

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

APLICATIVO MÓVEL PARA DIVULGAÇÃO DE PRODUTOS

MARCO AURÉLIO RUTZEN

BLUMENAU
2015

2015/2-12

MARCO AURÉLIO RUTZEN

APLICATIVO MÓVEL PARA DIVULGAÇÃO DE PRODUTOS

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Marcel Hugo, Mestre - Orientador

**BLUMENAU
2015**

2015/2-12

APLICATIVO MÓVEL PARA DIVULGAÇÃO DE PRODUTOS

Por

MARCO AURÉLIO RUTZEN

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: _____
Prof. Marcel Hugo, Mestre – Orientador, FURB

Membro: _____
Prof. Everaldo Artur Grahl, Mestre – FURB

Membro: _____
Prof. Luciana Pereira de Araújo, Mestre – FURB

Blumenau, 08 de Dezembro de 2015

Dedico este trabalho a todos os amigos e familiares, especialmente aqueles que me ajudaram diretamente na realização deste.

AGRADECIMENTOS

À minha família, que me apoiou em todos os momentos e forneceu a estrutura necessária para chegar a este momento.

Aos meus amigos Juan Pablo da Rocha e Paulo Ricardo de Mello pela ajuda e pelos momentos de descontração.

Ao meu orientador, professor Marcel Hugo, por ter contribuído com suas sugestões, incentivando a conclusão deste trabalho e acreditado nele em todos os momentos.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau, que contribuíram com seus ensinamentos e experiências durante todo o curso.

Pessimismo leva à fraqueza, otimismo ao poder.

William James

RESUMO

Muitos usuários não têm conhecimento dos produtos e informações das empresas que estão presentes perto de sua residência, como por exemplo floriculturas, farmácias, lojas de vestuários, eletroeletrônicos, revendedoras de carros, entre outros. Isto pode ser um problema gerado pela falta de divulgação da empresa e seus produtos, por não ter um site ou um aplicativo móvel a seu favor. Este trabalho apresenta um aplicativo para a plataforma Android que oferece uma alternativa para que isto não aconteça, disponibilizando funcionalidades para que os usuários tenham a oportunidade de pesquisar e interagir com empresas e produtos presentes em sua região, sendo que os produtos e informações da empresa são cadastrados a partir de um módulo web. Para facilitar a manutenção do sistema foi criado um mecanismo para geração de cadastro de tipos de produtos baseado em metadados, fazendo com que não sejam necessárias alterações em código fonte, banco de dados e telas do sistema referentes a cadastros de produtos. Foi desenvolvido um *web service* juntamente com as bibliotecas GSON e Jersey para comunicação com o aplicativo móvel. Na persistência dos dados foi utilizado o SGDB MySQL, além do banco de dados nativo do Android SQLite. Os resultados obtidos demonstram de forma positiva a utilidade do sistema, que consegue proporcionar aos usuários conhecimento e interação com as empresas de sua região. Pode-se afirmar que as funcionalidades desenvolvidas para a manutenção dos cadastros dos tipos de produto se demonstraram bastante úteis pois diminuem consideravelmente o custo e tempo de manutenção de cadastros de produtos. Sendo assim, é válido afirmar que as ferramentas e tecnologias utilizadas se demonstraram satisfatórias para o desenvolvimento do sistema.

Palavras chaves: Aplicativo móvel. Metadados. Android.

ABSTRACT

Many users are unaware of goods and information from companies that are present near your residence, such as florists, pharmacies, clothing stores, electronics, car dealerships, among others. This can be a problem caused by the lack of disclosure of the company and its products, for not having a website or a mobile application on their behalf. This paper presents an application for the Android platform that offers an alternative to prevent this from happening, providing functionality for users to have the opportunity to research and interact with companies and products present in your area, and the company's products and information are registered from a web module. To facilitate the maintenance of the system was a mechanism for generating registration kinds of products based on metadata, making them unnecessary changes to source code, database and system screens concerning products of entries. A web service was developed together with the Gson and Jersey libraries to communicate with the mobile application. The persistence of the data was used DBMS MySQL, in addition to native database SQLite Android. The results show a positive system utility, which can provide users with knowledge and interaction with businesses in your area. It can be said that the features developed for the maintenance of records of the types of product have proved quite useful as it significantly decrease the cost and time of product maintenance records. Therefore, it is fair to say that the tools and technologies used have proved satisfactory for the development of the system.

Key-words: Mobile app. Metadata. Android.

LISTA DE FIGURAS

Figura 1– Telas de pesquisa de estabelecimentos e resultados da pesquisa no aplicativo Guia Fácil	20
Figura 2 – Telas de pesquisa de categorias e pesquisa de produtos	21
Figura 3 – Tela de comparação de preços, avaliações e detalhes no aplicativo Buscapé	21
Figura 4 – Página inicial gerada pela ferramenta	22
Figura 5 – Página gerada para listagem das entidades	23
Figura 6 – Página gerada para inserção de registros	23
Figura 7 – Diagrama de casos de uso do módulo web	27
Figura 8 – Diagrama de casos de uso do aplicativo móvel	27
Figura 9 – Diagrama de Atividades do processo de personalização de tipo de produto	28
Figura 10 – Diagrama de Atividades do processo de atendimento de requisição	29
Figura 11 – Modelo de Entidade Relacionamento do módulo web	30
Figura 12 – Modelo de Entidade-Relacionamento do aplicativo móvel	31
Figura 13 – Primeira parte do diagrama de classes do módulo web	32
Figura 14 – Segunda parte do diagrama de classes do módulo web	33
Figura 15 – Terceira parte do diagrama de classes do módulo web.....	34
Figura 16 – Diagrama de classes do aplicativo móvel	35
Figura 17 – Tela de autenticação do módulo web	45
Figura 18 – Início da página de cadastro de Empresa	45
Figura 19 – Menu completo do módulo web.....	46
Figura 20 – Requisição atendida pelo administrador	47
Figura 21 – Página de cadastro de tipos de produto	48
Figura 22 – Página para manter ramos de atividade.....	48
Figura 23 – Página para manter usuários.....	49
Figura 24 – Página para personalizar tipos de produto	49
Figura 25 – Página para manter produtos	50
Figura 26 – Página para visualizar perguntas e comentários.....	51
Figura 27 – Tela de autenticação do aplicativo móvel	51
Figura 28 – Tela para cadastro de usuário do aplicativo móvel	52
Figura 29 – Tela do Menu Principal do aplicativo móvel	52
Figura 30 – Tela de pesquisa de empresas do aplicativo móvel.....	53

Figura 31 – Tela de informações da empresa do aplicativo móvel	54
Figura 32 – Tela de pesquisa de tipo de produto do aplicativo móvel	54
Figura 33 – Tela de pesquisa de produtos do aplicativo móvel.....	55
Figura 34 – Tela para pesquisa avançada de produtos	56
Figura 35 – Tela de informações do produto no aplicativo móvel.....	56
Figura 36 – Tela para visualizar e criar perguntas para o produto no aplicativo móvel	57
Figura 37 – Tela para visualizar e criar comentários para o produto no aplicativo móvel	58
Figura 38 – Telas para visualizar os produtos e empresas favoritos	58

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais do módulo web	25
Quadro 2 - Requisitos não funcionais do módulo web.....	26
Quadro 3 - Requisitos funcionais do aplicativo móvel.....	26
Quadro 4 - Requisitos não funcionais do aplicativo móvel.....	26
Quadro 5 – Método de inserção de requisição	37
Quadro 6 – Criação da classe <code>QueryBuilder</code>	37
Quadro 7 – Classe de acesso ao banco utilizada pelo <i>web service</i>	38
Quadro 8 – Classe do aplicativo móvel responsável pelas requisições de comentários.....	39
Quadro 9 – Classe do <i>web service</i> responsável pelas requisições de comentários.....	40
Quadro 10 – Método utilizado para gerar o cadastro dos tipos de produto.....	41
Quadro 11 – Métodos para criar e alterar tabelas de tipos de produto na base de dados	41
Quadro 12 – Método responsável por gerar comandos SQL para criar tabelas de tipos de produto.....	42
Quadro 13 - Método responsável por gerar comandos SQL para alterar tabelas de tipos de produto.....	44
Quadro 14 – Comparativo com trabalhos correlatos do aplicativo móvel desenvolvido.....	60
Quadro 15 – Comparativo com trabalhos correlatos do módulo web desenvolvido.....	61
Quadro 16 – Detalhamento do Caso de Uso 02 – Manter o cadastro de produtos	66
Quadro 17 – Detalhamento do Caso de Uso 06 – Cadastrar requisições.....	67
Quadro 18 – Detalhamento do Caso de Uso 03 – Personalizar tipo de produto	67
Quadro 19 – Detalhamento do Caso de Uso 10 – Manter o cadastro de tipos de produtos.....	68
Quadro 20 – Detalhamento do caso de uso 03 – Pesquisar produtos.....	69
Quadro 21 – Detalhamento do Caso de Uso 04 – Pesquisar empresas	70
Quadro 22 – Tabela Empresa	71
Quadro 23 – Tabela Ramo.....	71
Quadro 24 – Tabela Imagem.....	72
Quadro 25 – Tabela <code>Requisicao</code>	72
Quadro 26 – Tabela <code>TipoProduto</code>	72

Quadro 27 – Tabela Atributo	72
Quadro 28 – Tabela AtributoEmpresa	73
Quadro 29 – Tabela Usuario	73
Quadro 30 – Tabela Comentario	73
Quadro 31 – Tabela Pergunta	74
Quadro 32 – Tabela Empresa	75
Quadro 33 – Tabela EmpresaFavorita	75
Quadro 34 – Tabela Usuario	75
Quadro 35 – Tabela TipoProduto.....	76
Quadro 36 – Tabela TipoProdutoEmpresa	76
Quadro 37 – Tabela Produto	76
Quadro 38 – Tabela ProdutoFavorito	76
Quadro 39 – Tabela RamoAtividade	76
Quadro 40 – Tabela Imagem.....	77

LISTA DE ABREVIATURAS E SIGLAS

ADT – *Android Developer Tools*

CEP – Código de Endereçamento Postal

CNPJ – Cadastro Nacional da Pessoa Jurídica

CRUD – *Create, Read, Update and Delete*

EA – Enterprise Architect

IDE – *Integrated Development Environment*

JSF – *Java Server Faces*

JSON – *JavaScript Object Notation*

MER – Modelo Entidade Relacionamento

NoSQL – Not Only SQL

REST – *Representational State Transfer*

RF – Requisito Funcional

RNF – Requisito Não Funcional

SGBD – Sistema de Gerenciamento de Banco de Dados

SQL – Structured Query Language

UC – *Use Case*

UI – *User Interface*

URL – *Uniform Resource Locator*

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 OBJETIVOS.....	16
1.2 ESTRUTURA.....	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 APLICATIVOS MÓVEIS.....	17
2.2 METADADOS	18
2.3 TRABALHOS CORRELATOS	19
3 DESENVOLVIMENTO	24
3.1 LEVANTAMENTO DE INFORMAÇÕES	24
3.2 ESPECIFICAÇÃO	25
3.2.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	25
3.2.2 MODELO DE CASOS DE USO	26
3.2.3 DIAGRAMAS DE ATIVIDADES	28
3.2.4 MODELO DE ENTIDADE-RELACIONAMENTO	29
3.2.5 DIAGRAMAS DE CLASSES	31
3.3 IMPLEMENTAÇÃO	35
3.3.1 Técnicas e ferramentas utilizadas.....	35
3.3.1.1 Ferramentas.....	36
3.3.1.2 Persistência dos dados	36
3.3.1.3 Comunicação com o servidor	38
3.3.1.4 Geração de cadastros baseada em metadados.....	40
3.3.2 Operacionalidade da implementação	44
3.3.2.1 Módulo Web	44
3.3.2.2 Aplicativo Móvel	51
3.4 RESULTADOS E DISCUSSÕES.....	59
3.4.1 COMPARATIVO DO APLICATIVO MÓVEL	59
3.4.2 COMPARATIVO DO MÓDULO WEB	60
4 CONCLUSÕES	62
4.1 EXTENSÕES	63
REFERÊNCIAS	64
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO DO MÓDULO WEB	66

APÊNDICE B – DESCRIÇÃO DOS CASOS DE USO DO APLICATIVO MÓVEL.....	69
APÊNDICE C – DICIONÁRIO DE DADOS DO MÓDULO WEB.....	71
APÊNDICE D – DICIONÁRIO DE DADOS DO APLICATIVO MÓVEL.....	75

1 INTRODUÇÃO

Já não é mais novidade que os dispositivos móveis conquistaram o mercado e são um objeto presente na rotina de muitas pessoas. Além de oferecerem um meio de entretenimento e comunicação, disponibilizam meios para simplificar tarefas do dia-a-dia. Atualmente, aproximadamente 43 milhões de brasileiros com 12 anos ou mais navegam pela internet utilizando dispositivos móveis (DATA FOLHA, 2014).

Investir em um aplicativo móvel é essencial para o crescimento da empresa, principalmente no início de sua jornada. Segundo E-Commerce Brasil (2014), “As pequenas e médias empresas que apostam primeiro em mobile crescem até duas vezes mais rápido a sua receita que os concorrentes que investem por último em tecnologia móvel”. Há meios além de aplicativos móveis que conseguem oferecer benefícios para as empresas, mas este ainda é pouco aproveitado e pode ser uma grande oportunidade para gerar resultados para empresas que decidirem investir neste recurso.

É comum que empresas invistam em sites para tornar seus produtos e serviços mais disponíveis e visíveis ao público no ambiente digital. Já estão claras para muitos empreendedores as vantagens do investimento de migrar seus negócios para a internet. Mas há outra possibilidade que ainda é pouco explorada, principalmente por pequenas empresas, e que também pode gerar grandes resultados: os aplicativos para celular (SOUNDY, 2015).

Os dispositivos móveis são um meio de marketing para as empresas. Aplicativos podem oferecer competitividade, possibilitando a divulgação da marca e seus produtos e permitir que estejam mais próximas de seus clientes. Desenvolver a identidade digital de uma empresa ou marca é tarefa obrigatória para quem quer aparecer no seu mercado de atuação (LEONEL, 2014).

A TiWorld (2014) destaca que “facilitar a vida do cliente e fazer com que ele seja fiel à sua empresa é o que sempre se busca, e isso pode ser facilmente atingido através de um aplicativo.”. Aplicativos móveis que conseguem oferecer serviços de utilidade conseguem criar um vínculo de segurança entre o cliente e a empresa. Deste modo, quando o cliente precisar de algum serviço relacionado aos oferecidos pela empresa, ela será facilmente lembrada e indicada.

Visando auxiliar empresas a aumentar o alcance da visibilidade em sua região, propõe-se neste trabalho o desenvolvimento de um aplicativo móvel para divulgação dos produtos, apoiado por uma aplicação web para cadastro de produtos. O aplicativo oferecerá recursos para que usuários consigam procurar produtos conforme seus interesses e necessidades. Além

disso, os usuários poderão deixar comentários nos produtos ou entrar em contato com a empresa a partir de perguntas para os produtos.

Para conseguir aderência em um número maior de produtos surge a necessidade de realizar constantes manutenções e modificações no sistema, como adicionar campos em telas, alterar código fonte, criar colunas e tabelas no banco de dados, liberar versões do produto, etc. Utilizando conceitos referentes a metadados, que são definidos como dados sobre dados, é possível automatizar estas tarefas repetitivas que influenciam no tempo e custo para manutenção do sistema.

1.1 OBJETIVOS

O objetivo geral deste trabalho é disponibilizar um aplicativo móvel para visualização de produtos e empresas, os quais poderão ser cadastrados a partir de um módulo web.

Os objetivos específicos deste trabalho são:

- a) disponibilizar para o administrador do sistema um recurso para geração dinâmica de cadastros de produtos baseado em metadados;
- b) disponibilizar para a empresa recurso para cadastro de produtos e informações;
- c) disponibilizar para o usuário recurso para busca de produtos e empresas.

1.2 ESTRUTURA

No primeiro capítulo tem-se a introdução ao tema principal deste trabalho com a apresentação da justificativa e dos objetivos.

No segundo capítulo apresenta-se a fundamentação teórica sobre aplicativos móveis, metadados e trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento da aplicação.

No quarto capítulo tem-se as conclusões e as sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos a serem apresentados nas seções a seguir, tais como aplicativos móveis e metadados, além dos trabalhos correlatos.

2.1 APLICATIVOS MÓVEIS

Os aplicativos móveis estão presentes em *smartphones* e *tablets*, e estes são desenvolvidos para as mais diversas utilidades, como acessar redes sociais, jogos, educação, músicas, comunicação, entre outras. Como estão presentes nos dispositivos móveis os aplicativos podem ser uma oportunidade para as empresas que buscam expandir seus meios de divulgação de produtos e estreitar o relacionamento com o cliente.

Um aplicativo *mobile* permite que haja melhorias na comunicação com o público-alvo de uma empresa em dois níveis: relacionamento e vendas. No primeiro caso, é possível aumentar o volume de interações entre consumidores e empresas de uma maneira muito mais estratégica e mensurável quando comparado aos métodos tradicionais, que envolvem redes sociais, por exemplo. Já na segunda situação, é possível dizer que se cria um novo canal de marketing da marca, onde é possível, ao mesmo tempo, facilitar a compra a partir de diferentes meios e divulgar novos produtos (SOUNDY, 2014).

O dispositivo móvel tem como grande diferencial estar sempre por perto e disponível para seu usuário durante todo o dia. Deste modo os aplicativos móveis são mais propensos a serem acessados. É comum as pessoas usarem seus dispositivos em uma fila de banco, na fila de espera de uma consulta médica ou em várias outras situações em que é necessário esperar por alguém ou por algo em específico. Isso faz com que as empresas tenham mais oportunidades de contato com seus públicos de interesse e de tráfego em seus sites e aplicativos (E-COMMERCE BRASIL, 2015).

Os aplicativos móveis conseguem se destacar dos *websites* por vários motivos, entre eles está o acesso *off-line*, que segundo a Soundy (2015), “Essa função é uma poderosa arma de marketing, pois o cliente pode utilizar seus serviços mesmo sem estar conectado a uma rede móvel ou *wifi*.” Os dispositivos móveis podem ser considerados um canal de marketing para a empresa, pois a partir de um aplicativo móvel é possível oferecer serviços diferenciados para os clientes, criando uma relação mais próxima.

O dispositivo móvel não é apenas um meio, mas serve também como uma extensão de sua empresa e de todos os seus discursos, metas, valores e propósitos. Quando você se preocupa em oferecer serviços que são relevantes para seus clientes, antecipando suas necessidades antes mesmo que eles pensem nelas, você promove o engajamento e a lealdade à marca, criando conexões emocionais e um relacionamento positivo para os dois lados (CHAMELLA, 2015).

Ter a seu favor um aplicativo móvel é uma vantagem que deve ser considerada, principalmente por aqueles que desejam expandir suas vendas e rede de clientes a partir da divulgação da marca. Segundo a Soundy (2015), “Uma empresa que não utiliza um aplicativo móvel reduz sua possibilidade de vendas e de divulgação da sua marca. A tecnologia a cada dia nos mostra uma nova ferramenta viável. Não devemos ter receio em usá-las.”.

2.2 METADADOS

Metadados são comumente definidos como dados que descrevem dados. Eles podem beneficiar aplicações que necessitam realizar consultas complexas com quantidades elevadas de dados, por isso é um conceito comum em sistemas de gerenciamento de banco de dados. Uma das primeiras utilizações do termo metadado foi em esquemas de banco de dados, onde esquemas são informações que descrevem o formato de um registro, como tipo, nome e tamanho das colunas de uma tabela (DAMASCENO, 2005, p. 25).

Na Figura 1 é apresentado o resultado de uma consulta por colunas de um esquema no sistema de gerenciamento de banco de dados (SGDB) MySQL. Nela são demonstradas informações referentes ao nome da tabela, nome da coluna, tipo da coluna, tamanho máximo da coluna e precisão numérica da coluna.

Figura 1 – Consulta realizada no esquema do banco de dados MySQL

TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION
carro	Datafabricacao	date	NULL	NULL
carro	Ano	int	NULL	10
carro	Comprimento	decimal	NULL	13
carro	BancoRegulagem	varchar	3	NULL
comentario	id	int	NULL	10
comentario	fk_empresa_comentario	int	NULL	10

Os metadados são constituídos de dados que quando interpretados descrevem um objeto, que poderá ser interpretado pela aplicação como uma ação, um comportamento, uma mensagem, uma resposta, um componente, entre outras possibilidades. Deste modo, além de facilitar a comunicação entre diferentes aplicações, é possível executar ações dinâmicas na aplicação conforme o resultado da interpretação dos dados, como por exemplo, orientar como uma tela deve ser construída ou como ela deve se comportar.

Metadados são utilizados para descrever as propriedades e o significado dos dados para melhor entendê-los, utilizá-los e gerenciá-los, em outras palavras, os metadados definem a sintaxe e a semântica dos objetos de dados das aplicações, permitindo que estas saibam como processar solicitações e tarefas. Através da sua definição, a troca de informações entre diferentes aplicações pode ser realizada (DAMASCENO, 2005, p. 25).

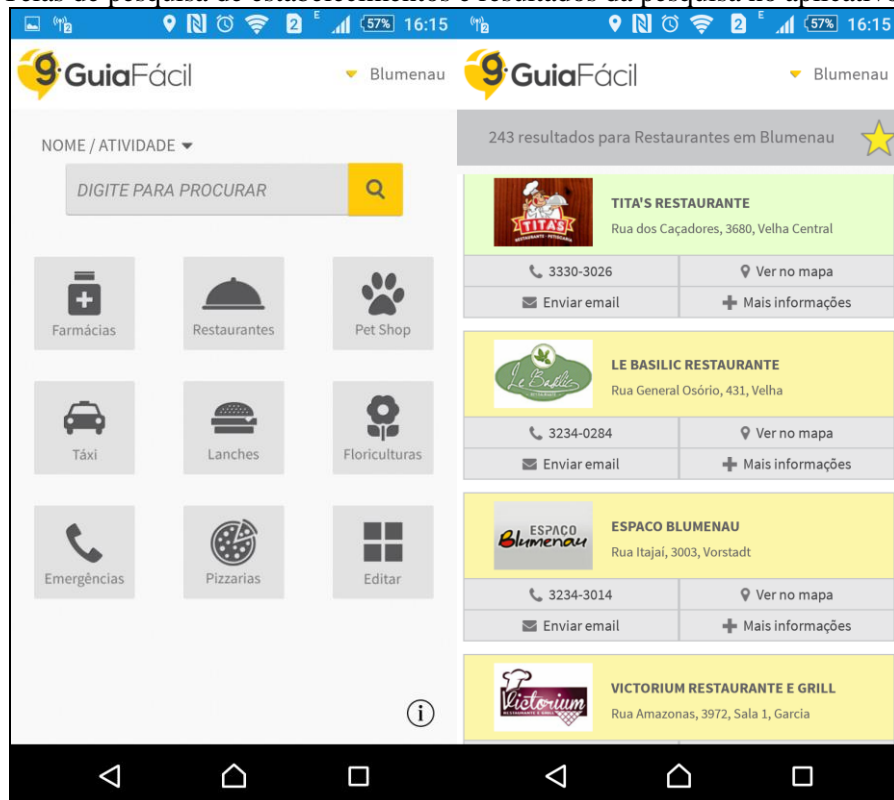
Com o uso dos conceitos referentes a metadados é possível automatizar tarefas repetitivas que influenciam no tempo e custo para manutenção do sistema, como criar novas tabelas, campos, alteração de telas, mudança em códigos fontes e entre outras. Os metadados podem ser utilizados para descrever objetos ou tornar pública sua existência. Eles disponibilizam, descrevem, localizam e auxiliam na compreensão dos dados, transformando-os em conhecimento (BENACCHIO; VAZ, 2008, p. 33).

2.3 TRABALHOS CORRELATOS

Pode-se citar como trabalhos correlatos as ferramentas comerciais Guia Fácil (2015) e Buscapé (2015), além do Trabalho Final de Graduação de Felipe Gabriel Arend (2011). Os dois primeiros oferecem um meio de divulgação de produtos enquanto o terceiro disponibiliza uma ferramenta para geração de código para operações *Create, Read, Update and Delete* (CRUD) baseada em metadados.

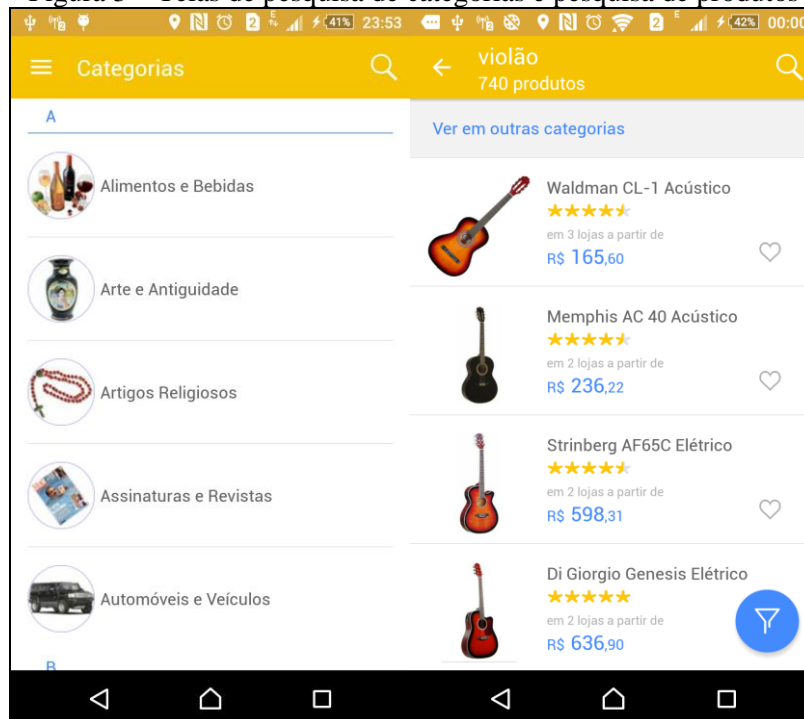
O Guia Fácil para dispositivos móveis disponibiliza mapas, telefones e endereços comerciais das principais cidades de Santa Catarina. Nele é possível encontrar informações sobre lojas, restaurantes, hotéis, farmácias, entre outros (GUIA FÁCIL COMUNICAÇÃO, 2015). O aplicativo está disponível para *download* nas plataformas Android e iOS, com a possibilidade de acesso pelo navegador. Ele permite realizar buscas por endereços, código de endereçamento postal (CEP), nome, atividade e/ou telefone. Na pesquisa são listadas informações básicas sobre os estabelecimentos, como telefone, endereço, breve descrição, visualizar a localização no mapa e enviar um e-mail (Figura 2).

Figura 2– Telas de pesquisa de estabelecimentos e resultados da pesquisa no aplicativo Guia Fácil



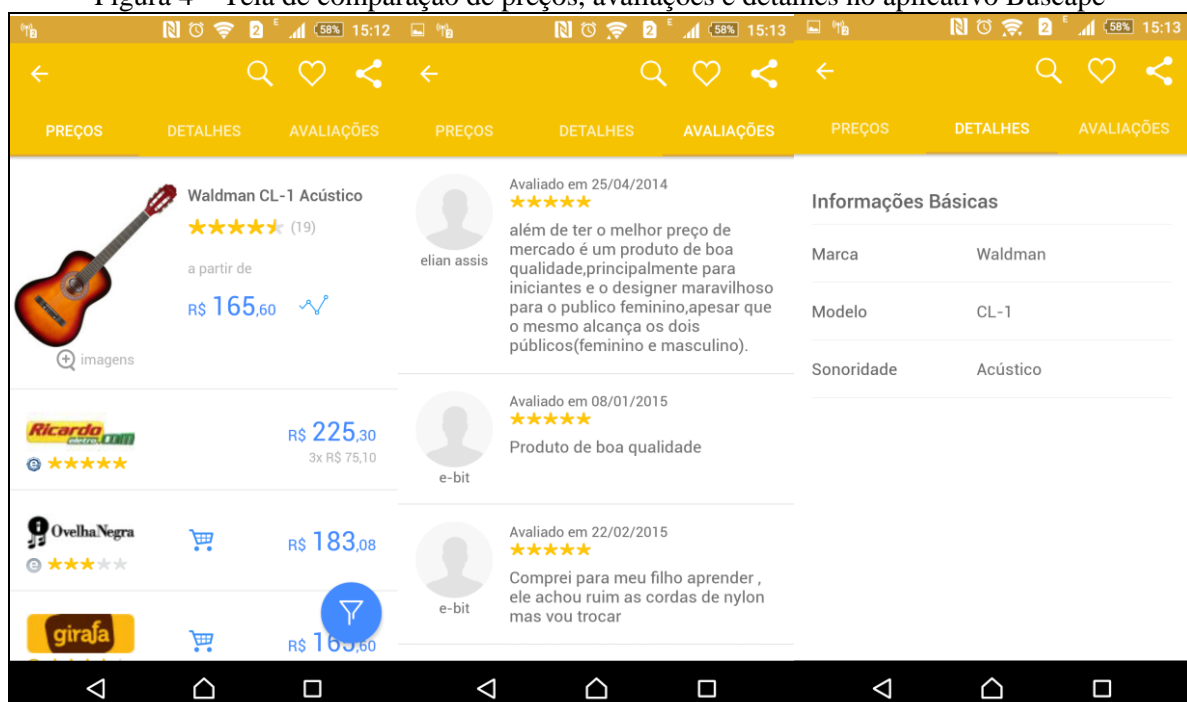
O Buscapé serve como um comparador de preços, lojas e produtos, com o intuito de incentivar a compra consciente e oferecer a oportunidade de o usuário pagar um preço justo pelo que deseja comprar (BUSCAPÉ, 2015). Disponível para *download* nas plataformas Android e iOS e com a possibilidade de acesso pelo navegador, o Buscapé (2015) permite o usuário realizar buscas por nome, marca, modelo e categoria dos produtos (Figura 3).

Figura 3 – Telas de pesquisa de categorias e pesquisa de produtos



Para auxiliar o consumidor conseguir a melhor experiência, além da comparação de preços a ferramenta disponibiliza um indicador para as empresas e o produto, em conjunto com a possibilidade de visualizar e realizar avaliações para os produtos. É possível também marcar os produtos como favoritos para visualizar novamente em outro momento, além de receber uma notificação quando os preços do produto diminuïrem (Figura 4).

Figura 4 – Tela de comparação de preços, avaliações e detalhes no aplicativo Buscapé



Comparando os trabalhos correlatos citados referentes ao aplicativo móvel é possível identificar que o Buscapé (2015) tem como principal foco a pesquisa por produtos e comparação de preços independente da região do usuário. O Guia Fácil (2015) é mais voltado para pesquisas de empresas e visualização de suas informações básicas, além de oferecer a possibilidade de o usuário selecionar a região que deseja realizar a busca. Este trabalho leva em consideração no seu desenvolvimento os recursos oferecidos pelos dois aplicativos.

Arend (2011) propôs em seu Trabalho Final de Graduação, o desenvolvimento de uma ferramenta para geração de código para operações CRUD para sistemas web com *Java Server Faces* (JSF). A ferramenta não tem como objetivo gerar um sistema pronto para uso, mas facilitar o desenvolvimento de aplicações web. Para a geração são utilizadas informações da base de dados com modelos gerados por um motor de *templates*. As entidades e modelos do sistema são gerados a partir de metadados que são obtidos da base de dados. Para ajudar na geração de arquivos e diretórios do sistema é utilizado o motor de *templates* Apache Velocity.

Na Figura 5 é apresentada a página inicial do sistema. As opções disponíveis são geradas pela ferramenta conforme as entidades presentes no banco de dados.

Figura 5 – Página inicial gerada pela ferramenta



Fonte: Arend (2011, p.35).

Após selecionar uma das opções do menu principal, é apresentada a página de listagem das informações da entidade (Figura 6). Esta página disponibiliza as opções para adicionar, visualizar, alterar ou excluir novos registros.

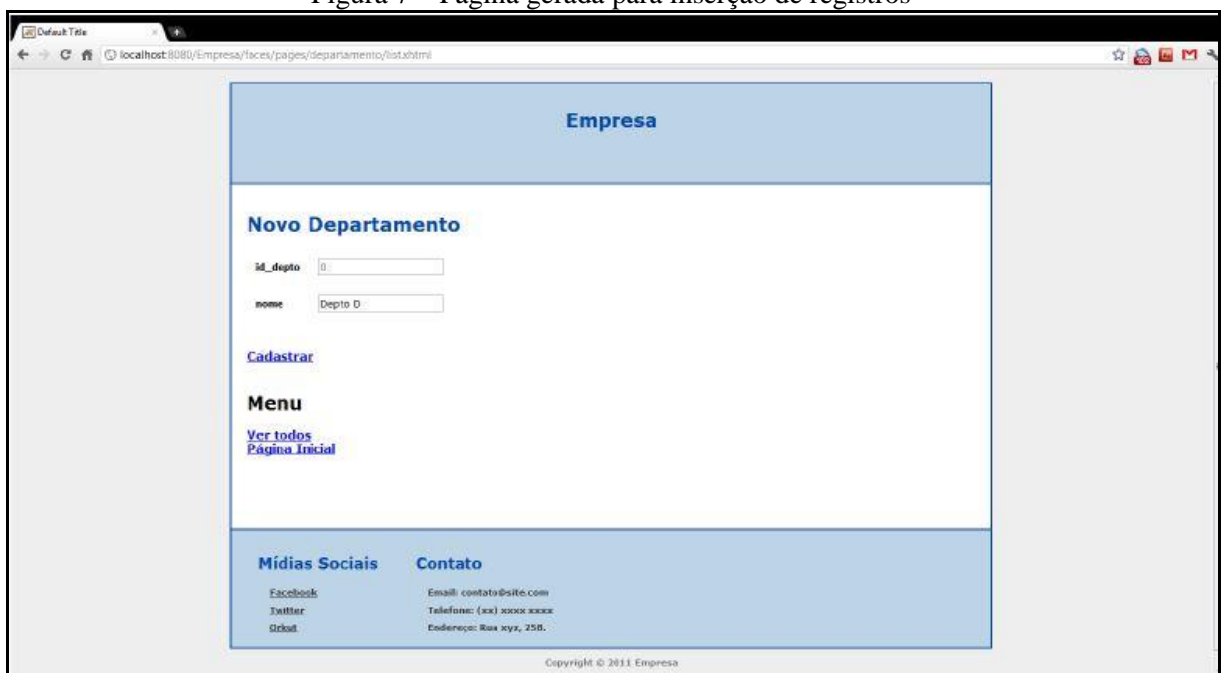
Figura 6 – Página gerada para listagem das entidades



Fonte: Arend (2011, p.36).

As páginas para operações CRUD são bem similares e apresentam o formulário com todos os campos. Na Figura 7 é apresentada a página gerada para inserção de registros.

Figura 7 – Página gerada para inserção de registros



Fonte: Arend (2011, p.36).

3 DESENVOLVIMENTO

Neste capítulo são descritos os aspectos técnicos do desenvolvimento do sistema, composto pelo levantamento das informações e sua especificação, que contém requisitos funcionais e não funcionais, diagramas de casos de uso, diagramas de atividades, modelo de entidade relacionamento, e diagramas de classes. Em seguida, são apresentadas as técnicas e ferramentas utilizadas, além da operacionalidade do sistema, os resultados obtidos e discussões pertinentes.

3.1 LEVANTAMENTO DE INFORMAÇÕES

O aplicativo móvel P-Market foi desenvolvido com a intenção de oferecer aos usuários e empresas um meio de interação maior em sua região, possibilitando assim que as empresas tenham maior visibilidade para os usuários mais próximos de sua localização, considerando bairro, cidade e estado. A aplicação web P-Market foi desenvolvida para as empresas e é focado em oferecer aderência a vários tipos de produtos dos mais variados ramos, podendo-se citar cosméticos, automotivo, eletroeletrônicos, farmacêutico, informática, materiais de construção, vestuário, têxtil, entre outros.

As funcionalidades que a empresa tem à sua disposição são baseadas na divulgação de seus produtos e contato com seus usuários, a partir de comentários e perguntas. Após se registrar no sistema ela pode iniciar o processo para cadastrar os seus produtos que serão disponibilizados para pesquisa no aplicativo móvel, bastando personalizar o tipo de produto desejado e escolhendo as características do produto. Se necessário ela pode abrir uma requisição para o administrador do sistema criar um novo tipo de produto, como por exemplo violões, pianos, *notebooks*, *smartphones*, entre outros. Para facilitar a comunicação com os usuários e possibilitar que a empresa esteja ciente do *feedback* dos mesmos, é disponibilizado um painel para responder as perguntas e visualizar os comentários feitos para seus produtos.

As funcionalidades disponíveis para os usuários são baseadas na procura de produtos em sua região, a possibilidade de marcar produtos e empresas como favoritos, para posterior acesso a esta informação *off-line*, e contato com as empresas por meio de comentários e perguntas. Para o administrador são oferecidas funcionalidades para manter os tipos de produto do sistema, a qual oferece a possibilidade de geração de cadastro do produto, além da manutenção das empresas e usuários do sistema.

3.2 ESPECIFICAÇÃO

Nesta seção são apresentados os Requisitos Funcionais (RF) e Não Funcionais (RNF) do sistema, os modelos de Casos de Uso (UC), diagramas de atividades e de classes, além do Modelo Entidade-Relacionamento (MER). Com exceção dos requisitos, a modelagem foi criada utilizando a ferramenta Enterprise Architect (EA).

3.2.1 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Nesta subseção são apresentadas as principais funcionalidades do sistema. O Quadro 1 apresenta os requisitos funcionais definidos para o módulo web e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Quadro 1 - Requisitos funcionais do módulo web

Requisitos Funcionais	Caso de Uso
RF01: O módulo web deverá permitir a empresa se cadastrar.	UC01
RF02: O módulo web deverá permitir a empresa manter o cadastro de produtos.	UC02
RF03: O módulo web deverá permitir a empresa personalizar as características de tipos de produtos.	UC03
RF04: O módulo web deverá permitir a empresa responder as perguntas feitas pelos usuários para seus produtos.	UC04
RF05: O módulo web deverá permitir a empresa visualizar os comentários feitos pelos usuários para seus produtos.	UC05
RF06: O módulo web deverá permitir a empresa enviar requisições de alteração e inclusão de tipos de produto para o administrador.	UC06
RF07: O módulo web deverá permitir a empresa visualizar as respostas de requisições feitas para o administrador.	RF07
RF08: O módulo web deverá permitir o administrador manter o cadastro de usuários.	UC08
RF09: O módulo web deverá permitir o administrador manter o cadastro de empresas.	UC09
RF10: O módulo web deverá permitir o administrador manter o cadastro de tipos de produtos.	UC10
RF11: O módulo web deverá permitir o administrador manter requisições de alterações e inclusões de tipos de produto	UC11
RF12: O módulo web deverá permitir o administrador manter ramos de atividade	UC12
RF13: O módulo web deverá permitir a empresa efetuar a autenticação	UC13

O Quadro 2 lista os requisitos não funcionais previstos para o módulo web.

Quadro 2 - Requisitos não funcionais do módulo web

Requisitos Não Funcionais
RNF01: O módulo web deverá ser desenvolvido em ambiente Java.
RNF02: O módulo web deverá ser acessível via Google Chrome (Versão 41.0.2272.101 ou superior).
RNF03: O módulo web deverá trocar informações com o aplicativo móvel a partir de <i>web services</i> .
RNF04: O módulo web deverá utilizar o banco de dados MySQL.

O Quadro 3 lista os requisitos funcionais previstos para o aplicativo móvel.

Quadro 3 - Requisitos funcionais do aplicativo móvel

Requisitos Funcionais	Caso de Uso
RF01: O aplicativo móvel deverá permitir o usuário manter seu cadastro.	UC01
RF02: O aplicativo móvel deverá permitir o usuário efetuar a autenticação.	UC02
RF03: O aplicativo móvel deverá permitir o usuário pesquisar produtos.	UC03
RF04: O aplicativo móvel deverá disponibilizar suas informações <i>off-line</i> , desde que o produto pesquisado seja cadastrado por uma empresa marcada como favorita.	UC03
RF05: O aplicativo móvel deverá permitir o usuário pesquisar empresas.	UC04
RF06: O aplicativo móvel deverá permitir o usuário marcar produtos para serem acompanhados.	UC05
RF07: O aplicativo móvel deverá permitir o usuário marcar empresas para serem acompanhadas.	UC06
RF08: O aplicativo móvel deverá permitir o usuário visualizar as empresas que estão sendo acompanhadas.	UC07
RF09: O aplicativo móvel deverá permitir o usuário visualizar os produtos que estão sendo acompanhados.	UC08
RF10: O aplicativo móvel deverá permitir o usuário criar perguntas para um produto.	UC09
RF11: O aplicativo móvel deverá permitir o usuário criar comentários para um produto.	UC10
RF12: O aplicativo móvel deverá permitir o usuário visualizar as respostas feitas pela empresa para as perguntas.	UC11

O Quadro 4 lista os requisitos não funcionais previstos para o aplicativo móvel.

Quadro 4 - Requisitos não funcionais do aplicativo móvel

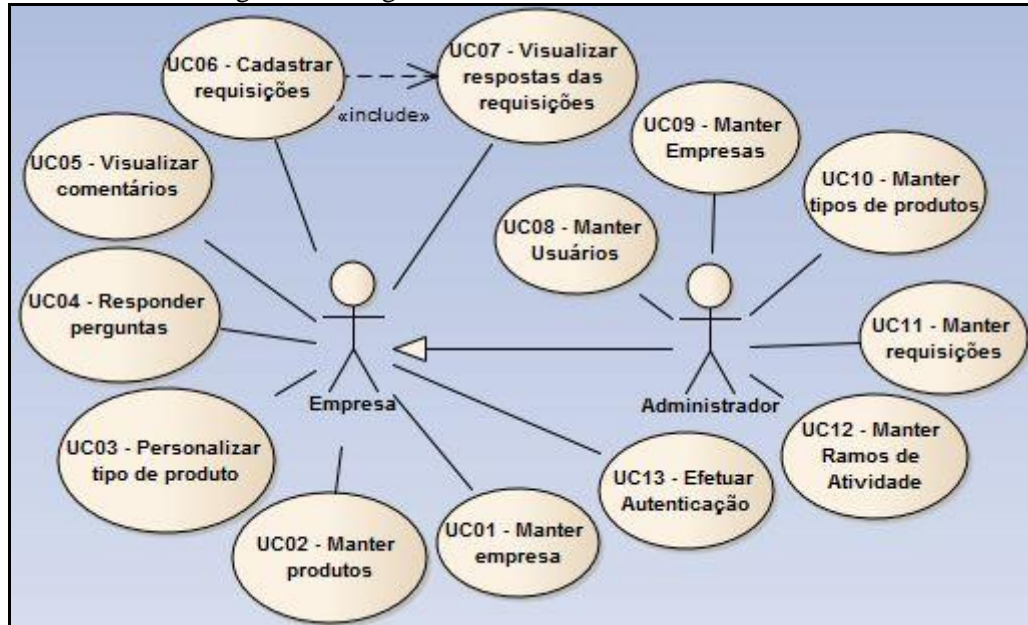
Requisitos Não Funcionais
RNF01: O aplicativo móvel deverá ser desenvolvido na plataforma Android sendo compatível com a versão 5.0.1 ou superior.
RNF02: O aplicativo móvel deverá trocar informações com o módulo web a partir de <i>web services</i> .

3.2.2 MODELO DE CASOS DE USO

Nesta subseção são apresentados os diagramas de casos de uso do módulo web e do aplicativo móvel, sendo que a descrição expandida dos principais casos de uso é apresentada nos Apêndices A e B. Na Figura 8 é apresentado o diagrama de casos de uso do módulo web. Nela estão contidos o ator *Empresa*, que tem como principal caso de uso o cadastro dos

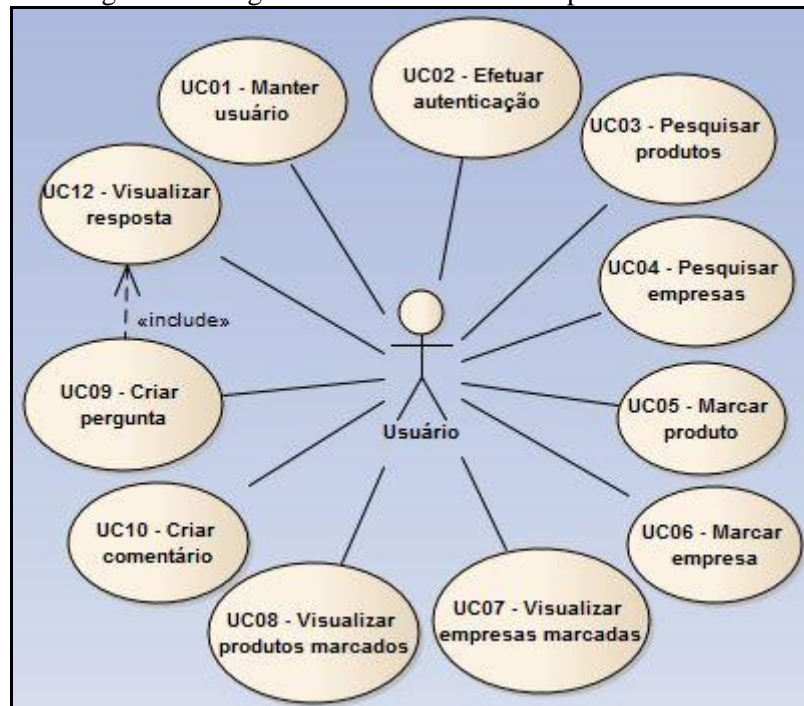
produtos, e Administrador que tem como principais casos de uso manter tipos de produtos, requisições e ramos de atividade.

Figura 8 – Diagrama de casos de uso do módulo web



Na Figura 9 tem-se o diagrama de casos de uso do aplicativo móvel. Ela apresenta o ator Usuario que tem como principais casos de uso pesquisar por empresas e produtos, marca-los como favoritos e criar perguntas e comentários para os mesmos.

Figura 9 – Diagrama de casos de uso do aplicativo móvel

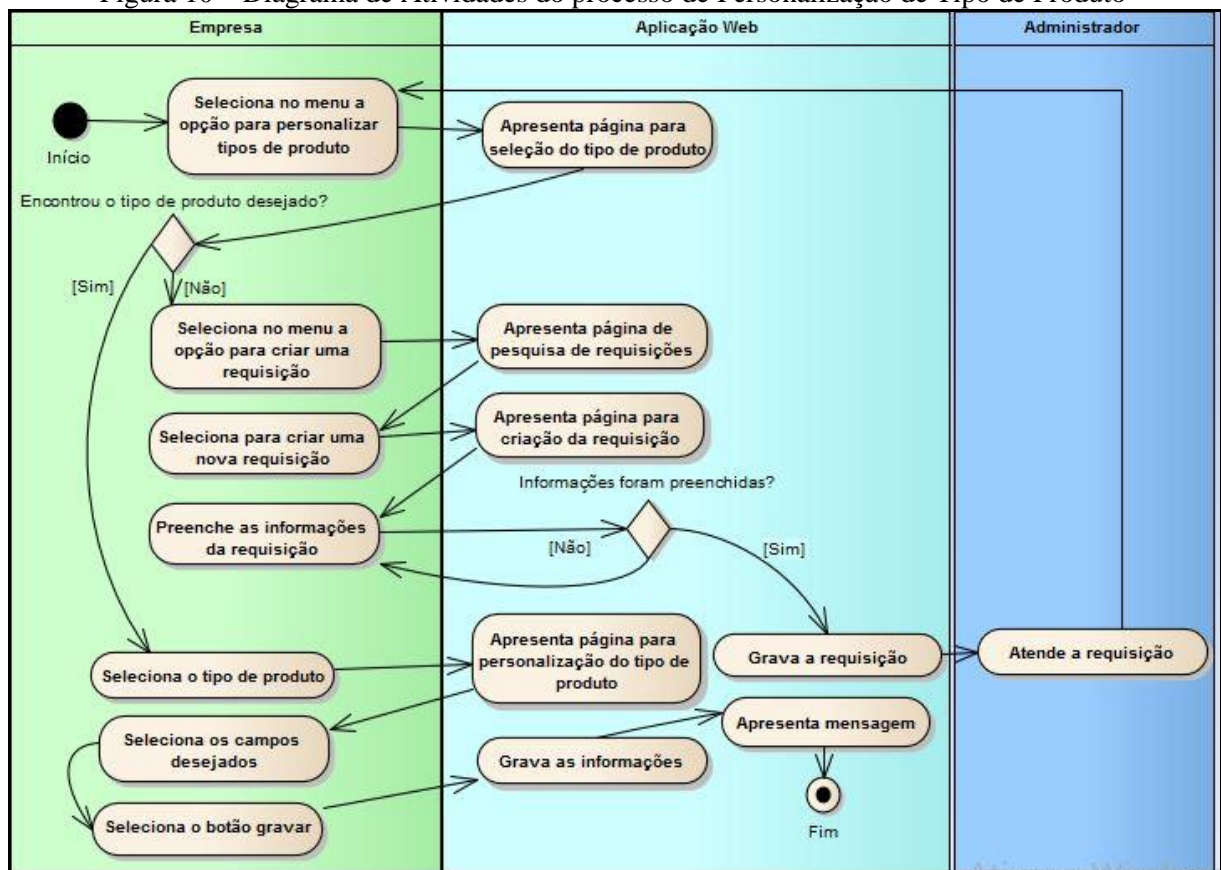


3.2.3 DIAGRAMAS DE ATIVIDADES

Nesta subseção são apresentados os diagramas de atividades necessários para entendimento das principais funcionalidades do sistema.

Na Figura 10 são demonstradas as atividades executadas pela empresa, pela aplicação web e pelo administrador no processo de personalização de tipo de produto, o qual é necessário para a empresa conseguir dar início ao cadastro de produtos. Logo após realizar a autenticação, a empresa seleciona no menu a opção referente a personalização do tipo de produto. Caso o tipo de produto requerido ainda não exista, a empresa acessa no menu a opção referente a requisições e então seleciona para criar uma nova requisição. Após informar todos os campos conforme sua necessidade, a requisição poderá ser gravada e entrará na fila do administrador para ser atendida. Quando a requisição estiver atendida a empresa pode retornar ao ponto onde pesquisa pelo tipo de produto desejado para personalização. Na página de personalização, a empresa seleciona os campos que deseja cadastrar para seu tipo de produto e seleciona a opção para gravar. Após a gravação ser feita pela aplicação será possível cadastrar produtos referentes ao tipo de produto personalizado.

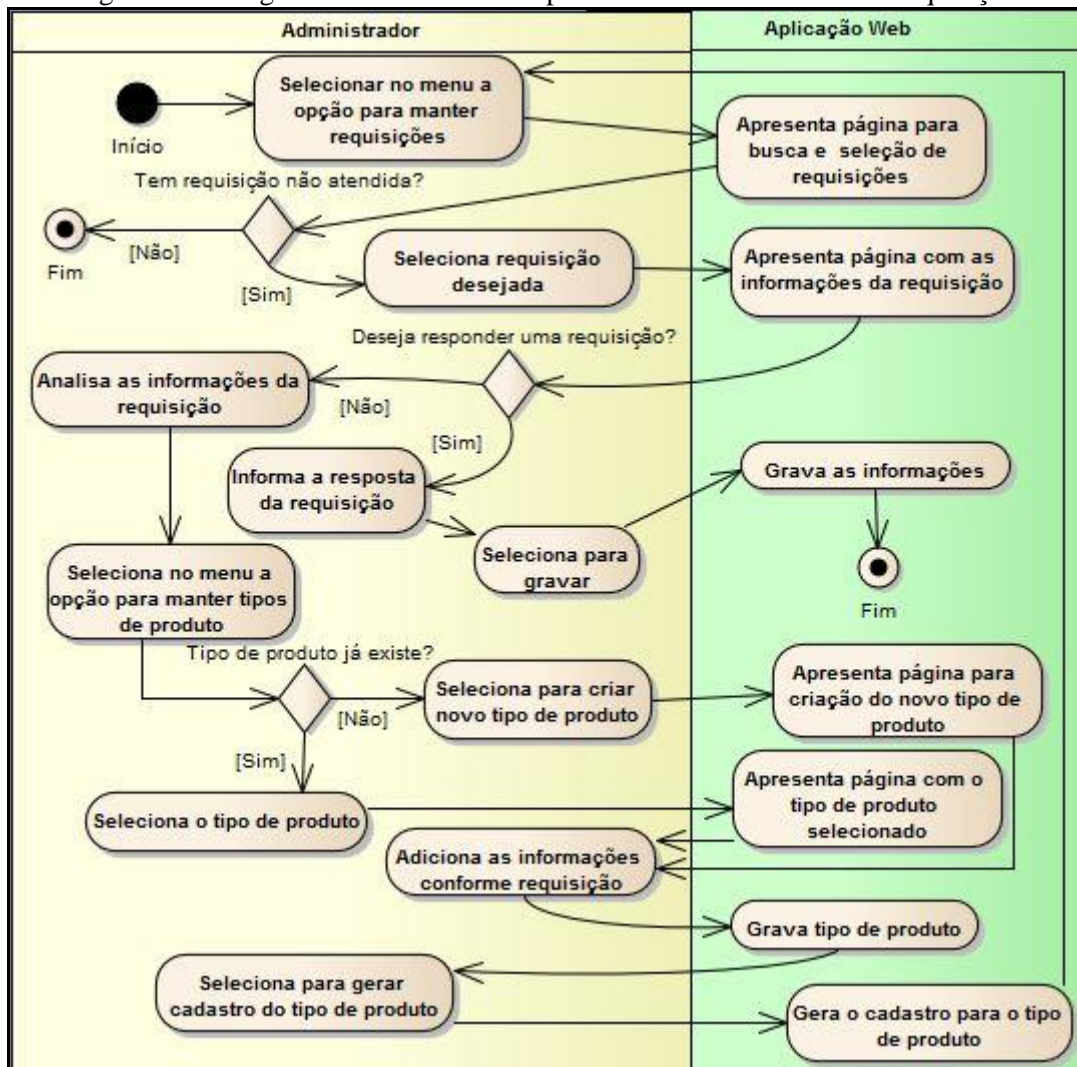
Figura 10 – Diagrama de Atividades do processo de Personalização de Tipo de Produto



Na Figura 11 são apresentadas as atividades executadas pelo administrador e da aplicação web no processo de atendimento de requisição. Logo após realizar a autenticação, o

administrador seleciona no menu a opção referente as requisições. Caso seja apresentada alguma requisição não atendida o administrador seleciona para visualização. Após analisar as informações contidas na requisição, o administrador seleciona no menu a opção referente a tipos de produto. Se o tipo de produto já foi cadastrado então seleciona para edição, senão seleciona para criar um novo cadastro. Na página de cadastro de tipo de produto, o administrador adiciona as informações conforme a requisição e então grava as informações. Para enfim disponibilizar o tipo de produto para a empresa, o administrador seleciona para gerar o cadastro do tipo de produto, deste modo a aplicação web processa as informações e apresenta mensagem de sucesso.

Figura 11 – Diagrama de Atividades do processo de Atendimento de Requisição



3.2.4 MODELO DE ENTIDADE-RELACIONAMENTO

Nesta subseção são apresentados os Modelos de Entidade Relacionamento do módulo web e do aplicativo móvel. Os dicionários de dados estão demonstrados nos Apêndices C e D.

A Figura 12 mostra o MER do módulo web. A tabela *Empresa* armazena as informações das empresas que foram cadastradas no módulo web enquanto a tabela *Usuario* armazena as informações dos usuários que foram cadastrados no aplicativo móvel.

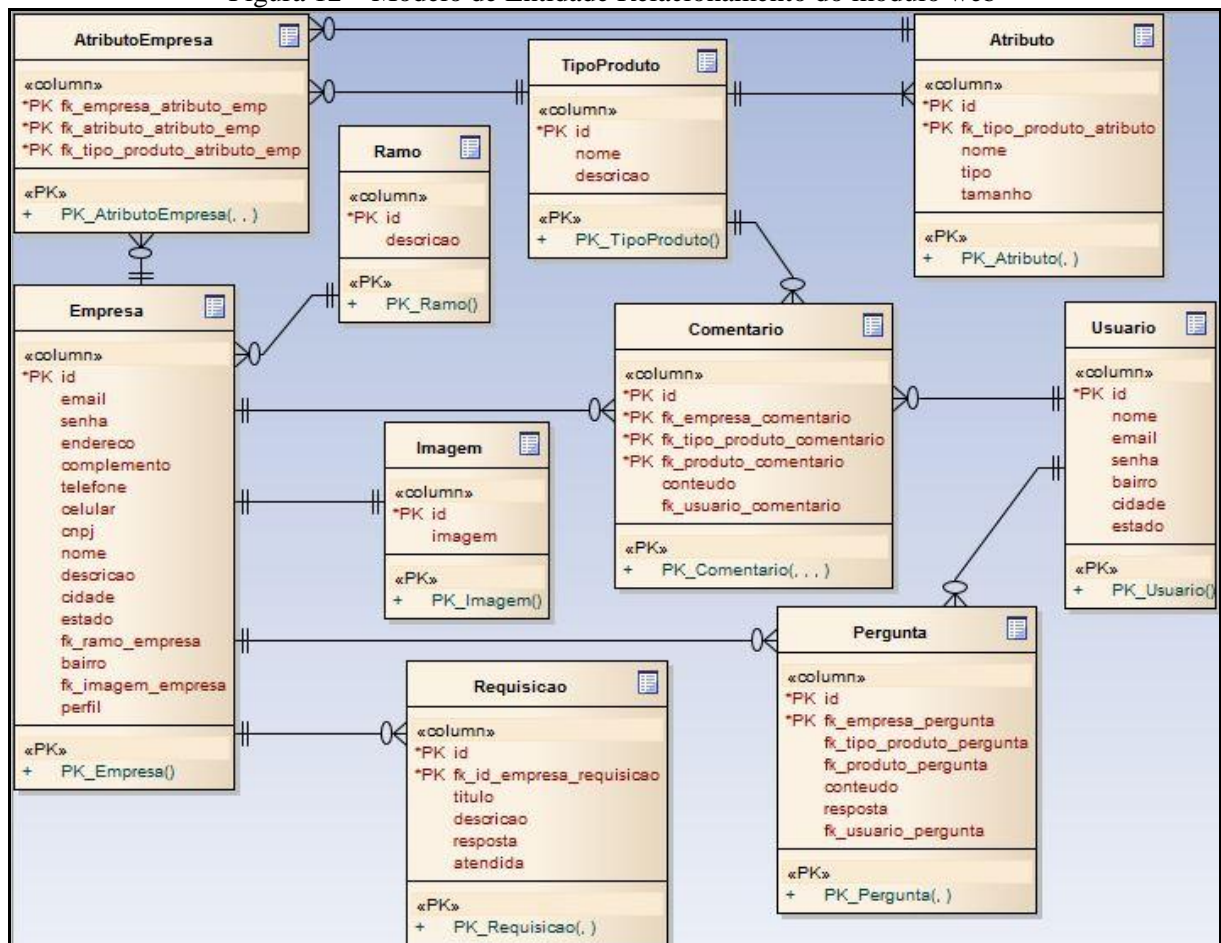
A tabela *Ramo* é utilizada para armazenar os ramos de atividades cadastrados pelo administrador para utilização no cadastro das empresas e a tabela *Imagem* é utilizada para armazenar as imagens da empresa e dos produtos cadastrados.

A tabela *Requisicao* é utilizada para armazenar as requisições criadas pelas empresas para os administradores em conjunto com a resposta do administrador para a empresa.

A tabela *TipoProduto* é utilizada para armazenar os tipos de produto cadastrados pelo administrador, enquanto a tabela *Atributo* armazena as informações das características dos tipos de produto. A tabela *AtributoEmpresa* armazena as características dos tipos de produtos personalizados pela empresa.

As tabelas *Comentario* e *Pergunta* são utilizadas para armazenar os comentários e perguntas feitos pelos usuários a partir do aplicativo móvel.

Figura 12 – Modelo de Entidade Relacionamento do módulo web

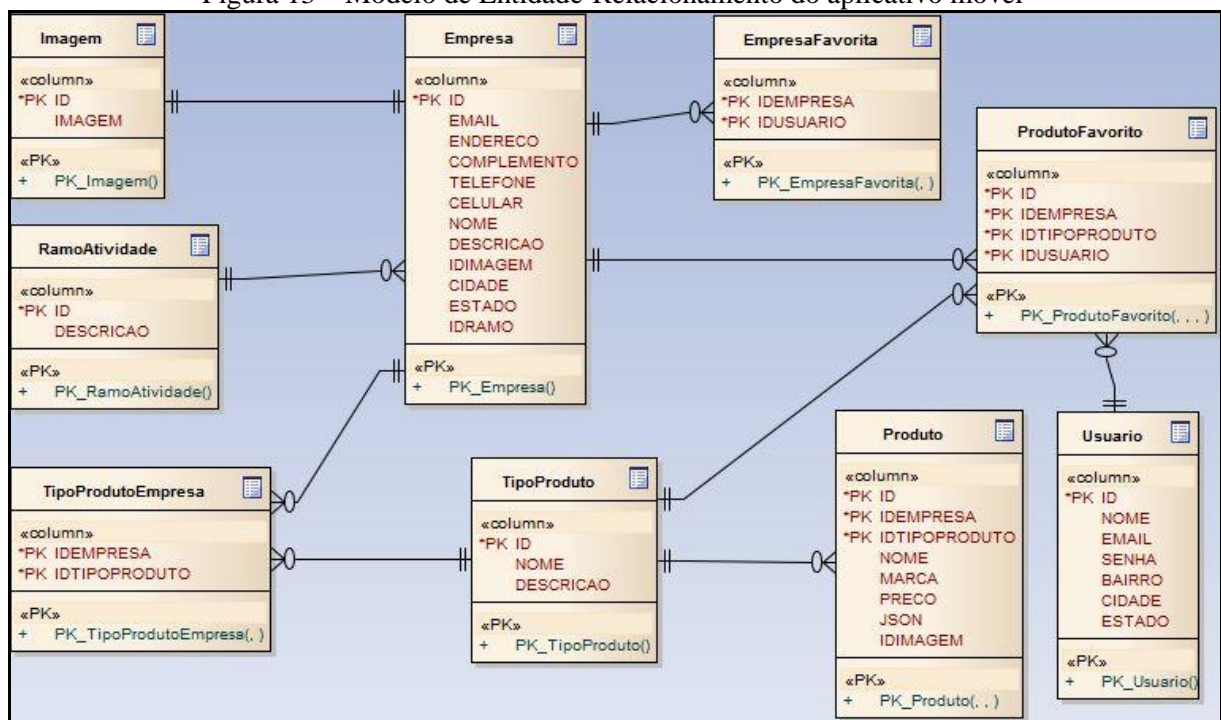


Na Figura 13 é apresentado o MER do aplicativo móvel. As tabelas *Empresa* e *Usuario* servem para armazenar as informações da empresa e do usuário. A tabela *EmpresaFavorita* é utilizada para armazenar quais são as empresas marcadas como favoritas pelo usuário. A tabela *TipoProdutoEmpresa* serve para armazenar quais são os tipos de produtos utilizados pelas empresas.

A tabela *RamoAtividade* é utilizada para armazenar os ramos de atividades das empresas armazenadas na tabela *Empresa*. Já a tabela *Imagem* serve para armazenar as imagens que são pertencentes as empresas e produtos.

A tabela *TipoProduto* é utilizada para armazenar os tipos de produtos das empresas marcadas como favoritas. A tabela *Produto* serve para armazenar os produtos marcados como favoritos e os produtos pesquisados das empresas marcadas como favoritas. A tabela *ProdutoFavorito* é utilizada para armazenar quais são os produtos marcados como favoritos.

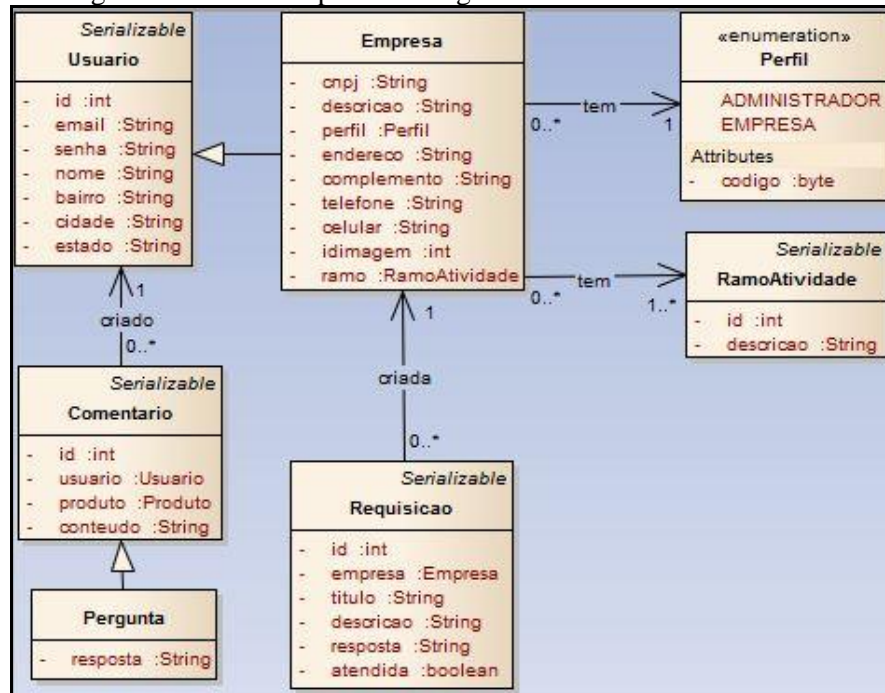
Figura 13 – Modelo de Entidade-Relacionamento do aplicativo móvel



3.2.5 DIAGRAMAS DE CLASSES

Nesta subseção são apresentados os diagramas de classes do módulo web e do aplicativo móvel. O diagrama de classes do módulo web foi separado em três partes para facilitar a visualização e o entendimento. Na Figura 14 é apresentada a primeira parte do diagrama de classes do módulo web e em seguida uma breve descrição sobre as classes envolvidas.

Figura 14 – Primeira parte do diagrama de classes do módulo web



As classes `Empresa` e `Usuario` contém informações referentes aos usuários do módulo web e da aplicação móvel. Enquanto o primeiro é pré-requisito para a utilização do módulo web o segundo é necessário para utilização do aplicativo móvel.

Para diferenciar quando é uma empresa ou um administrador acessando o sistema foi criado o *enum* `Perfil`, que é utilizado na classe `Empresa`.

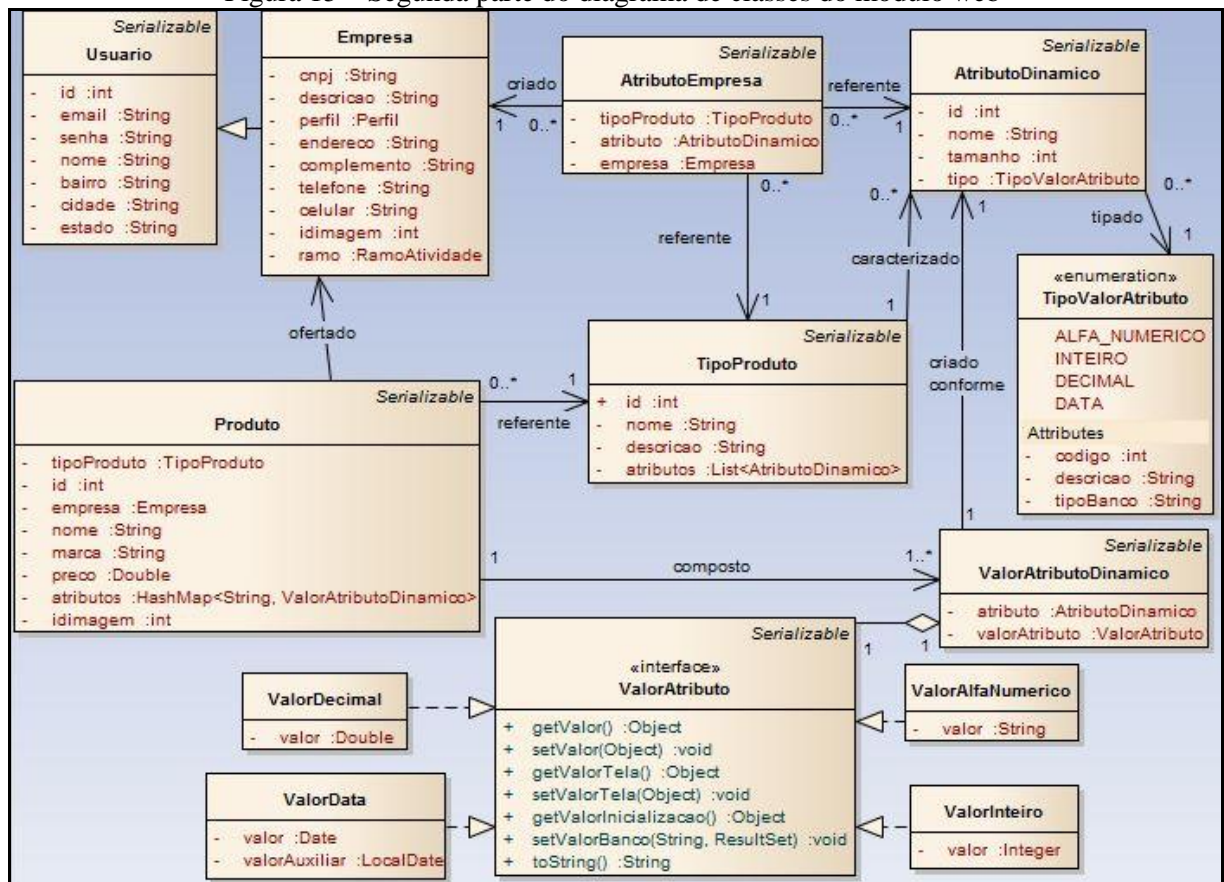
A classe `RamoAtividade` serve para armazenar o ramo de atividade da empresa.

A classe `Requisicao` é utilizada para armazenamento das informações referentes as requisições que são criadas pela empresa e respondidas pelo administrador.

As classes `Comentario` e `Pergunta` são utilizadas para carregamento dos comentários e perguntas feitos pelos usuários na aplicação móvel.

Na Figura 15 é apresentada a segunda parte do diagrama de classes do módulo web.

Figura 15 – Segunda parte do diagrama de classes do módulo web



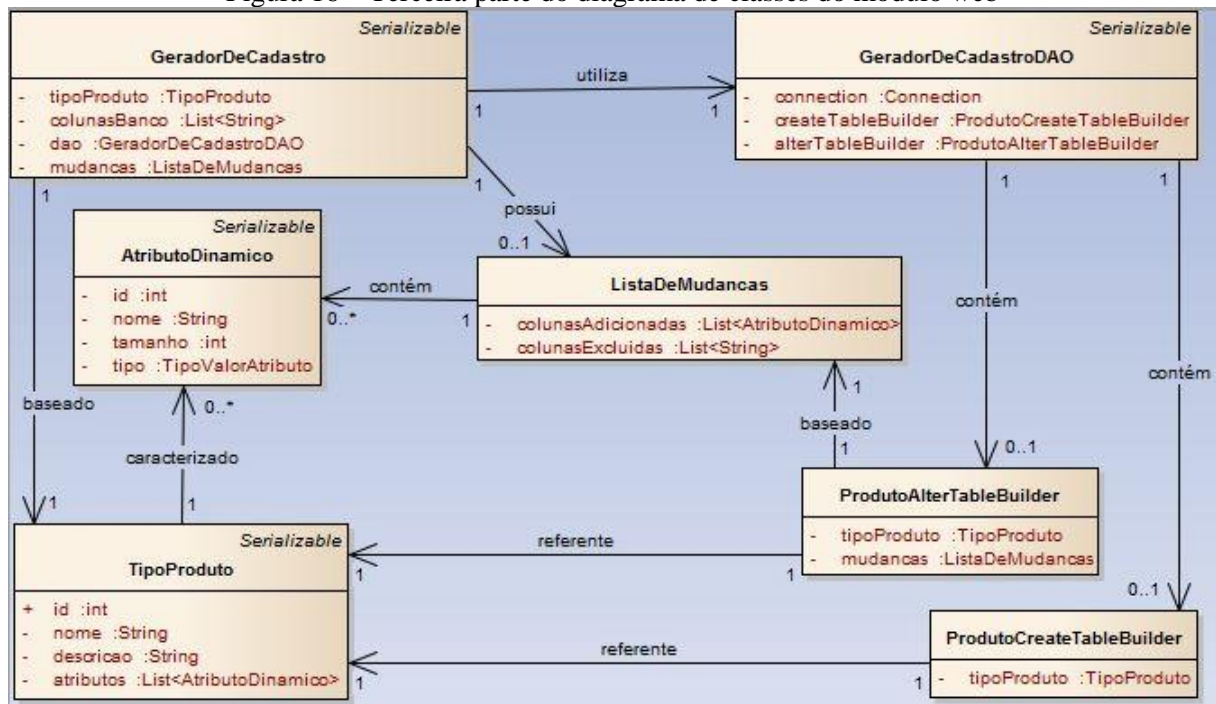
A classe `TipoProduto` contém as informações referentes a um tipo de produto que pode ser cadastrado pelo administrador. O tipo de produto tem suas características armazenadas em classes `AtributoDinamico` e o tipo de dado da característica é definido conforme o *enum* `TipoValorAtributo`.

A classe `Produto` tem as informações básicas de um produto e contém instâncias de `ValorAtributoDinamico` com os dados referentes aos seus atributos dinâmicos. Os dados de um `ValorAtributoDinamico` são armazenados utilizando a *interface* `ValorAtributo`, que é utilizada para facilitar o manuseio de diferentes tipos de dados para comunicação com as páginas web e a gravação no banco de dados. Os tipos de dados que são manuseados pela *interface* `ValorAtributo` foram criados conforme o *enum* `TipoValorAtributo` e são caracterizados pelas classes `ValorDecimal`, `ValorData`, `ValorAlfaNumerico` e `ValorInteiro`.

A classe `AtributoEmpresa` é utilizada para armazenar as informações da personalização do tipo de produto, que pode ser feita pela `Empresa` a partir do módulo web.

Na Figura 16 é apresentada a terceira parte do diagrama de classes do módulo web, que aborda a parte de geração de cadastros.

Figura 16 – Terceira parte do diagrama de classes do módulo web



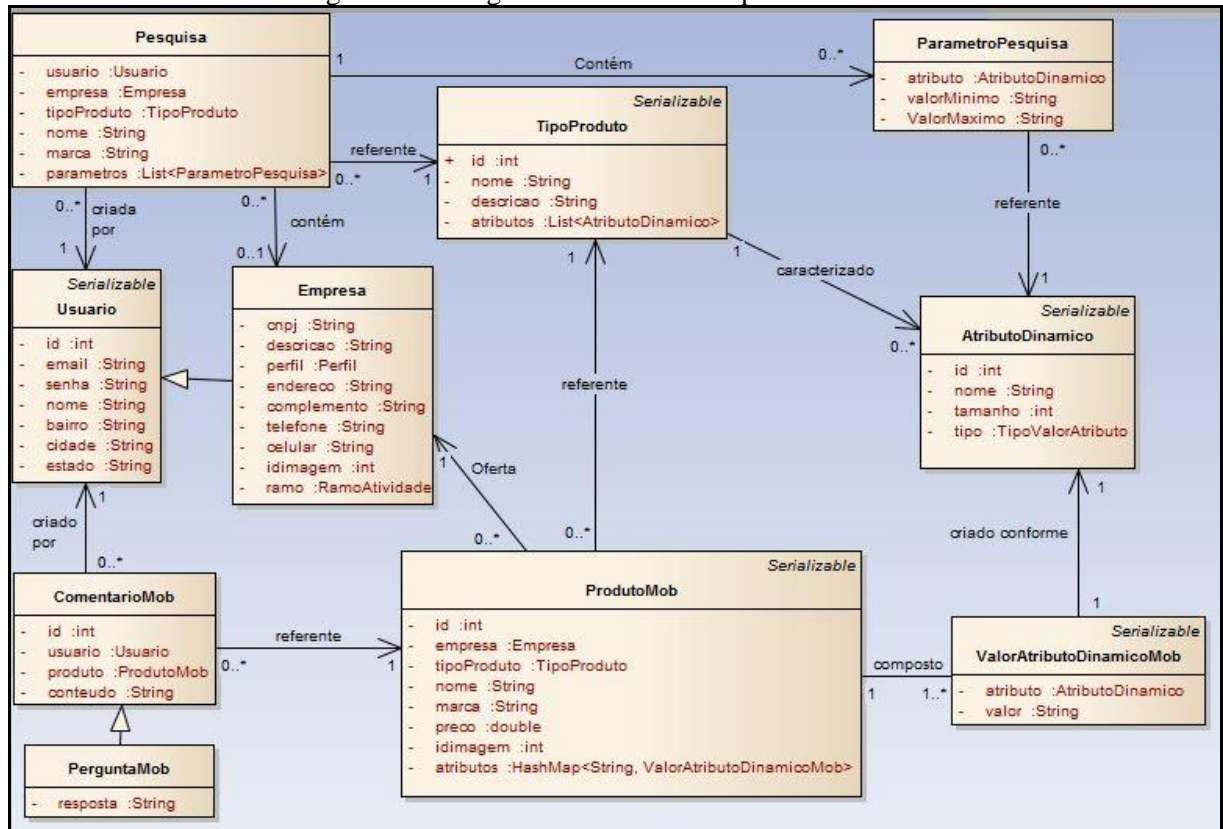
A classe `GeradorDeCadastro` é utilizada pelo sistema para processar a geração de cadastros dos tipos de produtos. Esta por sua vez faz uso da classe `GeradorDeCadastroDAO` que é utilizada para realizar as alterações no banco de dados.

As classes `ProdutoAlterTableBuilder` e `ProdutoCreateTableBuilder` são utilizadas pela classe `GeradorDeCadastroDAO` para gerar os comandos de criação e alteração dos tipos de produtos que estão tendo seu cadastro gerado.

Por fim, a classe `ListaDeMudancas` é criada pela classe `GeradorDeCadastro` e é passada para a classe `ProdutoAlterTableBuilder` ter conhecimento que quais são as alterações devem ser feitas no cadastro do tipo de produto.

Na Figura 17 é apresentado o diagrama de classes do aplicativo móvel.

Figura 17 – Diagrama de classes do aplicativo móvel



As classes Empresa, Usuario, TipoProduto e AtributoDinamico são provenientes do módulo web.

As classes ComentarioMob e PerguntaMob são utilizadas para manusear os comentários e perguntas criados pelos usuários.

As classes Pesquisa e ParametroPesquisa são utilizadas no envio das pesquisas por produtos para o *web service*.

As classes ProdutoMob e ValorAtributoMob são utilizadas para manusear os produtos carregados no aplicativo móvel.

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as técnicas e ferramentas utilizadas, além da operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

Nesta subseção são apresentadas ferramentas utilizadas, seguidas pelas técnicas utilizadas, separadas em persistência dos dados e comunicação com o servidor.

3.3.1.1 Ferramentas

No desenvolvimento do módulo web foi utilizado o *Integrated Development Environment* (IDE) Eclipse, na linguagem de programação Java. Como *framework* para construção da *User Interface* (UI) foi usado o JSF juntamente com os componentes da suíte *open-source* PrimeFaces. Para persistência dos dados o foi utilizado o SGDB MySQL.

No desenvolvimento do aplicativo móvel também foi utilizada a IDE Eclipse, porém com o *plug-in Android Developer Tools* (ADT). Já para a persistência de dados foi utilizado o SGDB SQLite, que é nativo da plataforma Android, e para a troca de dados com o *web service*, a biblioteca GSON. Para execução e depuração foi utilizado o *smartphone* Motorola G 3 com o sistema operacional Android na versão 5.1.1.

No desenvolvimento do *web service* também foi utilizada a IDE Eclipse, na linguagem de programação Java em conjunto com o *framework* Jersey e a biblioteca GSON para troca de dados com o aplicativo móvel. Assim como no módulo web, para persistência dos dados foi utilizado o SGDB MySQL.

3.3.1.2 Persistência dos dados

O armazenamento dos dados do sistema, tanto no módulo web quanto no aplicativo móvel e *web service*, foi desenvolvido tendo como objetivo separar as regras de negócio do acesso ao banco de dados. Deste modo, para cada entidade do banco de dados foi criada uma classe com os comandos e consultas, para facilitar a reutilização do código. No Quadro 5 é apresentado método inserir da classe responsável pela inserção, alteração, exclusão e consulta de requisições. O comando Structured Query Language (SQL) de inserção desta função é gerado por uma classe auxiliar, que é explicada logo a seguir.

Quadro 5 – Método de inserção de requisição

```

public void inserir(Requisicao requisicao) throws SQLException {
    String sql = queryBuilder.getInsert();

    try (PreparedStatement stmt = this.connection.prepareStatement(sql,
        Statement.RETURN_GENERATED_KEYS)) {
        stmt.setInt(1, requisicao.getEmpresa().getId());
        stmt.setString(2, requisicao.getTitulo());
        stmt.setString(3, requisicao.getDescricao());
        stmt.setString(4, requisicao.getResposta());

        if (requisicao.isAtendida())
            stmt.setInt(5, 1);
        else
            stmt.setInt(5, 0);

        stmt.execute();

        ResultSet rs = stmt.getGeneratedKeys();
        if (rs.next()) {
            requisicao.setId(rs.getInt(1));
        }
    }
}

```

Com o intuito de facilitar a manutenção dos comandos em SQL para inserção e alteração foi criada a classe `QueryBuilder`, que é utilizada no módulo web e no *web service*. Para utilizá-la basta criar uma instância na criação da classe de acesso ao banco e informar as chaves e campos da entidade que será trabalhada. Para então gerar o comando de *insert* ou *update* é necessário apenas chamar os métodos `getInsert` e `getUpdate`. No Quadro 6 é apresentada a sua criação utilizando a entidade `Atributo`. É válido mencionar que não é necessário incluir o campo de `ID` porque ele é criado como auto incremental no banco de dados.

Quadro 6 – Criação da classe `QueryBuilder`

```

private AtributoEmpresaDAO atributoEmpresaDAO;

public AtributoDinamicoDAO(TipoProduto tipoProduto) throws SQLException {
    this.connection = Database.getConnection();
    this.tipoProduto = tipoProduto;
    prepararSQLBuilder();
}

protected void prepararSQLBuilder() {
    queryBuilder = new QueryBuilder("ATRIBUTO");

    queryBuilder.adicionarChave("FK_TIPO_PRODUTO_ATRIBUTO");

    queryBuilder.adicionarCampo("NOME");
    queryBuilder.adicionarCampo("TIPO");
    queryBuilder.adicionarCampo("TAMANHO");
}

```

Na parte de *web service* são reaproveitadas as classes de acesso ao banco utilizadas no módulo web. Quando identificado que eram necessárias funcionalidades diferentes das que

estavam disponíveis, foram criadas novas classes herdando comportamentos da classe utilizada no módulo web para reutilização de código. No Quadro 7 é apresentada uma classe de acesso ao banco criada para o *web service*, porém herdando comportamentos da classe de acesso ao banco da mesma entidade utilizada no módulo web.

Quadro 7 – Classe de acesso ao banco utilizada pelo *web service*

```
public class PerguntaWsDAO extends PerguntaDAO {

    private Connection connection;

    public PerguntaWsDAO() throws SQLException {
        super();
        this.connection = Database.getConnection();
    }

    public void adicionar(PerguntaMob pergunta) throws SQLException {
        String sql = getQueryBuilder().getInsert();

        try (PreparedStatement stmt =
this.connection.prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {
            stmt.setInt(1,
pergunta.getProduto().getEmpresa().getId());
            stmt.setInt(2,
pergunta.getProduto().getTipoProduto().getId());
            stmt.setInt(3, pergunta.getProduto().getId());
            stmt.setString(4, pergunta.getConteudo());
            stmt.setString(5, pergunta.getResposta());
            stmt.setInt(6, pergunta.getUsuario().getId());
            stmt.execute();

            ResultSet rs = stmt.getGeneratedKeys();
            if (rs.next()) {
                pergunta.setId(rs.getInt(1));
            }
        }
    }
}
```

3.3.1.3 Comunicação com o servidor

Na sincronização de dados da aplicação móvel com o servidor foi utilizado um *web service* baseado na arquitetura *Representational State Transfer* (REST) com o *framework* Jersey, além da biblioteca GSON para serialização e desserialização de objetos. As classes que fazem a intermediação entre o aplicativo móvel e a base de dados foram separadas em classes representadas por entidades da aplicação, como por exemplo empresa, usuário, pergunta e comentário. Caso o leitor queira conhecer melhor as tecnologias GSON e REST, sugiro consultar GitHub (2015) e Jersey (2015).

No Quadro 8 é apresentada a classe responsável por enviar requisições de comentários para o *web service*. Para conseguir utilizá-la é necessário passar no construtor métodos que

serão executados ao receber a resposta do servidor ou caso ocorra um erro de conexão. O método `getLista` envia uma requisição para o *web service* retornar os comentários referentes a um produto. Para enviar o produto na requisição, ele é armazenado em um *JavaScript Object Notation* (JSON) logo após ser serializado utilizando a biblioteca GSON. Para criar a requisição são passados os métodos que serão executados em caso de resposta ou erro de conexão, em conjunto com o endereço do *web service* e o tipo da requisição.

Quadro 8 – Classe do aplicativo móvel responsável pelas requisições de comentários

```
public class ComentarioWS {

    private static final String TAG_COMENTARIOS_REQUEST =
"TAG_COMENTARIOS_REQUEST";
    private Listener<JSONObject> listenerResposta;
    private ErrorListener listenerErro;

    public ComentarioWS(Response.Listener<JSONObject> listenerResposta,
        Response.ErrorListener listenerErro) {
        this.listenerResposta = listenerResposta;
        this.listenerErro = listenerErro;
    }

    public void getLista(ProdutoMob produtoMob) throws JSONException {
        JSONObject obj = new JSONObject();
        obj.put("produto", new Gson().toJson(produtoMob));

        JsonObjectRequest jsonObjReq = new
JsonObjectRequest(Method.POST,
                    LinkWS.LISTAR_COMENTARIOS.getLink(), obj,
listenerResposta,
                    listenerErro);

        ApplicationController.getInstance().addToRequestQueue(jsonObjReq,
                    TAG_COMENTARIOS_REQUEST);
    }
}
```

No Quadro 9 é apresentada a classe `ComentarioResource` que faz parte do *web service* e é responsável por receber, processar e responder requisições enviadas pelo aplicativo móvel. Após receber a requisição em forma de JSON é feita a desserialização do produto enviado utilizando a biblioteca GSON. Em seguida é utilizada a classe de acesso ao banco `ComentarioWsDAO` para carregar os comentários referentes ao produto enviado pela requisição. Com a lista de comentários carregada, é feita a serialização da mesma utilizando novamente a biblioteca GSON. Para então enviar o resultado para o aplicativo móvel, a lista serializada é armazenada em um JSON e convertida para *String*.

O endereço que é utilizado pelo aplicativo móvel para acessar o *web service*, é baseado no nome do *web service* juntamente com as anotações `@Path` presentes na classe responsável pelas requisições. Ainda no Quadro 9 é possível identificar o caminho utilizado para as requisições de comentários. O método `comentarios` requer um *Uniform Resource Locator*

(URL) `http://[ip_servidor]:[porta]/WebServiceTCC/comentarios/listar`, sendo que o `[ip_servidor]` deve ser substituído pelo endereço IP do servidor e `[porta]` deve ser substituído pelo número da porta do servidor.

Quadro 9 – Classe do *web service* responsável pelas requisições de comentários

```
@Path("/comentarios")
public class ComentarioResource {

    @POST
    @Path("/listar")
    @Consumes("application/json")
    @Produces("application/json")
    public String comentarios(String info) throws NoContentException {
        try {
            JSONObject json = new JSONObject(info);

            ProdutoMob produto = new
Gson().fromJson(json.getString("produto"),
                ProdutoMob.class);

            ComentarioWsDAO dao = new ComentarioWsDAO();

            // Buscar os comentários do produto
            List<ComentarioMob> comentarios = dao.getListas(produto);

            // Preparar retorno
            JSONObject retorno = new JSONObject();
            retorno.put("result", new Gson().toJson(comentarios));

            return retorno.toString();
        } catch (SQLException e) {
            throw new NoContentException("Ocorreu um erro de
conexão!");
        } catch (JSONException e) {
            throw new NoContentException("Ocorreu um erro de envio de
dados!");
        }
    }
}
```

3.3.1.4 Geração de cadastros baseada em metadados

Para desenvolver a geração dos cadastros de tipos de produtos no módulo web foi criada a classe `GeradorDeCadastro` como base. Como pré-requisito da classe é requerido um tipo de produto, que deve ser passado logo no construtor. Após isto é necessário apenas chamar o método `gerarCadastro`.

No Quadro 10 é apresentado o método `gerarCadastro`. O primeiro passo executado por ele é verificar se o cadastro do tipo de produto já foi gerado anteriormente, fazendo uma consulta no banco de dados para verificar se a tabela já existe. Se já existir, inicialmente é carregada a lista de metadados das colunas da tabela, para que seja feita a comparação entre ela e os atributos do tipo de produto. Para identificar as colunas que devem ser adicionadas e

as que devem ser excluídas do cadastro do tipo de produto. Estas colunas identificadas são adicionadas em uma lista de mudanças e então passadas para a classe `GeradorDeCadastroDAO` utilizando o método `alterarColunasDaTabela` para processamento das alterações no banco de dados. Caso o tipo de produto ainda não tenha tabela no banco de dados é chamada a mesma classe para processar a criação da nova tabela, porém utilizando o método `adicionarTabela`.

Quadro 10 – Método utilizado para gerar o cadastro dos tipos de produto

```
public void gerarCadastro() throws SQLException, GeracaoException {
    if (dao.existeTabela(tipoProduto)) {
        colunasBanco = dao.getListasColunasTabela(tipoProduto);

        // Se a tabela do produto já existir
        mudancas = new ListaDeMudancas();

        identificarColunasAdicionadas(mudancas);
        identificarColunasExcluidas(mudancas);

        if (!mudancas.isEmpty())
            dao.alterarColunasDaTabela(tipoProduto, mudancas);
        else
            throw new GeracaoException("Geração cancelada. Nenhuma
alteração foi encontrada.");
    } else {
        // Quando a tabela do produto ainda não existe
        dao.adicionarTabela(tipoProduto);
    }
}
```

O Quadro 11 apresenta os métodos `adicionarTabela` e `alterarColunasDaTabela` da classe `GeradorDeCadastroDAO` responsáveis por adicionar e alterar as tabelas de tipos de produtos na base de dados.

Quadro 11 – Métodos para criar e alterar tabelas de tipos de produto na base de dados

```
public void adicionarTabela(TipoProduto tipoProduto) throws SQLException {
    String sql = getCreateTableBuilder(tipoProduto).getComando();

    try (PreparedStatement stmt = this.connection.prepareStatement(sql))
    {
        stmt.execute();
    }
}

public void alterarColunasDaTabela(TipoProduto tipoProduto,
    ListaDeMudancas mudancas) throws SQLException {
    String sql = getAlterTableBuilder(tipoProduto,
mudancas).getComando();

    try (PreparedStatement stmt = this.connection.prepareStatement(sql))
    {
        stmt.execute();
    }
}
```

O método `adicionarTabela` utiliza o método `getComando` da classe `ProdutoCreateTableBuilder` para processar o comando SQL que será utilizado na criação da tabela do tipo de produto no banco de dados. Já o método `alterarColunasDaTabela` utiliza o método `getComando` da classe `ProdutoAlterTableBuilder` para processar o comando SQL que será utilizado na alteração e exclusão das colunas da tabela do tipo de produto.

No Quadro 12 é apresentado o método `getComando` pertencente a classe `ProdutoCreateTableBuilder`, seguido de uma explicação.

Quadro 12 – Método responsável por gerar comandos SQL para criar tabelas de tipos de produto

```
public String getComando() {
    StringBuilder sb = new StringBuilder();
    // Comando inicial
    sb.append("CREATE TABLE ");
    // Nome Tabela
    sb.append(StrUtil.padronizar(tipoProduto.getNome()));
    // início
    sb.append("(");
    // ID como Chave
    sb.append("ID INT NOT NULL AUTO_INCREMENT, ");
    // FK Empresa e FK Imagem
    sb.append(getNomeFkEmpresa()).append(" INT NOT NULL, ");
    sb.append(getNomeFkImagem()).append(" INT NOT NULL, ");
    // Campos padrões: Nome, Marca, Preço
    sb.append("NOME VARCHAR(40) NULL,");
    sb.append("MARCA VARCHAR(60) NULL,");
    sb.append("PRECO DECIMAL(13,4) NULL,");
    // Adiciona Colunas
    for (AtributoDinamico atributo : tipoProduto.getAtributos()) {
        sb.append(StrUtil.padronizar(atributo.getNome()));
        sb.append(" ");
        sb.append(getTipoBanco(atributo.getTipo(),
atributo.getTamanho()));
        sb.append(" NULL, ");
    }
    // Adiciona ID como Primary Key
    sb.append("PRIMARY KEY (ID, ").append(getNomeFkEmpresa()).append(",
");
    // FK EMPRESA
    sb.append(" CONSTRAINT ").append(getNomeFkEmpresa());
    sb.append(" FOREIGN KEY (").append(getNomeFkEmpresa()).append(") ");
    sb.append(" REFERENCES EMPRESA (ID) ");
    sb.append(", ");
    // FK IMAGEM
    sb.append(" CONSTRAINT ").append(getNomeFkImagem());
    sb.append(" FOREIGN KEY (").append(getNomeFkImagem()).append(") ");
    sb.append(" REFERENCES IMAGEM (ID) ");
    // Fim
    sb.append(")");
    // Retorna o Select
    return sb.toString().toUpperCase();
}
```

Como são utilizadas muitas concatenações de `String` é utilizada a classe `StringBuilder` por boas práticas de programação.

Primeiramente é concatenado o comando `Create Table` correspondente à criação de uma nova tabela no banco de dados e em seguida o nome da tabela do tipo de produto. Para conseguir utilizar o nome do tipo de produto como nome da tabela é chamado o método `padronizar` da classe `StrUtil`, responsável por retirar todos os espaços, acentuações e caracteres especiais como pontos, colchetes, parênteses do nome do tipo de produto.

Após o comando inicial, é adicionado o campo `ID` como auto-incremental, além das chaves estrangeiras correspondentes à empresa e à imagem. Como no MySQL não é possível criar chaves estrangeiras com nome iguais, mesmo em tabelas diferentes, o nome delas é sempre montado concatenando `FK_ID_EMPRESA_` e `FK_ID_IMAGEM` com o nome do tipo de produto padronizado. Em seguida são adicionados os campos `nome`, `marca` e `preço` que são padrões em todas as tabelas de tipos de produto.

Para adicionar os campos específicos do tipo de produto, é feita uma iteração sobre os atributos dinâmicos. Assim como no nome da tabela, é chamado o método responsável pela padronização para processar o nome do atributo. Feito isto é concatenado o tipo do atributo dinâmico juntamente com o tamanho, quando necessário. Os nomes dos tipos são armazenados no *enum* `TipoValorAtributo` e carregados com o método `getTipoBanco`.

Com todos os campos adicionados, é necessário informar quais são as chaves primárias da tabela e referenciar as chaves estrangeiras da empresa e da imagem. Como chaves primárias é padrão para todos os tipos de produto o campo `ID` é a chave estrangeira da empresa.

No Quadro 13 é apresentado o método `getComando` pertencente a classe `ProdutoAlterTableBuilder`. Seguindo a mesma prática de concatenação apresentada no comando anterior, primeiramente é adicionado o comando `Alter Table` que é correspondente à alteração de uma tabela.

Para adicionar as novas colunas é realizada uma iteração concatenando o comando `ADD COLUMN` em conjunto com o nome e tipo do atributo dinâmico. Já para retirar as colunas excluídas é feita uma iteração concatenando o comando `DROP COLUMN` em conjunto com o nome do atributo dinâmico.

Quadro 13 - Método responsável por gerar comandos SQL para alterar tabelas de tipos de produto

```

public String getComando() {
    StringBuilder sb = new StringBuilder();
    // Comando inicial
    sb.append("ALTER TABLE ");
    // Nome Tabela
    sb.append(StrUtil.padronizar(tipoProduto.getNome())).append(" ");
    // Comandos para adicionar colunas
    List<AtributoDinamico> adicionadas =
mudancas.getColunasAdicionadas();
    // Iteração sobre as colunas adicionadas
    for (int i = 0; i < adicionadas.size(); i++) {
        AtributoDinamico coluna = adicionadas.get(i);
        // Comando para adicionar coluna
        sb.append("ADD COLUMN ");
        sb.append(StrUtil.padronizar(coluna.getNome())).append(" ");
        sb.append(getTipoBanco(coluna.getTipo(),
coluna.getTamanho())).append(" NULL");
        if (i < adicionadas.size() - 1) {
            sb.append(",");
        }
    }
    // Comandos para excluir colunas
    List<String> excluidas = mudancas.getColunasExcluidas();
    // Verifica se é necessário separador
    if (excluidas.size() > 0 && adicionadas.size() > 0) {
        sb.append(", ");
    }
    // Adiciona comandos para excluir colunas
    for (int i = 0; i < excluidas.size(); i++) {
        // Comando para excluir coluna
        sb.append(" DROP COLUMN
").append(StrUtil.padronizar(excluidas.get(i)));
        if (i < excluidas.size() - 1) {
            sb.append(",");
        }
    }
    return sb.toString();
}

```

3.3.2 Operacionalidade da implementação

Nesta subseção é apresentada a operacionalidade do sistema desenvolvido, separada pelas funcionalidades do administrador e da empresa no módulo web bem como do usuário no aplicativo móvel.

3.3.2.1 Módulo Web

Na Figura 18 é apresentada a página que permite efetuar a autenticação, utilizando e-mail e senha, além da opção de cadastrar uma nova empresa.

Figura 18 – Tela de autenticação do módulo web

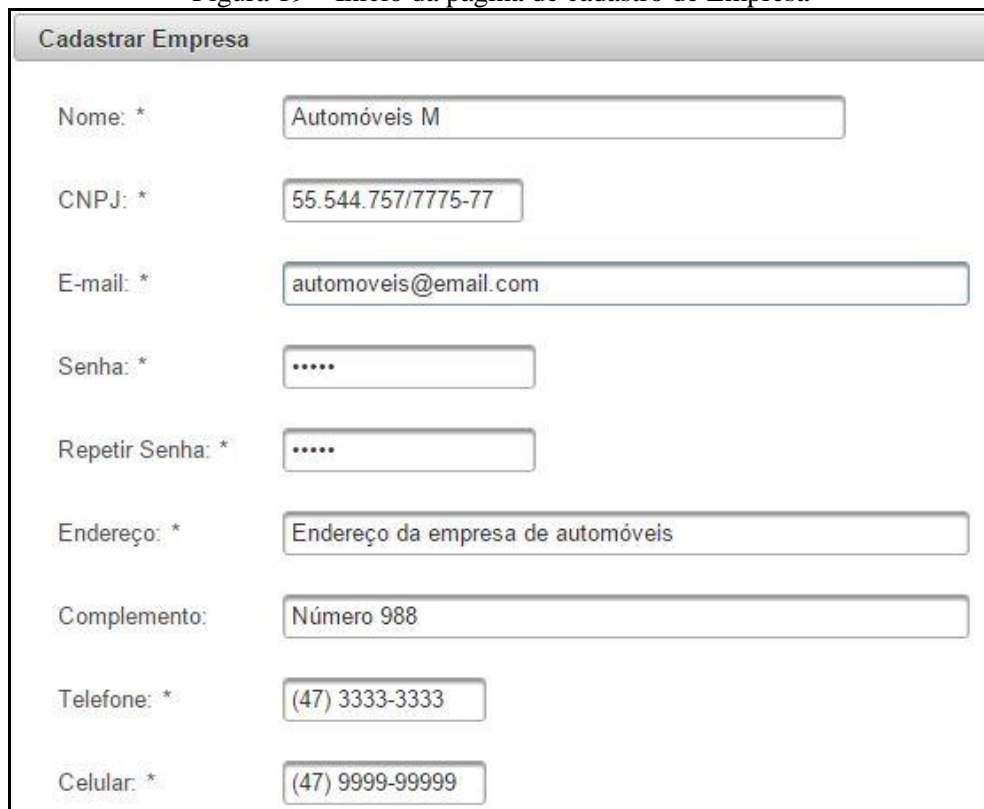


Tela de autenticação do módulo web. O formulário contém os seguintes elementos:

- Label: E-mail: *
- Input de texto para o e-mail.
- Label: Senha: *
- Input de texto para a senha.
- Botão: Entrar
- Link: [Novo por aqui?](#)

Na Figura 19 é apresentado o início da página de cadastro de Empresa, da qual é necessário informar nome, Cadastro Nacional da Pessoa Jurídica (CNPJ), e-mail, senha, endereço, complemento, telefone, celular, bairro, cidade, estado, ramo de atividade, descrição da empresa e imagem. Após efetuar o cadastro a página será redirecionada para a página inicial da aplicação, sem necessidade de efetuar a autenticação. A página utilizada para manter a empresa contém os mesmos campos e informações apresentados na página de cadastro.

Figura 19 – Início da página de cadastro de Empresa



Cadastrar Empresa

Nome: *	Automóveis M
CNPJ: *	55.544.757/7775-77
E-mail: *	automoveis@email.com
Senha: *
Repetir Senha: *
Endereço: *	Endereço da empresa de automóveis
Complemento:	Número 988
Telefone: *	(47) 3333-3333
Celular: *	(47) 9999-99999

Na Figura 20 é apresentado o menu completo do módulo web, separado pelas seções Geral, Produto, Administrador e Outros.

Figura 20 – Menu completo do módulo web



Na seção Geral, a opção Minha Empresa permite que sejam alteradas informações de cadastro da empresa enquanto a opção Requisições permite a empresa criar requisições para o administrador e permite o administrador responder as requisições criadas pelas empresas.

Na seção Produto, a opção Personalizar Tipo de Produto permite a empresa personalizar as características dos tipos de produtos disponíveis para cadastro, a opção Meus Produtos permite a empresa manter seus produtos e a opção Perguntas e Comentários permite a empresa visualizar os comentários e responder as perguntas feitas para seus produtos a partir do aplicativo móvel.

Na seção Administrador, as opções disponíveis permitem manter o cadastro de empresas, tipos de produto, usuários e ramos de atividade.

Na seção Outros, a opção Sair permite que seja feito o *logout* da aplicação.

Na Figura 21 é apresentada a página utilizada para a empresa e o administrador manterem uma requisição. Para que seja possível a empresa criar uma requisição para o administrador, é necessário informar título e descrição detalhada da necessidade. Após o

administrador responder a requisição não será mais possível excluí-la e a opção de marcar a requisição como atendida ficará habilitada para a empresa finalizá-la.

Figura 21 – Requisição atendida pelo administrador

Gravar Excluir ✓ Requisição Atendida! Voltar

Titulo da Requisição

Cadastrar Carros

Descrição da Requisição

Boa tarde.
Preciso que seja possível cadastrar carros no sistema!
As informações necessárias são: Câmbio, Combustível, Ar condicionado, Alarme, Banco com

175 caracteres restantes.

Resposta para Requisição

O carro foi adicionado para cadastro, qualquer dúvida entre em contato.

329 caracteres restantes.

3.3.2.1.1 Administrador

As principais funcionalidades do administrador são baseadas em manter os tipos de produto, gerar o cadastro dos tipos de produto, manter os ramos de atividade e responder as requisições feitas pela empresa. Para conseguir lidar com situações desconfortáveis, como por exemplo nomes ou endereços indevidos, foram disponibilizadas páginas para o administrador manter as empresas e usuários.

Na Figura 22 é apresentada a página de tipos de produto. Para cadastrar um tipo de produto é necessário informar nome e descrição, além de seus atributos que são definidos

por nome, tipo e tamanho. Após todas as informações terem sido informadas, o administrador deve gravar o tipo de produto e então poderá gerar o cadastro do mesmo. Após este procedimento o tipo de produto estará disponível para as empresas.

Figura 22 – Página de cadastro de tipos de produto

The screenshot shows a web interface for registering product types. At the top, there are buttons for 'Gravar', 'Gerar Cadastro', and 'Voltar'. The main form is divided into two sections: 'Tipo de Produto' and 'Atributos'.

Tipo de Produto:

- Nome: *
- Descrição: *

Atributos:

- Nome: *
- Tipo:
- Tamanho: *
-

Below the form is a table listing existing product types:

Nome	Tipo	Tamanho	Ação
Câmbio	Alfanumérico	40	Remover
Combustível	Alfanumérico	40	Remover
Ar Condicionado	Alfanumérico	1	Remover

Na Figura 23 é apresentada a página utilizada para manter os ramos de atividades disponíveis para a empresa selecionar em seu cadastro.

Figura 23 – Página para manter ramos de atividade

The screenshot shows a web interface for maintaining activity branches. At the top, there are buttons for 'Gravar' and 'Voltar'. The main form is titled 'Ramo de atividade' and contains a single field:

- Descrição: *

Na Figura 24 é apresentada a página utilizada para manter os usuários, que são cadastrados a partir do aplicativo móvel.

Figura 24 – Página para manter usuários

3.3.2.1.2 Empresa

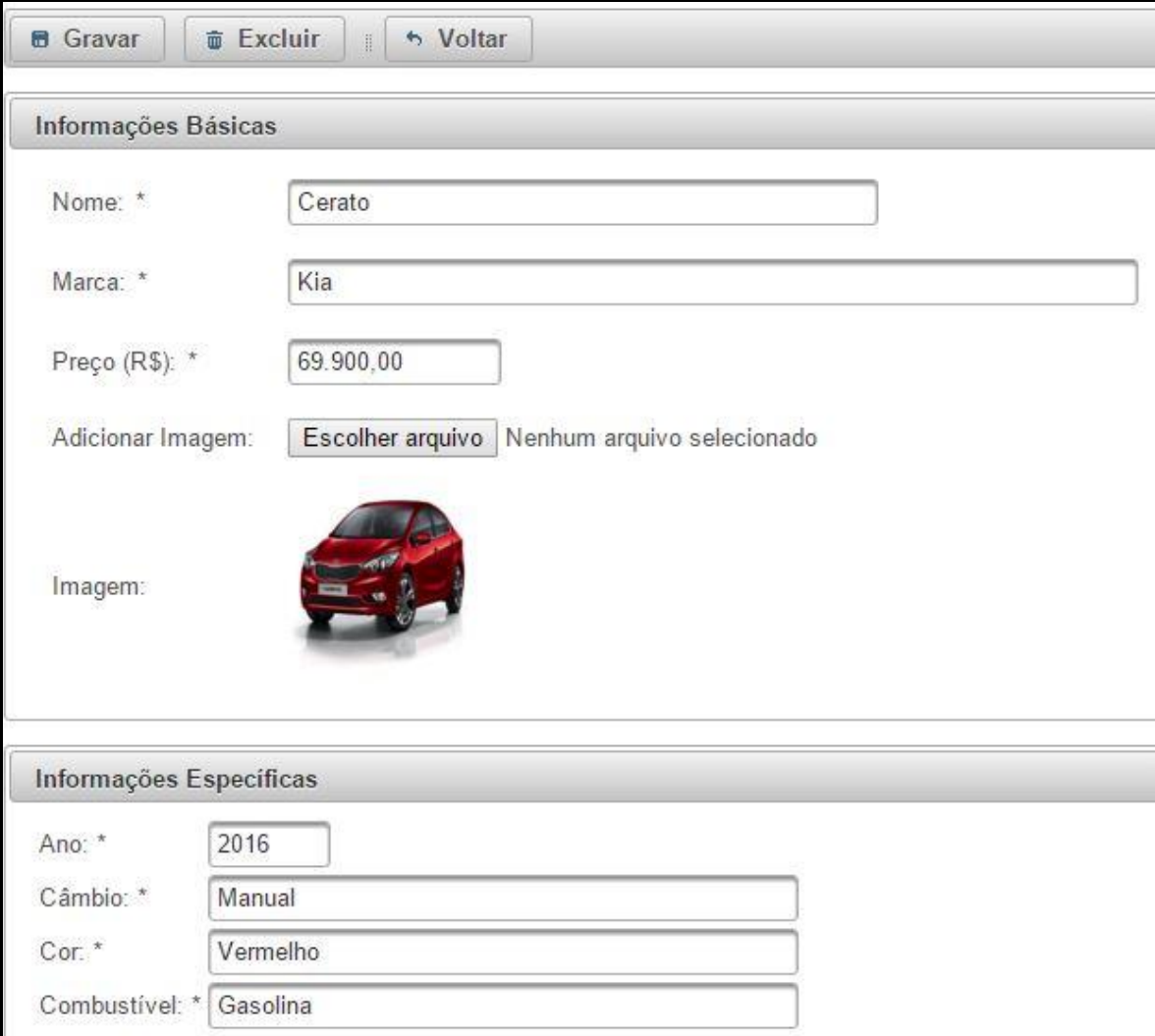
As funcionalidades da empresa são baseadas em manter o seu cadastro, criar requisições para o administrador, personalizar os tipos de produto, cadastrar produtos, visualizar comentários e responder perguntas feitos a partir do aplicativo móvel.

Na Figura 25 é apresentada a página utilizada pela empresa para personalizar os tipos de produtos. Para selecionar as características que estão disponíveis para serem informadas no cadastro do produto, a empresa deve colocá-las no painel direito e então selecionar para gravar.

Figura 25 – Página para personalizar tipos de produto

Na Figura 26 é apresentada a página responsável por manter produtos. No primeiro painel nomeado de informações básicas estão presentes as características que são comuns em todos os tipos de produto, são elas nome, marca, preço e imagem. No segundo painel, nomeado de informações específicas estão presentes as características do tipo de produto que foram selecionadas a partir da página de personalização de tipos de produto. Caso a empresa tente realizar a gravação do produto e algum campo não tenha sido informado, será apresentada uma mensagem indicando o campo faltante e a gravação não será completada.

Figura 26 – Página para manter produtos



The screenshot shows a web interface for managing products. At the top, there are three buttons: 'Gravar' (Save), 'Excluir' (Delete), and 'Voltar' (Back). Below this is a section titled 'Informações Básicas' (Basic Information) with the following fields:

- Nome: * (Name): Cerato
- Marca: * (Brand): Kia
- Preço (R\$): * (Price): 69.900,00
- Adicionar Imagem: (Add Image): Escolher arquivo (Choose file) Nenhum arquivo selecionado (No file selected)
- Imagem: (Image): A red car image.

Below the basic information is a section titled 'Informações Específicas' (Specific Information) with the following fields:

- Ano: * (Year): 2016
- Câmbio: * (Gear): Manual
- Cor: * (Color): Vermelho (Red)
- Combustível: * (Fuel): Gasolina (Gasoline)

Na Figura 27 é apresentada a página responsável pela visualização das perguntas e comentários. Além de permitir filtrar as informações pelos campos presentes da tabela, ela permite que a empresa responda as perguntas. Para isso é necessário apenas selecionar a pergunta e então será aberta uma tela com um campo para preenchimento.

Figura 27 – Página para visualizar perguntas e comentários

Perguntas				
Pergunta	Resposta	Produto	Tipo do Produto	Nome do Usuário
Banco de couro está incluso no preço?	Sim, está incluso.	Honda Civic	Carro	Ney Rubens
Quantos quilômetros faz por litro?	Aguardando resposta.	Honda Civic	Carro	Marco Aurélio Rutzen

Observação: Clique na pergunta desejada para responder!

Comentários			
Comentário	Produto	Tipo do Produto	Nome do Usuário
Um dos melhores em conforto e economia.	Honda Civic	Carro	Ney Rubens
Carro muito bom!!	Honda Civic	Carro	Marco Aurélio Rutzen

3.3.2.2 Aplicativo Móvel

A seguir serão apresentadas as funcionalidades do aplicativo móvel desenvolvido. Elas foram construídas para serem utilizadas por usuários, que são de modo geral pessoas físicas. No aplicativo é permitido que sejam feitas pesquisas por empresas e produtos, além da possibilidade de marcá-los como favoritos para acesso *off-line*.

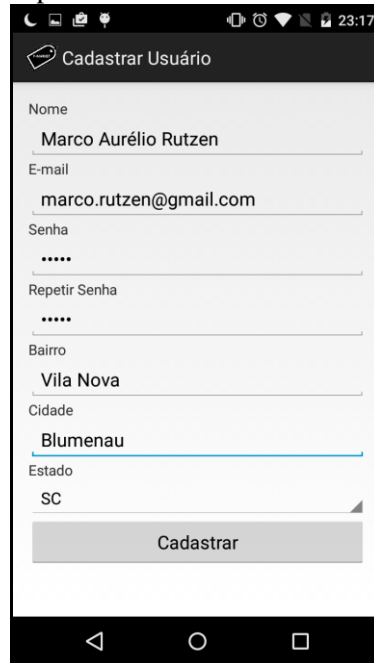
Na Figura 28 é apresentada a tela de autenticação do aplicativo móvel. Além da autenticação ela disponibiliza a opção para realizar um novo cadastro. Caso o usuário já tenha utilizado o aplicativo é possível realizar a autenticação *off-line*, deste modo será possível visualizar as empresas e produtos que foram marcados como favoritos sem conexão com a internet.

Figura 28 – Tela de autenticação do aplicativo móvel



Na Figura 29 é apresentada a tela de cadastro de usuário. Nesta tela é necessário informar os campos nome, e-mail, senha, bairro, cidade e estado. Para manter o cadastro do usuário é utilizada uma tela com os mesmos campos e, com exceção da senha, todos os campos são obrigatórios.

Figura 29 – Tela para cadastro de usuário do aplicativo móvel



The screenshot shows a mobile application interface for user registration. The title bar at the top reads "Cadastrar Usuário". Below the title, there are several input fields: "Nome" (filled with "Marco Aurélio Rutzen"), "E-mail" (filled with "marco.rutzen@gmail.com"), "Senha" (filled with "*****"), "Repetir Senha" (filled with "*****"), "Bairro" (filled with "Vila Nova"), "Cidade" (filled with "Blumenau"), and "Estado" (filled with "SC"). A "Cadastrar" button is located at the bottom of the form. The Android navigation bar is visible at the very bottom.

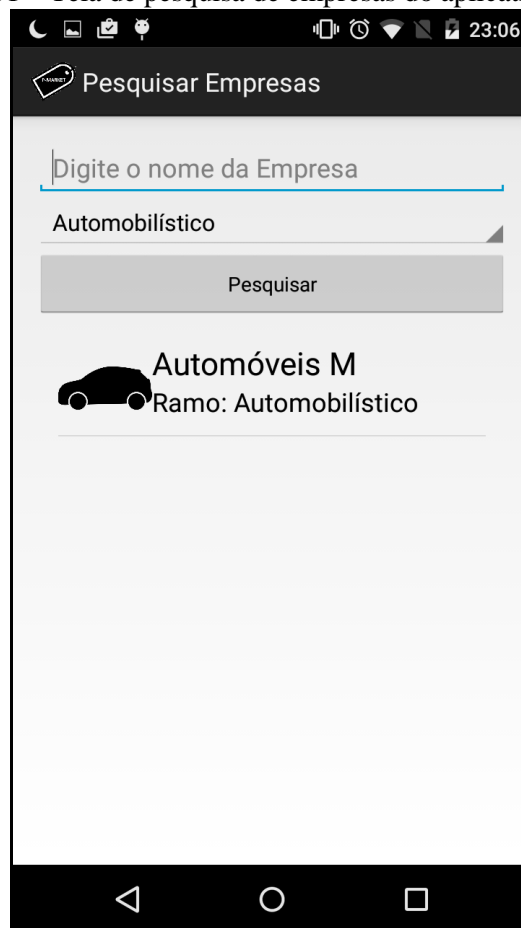
Na Figura 30 é apresentada a tela do menu principal do aplicativo. A partir dela é possível iniciar uma pesquisa por empresa ou por produtos, selecionar para visualizar as empresas e produtos favoritos, alterar as informações de cadastro do usuário e voltar para a tela de autenticação.

Figura 30 – Tela do Menu Principal do aplicativo móvel



Na Figura 31 é apresentada a tela de pesquisa de empresas. Nela é possível pesquisar pelo nome da empresa e é necessário informar o ramo de atividade desejado. A pesquisa irá listar primeiramente as empresas que se encontram no mesmo bairro, cidade e estado conforme o cadastro do usuário. Após o resultado ser apresentado, o usuário deve selecionar a empresa desejada e então as informações da empresa serão apresentadas.

Figura 31 – Tela de pesquisa de empresas do aplicativo móvel



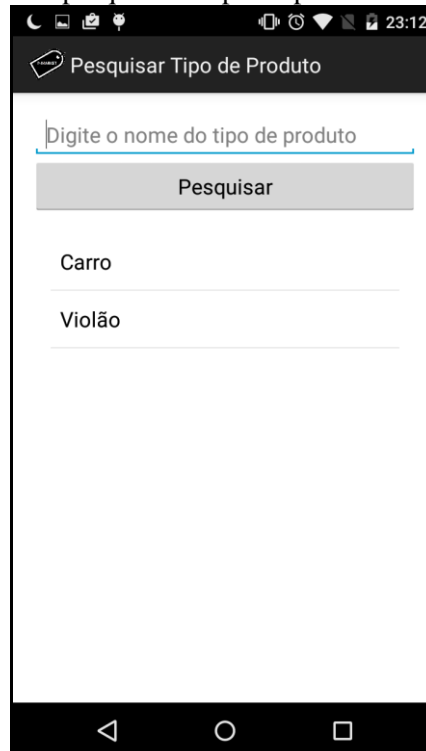
Na Figura 32 é apresentada a tela de informações da empresa. Esta tela pode ser chamada a partir da lista de favoritos, da tela de pesquisa de empresas ou da tela de informações do produto. Nesta tela são apresentadas as informações de nome, ramo de atividade, endereço, telefone, celular, uma breve descrição sobre a empresa e uma imagem. Além disto, é possível visualizar os produtos e marcar a empresa como favorita, selecionando a estrela que está logo ao lado da imagem da empresa.

Figura 32 – Tela de informações da empresa do aplicativo móvel



Na Figura 33 é apresentada a tela de pesquisa de tipo de produto. Ela é utilizada no processo de pesquisa de produtos, onde é necessário selecionar um tipo de produto desejado para que então seja apresentada a tela de pesquisa de produtos.

Figura 33 – Tela de pesquisa de tipo de produto do aplicativo móvel



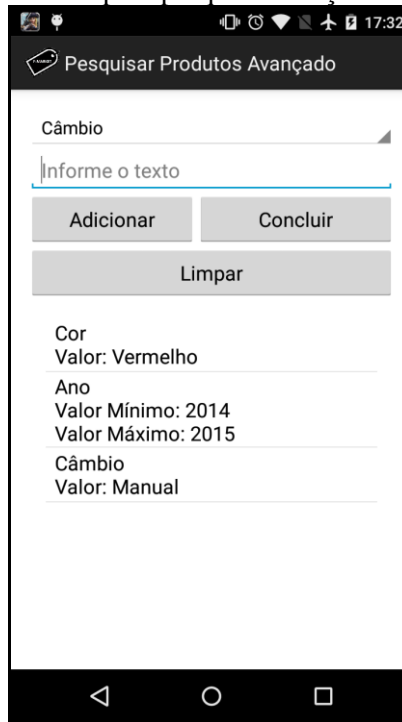
Na Figura 34 é apresentada a tela de pesquisa de produtos. Nela é possível realizar a pesquisa por produtos pelo nome e marca do produto. Caso apenas estes campos não satisfaçam as necessidades do usuário, é possível selecionar para realizar uma pesquisa avançada, onde é possível aplicar um filtro mais específico na pesquisa. Após selecionar um produto na lista, será apresentada a tela com as informações do produto.

Figura 34 – Tela de pesquisa de produtos do aplicativo móvel



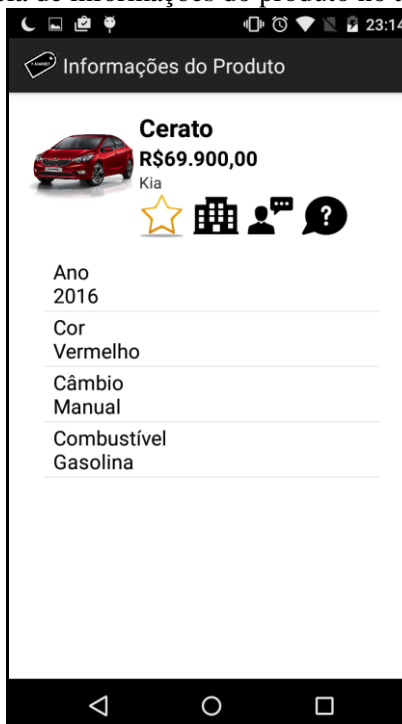
Na Figura 35 é apresentada a tela de pesquisa de produtos avançada. Ela permite que a pesquisa seja filtrada pelas características do tipo de produto. Primeiramente deve-se informar a característica desejada, se ela for referente a um texto será disponibilizado apenas um campo para informar o valor, caso contrário serão apresentados dois campos, um para o valor inicial e outro para o valor final. Para apresentar o resultado é necessário clicar em concluir, que então será apresentada a tela de pesquisa de produtos novamente, com os devidos produtos filtrados.

Figura 35 – Tela para pesquisa avançada de produtos



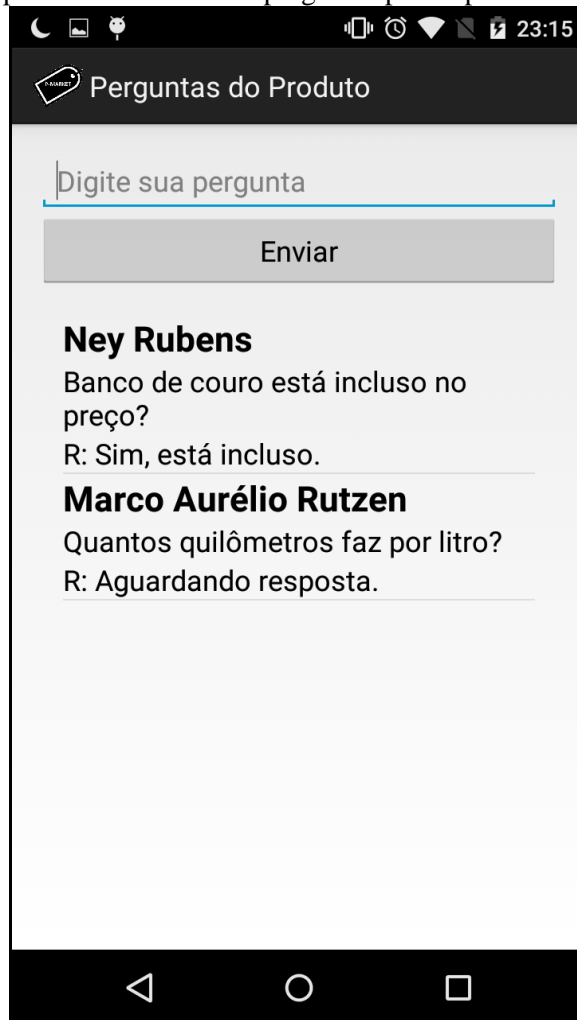
Na Figura 36 é apresentada a tela de informações do produto. Nela são mostradas todas as informações personalizadas e cadastradas pela empresa a partir do módulo web. Além disso é possível marcar o produto como favorito, selecionar para visualizar as informações da empresa a qual o produto pertence e selecionar para visualizar e criar comentários e perguntas para o produto.

Figura 36 – Tela de informações do produto no aplicativo móvel



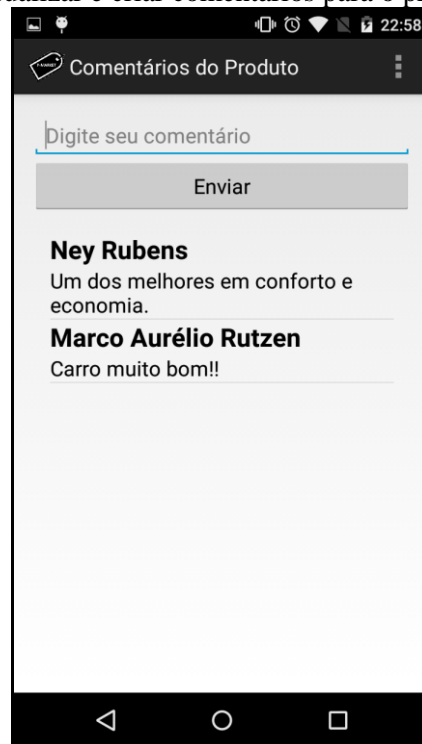
Na Figura 37 é apresentada a tela para visualizar e criar comentários para o produto. Após inserir a pergunta desejada e enviar, a empresa consegue visualizá-la a partir do módulo web e então respondê-la.

Figura 37 – Tela para visualizar e criar perguntas para o produto no aplicativo móvel



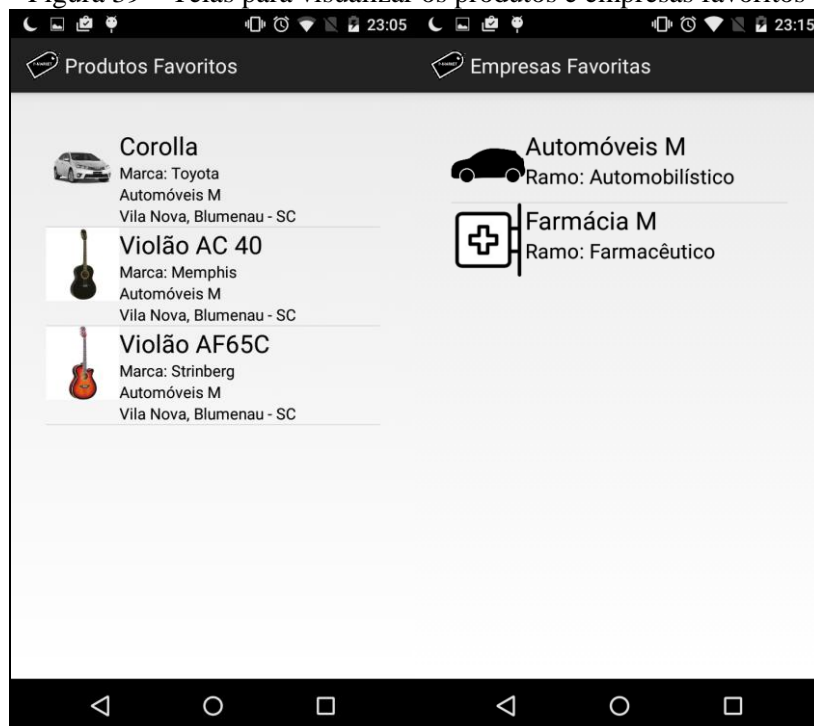
Na Figura 38 é apresentada a tela para visualizar e criar comentários para o produto. Após criar o comentário, a empresa consegue visualizá-lo a partir do módulo web e ficar a par das opiniões dos usuários referentes ao produto.

Figura 38 – Tela para visualizar e criar comentários para o produto no aplicativo móvel



Na Figura 39 são apresentadas as telas para visualizar produtos e empresas favoritos. Quando é feita a autenticação utilizando o modo *off-line* ou não há conexão com a internet ainda é possível acessar estas telas e visualizar os produtos e empresas marcados como favoritos, os quais ficaram armazenados desde a última visualização.

Figura 39 – Telas para visualizar os produtos e empresas favoritos



3.4 RESULTADOS E DISCUSSÕES

Este trabalho teve como objetivo desenvolver aplicativo móvel para divulgação de produtos e empresas, tendo como complemento o desenvolvimento de um módulo web com facilidade de manutenção utilizando conceitos de metadados e funcionalidades para as empresas cadastrarem seus produtos e informações.

O aplicativo móvel, além de oferecer recursos para pesquisa de produtos e empresas, permite que estes sejam marcados como favoritos para serem acessados mesmo sem conexão com a internet e permite criar perguntas e comentários para os produtos. Quando realizadas pesquisas, primeiro são apresentados os resultados conforme o cadastro do usuário, considerando bairro, cidade e estado.

O módulo web oferece à empresa recursos para cadastro de seus produtos, com a opção de personalizar quais são as informações dos tipos de produto que ela deseja informar. Além disso é possível visualizar os comentários e responder perguntas feitas pelos seus usuários para seus produtos.

O módulo web oferece ao administrador recursos para gerar o cadastro de produtos dos sem precisar realizar alterações no código fonte e no banco de dados. Também é disponibilizado um meio para atender requisições das empresas, que podem pedir alterações nos tipos de produtos e a geração do cadastro de novos tipos de produtos.

3.4.1 COMPARATIVO DO APLICATIVO MÓVEL

No Quadro 14 é apresentado um comparativo entre as principais funcionalidades do aplicativo móvel desenvolvido e os trabalhos correlatos Buscapé (2015) e Guia Fácil (2015).

Quadro 14 – Comparativo com trabalhos correlatos do aplicativo móvel desenvolvido

Funcionalidade	Buscapé (2015)	Guia Fácil (2015)	P-Market
Pesquisar por Empresas	Não	Sim	Sim
Pesquisar por Produtos	Sim	Não	Sim
Pesquisar por produtos de uma determinada empresa	Não	Não	Sim
Pesquisas apresentam resultados conforme localização informada.	Não	Sim	Sim
Pesquisas por ramo de atividade da empresa	Não	Sim	Sim
Pesquisas por tipo de produto	Sim	Não	Sim
Acesso <i>off-line</i>	Não	Não	Sim
Marcar empresa como favorita	Não	Não	Sim
Marcar produto como favorito	Sim	Não	Sim
Criar perguntas para os produtos	Não	Não	Sim
Criar comentários para os produtos	Sim	Não	Sim
Avaliar produtos	Sim	Não	Não
Comparação de preços	Sim	Não	Não
Informações da empresa	Não	Sim	Sim

O aplicativo Buscapé (2015) tem um foco maior em produtos e comparação de preços independente da região do usuário, possibilita a pesquisa por tipos de produto, realizar comentários e avaliar produtos, além de conseguir marcar o produto como favorito.

O aplicativo Guia Fácil (2015) tem um foco maior em disponibilizar informações de empresas conforme a cidade selecionada pelo usuário, além de permitir escolher qual é o ramo de atividade da empresa, como farmácias, restaurantes, lanches, floricultura, etc.

O aplicativo P-Market, desenvolvido neste trabalho, possui a maioria das funcionalidades presentes nos dois aplicativos citados, mas também possibilita o acesso *off-line* dos favoritos, pesquisas por produtos apenas de uma determinada empresa, criação de perguntas referentes aos produtos e marcação de empresas como favoritas.

3.4.2 COMPARATIVO DO MÓDULO WEB

No Quadro 15 é apresentado um comparativo entre a geração de cadastros do módulo web desenvolvido e a ferramenta para geração de código para operações CRUD presente no trabalho correlato do Felipe Gabriel Arend (2011).

Quadro 15 – Comparativo com trabalhos correlatos do módulo web desenvolvido

Funcionalidade	Ferramenta desenvolvida por Arend (2011)	P-Market
Geração dos cadastros baseada em metadados	Sim	Sim
Geração dos cadastros baseia-se em que tipo de dado	Entidades da base de dados	Tipos de produto cadastrados no sistema
Geração dos cadastros cria arquivos nos diretórios do sistema	Sim	Não
Forma de apresentação dos dados	Para cada entidade é gerada uma página	Apresenta sempre a mesma página, mas com os campos gerados conforme o produto
Tipo de cadastros que são gerados	Qualquer entidade do banco de dados	Baseado em produtos

A ferramenta desenvolvida por Arend (2011) realiza a geração dos cadastros do sistema baseando-se nas entidades presentes no banco de dados, obtendo os metadados da base informada. Para cada entidade a geração cria arquivos nas pastas do sistema e após iniciar o sistema estes cadastros estarão disponíveis no menu da aplicação.

A parte de geração de cadastro do P-Market desenvolvida por este trabalho baseia-se no cadastro dos tipos de produtos e estes são gerados individualmente. Para a representação dos cadastros é apenas utilizada uma página que tem seus campos gerados quando selecionado um produto para visualização. Esta página apresenta os campos nome, marca, preço e imagem que são padrões em todos os cadastros, além das informações específicas de cada produto.

4 CONCLUSÕES

Este trabalho teve como objetivos o desenvolvimento de um aplicativo móvel para divulgação de produtos de empresas para a plataforma Android e um módulo web com facilidade de manutenção, a partir do uso de metadados para geração dos cadastros dos produtos. Os objetivos foram alcançados, tendo todos os requisitos atendidos.

O módulo web fornece ao administrador um recurso para geração dos cadastros dos tipos de produto baseada em metadados e seu resultado pode ser utilizado pelas empresas a partir de uma página que se adequa conforme o tipo de produto que desejasse cadastrar.

O módulo web fornece às empresas recursos para personalização de tipos de produtos e cadastro de produtos para que sejam divulgados a partir do aplicativo móvel desenvolvido. Além disto, tem a possibilidade de criar requisições para o administrador criar novos tipos de produtos ou incluir novas características. É possível também visualizar e responder perguntas feitas pelos usuários para seus produtos.

O aplicativo móvel fornece aos usuários recursos para pesquisa de produtos e empresas, que apresenta primeiramente os resultados mais próximos de sua região, considerando bairro, cidade e estado. Além disto, é possível marcar produtos e empresas como favoritos para posterior acesso *off-line* e realizar perguntas e comentários para os produtos das empresas.

O módulo web foi desenvolvido na linguagem Java juntamente com o framework JSF e a suíte *open-source* PrimeFaces através da IDE Eclipse. Para a persistência dos dados foi utilizado o banco de dados MySQL. O aplicativo móvel também foi desenvolvido na linguagem Java utilizando a IDE Eclipse, porém com o *plugin* ADT. Para persistência foi utilizado o banco de dados Not Only SQL (NoSQL) SQLite. Foi desenvolvido também um *web service* para comunicação do aplicativo móvel com o servidor, na linguagem Java juntamente com a IDE Eclipse e para persistência dos dados o banco de dados MySQL. Para comunicação entre o *web service* e o aplicativo móvel foi utilizada a biblioteca GSON para serialização e desserialização das informações.

Com base nos resultados obtidos, avalia-se de forma positiva a utilidade do sistema, pois proporciona aos usuários conhecimento e interação a partir de comentários e perguntas com as empresas que se encontram mais próximas conforme bairro, cidade e estado informados em seu cadastro. Além disto, proporciona ao administrador do sistema recursos que facilitam a comunicação com as empresas e na manutenção dos cadastros dos produtos, sem precisar realizar alterações em códigos fonte e banco de dados. Deste modo, também é

válido afirmar que as ferramentas e tecnologias utilizadas se demonstraram satisfatórias para o desenvolvimento do sistema.

4.1 EXTENSÕES

Para trabalhos futuros, podem-se citar várias melhorias e complementos para o sistema desenvolvido. A seguir são apresentados os principais:

- a) permitir que sejam informadas máscaras para as características dos tipos de produto, deste modo quando um campo for de telefone, por exemplo, facilitará na visualização e compreensão da informação;
- b) oferecer recursos para listagem dos produtos e empresas mais acessados por região e período, para que os usuários, empresas e até mesmo o administrador consigam extrair conhecimento destes;
- c) permitir que sejam adicionadas mais imagens para os produtos, no momento é possível adicionar apenas uma;
- d) realizar a normalização do endereço da empresa e do usuário e posteriormente oferecer recursos para que as pesquisas por empresas e produtos apresentem resultados próximos a região em que o usuário se encontra utilizando geolocalização;
- e) oferecer recursos para monetização com *banners*.

REFERÊNCIAS

- AREND, Felipe G. **Geração de operações CRUD a partir de metadados**. 2011. 72 f. Trabalho Final de Graduação (Bacharelado em Ciências da Computação) - Departamento de Eletrônica e Computação, Universidade Federal de Santa Maria, Santa Maria.
- BENACCHIO, Alcione; VAZ, Maria S.M.G. **Metapadrão** - Descrição e Integração de Padrões de Metadados. [S.l.], 2008. Disponível em: <http://www.unieuro.edu.br/sitenovo/revistas/downloads/ruti_01_04_Metapadrao.pdf>. Acesso em: 22 nov. 2015.
- BUSCAPÉ. **Conheça o Buscapé**. [2015?]. Disponível em: <<http://www.buscape.com.br/conheca-o-buscape>>. Acesso em: 05 abr. 2015.
- DAMASCENO, L. L. **Interoperabilidade de metadados em aplicações distribuídas: desenvolvimento de ferramentas para validação de metamodelos**. 2005. 113 f. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas.
- DATA FOLHA. **43 milhões de brasileiros acessam internet por dispositivos móveis**. São Paulo, n. 425, jan. 2014. Disponível em: <<http://datafolha.folha.uol.com.br/mercado/2014/01/1400618-43-milhoes-de-brasileiros-acessam-internet-por-dispositivos-moveis.shtml>>. Acesso em: 05 abr. 2015.
- E-COMMERCE BRASIL. **Aplicativo mobile: para aumentar os acessos e vendas**. [S.I.], jun. 2015. Disponível em: <<https://www.ecommercebrasil.com.br/artigos/aplicativo-mobile-para-aumentar-os-acessos-e-vendas/>>. Acesso em: 20 out. 2015.
- E-COMMERCE BRASIL. **Mobile impulsiona receita de pequenas e médias empresas, informa estudo**. [S.I.], jan. 2014. Disponível em: <<http://www.ecommercebrasil.com.br/noticias/mobile-impulsiona-receita-de-pequenas-e-medias-empresas-informa-estudo/>>. Acesso em: 05 abr. 2015.
- GITHUB. **A Java serialization library that can convert Java Objects into JSON and back**. 2015. Disponível em: <<https://github.com/google/gson>>. Acesso em: 12 dez. 2015.
- GUIA FÁCIL COMUNICAÇÃO. **Quem somos**. [2015?]. <http://www.guiafacilcomunicacao.com/#quem_somos>. Acesso em: 05 abr. 2015.
- JERSEY. **About**. 2015. Disponível em: <<https://jersey.java.net/>>. Acesso em: 12 dez. 2015.
- LEONEL, Thais. A importância do Marketing Digital. **WSI**, São Paulo, n. 151, 2014. Disponível em: <<http://www.wsiconsultores.com.br/importancia-marketing-digital/>>. Acesso em: 05 abr. 2015.
- SOFTWARE, Chamella. **É hora de investir em estratégias de marketing para dispositivos móveis**. [S.I.], 2015. Disponível em: <<http://chamella.com.br/blog/e-hora-de-investir-em-estrategias-de-marketing-para-dispositivos-moveis/>>. Acesso em: 22 nov. 2015.
- SOUNDY, News. **Aplicativo móvel, seu site no bolso do seu cliente**. São Paulo, n. 585, 2015. Disponível em: <<http://soundy.cc/aplicativo-movel-seu-site-no-bolso-do-seu-cliente/>>. Acesso em: 05 abr. 2015.
- SOUNDY, News. **Entenda a importância do aplicativo mobile para o sucesso nos negócios**. São Paulo, 2014. Disponível em: <<http://soundy.cc/entenda-a-importancia-do-aplicativo-mobile-para-o-sucesso-nos-negocios/>>. Acesso em: 05 abr. 2015.

TIWORLD. **Por que desenvolver um aplicativo para sua empresa.** Rio de Janeiro, n. 180, 2014. Disponível em: <<http://www.tiworld.com.br/aplicativo-para-sua-empresa-2/>>. Acesso em: 05 abr. 2015.

APÊNDICE A – Descrição dos Casos de Uso do Módulo Web

Este Apêndice apresenta a descrição dos principais casos de uso do módulo web, apresentados na seção 3.2.2. No Quadro 16 é apresentado o detalhamento do caso de uso Manter o cadastro de produtos.

Quadro 16 – Detalhamento do Caso de Uso 02 – Manter o cadastro de produtos

UC02 – Manter produtos

Permite a empresa criar um novo produto a partir de um tipo de produto, para posteriormente ser disponibilizado no aplicativo móvel, permitindo que seja consultado por usuários. Além disto a empresa pode efetuar consultas a todos os produtos cadastrados.

Ator(es): Empresa, Administrador.

Constraints:

Pré-condição: A empresa deve possuir um cadastro.

Pré-condição: A empresa deve estar conectada ao módulo web.

Pré-condição: O tipo de produto referente ao produto deve estar cadastrado.

Pré-condição: A empresa deve ter personalizado os campos do tipo de produto referente ao produto.

Pós-condição: A empresa realizou o cadastro do produto.

Cenário Principal:

1. Empresa seleciona a opção “Meus Produtos” no menu principal.
2. Aplicação exibe a página para pesquisa de tipos de produto.
3. Empresa seleciona o tipo de produto desejado.
4. Empresa clica no botão “Selecionar”.
5. Aplicação exibe página para pesquisa de produtos.
6. Empresa clica no botão “Novo”.
7. Aplicação exibe página para cadastro do produto.
8. Empresa informa nome, marca, preço e imagem, além dos campos específicos.
9. Empresa clica no botão “Gravar”.
10. Aplicação grava as informações no banco de dados.
11. Aplicação apresenta mensagem de sucesso.

Cenário de Alteração:

6. Empresa seleciona o produto desejado.
7. Empresa clica no botão “Editar”.
8. Aplicação exibe página com as informações do produto.
9. Empresa altera as informações conforme desejado.
10. Empresa clica no botão “Gravar”.
11. Aplicação grava as informações no banco de dados.
12. Aplicação apresenta mensagem de sucesso.

Cenário de Exclusão:

6. Empresa seleciona o produto desejado.
7. Empresa clica no botão “Editar”.
8. Aplicação exibe página com as informações do produto.
9. Empresa clica no botão “Excluir”.
10. Aplicação exclui o produto no banco de dados.
11. Aplicação apresenta mensagem de sucesso.
12. Aplicação redireciona a página para a pesquisa de produtos.

No Quadro 17 é apresentado o detalhamento do caso de uso Cadastrar requisições.

Quadro 17 – Detalhamento do Caso de Uso 06 – Cadastrar requisições

UC06 – Cadastrar requisições

Permite a empresa enviar uma requisição para pedir a inclusão ou alteração de campos de tipos de produtos para o administrador do sistema.

Ator(es): Empresa.

Constraints:

Pré-condição: A empresa deve possuir um cadastro.

Pré-condição: A empresa deve estar conectada ao módulo *web*.

Pós-condição: A empresa cadastrou uma nova requisição.

Cenário Principal:

1. Empresa seleciona a opção “Requisições” no menu principal.
2. Aplicação exibe a página para pesquisa de requisições.
3. Empresa clica no botão “Nova Requisição”.
4. Aplicação exibe a página para cadastro de requisições.
5. Empresa informa título e descrição da requisição conforme suas necessidades.
6. Empresa clica no botão “Gravar”.
7. Aplicação grava a requisição no banco de dados.
8. Aplicação mostra mensagem de sucesso.

Cenário de Alteração:

3. Empresa seleciona a requisição desejada.
4. Empresa clica no botão “Visualizar”.
5. Aplicação exibe a página com as informações da requisição.
6. Empresa altera informações conforme necessidade.
7. Empresa clica no botão “Gravar”.
8. Aplicação grava a requisição no banco de dados.
9. Aplicação mostra mensagem de sucesso.

No Quadro 18 é apresentado o detalhamento do caso de uso Personalizar tipo de produto.

Quadro 18 – Detalhamento do Caso de Uso 03 – Personalizar tipo de produto

UC03 – Personalizar tipo de produto

Permite a empresa personalizar as características dos tipos de produtos desejados.

Ator(es): Empresa.

Constraints:

Pré-condição: A empresa deve possuir um cadastro.

Pré-condição: A empresa deve estar conectada ao módulo *web*.

Pré-condição: O tipo de produto referente ao produto deve estar cadastrado.

Pós-condição: A empresa realizou a personalização do tipo de produto.

Cenário Principal:

1. Empresa seleciona a opção “Personalizar Tipo de Produto” no menu principal.
2. Aplicação exibe a página para pesquisa de tipos de produto.
3. Empresa seleciona o tipo de produto desejado.
4. Empresa clica no botão “Personalizar”.
5. Aplicação exibe página para personalização do tipo de produto.
6. Empresa altera as características do tipo de produto desejadas para cadastro ao lado direito do quadro.
7. Empresa clica no botão “Gravar”.
8. Aplicação grava a personalização do tipo de produto no banco de dados.
9. Aplicação apresenta mensagem de sucesso.

No Quadro 19 é apresentado o detalhamento do caso de uso Manter o cadastro de tipos de produtos.

Quadro 19 – Detalhamento do Caso de Uso 10 – Manter o cadastro de tipos de produtos

UC10 – Manter tipos de produtos

Permite ao administrador manter o cadastro de tipos de produtos e suas características.

Ator(es): Administrador.

Constraints

Pré-condição: O administrador deve possuir um cadastro.

Pré-condição: O administrador deve estar conectado no módulo web.

Pós-condição: O administrador realizou o cadastro do tipo de produto.

Cenário Principal

1. Administrador seleciona a opção “Tipos de produto” no menu principal.
2. Aplicação exibe a página para pesquisa de tipos de produto.
3. Administrador clica no botão “Novo”.
4. Aplicação exibe a página para cadastro de tipos de produto.
5. Administrador informa os campos nome e descrição, além dos atributos que são constituídos por nome, tipo e tamanho.
6. Administrador clica no botão “Gravar”.
7. Aplicação grava o tipo de produto no banco de dados.
8. Aplicação apresenta mensagem de sucesso.

Cenário de Alteração

3. Administrador seleciona o tipo de produto desejado.
4. Administrador clica no botão “Editar”.
5. Aplicação exibe página com as informações do tipo de produto.
6. Administrador altera as informações conforme necessidade.
7. Administrador clica no botão “Gravar”.
8. Aplicação grava o tipo de produto no banco de dados.
9. Aplicação apresenta mensagem de sucesso.

APÊNDICE B – Descrição dos Casos de Uso do Aplicativo Móvel

Este Apêndice apresenta a descrição dos principais casos de uso do aplicativo móvel, apresentados na seção 3.2.2. No Quadro 20 é apresentado o detalhamento do caso de uso Pesquisar produtos.

Quadro 20 – Detalhamento do caso de uso 03 – Pesquisar produtos

UC03 – Pesquisar produtos

Permite ao usuário realizar pesquisas por produtos que foram cadastrados pelas empresas no módulo web.

Ator(es): Usuário.

Constraints

Pré-condição: O usuário deve ter efetuado seu cadastro.

Pré-condição: O sistema deve conter tipos de produto cadastrados.

Pré-condição: O sistema deve conter produtos cadastrados.

Pré-condição: O usuário deve ter efetuado autenticação.

Pós-condição: O usuário listou os produtos conforme sua necessidade.

Cenário Principal

1. Usuário seleciona a opção “Pesquisar Produtos” no menu principal.
2. Aplicação exibe tela para pesquisa de tipos de produto.
3. Usuário informa o nome tipo de produto desejado.
4. Aplicação exibe os tipos de produto conforme o nome informado pelo usuário.
5. Usuário seleciona o tipo de produto desejado.
6. Aplicação exibe tela para pesquisa de produtos.
7. Usuário informa nome e marca do produto desejado.
8. Usuário clica no botão “Pesquisar”.
9. Aplicação exibe os produtos conforme nome e marca informados, listando primeiramente os produtos de empresas do bairro, cidade e estado conforme cadastro do Usuário.

Cenário Alternativo

1. No passo 8 o Usuário clica no botão “Pesquisa Avançada”.
2. Aplicação exibe a tela de pesquisa avançada de produtos.
3. Usuário informa as características e os valores para ser aplicado como filtro na pesquisa.
4. Usuário clica no botão “Concluir”.
5. Aplicação exibe os produtos conforme nome, marca e filtros informados, listando primeiramente os produtos de empresas do bairro, cidade e estado conforme cadastro do Usuário.

No Quadro 21 é apresentado o detalhamento do caso de uso Pesquisar empresas.

Quadro 21 – Detalhamento do Caso de Uso 04 – Pesquisar empresas

UC04 – Pesquisar empresas

Permite ao usuário realizar pesquisas por empresas que realizaram seu cadastrado no módulo web.

Ator(es): Usuário.

Constraints

Pré-condição: O sistema deve conter ramos de atividades cadastrados.

Pré-condição: O sistema deve conter empresas cadastradas.

Pré-condição: O usuário deve ter efetuado seu cadastro.

Pré-condição: O usuário deve ter efetuado autenticação.

Pós-condição: O usuário listou as empresas conforme sua necessidade.

Cenário Principal

1. Usuário seleciona a opção “Pesquisar Empresas” no menu principal.
2. Aplicação exibe tela para pesquisa de empresas.
3. Usuário informa nome da empresa e seleciona o ramo de atividade desejado.
4. Usuário clica no botão “Pesquisar”.
5. Aplicação exibe as empresas conforme nome e ramo de atividade informados, listando primeiramente as empresas que se encontram no bairro, cidade e estado conforme o cadastro do Usuário.

APÊNDICE C – Dicionário de Dados do Módulo Web

Este apêndice apresenta as descrições das tabelas do banco de dados do módulo web.

Os tipos de dados utilizados são:

- a) `integer`: armazena números inteiros;
- b) `varchar`: armazena caracteres alfanuméricos;
- c) `medium blob`: armazena arquivos em geral.

No Quadro 22 é apresentado o dicionário de dados da tabela `Empresa`.

Quadro 22 – Tabela `Empresa`

Empresa					
Armazena as empresas					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	código	INTEGER	-	Sim	Não
email	E-mail	VARCHAR	80	Não	Não
senha	Senha	VARCHAR	15	Não	Não
endereco	Endereço	VARCHAR	80	Não	Não
complemento	Complemento do Endereço	VARCHAR	80	Não	Não
telefone	Telefone	VARCHAR	11	Não	Não
celular	Celular	VARCHAR	11	Não	Não
cnpj	CNPJ	VARCHAR	15	Não	Não
nome	Nome da empresa	VARCHAR	40	Não	Não
descricao	Descrição da Empresa	VARCHAR	400	Não	Não
cidade	Cidade	VARCHAR	30	Não	Não
estado	Sigla do estado	VARCHAR	2	Não	Não
fk_ramo_empresa	Chave do ramo de atividade	INTEGER	-	Não	Sim
bairro	Bairro	VARCHAR	30	Não	Não
fk_imagem_empresa	Chave da imagem	INTEGER	-	Não	Sim
perfil	Código do perfil	INTEGER	-	Não	Não

No Quadro 23 é apresentado o dicionário de dados da tabela `Ramo`.

Quadro 23 – Tabela `Ramo`

Ramo					
Armazena os ramos de atividade					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código	INTEGER	-	Sim	Não
descricao	Descrição	VARCHAR	45	Não	Não

No Quadro 24 é apresentado o dicionário de dados da tabela Imagem.

Quadro 24 – Tabela Imagem

Imagem					
Armazena as imagens					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
Id	Código	INTEGER	-	Sim	Não
Imagem	Descrição	MEDIUM BLOB	-	Não	Não

No Quadro 25 é apresentado o dicionário de dados da tabela Requisicao.

Quadro 25 – Tabela Requisicao

Requisicao					
Armazena as requisições					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
Id	Código	INTEGER	-	Sim	Não
fk_id_empresa_requisicao	Código da empresa	INTEGER	-	Sim	Sim
Titulo	Titulo	VARCHAR	40	Não	Não
descricao	Descrição	VARCHAR	400	Não	Não
Resposta	Resposta	VARCHAR	400	Não	Não
Atendida	Atendida	INT	-	Não	Não

No Quadro 26 é apresentado o dicionário de dados da tabela TipoProduto.

Quadro 26 – Tabela TipoProduto

TipoProduto					
Armazena os tipos de produto					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
Id	Código	INTEGER	-	Sim	Não
Nome	Nome	VARCHAR	40	Não	Não
descricao	Descrição	VARCHAR	200	Não	Não

No Quadro 27 é apresentado o dicionário de dados da tabela Atributo.

Quadro 27 – Tabela Atributo

Atributo					
Armazena os atributos dos tipos de produto					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
Id	Código	INTEGER	-	Sim	Não
fk_tipo_produto_atributo	Código tipo de produto	INTEGER	-	Sim	Sim
Nome	Nome	VARCHAR	40	Não	Não
Tipo	Tipo	INT	-	Não	Não
Tamanho	Tamanho	INT	-	Não	Não

No Quadro 28 é apresentado o dicionário de dados da tabela *AtributoEmpresa*.

Quadro 28 – Tabela *AtributoEmpresa*

AtributoEmpresa					
Armazena os atributos dos tipos de produtos personalizados					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
fk_empresa_atributo_emp	Código empresa	INTEGER	-	Sim	Sim
fk_atributo_atributo_emp	Código atributo	INTEGER	-	Sim	Sim
fk_tipo_produto_atributo_emp	Código tipo de produto	INTEGER	-	Sim	Sim

No Quadro 29 é apresentado o dicionário de dados da tabela *Usuario*.

Quadro 29 – Tabela *Usuario*

Usuario					
Armazena os usuários					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código	INTEGER	-	Sim	Não
nome	Nome	VARCHAR	40	Não	Não
email	E-mail	VARCHAR	80	Não	Não
senha	Senha	VARCHAR	15	Não	Não
bairro	Bairro	VARCHAR	30	Não	Não
cidade	Cidade	VARCHAR	30	Não	Não
estado	Sigla do Estado	VARCHAR	2	Não	Não

No Quadro 30 é apresentado o dicionário de dados da tabela *Comentario*.

Quadro 30 – Tabela *Comentario*

Comentario					
Armazena os comentários					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código	INTEGER	-	Sim	Não
fk_empresa_comentario	Código empresa	INTEGER	-	Sim	Sim
fk_tipo_produto_comentario	Código tipo de produto	INTEGER	-	Sim	Sim
fk_produto_comentario	Código produto	INTEGER	-	Sim	Sim
conteudo	Conteúdo	VARCHAR	200	Não	Não
fk_usuario_comentario	Código usuário	INTEGER	-	Não	Sim

No Quadro 31 é apresentado o dicionário de dados da tabela Pergunta.

Quadro 31 – Tabela Pergunta

Pergunta					
Armazena as perguntas					
Nome Coluna	Descrição	Tipo	Tamanho	Chave Primária	Chave Estrangeira
id	Código	INTEGER	-	Sim	Não
fk_empresa_pergunta	Código empresa	INTEGER	-	Sim	Sim
fk_tipo_produto_pergunta	Código tipo de produto	INTEGER	-	Sim	Sim
fk_produto_pergunta	Código produto	INTEGER	-	Sim	Sim
conteudo	Conteúdo	VARCHAR	200	Não	Não
resposta	Resposta	VARCHAR	200	Não	Não
fk_usuario_pergunta	Código usuário	INTEGER	-	Não	Sim

APÊNDICE D – Dicionário de Dados do Aplicativo Móvel

Este apêndice apresenta as descrições das tabelas do banco de dados do módulo web.

Os tipos de dados utilizados são:

- a) `integer`: armazena números inteiros;
- b) `text`: armazena caracteres alfanuméricos;
- c) `blob`: armazena arquivos em geral;
- d) `real`: armazena números decimais.

No Quadro 32 é apresentado o dicionário de dados da tabela `Empresa`.

Quadro 32 – Tabela `Empresa`

Empresa			
Armazena as empresas			
Nome Coluna	Descrição	Tipo	Chave Primária
ID	Código	INTEGER	Sim
EMAIL	E-mail	TEXT	Não
ENDERECO	Endereço	TEXT	Não
COMPLEMENTO	Complemento	TEXT	Não
TELEFONE	Telefone	TEXT	Não
CELULAR	Celular	TEXT	Não
NOME	Nome	TEXT	Não
DESCRICA0	Descrição	TEXT	Não
IDIMAGEM	Código imagem	INTEGER	Não
CIDADE	Cidade	TEXT	Não
ESTADO	Estado	TEXT	Não
IDRAMO	Código do ramo de atividade	INTEGER	Não

No Quadro 33 é apresentado o dicionário de dados da tabela `EmpresaFavorita`.

Quadro 33 – Tabela `EmpresaFavorita`

EmpresaFavotira			
Armazena as empresas favoritas			
Nome Coluna	Descrição	Tipo	Chave Primária
IDEMPRESA	Código empresa	INTEGER	Sim
IDUSUARIO	Código usuário	INTEGER	Sim

No Quadro 34 é apresentado o dicionário de dados da tabela `Usuario`.

Quadro 34 – Tabela `Usuario`

Usuario			
Armazena os usuários			
Nome Coluna	Descrição	Tipo	Chave Primária
ID	Código usuário	INTEGER	Sim
NOME	Nome	TEXT	Não
EMAIL	E-mail	TEXT	Não
SENHA	Senha	TEXT	Não
BAIRRO	Bairro	TEXT	Não
CIDADE	Cidade	TEXT	Não
ESTADO	Sigla do Estado	TEXT	Não

No Quadro 35 é apresentado o dicionário de dados da tabela `TipoProduto`.

Quadro 35 – Tabela TipoProduto

TipoProduto			
Armazena os tipos de produto			
Nome Coluna	Descrição	Tipo	Chave Primária
ID	Código usuário	INTEGER	Sim
NOME	Nome	TEXT	Não
DESCRICAÇÃO	Descrição	TEXT	Não

No Quadro 36 é apresentado o dicionário de dados da tabela TipoProdutoEmpresa.

Quadro 36 – Tabela TipoProdutoEmpresa

TipoProdutoEmpresa			
Armazena os tipos de produto da empresa			
Nome Coluna	Descrição	Tipo	Chave Primária
IDEMPRESA	Código empresa	INTEGER	Sim
IDTIPOPRODUTO	Código tipo de produto	INTEGER	Sim

No Quadro 37 é apresentado o dicionário de dados da tabela Produto.

Quadro 37 – Tabela Produto

Produto			
Armazena os produtos			
Nome Coluna	Descrição	Tipo	Chave Primária
ID	Código	INTEGER	Sim
IDEMPRESA	Código empresa	INTEGER	Sim
IDTIPOPRODUTO	Código tipo de produto	INTEGER	Sim
NOME	Nome	TEXT	Não
MARCA	Marca	TEXT	Não
PREÇO	Preço	REAL	Não
JSON	Produto no formato JSON	TEXT	Não
IDIMAGEM	Código imagem	INTEGER	Não

No Quadro 38 é apresentado o dicionário de dados da tabela ProdutoFavorito.

Quadro 38 – Tabela ProdutoFavorito

ProdutoFavorito			
Armazena os produtos favoritos			
Nome Coluna	Descrição	Tipo	Chave Primária
ID	Código do produto	INTEGER	Sim
IDEMPRESA	Código empresa	INTEGER	Sim
IDTIPOPRODUTO	Código tipo de produto	INTEGER	Sim
IDUSUARIO	Código usuário	INTEGER	Sim

No Quadro 39 é apresentado o dicionário de dados da tabela RamoAtividade.

Quadro 39 – Tabela RamoAtividade

RamoAtividade			
Armazena os ramos de atividade			
Nome Coluna	Descrição	Tipo	Chave Primária
ID	Código	INTEGER	Sim
DESCRICAÇÃO	Descrição	TEXT	Não

No Quadro 40 é apresentado o dicionário de dados da tabela Imagem.

Quadro 40 – Tabela Imagem

Imagem			
Armazena as imagens			
Nome Coluna	Descrição	Tipo	Chave Primária
ID	Código	INTEGER	Sim
IMAGEM	Imagem	BLOB	Não