

ESTUDO DA TECNOLOGIA PHONEGAP/CORDOVA E A APLICAÇÃO EM UM ESTUDO DE CASO

Férilon M. Piran*

Alexandre T. Lazzaretti**

RESUMO

Com o constante crescimento no uso de dispositivos móveis a demanda por aplicações para estes dispositivos também cresce. Isso motivou o estudo sobre as tecnologias de desenvolvimento de aplicações móveis, dentre as quais estão classificadas como: a tecnologia nativa das plataformas, a tecnologia web, e a tecnologia híbrida. Dessas a tecnologia híbrida combina as características das outras duas citadas. Com o estudo escolheu-se o framework PhoneGap/Cordova que representa a tecnologia híbrida, pelo fato de atender e se adaptar as características da aplicação proposta no caso de uso, que consiste no desenvolvimento de uma aplicação multiplataforma para interação entre usuários, através de um mural de mensagens e recados, onde o usuário poderá visualizar as mensagens de acordo com a categoria do assunto, e postar novas mensagens e recados no mural para interagir com outros usuários. Este trabalho tem como principal objetivo a aplicação da tecnologia híbrida através do framework PhoneGap/Cordova no desenvolvimento de uma aplicação móvel multiplataforma apresentando suas particularidades e características, assim como os passos para sua instalação e comandos para criar um projeto, adicionar as plataformas móveis, estrutura do projeto e testes da aplicação.

Palavras-chave: Aplicações Móveis. Aplicações Híbridas. Dispositivos Móveis.

* Férilon M. Piran, aluno do curso de Sistemas de Informação para Internet pelo Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense / Campus Passo Fundo (IFSul).

E-mail: ferllon.mp@gmail.com Rua João Flores, 488 - Passo Fundo-RS.

** Alexandre T. Lazzaretti, professor orientador, Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense / Campus Passo Fundo (IFSul).

E-mail: alexandre.lazzaretti@passofundo.ifsul.edu.br Estrada Perimetral Leste, 150 - Passo Fundo-RS.

INTRODUÇÃO

Com o crescimento das vendas dos dispositivos móveis (DRSKA, 2013.), a demanda por aplicações móveis também tem acompanhado abrindo uma nova oportunidade de mercado, que consiste em criar aplicações para esses dispositivos, que estão presentes no cotidiano das pessoas seja para trabalho ou lazer. Em uma pesquisa da Appnition (MERCADO, 2013), foi divulgado que a receita com aplicativos móveis vai passar de US\$ 72 bilhões em 2013 para US\$ 151 bilhões em 2017, essa projeção inclui receitas de vendas de aplicativos móveis pagos, vendas in-app, publicidade e vendas de bens e serviços através dos aplicativos. Este último item, segundo a pesquisa, corresponderá a cerca de 75% desse valor. Além dos smartphones e tablets, já bem populares, as novas plataformas como Smart Tvs e carros conectados serão uma nova oportunidade para o mercado de aplicativos. As aplicações móveis são sistemas desenvolvidas para ser instaladas ou rodar nessas plataformas, e podem ser voltadas para o lazer, entretenimento, jogos, redes sociais ou sistemas corporativos utilizados para trabalhar, entre outros.

Um dos problemas encontrados no desenvolvimento dessas aplicações é a escolha da tecnologia, uma vez que existem várias plataformas, sistemas operacionais móveis diferentes, versões de sistemas operacionais diferentes, tamanhos de telas diferentes. Este trabalho propõe um estudo sobre as tecnologias de desenvolvimento de aplicações móveis, abordando as principais plataformas e tecnologias para criação destas aplicações.

Para o desenvolvimento deste trabalho observou-se pesquisas e trabalhos relacionados com os temas, Desenvolvimento em Smartphones – Aplicativos Nativos e Web que aborda as tecnologias Nativas e Web, Revisão de Literatura de Frameworks de Desenvolvimento Móvel Multiplataforma, Desenvolvimento de aplicações multiplataforma para dispositivos móveis entre outras referências citadas no trabalho.

A tecnologia adotada para o desenvolvimento da aplicação referente ao estudo de caso é o framework PhoneGap, que possibilita o desenvolvimento de uma aplicação que seja multiplataforma utilizando tecnologias web como, Linguagem de Marcação de Hipertexto (HTML5), JavaScript e Linguagem de Folhas de Estilo (CSS3). Essa tecnologia possui suporte aos recursos de hardware das plataformas através de Interface de Programação de Aplicações (API), no decorrer deste trabalho será apresentado as características da tecnologia utilizada.

O trabalho está organizado em três partes principais: na primeira são apresentadas as principais tecnologias para o desenvolvimento de aplicações móveis, que são: tecnologia

nativa das plataformas, tecnologia web e a tecnologia híbrida; na segunda parte são apresentadas as características da tecnologia híbrida através do framework PhoneGap/Cordova, aplicada no estudo de caso proposto neste trabalho, por fim, apresenta-se os resultados obtidos com a pesquisa e aplicação do estudo de caso no desenvolvimento de uma aplicação multiplataforma para dispositivos móveis.

1. REFERENCIAL TEÓRICO

1.1 Aplicações Nativas

As aplicações móveis nativas são desenvolvidas para executar em uma plataforma específica, sendo instalada diretamente no sistema operacional da plataforma e funcionam em modo off-line dispensando a conexão com a internet. Para se desenvolver uma aplicação nativa precisa-se escrever a aplicação na linguagem nativa de cada plataforma, como por exemplo Java e XML no caso do Android, Objective-C no caso do iOS e no caso do Windows Phone C#, nesse modelo de aplicação os recursos das plataformas como acelerômetro, geolocalização, contatos, câmera entre outros podem ser amplamente explorados. (TOLEDO; DEUS, s.d.)

Esse tipo de aplicação exige mais conhecimento por parte do programador que precisa conhecer cada uma das linguagens de programação das plataformas que pretende desenvolver, e isso pode ser considerado uma dificuldade, pois demanda mais tempo de aprendizado das linguagens, maior tempo de desenvolvimento pois para escrever um aplicativo para as três plataformas citadas é preciso escrever três aplicativos em três linguagens de programação diferentes, essas aplicações ficam disponíveis nas lojas oficiais. (TOLEDO; DEUS, s.d.)

Na Tabela 1 segue uma ilustração das tecnologias utilizadas entre as plataformas.

Tabela 1 - Tecnologias utilizadas entre as aplicações nativas.

Sistema Operacional	Linguagem de desenvolvimento	IDE	Sistema Operacional de desenvolvimento
Android	Java e XML	Eclipse / Android Studio	Windows, Linux, OS X
iOS	Objective-C	XCode	OS X
Windows Phone	C#	Visual Studio	Windows

Fonte: Transcrita site Devmedia.

O Android nativamente usa linguagem orientada a objetos Java e XML para o desenvolvimento de aplicações nativas. De acordo com MEDNIKES. et. al. (2013. P.53)

Java é uma linguagem orientada a objetos que se concentra não em tipos primitivos, mas, sim, em objetos – combinações de dados e procedimentos para operações nesses dados. Uma classe define campos (dados) e métodos (procedimentos) que compõem o objeto.

As aplicações nativas para iOS são desenvolvidas com a linguagem Objective-C. “Objective-C é uma linguagem orientada a objetos que adicionada ao C a transmissão de mensagens em estilo Smalltalk. A linguagem é um superconjunto da linguagem C, fornecendo estruturas que lhe permitem definir classes e objetos”. (ALLAN, 2013, p.70)

A plataforma Windows Phone tem suas aplicações nativas desenvolvidas na linguagem C#, ele é baseado em C++ e Java mas também inclui algumas ideias de Delphi e Visual BASIC, o seu propósito é ser uma linguagem de programação para o desenvolvimento de software baseado em componentes, especialmente para tal desenvolvimento no framework .NET. (SEBESTA, 2010)

1.2 Aplicações Web

Este modelo de aplicação depende de conexão com a internet para poder ser acessada, pois essa aplicação não é instalada na plataforma como a nativa, e sim acessada através do navegador da plataforma, como se fosse um site. Um ponto determinante para o bom funcionamento desse modelo de aplicação e a conexão com a internet que influencia diretamente no funcionamento da aplicação como por exemplo, velocidade, estabilidade, e disponibilidade. Um outro ponto dessas aplicações é que elas não tem acesso aos recursos de hardware da plataforma como câmera, contatos, acelerômetro entre outros. E é preciso se levar em conta na hora de se desenvolver uma aplicação web os diferentes tamanhos de tela dos inúmeros dispositivos existentes, assim como as versões de navegadores dos dispositivos que podem interpretar de forma diferente o código da aplicação e apresentar erros, isso pode ser considerado uma dificuldade na hora de se desenvolver uma aplicação web. (TOLEDO; DEUS, s.d.)

Um ponto positivo desse modelo de aplicação é que as linguagens utilizadas são mais comuns à maioria dos desenvolvedores, tornando mais fácil o desenvolvimento da aplicação,

outro ponto é que se desenvolve uma aplicação que será executada por todas as plataformas tornando o processo de desenvolvimento mais ágil, uma vez que apenas uma aplicação é desenvolvida para múltiplas plataformas. (TOLEDO; DEUS, s.d.)

Este tipo de aplicação é desenvolvida com linguagem de programação que é interpretada pelos navegadores web, tais como HTML5, CSS3, JavaScript, entre outras.

HTML é uma linguagem de marcação de hipertexto que é utilizada para a publicação de conteúdo de texto e multimídia na internet, baseado no conceito de Hypertext, uma forma não linear de organizar conteúdos através de um conjunto de elementos ligados por conexões, assim esses elementos ligados formam uma grande rede de comunicação. (EIS; FERREIRA, 2012)

CSS é uma linguagem de folha de estilo em cascata que é usada para formatar as informações escritas em HTML que pode ser qualquer coisa, como texto, áudio, vídeo e etc. (EIS; FERREIRA. 2012)

O JavaScript foi criado pela Netscape baseado no C, C++ e no Perl. Ela é uma linguagem de scripting interpretada que deve ser usada em conjunto com outro tipo de linguagem como HTML por exemplo. (REMOALDO, 2008)

1.3 Aplicações Híbridas

Aplicações híbridas são desenvolvidas para rodar em multiplataforma usando tecnologia web como HTML5, CSS e JavaScript, porém esse tipo de aplicação diferencia-se da web pois ela é instalada na plataforma e funcionam em modo off-line. Essas aplicações unem características das aplicações web e nativas, ou seja uma aplicação híbrida é multiplataforma como a web, e assim como as nativas tem acesso aos recursos de hardware da plataforma através de API's, e é instalada na plataforma funcionando em modo off-line e on-line. (FROZZA, s.d.)

Entre os frameworks mais utilizados para o desenvolvimento de aplicações híbridas encontra-se o PhoneGap/Cordova ele serve como uma ponte entre as aplicações e os recursos de hardware das plataformas. (FROZZA, s.d.)

Essas aplicações são desenvolvidas com tecnologia web como HTML5 e um container nativo que permite acessar os recursos da plataforma, o mais usado é Open Source Apache Cordova, o PhoneGap que é uma distribuição do Cordova, eles oferecem um conjunto de API's que permite acessar os recursos dos dispositivos através da aplicação em HTML5, nesse tipo de tecnologia a parte principal do código fonte é escrito em HTML5, ficando

apenas a parte de comunicação com os recursos da plataforma como código específico para cada plataforma desejada. Com isso o desenvolvimento fica mais rápido e mais barato a medida que a aplicação desenvolvida com essa tecnologia é multiplataforma, fica disponível nas lojas de aplicativos nativos de cada plataforma assim como as aplicações nativas. (CHEDE, 2013)

2. RESULTADOS E DISCUÇÕES

2.1 Escolha da Tecnologia PhoneGap

Dentre as tecnologias estudadas a escolhida para aplicação no caso de uso foi a tecnologia híbrida para o desenvolvimento de aplicações móveis, pois se trata de uma aplicação multiplataforma, com o objetivo de ser instalada na plataforma do usuário para que ele possa interagir a partir da postagem de mensagens em mural digital. Essa tecnologia permite criar apenas uma aplicação, a qual se adapta as plataformas móveis sem a necessidade de se criar uma aplicação para cada plataforma. Com essa tecnologia a aplicação é criada com uso das tecnologias web, o que facilita o trabalho uma vez que não é necessário o conhecimento específico de cada uma das tecnologias usadas para a criação de aplicações nativas das plataformas.

Outro ponto determinante para a escolha é o tempo de desenvolvimento, uma vez que uma única aplicação atende as múltiplas plataformas pretendidas, isso é um facilitador na hora de se desenvolver uma aplicação móvel. Com base nestes pontos citados a escolha foi o uso da tecnologia híbrida através do framework PhoneGap, sua escolha se dá por se tratar de uma tecnologia open source, com API's¹ para os mais diversos recursos dos dispositivos, documentação disponível. Outro determinante na escolha é a possibilidade de disponibilizar a aplicação desenvolvida com essa tecnologias nas lojas oficiais das plataformas assim como as aplicações nativas, o que ajuda na popularização da aplicação.

2.2 PhoneGap

PhoneGap é um framework open source para o desenvolvimento de aplicações multiplataforma usando as tecnologias web HTML5, JavaScript e CSS. Ele utiliza tecnologia

¹ API – Interface de Programação de Aplicação.

de contêiner de aplicativo que possibilita a criação de aplicações nativas que são instaladas nos dispositivos móveis e possibilita o acesso aos recursos de hardware das plataformas como câmera, contatos, geolocalização e etc., através de API's, seu funcionamento pode ser online ou off-line. (PHONEGAP, s.d.)

A seguir são descritos alguns dos principais pontos desta tecnologia.

2.2.1 Interface WebView

A interface do usuário é desenvolvida com HTML5², CSS3³ e JavaScript, essa camada de interface é interpretada por um navegador web que ocupa 100% da largura e 100% da altura da tela do dispositivo, como ilustrado na Figura 1. (TRICE, 2012)

Figura 1 – Representação da WebView do PhoneGap.



Fonte: Site PhoneGap.

A aplicação é desenvolvida para ocupar todo o espaço da tela sem a decoração de um navegador comum, a visão web usada pelo PhoneGap é a mesma usada pelo sistema operacional nativo, como a classe Objective-C UIWebView no caso do iOS e android.webkit.WebView no caso do Android, é preciso certificar-se das diferenças de renderização WebView de cada sistema operacional para que a aplicação funcione perfeitamente em todos os dispositivos. (TRICE, 2012)

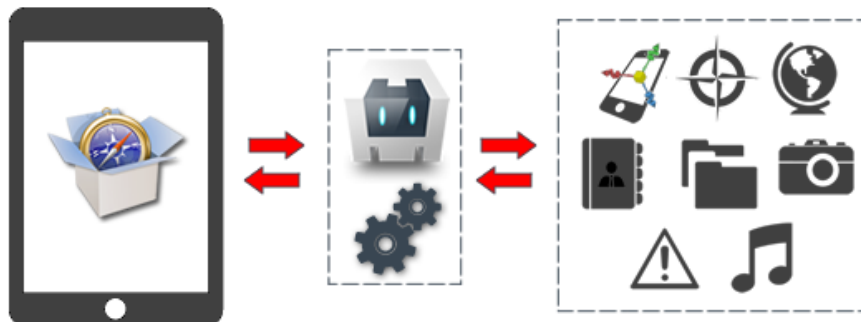
² HTML – Linguagem de Marcação de Hypertexto.

³ CSS – Linguagem de Folha de Estilo.

2.2.2 Recursos Nativos.

O PhoneGap fornece uma API que possibilita ao desenvolvedor acessar os recursos do sistema operacional nativo usando JavaScript, onde o desenvolvedor cria a lógica da aplicação usando JavaScript e a API se encarrega da comunicação com o sistema operacional nativo da plataforma, como mostra a Figura 2. (TRICE, 2012)

Figura 2 - Comunicação entre Interface e recursos nativos através da API.



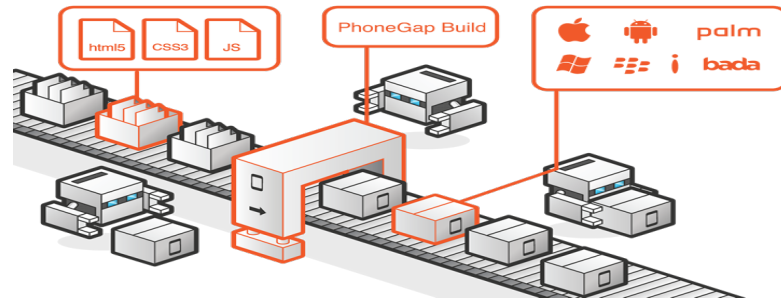
Fonte: Site PhoneGap.

2.2.3 Adobe PhoneGap Build

O Adobe PhoneGap Build (<https://build.phonegap.com/>) é uma ferramenta desenvolvida pela Adobe que facilita a compilação do projeto. Com ele é possível compilar o projeto para seis plataformas diferentes ao mesmo tempo, o projeto pode ser enviado no formato .zip ou lido diretamente do GitHub (<https://github.com>) que possui integração com o Adobe PhoneGap Build. (BUILD, s.d.).

As aplicações desenvolvidas com o PhoneGap são construídas com HTML5, CSS3 e JavaScript. Porém a aplicação final é um arquivo binário que pode ser distribuído nas lojas oficiais de cada plataforma, como por exemplo, Google Play do Android, App Store no caso do iOS e MarketPlace no caso do Windows Phone. Isso é possível através do serviço de nuvem da Adobe PhoneGap Build que empacota o arquivo desenvolvido com as tecnologias web gerando aplicações nos formatos de aplicativos nativos de cada plataforma desejada, como ilustra a Figura 3. (TRICE, 2012)

Figura 3 – Construção da aplicação para multiplataforma através do Adobe PhoneGap Build.



Fonte: Site PhoneGap.

2.2.4 Uso de Banco de Dados

A aplicação desenvolvida com PhoneGap age com um cliente que se comunica com servidor de aplicativos para receber os dados, onde o servidor de aplicativo lida com as regras de negócio e se comunica com a base de dados back-end, como ilustrado na Figura 4. (TRICE, 2012)

Figura 4 - Comunicação Cliente-Servidor.



Fonte: Site PhoneGap.

Os servidores de aplicativo geralmente são servidores web com Apache, ISS, etc..., que usam linguagens de script como Java, PHP, NET, etc..., o PhoneGap é independente das tecnologias de back-end, podendo trabalhar com qualquer servidor de aplicativo através dos protocolos padrão da web, onde o servidor executa as regras de negócio e recupera ou persiste dados a partir de uma base de dados que geralmente é relacional, mas pode ser qualquer outra estrutura ou mecanismo para persistência de dados. (TRICE, 2012)

Geralmente a comunicação com a base de dados não é direta, ela é intermediada pelo servidor de aplicativo, onde o cliente, que no caso é a aplicação desenvolvida com PhoneGap, faz solicitações HTTP padrão para conteúdo HTML, como REST-Full XML service, JSON services, ou SOAP. (TRICE, 2012)

As aplicações PhoneGap geralmente utilizam um modelo de aplicação de tela única, onde a lógica da aplicação fica em uma única página HTML. Essa página fica carregada na memória e todos os dados são exibidos através do HTML DOM, os dados vêm do servidor de aplicativo através de requisições Ajax, e as variáveis são mantidas em memória dentro do JavaScript. As aplicações multi-páginas com PhoneGap podem ser construídas, porém não são recomendadas porque os valores em memória das variáveis são perdidos ao carregar uma página separada. (TRICE, 2012)

O PhoneGap também oferece suporte para o SQLite, tecnologia para persistência de dados independente e transicional, presente na maioria das plataformas de dispositivos móveis. Esse suporte se dá através de um plugin que devem ser instalados junto ao projeto, por exemplo para a plataforma Android e iOS pode-se instalar o plugin ao seu projeto através do uso do Cordova Interface de linha de Comando (CLI), com o comando: (HIRAL, 2014)

- `cordova plugin add https://github.com/brodysoft/Cordova-SQLitePlugin.git`

Após a instalação do plugin no projeto, basta criar as funções para manipulação do banco de dados no arquivo JavaScript do seu projeto, a Figura 5 ilustra o exemplo de três funções para a manipulação da base de dados, a primeira função é para criar ou abrir o banco de dados, a segunda função é para criar uma tabela onde os dados serão inseridos, e a terceira função mostra um exemplo de como inserir dados em uma tabela do banco de dados. (HIRAL, 2014).

Figura 5 - Funções para manipulação do Banco de Dados SQLite.

```
// CRIAR OU ABRIR BANCO DE DADOS
document.addEventListener("deviceready", onDeviceReady, false);
var db;
function onDeviceReady() {
  db = window.sqlitePlugin.openDatabase({name: "DB"});
}

// CRIAR TABELA
db.transaction(function(tx) {
  tx.executeSql('CREATE TABLE IF NOT EXISTS test_table \n\
(id integer primary key, data text, data_num integer)');
});

// INSERIR DADOS NO BANCO DE DADOS
db.transaction(function(tx) {
  tx.executeSql("INSERT INTO test_table (data, data_num) VALUES (?,?)",
  ["test", 100],
  function(tx, res) {
    console.log("insertId: " + res.insertId + " -- probably 1");
    console.log("rowsAffected: " + res.rowsAffected + " -- should be 1");
  });
});
```

Fonte: Do autor.

2.2.5 API's de suporte as plataformas

O PhoneGap oferece suporte para plataformas móveis através de API's de desenvolvimento, que possibilitam acesso aos recursos de hardware dos dispositivos, como câmera, contatos, geolocalização, acelerômetro, mídias, arquivos entre outros, isso permite desenvolver aplicação com funções semelhantes as nativas, na Figura 6 apresenta-se as API's disponíveis para cada plataforma. (PHONEGAP, s.d.).

Figura 6 - API's de desenvolvimento PhoneGap.

	amazon- fireos	android	blackberry10	Firefox OS	ios	Ubuntu	wp7 (Windows Phone 7)	wp8 (Windows Phone 8)	win8 (Windows 8)	tizen
Platform APIs										
Accelerometer*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Capture*	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗
Compass*	✓	✓	✓	✗	✓ (3GS+)	✓	✓	✓	✓	✓
Connection*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Contacts*	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
Device*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Events	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
File*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Geolocation*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Globalization*	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗
InAppBrowser*	✓	✓	✓	✗	✓	✓	✓	✓	(partial)uses iframe	✗
Media*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Notification*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Splashscreen*	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
Storage	✓	✓	✓	✗	✓	✓	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓ localStorage & indexedDB	✓

Fonte: Site PhoneGap.

2.3 Instalação do PhoneGap e Cordova

Para trabalhar com o PhoneGap/Cordova é preciso instalar o Cordova Interface de Linha de Comando (CLI), ferramenta que permite a criação de novos projetos, construí-los em plataformas diferentes e executa-los em emuladores ou em dispositivos reais. Para executar as tarefas de linha de comando da CLI é necessário que se tenha o kit de Desenvolvimento de Software (SDK) de cada plataforma que se pretende trabalhar instaladas no computador. A Tabela 2 apresenta uma ilustração das combinações que a CLI suporta. (PHONEGAP, s.d.).

Tabela 2 - Ilustração das combinações da CLI.

	iOS	Amazon Fire OS	Android	BlackBerry	Windows Phone7	Windows Phone 8	Windows 8	Firefox OS
Linux	X	√	√	√	X	X	X	√
Mac OS	√	√	√	√	X	X	X	√
Windows	X	√	√	√	√	√	√	√

Fonte: Site PhoneGap.

A seguir são descritos os passos e requisitos para instalação do PhoneGap e Cordova.

I. Pré-requisitos

- 1 - Instalação do Node.js⁴ (<http://nodejs.org/>), os pacotes são distribuídos como pacotes npm.
- 2 - Instalação do Git Cliente⁵ (<http://git-scm.com/>), o Cordova faz uso dele em segundo plano.

II. Instalação

Linux/Unix:

- `sudo npm install -g phonegap`
- `sudo npm install -g cordova`

Windows:

- `C:\> npm install -g phonegap`
- `C:\> npm install -g cordova`

III. Criar projeto

Para criar um projeto pelo terminal, é necessário criar uma pasta onde serão armazenados os projetos, acesse a pasta pelo terminal e digite o comando:

- `cordova create teste com.exemplo.teste Teste`

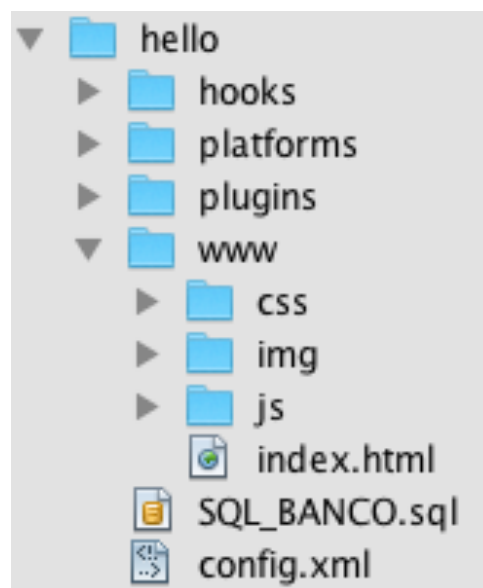
⁴ NodeJS – Ferramenta para construção de aplicações de rede escaláveis.

⁵ Git Client – Ferramenta para controle de versão de projetos.

O primeiro argumento “teste” é o nome de um diretório que será criado pelo Cordova e que conterá a estrutura do projeto dentro dele, o segundo argumento “com.exemplo.teste” é um identificador de domínio reverso e o terceiro argumento “Teste” é o título de exibição da aplicação.

A estrutura do projeto criado pode ser vista na Figura 7, onde dentro da pasta “www” temos o arquivo index.html que é a página principal da aplicação que deve ser editada para criar a aplicação desejada.

Figura 7 - Estrutura do projeto PhoneGap/Cordova.



Fonte: Do autor.

IV. Adicionar o projeto à plataforma e teste:

Para adicionar o projeto a uma plataforma como iOS por exemplo é necessário estar dentro do diretório do projeto e executar o comando:

- `cordova plataforma add ios`

E assim para todas as plataformas que pretende executar a aplicação, basta substituir o “iOS” pela plataforma desejada, como “android” por exemplo, lembrando que para funcionar é necessário estar com o SDK da plataforma desejada instalado em seu computador. Também é possível listar as plataformas adicionadas da aplicação com o comando:

- *cordova platforms ls*

Também é possível remover uma plataforma com o comando:

- *cordova rm plataforma android*

Para testar a aplicação em um emulador ou em um dispositivo basta digitar o comando:

- *cordova run android*

Também é possível abrir o projeto no Ambiente Integrado de Desenvolvimento (IDE) como o Eclipse ou Android Studio usados para aplicações Android ou XCode no caso do iOS, podendo-se testar a aplicação a partir da IDE.

2.4 Estudo de caso

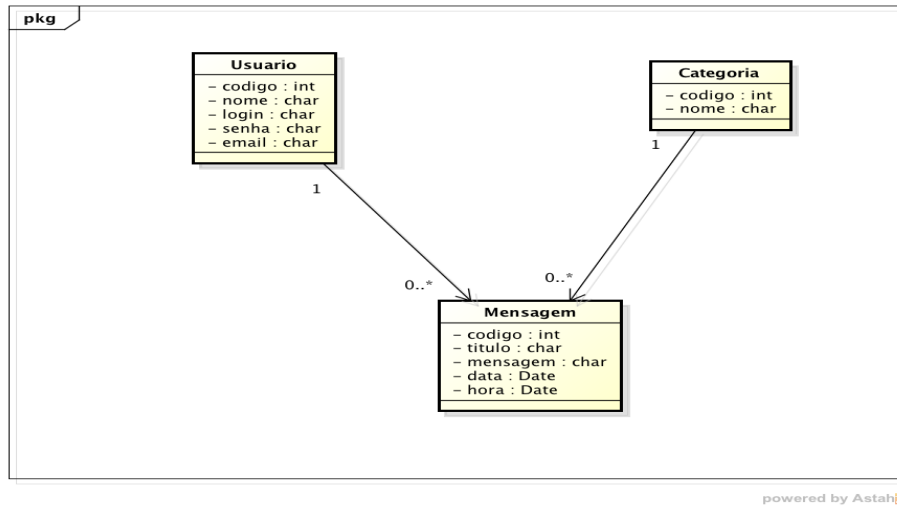
O estudo de caso consiste no desenvolvimento de uma aplicação para dispositivos móveis que seja multiplataforma, com o objetivo de ser um mural digital para recados e mensagens, onde o usuário irá visualizar todos os recados e mensagens listados em ordem decrescente por data e hora, posteriormente o usuário poderá filtrar as mensagens exibidas por categorias de assuntos, as mensagens postadas pelo usuário que está visualizando contam com uma opção de apagar. As mensagens são compostas por título, categoria, mensagem, e-mail do usuário que está postando, data e hora da postagem. O usuário informa o título, seleciona a categoria, escreve a mensagem de no máximo cento e cinquenta (150) caracteres, os demais dados vêm do seu cadastro na aplicação. A seguir são mostrados o diagrama de classe, interfaces e código fonte da aplicação.

2.4.1 Diagrama de classes

O diagrama de classes apresentado na Figura 8 demonstra as classes da aplicação móvel, que conta com um servidor remoto com a base de dados onde as informações de usuário e mensagens são armazenados, esta base de dados foi desenvolvida com a tecnologia

PostgreSQL, e é acessada remotamente através de requisições feitas em JavaScript utilizando a tecnologia Ajax.

Figura 8 - Diagrama de classes do caso de uso.

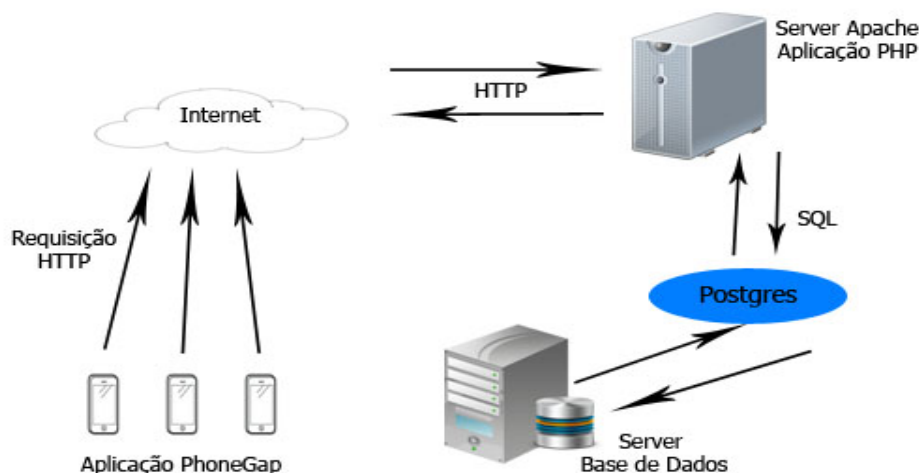


Fonte: Do autor.

2.4.2 Comportamento da Aplicação

A figura 9 apresenta uma ilustração sobre o comportamento da aplicação durante sua execução, onde as requisições de dados são realizadas através da internet com protocolo padrão web a um servidor remoto Apache, que contém uma aplicação PHP responsável pela comunicação com a base de dados.

Figura 9 - Ilustração do comportamento da aplicação.



Fonte: Do autor.

2.4.3 Interfaces da aplicação

Para visualizar ou postar mensagens, anteriormente o usuário precisa realizar o cadastro e posteriormente fazer login. A tela inicial disponibiliza as entradas de usuário e senha para autenticação, permitindo o acesso as funções do sistema, também disponibiliza um botão para realizar o cadastro de novo usuário, caso o mesmo ainda não seja cadastrado, como mostra a Figura 10.

Figura 10 - Tela de início da aplicação.

A imagem mostra a tela de início da aplicação. No topo, há uma barra preta com o texto "INÍCIO" em branco. Abaixo, há um formulário com dois campos de entrada: "USUÁRIO:" e "SENHA:". O campo "SENHA:" está selecionado com um cursor de texto. Abaixo dos campos, há dois botões: "ENTRAR" e "NOVO USUÁRIO". Na base da tela, há uma barra preta com o texto "TSPI IFSUL - PF/RS" em branco.

Fonte: Do autor.

Na tela de novo usuário é necessário informar o nome, usuário, senha, confirmação de senha e e-mail. Para efetivar o cadastro de usuário na aplicação, todos os campos são obrigatórios, a Figura 11 ilustra a tela de cadastro de novo usuário onde ele deve informar seus dados como, nome, usuário, senha, confirmação de senha e e-mail para realizar o cadastro e ter acesso as mensagens da aplicação. Essa tela conta com um botão de “Cadastro” para o usuário finalizar o seu cadastro e um botão chamado “Início” para o usuário voltar a tela de início e fazer sua autenticação na aplicação.

Figura 11 - Tela de cadastro de usuário.

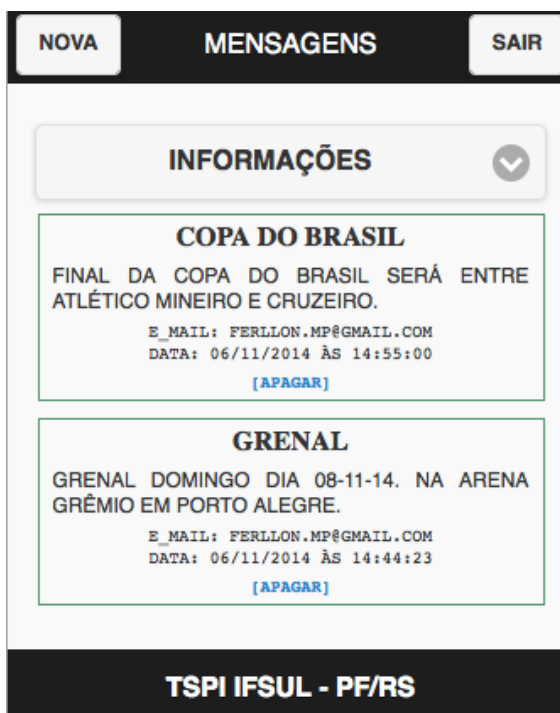


A tela de cadastro de usuário apresenta um layout com um cabeçalho escuro contendo dois botões: "INÍCIO" e "CADASTRO". Abaixo, há campos de entrada para "NOME:", "USUÁRIO:", "SENHA:", "CONFIRMA SENHA:" e "E-MAIL:". Um botão "CADASTAR" está posicionado na base dos campos. Na barra de rodapé, o texto "TSPI IFSUL - PF/RS" é exibido.

Fonte: Do autor.

Após a autenticação do usuário, ele é redirecionado para a tela que representa o mural de postagens, onde as mensagens e recados podem ser visualizados. As mensagens são compostas por título, categoria, mensagem, e-mail do usuário que realizou a postagem, data, hora e uma opção para serem deletadas nas mensagens postadas pelo usuário autenticado na aplicação, são listadas todas as mensagens em ordem decrescente por data e hora, podendo o usuário filtrá-las por categorias, como na Figura 12.

Figura 12 - Tela das mensagens.



Fonte: Do autor.

Esta tela ainda é composta por dois botões para as funções de postar uma nova mensagem e sair, caso escolha a opção sair o usuário sai da aplicação encerrando sua sessão, caso a opção seja nova mensagem o usuário é redirecionado para a tela de postagem de mensagens onde ele deverá informar o título da mensagem, selecionar a categoria e escrever o recado ou mensagem de no máximo cento e cinquenta (150) caracteres. Todos os campos são obrigatórios, a mensagem ainda é composta por data, hora e e-mail do usuário que postou a mensagem, essas informações são inseridas pela aplicação de forma automática, pegando a data e hora do sistema na hora da postagem da mensagem e o e-mail do cadastro do usuário, a Figura 13 ilustra a tela de postagem das mensagens.

Figura 13 - Tela nova mensagem.

Fonte: Do autor.

Essa tela conta com três botões, MSGs, Sair e Postar. Caso escolha a opção “MSGs” o usuário é redirecionado para a tela de mensagens, caso opte por escolher sair ele sai da aplicação encerrando sua sessão, a opção postar registra a mensagem na base de dados e redireciona o usuário para a tela de mensagens.

2.4.4 Código fonte

No projeto desenvolvido a partir do estudo de caso com a tecnologia PhoneGap e Cordova utilizou-se o framework jQueryMobile (<http://jquerymobile.com/>), compatível com PhoneGap e Cordova, para a construção da visão, interface com o usuário, a Figura 14 apresenta um trecho do código para a utilização do jQueryMobile juntamente com o PhoneGap, e o link para o arquivo JavaScript que faz o controle das funções da aplicação, chamado de “meuScript”.

O jQuery Mobile é um framework open open source baseado em HTML5, projetado para o desenvolvimento de sites e aplicativos para dispositivos móveis.

Figura 14 - Script para jQuery Mobile e para controle da aplicação.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <!-- WARNING: for iOS 7, remove the width=device-width and height=device-heig
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-sca
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <meta name="msapplication-tap-highlight" content="no" />
    <title>Férлон - Testando Cordova</title>
    <!-- Aqui para JavaScript -->
    <script type="text/javascript" src="js/meuScript.js"></script>
    <!-- Links para framwork JQuery Mobile -->
    <script type="text/javascript" src="js/jquery.js"></script>
    <script type="text/javascript" src="js/jquery.mobile-1.4.3.min.js"></script>
    <link rel="stylesheet" href="js/jquery.mobile-1.4.3.min.css" />
  </head>
  <body>

```

Fonte: Do autor.

A Figura 15 apresenta um trecho do código da visão que representa a tela de início da aplicação. Todas as telas da aplicação foram criadas em um único arquivo HTML que é o arquivo index.html criado junto com projeto através do terminal com o comando:

- *"cordova create teste com.exemplo.teste Teste"*

Esse arquivo index.html fica dentro do diretório www, dentro deste diretório encontra-se também o diretório js onde encontra-se o arquivo JavaScript que faz o controle das funções da aplicação e comunicação com o banco de dados remoto através de requisições feitas em Ajax nas funções em JavaScript, ainda neste diretório encontra-se os diretórios css e img, que contém a folha de estilo e as imagens da aplicação respectivamente.

Figura 15 - Código da tela de Início.

```

<!-- ##### LOGIN ##### -->
<div data-role="page" id="login" data-theme="a">
  <div data-role="header" data-theme="b">
    <h1>Início</h1>
  </div><!-- /header -->
  <div data-role="content">
    <form action="#" method="post" name="formLogin">
      <label for="login">Usuário:</label>
      <input type="text" name="login" id="login" data-mini="true"/>
      <label for="senha">Senha:</label>
      <input type="password" name="senha" id="senha" data-mini="true"/>
      <input type="button" onclick="verificaLogin()" value="Entrar"
        data-mini="true" data-theme="b"/>
    </form>
    <a href="#cadastroUser" data-role="button" data-theme="b"
      data-mini="true">Novo Usuário</a>
  </div><!-- /content -->
  <div id="conteudo"></div>
  <div data-role="footer" data-theme="b">
    <h4>TSPI IFSUL - PF/RS</h4>
  </div><!-- /footer -->
</div><!-- FINAL PAGINA DE LOGIN -->

```

Fonte: Do autor.

O controle das ações da aplicação assim como a validação de campos é realizado através de um arquivo de JavaScript chamado de “meuScript”, que contém as funções da aplicação como por exemplo verificação de login e senha através de requisições de conteúdo feitas a base de dados, como exemplificado na Figura 16.

Figura 16 - Função JavaScript.

```

function verificaLogin() {
    var var_login = document.formLogin.login.value;
    var var_senha = document.formLogin.senha.value;
    //Limpa o Forma
    document.formLogin.reset();
    if (var_login == "") {
        alert("Preenha Login");
    } else {
        if (var_senha == "") {
            alert("Preencha a Senha");
        } else {
            $.ajax({
                type: "POST",
                url: url_serv,
                data: ({operacao: "verificaLogin",
                    login: var_login,
                    senha: var_senha}),
                cache: false,
                dataType: "text",
                success: resLogin
            }).error(function() {
                alert('Erro Requisição');
            });
        }
    }
}

function resLogin(res) {
    if (res == "true") {
        location.href = "index.html#mensagens";
        mensagens();
    } else {
        alert("Erro de Login!");
        location.href = "index.html";
    }
}
};

```

Fonte: Do autor.

Os dados vêm de uma base de dados desenvolvida com a tecnologia PostgreSQL e fica em um servidor remoto, que é acessado pelos dispositivos de forma remota usando requisições em Ajax dentro das funções JavaScript, como pode ser visto na Figura 17.

Figura 17 - Requisição em Ajax ao servidor remoto.

```
$.ajax({  
    type: "POST",  
    url: url_serv,  
    data: ({operacao: "verificaLogin",  
            login: var_login,  
            senha: var_senha}),  
    cache: false,  
    dataType: "text",  
    success: resLogin  
}).error(function() {  
    alert('Erro Requisição');  
});
```

Fonte: Do autor.

CONCLUSÃO

Este trabalho propôs o estudo da tecnologia de desenvolvimento de aplicações móveis híbridas através desenvolvimento de uma aplicação móvel com o uso do framework PhoneGap/Cordova. Com a aplicação desta tecnologia foi possível abordar as características do framework, que permite desenvolver uma aplicação com as tecnologias web HTML5, CSS e JavaScript e posteriormente converte-lá em uma aplicação móvel nativa que pode ser instalada nos dispositivos e distribuída nas lojas oficiais de cada uma das plataformas.

Essa tecnologia permite desenvolver uma aplicação multiplataforma para dispositivos móveis sem a necessidade de conhecimento específico nas linguagens de programação nativa das mesmas, o que pode ser considerado um ponto positivo pois isso permite desenvolver uma aplicação em menos tempo e com menor custo. Outro ponto positivo é a possibilidade de manipular os recursos de hardware dos dispositivos como câmera, contatos, geolocalização, entre outros, ela ainda permite combinação com outras tecnologias como por exemplo jQuery Mobile e Sencha Touch.

STUDY OF TECHNOLOGY PHONEGAP/CORDOVA AND APPLICATION IN A CASE STUDY

ABSTRACT

With the steady increase in the use of mobile devices demand for applications for this device also increase. This motivated the study of the technologies of mobile application development, among which are classed as native technology platforms, web technology, and hybrid technology. Hybrid technology which combines the characteristics of the two other cited. In this study we choose to PhoneGap / Cordova framework that represents the hybrid technology, the fact that better adapt to the proposed application use case that consists in developing a multiplatform application for interaction between users by a message board and messages, where the user can view the messages according to subject category, and post new messages and messages on wall to interact with other users. This work aims the implementation of the PhoneGap technology presenting their features and particularities.

Keywords: Mobile Applications. Hybrid Applications. Mobile Devices.

REFERÊNCIAS

ALLAN, Alasdair. Aprendendo Programação iOS: Do Xcode à App Store. 1º ed. São Paulo: Novatec, 2013. Disponível em:
<<http://books.google.com.br/books?id=x6qjAwAAQBAJ&pg=PA19&dq=aplicativos++nativos&hl=pt-BR&sa=X&ei=17wyVKf7NcfGgwT27ILACQ&ved=0CBwQ6AEwAA#v=onepage&q=aplicativos%20%20nativos&f=false>> Acesso em: 20 Set. 2014.

BUILD, Adobe PhoneGap. What is Build. Disponível em:
<http://docs.build.phonegap.com/en_US/introduction_what_is_build.md.html#What%20is%20Build> Acesso em: 07 Nov. 2014.

CHEDE, CEZAR (ctaurion). Desenvolvimento de apps – Parte 2: híbrido, nativo ou web?. IBM DeveloperWorks, 2013. Disponível em:
<https://www.ibm.com/developerworks/community/blogs/ctaurion/entry/desenvolvimento_de_apps-parte_2_hibrido_nativo_ou_web?lang=en> Acesso em: 09 Out. 2014.

DRSKA, Moacir. Vendas Globais de Dispositivos Móveis Crescem 37,4% no 1º trimestre. São Paulo: Valor Econômico, 2013. Disponível em:
<<http://www.valor.com.br/empresas/3117700/vendas-globais-de-dispositivos-moveis-crescem-374-no-1>> Acesso em: 10 nov. 2014.

DEVMEDIA. Aplicações Móveis: Nativas ou Web?. S.d. Disponível em: <<http://www.devmedia.com.br/aplicacoes-moveis-nativas-ou-web/30392>> Acesso em: 10 nov. 2014.

EIS, Diego; FERREIRA, Elcio. HTML5 e CSS3 com farinha e pimenta. São Paulo: Tableless, 2012. Disponível em: <<http://books.google.com.br/books?id=nDKMAwAAQBAJ&printsec=frontcover&dq=html5+and+css3&hl=pt-BR&sa=X&ei=kAEzVO7CI5eRgwTI74GoCQ&ved=0CGAQ6AEwBw#v=onepage&q=html5%20and%20css3&f=false>> Acesso em: 04 Out. 2014.

HIRAL. Integrate SQLite Plugin in PhoneGap for Android and iOS. 2014. Disponível em: <<http://www.excellentwebworld.com/integrate-sqlite-plugin-in-phonegap-for-android-and-ios/>> Acesso em: 17 nov. 2014.

MERCADO de aplicativos móveis vai movimentar US\$ 151 bilhões. Revista Exame. 2013. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/mercado-de-aplicativos-moveis-vai-movimentar-us-151-bilhoes>>. Acesso em: 10 nov. 2014.

PHONEGAP. About The Project. Disponível em: <<http://phonegap.com/about/>> Acesso em: 13 Out. 2014.

ROMEALDO, Pedro. O Guia Prático do Dreamweaver CS3 com PHP, JavaScript e Ajax. 1º ed. Lisboa: Inova, 2008. Disponível em: <<http://books.google.com.br/books?id=3BlbiA4z4n0C&pg=PA123&dq=Java+Script&hl=pt-BR&sa=X&ei=S742VPveIYyRgwTA6IGQDw&ved=0CG4Q6AEwCQ#v=onepage&q=Java%20Script&f=false>> Acesso em: 04 Out. 2014.

SEBESTA, Robert W. Conceitos de linguagem de programação. Porto Alegre: Artmed, 2010. Disponível em: <<http://books.google.com.br/books?id=vPldwBmt-9wC&pg=PA125&dq=Linguagem+C%23&hl=pt-BR&sa=X&ei=G9YyVLaDIcTGgwTo2oI4&ved=0CEIQ6AEwBw#v=snippet&q=conceito%20C%23&f=false>> Acesso em: 20 Set. 2014.

TOLEDO, Jan Miszura; DEUS, Gilmar Divino de. Desenvolvimento em Smartphones - Aplicativos Nativos e Web. Disponível em: <<http://www.cpgls.ucg.br/7mostra/Artigos/AGRARIAS%20EXATAS%20E%20DA%20TERRA/Desenvolvimento%20em%20Smartphones%20-%20Aplicativos%20Nativos%20e%20Web.pdf>> Acessado em: 20 Set. 2014.

TRICE, Andrew. PhoneGap Explained Visually. 2012. Disponível em: <<http://phonegap.com/2012/05/02/phonegap-explained-visually/>> Acesso em: 13 Out. 2014.