferramenta Phonegap, o desenvolvedor pode optar por apenas possuir a licença de desenvolvedor Apple, e realizar a compilação em cloud, através de uma máquina virtual com o sistema operacional Mac OSX instalado, resultando em alta economia.

(CHARLAND; LEROUX, 2011) considera como os pilares da performance para uma aplicação móvel: Latência e tempo de execução. Neste aspecto, sem dúvidas, uma aplicação nativa é consideravelmente mais rápida do que uma linguagem interpretada como Javascript, sem contar que o tempo de inicialização de código é muito maior, na linguagem interpretada. Porém a performance deve ser levada em conta com o contexto da aplicação: para uma aplicação de simples CRUD, e utilização de simples API's do dispositivo como câmera, rede, acelerometro, áudio o tempo de resposta é aceitável. Se o contexto for aplicações robustas, como por exemplo utilização de gráficos 3D em tempo real, como jogos, a melhor solução com certeza é o desenvolvimento nativo.

Construir uma aplicação móvel utilizando Javascript, HTML, CSS para diversos dispositivos ao invés de criar uma aplicação nativa para cada dispositivo, significa utilizar menos código, conforme (CHARLAND; LEROUX, 2011), a utilização de menos código para desenvolvimento, significa menos manutenções e manutenções mais rápidas.

É possível reduzir ainda mais o código-fonte ao utilizar frameworks, utilizando bibliotecas com diversas funções prontas, facilitando a criação do aplicativo e ainda poupa que o desenvolvedor precise ter conhecimentos avançados em Javascript.

3.CONCEITO DE FRAMEWORK E SUAS CARACTERÍSTICAS

Um framework é uma biblioteca de recursos, que podem compartilhar diversos tipos de funções como por exemplo: Busca de strings, funções de matemática, funções para tratamento de imagens, funções para a manipulação de arquivos, etc. Dentro da estrutura do framework, que costuma ser uma estrutura de pastas na maioria das linguagens, incluem documentação, código-fonte, e até múltiplas versões, para retrocompatibilidade. Frameworks não são obrigados à incluir o seu código-fonte, também podem distribuir a versão final dos arquivos para serem utilizados, porém, não costuma ser comum.

Em um pacote de framework, pode ter incluído, outros pacotes de framework também. As vantagens da utilização de um framework são inúmeras, entre elas:

- **Segurança:** Existem diversos frameworks populares, que possuem vasta comunidade, e são revisados diariamente, com diversos bugs sendo reportados, analisados, e corrigidos.
- **Desenvolvimento ágil:** Com a utilização de um framework, um projeto pode ser desenvolvido com uma equipe menor, e muito mais rapidamente, pois diversas funções complexas estão disponíveis para uso.
- **Suporte da Comunidade:** No caso de utilização de frameworks de grande porte, a maioria possui comunidades ativas que estão dispostas à responder dúvidas, corrigir bugs e até sugerir as melhores práticas.

- Oportunidades de Trabalho: Ao utilizar um framework popular no mercado e ganhar experiência em sua utilização, você estará se preparando para um fecho maior no mercado.

Porém, a utilização de um framework, também tem suas desvantagens, tais como:

- Performance: Nem sempre a utilização de um framework significa melhor performance, em muitos casos a utilização de um framework, pode ter um impacto significativo em uma aplicação.
- O tamanho da aplicação: O tamanho da aplicação deve ser um fator prioritário na hora da utilização de um framework. Se a aplicação é pequena, e não há alguma necessidade da utilização de um framework, em muitos casos, é melhor não utilizar.

4. PHONEGAP – FRAMEWORK OPEN SOURCE

PhoneGap é um framework open source e gratuito, baseado na licença Apache, atualmente está sendo utilizado por cerca de 400,000 desenvolvedores e já foi baixado cerca de 1 milhão de vezes. Pode construir aplicações simples, ou robustas utilizando HTML 5, CSS e Javascript, conectando-se ao backend de webservice criado em outras linguagens como: PHP, Java, C#, entre outras. O framework foi criado em 2011 de Outubro pela empresa Nitobi, que foi posteriormente adquirida pela Adobe.

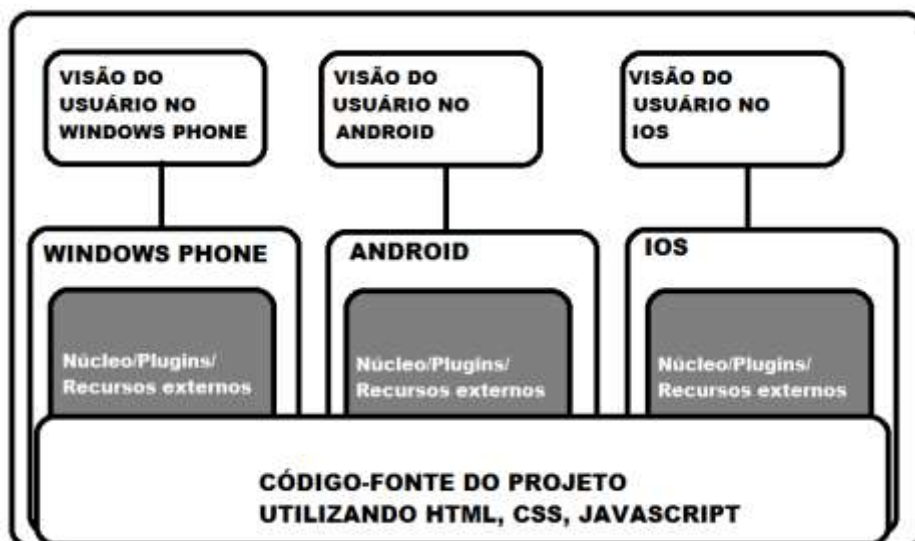
"Eu acho que a maior razão na qual o PhoneGap foi criado é porque nós queríamos voltar à criar aplicações web usando nativamente o HTML,CSS e Javascript, e não queríamos ficar presos em uma plataforma mobile." (BRIAN LEROUX, SR. COMPUTER SCIENTIST, PHONEGAP, 2012). Ou seja, a razão da criação do PhoneGap, é fazer com que desenvolvedores web, se tornem desenvolvedores mobile, sem ter que aprender uma nova linguagem, ou se prender à uma única plataforma.

Ao se tornar open-source, A Adobe como parte de aquisição, contribuiu para o Apache e tornou-o Open Source. A Adobe continuou com o nome PhoneGap, e a Apache nomeou-o como Cordova, onde os dois projetos são basicamente iguais, contendo mínimas alterações entre si.

"É claro, o ambiente de smartphones é de rápida mudança e rápida evolução de ambiente. Como tal, soluções precisam de um igual agilizamento." (KERRI SHOOTS; PHONEGAP FOR ENTERPRISE, 2014).

O diagrama abaixo nos permite ter uma noção básica, sobre o núcleo do PhoneGap:

Figura 2 – Estrutura básica do PhoneGap.



Fonte: SHOTTS, KERRI. PHONEGAP FOR ENTERPRISE p.12. Adaptado pelo autor.

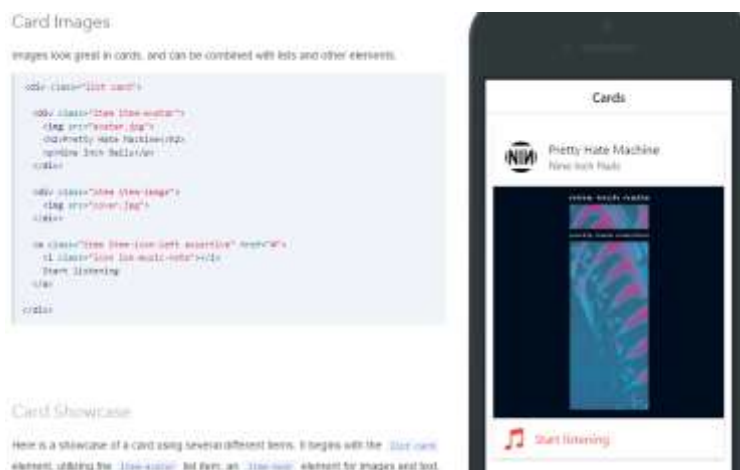
5. IONIC – FRAMEWORK FRONT END

O framework Ionic, é gratuito e open-source, com uma biblioteca CSS e Javascript completa, e possui uma detalhada documentação, além de uma comunidade enorme. Possui diversas ferramentas que auxiliam o desenvolvedor em sua criação, como por exemplo o Ionic Live, que permite ao desenvolvedor testar a aplicação em tempo real através do próprio navegador.

Possui integração nativa com o AngularJS e o NGCordova, facilitando ainda mais o seu uso, além de exemplos de seus métodos e até projetos exemplos.

Alguns exemplos de componentes abaixo:

Figura 3 – Documentação do Ionic.



Fonte: Ionic – Documentação do Ionic.

Figura 4 – Exemplo da utilização de classes do PhoneGap.

on-hold

[View Source](#)

[Improve this doc](#)

Touch stays at the same location for 500ms. Similar to long touch events available for AngularJS and jQuery.

Usage

```
<button on-hold="onHold()" class="button">Test</button>
```

Fonte: Ionic – Documentação do Ionic.

Nos dois exemplos demonstrados, podemos analisar como facilmente podemos criar templates utilizando apenas HTML e CSS, sem conhecimento algum de Javascript. No segundo exemplo, podemos verificar como facilmente podemos utilizar a função `onHold()` em um botão, que na qual, ao usuário pressionar este botão por 500 milissegundos, uma ação desejada ocorrerá.

6. METODOLOGIA DO TRABALHO

6.1 LARAVEL – FRAMEWORK PHP

Laravel é um framework PHP voltado para aplicações web, é open-source e está sobre a licença do MIT. Foi criado em Junho de 2011 por Taylor Otwell, desenvolvedor .NET e PHP, com um simples ideal: criar um framework onde programar fosse divertido. Em apenas 4 anos,, Laravel é apontado em diversas fontes e está entre o primeiro ou segundo lugar, dos melhores frameworks de PHP.

Certamente, o que faz o framework Laravel ser um dos mais utilizados no mundo todo hoje, é a facilidade de aprendizado, seus recursos, a comunidade principalmente, e uma documentação clara, e por final, sua estabilidade. Sua modularidade é tão robusta, que permite que o desenvolvedor utilizar diversos Design Patterns, e criar sua aplicação utilizando o design pattern desejado, por exemplo: O desenvolvedor pode utilizar o design pattern Facade, ou o design pattern MVC, dentre outros.

Requerimentos para uma instalação do Laravel:

- PHP maior ou igual à 5.5.9.
- Extensão OpenSSL.
- Extensão PHP PDO.
- Extensão MBString.
- Extensão Tokenizer PHP.

Com apenas estes requerimentos já é possível utilizar o Laravel, e desfrutar de diversas funcionalidades do Laravel, segue algumas inclusas:

- Módulo de Routing (Utilizado para criar rotas HTTP).
- Módulo Middleware (Utilizado para filtro de requisições HTTP).
- Módulo Responses (Utilizado para respostas HTTP, como por exemplo: JSON,File)
- Módulo Cache (Utilizado para o cache entre sua aplicação e o banco de dados)

- Módulo Eloquent (ORM ou Object Relational Mapping, módulo que faz o relacionamento do banco de dados em forma de objetos).
- Módulo Mail (Responsável por todas as funções como por exemplo, enviar emails)

Entre outros diversos módulos disponíveis logo após a instalação.

6.2 MYSQL: BANCO DE DADOS

O terceiro banco de dados mais utilizado no mundo todo, open-source e criado utilizando C/C++, e adquirido pela Oracle, o MYSQL é um dos bancos de dados preferidos de todos os programadores, pelo seu fácil uso e pelo seu potencial.

A razão para a utilização neste protótipo, é simples, é um banco de dados seguro, rápido, e ótimo para aplicações de pequeno à medio porte, além de ser integrado facilmente com PHP.

6.3 ANGULARJS: JAVASCRIPT FRAMEWORK

O AngularJS é com certeza, um framework bem recente no mercado, criado pela Google, muito utilizado para aplicações MVC, permite a extensão da linguagem HTML para a criação de aplicações 100% dinâmicas.

De certa forma, o AngularJS permite a utilização do MVC no lado do client, utilizando AJAX para as requisições, e organizando suas aplicações MVC não apenas no lado do servidor, mas também no lado do client.

O motivo da utilização desta tecnologia no protótipo, é otimizar e facilitar, o uso do Javascript no protótipo, além de fácil manipulação de dados.

Ao utilizar o AngularJS em conjunto ao Laravel, isso nos permite que tenhamos dois frameworks agindo em conjunto e sincronizados, um no lado do client, e outro no lado do servidor. Outras razões pela qual decidi optar pelo AngularJS são suas principais funcionalidades: Diretivas, Binding de dados, filtros, módulos e rotas, onde, um

desenvolvedor com experiência básica no Laravel ou um framework como o ASP.NET MVC, já é familiarizado com estes conceitos e práticas.

7. PROTÓTIPO

Após uma breve introdução sobre todas as ferramentas utilizadas para o desenvolvimento deste protótipo, posso por fim, detalhar todos aspectos deste projeto.

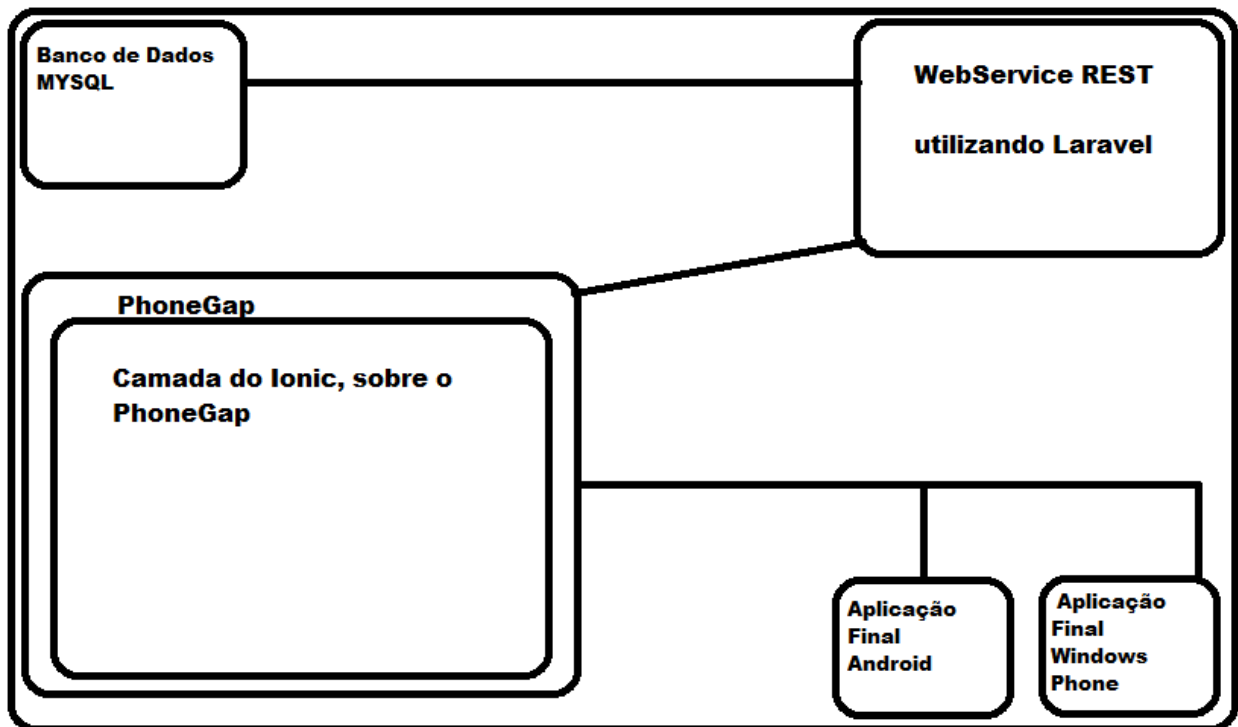
A ideia base deste protótipo é o desenvolvimento de uma aplicação empresarial, na qual requer um banco de dados para a persistência destes dados, suas regras de negócio, que manipulará os dados e por fim a aplicação, envolvendo todos os aspectos.

O protótipo apresenta uma aplicação simples, onde o objetivo é, além da utilização de todas as tecnologias presentes, utilizar as funções básicas do hardware do aparelho como: áudio e câmera. Realizando um CRUD de cadastro de usuários com o uso da câmera para a foto de profile, e áudio para gravar uma mensagem de voz ,utilizando o Laravel como backend, o Ionic como FrontEnd, e o PhoneGap como a arquitetura principal.

Este protótipo visa utilizar todas as ferramentas apresentadas neste documento até o seguinte momento, de forma à demonstrar a utilização do PhoneGap + Laravel + Ionic para o desenvolvimento rápido de aplicações mobile, voltadas para o uso empresarial, onde normalmente o custo estimado de projeto é baixo, onde o tempo de desenvolvimento é baixo, e a aplicação deve ser totalmente funcional, e atender os requisitos da empresa.

Na imagem abaixo, demonstro um breve resumo da arquitetura do protótipo, e como as tecnologias estão sendo utilizadas em conjunto:

Figura 5 – Estrutura do protótipo.



Fonte: elaborado pelo autor.

Analisando a imagem, podemos verificar que o PhoneGap, ao final do protótipo, nos providenciará as versões finais do aplicativo, na versão nativa de cada plataforma, no caso Android e Windows Phone.

7.1 PROTÓTIPO: SEGURANÇA

"Independente da arquitetura, é importante notar que o servidor de banco de dados, não tem nenhuma exposição direta com o lado de fora da aplicação." (KERRI SHOTTS, PHONEGAP FOR ENTERPRISE, 2014).

Ao trabalhar com uma aplicação que utiliza web service como backend, uma grande preocupação do desenvolvedor, é a segurança. Como desenvolvedor web, e com uma pequena experiência utilizando e criando web services, há dois níveis de autenticação no mercado atual, uma chamada: OAuth, e a outra: Tokens.

O acesso à web services acontece com requisições HTTP, por exemplo: GET, POST, etc e o retorno será em JSON ou XML, portanto, mesmo que possua um login e senha, ainda não é suficiente para a proteção das rotas HTTP, pois o cabeçalho apenas solicitará as informações, e não contém nenhum tipo de lógica envolvida, portanto é necessário a utilização de tokens ou OAuth para uma melhor proteção.

Visando o foco principal deste artigo é o protótipo, irei detalhar de maneira básica o funcionamento deste tópico.

7.11 OAuth:

É um framework de autorização que permite o acesso de uma aplicação terceira à prover acesso limitado à um serviço HTTP limitada por domínios (2-legged), e limitada por usuário e domínio (3-legged).

7.12 Tokens:

Tokens podem ser entendidos como “chaves de acesso”, que são compartilhadas entre o cliente-servidor.

Há diversas maneiras de se implementar um token dentro de uma aplicação, entre elas:

- Servidor: Um token é armazenado no servidor, e o cliente apenas realiza a autenticação com este token.
- Cliente / Servidor: Neste modo, o usuário se autentica com login e senha e há duas possibilidades para o token: Ou o token pode ser armazenado numa tabela do banco de dados, e através de scripts ela ser regenerada à cada tempo, ou através de programação no servidor, o token se regenera de tempos em tempos. À cada acesso, o token é armazenado temporariamente no cliente, geralmente por SESSION, e quando ocorre à regeneração, é solicitado novamente ao servidor.

8 PROTÓTIPO: CONFIGURAÇÃO NECESSÁRIA

Apresento por meio deste capítulo, a linguagem, IDE e ferramentas utilizadas para o desenvolvimento deste projeto, que serão detalhados nos itens seguintes.

Como linguagem base para a criação deste projeto, no backend foi utilizado PHP em conjunto com framework como descrito neste documento, para criação do servidor, que no caso é o webservice REST onde possui toda sua lógica, validações e persistências com o banco de dados.

Para a criação do front end, foi utilizado Javascript, HTML 5, CSS, em conjunto com os frameworks descritos neste documento.

Em recursos de hardware, para o desenvolvimento do projeto, testes e hospedagem do servidor, foi utilizado meu próprio laptop, Dell XPS L502X, com processador i3 2330M 2.20 GHz, com 4 GB memória RAM, e SSD de 256 GB. O grande diferencial das linguagens escolhidas, é o consumo baixo de hardware, portanto, ainda com hardware inferior seria possível o desenvolvimento deste projeto sem qualquer impacto.

8.1 PROTÓTIPO: IDE

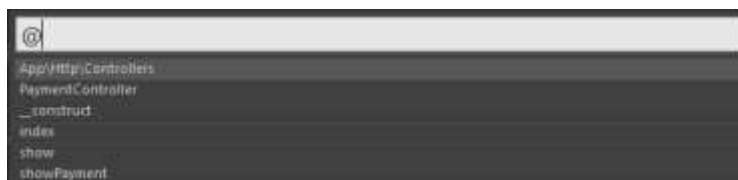
A escolha de uma boa interface de desenvolvimento para o projeto auxilia na produtividade, e a IDE que eu decidi utilizar para este projeto, foi o Sublime Text 3. O motivo de minha escolha, foi pelo fato de ser uma IDE que consome pouquíssimos recursos, leve, multi-plataforma, com vantagens de instalação de plugins, multi-cursors, instalação de temas, entre outros.

Um recurso muito interessante é o Multi-Edit, que permite que você substitua tags e palavras rapidamente, e o Emmet, que permite a construção de blocos de códigos

utilizando templates que o próprio desenvolvedor pode criar, e permite aumentar sua produtividade.

Outro recurso interessante, é o Goto Symbols, que permite que você encontre métodos rapidamente:

Figura 6 – Busca de métodos no Sublime Text.



Fonte: print screen da aplicação no sistema operacional Windows 10.

8.2 PROTÓTIPO: PHP

A escolha da linguagem PHP para este projeto, foi a minha paixão e inspiração por esta linguagem, criada em C/C++, uma linguagem open-source e de alta performance, de código livre e com uma comunidade online que cresce rapidamente, não é à toa que a maior rede social do mundo hoje, o Facebook, escolheu ela em seus primórdios. O que faz esta linguagem ser admirada por diversos programadores, é sua facilidade de aprendizado e seu dinamismo.

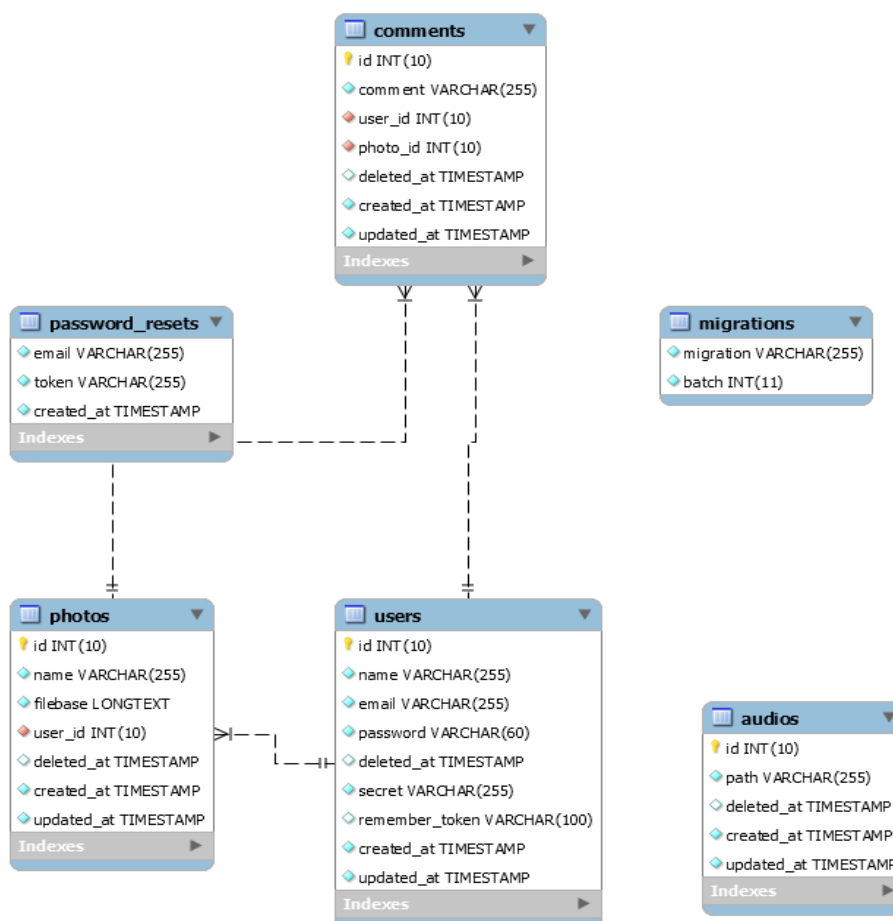
Outro ponto positivo para esta linguagem, é sua documentação bem detalhada e com diversos exemplos, onde os desenvolvedores contribuem diretamente com a documentação, o que facilita ainda mais o aprendizado do desenvolvedor, aumentando rapidamente sua linha de aprendizagem.

8.3 PROTÓTIPO: MODELAGEM DER

Para a modelagem do MER, foi utilizado o MYSQL WorkBench, ferramenta gratuita da Oracle, uma suíte completa, que além de criar modelagens permite ao desenvolvedor realizar consultas ao banco de dados, criar procedures, realizar backups, restore, realizar relatórios, exportação e importação de dados, e até realizar melhorias de performance no banco de dados.

Abaixo, o modelo entidade relacionamento, que fora designado totalmente de forma simples baseando-se apenas em um protótipo com o objetivo maior, de mostrar em si suas tecnologias e a forma em que foram aplicadas.

Figura 7 – Modelo Entidade Relacionamento do protótipo.



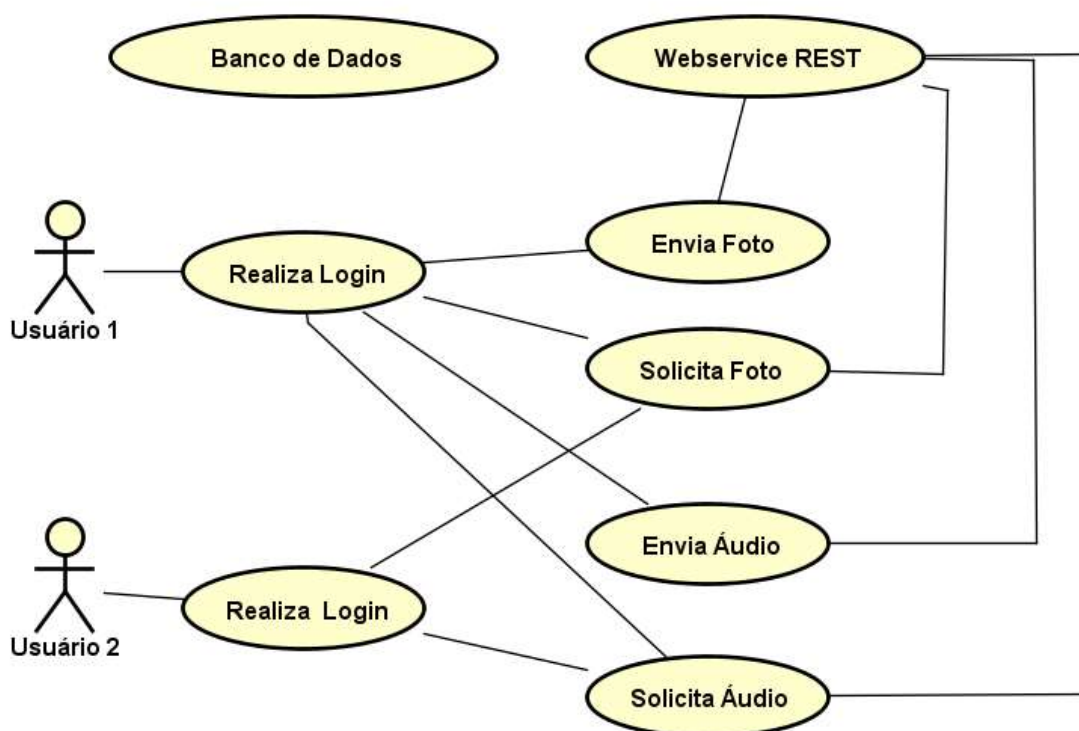
Fonte: elaborado pelo autor.

8.4 PROTÓTIPO: DIAGRAMA DE CASOS DE USO

Para a criação do diagrama de casos de uso, também foi utilizado o software Astah, além da criação de diagramas de classes, o software também permite a criação de diagrama de casos de uso. Novamente a decisão para a escolha deste software, foi por sua praticidade na utilização.

Neste diagrama de casos de uso, é demonstrado as possíveis utilização do protótipo, onde o usuário solicita ou envia fotos e áudios para o servidor através do webservice REST.

Figura 8 – Diagrama de casos de uso do protótipo.



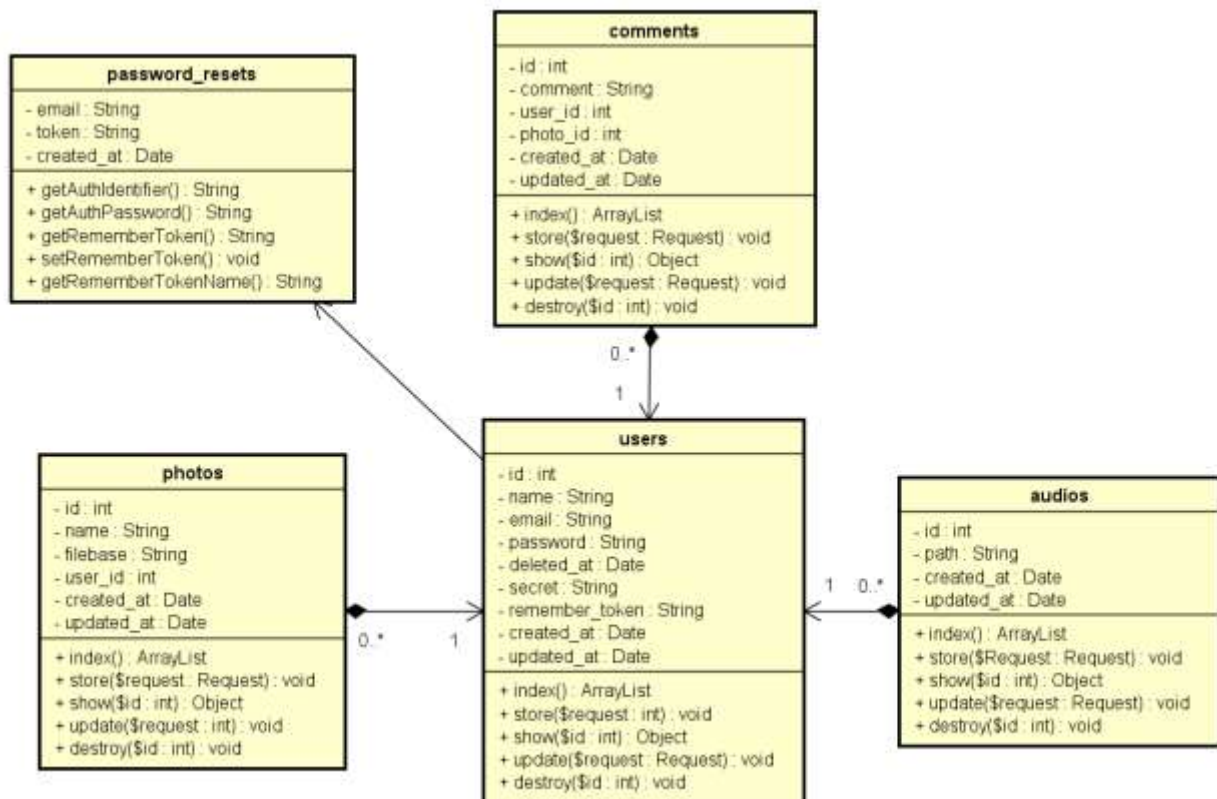
Fonte: print screen do diagrama utilizando o software Astah.

8.5 PROTÓTIPO: DIAGRAMA DE CLASSES

Para a criação do diagrama de classes, foi utilizado o software Astah, um software especializado para a criação de diagrama de classes, sua interface é simples, e possui usabilidade fácil, o que permite que todo o projeto seja desenhada de forma simples e rápida. Além de funcional, o software possui teclas de atalho que garante maior produtividade ao desenvolvedor.

Através deste diagrama de classes, podemos ver a simplicidade dos métodos, no caso estes métodos como: Index, Store, Show, Update e Destroy, são gerados automaticamente pelo Laravel, facilitando toda a operação do CRUD.

Figura 9 – Diagrama de classes do protótipo.



Fonte: print screen do diagrama utilizando o software Astah.

8.6 PROTÓTIPO: TELAS

Neste capítulo, demonstrarei algumas telas referentes ao protótipo em algumas das visões de login, cadastro de áudio, cadastro de fotos e o funcionamento em multi-plataforma, aplicando todas as tecnologias citadas nos capítulos anteriores neste documento.

8.6.1 PROTÓTIPO – TELAS: LOGIN

Tela de login do protótipo, pela qual o usuário realiza o login através dos campos email e senha.

Figura 10 – Tela de login no Windows Phone.

prototipo

.

Entrar

Fonte: print screen da aplicação no sistema operacional Windows Phone

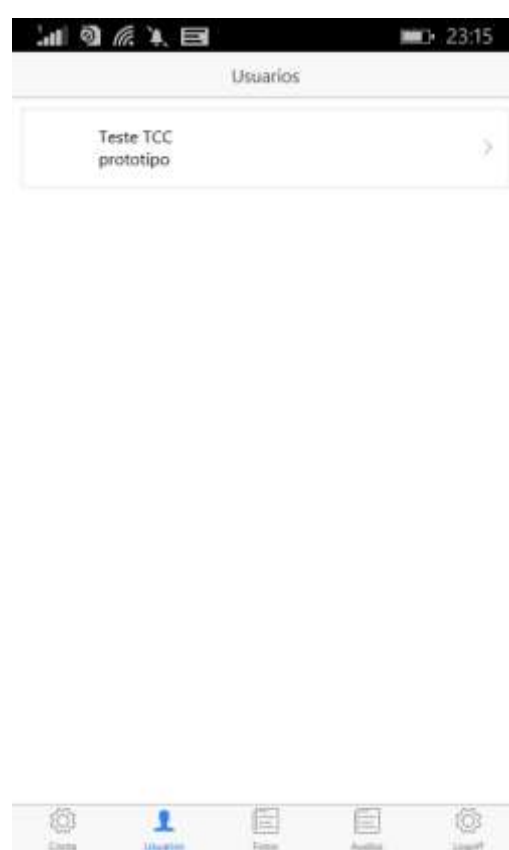
8.62 PROTÓTIPO – TELAS: CONTA E USUÁRIOS

Respectivas telas dos setores do aplicativo, tais como: usuários, conta. Onde é possível a visualização dos usuários cadastrados no sistema, e a alteração de senha do usuário atual.

Figura 11 – Tela Conta.

Figura 12 – Tela

Usuários.



Fonte: print screen da aplicação no sistema operacional

Windows Phone

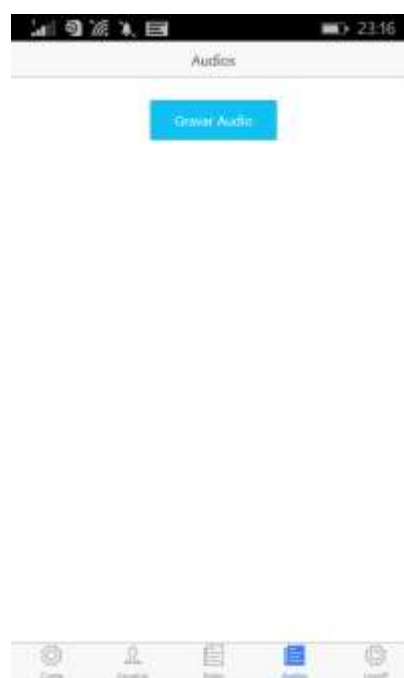
8.63 PROTÓTIPO – TELAS: FOTOS E ÁUDIOS

Nas figuras abaixo, as telas fotos e áudios, onde é possível que o usuário tire uma foto, ou grave um áudio, e será armazenado no servidor.

Figura 13 – Tela Fotos.



Figura 14 – Tela áudios.



Fonte: print screen da aplicação no sistema operacional Windows Phone.

8.64 PROTÓTIPO – TELAS: FOTOS E CÂMERA

Telas pré-envio e pós-envio da foto, com popup amigável para o usuário, também em seguida, a chamada da câmera do dispositivo.

Figura 15 – Pré-envio da foto.

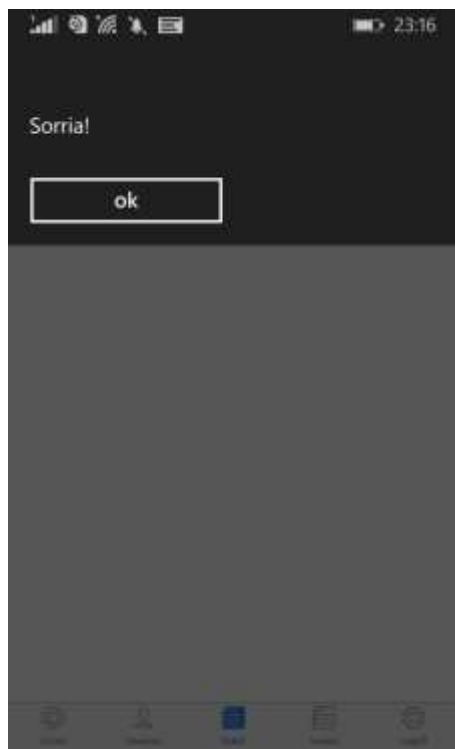
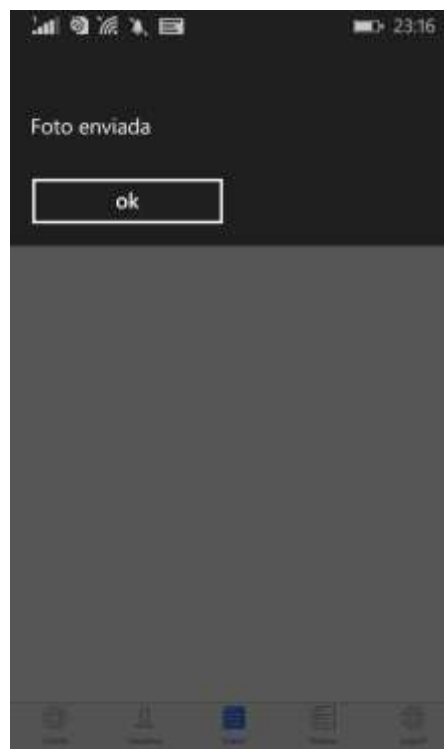
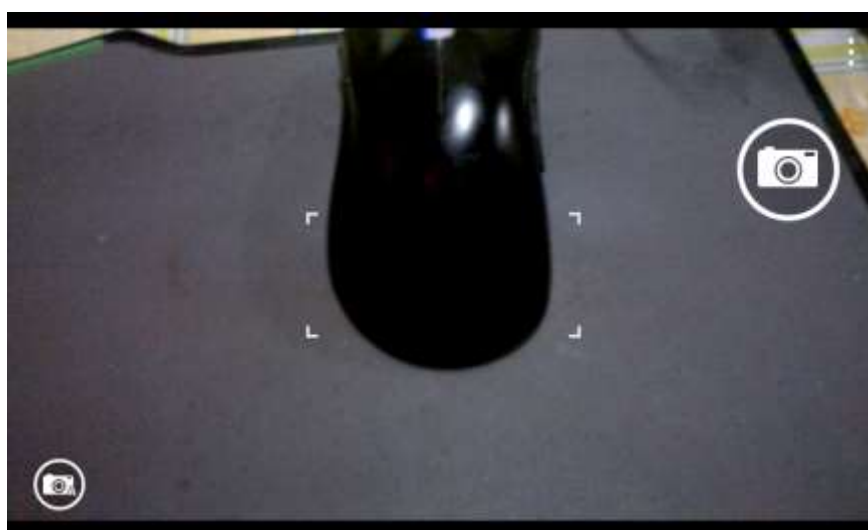


Figura 16 – Pós-envio da foto.



Fonte: print screen da aplicação no sistema operacional Windows Phone.

Figura 17 – Chamada da câmera do dispositivo.



Fonte: print screen da aplicação no sistema operacional Windows Phone.

8.65 PROTÓTIPO – TELAS: FOTOS PÓS-ENVIO

Após o envio da foto, o usuário final pode ver a foto que foi enviada para o servidor.

Figura 18 – Pós-envio da foto.

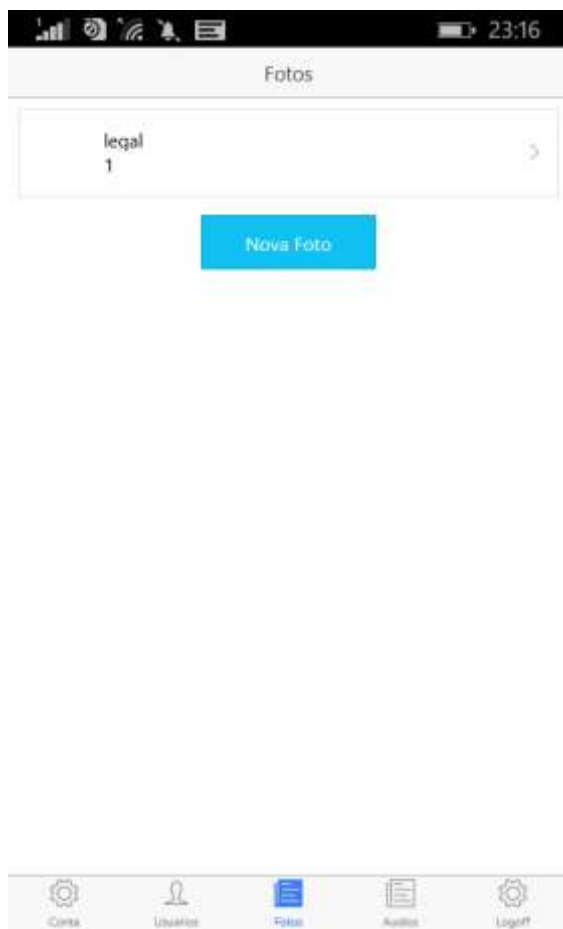


Figura 19 – Visualização da foto.

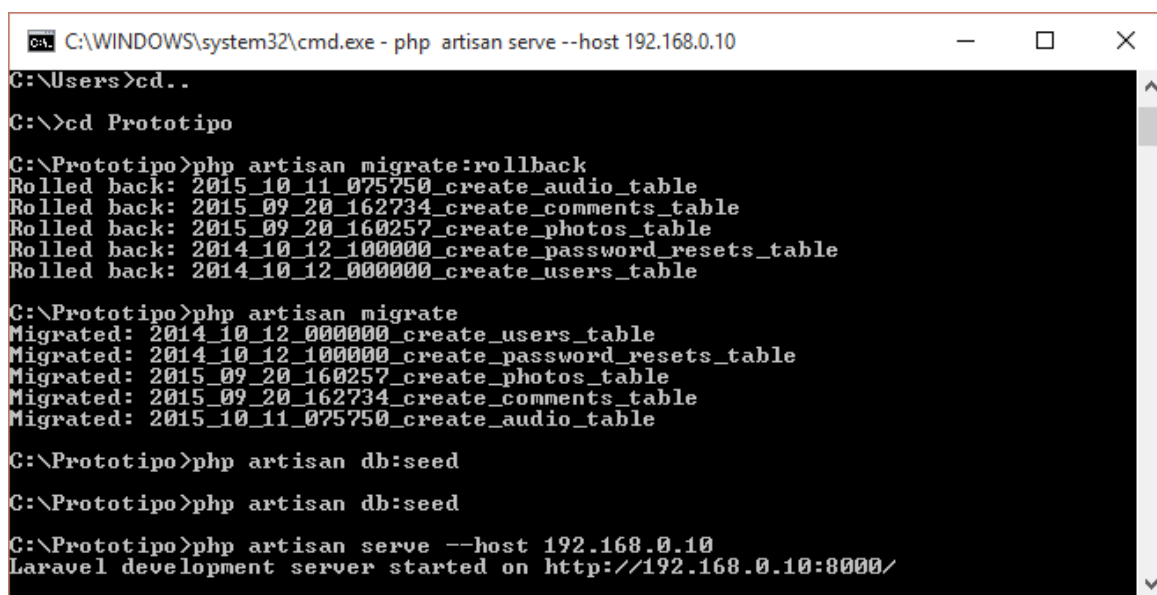


Fonte: print screen da aplicação no sistema operacional Windows Phone.

8.66 PROTÓTIPO – TELAS: SERVIDOR E BANCO DE DADOS

Nas telas abaixo, é possível verificar a inicialização do webservice, onde será responsável pelo armazenamento dos dados e arquivos gerados pela aplicação, e também o acesso ao banco de dados em MYSQL.

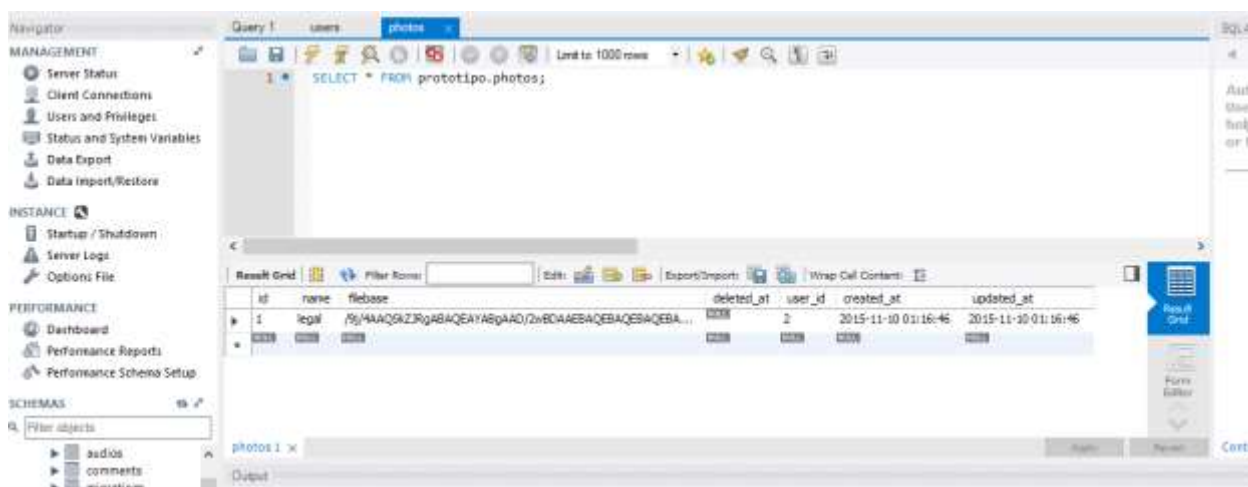
Figura 20 – Iniciando o servidor.



```
C:\WINDOWS\system32\cmd.exe - php artisan serve --host 192.168.0.10
C:\Users>cd..
C:\>cd Prototipo
C:\Prototipo>php artisan migrate:rollback
Rolled back: 2015_10_11_075750_create_audio_table
Rolled back: 2015_09_20_162734_create_comments_table
Rolled back: 2015_09_20_160257_create_photos_table
Rolled back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_000000_create_users_table
C:\Prototipo>php artisan migrate
Migrated: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrated: 2015_09_20_160257_create_photos_table
Migrated: 2015_09_20_162734_create_comments_table
Migrated: 2015_10_11_075750_create_audio_table
C:\Prototipo>php artisan db:seed
C:\Prototipo>php artisan db:seed
C:\Prototipo>php artisan serve --host 192.168.0.10
Laravel development server started on http://192.168.0.10:8000/
```

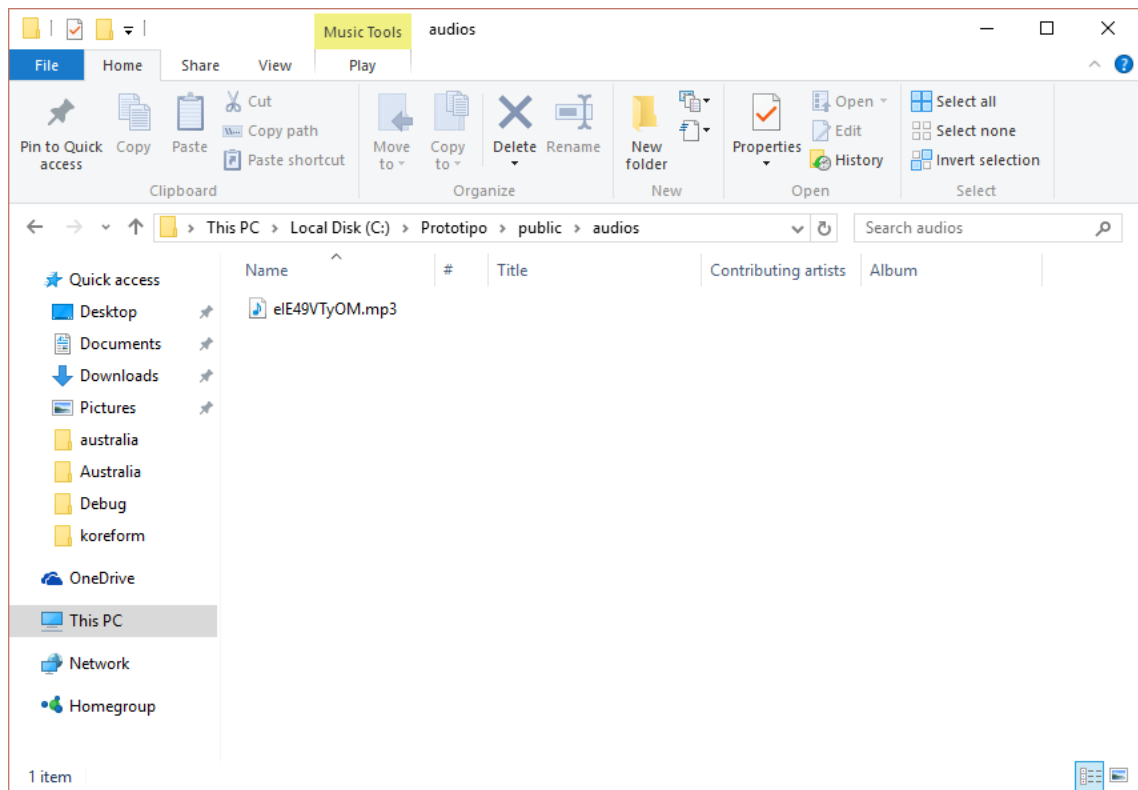
Fonte: print screen do webservice no sistema operacional Windows 10.

Figura 21 – Verificação do registro no banco de dados.



Fonte: print screen do MYSQL no sistema operacional Windows 10.

Figura 22 – Verificação da gravação de áudio.

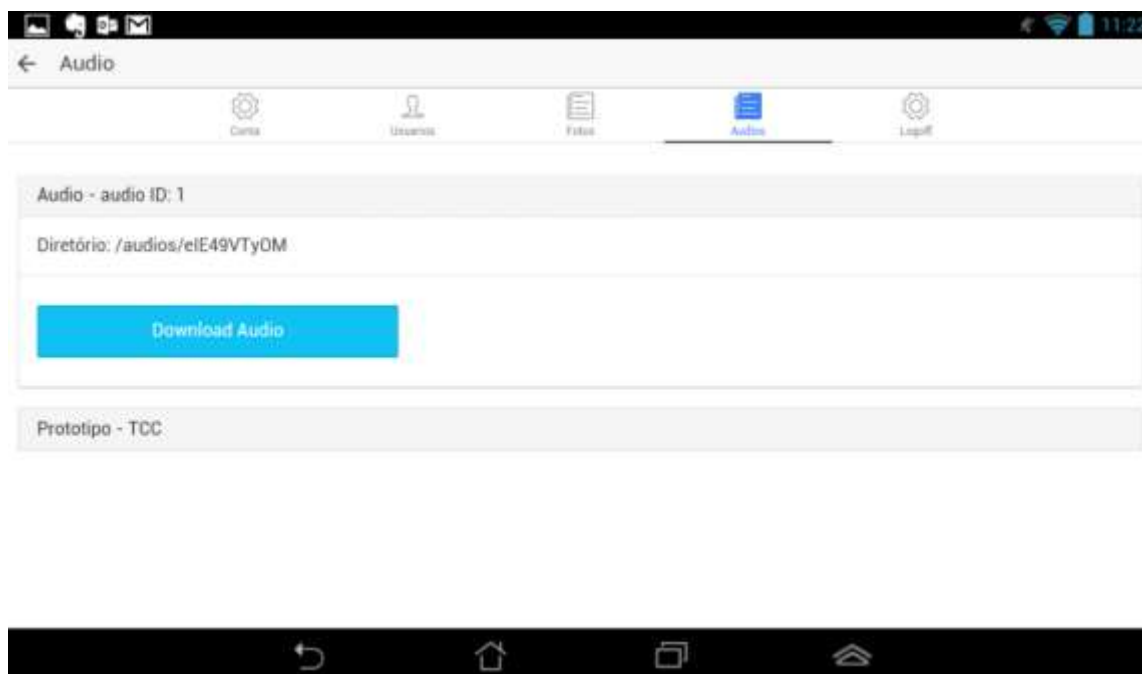


Fonte: print screen da pasta de áudios no sistema operacional Windows

10.

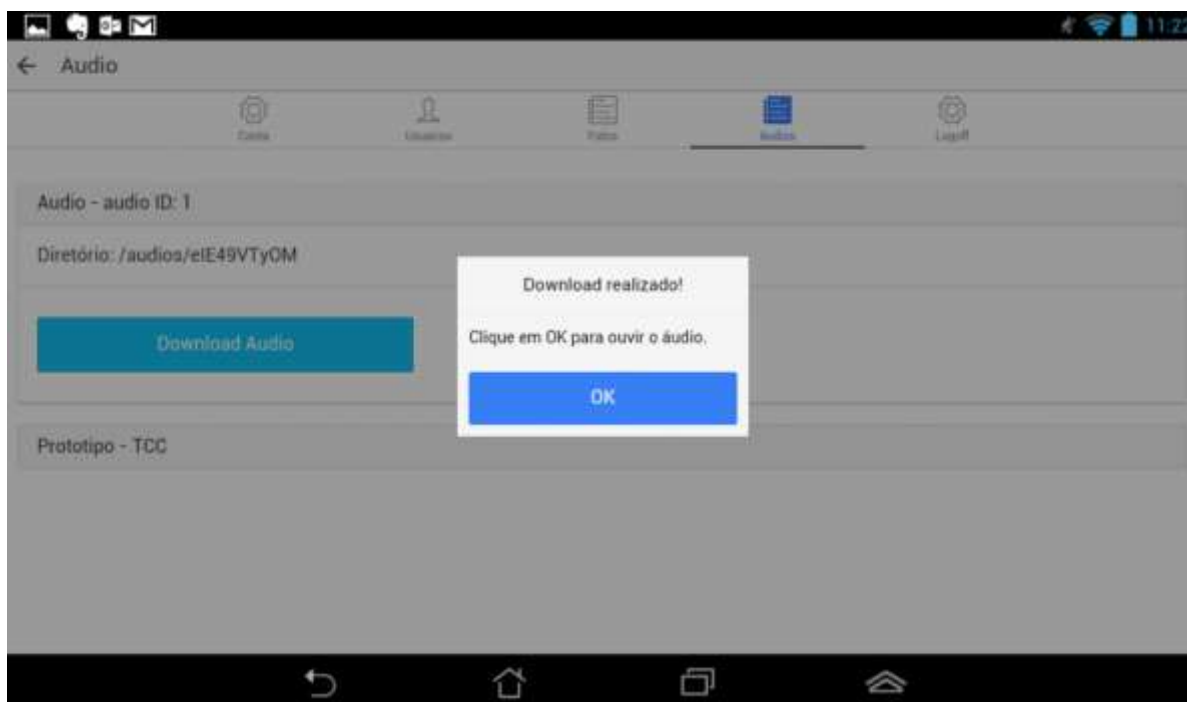
Telas do pós-envio do áudio, onde nesse caso é possível que o usuário ouça o áudio que foi enviado ao servidor.

Figura 23 – Pós-envio do áudio.



Fonte: print screen da aplicação no sistema operacional Android.

Figura 24 – Pós-download do áudio.



9. CONSIDERAÇÕES FINAIS

A utilização da ferramenta PhoneGap, em conjunto com todas as demais apresentadas neste documento, foram extremamente satisfatórias, em minha visão como desenvolvedor web, com conhecimentos básicos em desenvolvimento mobile atendeu todas as expectativas, que no caso era de desenvolver uma aplicação mobile com todos os conhecimentos obtidos nestes anos de experiência em minha carreira de desenvolvimento web.

Contudo, cabe à avaliação do desenvolvedor rever os requisitos que é esperada em sua aplicação. Para aplicações de pequeno e médio porte e de contexto empresarial, a ferramenta PhoneGap cumpriu todos os aspectos na qual foi exigido neste protótipo, porém, para um projeto de grande porte, cabe uma avaliação precisa e que deve ser melhor avaliada.

Vale também realçar, que há conhecimentos necessários e que devem ser utilizados em conjunto com o PhoneGap para que se tenha êxito na criação de uma aplicação, como demonstrado neste documento, como por exemplo a criação de um webservice REST, conhecimentos da ferramenta AngularJS, que permitiu que todas as telas fossem dinâmicas, e também ao menos o conhecimento intermediário de Javascript, dentre as outras demais citadas.

Todo o estudo realizado neste protótipo, serve de base para futuros estudos e melhores análises, tais como: desenvolvimento de aplicações à grande porte, mensuração do desempenho da ferramenta PhoneGap com outros frameworks RESTs, e em outras linguagens de programações.

Trabalhos futuros que podem ser desenvolvidos à partir deste estudo:

- Benchmark entre aplicações híbridas e entre aplicações nativas mobile.
- Desenvolvimento de aplicações mobile de pequeno à medio porte.

- Comparação da ferramenta PhoneGap com as demais disponíveis no mercado.

10.REFERÊNCIAS BIBLIOGRÁFICAS

SHOTTS, KERRI. *PHONEGAP FOR ENTERPRISE*. United Kingdom: Packt Publishing Ltd, 2014.

GHATOL, ROHIT. *BEGINNING PHONEGAP*. Indianapolis: John Wiley & Sons, Inc, 2012.

CHARLAND, LEROUX. *COMMUNICATIONS OF THE ACM*. ACM QUEUE, 2011.

OnLine: <http://dl.acm.org/citation.cfm?id=1941504> , recuperado em 11/11/2015.

Tecmundo. OnLine: <http://www.tecmundo.com.br> , recuperado em 05/09/2015.

Apple. OnLine: <http://www.apple.com/hotnews/thoughts-on-flash/> , recuperado em 05/09/2015.

HostMyApple. OnLine: <https://www.hostmyapple.com/macvps.html> , recuperado em 18/11/2015.

Phonegap. OnLine: <http://phonegap.com/> , recuperado em 28/08/2015.

FastShop. OnLine: http://www.fastshop.com.br/loja/mac-apple/macbook-pro-de-13-3-core-i5--500gb-aemd101bza-prd--6-md101bza-fast?cm_re=FASTSHOP%3aSub-departamento%3aMac--Vitrine+03--AEMD101BZA, recuperado em 18/11/2015.

Forbes. OnLine: <http://www.forbes.com/sites/joshsteimle/2014/02/12/mobile-is-the-future-of-everything/> , recuperado em 01/11/2015.