

**ALLAN THOMAS OLIVEIRA FRANCO
FLÁVIO FRANCISCO DO NASCIMENTO**

APLICAÇÃO HÍBRIDA PARA A CPA USANDO IONIC

**UNIVERSIDADE DO VALE DO SAPUCAÍ
POUSO ALEGRE – MG
2016**

**ALLAN THOMAS OLIVEIRA FRANCO
FLÁVIO FRANCISCO DO NASCIMENTO**

APLICAÇÃO HÍBRIDA PARA A CPA USANDO IONIC

Pré-projeto de pesquisa apresentado à disciplina de TCC I do Curso de Sistemas de Informação como requisito parcial para obtenção de créditos.

Orientador: Prof. MSc. Márcio Emílio Cruz Vono de Azevedo.

**UNIVERSIDADE DO VALE DO SAPUCAÍ
POUSO ALEGRE – MG
2016**

SUMÁRIO

INTRODUÇÃO	3
2 JUSTIFICATIVA	4
3 OBJETIVOS	5
3.1 Objetivo Geral	5
3.2 Objetivos Específicos	5
4 QUADRO TEÓRICO	6
4.1 Ionic	6
4.1.1 Cordova	7
4.1.2 CSS	7
4.1.3 Javascript	7
4.2 AngularJS	8
4.3 HTML5	8
4.4 PHP	9
4.5 REST	10
4.6 MySQL	11
REFERÊNCIAS	12

INTRODUÇÃO

No mundo em que vivemos, com o desenvolvimento tecnológico e os *smartphones* em ascensão, as pessoas sempre olham seu dispositivo como recurso para resolver um problema ou como ele pode facilitar uma determinada tarefa, como uma transação bancária, uma pesquisa em um site de buscas ou simplesmente para acessar uma rede social. Esses dispositivos, que estão na mão de praticamente todos os universitários, podem ser usados como grandes aliados de uma instituição de ensino. Hoje em dia, o desenvolvimento e a utilização de *softwares* não se limita apenas aos *desktops* como era há alguns anos atrás, cada vez mais os sistemas seguem atrelados aos dispositivos móveis, com toda a facilidade e mobilidade que eles nos proporcionam.

Diante deste cenário, o projeto consiste em criar uma aplicação que possibilite à universidade coletar opiniões dos alunos em relação à sua infraestrutura, professores, instalações, entre outros, algo que já é feito semestralmente através do portal do aluno, mas que pode ser aprimorado com o uso de um aplicativo destinado especialmente para esse objetivo. Uma pesquisa que seria realizada através do portal da universidade, passa a ser feita por um aplicativo móvel e torna ainda mais atrativa para quem possa vir a respondê-la.

A tecnologia que será usada no desenvolvimento do projeto, que é chamada de híbrida, permite que um único aplicativo possua suporte para mais de uma plataforma *mobile*, como por exemplo o *Android* e o *iOS*, tudo isso a partir do *framework* Ionic. De acordo com Khanna e Harlington (2016), aplicativos híbridos são semelhantes aos aplicativos nativos, eles são desenvolvidos com um único código base e utilizados em mais de uma plataforma, além possuírem comunicação com o *hardware* e também serem instalados no dispositivo. Segundo IONIC (2016) o Ionic é um poderoso SDK¹ HTML5 que ajuda a construir aplicativos móveis usando tecnologias web como HTML², CSS³ e Javascript, que são voltados principalmente para a interface gráfica do aplicativo.

¹ *Software Development Kit*

² *HyperText Markup Language*

³ *Cascading Style Sheets*

2 JUSTIFICATIVA

A escolha por desenvolver um aplicativo usando Ionic e seus conceitos, deve-se pela sua excelente forma de construir um App¹ *mobile* multiplataforma, além das tecnologias usadas por ele que estão mais robustas e em constantes atualizações. A principal plataforma utilizada será o *Android* devido à sua grande popularidade e por ser um dos sistemas operacionais mais usados mundialmente.

A relevância do aplicativo para a universidade é grande, pois facilitará a pesquisa feita pela CPA, a qual muitos alunos deixam de responder por falta da praticidade. Um aplicativo destinado especialmente para ela, com fácil utilização, intuitivo e que emite alertas aos usuários quando a pesquisa estiver disponível. Pode ajudar a pesquisa a atingir seu público por mais de um meio, aumentando a quantidade de entrevistados, o que, no final, garante ainda mais informações.

O trabalho também auxiliará os alunos que se interessarem por aprender como funciona o desenvolvimento de um aplicativo híbrido, além de todas as tecnologias e *frameworks* que estão envolvidos no processo, que são atuais e muito requisitadas no mercado de trabalho.

¹ Abreviação para a palavra *Application*

3 OBJETIVOS

Descreveremos a seguir os objetivos pretendidos pela presente pesquisa.

3.1 Objetivo Geral

Desenvolver um aplicativo móvel multiplataforma a partir do *framework* Ionic, que deve possibilitar aos alunos da UNIVÁS¹ responderem ao questionário semestral que é disponibilizado pela CPA² afim de coletar informações e opiniões referentes à universidade.

3.2 Objetivos Específicos

Afim de atingir o objetivo geral da pesquisa, estão descritos a seguir os objetivos específicos:

- Efetuar o levantamento de requisitos do sistema, bem como das tecnologias que serão usadas no desenvolvimento;
- Planejar o desenvolvimento do projeto a partir dos requisitos coletados;
- Desenvolver o aplicativo que permita aos alunos responderem o questionário da CPA.

Conforme esses objetivos, espera-se demonstrar de maneira simples e efetiva o funcionamento do Ionic e seu método de desenvolvimento híbrido, tirando o maior proveito possível com um software que auxiliará a pesquisa que é realizada pela CPA.

¹ Universidade do Vale do Sapucaí

² Comissão Própria de Avaliação

4 QUADRO TEÓRICO

Para que se possa desenvolver uma aplicação são necessárias algumas tecnologias como linguagens de programação e *frameworks*, falaremos no capítulo a seguir quais são eles e suas principais características.

4.1 Ionic

Segundo IONIC (2016) o *framework* foi construído por Ben Sperry, Adam Bradley e Max Lynch em Drifty, uma empresa de *software* independente e que possui diversos produtos reconhecidos no mercado. O Ionic foi desenvolvido para possibilitar que aplicativos *mobile* fossem desenvolvidos em sua base com HTML5. Mesmo com conhecimentos não muito aprofundados em tecnologias web, é possível criar um aplicativo usando Ionic, que é um grande aglomerado de tecnologias que trabalham juntas para fazer do desenvolvimento web um caminho estreito para o mundo *mobile*.

Existem três principais modos de construção de um aplicativo móvel, são eles: aplicativos nativos, sites móveis e aplicativos híbridos, cada um deles apresenta diversas vantagens e desvantagens, o Ionic se encaixa na terceira opção. Wilken (2015) afirma que o desenvolvimento de aplicativos móveis se tornou um requisito inicial para qualquer desenvolvedor. O Ionic facilita esse desenvolvimento, ele faz com que um aplicativo móvel híbrido seja como um aplicativo nativo, através de recursos próprios que interpretam o código por partes, nas quais a interface gráfica é controlada por uma espécie de navegador isolado chamado de *WebView*, e também um aplicativo nativo que controla o *hardware*. Além do AngularJS, o Ionic possui o Cordova, que funciona carregando todo o código da aplicação web para apresentá-lo ao usuário, além de realizar a comunicação com o dispositivo.

O uso de AngularJS atualmente é exigido pelo Ionic, isso a fim de potencializar seu desenvolvimento. Além de AngularJS, o Ionic também usa Javascript e possui CSS nativo.

4.1.1 Cordova

Segundo Apache (2016) o Apache Cordova é um *framework* de código aberto para desenvolvimento móvel, com ele os aplicativos são direcionados para cada plataforma específica, contando com ligações API compatíveis para realizar o acesso ao *hardware* de cada dispositivo. Sem os recursos do Cordova o Ionic também pode ser executado, mas seu resultado só é visível através de um navegador, o Cordova é necessário para que a aplicação seja executada nos dispositivos.

4.1.2 CSS

Segundo Silva (2011), CSS é a abreviação para o termo em inglês *Cascading Style Sheet*, traduzido para o português como folhas de estilo em cascata. A linguagem CSS tem por finalidade estilizar uma estrutura HTML para a apresentação de elementos. Como por exemplo: cores de fontes, tamanhos de texto, bordas arredondadas entre outros. De acordo com Lie e Bos (2005) os primeiros vestígios de CSS são do ano de 1994, quando surgiu a necessidade de melhorar a aparência de uma página web tendo como *layout* um jornal, naquela época algumas coisas ainda eram impossíveis de se fazer, coisas que hoje em dia com o CSS3, encontramos facilmente em qualquer página web.

Sendo assim, entendemos que o CSS faz todo o trabalho de estilizar e deixar páginas web com uma interface amigável e melhor apresentada do que somente quando se tem HTML puro.

4.1.3 Javascript

Segundo Silva (2010), o Javascript foi criado pela Netscape em conjunto com a Sun Microsystems com o objetivo de adicionar interatividade a páginas web que por padrão são estáticas. Com HTML puro não é possível processar dados ou enviá-los ao servidor, o Javascript foi desenvolvido para rodar no lado do cliente (*client-side*) assim, apenas um navegador é necessário para fazermos funcionar um código escrito em Javascript.

De acordo com Flanagan (2006), o Javascript é interpretado ao invés de compilado, por isso muitas vezes é considerado uma linguagem de *script* ao invés de linguagem de programa-

ção. Apesar de levar parte do nome, o Javascript não é uma versão simplificada do Java, que é uma linguagem desenvolvida também pela Sun Microsystems.

4.2 AngularJS

Segundo AngularJS (2016), o AngularJS é um *framework* para desenvolvimento de aplicações web, foi criado em 2009 por Misko Hevery e Adams Abrons, porém adotado posteriormente pelo Google.

Branas (2014) diz que devido à sua forma de escrita simples e direta, reutilizável e que possibilita uma aplicação sustentável, a escrita do código com Angular ao seu final elimina uma enorme quantidade de código desnecessário, assim uma equipe de desenvolvimento mantém seu foco no trabalho que é realmente importante.

De acordo com Green e Seshadri (2014), com a evolução das tecnologias web, as aplicações foram se tornando maiores e mais robustas, isso acarretou em um aumento da complexidade para os seus administradores. Desenvolver utilizando Javascript/JQuery não estava gerando desempenho suficiente e a manutenção de código à longo prazo estava prejudicada. O AngularJS surgiu para atender essas necessidades que surgiram, com ele muitas tecnologias acabaram por ser dispensáveis ou pouco utilizadas.

O AngularJS utiliza como modelo de arquitetura o padrão MVC, que divide o sistema em três partes distintas e modulares: o modelo (*model*), a visão (*view*) e o controlador (*controller*). O modelo corresponde aos dados, visão é a interface gráfica do aplicativo com o usuário e o controlador corresponde à lógica de negócios. Cada uma dessas partes separa o código em grupos, o que proporciona vantagens como o dimensionamento e a organização de tarefas, na qual cada uma faz somente o que é designado, além disso o código se torna reutilizável e de fácil manutenção.

4.3 HTML5

Segundo Eis e Ferreira (2012), o HTML é um dos três pilares da web, uma linguagem de marcação para publicação de conteúdo (texto, audio, imagem etc) para web. Seu conceito de Hipertexto é uma forma de organizar o conteúdo de maneira não linear, as marcações que referenciam os elementos (*tags*), também os identificam para o CSS e o Javascript tratá-los.

O HTML que usamos hoje em dia, na versão 5, é uma evolução da versão criada por Tim Berners-Lee. O HTML5 chegou ao seu estágio final a partir de um grupo chamado WHATWG, formado por desenvolvedores de empresas como Apple, Mozilla e Opera, que estavam insatisfeitos com o caminho que a web estava tomando. A versão 5 trouxe para o HTML uma série de melhorias que o tornou muito mais potente, como a manipulação de características de elementos de forma que a aplicação continue funcional e leve, descartando a necessidade de scripts enormes. A manipulação de formulários é outro recurso que foi implementado no HTML5, que reduz drasticamente o trabalho no tratamento de informações que são enviadas para o servidor.

No HTML5 a semântica se faz mais presente, com menor quantidade de códigos, pouca necessidade de instalação de plugins e consequentemente ganho de performance.

4.4 PHP

O PHP¹ é uma linguagem *open source*, que em tradução livre, significa código aberto, ou seja, qualquer um pode contribuir para seus aprimoramentos. De acordo com PHP (2016), o que o difere do Javascript, por exemplo, é que o PHP é executado no lado do servidor, o que faz com que o código seja processado por um sistema dedicado especialmente para essa função, e não na máquina do usuário. Desta forma, o navegador recebe apenas os resultados do processamento, o que é indispensável quando se é necessário ter sigilo e segurança com as informações.

Segundo Niederauer (2004), o PHP é uma das linguagens de programação mais conhecidas e utilizadas mundialmente. Sua função, é principalmente trazer interações e conteúdos dinâmicos para páginas web, em que apenas o HTML não é suficiente. Um site de notícias, por exemplo, só possui atualizações em tempo real por causa do PHP ou uma linguagem semelhante, com a função de recuperar informações de um banco de dados e transmiti-las para o usuário.

Para que seja possível a comunicação entre uma aplicação móvel e um servidor de banco de dados através do PHP, é necessário um serviço web realizando esse intermédio. Para essa função será utilizado o *Slim Framework*, segundo Framework (2016), o *Slim Framework* é uma ferramenta que possibilita a criação de APIs² REST³ de maneira sucinta e poderosa através do

¹ *Personal Home Page*

² *Application Programming Interface*

³ *Representational State Transfer*

protocolo HTTP⁴.

4.5 REST

Segundo Saudate (2014), o REST criado por Roy Fielding, nada mais é que um estilo de desenvolvimento de *web services*, em que a tecnologia trabalha com base no protocolo HTTP e seus respectivos métodos, realizando chamadas através de URL's⁵.

A URL deve ser única por recurso, ou seja, para cada função que se deseja ter, é necessário uma URL diferente, pois é ela quem define o procedimento que será feito e qual será a interação com o lado do cliente. Em conjunto com a URL, é necessário trabalhar com os métodos HTTP, pois são eles quem a API interpreta para saber o que deverá ser feito. Na versão corrente do HTTP, 1.1, estão disponíveis oito principais métodos, são eles:

- GET
- POST
- PUT
- DELETE
- OPTIONS
- HEAD
- TRACE
- CONNECT

REST trabalha com dois principais formatos de envio e retorno de informações, que são o XML⁶ e o JSON⁷. Ambos os formatos possuem a mesma proposta, porém, com algumas diferenças, principalmente na implementação. No projeto será usado o formato JSON, por conta de sua performance, menor complexidade, tamanho reduzido quando comparado com o XML, e também por ser muito usado atualmente.

⁴ *Hypertext Transfer Protocol*

⁵ *Uniform Resource Locator*

⁶ *eXtensible Markup Language*

⁷ *JavaScript Object Notation*

De acordo com Webber, Parastatidis e Robinson (2010), a web atual é em sua maioria, desenvolvida para pessoas, e a necessidade de várias formas acesso ou mais de uma plataforma para um único recurso ou aplicativo, faz a exigência da construção de um sistema distribuído. A facilidade para trabalhar com sistemas distribuídos é a grande vantagem de uso das REST API's, elas possibilitam a comunicação de dados entre domínios distintos, justamente a necessidade de uma aplicação *mobile*, diferente do que acontece nos websites por exemplo, em que ambos estão na mesma rede e se comunicam diretamente, sem a necessidade de um *web service* realizando um intermédio.

4.6 MySQL

Segundo Jobstraibizer (2010), o MySQL foi criado na Suécia por três desenvolvedores, Allan Larsson, David Amark e Michael Widenius, o sistema foi posteriormente adquirido pela Sun Microsystems, empresa que hoje pertence à Oracle.

Niederauer (2005) diz que o MySQL é um SGBD⁸ relacional que utiliza como padrão a linguagem SQL⁹, largamente utilizado para websites e aplicações web, principalmente quando se necessita de velocidade ao acessar um banco de dados muito grande. Sua alta popularidade está também relacionada à disponibilidade para diferentes sistemas operacionais como o Linux, Mac e Windows. Entre suas vantagens estão a performance, escalabilidade e confiabilidade, além de ser compatível com diversas linguagens de programação como o PHP, Java, Python e C++. O MySQL é um banco de dados excelente, tanto para pequenos websites como para grandes portais e aplicações web.

⁸ Sistema de Gerenciamento de Bancos de Dados

⁹ *Structured Query Language*

REFERÊNCIAS

- ANGULARJS. : **AngularJS Documentation**: what is angular? 2016. Disponível em: <<https://docs.angularjs.org/guide/introduction>>. Acesso em: 15 de Fevereiro de 2016.
- APACHE, S. F. : **Cordova Documentation**. 2016. Disponível em: <<https://cordova.apache.org/docs/en/6.0.0/guide/overview/>>. Acesso em: 15 de Fevereiro de 2016.
- BRANAS, R. : **AngularJS Essentials**. Birmingham: Packt Publishing Ltd, 2014.
- EIS, D.; FERREIRA, E. : **HTML5 e CSS3** com farinha e pimenta. São Paulo: Tableless, 2012.
- FLANAGAN, D. : **JavaScript**: the definitive guide. Sebastopol: O'Reilly Media, Inc, 2006.
- FRAMEWORK, S. : **Slim Framework Documentation**. 2016. Disponível em: <<http://www.slimframework.com/docs/>>. Acesso em: 16 de Março de 2016.
- GREEN, B.; SESHADRI, S. : **AngularJS**. São Paulo: Novatec Editora Ltda, 2014.
- IONIC. : **IONIC Documentation**. 2016. Disponível em: <<http://ionicframework.com/docs/overview/>>. Acesso em: 13 de Fevereiro de 2016.
- JOBSTRAIBIZER, F. : **Criação de bancos de dados com MySQL**. São Paulo: Universo dos Livros Editora Ltda, 2010.
- KHANNA, R.; HARLINGTON, M. : **Getting Started with Ionic**. Birmingham: Packt Publ., 2016.
- LIE, H. W.; BOS, B. : **Cascading Style Sheets**: designing for the web, portable documents. Boston: Addison-Wesley Professional, 2005.
- NIEDERAUER, J. : **Desenvolvendo websites com PHP**. São Paulo: Novatec Editora Ltda, 2004.
- NIEDERAUER, J. : **Integrando PHP 5 com MySQL**. São Paulo: Novatec Editora Ltda, 2005.
- PHP. : **PHP Documentation**: o que é o php? 2016. Disponível em: <http://php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 16 de Março de 2016.
- SAUDATE, A. : **REST**: construa api's inteligentes de maneira simples. São Paulo: Editora Casa do Código, 2014.
- SILVA, M. S. **JavaScript**: guia do programador. São Paulo: Novatec Editora Ltda, 2010.
- SILVA, M. S. : **CSS3**: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das css3. São Paulo: Novatec Editora Ltda, 2011.
- WEBBER, J.; PARASTATIDIS, S.; ROBINSON, I. : **REST in practice**: hypermedia and systems architecture. Sebastopol: O'Reilly Media, Inc., 2010.
- WILKEN, J. : **Ionic In Action**: hybrid mobile apps with ionic and angularjs. Shelter Island: Manning Publications, 2015.