

SISTEMA WEB E APLICATIVO MOBILE PARA ENCONTRAR E CLASSIFICAR PROFISSIONAIS LIBERAIS NA CIDADE DE CRICIÚMA

Diógenes Lüdtke Zilli¹

Daniel Basílio Marcelo²

Resumo: Um problema da região sul de Santa Catarina é a dificuldade para encontrar profissionais qualificados para prestação de serviços informais, como pedreiro, eletricitista e pintor por exemplo. Com a chegada da internet, várias soluções tecnológicas surgiram voltadas para esse novo meio de comunicação. Uma dessas soluções veio após a criação dos *smartphones*, onde um novo mercado foi criado para o consumo de aplicativos, que solucionam problemas do cotidiano da população. O problema apontado pode ser solucionado com a junção dessas duas tecnologias, um novo produto, chamado Profissional Indicado, que indique profissionais qualificados para diversos tipos de serviços que podem ser necessários para a população de uma região. Um sistema *web* e um sistema *mobile* que armazena informações dos profissionais da cidade de Criciúma, uma solução criada com tecnologias aprendidas durante o decorrer do curso de Engenharia da Computação, como *JavaEE* na disciplina de orientação a objetos, *Javascript*, *HTML* e *CSS* na disciplina de Tecnologias *Web*. Além de outras soluções como *AngularJS*, *Wildfly Swarm*, *Postgresql*, *Maven*, e o *Ionic* para a solução *mobile*.

Palavras-chave: Profissionais Liberais, Prestação de Serviços, Sistema, *Web*, *Mobile*.

1 INTRODUÇÃO

Uma pesquisa feita pelo Instituto Brasileiro de Ética Concorrencial (ETCO) e pelo Ibre/FGV mostram que o movimento da economia pelo mercado informal girou em torno de R\$957 Bilhões em 2015 no Brasil, o que é cerca de 16,2% do PIB do país [1]. Um mercado de quase R\$1 Trilhão que movimenta a economia de milhares de pessoas que não possuem “carteira assinada”. A profissionalização através de um meio de divulgação direto para os seus serviços na internet, que contribuiria para o aumento do faturamento, e assim, de investimento e legalização, aumentaria a arrecadação do governo nesse mercado inexplorado.

Na maioria das vezes, quando precisamos contratar algum tipo de serviço profissional desse mercado, hoje nós precisamos utilizar o método de recomendação

¹ Graduando em 2019/1. E-mail: dogezilli@gmail.com

² Prof. Daniel Basílio Marcelo. E-mail: daniel.marcelo@satc.edu.br

“boca a boca”. Geralmente vamos pedir indicação de um profissional que precisamos para conhecidos, familiares, amigos, ou colegas de trabalho, ou seja, o nosso alcance fica limitado a nossa rede de relacionamentos.

Não existe hoje um lugar específico na nossa cidade que classifica todos os profissionais disponíveis na nossa região, nem um histórico de avaliações dos serviços prestados por esses profissionais.

A ideia para o projeto surgiu desse pensamento, e de uma necessidade observada por empresários que possuem pequenas empresas e profissionais autônomos que prestam serviços como eletricista, pedreiro e pintura na cidade de Criciúma, onde eles não possuíam um canal de divulgação direto para que novos clientes pudessem encontrar seus dados de forma mais direta. Em uma pesquisa feita na região, entrando em contato com profissionais autônomos, foi verificado que não existe nenhuma ferramenta que realiza esse tipo de serviço focado, na internet.

Tendo em vista o problema em encontrar um local com dados de profissionais qualificados na cidade de Criciúma, foi criado o Profissional Indicado, uma solução *web*, um site, e uma solução *mobile*, um aplicativo, que armazena as informações desses profissionais e indica o mais qualificado para o usuário do sistema.

O sistema permite o cadastro de profissionais, para que assim os perfis fiquem disponíveis na plataforma *web* e no aplicativo *mobile*, e então o usuário terá a possibilidade de pesquisar um perfil através do nome do profissional, ou então pelo tipo de serviço desejado. Os usuários ainda poderão avaliar o perfil com comentários de como foi o atendimento e a execução do serviço daquele profissional, além de terem a possibilidade de dar uma nota de zero a cinco estrelas junto com a avaliação.

As aplicações *web*, que são sistemas informatizados que rodam através dos navegadores de computadores e *notebooks*, e aplicações *mobile*, que são aplicativos que rodam através dos *smartphones* e *tablets*, são evoluções de soluções tecnológicas utilizadas após a criação da internet. Hoje em dia vários produtos são comercializados através desses dois meios, o que gerou uma revolução no dia a dia das pessoas em como eles acessam alguns tipos de serviços e comércio.

Várias empresas foram criadas a partir dessas tecnologias, empresas que começaram suas vidas na *web* como o *Google*, que é o maior sistema de pesquisa da internet, e o *Facebook*, que é a maior rede social do mundo. Temos também empresas

que foram criadas na era dos aplicativos, como o *Uber*, que é a maior empresa de motoristas particulares do mundo, e o *Airbnb*, maior empresa de aluguel de imóveis, tudo isso através do alcance da palma da sua mão.

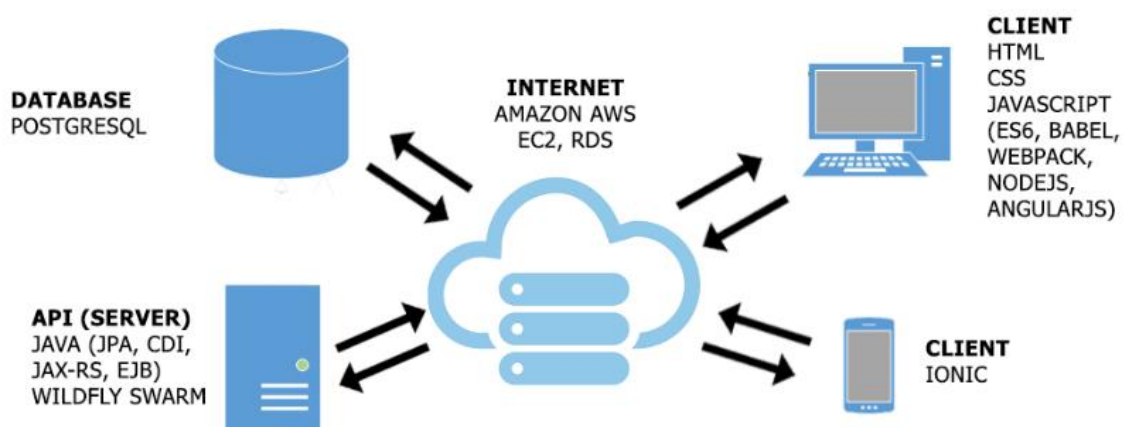
Um concorrente direto nesse mercado a nível nacional é o site GetNinjas, que possui uma proposta diferente. O cliente cadastra o pedido de um serviço que ele precisa, passando informações específicas da sua necessidade, e os profissionais recebem uma notificação para que eles enviem uma proposta de orçamento através do site para o cliente, para que ele escolha o profissional de sua preferência.

A abordagem do Profissional Indicado seria disponibilizar os perfis dos profissionais para que os usuários tenham acesso às informações, e selecionem o profissional através do histórico de avaliações e comentários do perfil, e assim entre em contato diretamente com o profissional, sem a interferência do site.

2 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento da solução, uma arquitetura de sistema teve que ser projetada para que atendesse os requisitos mínimos para desenvolver um produto que atendesse as necessidades do mercado. Essa arquitetura é como as diversas tecnologias necessárias se comunicam para que o sistema entregue o que é prometido. A arquitetura do sistema é apresentada na figura abaixo.

Figura 1: Arquitetura do sistema



O desenvolvimento dessa arquitetura foi desenvolvido com ferramentas próprias para o desenvolvimento de *software*, sendo elas a *IDE*, ou ambiente de desenvolvimento, *Intellij* para a criação do *back-end* em *Java*, sendo que o *back-end* é todo o código e ferramenta que roda no servidor, aonde vão ser feitas as requisições para que ele retorne os dados para o usuário. Foi utilizado o *Visual Studio Code* para a programação do *front-end* da aplicação, com *HTML*, *CSS*, *Javascript*, *AngularJS* e para a aplicação *mobile* em *Ionic*. O *front-end* é a parte visual do sistema, são os formulários e telas exibidos ao acessar as páginas do site pelo navegador. E para o gerenciamento do banco de dados foi utilizado o *PostgreSQL* em conjunto com a ferramenta *pgAdmin 4*.

2.2 JAVA – LINGUAGEM BACK-END

A linguagem de programação *Java* foi criada pela *Sun Microsystems* no início da década de 90. Mas que só veio aparecer na cena de desenvolvimento de *software* em 1995, onde rapidamente ganhou popularidade. *Java* também é o termo para descrever uma tecnologia composta pela *Java Virtual Machine* (JVM, ou Máquina Virtual do *Java* em tradução livre), bibliotecas embarcadas, compiladores, todos esses são elementos que estabelecem o *Java Development Kit* (JDK, ou Kit de Desenvolvimento *Java* em tradução livre) [2]. Ela foi utilizada no lado do servidor da aplicação. Ela que cria toda a lógica do sistema, todas as entidades mapeadas, necessárias para armazenar as informações do sistema.

2.3 MAVEN – GERENCIADOR DE DEPENDÊNCIAS

Para gerenciar as dependências de um projeto utilizando a linguagem de programação *Java* nós utilizamos o *Maven*. Ele aplica padrões para a infraestrutura do projeto, para criar uma compressão das bibliotecas e aumentar a produtividade do desenvolvedor, ao prover um arquivo que gerencia as dependências ou bibliotecas daquele projeto [3]. Foi usada em conjunto com o *Java* no lado servidor da aplicação. É com ela que podemos compilar todo o código em *Java*, e disponibilizar em um servidor *web*, nesse caso o *WildFly Swarm*.

2.4 WILDFLY SWARM – SERVIDOR DE APLICAÇÃO

Servidores web de aplicação são necessários para serviços que rodam na web. O *Wildfly Swarm* é um contêiner de um servidor web para publicar aplicações e serviços usando várias especificações do *JavaEE*. Ele trabalha usando o *Maven* como gerenciador de dependências e também para dar *build* no projeto, ou seja, compilar a aplicação para que ela rode na *web* [4]. Ele é basicamente o servidor de aplicação para que o projeto trabalhe na *web*, e com ele é possível gerenciar as conexões com banco de dados e autenticações. Através dele recebemos as requisições feitas pelo sistema *web* e pelo sistema *mobile*.

2.5 ARQUITETURA REST

Um conceito criado em uma tese de doutorado de Roy Fielding, publicado em junho de 2000, chamado de *Representational State Transfer* (REST, ou Estado de Transferência Representacional em tradução livre) é um estudo que apresenta o modelo de arquitetura *REST* como um padrão abstrato a ser utilizado no desenvolvimento de aplicações *web*, para guiar e redesenhar a definição de *HTTP* (*Hypertext Transfer Protocol*, ou Protocolo de Transferência Hipertexto em tradução livre) e as *URLs* (*Uniform Resource Identifier*, ou Identificador de Recurso Universal em tradução livre) [5]. Através dessa arquitetura é possível comunicar o *front-end* com o *back-end* através do protocolo *HTTP*, utilizando os métodos *GET*, *POST*, *PUT* e *DELETE*, enviando e recebendo requisições, passando essas informações através de *JSONs* (*JavaScript Object Notation*, ou Notação de Objetos JavaScript em tradução livre), um formato de objeto utilizado na *web* para enviar e receber dados. Essa arquitetura é implementada no *back-end* junto da linguagem *Java* com a biblioteca *JAX-RS*.

2.6 POSTGRESQL – BANCO DE DADOS

Um grupo de voluntários criou um projeto *open-source* (código-aberto) de um banco de dados objeto relacional chamado *PostgreSQL* [6]. Um banco de dados é uma aplicação que lhe permite armazenar e obter de volta dados com eficiência. O

modo que o banco de dados objeto relacional organiza seus dados é através de tabelas, que possuem colunas e linhas, sendo que as tabelas podem se relacionar entre si através dos ids de cada registro. O banco de dados *PostgreSQL* fica no lado do servidor da aplicação, e se comunica diretamente com a linguagem de *back-end Java* através do servidor de aplicação *web WildFly Swarm*.

2.7 ANGULARJS – FRAMEWORK FRONT-END

De acordo com Ralston[7], o *AngularJS* é um *framework front-end* que foi desenvolvido em 2009 por dois funcionários da *Google*, Misko Hevery e Adam Abroms, sendo escrito totalmente em *Javascript* e rodando apenas no lado cliente da aplicação, ou seja, no navegador em aplicações *web* ou em aplicativos *mobile*. Ele é conhecido por construir *Single Page Applications* (SPA, em tradução livre, Aplicações de uma única página), que são aplicações *web* que fornecem a experiência para o usuário de como se estivessem usando um sistema *desktop*.

Outra característica do *AngularJS* é o *two-way data-binding*, já que a arquitetura utilizada pelo *framework* é o *MVC* (*Model, View, Controller*, em tradução livre, Modelo, Visualização e Controle), os dados do *DOM*, ou do *Model* (*HTML* ou *HyperText Markup Language*, em tradução livre, Linguagem de Marcação de Hipertexto), são acessados diretamente pela *Controller* (*Javascript*) e vice-versa, sendo que a *View* (*CSS* ou *Cascading Style Sheets*, em tradução livre Folha de Estilos em Cascata), serve para modelar os componentes no *HTML*. Basicamente o *AngularJS* é usado no lado *front-end* da aplicação, ou seja, nos navegadores e no aplicativo. É ele que recebe os dados do *back-end* e exibe na sua tela do navegador ou aplicativo.

2.8 BOOTSTRAP – FRAMEWORK FRONT-END

Para facilitar o desenvolvimento do *front-end* do site *Twitter*, os desenvolvedores da empresa criaram um poderoso *framework*, o *Bootstrap*, que possui uma série de classes *CSS* e funções *Javascript* que facilitam o desenvolvimento *front-end*. Ele possui design responsivo que suporta todos os navegadores, seja no *notebook*, *tablet* ou *smartphones*.

O desenvolvedor não precisa se preocupar e trabalhar com o CSS para fazer seu site parecer mais bonito ou ter suporte a princípios de *design* responsivo, a não ser que seja necessário por alguma nova característica que a biblioteca não fornece e que o desenvolvedor queira implementar [8]. Ele foi utilizado para criar os formulários e estilizar as páginas *html* que rodam no navegador, deixando com uma aparência padronizada e mais bela.

2.9 JPA – API DE PERSISTÊNCIA DO JAVA

Uma solução para o mapeamento de objetos relacionais na linguagem *Java* é o *framework JPA* (*Java Persistence API*, ou API de Persistência do Java em tradução livre. É uma *API* (*Application Programming Interface*, ou Interface de Programação de Aplicações em tradução livre), e tudo que você precisa fazer é criar suas entidades de negócio e usar as anotações apropriadas do *JPA* para criar o mapeamento dos relacionamentos entre a entidade e o banco de dados. A *API* vai cuidar de todo o trabalho e fazer o mapeamento entre o *back-end* e o bando de dados funcionar perfeitamente [9]. Basicamente ele é utilizado como anotações dentro da linguagem *Java* para mapear o banco de dados nas entidades criadas no *back-end*.

2.10 CDI – INJEÇÃO DE DEPENDÊNCIA

A biblioteca de Injeção de Dependência do *Java* é o *CDI*. Ela se refere ao processo de fornecimento de uma dependência externa a um componente como por exemplo, um objeto de acesso a dados (*DAO*, *Data Access Object*) pode depender de uma conexão de banco de dados, o *CDI* faz isso para você, assim ele pode ajudar a tornar seu código arquitetonicamente mais puro [10]. Ele foi utilizado através de anotações na linguagem *Java* para injetar a conexão com o banco de dados dentro das entidades criadas no *back-end*.

2.11 JAX-RS – IMPLEMENTAÇÃO REST DO JAVA

Para implementar a arquitetura *REST* no *Java*, nós utilizamos o *JAX-RS*, que é a *API* do *JavaEE* para aplicações *web REST*, ela implementa o padrão de

arquitetura *REST* e prove de uma maneira simples a criação de serviços *REST*. Através de anotações dos métodos *HTTP*, *GET*, *POST*, *PUT*, *DELETE*, conseguimos disponibilizar *endpoints* do *back-end* para o *front-end*, para que a nossa aplicação consuma os serviços disponibilizados pela *API* [11]. Através de anotações da linguagem *Java*, foi mapeado os métodos *HTTP* que são chamados através de requisições feitas no *front-end*, pelo navegador ou pelo aplicativo.

2.12 *EJB – JAVA BEAN*

Um dos principais componentes da plataforma *Java Enterprise Edition* (*Java EE*) para desenvolvimento de sistemas corporativos é o *EJB*. Com ele é possível criar aplicações modulares, separar suas funcionalidades e as reutilizar de forma simples e objetiva. Ao utilizar *EJB* é possível construir aplicações distribuídas, totalmente seguras e com um eficaz tratamento de transações. [12]

Basicamente eles são executados em um container *EJB* no lado do servidor da aplicação. A principal característica do *EJB* é a rapidez em desenvolver soluções, pois, ele fornece um conjunto de componentes prontos, o que torna o desenvolvimento eficiente.

2.13 *IONIC – FRAMEWORK MOBILE*

O *framework* de desenvolvimento *mobile open-source Ionic* utiliza *HTML5*, *CSS* e *Javascript* para construir aplicações *mobile* que possuem uma experiência de usuário que se aproxima ao desenvolvimento nativo. Ele constrói suas aplicações através da tecnologia *SASS* (*Syntactically Awesome Style Sheets*, ou Folha de Estilos Sintática e Incrível em tradução livre), provendo componentes de interface para o usuário, facilitando para o desenvolvedor a criação de interfaces complexas para aplicações *mobile*, além de utilizar o *framework AngularJS* para aprimorar a experiência do usuário [13]. Ele foi utilizado para criar toda a aplicação *mobile*.

2.13 ECMASCRIPT6 E BABEL – JAVASCRIPT AVANÇADO NOS NAVEGADORES

Um comitê chamado *TC39* tem a responsabilidade de definir as novas características da linguagem *Javascript* que são lançadas todo ano. A versão *ECMAScript6*, que também é chamada de *ES6* ou *ES2015*, lançada em 2015, foi um marco para a linguagem *Javascript* por todas as novas características que foram implementadas. [14]

E para que os navegadores possam interpretar essas novas características da linguagem *Javascript*, é necessário a utilização de um transpilador chamado de *Babel*, que converte todo o código *ES6* para a versão *ES5* do *Javascript*, suportada pelos navegadores.

2.14 WEBPACK – BUNDLER DE MÓDULOS

O padrão das aplicações web hoje em dia são as *Single Page Applications* (*SPA*, ou Aplicações de uma única página em tradução livre), e uma ferramenta que facilita o desenvolvimento ao organizar o projeto em módulos. Ele é um *bundler* de módulos porque pega todas as dependências do projeto, e mimifica os arquivos *Javascript*, *HTML* e *CSS* em arquivos únicos, além de injetar apenas as dependências que realmente vão ser utilizadas no projeto. [15].

2.15 NODEJS – JAVASCRIPT NO LADO DO SERVIDOR

NodeJS é um poderoso ambiente de execução *Javascript* que roda encima da *engine V8* do *Google Chrome* [16]. Com isso é possível fazer rodar o *Javascript* no console do computador, e conseqüentemente ela também virou uma linguagem *server-side*, sendo que primeiramente ela foi criada para rodar apenas na web, ou seja, no lado cliente da aplicação. A partir dessa inovação, os desenvolvedores de *Javascript* ganharam uma imensa vantagem, pois com apenas uma linguagem eles podem desenvolver tanto o *front-end* quanto o *back-end* da aplicação. Foi a partir do *NodeJS* que ferramentas como o *Webpack* e o *Babel* tiveram a possibilidade de serem criadas.

2.16 AWS – EC2 E RDS

Através da computação em nuvem, a *Amazon* oferece o serviço de hospedar servidores de aplicação em seus servidores espalhados pelo mundo.

O *Amazon Elastic Compute Cloud (Amazon EC2)* fornece uma capacidade de computação escalável na nuvem da *Amazon Web Services (AWS)*. O uso do *Amazon EC2* elimina a necessidade de investir em *hardware* inicialmente, portanto, você pode desenvolver e implantar aplicativos com mais rapidez [17].

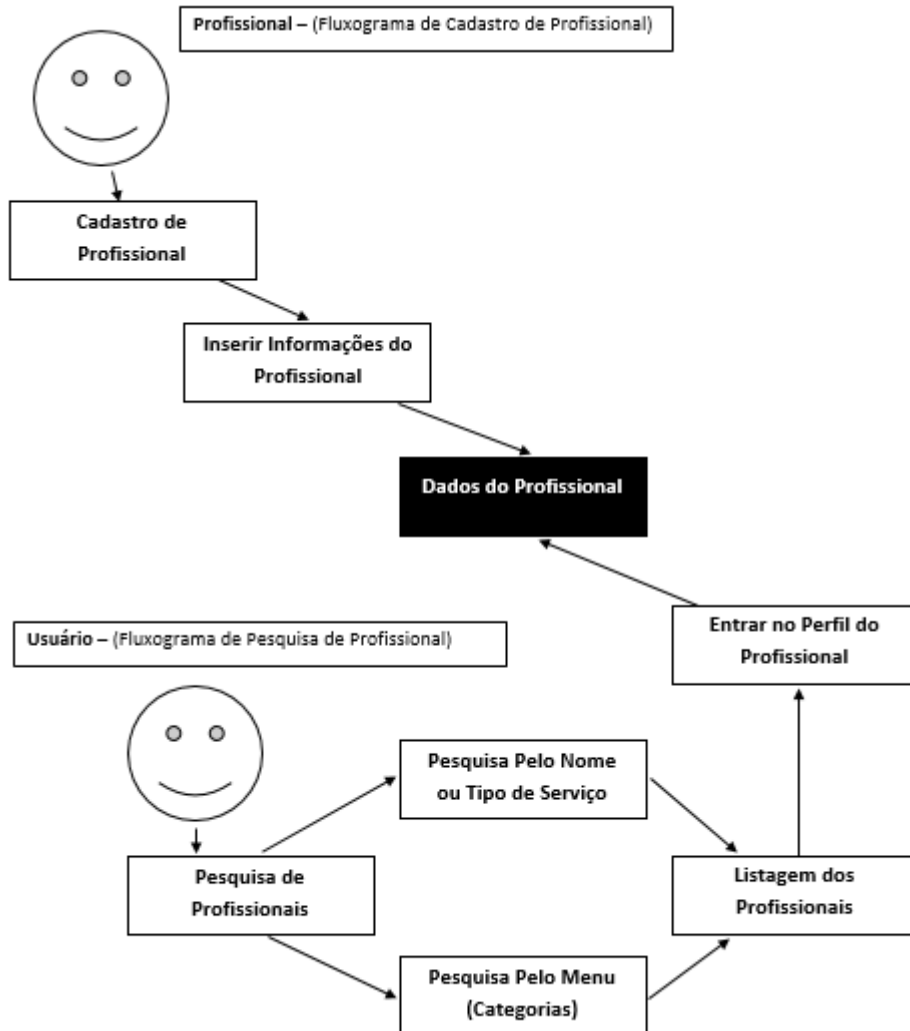
O *Amazon Relational Database Service (Amazon RDS)* facilita configurar, operar e escalar banco de dados relacionais na nuvem [18]. O serviço suporta o banco de dados *Postgres* que é utilizado nesse projeto, e ele administra a instância do banco no seu serviço em nuvem, criando a configuração do banco e disponibilizando para o administrador. Além desse serviço ele também cria *backups* dos bancos hospedados no serviço.

3 METODOLOGIA

O conjunto de tecnologias escolhidas para o desenvolvimento vem de encontro com o que está em alta no mercado na região. A linguagem Java para o back-end é por ser uma opção segura e robusta, surgiu na década de 90 e possui grande suporte da comunidade de desenvolvedores, além de ser uma escolha segura por ser uma das linguagens mais utilizadas mundialmente por desenvolvedores. No front-end o framework AngularJs é uma ferramenta que aumenta a produtividade do desenvolvedor, com ele foi possível criar o sistema web e os aplicativos mobile com o framework Ionic, que utiliza o AngularJs como linguagem de programação, economizando tempo de aprendizagem e desenvolvimento. Outro motivo pela escolha do Ionic é por ele ser uma ferramenta híbrida, em que com o mesmo código fonte consigo buildar um aplicativo para iOS e Android.

Abaixo temos uma imagem com o fluxograma de todo o funcionamento do sistema *web* do Profissional Indicado.

Figura 2: Fluxograma de processo sistema *web*.



Fonte: Autor (2018)

As tecnologias utilizadas para desenvolver a solução do problema encontrado foram cursadas durante a faculdade. Com o conjunto delas foi possível criar um produto que pudesse ser apresentado como uma solução viável para o problema definido.

A página *web* foi pensando em ter uma *homepage* principal em que os usuários pudessem entender de forma rápida e simples o propósito e o funcionamento do sistema. Existem dois botões principais na *homepage* que redirecionam para duas páginas diferentes do sistema. Um botão para cadastro de profissionais, e outro botão que encaminha o usuário para o menu de busca dos profissionais.

Figura 3: Homepage.



Fonte: Autor (2018)

Na página de cadastro, o perfil do profissional foi pensando para atender as características básicas do sistema, e as informações mais relevantes para o usuário, como o nome do profissional, a descrição dos serviços prestados, o endereço dele, contato com e-mail e telefone, e o tipo de serviço que ele presta, separados por três categorias principais, Automóveis, Profissionais e Residências. Cada categoria possui subcategorias para dar um filtro ainda mais qualificado para o usuário, para que ele tenha o filtro específico da atividade final que aquele profissional exerce em seu serviço.

Figura 4: Formulário de cadastro.

Cadastro do Profissional

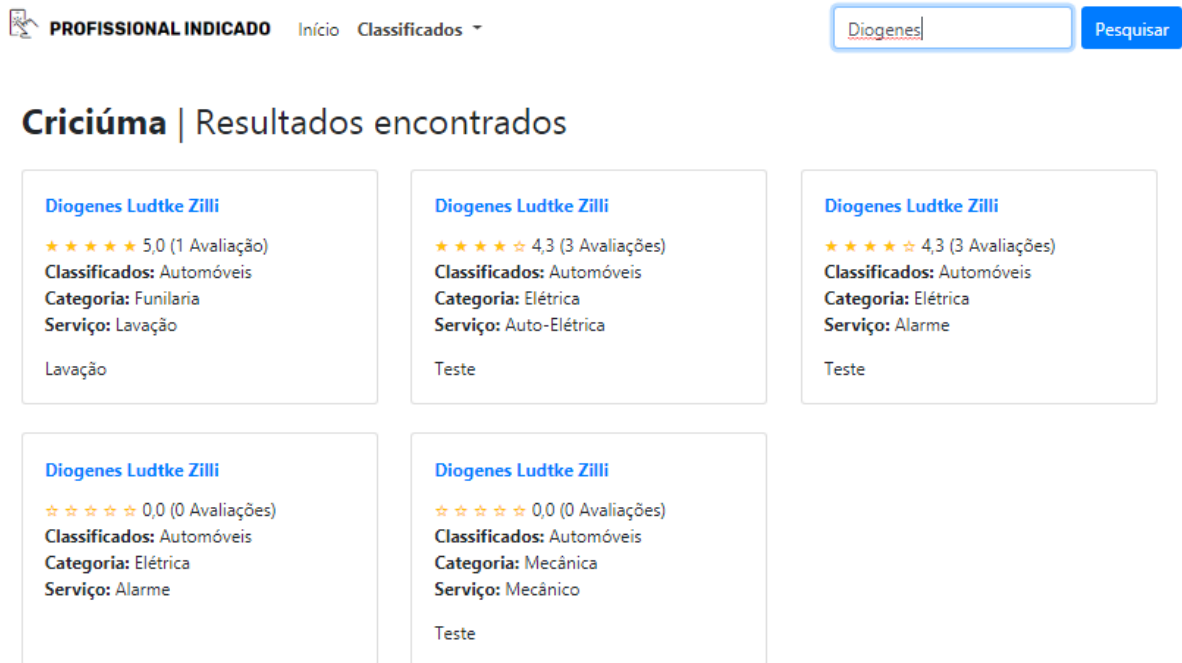
Usuário *		Senha *	
<input type="text"/>		<input type="text"/>	
E-mail	Site	Nome *	
<input type="text" value="nome@exemplo.com"/>	<input type="text"/>	<input type="text" value="Nome do profissional ou empresa"/>	
Telefone	Celular	Descrição do Serviço	
<input type="text" value="(xx) xxxxx-xxxx"/>	<input type="text" value="(xx) xxxxxx-xxxx"/>	<input type="text" value="Serviços que presta, horário de atendimento..."/>	
Cidade *		Classificados *	
<input type="text"/>		<input type="text"/>	
Rua	Número	Categoria *	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Bairro	Cep	Serviço *	
<input type="text"/>	<input type="text"/>	<input type="text"/>	

Fonte: Autor (2018)

Sendo que na categoria de Automóveis elas são divididas em três subcategorias, Elétrica, Funilaria e Mecânica, com mais 15 tipos de serviços divididos entre elas. Na categoria de Profissionais, são divididos em Aulas, Autônomos, Empresas, Engenheiros, Estética, Eventos e Saúde, com mais 60 tipos de serviços prestados divididos entre as categorias principais. E na parte de Residências existem três subcategorias, Animais, Reformas e Serviços Domésticos, tendo mais 19 tipos de serviços divididos entre elas.

Na página de busca dos profissionais existem dois métodos de busca principais. O filtro de pesquisa pelo nome ou tipo de serviço, no canto superior direito da tela, e os menus e sub menus das categorias daquela cidade. O aplicativo *mobile* foi pensado apenas para fazer a pesquisa desses profissionais, sendo assim ele é apenas um meio de consulta para ter as informações do profissional que ele necessita, filtrando primeiro pela cidade, aonde leva um sub menu das três categorias principais e assim por diante.

Figura 5: Pesquisa.



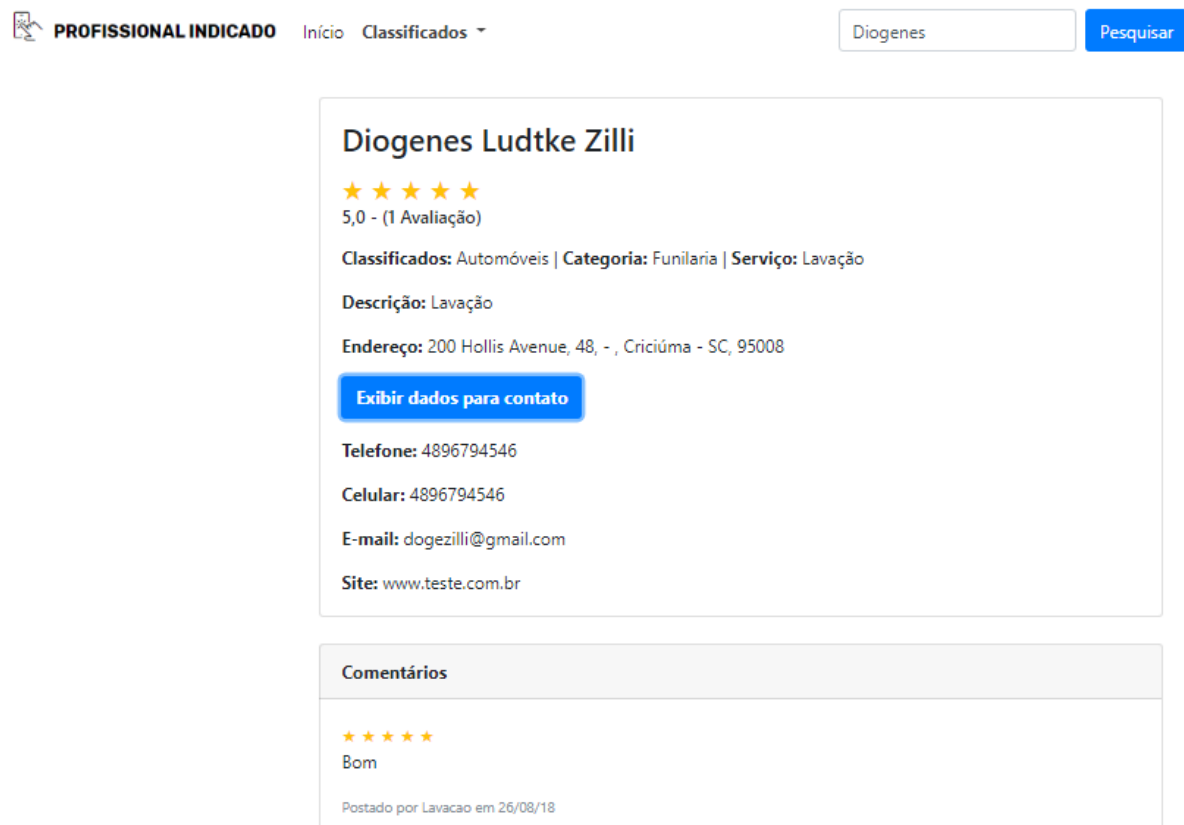
The screenshot shows a search interface with a search bar containing 'Diogenes' and a 'Pesquisar' button. Below the search bar, the text 'Criciúma | Resultados encontrados' is displayed. Five search results are shown in a grid. Each result card for 'Diogenes Ludtke Zilli' includes a star rating, the number of reviews, classification, category, and service type.

Star Rating	Reviews	Classification	Category	Service
★★★★★	5,0 (1 Avaliação)	Automóveis	Funilaria	Lavação
★★★★☆	4,3 (3 Avaliações)	Automóveis	Elétrica	Auto-Elétrica
★★★★☆	4,3 (3 Avaliações)	Automóveis	Elétrica	Alarme
☆☆☆☆☆	0,0 (0 Avaliações)	Automóveis	Elétrica	Alarme
☆☆☆☆☆	0,0 (0 Avaliações)	Automóveis	Mecânica	Mecânico

Fonte: Autor (2018)

Tanto no sistema *web* como pelo aplicativo o usuário pode conferir o perfil do profissional, verificar suas notas e avaliações e também pode dar uma nota e fazer uma avaliação para aquele profissional. Além de poder consultar os dados para contato daquele perfil, conforme imagem abaixo.

Figura 6: Perfil.



PROFISSIONAL INDICADO Início Classificados ▾

Diogenes

Diogenes Ludtke Zilli

★★★★★
5,0 - (1 Avaliação)

Classificados: Automóveis | Categoria: Funilaria | Serviço: Lavação

Descrição: Lavação

Endereço: 200 Hollis Avenue, 48, - , Criciúma - SC, 95008

Telefone: 4896794546
Celular: 4896794546
E-mail: dogezilli@gmail.com
Site: www.teste.com.br

Comentários

★★★★★
Bom

Postado por Lavacao em 26/08/18

Fonte: Autor (2018)

Na imagem podem observar o perfil do profissional, que contém ao topo o nome do mesmo, em seguida é exibido a média das notas de todas as suas avaliações, assim como a quantidade delas. Logo abaixo é exibido a informação sobre o tipo de serviço que aquele profissional presta, com um campo de descrição que o profissional pode preencher com informações adicionais como horário de atendimento e outros serviços específicos que ele presta. Temos então a seção que exibe as informações do endereço daquele profissional, com os dados da cidade, bairro, rua e CEP. Em seguida temos o botão Exibir dados para contato, que ao ser clicado exibe as informações para contato com aquele profissional, como telefone, celular, e-mail e site. E logo na última seção temos a listagem de todos os comentários daquele perfil,

onde é mostrado a opinião de cada cliente que postou uma avaliação, junto com a nota que ele deu aquele profissional.

3.1 METODOLOGIA DE PESQUISA DE USUÁRIOS

A busca pelos primeiros profissionais a serem cadastrados no sistema foi pelos meios de pesquisa convencionais, com a ferramenta de pesquisa *Google*, a recomendação de profissionais de pessoas conhecidas, e cartões de contato. Foi entrado em contato com esses profissionais e verificado se eles teriam interesse de disponibilizarem seus serviços de forma gratuita nesse novo serviço *web* e aplicativo que iria catalogar profissionais da cidade de Criciúma e ser um meio de avaliação para seus serviços prestados, além de ser uma ferramenta de divulgação e *marketing* para seus serviços, centralizando para os usuários daquela região informações importantes para a contratação de um profissional.

Na aplicação foi desenvolvido duas formas de geração de dados estatísticos. A primeira utilizando o *Google Analytics*, que é uma ferramenta do *Google* que monitora o tráfego de um site. Nesse caso ele monitora todos os acessos às categorias de serviços disponíveis no site. E outra forma de monitorar o acesso aos perfis dos profissionais, são quantas requisições teve o acesso ao perfil de determinado profissional, além da contagem de clicks em um botão dentro do perfil, para a exibição dos dados para contato daquele profissional.

Sendo assim, o profissional teria informações relevantes para o seu negócio, como o retorno de divulgação que a aplicação do Profissional Indicado está dando a ele, além das funcionalidades de avaliação feitas pelos usuários do sistema, o que gera um valor agregado para o profissional de quantos acessos seu perfil está recebendo, e como o seu perfil e serviço está sendo avaliado.

4 RESULTADOS

O resultado obtido foi muito gratificante e acima do esperado. Na parte técnica houve um aprendizado técnico e teórico de tecnologias em alta no mercado. A escolha das tecnologias mostrou algumas vantagens e desvantagens para algumas situações encontradas, como por exemplo ao escolher uma tecnologia consolidada

no mercado como o Java nos deu uma robustez e segurança no sistema, mas que em contrapartida pelo tamanho do projeto inicial, existe um consumo de memória grande para um serviço que poderia ser mais leve utilizando outra tecnologia.

Na parte de resultado de mercado, foram registrados no sistema 55 profissionais da cidade de Criciúma das mais diversas áreas, que se interessarão em criar seus perfis para que ficassem disponíveis para ser acessado através do site ou do aplicativo. O *feedback* recebido pelos profissionais é positivo, e mostra o potencial do projeto em divulgar e manter um sistema de avaliações organizado dos perfis dos profissionais e serviços divulgados.

O Engenheiro com mestrado, Jônata Biehl, que possui uma barbearia e foi um dos primeiros participantes do projeto disse o seguinte: *“Ter um aplicativo/site que reúna os prestadores de serviços categorizados em uma região como Criciúma, facilita o encontro destes pelos potenciais clientes regionais. Essa conexão permite que as negociações possam ser realizadas de forma muito prática, principalmente nos quesitos qualidade, já que o sistema pontua o prestador de serviço com melhor classificação, colocando-o entre os primeiros entre as buscas.”*

5 CONSIDERAÇÕES FINAIS

O projeto contribuiu muito a desenvolver na prática, a teoria aprendida no decorrer do curso. Foi um desafio criar uma aplicação *web* e *mobile* do começo ao fim, e entregar uma solução que já pode ser usada no mercado. O desenvolvimento de uma solução voltada para o mercado, foi algo de muito valioso na construção desse TCC. Pois algo que tem utilidade pública é algo que deve ser feito com o conhecimento adquirido durante esses anos de curso. A solução de catalogar profissionais liberais em um banco de dados, e disponibilizar essas informações através de um aplicativo se mostrou válida, pois o acesso a essas informações a novos moradores da cidade, ou até a pessoas que não sabem onde encontrar os melhores profissionais para a sua necessidade, se mostrou algo útil para o dia a dia de quem precisa de algum tipo de serviço. Para implementações futuras, pode ser elaborado um sistema de cadastro de solicitação de serviço por parte do usuário para determinada categoria de profissional, e os profissionais daquela categoria naquela região, podem assim receber uma notificação através do aplicativo com os dados do interessado em contratar o serviço.

REFERÊNCIAS

- [1] Globo. **“Economia informal cresce pela 1ª vez em mais de uma década, diz estudo”** Disponível em: < <http://g1.globo.com/economia/noticia/2016/06/economia-informal-cresce-pela-1-vez-em-mais-de-uma-decada-diz-estudo.html>> Acesso em: 10 setembro. 2018.
- [2] MILCZAREK, Tomasz; SAKOWICZ, Bartosz; NAPIERALSKI, Andrzej. **Job Recruitment System Based on Java Enterprise Edition 5 Platform**. Poland, 2008.
- [3] Maven. **“What is Maven?”** Disponível em: < <https://maven.apache.org/guides/getting-started/>>. Acesso em: 14 agosto. 2018.
- [4] DZone. **“Building Microservices With Wildfly Swarm”** Disponível em: < <https://dzone.com/articles/building-microservices-with-wildfly-swarm>> Acesso em: 14 agosto. 2018.
- [5] LI, Hongjun. **RESTful Web Service Frameworks in Java**. China, 2011.
- [6] SULTANA, Seema; DIXIT, Sunanda. **Indexes in PostgreSQL**. 2011.
- [7] RALSTON, Steve. **Introduction to AngularJS**. Disponível em: <<https://www.webdesignerdepot.com/2013/04/an-introduction-to-angularjs/>>. Acesso em: 14 agosto 2018.
- [8] BALASUBRAMANEE, Viknes; WIMALASENA, Chathuri; SINGH, Raminder; PIERCE, Marlon. **Twitter Bootstrap and AngularJS**. USA, 2013.
- [9] DZone. **“Java Persistence API Introduction (Part 1)”** Disponível em: < <https://dzone.com/articles/java-persistence-api-introduction-part-1>> Acesso em: 14 agosto. 2018.
- [10] DZone. **“CDI Dependency Injection - An Introductory Java EE Tutorial Part 1”** Disponível em: < <https://dzone.com/articles/cdi-di-p1>> Acesso em: 14 agosto. 2018.
- [11] ReadLearnCode. **“What is JAX-RS 2.0?”** Disponível em: < <https://readlearncode.com/java-ee/java-ee-what-is-jax-rs-2-0/>> Acesso em: 14 agosto. 2018.
- [12] DEBNATH, Manoj. **“What Are Enterprise Java Beans (EJB)?”** Disponível em: < <https://www.developer.com/java/ejb/what-are-enterprise-java-beans-ejb.html>> Acesso em: 12 outubro. 2018.
- [13] YANG, Yahui; ZHANG, Youbing; XIA, Pengtao; LI, Bo; REN, Zezhi. **Mobile Terminal Development Plan of Cross-platform Mobile Application Service Platform based on Ionic and Cordova**. China, 2017.
- [14] ZAKAS, Nicholas. **Understanding ECMAScript 6**. EUA, 2016.

[15] DIFFENDERFER, Aaron. “**The Wonderful Wide World of webpack: Unpacking Awesomeness**” Disponível em: < <https://keyholesoftware.com/2018/06/28/wide-world-of-webpack/>> Acesso em: 12 outubro. 2018.

[16] HELLER, Martin. “**What is Node.js? The JavaScript runtime explained**” Disponível em: < <https://www.infoworld.com/article/3210589/node-js/what-is-nodejs-javascript-runtime-explained.html>> Acesso em: 12 outubro. 2018.

[17] AWS. “**O que é o Amazon EC2?**” Disponível em: < https://docs.aws.amazon.com/pt_br/AWSEC2/latest/UserGuide/concepts.html> Acesso em: 12 outubro. 2018.

[18] AWS. “**Amazon Relational Database Service (RDS)**” Disponível em: < <https://aws.amazon.com/pt/rds/>> Acesso em: 12 outubro. 2018.

ABSTRACT

One problem in the southern region of Santa Catarina is the difficulty in finding qualified professionals to provide informal services, such as mason, electrician and painter for example. With the arrival of the internet, several technological solutions have emerged focused on this new medium of communication. One such solution came after the creation of smartphones, where a new market was created for the consumption of applications, which solve daily problems of the population. The problem can be solved by joining these two technologies, a new product, called Professional Indicado, which indicates qualified professionals for different types of services that may be necessary for the population of a region. A web system and a mobile system that stores information from professionals in the city of Criciúma, a solution created with technologies learned during the course of Computer Engineering, such as JavaEE in the discipline of object orientation, Javascript, HTML and CSS in the discipline of Web Technologies. In addition to other solutions such as AngularJS, Wildfly Swarm, Postgresql, Maven, and Ionic for the mobile solution.

Key-words: *Liberal Professional, Service Provision, System, Web, Mobile.*

AGRADECIMENTOS

Agradeço primeiramente a Deus e a minha família que me proporcionou a oportunidade de entrar na faculdade e cursar Engenharia da Computação. Agradeço a meus amigos e minha namorada Isadora Farias Lummertz pelo apoio e compreensão no processo do Trabalho de Conclusão de Curso. Agradeço ao apoio de Jhony Alceu Pereira que contribuiu com dicas para soluções técnicas quando necessitado e pelo apoio de iniciar e levar o projeto em frente. Agradeço ao orientador Daniel Basílio Marcelo que soube como organizar da melhor forma possível a apresentação das ideias no artigo e que deu dicas valiosas em como levar o projeto adiante. Agradeço ao coordenador do curso Gustavo dos Santos de Lucca pela orientação no decorrer do curso de Engenharia.