



UNIVERSIDADE FEDERAL DE MATO GROSSO  
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM  
SISTEMAS DE INFORMAÇÃO

**RELATÓRIO DE ESTÁGIO SUPERVISIONADO  
DESENVOLVIMENTO DE APLICAÇÕES HÍBRIDAS  
UTILIZANDO O APACHE CORDOVA: UM CASE EM  
GAMIFICAÇÃO**

**AARON LESBÃO DUMONT**

CUIABÁ – MT

2015

UNIVERSIDADE FEDERAL DE MATO GROSSO  
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM  
SISTEMAS DE INFORMAÇÃO

**RELÁTÓRIO DE ESTÁGIO SUPERVISIONADO  
DESENVOLVIMENTO DE APLICAÇÕES HÍBRIDAS  
UTILIZANDO O APACHE CORDOVA: UM CASE EM  
GAMIFICAÇÃO**

**AARON LESBÃO DUMONT**

Relatório apresentado Instituto de  
Computação da Universidade Federal de  
Mato Grosso, para obtenção do título de  
Bacharel em Sistemas de Informação.

CUIABÁ – MT

2015

UNIVERSIDADE FEDERAL DE MATO GROSSO  
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM  
SISTEMAS DE INFORMAÇÃO

AARON LESBÃO DUMONT

Relatório de Estágio Supervisionado apresentado à Coordenação do Curso de Sistemas de Informação como uma das exigências para obtenção do título de Bacharel em Sistemas de Informação da Universidade Federal de Mato Grosso

Aprovado por:

---

Prof. MSc. Jivago Medeiros Ribeiro  
Instituto de Computação  
(ORIENTADOR)

---

Prof. MSc. Karen da Silva Figueiredo  
(SUPERVISOR)

---

Prof. Msc. Raphael de Souza Rosa Gomes  
Instituto de Computação  
(Coordenador de Estágios)

## **DEDICATÓRIA**

*À minha família pelo apoio, por acreditarem em mim, me incentivarem e sempre me apoiarem nas decisões que tomei.*

*Ao Professor Mestre Jivago Medeiros, por acreditar no meu potencial, não desistir e sempre me incentivar.*

## **AGRADECIMENTOS**

*Agradeço primeiramente a Deus, por sempre ter iluminado o meu caminho, me orientado e não deixado eu desistir.*

*Agradeço a todos os professores que me transmitiram seus conhecimentos, sendo de suma importância para minha vida acadêmica e profissional.*

*Agradeço a minha família por me apoiarem e acreditarem em mim.*

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>8</b>
<b>LISTA DE TABELAS.....</b>	<b>9</b>
<b>LISTA DE SIGLAS E ABREVIATURAS .....</b>	<b>10</b>
<b>RESUMO .....</b>	<b>11</b>
<b>INTRODUÇÃO .....</b>	<b>12</b>
<b>1. REVISÃO DE LITERATURA .....</b>	<b>14</b>
1.1 GAMIFICAÇÃO .....	14
1.1.1 Gamificação na Educação.....	14
1.2 DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS.....	15
1.2.1 Desenvolvimento Híbrido.....	17
1.2.2 Apache Cordova e Adobe PhoneGap.....	18
1.2.3 Cordova Plugins.....	19
<b>2. MATERIAIS, TÉCNICAS E MÉTODOS.....</b>	<b>21</b>
2.1 ECLIPSE IDE.....	21
2.2 PLATAFORMA ANDROID .....	22
2.3 HTML5, CSS3 E JAVASCRIPT .....	22
2.4 JQUERY E JQUERY MOBILE .....	24
2.5 APACHE CORDOVA .....	25
2.5.1 Instalação do Apache Cordova, Criação e Execução dos projetos.....	25
2.6 PERSISTÊNCIA DE DADOS .....	27
2.7 MÉTODO DE DESENVOLVIMENTO .....	27
<b>3. RESULTADOS .....</b>	<b>29</b>
3.1 GiC MASTER E GiC PLAYER .....	29
3.1.1 GiC Master.....	29
3.1.2 GiC Player.....	29
3.2 FUNCIONALIDADES DESENVOLVIDAS .....	30
3.2.1 Login.....	30
3.2.2 Cadastro .....	32
3.2.3 Funcionalidades GiC Master.....	32
3.2.4 Funcionalidades GiC Player .....	35
<b>4. DIFICULDADES ENCONTRADAS .....</b>	<b>36</b>
<b>5. CONCLUSÕES.....</b>	<b>37</b>
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>38</b>
<b>ANEXOS .....</b>	<b>40</b>
ANEXO I – MODELO ENTIDADE RELACIONAMENTO DAS TABELAS CRIADAS .....	40
ANEXO II – DIAGRAMA DE CASO DE USO DE CRIAÇÃO DE UM JOGO.....	40
ANEXO III – DIAGRAMA DE CASO DE USO DA INSCRIÇÃO DO PLAYER NO JOGOS. ....	41
<b>APÊNDICES.....</b>	<b>42</b>
APÊNDICE I – CLASSE USERSERVICECONTROLLER.JAVA (PLUGIN) EM JAVA.....	42
APÊNDICE II – CLASSE INDEX.HTML (TELA DE LOGIN) EM HTML .....	43
APÊNDICE III – CÓDIGO EM JAVASCRIPT CHAMANDO O PLUGIN USERSERVICECONTROLLER, UTILIZANDO CORDOVA. ....	44

APÊNDICE IV – FUNÇÕES PARA O CALLBACK DO PLUGIN USERSERVICECONTROLLER EM JAVASCRIPT UTILIZANDO O PLUGIN DIALOG DO CORDOVA. ....	44
--	----

## LISTA DE FIGURAS

FIGURA 1 - PROCESSO DE EMPACOTAMENTO DA APLICAÇÃO DO APACHE CORDOVA (WARGO, 2015).	18
FIGURA 2 - APACHE CORDOVA ARQUITETURA DA APLICAÇÃO (WARGO, 2015).....	19
FIGURA 3 - INSTALAÇÃO DE PLUGIN DO CORDOVA. ....	20
FIGURA 4 - LISTA DE PLUGINS DO APACHE CORDOVA DISPONÍVEL EM SEU SITE. ....	21
FIGURA 5 - TELA DO ECLIPSE COM PROJETOS E CLASSES ABERTAS. ....	22
FIGURA 6 - CÓDIGO HTML. ....	23
FIGURA 7 - CÓDIGO EM JAVASCRIPT. ....	24
FIGURA 8 - INSTALAÇÃO APACHE CORDOVA ATRÉS DO NODE.JS. ....	25
FIGURA 9 - COMANDOS DO CORDOVA DE CRIAÇÃO DE PROJETO E INCLUSÃO DA PLATAFORMA ANDROID. .....	26
FIGURA 10 - PASTAS DO PROJETO CRIADO PELO APACHE CORDOVA. ....	26
FIGURA 11 - A) TELA DE LOGIN DO GiC MASTER B) TELA DE LOGIN DO GiC PLAYER. ....	30
FIGURA 12 - A) TELA DE BEM VINDO GiC MASTER B) TELA BEM VINDO GiC PLAYER. ....	31
FIGURA 13 - A) TELA DE ERRO DE LOGIN GiC MASTER B) TELA DE ERRO DE LOGIN GiC PLAYER. ....	32
FIGURA 14 - A) TELA DE CADASTRO DE USUÁRIO GiC MASTER B) TELA DE CADASTRO DE USUÁRIO GiC PLAYER. ....	32
FIGURA 15 - TELA DE FUNCIONALIDADES DA GiC MASTER. ....	33
FIGURA 16 - TELA DA FUNCIONALIDADE PLANEJAMENTO DA GiC MASTER. ....	34
FIGURA 17 - TELA DA OPÇÃO NOVO JOGO DA GiC MASTER. ....	34
FIGURA 18 - TELA DA OPÇÃO ADICIONAR JOGADORES DA GiC MASTER. ....	35
FIGURA 19 - TELA DE FUNCIONALIDADES DA GiC PLAYER. ....	35
FIGURA 20 - MODELO ENTIDADE RELACIONAMENTO .....	40
FIGURA 21 - DIAGRAMA CASO DE USO DA CRIAÇÃO DE UM JOGO. ....	40
FIGURA 22 - DIAGRAMA CASO DE USO DE INSCRIÇÃO DO PLAYER NO JOGO. ....	41

## LISTA DE TABELAS

TABELA 1 - COMPARAÇÃO ENTRE APLICATIVOS NATIVOS E WEB (XANTHOPOULOS & XINOGALOS, 2013).....	17
---	----

## LISTA DE SIGLAS E ABREVIATURAS

ABNT	Associação Brasileira de Normas Técnicas
CPU	<i>Central Processing Unit</i> – Unidade Central de Processamento
SO	Sistema Operacional
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
SQL	<i>Structured Query Language</i>
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicação)
BD	Banco de Dados
GPS	<i>Global Position System</i> (Sistema de Posição Global)
SDK	<i>Software Development Kit</i> (Kit de Desenvolvimento de Software)
REST	<i>Representational State Transfer</i> (Transferência de Estado Representacional)

## RESUMO

Este trabalho de conclusão de curso descreve as atividades decorrentes do período de estágio realizado no LAVI pelo discente Aaron Lesbão Dumont, supervisionado pela Prof<sup>ª</sup> Ms. Karen Figueiredo e orientado pelo Prof Ms. Jivago Medeiros.

O estágio teve por objetivo explorar a utilização de tecnologias híbridas no desenvolvimento de aplicativos para dispositivos móveis, como por exemplo, smartphones e tablets.

A atualização do HTML passou a possibilitar o desenvolvimento de aplicativos móveis híbridos, porém apenas com essa tecnologia não é possível acessar recursos nativos dos dispositivos, é necessário o auxílio de frameworks como por exemplo, o Appcelerator Titanium, Apache Cordova e Adobe PhoneGap, entre outros que permitem esse acesso para as diferentes plataformas (iOS, Android, Windows Phone, entre outras) e ao final do processo de desenvolvimento seja gerado um aplicativo específico para cada SO ao passo que a maior parte do código desenvolvido seja aproveitado em cada plataforma, diminuindo a necessidade de esforço específico para cada plataforma. No desenvolvimento desse trabalho de Estágio Supervisionado, além das tecnologias HTML5, CSS3 e JavaScript, foram utilizados o framework Apache Cordova, a biblioteca JavaScript jQuery Mobile e outras tecnologias adjacentes.

Os aplicativos desenvolvidos durante o estágio mostraram que a utilização de tecnologias híbridas, como por exemplo o Apache Cordova, para o desenvolvimento de aplicativos para dispositivos móveis, podem ser vistos como uma importante alternativa no que tange a produtividade para casos semelhantes ao que foi abordado neste trabalho.

## INTRODUÇÃO

Nos dias atuais, os dispositivos móveis deixaram de ser apenas um artigo de luxo e passaram a ser uma necessidade, pois eles fornecem recursos indispensáveis para a atualidade. As aplicações para esses dispositivos despertam um grande interesse científico e econômico. Muitas empresas estão levando seus sistemas para os dispositivos móveis, além de que o mercado de aplicativos para esses dispositivos vem crescendo cada vez mais, porém, tanto as empresas como os desenvolvedores, têm que escolher em qual plataforma das diferentes vão (Android, BlackBerryOS, FirefoxOS, iOS, Ubuntu Touch OS, Windows Phone, entre outros) investir, visto que se focar em mais de uma possivelmente aumentara os gastos com o desenvolvimento (XANTHOPOULOS & XINO GALOS, 2013; PALMIERI, SINGH & CICCHETTI, 2012; IDC, 2015).

A plataforma mais utilizada é o Android, não só por haver aparelhos que utilizam esse Sistema Operacional (SO) com um preço mais acessível como também por ser uma plataforma *open source*. Entre os SOs existentes, o Android é o mais utilizado no mundo em 2015, 82,2%, e em segundo lugar está o iOS, com 13,9%, mesmo assim, é interessante para o desenvolvedor que a sua aplicação possa estar em todas as plataformas possíveis, abrangendo um público muito maior. O empecilho para que isso ocorra é que para cada uma é necessário desenvolver um aplicativo na linguagem nativa do dispositivo, neste caso, para o Android, Java, e para o iOS, Objective-C (XANTHOPOULOS & XINO GALOS, 2013; PALMIERI, SINGH & CICCHETTI, 2012; IDC, 2015).

Uma forma de atenuar o conflito no desenvolvimento de aplicativos para dispositivos móveis é por meio da utilização de frameworks para o desenvolvimento de aplicativos híbridos. Estes aplicativos permitem que parte do código desenvolvido seja reaproveitado nas diferentes plataformas. Esse tipo de desenvolvimento utiliza as mesmas ferramentas para o desenvolvimento web, como HTML, CSS e JavaScript; um framework para desenvolvimento híbrido, como o Apache Cordova; e a linguagem nativa, como Java para Android e Objective-C para iOS, que proporcionará acesso às APIs dos dispositivos, permitindo, dessa forma, que o aplicativo desenvolvido acesse a Câmera ou o GPS do dispositivo

(XANTHOPOULOS & XINOGALOS, 2013; PALMIERI, SINGH & CICCETTI, 2012; IDC, 2015).

Esse relatório, requisito para a obtenção do título de Bacharel em Sistemas de Informação, descreverá os conceitos e práticas empregadas no estágio supervisionado no Instituto de Computação da UFMT. O estágio teve como objetivo explorar o framework Apache Cordova e demais tecnologias adjacentes no desenvolvimento de aplicativos híbridos, tendo como *case* um projeto de gamificação em educação.

Durante o estágio, foram desenvolvidos dois aplicativos híbridos para dispositivos móveis, focando em smartphones e *tablets* com o sistema operacional Android para auxiliar o projeto de gamificação de disciplinas realizado pela Professora Karen Figueiredo, supervisora do estágio.

O desenvolvimento foi realizado por módulos solicitados pela supervisora do projeto. Após a descrição de determinado módulo, ficava a cargo do estagiário e do professor orientador Jivago Medeiros como deveria ser feito o desenvolvimento, quais tabelas iriam ser criadas e a criação de webservices para solicitações ao servidor. O professor orientador também foi responsável por determinar quais frameworks, plugins e demais ferramentas que seriam utilizadas para o desenvolvimento da aplicação.

Foram realizadas reuniões semanais para demonstração do desenvolvimento da aplicação e *feedback* referentes aos módulos em questão.

Os demais capítulos desse documento são:

- Capítulo 1: argumentará sobre a revisão de literatura referente a teoria utilizada no estágio.
- Capítulo 2: descreve os materiais, técnicas e métodos empregados no estágio para alcançar seus objetivos.
- Capítulo 3: descrição dos resultados, utilizando-se de recursos como imagem, gráficos e tabelas para uma melhor demonstração.
- Capítulo 4: dificuldades encontradas ao longo do período de estágio.
- Capítulo 5: descreve as conclusões obtidas com o desenvolvimento do trabalho.

# 1. REVISÃO DE LITERATURA

## 1.1 Gamificação

O que é gamificação?

De um lado, empresários querendo que seus funcionários produzam mais e fiquem mais qualificados, ou professores querendo que seus alunos aprendam e produzam mais. De outro, profissionais desestimulados, sem reconhecimentos, ou alunos desinteressados no conteúdo, distraídos com smartphones e outras tecnologias. É esse o paradoxo que muitas empresas e escolas vivem no Brasil e em outras partes do mundo (VIANNA et al., 2013; FIGUEIREDO, 2015).

Uma possível alternativa para esses grupos expostos é a utilização da gamificação, onde conceitos de games são utilizados para despertar interesse, agilizar o aprendizado, tornar atividades tediosas e/ou repetitivas em dinâmicas e agradáveis, estimular o cooperativismo, em ambientes fora dos games. Mesmo estando fora do ambiente dos games, as pessoas têm a sensação de estarem jogando. Para comprovar isso, a pesquisa realizada pelas empresas SIOUX e Blend apresenta, do total de pessoas que participaram da pesquisa, 92,7% classificam jogar games como entretenimentos que lhes agradam (VIANNA et al., 2013).

Nos games há sempre um objetivo a ser alcançado e o jogador tem várias possibilidades de chegar ao resultado. Isso torna o jogo mais atrativo, estimulando e motivando os jogadores a continuarem. Além de que, segundo Vianna, jogar é uma função significativa que encerra determinado sentido, havendo sempre algo em jogo, um objetivo a ser alcançado, conferindo um sentido a ação.

Esses conceitos, estratégias, que os jogos possuem para deixar as pessoas interessadas e muitas vezes imersas nessa realidade, podem ser utilizados em contextos fora dos jogos para se obter resultados semelhantes (VIANNA et al., 2013).

### 1.1.1 Gamificação na Educação

Um dos primeiros ambientes que se tem contato com a gamificação é na escola. Ao se ganhar uma estrela ou pontos por entregar a tarefa ou responder uma pergunta corretamente, a pessoa tem a sensação de prazer por ter se saindo bem e,

provavelmente, estar à frente dos demais são conceitos de jogos sendo aplicados (FADEL & ULBRICHT, 2014).

Segundo a Pesquisa Game Brasil 2015, em uma comparação entre 2013 e 2015, o número de pessoas que jogam utilizando smartphones subiu de 71,3% para 82,8% ultrapassando a quantidade de jogadores de consoles (56,2%) e de computadores (71,3%). A combinação de gamificação de disciplinas e a utilização de smartphones para jogá-las traria não só comodidade, mas também o interesse de participar do game.

Com a gamificação os alunos, mesmo estando em um ambiente escolar, se sentem como se estivessem dentro de um jogo devido as mecânicas de *games* empregada, dessa maneira a aprendizagem é mediada pelo desafio, pelo prazer e entretenimento, afinal o espaço de aprendizagem é compreendido por um local que potencializa o desenvolvimento de habilidades cognitivas, sociais e motoras, sendo ele escolar ou não (BUSARELLO, ULBICHT & FADEL, 2014).

## **1.2 Desenvolvimento para Dispositivos Móveis**

Dispositivos móveis são aparelhos de computação pequenos, sem fio, como smartphones e tablets. O desenvolvimento de aplicativos para esses dispositivos vem crescendo assim como a popularização deles.

Aplicativos que são desenvolvidos para esses dispositivos podem ser classificados como: *Native App* (ou Aplicativo Nativo) e *Web App* (ou Aplicativo Web) (CHARLAND & LEROUX, 2011; HUY & VANTHANH, 2012; SMUTNÝ, 2012; XANTHOPOULOS & XINOGALOS, 2013).

O processo que origina os Aplicativos Nativos será denominado neste trabalho como Desenvolvimento Nativo, devido ao fato de serem para um ambiente/plataforma específica, para isso fazem uso de uma linguagem/SDK específico, como por exemplo, o Java para Android, Objective C para iOS e C++ para Windows Phone. Nesse tipo de desenvolvimento não é possível aproveitar o código para que seja utilizado no desenvolvimento em uma plataforma diferente, sendo necessário, nesse caso, reescrever todo o código para que seja possível utilizar o aplicativo em outro sistema operacional. Isto faz com que seja necessário conhecimento específico das plataformas para as quais deseja-se desenvolver e

disponibilizar o aplicativo desenvolvido, sendo esse um dos motivos pelos quais considera-se o desenvolvimento nativo uma atividade onerosa para equipes e empresas. Todavia, no desenvolvimento nativo é possível explorar ao máximo a totalidade dos recursos disponíveis no dispositivo, como sensores, recursos de chamadas, de interface, de gráficos e processamento, permitindo a construção de aplicativos com muitos recursos, capazes de processar vídeos e imagens de grande tamanho, além de recursos 3D como jogos (PHAN, GINDER-VOGEL, & GOVIER, 2014; SMUTNÝ, 2012; XANTHOPOULOS & XINO GALOS, 2013).

Os aplicativos web possuem uma peculiaridade, eles só são acessados através de navegadores (Mozilla FireFox, Google Chrome, Apple Safari, Dolphin, entre outros) que estejam previamente instalados nos dispositivos móveis, uma vez que o conteúdo dos *web apps* encontram-se em servidores remotos.

Parte desses aplicativos são desenvolvidos utilizando tecnologias amplamente difundidas, como por exemplo, HTML, CSS e JavaScript tornando o processo de desenvolvimento mais ágil e menos oneroso. O HTML encontra-se em sua 5ª versão, essa, além de oferecer novos elementos (ex: <section>, <head>, <nav>) que ocasionou um aumento da semântica no processo de escrita de código, propôs a padronização pelos navegadores de uma série de API's com potencialidade de incrementar o potencial dos aplicativos web, como por exemplo: *Web Storage*, *Indexed Database API*, *File API*, *Web SQL Database*, *Offline Web* e *Geolocation API*. Adendo, novas bibliotecas e atualizações de bibliotecas JavaScript (ex: jQuery Mobile, jQTouch, SenchaTouch, entre outras) foram desenvolvidas com o intuito de tornar os aplicativos web mais dinâmicos, fluidos e atraentes, melhorando a codificação pelos desenvolvedores e melhorando a experiência dos usuários. Por outro lado, o principal ponto fraco dos *web apps* continuam a ser o acesso limitado aos recursos de hardware e dados dos dispositivos (HUY & VANTHANH, 2012; XANTHOPOULOS & XINO GALOS, 2013).

	Aplicativo Nativo	Aplicativo Web
Acesso as APIs	Total	Muito Baixa
Tempo de desenvolvimento	Médio/Alto	Baixo
Curva de conhecimento da linguagem	Alta	Média
Reaproveitamento	Quase nenhum	Praticamente Total

**Tabela 1 - Comparação entre Aplicativos Nativos e Web (XANTHOPOULOS & XINOGALOS, 2013).**

Nesse contexto, surge ainda os *Hybrid apps* (ou aplicativos híbridos) originários do processo que se denomina nesse trabalho de Desenvolvimento Híbrido. O desenvolvimento híbrido tenta combinar as vantagens do desenvolvimento web e nativo. Há diferentes frameworks e ferramentas para desenvolvimento híbrido, a citar: Appcelerator Titanium, Adobe PhoneGap e Apache Cordova. Apesar de utilizarem diferentes abordagens, de um modo geral, essas ferramentas permitem que parte do desenvolvimento seja feito utilizando tecnologias voltadas para o desenvolvimento web, como por exemplo HTML5 e JavaScript, e ao final do processo o aplicativo seja instalado no dispositivo da mesma forma que um aplicativo nativo, diminuindo parte do conhecimento específico necessário para a programação em diferentes plataformas permitindo que ao menos uma parte do código desenvolvido seja aproveitado em diferentes plataformas (*cross-plataform*) (XANTHOPOULOS & XINOGALOS, 2013).

### 1.2.1 Desenvolvimento Híbrido

O desenvolvimento híbrido é a criação de um aplicativo que possa ser executado em qualquer um desses SOs, ou pelo menos em mais de um deles. Com isso o desenvolvedor diminui o tempo de programação, que seria dobrado para cada sistema em que o aplicativo pudesse ser executado.

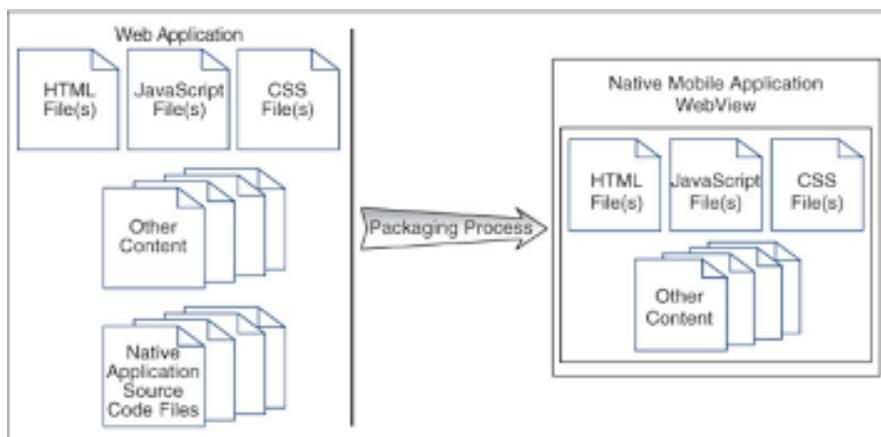
Nesse tipo de desenvolvimento, o código não é integralmente igual para todos os dispositivos. Em cada sistema, haverá algumas partes do código que deverão ser implementadas de acordo com a linguagem utilizada pelo sistema.

O acesso aos drivers do dispositivo deve ser desenvolvido na linguagem específica do sistema e é justamente nessa parte em que entram as bibliotecas que fazem a conexão da linguagem comum com a específica do sistema (XANTHOPOULOS & XINO GALOS, 2013).

### 1.2.2 Apache Cordova e Adobe PhoneGap

Iniciou-se o desenvolvimento de um Framework com uma proposta para o desenvolvimento de aplicativos *cross-plataform* baseado em *webviews* em 2008 no iPhoneDevCamp denominado Nitobi. Ele ganhou destaque entre os desenvolvedores de aplicativos para dispositivos móveis, porem seus criadores anunciaram que iriam doar o projeto para o Apache Software Foundation. Entretanto, antes que a doação ocorresse, a Adobe adquiriu a empresa Nitobi e denominou-o de PhoneGap, mesmo assim manteve a doação do projeto Nitobi. Na Apache Foundation, o projeto passou a se chamar Cordova e entre os mantedores do projeto está também a Adobe, que usa o Cordova como *core* do PhoneGap, este distribuído por ela (WARGO, 2015).

A Figura 1 ilustra o processo de empacotamento da aplicação que o Apache Cordova faz para enviar a aplicação para o dispositivo, ao contrário do que alguns desenvolvedores pensam, o Cordova não traduz, muito menos transforma, o código fonte em um código da linguagem nativa dos dispositivos, por exemplo Java para Android e Objective-C para iOS, o framework se utiliza da classe *WebView* para que através dela a aplicação seja executada (ANDROID, 2015; WARGO, 2015).

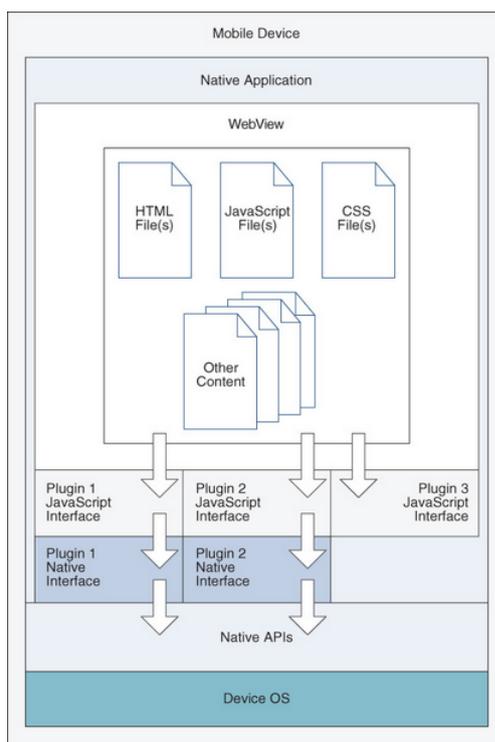


**Figura 1 - Processo de Empacotamento da Aplicação do Apache Cordova (Wargo, 2015).**

Na Figura 2 é demonstrado como a aplicação é executada após a instalação no dispositivo. Como se pode ver, a aplicação encontra-se dentro do *WebView*, será

por ela que a aplicação será executada, pode-se ver um exemplo nos aplicativos do Facebook e do Twitter onde ao clicar para abrir um link externo, nas versões do Android a partir da 5.0, em vez de abrir o navegador, ou uma lista de navegadores para selecionar um, é aberto diretamente pela aplicação (ANDROID, 2015; WARGO, 2015).

A aplicação utiliza-se de plugins para acessar as APIs nativas, esses plugins são codificados de duas maneiras: a primeira em JavaScript, as quais o front-end acessa, normalmente através do código em HTML; e a segunda em linguagem nativa do dispositivo, essas são chamadas pelas anteriores, em JavaScript, para executar determinadas ações normalmente relacionadas a chamadas no servidor ou acesso as APIs nativas do dispositivo, como por exemplo acessar a câmera ou o GPS (ANDROID, 2015; WARGO, 2015).



**Figura 2 - Apache Cordova Arquitetura da Aplicação (Wargo, 2015).**

### 1.2.3 Cordova Plugins

Plugins são arquivos contendo o código de funções e métodos, que permitem aplicações não nativas utilizarem recursos do dispositivo, como GPS, câmera, entre outros, que só estão disponíveis para aplicações nativas (CORDOVA, 2015).

Os plugins do cordova são desenvolvidos de tal forma que um mesmo pode ser utilizado em várias plataformas, dessa forma o desenvolvedor utiliza seus métodos na parte web da aplicação, através do JavaScript. A instalação é feita via linha de comando, como pode-se ver na Figura 3 (CORDOVA, 2015).

```
C:\DesenvolvimentoHibrido\GiCMaster>cordova plugin add cc.fovea.cordova.purchase
Fetching plugin "cc.fovea.cordova.purchase" via plugin registry
npm http GET http://cordova.iriscouch.com/registry/_design/app/_rewrite/cc.fovea.cordova.purchase/-/cc.fovea.cordova.purchase-3.11.0.tgz
npm http 200 http://cordova.iriscouch.com/registry/_design/app/_rewrite/cc.fovea.cordova.purchase/-/cc.fovea.cordova.purchase-3.11.0.tgz
Installing "cc.fovea.cordova.purchase" for android
Failed to install 'cc.fovea.cordova.purchase':Error: Variable(s) missing: BILLING_KEY
    at C:\Users\Aaron\AppData\Roaming\npm\node_modules\cordova\node_modules\cordova-lib\src\plugman\install.js:299:23
    at _fulfilled (C:\Users\Aaron\AppData\Roaming\npm\node_modules\cordova\node_modules\q\q.js:787:54)
    at self.promiseDispatch.done (C:\Users\Aaron\AppData\Roaming\npm\node_modules\cordova\node_modules\q\q.js:816:30)
    at Promise.promise.promiseDispatch (C:\Users\Aaron\AppData\Roaming\npm\node_modules\cordova\node_modules\q\q.js:749:13)
    at C:\Users\Aaron\AppData\Roaming\npm\node_modules\cordova\node_modules\q\q.js:557:44
    at flush (C:\Users\Aaron\AppData\Roaming\npm\node_modules\cordova\node_modules\q\q.js:108:17)
    at process._tickCallback (node.js:419:13)
```

**Figura 3 - Instalação de Plugin do Cordova.**

Como pode-se ver na Figura 2 os plugins possuem uma interface JavaScript que acessa os plugins nativos da plataforma para que seja possível acessar os recursos dos dispositivos.

Uma das vantagens do Apache Cordova em relação a outras tecnologias para desenvolvimento híbrido é a gama de plugins disponíveis, contando atualmente com mais de 750 plugins. Pode-se ver na Figura 4 uma lista de plugins do Cordova, disponíveis em seu próprio site, podendo também verificar as plataformas que é possível utilizar o plugin.

cordova-plugin-whitelist Cordova Whitelist Plugin	v1.0.0 by stevegill	Apache 80.810 downloads last month Last updated 140 days ago
cordova-plugin-device Cordova Device Plugin	v1.0.1 by stevegill	Apache 52.149 downloads last month Last updated 140 days ago
cordova-plugin-splashscreen Cordova Splashscreen Plugin	v2.1.0 by stevegill	Apache 50.215 downloads last month Last updated 140 days ago
cordova-plugin-statusbar Cordova StatusBar Plugin	v1.0.1 by stevegill	Apache 44.810 downloads last month Last updated 140 days ago
cordova-plugin-console Cordova Console Plugin	v1.0.1 by stevegill	Apache 43.431 downloads last month Last updated 140 days ago
ionic-plugin-keyboard Ionic Keyboard Plugin	v1.0.7 by ionic	Apache 38.188 downloads last month Last updated 56 days ago

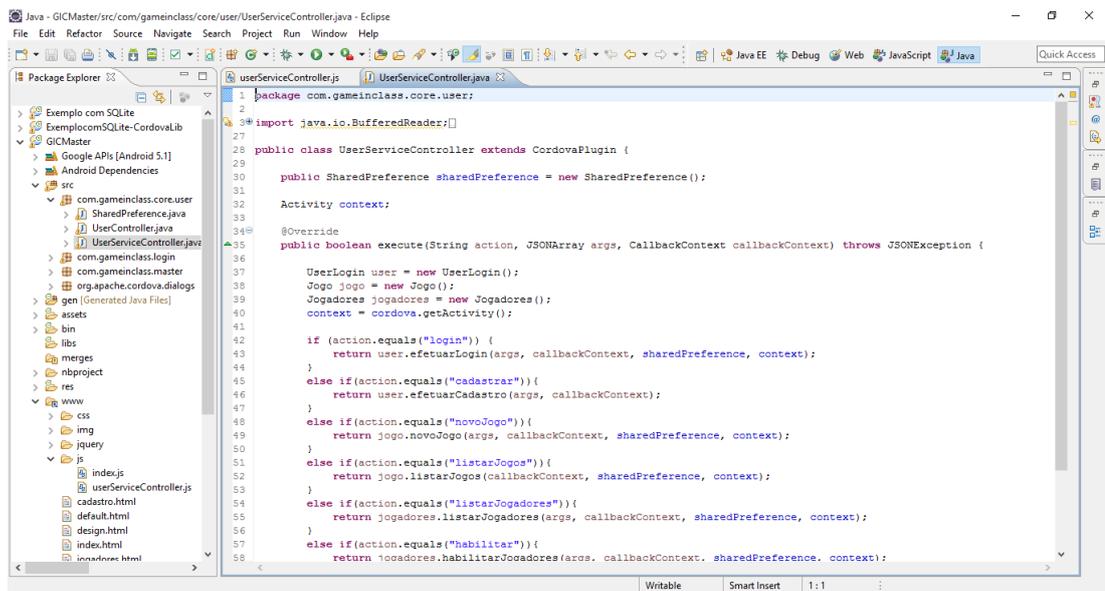
**Figura 4 - Lista de Plugins do Apache Cordova disponível em seu site.**

## 2. MATERIAIS, TÉCNICAS E MÉTODOS

Para a realização do projeto, foram utilizados diversos materiais, técnicas e ferramentas para se chegar ao resultado. Este capítulo tratara deles, descrevendo como eles funcionam e a maneira que foi utilizada no projeto.

### 2.1 Eclipse IDE

A Eclipse é a IDE mais utilizada no mundo, segundo eclipse.org. Nesta IDE, podem ser desenvolvidas aplicações em vários tipos de linguagens, uma das razões por ter sido utilizada no desenvolvimento do estágio aqui relatado, facilitando o desenvolvimento de aplicações híbridas que, neste caso, utilizou-se das linguagens HTML, CSS, JavaScript e JAVA (ECLIPSE, 2015).



**Figura 5 - Tela do Eclipse com projetos e classes abertas.**

Na IDE Eclipse, ilustrada pela Figura 5, pode ser instalado o ADT (Android Development Tools) que amplia os seus recursos facilitando a criação de novos projetos Android de forma mais rápida e com mais recursos.

## 2.2 Plataforma Android

O Android fornece um SDK para que os desenvolvedores possam criar aplicações, em Java, para essa plataforma. O SDK oferece vários recursos, como o emulador de um dispositivo móvel com o SO Android, porém o mais importante é que ele é responsável por compilar o código escrito em um APK, que é o pacote de instalação de aplicativos no Android, ele contém todo o conteúdo do aplicativo e a extensão do arquivo gerado é *.apk* (ANDROID, 2015).

Uma das razões de utilizarmos o Android como plataforma de estudo para a qual a aplicação será desenvolvida é que 82,2% dos smartphones o utilizam como sistema operacional (IDC, 2015).

Outra razão é que o Android é uma plataforma *open-source* facilitando tanto na criação do aplicativos por poder contar com ferramentas gratuitas e outras vantagens, como também para instalar o aplicativo no dispositivo para testa-lo.

## 2.3 HTML5, CSS3 e JavaScript

O HTML é uma linguagem empregada para desenvolvimento de sites, porém, na sua 5ª versão, a evolução foi tamanha que, hoje, é possível desenvolver aplicativos mobile usando essa linguagem.

Com o HTML5, a utilização de vídeos, animações e outros recursos, presentes em aplicativos, ficaram muito mais fácil, não só para o desenvolvedor, mas também na interação com o usuário.

```

46<div data-role="page" id="pgCadastro">
47  <div data-role="header">
48    <h1>Game In Class: Master</h1>
49  </div>
50  <div data-role="content">
51    <form action="" id="frmNovoJogo" name="frmNovoJogo" onsubmit="validarEmail(this);">
52      <div data-role="fieldcontain">
53        <label for="nome">Nome:</label>
54        <input type="text" name="nome" id="nome" required="required"/>
55        <label for="descricao">Descriç&cedil;&atilde;o:</label>
56        <textarea name="descricao" id="descricao" rows="5" cols="25"></textarea>
57        <label for="dtInicioInscricao">Data de In&ccedil;&otilde;eg:</label>
58        <input type="date" data-role="datebox" name="dtInicioInscricao" id="dtInicioInscricao"
59          required="required" />
60        <label for="dtFimInscricao">Data de Fim das In&ccedil;&otilde;eg:</label>
61        <input type="date" data-role="date" name="dtFimInscricao" id="dtFimInscricao" class="data"
62          required="required" />
63        <label for="dtInicioJogo">Data de In&iacute;&ccedil;&otilde;eg do Jogo:</label>
64        <input type="date" data-role="date" name="dtInicioJogo" id="dtInicioJogo" class="data"
65          required="required" />
66        <label for="dtFimJogo">Data de Fim do Jogo:</label>
67        <input type="date" data-role="date" name="dtFimJogo" id="dtFimJogo" class="data" required="required" />
68      </div>
69      <div>
70        <button id="btnNovoJogo" class="ui-shadow ui-btn ui-icon-plus ui-btn-icon-left ui-corner-all">CRIAR</button>
71      </div>
72    </form>
73  </div>
74  <div data-role="footer">
75    <h4>UFMT</h4>
76  </div>
77 </div>
78 <script type="text/javascript" src="cordova.js"></script>
79 <script type="text/javascript" charset="utf-8" src="jquery/jquery-1.11.2.min.js"></script>
80 <script type="text/javascript" charset="utf-8" src="jquery/jquery.mobile-1.4.5.js"></script>

```

Figura 6 - Código HTML.

Assim como o HTML, o CSS, uma linguagem de programação muito utilizada em conjunto com o HTML para definir a aparência ou o estilo da página web, vem evoluindo e hoje está em sua 3ª versão a qual dá mais autoridade em relação ao controle dos elementos da página para o designer.

Com o avanço dessas linguagens, para o conteúdo e outra para a forma, o desenvolvimento, que antes era apenas para página web, passou a ser também para aplicativos *mobile*. Essas novas versões dão suporte para diversos navegadores e plataformas, com isso um mesmo código pode ser acessado por um PC, Smartphone ou Tablets Android ou IOS, e apresentar as mesmas funcionalidades para todos. Além de que, para os dispositivos móveis, é possível criar aplicações utilizando essas linguagens e seus novos recursos.

Para que a interação seja mais fluida e dinâmica, é utilizado o JavaScript para executar os eventos ocorridos na página ou aplicação. Essa linguagem oferece

muitos recursos e um nível de resposta mais dinâmica, por ser uma linguagem *client-side*.

Há muitos frameworks que utilizam a linguagem JavaScript, como jQuery e jQuery Mobile, oferecendo mais recursos para o desenvolvedor e para o usuário. Esses frameworks deixam mais agradável a interação, além de auxiliar na facilidade do desenvolvimento contendo métodos e funções já criadas e prontas para a utilização.

## 2.4 jQuery e jQuery Mobile

jQuery é uma biblioteca JavaScript criada para otimizar a velocidade em que o programador desenvolve nas quais as aplicações rodam. Isso é possível por ela ser rica em recursos, evitando que o programador tenha que desenvolver uma função ou método do zero, além de que os recursos contidos nela foram testados e modificados para que trabalhem da melhor maneira possível, mesmo com todos esses recursos a biblioteca ainda mantém um tamanho pequeno evitando tornar a aplicação mais lenta.

Na Figura 7 é possível verificar a utilização do JavaScript nas linhas 20 e 30 para realizar a chamada do plugin *dialogs*, que é adicionado ao projeto e funcionará conforme citado na seção **1.2.3 Cordova Plugins**.

```

10<<script type="text/javascript" charset="utf-8">
11  document.addEventListener("deviceready", onDeviceReady, false);
12  function onDeviceReady() {
13    $("#frmNovoJogo").on("submit",function(e){
14      var form_data = jQuery('#frmNovoJogo').serializeArray();
15      e.preventDefault();
16      cordova.exec( successCreate, errorCreate, "UserServiceController", "novoJogo", form_data);
17    });
18  }
19  function successCreate(retorno){
20    navigator.notification.alert(
21      retorno.message,
22      function(){
23        $.mobile.changePage("planejamento.html");
24      },
25      "Pronto!",
26      "OK"
27    );
28  }
29  function errorCreate(retorno){
30    navigator.notification.alert(
31      retorno.message,
32      null,
33      "Lamento!",
34      "OK"
35    );
36  }
37
38  function validaFimInscricao(){
39    if($("#dtFimInscricao").text < $("#dtInicioInscricao").text){
40      alert("Data Fim menor que a data de inicio");
41    }
42  }
43 </script>
44<div data-cs="3" data-kind="parent" id="MsgCadastro">

```

Figura 7 - Código em JavaScript.

Seus recursos são tantos que o slogan é “Escrever menos, fazer mais” e o jQueryMobile veio para reforçar mais essa ideia. Com esse framework é possível escrever um site ou uma aplicação que trabalhe em qualquer plataforma, seja smartphones, tablets ou desktops de diversos sistemas operacionais.

Com isso passamos a ter um desenvolvimento híbrido, já que o mesmo código é utilizado em diversas plataformas com SOs diferentes.

## 2.5 Apache Cordova

No desenvolvimento híbrido, utiliza-se linguagens web, como HTML, CSS e JavaScript, para a codificação de uma parte da aplicação a ser desenvolvida, porém através dessas linguagens, não é possível acessar os componentes, como câmera, GPS, acelerômetro, dos dispositivos. Para isso, é necessária uma biblioteca que permita esse acesso.

O desenvolvedor precisa de um conjunto de APIs de dispositivos para que o aplicativo que ele está desenvolvendo possa acessar os componentes dos aparelhos que instalarão seu aplicativo. O Apache Cordova é um dos frameworks que possui esses conjuntos de APIs.

O Cordova permite o desenvolvimento multi-plataforma (*cross-plataform*), utilizando as linguagens web e diminuindo a necessidade da utilização de linguagens nativas.

### 2.5.1 Instalação do Apache Cordova, Criação e Execução dos projetos.

É recomendável pela documentação<sup>1</sup> oficial a instalação através da linha de comando pelo Node.js, utilizando o comando ilustrado pela Figura 8.

```
> npm install -g cordova
```

Figura 8 - Instalação Apache Cordova através do Node.js.

A Figura 9 demonstra o comando que é utilizado para a criação de um projeto utilizando o Cordova, após a expressão *cordova create*, vem o primeiro argumento (GiCMaster), que informa o nome da pasta onde ficará o projeto, o segundo argumento (com.gameinclass.master) é opcional, apenas quando o terceiro

---

<sup>1</sup> <https://cordova.apache.org/docs/en/latest/index.html>

argumento também não for descrito, ele serve para criar o projeto com um identificador de estilo de domínio reverso, e o terceiro argumento (“GiC Master”) proporciona um título par a aplicação, tanto o segundo como o terceiro argumentos podem ser alterados posteriormente no arquivo config.xml (APACHE, 2015).

```
C:\DesenvolvimentoHibrido>cordova create GiCMaster com.gameinclass.master "GiC Master"
Creating a new cordova project with name "GiC Master" and id "com.gameinclass.master" at locati
on "C:\DesenvolvimentoHibrido\GiCMaster"

C:\DesenvolvimentoHibrido>cd GiCMaster

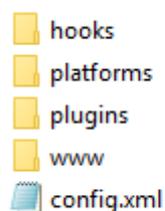
C:\DesenvolvimentoHibrido\GiCMaster>cordova platform add android
npm http GET https://registry.npmjs.org/cordova-android/3.7.1
npm http 200 https://registry.npmjs.org/cordova-android/3.7.1
npm http GET https://registry.npmjs.org/cordova-android/-/cordova-android-3.7.1.tgz
npm http 200 https://registry.npmjs.org/cordova-android/-/cordova-android-3.7.1.tgz
Adding android project...
Creating Cordova project for the Android platform:
  Path: platforms\android
  Package: com.gameinclass.master
  Name: GiC Master
  Android target: android-21
Copying template files...
Project successfully created.

C:\DesenvolvimentoHibrido\GiCMaster>cordova build
```

**Figura 9 - Comandos do Cordova de criação de projeto e inclusão da plataforma Android.**

Após esses comandos o projeto está pronto para ser desenvolvido, o exemplo dado acima foi apenas para a plataforma Android, porém para as demais plataformas basta alterar o comando *cordova platform add android* alterando o *android* para a plataforma desejada como por exemplo *ios*, para SO iOS da Apple, ou *wp8*, para SO Windows Phone 8.

A estrutura de pastas que o Cordova cria após o comando *cordova create* é a descrita na Figura 10. Na pasta *platform* ficara armazenado as informações referentes a plataforma adicionada ao projeto, na pasta *plugins* armazena-se os *plugins* adicionados ao projeto, como eles normalmente podem ser utilizados em mais de uma plataforma, basta adiciona-lo uma vez, e na pasta *www* ficam os códigos referentes HTML, CSS e JavaScript, sendo esses responsáveis por criar a parte visual da aplicação e acessar os *plugins*.



**Figura 10 - Pastas do projeto criado pelo Apache Cordova.**

## 2.6 Persistência de Dados

A persistência e recuperação de dados nos dispositivos móveis pode ser feita de várias maneiras, foi abordado duas maneiras para o projeto, persistência de dados no dispositivo e em servidores remotos via *web service*.

Toda plataforma tem por *default* uma ferramenta de persistir os dados no dispositivo, no caso do Android é através do SQLite. Dessa maneira, o aplicativo pode salvar no dispositivo determinados dados para evitar a necessidade do usuário de preenche-los constantemente, como por exemplo, os dados de acesso no aplicativo (ex: login e senha).

Porém, mesmo havendo a possibilidade de persistir dados no próprio dispositivo, algumas aplicações podem necessitar persistir e recuperar dados em servidores\bancos de dados remotos, nesses casos pode-se utilizar *web service*. O *web service* envia os dados para o servidor, o qual responde com um status informando se a persistência foi bem sucedida ou não. Para requisição dos dados que se encontram no servidor, o *web service* faz uma requisição par ao servidor, que retorna um status informando se foi possível ou não encontrar os dados requeridos e caso o status seja de confirmação, é enviado também o que foi solicitado.

## 2.7 Método de Desenvolvimento

O desenvolvimento foi separado por módulos, os quais eram passados para o estagiário em reuniões com a Prof<sup>a</sup> Karen onde as necessidades do projeto eram apresentadas. Posteriormente, o estagiário reunia-se com o orientador para traçarem estratégias de como ocorreria o desenvolvimento.

Persistência e recuperação de dados remotos: todos os dados trabalhados no aplicativo foram armazenados em um banco de dados MySQL 5.5 em um servidor remoto e para acessar e persistir esses dados foram utilizados *web services* REST<sup>2</sup> escritos na linguagem PHP.

Foi decidido que o layout utilizado seria o *springboard*, visto que é um layout simples, de fácil interação, além de funcionar bem em várias plataformas. Porém, esse só era definitivo mediante a aprovação da Prof<sup>a</sup>, tendo de ser refeito, mas

---

<sup>2</sup> REST é um protocolo de comunicação, baseado no protocolo de hipermídia HTTP. Porém ele não impõe restrições ao formato da mensagem, apenas no comportamento dos componentes envolvidos.

mantendo o mesmo estilo, para que atendesse de melhor as necessidades do projeto e se comportasse de uma forma aceitável em plataformas diferentes (NEIL, 2012).

### **3. RESULTADOS**

Durante a execução do estágio supervisionado foram desenvolvidos dois aplicativos híbridos (GiC Player e GiC Master) como forma de explorar as tecnologias descritas no Capítulo 2. Neste capítulo será apresentado e descrito cada um desses aplicativos ilustrando-os por meio de imagens obtidas da execução dos aplicativos em um aparelho Xperia Z3 e Android 5.1.

#### **3.1 GiC Master e GiC Player**

O estágio supervisionado tinha o intuito de desenvolver dois aplicativos para dispositivos móveis chamados GiC Master e GiC Player, sendo parte de um projeto de Gamificação na Educação da Profa. Karen Figueiredo. Esses aplicativos têm como objetivo proporcionar um ambiente de Gamificação na Educação, provendo elementos de jogos a serem utilizados por professores e alunos de disciplinas de curso de nível superior. Nas próximas duas subseções, 3.1.1 e 3.1.2 ser descrito em detalhes a proposta relacionada a esses dois aplicativos.

##### **3.1.1 GiC Master**

Será utilizado para efetuar o cadastro do professor no aplicativo, o qual utilizará como login para acessá-lo. O professor, após efetuar o login, poderá criar os jogos com os períodos de inscrição, início e fim do jogo, criar o ambiente do jogo e como ele ocorrerá. Também será possível autorizar a participação dos alunos, pois o aplicativo será utilizado em conjunto com alguma disciplina, sendo assim o aluno deve estar matriculado na mesma. O game poderá ter pontuação, o qual através do aplicativo o professor irá atribuir ao aluno (jogador).

##### **3.1.2 GiC Player**

Assim como o GiC Master, no GiC Player o aluno terá que fazer seu cadastro para poder fazer login no aplicativo. Após o login o aluno poderá ver a lista de jogos disponíveis para o período em que ele está logado. Após escolher o jogo, se inscrever e obter a autorização do criador do jogo (professor), ele participará do jogo, ou jogos, através do aplicativo. Haverá uma opção em que o aluno poderá ver a sua pontuação e também a sua posição no ranking do jogo.

### 3.2 Funcionalidades Desenvolvidas

Devido ao cronograma apertado, não foi possível desenvolver todas as funcionalidades dos aplicativos GiC Master e GiC Player descritas em detalhe nas duas sessões anteriores, todavia, foi desenvolvido um número considerável dessas funcionalidades permitiu-se explorar. Nesta subseção apresentaremos detalhadamente, explicando a utilização do Apache Cordova e demais tecnologias adjacentes no processo de desenvolvimento dos aplicativos híbrido, sendo que as funcionalidades comuns entre os dois aplicativos serão apresentadas em uma única subseção.

#### 3.2.1 Login

Na Figura 11 é demonstrado a tela inicial das aplicações, onde é feito o login para acessar as funcionalidades. A Figura 11 (a) mostra a tela do aplicativo voltado para os professores. Nele deve-se entrar o o CPF e a senha previamente cadastrada. Já na Figura 11 (b), é a tela inicial do sistema para os alunos (players) onde eles acessaram com o RGA e a senha préviamente cadastrada. Em ambos ha o botão para efetuar o cadastro nos respectivos aplicativos e o link para solicitar uma nova senha, caso o usuário tenha esquecido.

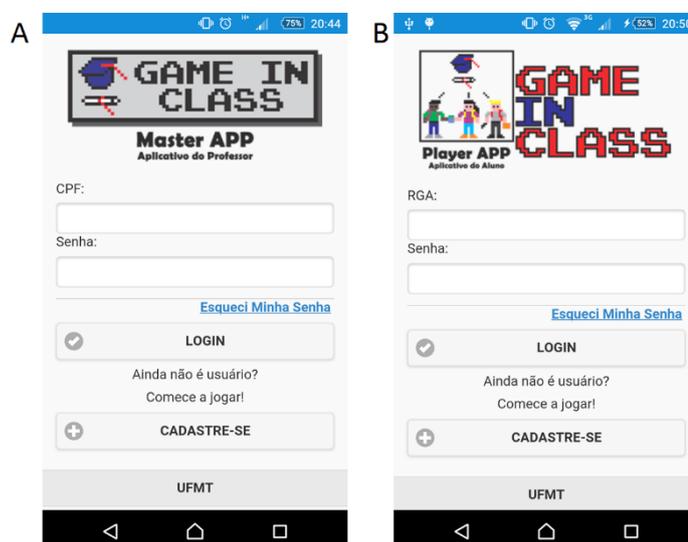
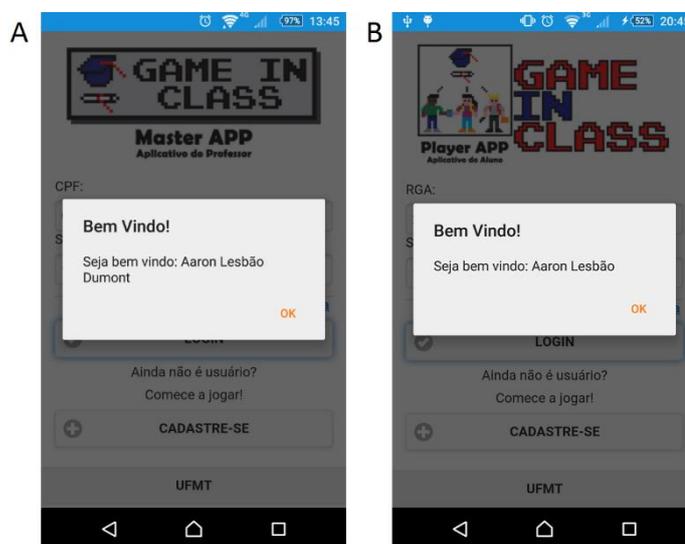


Figura 11 - a) Tela de login do GiC Master b) Tela de login do GiC Player.

Ao clicar no botão login, a aplicação fará uma requisição via *web service* para o servidor verificar se o usuário informado (login e senha) encontram-se registado no Banco de Dados. Após a verificação, o servidor irá enviar uma resposta

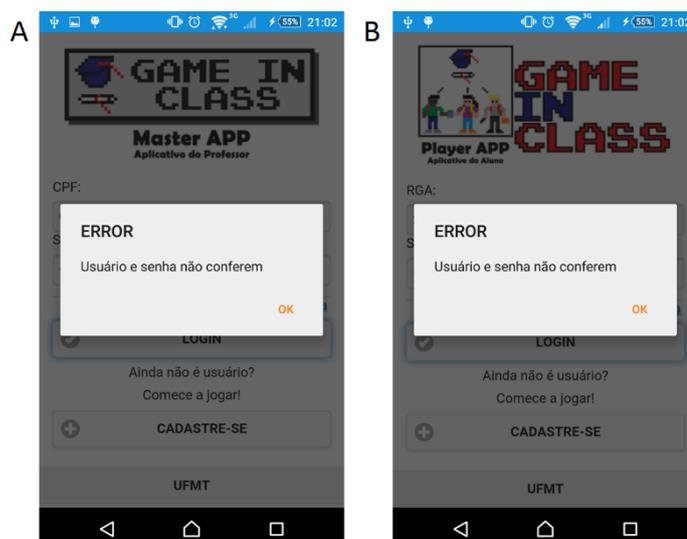
para a aplicação, sendo positiva caso tenha encontrado, ou negativa caso não o encontre. Esse processo só é possível graças ao Apache Cordova, que permite chamar um método na linguagem nativa do sistema através do JavaScript, dessa forma ele utiliza a API de acesso à internet para fazer a requisição.

Quando o retorno do servidor é positivo, o usuário consegue acessar o sistema, porém antes aparece a mensagem de Boas Vindas da aplicação, como pode ser visualizado na Figura 12 sendo que o A é referente a aplicação GiC Master e a B referente a aplicação GiC Player.



**Figura 12 - a) Tela de Bem Vindo GiC Master b) Tela Bem Vindo GiC Player.**

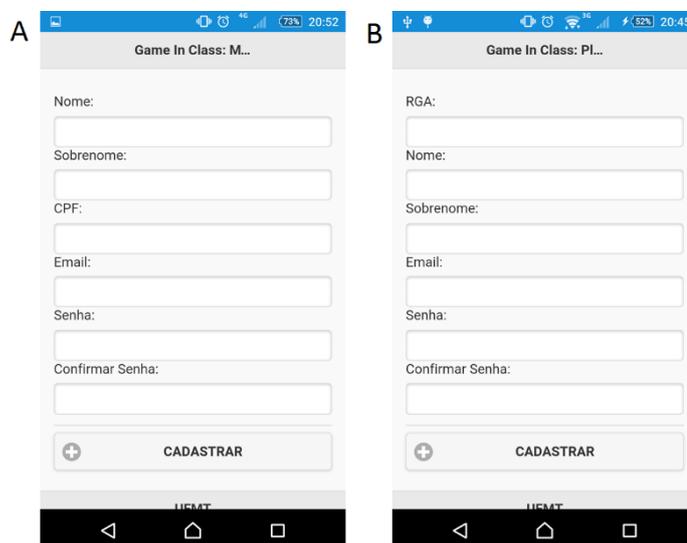
Um fluxo alternativo para o login é demonstrado na Figura 13, caso as informações estiverem incorretas o servidor enviara um retorno negativo, não permitindo que o usuário acesse a aplicação, nesse caso é exibido a mensagem demonstrada nas figuras abaixo.



**Figura 13 - a) Tela de erro de login GiC Master b) Tela de erro de login GiC Player.**

### 3.2.2 Cadastro

Caso o usuário, tanto o professor como o aluno, ainda não possuam cadastro, ele pode ser efetuado diretamente pelo aplicativo clicando no botão CADASTRAR respectivo de cada sistema. Após selecionar a opção de cadastrar-se, o sistema exibirá as telas demonstradas na Figura 14.



**Figura 14 - a) Tela de cadastro de usuário GiC Master b) Tela de cadastro de usuário GiC Player.**

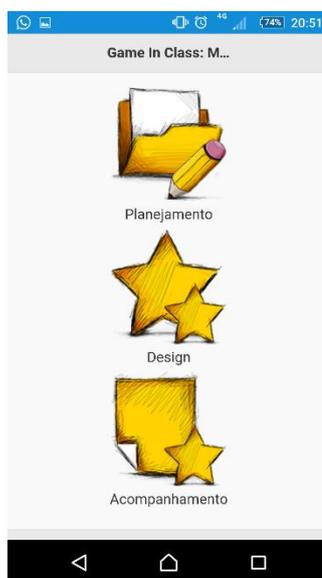
### 3.2.3 Funcionalidades GiC Master

As próximas imagens apresentam a tela de funcionalidades de cada aplicação. Será descrito primeiro as funcionalidades do aplicativo GicMaster, começando pela tela após o login, que demonstra as funcionalidades do usuário.

### 3.2.3.1 Planejamento

Na Figura 15 encontramos as funcionalidades referentes ao Mestre do jogo (professor) com elas ele poderá planejar como será o jogo, montar o design e acompanhar o andamento do jogo, verificando também a desenvoltura dos participantes (alunos) do jogo.

Pode-se observar na Figura 15 a utilização do *layout springboard* para o desenvolvimento dos menus, embora esse *layout* tenha sido utilizado em toda aplicação, fica nítido seu *design* nesta tela.



**Figura 15 - Tela de funcionalidades da GiC Master.**

Ao acessar o planejamento, aparecerá para o usuário as opções de criar novo jogo e de adicionar os jogadores nos jogos, essa função é necessária pois o professor deverá verificar se o aluno que está interessado em jogar um de seus jogos está matriculado na disciplina referente ao jogo. Pode-se verificar ambas as funcionalidades na Figura 16.



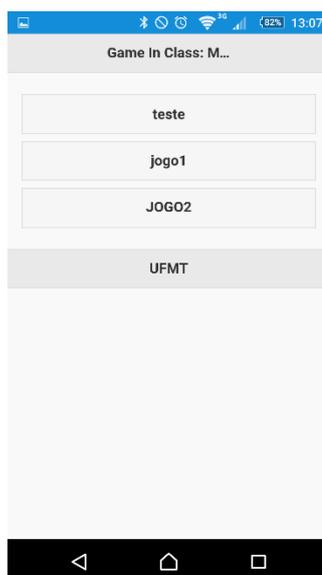
**Figura 16 - Tela da funcionalidade Planejamento da GiC Master.**

A Figura 17 demonstra os campos que devem ser preenchidos para se criar um novo jogo. Após o preenchimento de todos os campos, clica-se em “CRIAR” para criar o novo jogo.



**Figura 17 - Tela da opção Novo Jogo da GiC Master.**

Após a criação do jogo, o mesmo ficara disponível para os players se inscreverem, porém, o professor deverá aceitar ou não a inscrição do aluno em seu jogo. Para isso, o professor deverá selecionar um entre os seus jogos, como é apresentado na Figura 18.



**Figura 18 - Tela da opção Adicionar Jogadores da GiC Master.**

### 3.2.4 Funcionalidades GiC Player

Já na Figura 19 podemos observar as funcionalidades voltadas para o player (aluno) as quais são inscrever-se, na qual ele selecionara os jogos disponíveis para se inscrever, meus jogos, que mostra os jogos nos quais ele se inscreveu, e minhas estatísticas, onde o usuário pode acompanhar suas estatísticas em cada jogo.



**Figura 19 - Tela de funcionalidades da GiC Player.**

## 4. DIFICULDADES ENCONTRADAS

Para alcançar a eficiência e produtividade prometida pelos frameworks de desenvolvimento híbrido é necessário que haja estudo e aprendizado tanto do framework quanto também das tecnologias adjacentes. O Apache Cordova apresenta algumas peculiaridades em relação a construção e execução do aplicativo, muitas das quais trouxeram dificuldades e uma certa curva de aprendizagem até que conseguiu-se seguir em frente com o desenvolvimento dos aplicativos.

Conforme descrito, desenvolver plugins específicos, como o que foi descrito na seção 2.5.1 utilizando a linguagem JAVA para codificá-lo e o *framework* Apache Cordova para acessá-lo através de JavaScript, foi uma experiência nova para o desenvolvedor, havendo dificuldade em entender o funcionamento e a interação entre essas tecnologias.

Em relação as tecnologias como HTML, CSS e JavaScript, havia pouca experiência, em especial, houve algumas dificuldades em se trabalhar com bibliotecas JavaScript, no caso o jQuery e jQuery Mobile, esse por haver poucos exemplos, e em conseguir ajustar o *layout* da aplicação utilizando CSS, mesmo utilizando os arquivos CSS disponibilizados pelo jQuery Mobile.

Também havia pouca experiência em relação a utilização de *web service* e persistência de dados, visto que o estagiário não havia trabalhado com PHP antes, e esta é uma parte de suma importância para o projeto, afinal é a maneira em que vai ser persistir os dados no servidor.

## 5. CONCLUSÕES

No estágio supervisionado descrito nesse trabalho o aluno teve a oportunidade de desenvolver aplicações utilizando tecnologias de desenvolvimento híbrido propiciando a criação de dois aplicativos com um *case* real que aborda a gamificação na educação.

Durante a criação destes aplicativos verificou-se que o desenvolvimento híbrido de aplicativos é uma maneira eficiente, rápida e fácil de atender o mercado de aplicativos móveis. Já que esse mercado está sempre em expansão criando cada dia novos dispositivos e atualizando seus Sistemas Operacionais e as tecnologias utilizadas conseguiram se adaptar a estes fatores.

Apesar de se ter tido a oportunidade de explorar tecnologias híbridas (em especial o Apache Cordova) no desenvolvimento de aplicativos para dispositivos móveis, tendo instalado, criado dois projetos, adicionado plugins disponíveis no repositório do cordova, criado plugins próprios, trabalhado com webservices explorando alguns recursos do Android SDK, devido ao cronograma apertado não foi possível desenvolver todas as funcionalidades que seriam necessárias para a realização de testes de software e amadurecimento dos aplicativos para que pudéssemos realizar estudos de caso no *case* (gamificação na educação) abordado. Como trabalhos futuros, espera-se amadurecer as aplicações desenvolvidas e realizar estudos de caso que permitam avaliar em *cases* reais não somente o desenvolvimento híbrido, mas também a proposta de gamificação na educação apresentada.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

ANDROID, developers. The developer's guide, Disponível em <<http://developer.android.com/guide/index.html>>. Acesso em: 15 jul. 2015.

ANDROID, developers. Building Web Apps in WebView, Disponível em <<http://developer.android.com/intl/pt-br/guide/webapps/webview.html>>. Acesso em: 15 jul. 2015.

APACHE SOFTWARE FOUNDATION. The Command-Line Interface, Disponível em <<http://cordova.apache.org/docs/en/5.1.1/guide/cli/index.html>>. Acesso em: 27 out. 2015.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS; 1989. *Referências Bibliográficas*, NBR 6023. Rio de Janeiro.

CHARLAND, A., & LEROUX, B. (2011). mobile application Development : Web vs . native. Communications of the ACM, 54, 0–5. <http://doi.org/10.1145/1941487>

ECLIPSE. About the Eclipse Foundation, Disponível em <<https://eclipse.org/org/>>. Acesso em: 15 jul. 2015.

ECLIPSE, Getting Started. Disponível em <<https://eclipse.org/>>. Acesso em: 05 jul. 2015.

FADEL, Luciane Maria et al.; 2014. *Gameificação na educação*. 1ª Edição. São Paulo: Pimenta Cultural.

FIGUEIREDO, Karen; 2015. Proposta de Gamificação de Disciplinas em um Curso de Sistemas de Informação.

GOOGLE, INC.; Android Open Source Project. Disponível em <<http://source.android.com/>>. Acesso em: 20 jul. 2015.

HUY, N. P., & VANTHANH, D. (2012). Evaluation of mobile app paradigms. Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia - MoMM '12, 25. <http://doi.org/10.1145/2428955.2428968>

IDC RESEARCH, INC. Smartphone OS Market Share, 2015 Q2, Disponível em <<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>>. Acesso em: 30 out. 2015.

OGLIARI, Ricardo da Silva, BRITO, Robison Cris; 2014. *Android – Do Básico ao Avançado*. 1ª Edição. Rio de Janeiro: Editora Ciência Moderna.

NEIL, Theresa; 2012. *Padrões de Design para Aplicativos Móveis*. 1ª Edição. São Paulo: Novatec Editora Ltda.

PALMIERI, M., SINGH, I., & CICCHETTI, A. (2012). Comparison of Cross-Platform Mobile Development Tools of the 16th International Conference on Intelligence in Next Generation Network,

SMUTNY, P. (2012). Mobile development tools and cross-platform solutions. Proceedings of the 2012 13th International Carpathian Control Conference, ICC 2012, 653–656. <http://doi.org/10.1109/CarpathianCC.2012.6228727>

VIANNA, Ysmar et al.; 2013. *Gamification, Inc: como reinventar empresas a partir de jogos*. 1ª Edição. Rio de Janeiro : MJV Press.

WARGO, J. M. (2015). *Apache Cordova 4 Programming*. Person Education.

XANTHOPOULOS, S., & XINOGALOS, S. (2013). A Comparative Analysis of Cross-platform Development Approaches for Mobile Applications. Proceedings of the 6th Balkan Conference in Informatics, 213–220. <http://doi.org/10.1145/2490257.2490292>

## ANEXOS

### Anexo I – Modelo Entidade Relacionamento das Tabelas Criadas.

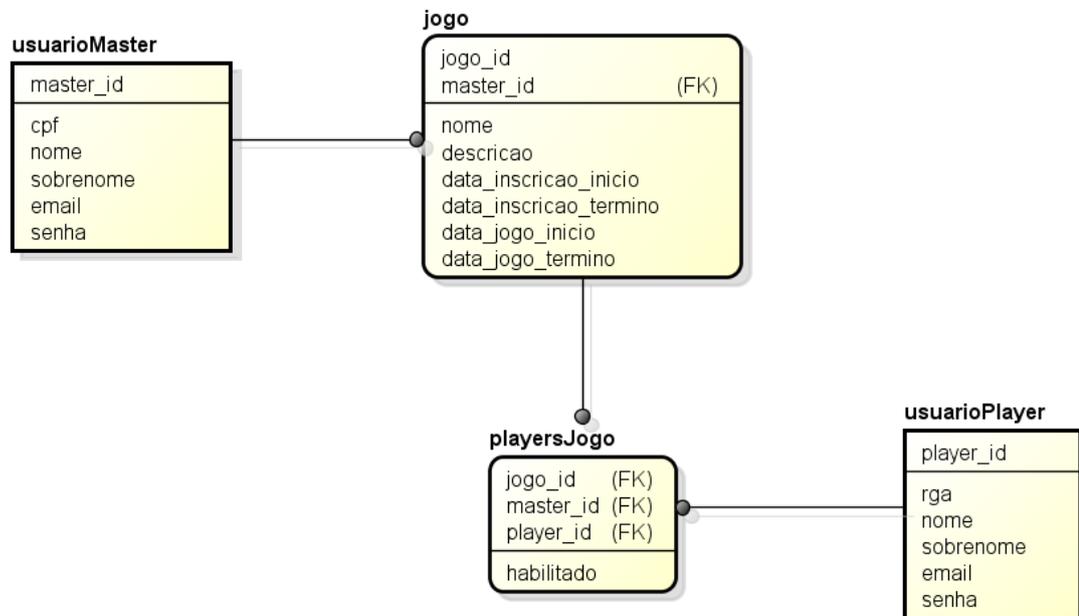


Figura 20 - Modelo Entidade Relacionamento

### Anexo II – Diagrama de Caso de Uso de criação de um jogo.

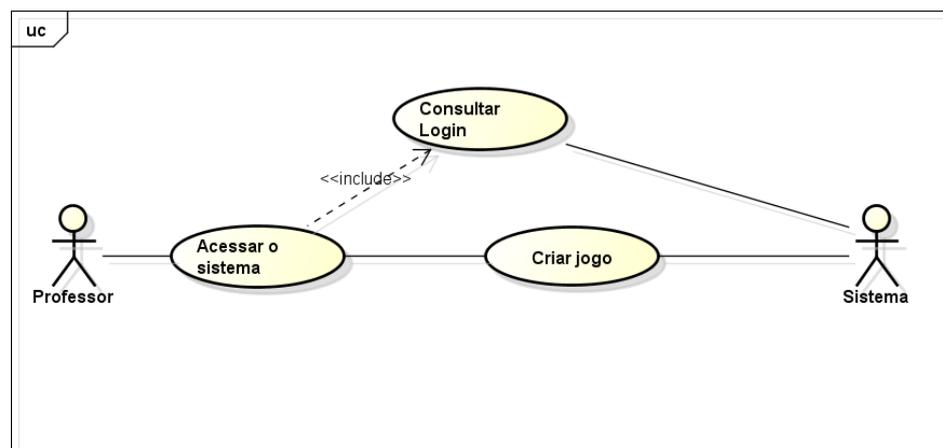
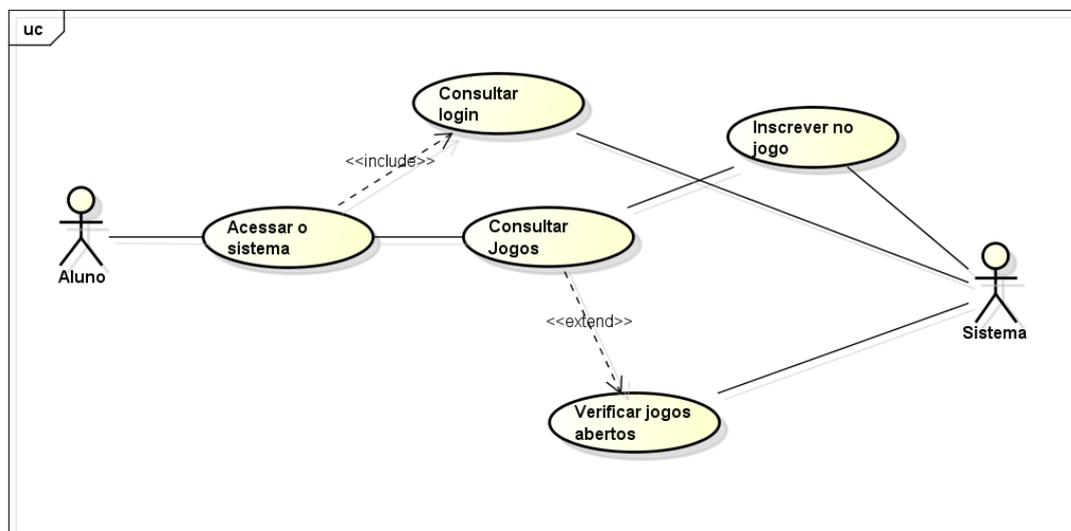


Figura 21 - Diagrama Caso de Uso da Criação de Um Jogo.

**Anexo III – Diagrama de Caso de Uso da Inscrição do Player no Jogos.****Figura 22 - Diagrama Caso de Uso de Inscrição do Player no Jogo.**

## APÊNDICES

### Apêndice I – Classe UserServiceController.java (Plugin) em JAVA.

```

package com.gameinclass.core.user;

import org.apache.cordova.CallbackContext;
import org.apache.cordova.CordovaPlugin;
import org.json.JSONArray;
import org.json.JSONException;

import com.gameinclass.login.Jogadores;
import com.gameinclass.login.Jogo;
import com.gameinclass.login.UserLogin;

import android.app.Activity;
import android.content.Context;
import android.util.Log;

public class UserServiceController extends CordovaPlugin {

    public SharedPreferences sharedPreferences = new
    SharedPreferences();

    Activity context;

    @Override
    public boolean execute(String action, JSONArray args,
    CallbackContext callbackContext) throws JSONException {

        UserLogin user = new UserLogin();
        Jogo jogo = new Jogo();
        Jogadores jogadores = new Jogadores();
        context = cordova.getActivity();

        if (action.equals("login")) {
            return user.efetuarLogin(args, callbackContext,
            sharedPreferences, context);
        }
        else if(action.equals("cadastrar")){
            return user.efetuarCadastro(args,
            callbackContext);
        }
        else if(action.equals("novoJogo")){
            return jogo.novoJogo(args, callbackContext,
            sharedPreferences, context);
        }
        else if(action.equals("listarJogos")){
            return jogo.listarJogos(callbackContext,
            sharedPreferences, context);
        }
        else if(action.equals("listarJogadores")){
            return jogadores.listarJogadores(args,
            callbackContext, sharedPreferences, context);
        }
        else if(action.equals("habilitar")){
            return jogadores.habilitarJogadores(args,
            callbackContext, sharedPreferences, context);
        }
    }
}

```

```

    }
    return false;
}
}

```

## Apêndice II – Classe index.html (Tela de Login) em HTML.

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="windows-1252" />
    <meta name="viewport" content="initial-scale=1, width=device-
width" />
    <link rel="stylesheet" type="text/css"
href="jquery/jquery.mobile-1.4.5.css">
    <title>Game In Class Master</title>
</head>
<body>
    <script type="text/javascript" charset="utf-8">

        document.addEventListener("deviceready", onDeviceReady,
false);

        function onDeviceReady() {
            $("form#frm-login-
pessoa").on("submit", function(e) {
                e.preventDefault();
                cordova.exec( successLogin, errorLogin,
"UserServiceController", "login",
[$("#cpf").val(), $("#senha").val()]);
            });
        }

        function errorLogin(retorno) {
            navigator.notification.alert(
                retorno.message, // message
                null,
                'ERROR', // title
                'OK' // buttonName
            );
        }

        function successLogin(retorno) {
            var confirmacao = confirm('Bem Vindo!\nSeja bem
vindo: ' + retorno.nome);
            if(confirmacao)
                $.mobile.changePage("menu.html");
        }

    </script>
    <div data-role="page" id="pgLogin">
        <div data-role="content">
            <div class="imagem" style="text-align: center;">
                
            </div>
            <!-- <div id="output">&nbsp;</div> -->
            <form id="frm-login-pessoa" name="frm-login-
pessoa">
                <div data-role="fieldcontain">

```

```

        <label for="cpf">CPF:</label>
        <input type="tel" name="cpf" id="cpf"
value="" required="required" />
        <label for="senha">Senha:</label>
        <input type="password" name="senha"
id="senha" value="" required="required" />
    </div>
    <div style="text-align: right;">
        <a id="esqueciSenha"
href="#esqueciSenha">Esqueci Minha Senha</a>
        <br/>
        <button type="submit" data-icon="check"
id="btnLogin">LOGIN</button>
    </div>
</form>
<div style="text-align: center;">
    <label>Ainda não é usuário?</label>
    <label>Comece a jogar!</label>
    <a href="cadastro.html" data-rel="external"
data-ajax="false" data-transition="slide"
class="ui-shadow ui-btn ui-icon-plus ui-btn-icon-
left ui-corner-all">CADASTRE-SE</a>
</div>
</div>
<div data-role="footer">
    <h4>UFMT</h4>
</div>
</div>
<script type="text/javascript" src="cordova.js"></script>
<script type="text/javascript" charset="utf-8"
src="jquery/jquery-1.11.2.min.js"></script>
<script type="text/javascript" charset="utf-8"
src="jquery/jquery.mobile-1.4.5.js"></script>
</body>
</html>

```

### Apêndice III – Código em JavaScript chamando o plugin UserServiceController, utilizando cordova.

```

function onDeviceReady() {
    $("#form#frm-login-pessoa").on("submit", function(e) {
        e.preventDefault();
        cordova.exec( successLogin, errorLogin,
"UserServiceController", "login", [$("#cpf").val(),
$("#senha").val()]);
    });
}

```

### Apêndice IV – Funções para o Callback do plugin UserServiceController em JavaScript utilizando o plugin dialog do Cordova.

```

function errorLogin(retorno) {
    navigator.notification.alert(
        retorno.message,
        null,
        'ERROR',
        'OK'
    );
}

```

```
    );  
}  
  
function successLogin(retorno) {  
    var confirmacao = confirm('Bem Vindo!\nSeja bem vindo: ' +  
retorno.nome);  
    if(confirmacao)  
        $.mobile.changePage("menu.html");  
}
```