



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

EDUARDO FRANCISCO PARANHOS

**SUPERCIDADE: APLICATIVO ANDROID E SOFTWARE WEB
DE REDE SOCIAL MUNICIPAL**

**Assis/SP
2017**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

EDUARDO FRANCISCO PARANHOS

**SUPERCIDADE: APLICATIVO ANDROID E SOFTWARE WEB
DE REDE SOCIAL MUNICIPAL**

Trabalho de conclusão de curso de Bacharel em Ciência da Computação apresentado ao Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando(a): Eduardo Francisco Paranhos
Orientador(a): Luiz Ricardo Begosso**

**Assis/SP
2017**

FICHA CATALOGRÁFICA

P223s PARANHOS, Eduardo Francisco.

SUPERCIDADE: APLICATIVO ANDROID E SOFTWARE WEB DE REDE SOCIAL MUNICIPAL / Eduardo Francisco Paranhos -- Assis, 2017.

46p.

Trabalho de Conclusão de Curso (Ciência da Computação) – Fundação Educacional do Município de Assis - FEMA.

Orientador: Dr. Luiz Ricardo Begosso.

1. Java. 2. Android. 3. Hibernate. 4. Restful. 5. Json.

CDD: 005.133
Biblioteca da FEMA

**SUPERCIDADE: APLICATIVO ANDROID E SOFTWARE WEB
DE REDE SOCIAL MUNICIPAL**

EDUARDO FRANCISCO PARANHOS

Trabalho de conclusão de curso de Bacharel em Ciência da Computação apresentado ao Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Professor Dr. Luiz Ricardo Begosso

Examinador: _____
Professor Osmar Aparecido Machado

DEDICATÓRIA

Dedico este trabalho a Deus, em primeiro lugar, aos meus familiares, principalmente aos meus pais, Dinei e Lourdes, que incansavelmente fizeram o possível pra me tornar o homem que sou hoje, a minha esposa Valéria, que se fez companheira em todos os momentos, ao meu filho Dinei Augusto, que é o grande motivo para que eu procure evoluir e servir como espelho, e aos meus amigos, que sempre acreditaram que eu seria capaz.

AGRADECIMENTOS

Agradeço a Deus, em primeiro lugar, por me dar conhecimento e sabedoria, para que eu pudesse superar cada etapa desta Graduação, por ter me abençoado a cada novo dia, não removendo barreiras do meu caminho, mas me tornando forte para superá-los e chegar ao final desta importante trajetória.

Agradeço a todos os meus familiares, mas em destaque, meu pai, que não importasse a dificuldade, levantava cedo e saía pra rotina diária de trabalho, muitas vezes doente, outras vezes cansado em virtude de poucas horas de sono pra nada faltar aos de nossa casa, a minha Mãe, guerreira, paciente, educadora, que passou e passa muitas noites em claro pra cuidar e proteger a todos de que ama, que tinha por sonho ver seus filhos “formados”, e sempre acreditou que esse dia chegaria. A minha esposa, que por tantas vezes disse que eu conseguiria, mesmo quando nem eu acreditasse nisso, e acertou.

Agradeço ao meu Orientador, Professor Dr. Luiz Ricardo Begosso, por seus ensinamentos, suas orientações, ideias e estímulos, do tipo “*Tá bom... Mas pode melhorar*”.

Agradeço a todos os professores da FEMA, porque em algum momento, todos, sem exceção, contribuíram para o meu crescimento educacional e profissional, me dando condições de ampliar meus conhecimentos e estrutura necessária para encarar o mercado de trabalho, e o melhor, sempre deixaram claro que a aprendizagem deve ser contínua e o conhecimento precisa ser alimentada a cada dia.

Agradeço aos meus amigos de sala, porque muitas foram as vezes, nos ajudamos, nos apoiamos para superar os obstáculos, inclusive aos que ficaram pelo caminho, porque também contribuíram em algum momento.

Agradeço a Direção da FEMA, por meio de seu Presidente, seu Diretor e membros do Conselho Curador, que possibilitam a qualificação de funcionários e seus familiares, de forma gratuita e sem distinção de direitos e deveres, tornando-nos pessoas mais qualificadas.

Agradeço aos meus companheiros de trabalho, meus chefes, que sempre que houve necessidade, adaptaram atividades ou escalas pra que eu pudesse participar de algum evento ou me preparar para provas.

Trabalhe para que teus clientes te procurem por ser o melhor, não o mais barato! (Cris Andrade)

RESUMO

SUPERCIDADE é um software de informação, em forma de rede social, que permite a comunicação entre Prefeituras e seus munícipes, formado por 2 módulos, distintos, que se interagem. Um dos módulos é um sistema desenvolvido para dispositivos Android e o outro uma aplicação web. Por meio do sistema Android, o usuário, poderá divulgar em rede social imagens do município, sugerir ideias, acompanhar publicações de notícias por parte da administração municipal, obras em andamento, acompanhar calendários cívicos da cidade, consultar endereços e/ou telefones das secretarias, órgãos de utilidades públicas e ainda, saber os eventos em destaque no município. O sistema web é o canal onde pessoas autorizadas e devidamente autenticadas alimentarão o banco de dados do sistema, que permitirá a comunicação entre as partes. Toda a alimentação de dados do sistema, por parte da administração municipal, será feita através desta aplicação.

O objetivo é criar um mecanismo de comunicação e interação de munícipes e a administração municipal. Contribuir com o crescimento da sociedade e a melhoria da qualidade de vida de todos.

Com este software, a população é quem levará as notícias e as solicitações, mas com foco específico em acontecimentos diretamente ligados a sua cidade, e não a uma rede social mundial como alguns aplicativos já fazem.

Para desenvolvimento da aplicação, serão utilizados as seguintes tecnologias: Linguagem Java, banco de Dados MySQL, frameworks como Hibernate, CDI, MAVEN, Spring Security, Spring OAUTH2, entre outros.

Concluindo, o SUPERCIDADE tem por prioridade apoiar a comunicação e o crescimento do município. E pode se tornar uma grande ferramenta de crescimento a toda sociedade.

Palavras-chave: Java, Android, Hibernate, Restful, JSON.

ABSTRACT

SUPERCIDADE is a software which works as a social networking that allows communication between City Hall and residents. Composed by two distinct modules which interact each other. One of them is the Android system and the other is the Web Site.

The Android system allows the users to post city's photos on the social network. Those posts can be made to demonstrate satisfaction (or dissatisfaction), support campaigns or support posts from other users, all for the benefit of the City. Another functionality is suggest ideas to unused places. Adding is possible to follow city's news, construction in progress, city's calendar. Moreover consult information like City Offices address and phone numbers also events dates. Furthermore, there is a specific channel to the users send message to the City Office, as a consequence that City Office can answer this message. The Web Site is a channel used by municipal authorized users to register information.

In conclusion, the main purpose of SUPERCIDADE is help with communication between city's members and support the growth of the city. Therefore it can become a great growth tool for the society.

Keywords: Java, Android, Hibernate, Restful, JSON, MySQL.

LISTA DE ILUSTRAÇÕES

Figura 1: Características do MySQL	20
Figura 2: Composição de uma tabela	21
Figura 3: Estrutura Analítica do Projeto SUPERCIDADE	27
Figura 4: Diagrama Entidade Relacionamento do Sistema SUPERCIDADE.....	29
Figura 5: Casos de Uso do Sistema SUPERCIDADE	30
Figura 6: Use Case Manter Categorias	31
Figura 7: Use Case Manter Secretarias.....	32
Figura 8: Use Case Manter Instalações Públicas	32
Figura 9: Use Case Manter Políticos	33
Figura 10: Use Case Manter Utilidades Públicas.....	34
Figura 11: Use Case Manter Usuários.....	34
Figura 12: Use Case Manter Calendários.....	35
Figura 13: Use Case Manter Eventos	36
Figura 14: Use Case Manter História.....	36
Figura 15: Use Case Manter Notícias	37
Figura 16: Use Case Responder Canal Direto.....	38
Figura 17: Use Case Manter Publicações.....	38
Figura 18: Use Case Manter Minhas Publicações	39
Figura 19: Use Case Inserir Registros	40
Figura 20: Use Case Inserir Idéias	41
Figura 21: Use Case Consultar Notícias.....	41
Figura 22: Use Case Consultar Calendário	42
Figura 23: Use Case Manter Canal Direto	43
Figura 24: Use Case Consultar Secretarias.....	43

Figura 25: Use Case Consultar Utilidades Públicas44

LISTA DE TABELAS

Tabela 1: Cronograma de Desenvolvimento do Sistema SUPERCIDADE	28
Tabela 2: Especificação da Use Case Manter Categorias.....	31
Tabela 3: Especificação da Use Case Manter Secretarias	32
Tabela 4: Especificação da Use Case Manter Instalações Públicas	33
Tabela 5: Especificação da Use Case Manter Políticos.....	33
Tabela 6: Especificação da Use Case Utilidades Públicas	34
Tabela 7: Especificação da Use Case Manter Usuários	35
Tabela 8: Especificação da Use Case Manter Calendários	35
Tabela 9: Especificação da Use Case Manter Eventos	36
Tabela 10: Especificação da Use Case Manter História	37
Tabela 11: Especificação da Use Case Manter Notícias	37
Tabela 12: Especificação da Use Case Manter Canal Direto	38
Tabela 13: Especificação da Use Case Manter Publicações.....	39
Tabela 14: Especificação da Use Case Manter Minhas Publicações	40
Tabela 15: Especificação da Use Case Inserir Registros	40
Tabela 16: Especificação da Use Case Inserir Idéias.....	41
Tabela 17: Especificação da Use Case Consultar Notícias	42
Tabela 18: Especificação da Use Case Consultar Calendário.....	42
Tabela 19: Especificação da Use Case Manter Canal Direto	43
Tabela 20: Especificação da Use Case Consultar Secretarias.....	44
Tabela 21: Especificação da Use Case Consultar Utilidades Públicas.....	44

SUMÁRIO

1. INTRODUÇÃO	14
1.1. OBJETIVO.....	14
1.2. JUSTIFICATIVA	15
1.3. MÉTODO DE DESENVOLVIMENTO	15
2. TECNOLOGIAS DE DESENVOLVIMENTO	17
2.1. JAVA	17
2.1.1. JAVA SERVER FACES.....	17
2.2. ANDROID.....	18
2.2.1. INTERFACE.....	18
2.2.2. ANDROID STUDIO	19
2.2.3. APLICATIVOS	19
2.3. MYSQL.....	20
2.4. REST	21
2.5. JSON.....	22
2.6. HIBERNATE	23
2.7. ECLIPSE	23
2.8. MAVEN.....	23
2.9. CDI	24
2.10. SPRING SECURITY	24
2.11. SPRING OAUTH2	25
3. DOCUMENTAÇÃO	27
3.1. ESTRUTURA ANALÍTICA DO PROJETO	27
3.2. CRONOGRAMA.....	28
3.3. DIAGRAMA DE CLASSE	29
3.4. DIAGRAMA DE CASOS DE USO.....	29
3.5. ESPECIFICAÇÃO DOS CASOS DE USO.....	30
3.5.1. Manter Categorias.....	31
3.5.2. Manter Secretarias	32
3.5.3. Manter Instalações Públicas	32
3.5.4. Manter Políticos	33

3.5.5.	Manter Utilidades Públicas.....	34
3.5.6.	Manter Usuários	34
3.5.7.	Manter Calendários	35
3.5.8.	Manter Eventos	36
3.5.9.	Manter História	36
3.5.10.	Manter Notícias	37
3.5.11.	Responder Canal Direto	38
3.5.12.	Manter Publicações	38
3.5.13.	Manter Minhas Publicações	39
3.5.14.	Inserir Registros	40
3.5.15.	Inserir Idéias.....	41
3.5.16.	Consultar Notícias	41
3.5.17.	Consultar Calendário.....	42
3.5.18.	Manter Canal Direto	43
3.5.19.	Consultar Secretarias.....	43
3.5.20.	Consultar Utilidades Públicas	44
4.	CONCLUSÃO.....	45
5.	REFERÊNCIAS	46

1. INTRODUÇÃO

O Sistema Operacional (SO) Android, baseado no núcleo Linux, projetado pela empresa de tecnologia Google, é criado principalmente para smartphones e tablets, mas já pode ser encontrado versões específicas para TVs, carros e relógios, ainda em consoles de vídeo games e eletro eletrônicos. Atualmente, é o SO mais utilizado em dispositivos móveis no mundo e o principal escolhido por programadores de aplicações móveis (em 2013 estimava-se que 71% de desenvolvedores desenvolviam Android¹). É também o mais popular entre empresas de tecnologia que procuram um software pronto, de baixo custo e personalizável para dispositivos de alta tecnologia.

O Android teve uma aceitação morna quando foi revelado em 2007, desde então, cresceu e se tornou o SO móvel mais utilizado no mundo, e uma das experiências de sistema móvel mais velozes. A mídia elogiou o código aberto do Android, tendo esta característica a principal força do sistema, o que fez com que outras gigantes do mercado mundial criassem ramificações a partir do Android. Segundo a empresa de pesquisa Canalys, em 2016 o SO Android representava 94,4% das vendas de smartphones no Brasil. Relata ainda, que o mesmo domina no mundo inteiro, exceto em países onde o mercado tem foco em aparelhos premium (EUA, Reino Unido e Japão).

Devido a grande aceitação, facilidade de utilização e enorme popularidade do SO Android, o sistema SUPERCIDADE, software de informação, trabalhará com esta tecnologia, associada a aplicação web, que formarão 2 módulos, distintos, que se interagem.

1.1. OBJETIVO

O objetivo principal do software, é permitir a comunicação e interação, entre munícipes de um município e principalmente entre munícipes e a administração municipal.

Ao notar pontos do município que merecem atenção, por apreciação ou por rejeição, o usuário poderá publicar a imagem, seu endereço, sua categoria e o seu ponto de vista sobre a mesma em rede social comum ao município. Os demais usuários da aplicação poderão possibilidade de apoiar, efetuar comentários sobre o assunto e até mesmo

¹ Fonte: <https://pt.wikipedia.org/wiki/Android>

compartilhar com outras mídias sociais. Os usuários poderão ainda, acompanhar notícias da administração municipal, calendário cívico, eventos, consultar locais, endereços e telefones úteis da cidade e informações sobre órgãos públicos.

A aplicação SUPERCIDADE permitirá que cada cidadão possa acompanhar o andamento de obras públicas, tenha um canal direto para troca de mensagens com as secretarias e dê ideias de melhorias a espaços atualmente sem utilidade.

Ao acompanhar as publicações ou mesmo o recebimento de mensagens, a secretaria interessada diretamente no assunto, poderá, caso tenha interesse, apresentar justificativa e ainda o planejamento para solução do problema. No caso de sugestões, poderá expressar sua posição sobre a mesma.

A prioridade da aplicação não é de se transformar em uma ferramenta de denúncias ou revoltas, e sim, de apoiar a comunicação e o desenvolvimento do município.

1.2. JUSTIFICATIVA

Atualmente, várias pessoas tentam melhorar a sociedade, e muitas vezes, por falta de apoio, e principalmente de informações, não conseguem mudar sua realidade, nem a realidade dos que estão a sua volta. Quando existe a possibilidade de uma interação e até mesmo, porque não, de uma mobilização, ao acreditarem que a união os torna mais fortes, esta perspectiva melhora.

O SUPERCIDADE, de maneira fácil, leve e rápida, pretende oferecer este apoio a população em geral, pretende apoiar o desenvolvimento municipal e ainda, aproximar pessoas com interesses em comum, independente de se conhecerem ou não.

1.3. MÉTODO DE DESENVOLVIMENTO

A elaboração da arquitetura do software, que inclui o diagrama de classes, diagrama de entidade relacionamento e diagrama de atividade foram desenvolvidos pelos programas Astah e DBDesigner.

Para desenvolvimento das aplicações, linguagem de programação dos módulos do sistema será Java, serão utilizadas a IDE Eclipse IDE for Java EE Developers, da fundação The Eclipse Foundation e a IDE Android Studio da Google para desenvolvimento Android.

O sistema de gerenciamento de banco de dados (SGBD) será o MySQL, da Sun Microsystems. O módulo web do sistema fará comunicação direta com a base de dados por meio de autenticações de segurança. Para que o módulo Android possa alimentar ou acessar as informações da aplicação, será desenvolvido um Webservice, também em Java, que permitirá a comunicação, por meio de especificação REST, que permite acesso a serviços da web, usando mensagens HTTP de forma explícita e representativa para se comunicar, que neste sistema, irá utilizar a notação comum para transferência de dados chamado JSON, formato de dados derivado do JavaScript, independente de linguagem. O mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante anotações Java.

2. TECNOLOGIAS DE DESENVOLVIMENTO

2.1. JAVA

Criada na década de 90, o Java é uma linguagem de programação interpretada orientada a objetos, que diferente de outras linguagens tradicionais, compiladas para código nativo, a linguagem Java é compilada para bytecode, que é executado por uma máquina virtual. Sua plataforma foi adotada mais rápido do que qualquer outra linguagem de programação de toda a história. Desenvolvida pela Sun Microsystems, em novembro de 2006 partes do Java foram disponibilizadas como software livre, sob licença GNU General Public License. A liberação completa do código fonte sob a GPL se deu em maio de 2007. Em 2009, a Oracle Corporation adquire a empresa responsável pela linguagem Java, a Sun Microsystems, com o objetivo de levar o Java e outros produtos a disposição de seus consumidores.

A linguagem Java foi projetada, tendo como principais objetivos: orientação a objetos, portabilidade independente de plataforma, recursos de redes com extensa biblioteca com rotinas de comunicações com protocolos (HTTP, TCP/IP e FTP) e segurança de redes, podendo executar programas via rede com restrições de execução.

Outras vantagens ainda podem ser destacadas, como, por exemplo, sintaxe similar ao C/C++ e C#, facilidades de internacionalização porque suporta nativamente caracteres Unicode, linguagem e ambiente de execução (JVM - Java Virtual Machine) de simples especificação, vasto conjunto de bibliotecas (ou APIs) e carga dinâmica de código (os programas são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização). (SANTOS, 2014)

2.1.1. JAVA SERVER FACES

Quando se menciona a necessidade de desenvolvimento web com o Java, a plataforma usada é o Java EE (Java Enterprise Edition), que é atraente, apresenta portabilidade para várias plataformas e é suportado por muitos fabricantes. Na busca de grande desempenho e facilidade para programar interfaces de usuário, o JSF (Java Server

Faces) traz a rapidez no desenvolvimento de interfaces de usuário e vem despontando como principal solução para programadores web.

O JSF possui um conjunto de componentes pré-fabricados de interface de usuário, um modelo de programação orientado a eventos e um modelo de componentes que permite desenvolvedores fornecerem componentes adicionais. Possui componentes simples, como inputs e botões, e outros bastantes sofisticados, como tabelas de dados e árvores. Possui códigos necessários para manipulação de eventos e organização de componentes, o que permite aos programadores se dedicarem a lógica de programação.

2.2. ANDROID

Android é sistema operacional baseado no núcleo Linux e atualmente é desenvolvido pela Google. Baseado numa interface de manipulação direta, é projetado para dispositivos móveis com tela sensível ao toque como smartphones e tablets, com interface específica para TV (Android TV), carro (Android Auto) e relógio de pulso (Android Wear). É o SO mais utilizado no mundo, e, em 2013, possuía a maior porcentagem de vendas mundiais de SO móveis e 71% dos desenvolvedores para dispositivos móveis utilizam Android.

Abaixo, tem-se exposto o conceito de um dos fundadores da Android Inc, em 2003, que posteriormente seria comprada pela Google, a qual, o manteria juntamente com outros dois sócios em sua equipe:

"dispositivos móveis mais inteligentes que estejam mais cientes das preferências e da localização do seu dono" (RUBIN, 2003)

2.2.1. INTERFACE

De interface padrão de usuária baseada em manipulação direta, a resposta é desenhada para ser imediata e com fluidez. Possui hardwares internos como acelerômetros, giroscópios e sensores de proximidade para responder à ações do usuário, como mudança da orientação de tela. Assim que ligados, são direcionados a uma tela inicial (homescreen), com a navegação primária e principal do sistema. Composta por ícones (figuras) de aplicativos e widgets (componentes de uma interface gráfica do usuário (GUI) que inclui janelas, botões, menus, ícones, barras de rolagem, etc..). Os ícones direcionam o usuário ao aplicativo correspondente, enquanto o widget mostra informações na própria tela inicial, como previsão do tempo, e-mails não lidos, notícias, etc..

Aplicativos de terceiros podem ser adquiridos por meio de uma loja virtual, o Google Play ou em outras lojas do tipo.

No topo da tela é apresentado uma status bar (barra de status), que dá informações sobre o dispositivo e sua conectividade, barra essa, que pode ser expandida e mostrar notificações de outros aplicativos importantes ou atualizações, como um novo email, um novo SMS e assim por diante.

O Android permite ainda utilizar-se de aplicativos que mudem o iniciador padrão, inclusive mudanças que atingem diretamente a experiência do usuário com o sistema.

2.2.2. ANDROID STUDIO

Android Studio é um ambiente de desenvolvimento integrado (IDE) para desenvolver para a plataforma Android. Foi anunciado em 16 de Maio de 2013 na conferência Google I/O. Android Studio é disponibilizado gratuitamente sob a Licença Apache 2.0.

Baseado no software IntelliJ IDEA de JetBrains, foi criado especificamente para desenvolvimento Android e disponível para plataformas Windows, Mac OS X e Linux e substituiu o Eclipse Android Development Tools (ADT) como IDE primária do Google de desenvolvimento nativo para Android. (MONTEIRO, 2016)

2.2.3. APLICATIVOS

Também conhecidos como apps estendem as funcionalidades do dispositivo Android, e geralmente, são desenvolvidos primariamente em Java, usando-se do sistema de desenvolvimento do software Android (SDK), que inclui ferramentas de desenvolvimento, depurador, bibliotecas, emulador, códigos de exemplos e tutoriais.

É enorme a quantidade de aplicativos desenvolvidos por terceiros. Tais aplicativos podem ser instalados por meio da loja virtual de aplicativos ou através de downloads e instalação do arquivo APK. A principal loja dos usuários Android é a Play Store.

2.3. MYSQL

O MySQL é um sistema gerenciador de banco de dados (SGBD) que utiliza linguagem SQL (Structured Query Language) como interface. Criado na Suécia na década de 1980, por dois suecos e um finlandês, hoje seu desenvolvimento e manutenção empregam mais de 400 profissionais no mundo inteiro e mais de 1000 contribuem testando o software. Em 2008, foi adquirido pela Sun Microsystems e posteriormente, em 2009 pela Oracle que compraria a Sun. Hoje ele suporta Unicode, Full Text Indexes, replicação, Hot Backup, GIS, OLAP e muitos outros recursos de bancos de dados. (CARVALHO, 2015)

O MySQL é o banco de dados *open source* mais popular do mundo, segundo a publicação da Casa do Código sobre o assunto. O seu SGBD, além de ser uma ferramenta gratuita sob licença de software livre, oferece ainda qualidade, robustez e segurança.

A seguir, um mapa mental que lista de forma bastante clara e visual as qualidades do MySQL:

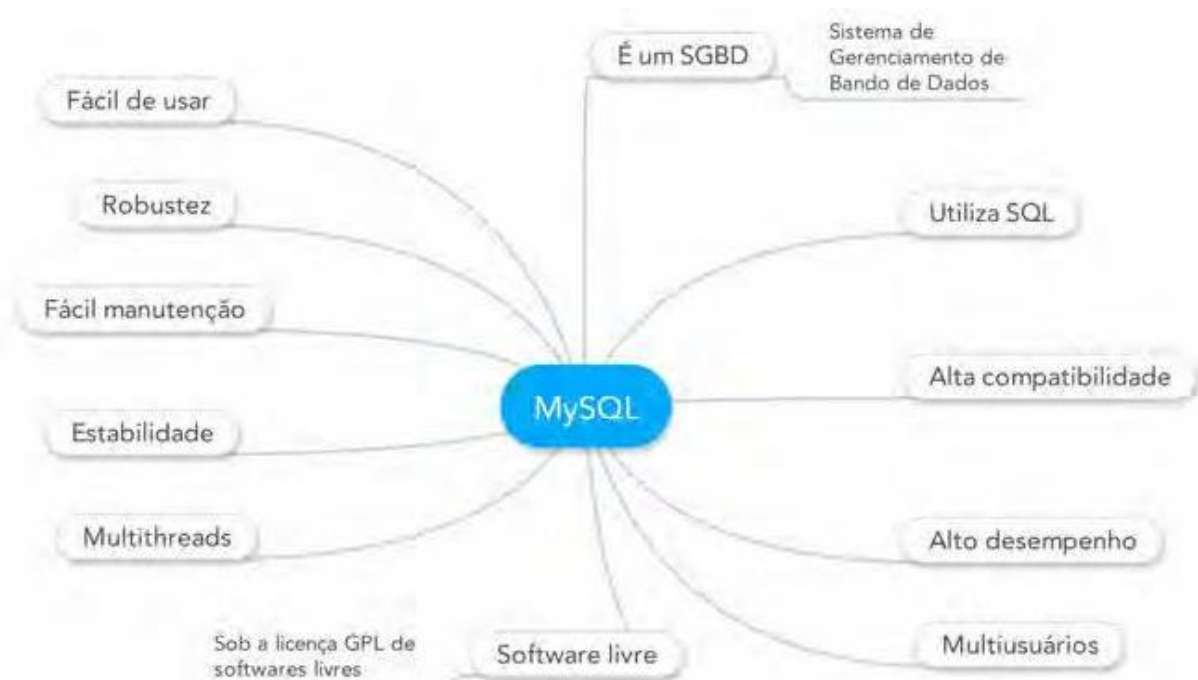


Figura 1: Características do MySQL

Fonte: MySQL – Comece com o principal banco de dados open source do mercado

Quando fala-se em banco de dados, remete-se a ideia de um local no qual é possível armazenar informações, para os mais variados fins, inclusão, consulta, modificação e

assim por diante. Basicamente, um banco de dados é constituído por tabelas e campos, linhas e colunas. Os campos são espaços reservados que armazenam um determinado dado, as linhas que podem ter um ou mais campos formam os registros, e as colunas são registros de um determinado campo de todos os registros. As tabelas então, são compostas por todos esses campos, linhas e colunas.

ID	NOME	DATA_NASCIMENTO	ID_ESTADO
1	DANIEL	2014-03-29	3
2	BRUNA	2004-06-04	1
3	VINICIUS S	2004-02-12	1
4	VITÓRIA	2004-12-17	5
5	THAIS	1987-08-27	1

Figura 2: Composição de uma tabela

Fonte: MySQL – Comece com o principal banco de dados open source do mercado

O MySQL possui características marcantes que o destacam frente a seus concorrentes, tais como: portabilidade com qualquer plataforma atual, compatibilidade com diversas linguagens de programação (Java, C/C++, Delphi, Visual Basic, Python, Perl, PHP, ASP e Ruby), excelente desempenho e estabilidade, pouco exigente quanto a requisitos de hardware, de fácil manuseio, possui uma interface gráfica de fácil utilização fornecida pela MySQL Inc, entre outras coisas. (WIKIPEDIA)

2.4. REST

A Representational State Transfer (REST), em português Transferência de Estado Representacional, é uma abstração da arquitetura da World Wide Web (*Web*), um estilo arquitetural que consiste de um conjunto coordenado de restrições arquiteturais aplicadas a componentes, conectores e elementos de dados dentro de um sistema de hipermídia distribuído. O REST ignora os detalhes da implementação de componente e

a sintaxe de protocolo com o objetivo de focar nos papéis dos componentes, nas restrições sobre sua interação com outros componentes e na sua interpretação de elementos de dados significantes.

O estilo arquitetural REST é aplicado no desenvolvimento de serviços web (web services), caracterizando assim um “RESTful”. Embora se refira a a princípios de arquitetura, na atualidade o mesmo tem sido utilizado amplamente interfaces web simples que utilizam XML, YAML ou JSON e HTTP.

É importante destacar que o REST oferece recursos (elementos de informação), que podem ser utilizados como um identificador global que podem manipular esses recursos, os componentes da rede se comunicam através de um padrão HTTP e trocam informações (arquivos ou ficheiros podem ser enviados ou recebidos). Quando um pedido é solicitado, nada mais será visto além do que foi solicitado, portanto, uma aplicação poderá interagir com o recurso conhecendo o identificador e a ação requerida. A aplicação deve compreender o formato da informação de volta, que pode ser um XML, YAML ou JSON, que pode ser qualquer conteúdo, inclusive uma imagem.

2.5. JSON

Em computação, JSON é um acrônimo para “JavaScript Object Notation”, é um formato de padrão aberto que utiliza texto legível a humanos para transmitir objetos de dados consistindo de pares atributo-valor. É o formato de dados mais comumente utilizado para comunicação assíncrona navegador/servidor, substituindo amplamente o XML. O JSON é um formato de dados independente de linguagem. Deriva do JavaScript, mas a partir de 2017 muitas linguagens de programação incluem código para gerar e analisar sintaticamente dados em formato JSON. O tipo de mídia da Internet oficial para o JSON é application/json. Nomes de arquivos JSON usam a extensão .json. (WIKIPEDIA)

A simplicidade do JSON o torna tão utilizado por programadores de todo o mundo, trata-se de uma formatação leve de dados, de fácil escrita e leitura.

JSON está constituído em duas estruturas: uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um *object*, record, struct, dicionário, hash table, keyed list, ou arrays associativas; uma lista ordenada de valores. Na maioria das

linguagens, isto é caracterizado como uma *array*, vetor, lista ou sequência. (www.json.org/json-pt.html)

Hoje, todas as linguagens de programação modernas, de uma forma ou de outra suportam estas estruturas.

2.6. HIBERNATE

O Hibernate é um framework para mapeamento objeto-relacional (*ORM – Object-Relational Mapping*) escrito em Java que facilita o mapeamento entre uma base tradicional de dados relacionais e o modelo objeto da aplicação. É um software livre de código aberto distribuído com a licença LGPL. O objetivo é minimizar a complexidade entre programas Java baseados no modelo orientado a objetos que precisam trabalhar com banco de dados relacionais, transformando, as classes Java para tabelas de dados (tipos de dados do Java para os da SQL), ele gera chamadas SQL e libera os desenvolvedores do trabalho manual de conversão de dados, mantendo o programa portátil para qualquer banco de dados SQL. O Hibernate gerencia as transações e a tecnologia de acesso à base de dados.

2.7. ECLIPSE

O Eclipse é uma IDE (*Integrated Development Environment*), que tem por objetivo otimizar o desenvolvimento, onde um grande numero de códigos são gerados em background, de forma rápida, ágil e leve. Embora seja uma IDE para desenvolvimento Java, suporta várias outras linguagens a partir de plug-ins como C/C++, PHP, ColdFusion, Python, Scala e plataforma Android.

O projeto Eclipse, criado pela IBM, distribuído como software livre sob licença EPL (Eclipse Public License), hoje é a IDE Java mais utilizada no mundo.

2.8. MAVEN

O Maven é uma ferramenta que gerencia projetos desde a sua criação. Ele mantém todas as informações do projeto em um único lugar, por meio do Project Object Model (POM), um arquivo de configuração do projeto onde são definidas todas as suas características,

como nome do projeto, desenvolvedores, repositórios de código fonte, dependências em bibliotecas externas. Trata-se de uma ferramenta que padroniza os projetos e diminui muito o tempo de desenvolvimento. Desenvolvido pela equipe do projeto Jakarta Turbine, que estavam procurando por uma maneira de deixar todos os projetos seguindo uma mesma estrutura ou um mesmo padrão, que queriam não ter mais que enviar os arquivos “.jar” das dependências para os sistemas de controle de versão, foi então que surgiu a primeira versão da ferramenta, que em 2005 foi completamente reescrita, resultando no Maven 2.

2.9. CDI

A especificação CDI (*Context and Dependency Injection*) dá enorme poder ao projeto, possibilitando para a aplicação recursos de injeção de dependência e ciclo de vida com escopos e contextos, de forma padronizada para o mundo Java e Java EE. Através do uso da sua implementação de referência, o Weld, os seus componentes podem injetá-los em qualquer classe. Integrar o CDI com outras especificações do Java EE, como a JPA e o JSF, além de desvendar os recursos avançados da especificação, permite ainda desacoplar o código e ter um design mais elegante para suas classes. Em aplicações web, muitas vezes os escopos de requisição são muito curtos e os de sessão muito longos, por isso, existe a necessidade de um escopo intermediário. Para a maioria dos desenvolvedores, o CDI foi a melhor adição ao Java EE à partir da versão 6. (CORDEIRO, 2014)

2.10. SPRING SECURITY

O Spring Security é um framework que trata de maneira extremamente robusta o controle de acesso, definindo de forma declarativa que terá acesso a que recursos da aplicação. Muitos tem a falsa impressão de que se trata de um framework de segurança, que torna a aplicação invulnerável, mas, isso não é verdade. O controle de acesso, com certeza, é uma parte fundamental, mas existem outros problemas que precisam ser tratados pelo desenvolvedor, tais como: ataques de negação de serviços (DOS: Denial of Service), entradas maliciosas, injeção de código entre outras.

O controle de acesso do Spring Security possui dois níveis: requisições que chegam a aplicação e invocações de métodos dos beans gerenciados pelo contexto do Spring. Esse

controle é feito por meio de autenticação e autorização. O primeiro processo, de autenticação, se dá quando o usuário se apresenta ao sistema informando sua identificação, por meio de login e senha ou até mesmo dados biométricos. Dada a autenticação, o framework analisa as informações e perfil do mesmo que são suas credenciais de acesso, desta forma definindo os limites de atuação do usuário. Após a autenticação e verificação das permissões dá-se a definição do segundo processo, o da autorização.

2.11. SPRING OAUTH2

Quando se aborda o tema de segurança de aplicações, principalmente em cenários onde API's REST estão contidas, algumas questões surgem com frequência:

- Como identificar as permissões de manipulações?
- As informações, de que forma serão codificadas ou decodificadas?
- Como definir as restrições de acesso?
- Quem se responsabiliza por guardar e fornecer dados de segurança?
- Como saber se uma requisição passou ou não por modificações?

É fundamental para o desenvolvedor, saber que assim como aplicações web, também as REST são vulneráveis. Existem vários mecanismos de proteção de serviços, mas bastante limitados. Já que a arquitetura REST não define procedimentos de segurança, porque temos apenas métodos HTTP como GET (para buscar), POST (para incluir), PUT (para atualizar), DELETE (para eliminar), desta forma o protocolo OAUTH permite que aplicações acessem e compartilhem serviços com segurança e confiabilidade.

O OAUTH é utilizado por Google, facebook, NETFLIX, Linkedin, github, Instagram, twitter e muitas outras aplicações. Atualmente, a versão utilizada é a 2.0, segura, simples e padronizada.

O OAUTH2 não utiliza "username" ou "password", mas sim "tokens", estabelece confiança entre recursos (servidor de identidade e cliente da aplicação), é fortemente integrado com TLS/SSL. Analisando o protocolo de segurança, pode-se chegar às seguintes conclusões:

- Prós:
 - Ótima integração de aplicações aos serviços de web sites;
 - Acesso permitido por escopo ou tempo limitado;

- Independência de compartilhamento de senhas com aplicações terceiras.
- Contras:
 - Pode exigir determinada complexidade de implementação (dependendo do recurso desejado);
 - Problemas de interoperabilidade;
 - Más implementações expõem falhas de segurança.

3. DOCUMENTAÇÃO

É fundamental no desenvolvimento de software que o mesmo tenha uma documentação detalhada, de simples entendimento e atualizada. Atendendo a esta máxima, os subtópicos a seguir apresentarão todas as informações referentes ao sistema SUPERCIDADE.

3.1. ESTRUTURA ANALÍTICA DO PROJETO

A estrutura analítica de projeto (EAP), do inglês *work breakdown structure* (WBS), em gerência de projetos, é uma estrutura de subdivisão das entregas e dos componentes do projeto em partes menores e de fácil gerenciamento. É uma estrutura em modelo de árvore, orientada aos prazos de entregas, ciclos e subprojetos. O EAP serve de base para o planejamento do projeto, é um processo bastante comum. A seguir, a estrutura analítica do projeto SUPERCIDADE:

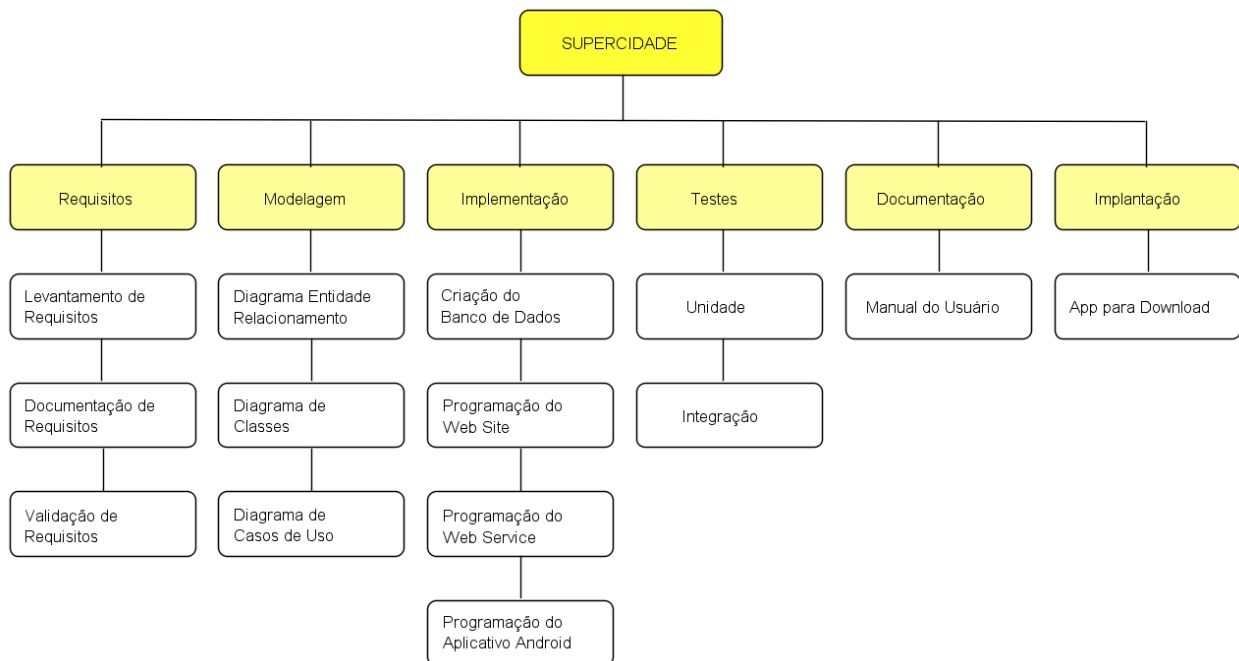


Figura 3: Estrutura Analítica do Projeto SUPERCIDADE

3.2. CRONOGRAMA

Dentro de um plano consistente de metas estratégicas para desenvolvimento de um sistema temos o cronograma de atividade, e tratando de softwares arrojados e complexos, cada vez mais se faz necessário muita organização. O cronograma é a representação do plano de projeto, com os recursos e o tempo de execução.

O não cumprimento de prazos, pode acarretar em atraso no lançamento ou instalação de um software, perdas de clientes e principalmente perdas de receitas.

Um bom projeto nasce do planejamento estratégico de uma organização ou de determinada necessidade. Se feito de forma correta e coerente, organizará as fases do projeto (início, planejamento, execução, controle e encerramento) ao decorrer do seu ciclo. A seguir, o cronograma do projeto SUPERCIDADE:

CRONOGRAMA	Nov		Dez		Jan		Fev		Mar		Abr		Mai		Jun		Jul		Ago		
	1º Q	2º Q	1º Q	2º Q	1º Q	2º Q	1º Q	2º Q	1º Q	2º Q	1º Q	2º Q	1º Q	2º Q	1º Q	2º Q	1º Q	2º Q	1º Q	2º Q	
Levantamento de Material	■	■																			
Pesquisa de Trabalhos Correlatos			■	■																	
Desenvolvimento Pré-Projeto					■	■															
Levantamento de Requisitos						■															
Mapa Mental							■														
Diagrama de Classes								■													
Diagrama Entidades Relacionamentos									■												
Abstração dos Casos de Uso										■	■										
Especificação dos Casos de Uso											■	■									
Entrega do TCC para Qualificação												■									
Criação do Banco de Dados													■	■	■						
Programação do Web Site														■	■	■					
Programação do Web Service																■					
Programação do App Android																	■	■	■		
Testes																					■
Envio/Retorno do TCC ao Orientador																					■
Apresentação Final																					■

Tabela 1: Cronograma de Desenvolvimento do Sistema SUPERCIDADE

3.3. DIAGRAMA DE CLASSE

O diagrama de classe é a representação gráfica do modelo de dados do sistema, com alto nível de abstração. Faz se importante, para entender a importância do DER, conhecer o que as expressões ENTIDADE, ATRIBUTO e RELACIONAMENTO significam. A “entidade” pode ser entendida como objetos do mundo real (pessoas, móveis, imóveis, veículos, etc..), os “atributos” são os dados, ou porque não entendermos como os campos referentes da entidade, e, o “relacionamento” são as ligações entre duas ou mais entidades, que demonstram como estas interagem e também, como seus atributos podem contribuir nesta interação. A seguir, o diagrama entidade relacionamento do SUPERCIDADE:

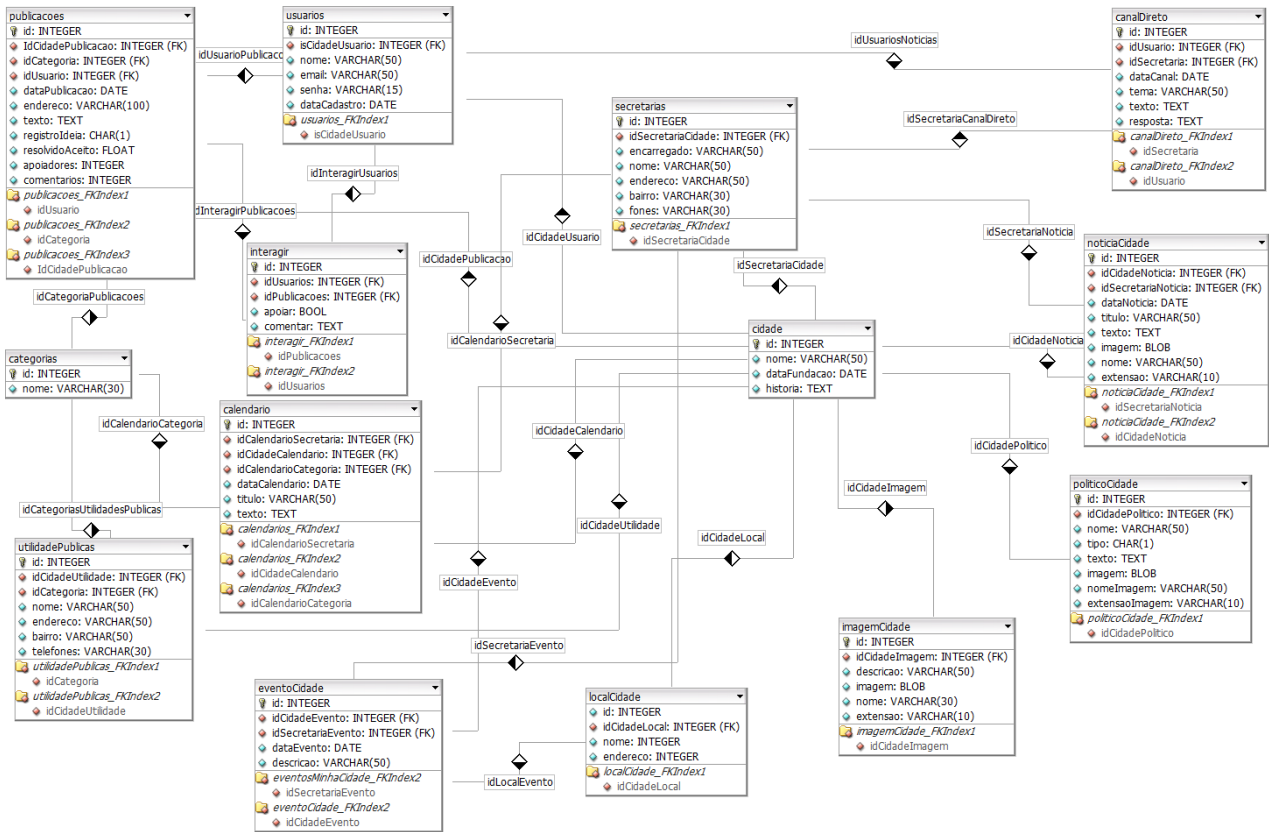


Figura 4: Diagrama Entidade Relacionamento do Sistema SUPERCIDADE

3.4. DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso documenta o que o sistema faz e as interações que o mesmo tem com o usuário. Este diagrama não se aprofunda em termos técnicos sobre as funcionalidades do sistema. É composto basicamente de quatro partes: cenário

(sequência de eventos), ator (usuários do sistema), caso de uso (funcionalidade) e comunicação (ligação entre ator e caso de uso). A seguir, o diagrama de casos de uso do SUPERCIDADE:

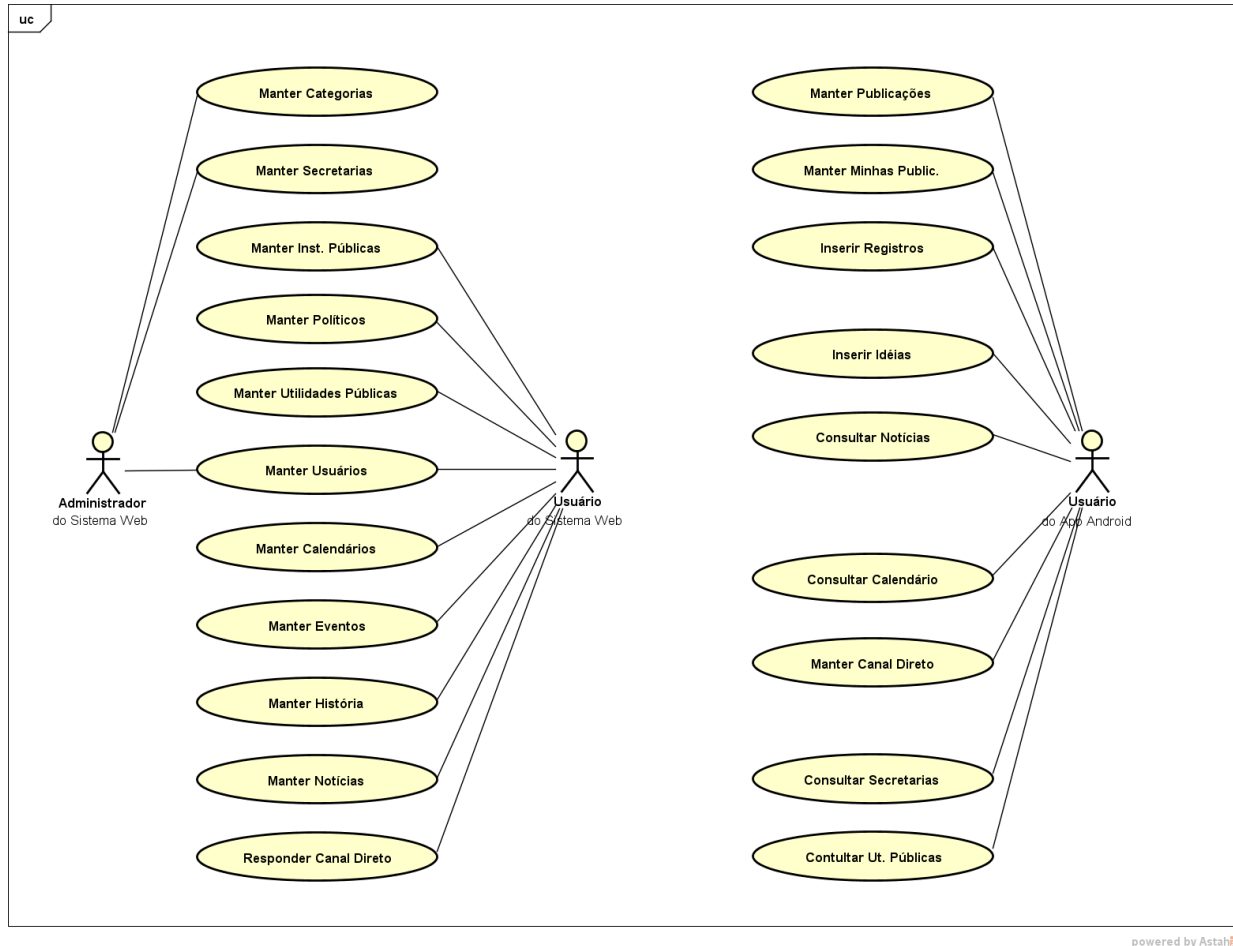


Figura 5: Casos de Uso do Sistema SUPERCIDADE

3.5. ESPECIFICAÇÃO DOS CASOS DE USO

A especificação dos casos de uso tem por finalidade demonstrar o comportamento do sistema (todo ou parte dele), seguindo alguns critérios, tais como: identificação, apresentação dos atores, pré-condições, fluxos e pós-condições. Cada descrição de caso de uso deve conter em alto nível o detalhamento de um ou mais requisitos funcionais do software. Esta especificação deve mostrar a interação entre o ator e o sistema sobre uma visão externa.

A identificação do caso de uso deve ter um nome que identifique a funcionalidade abordada e o nome dado na especificação deve ser o mesmo dado no diagrama. O ator representa a entidade externa que interage com o sistema, e pode ser um ou vários, como geralmente acontece. A pré-condição tem que ser testada antes de se iniciar o caso de uso, e a mesma tem de ser verdadeira, caso contrário, o acesso ao recurso é negado

e, deve ser agregada quando houver um pré-requisito significativo para a funcionalidade. O fluxo de eventos são as ações dos possíveis cenários do caso de uso e, geralmente, são acionados por estímulo de um ator ou por verificação de atributos da funcionalidade. A pós-condição realizará uma ação sempre que a condição dos fluxos realizados é verdadeira, é opcional, e deve ser utilizada quando for agregar valor ao caso de uso.

3.5.1. Manter Categorias

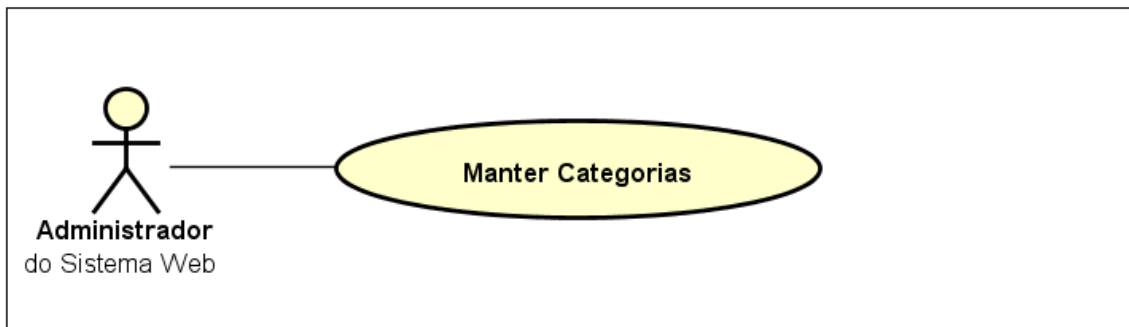


Figura 6: Use Case Manter Categorias

Finalidade/Objetivo	Permite atualizar o cadastro de categorias, que auxiliará na organização das publicações e utilidades públicas.
Ator(es)	Administrador do Sistema Web
Pré-condições	O Administrador do Sistema Web precisa estar autenticado
Fluxo Principal	A rotina apresenta uma listagem contendo as categorias já cadastradas. A partir daí o administrador poderá escolher um dos botões: <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 2: Especificação da Use Case Manter Categorias

3.5.2. Manter Secretarias

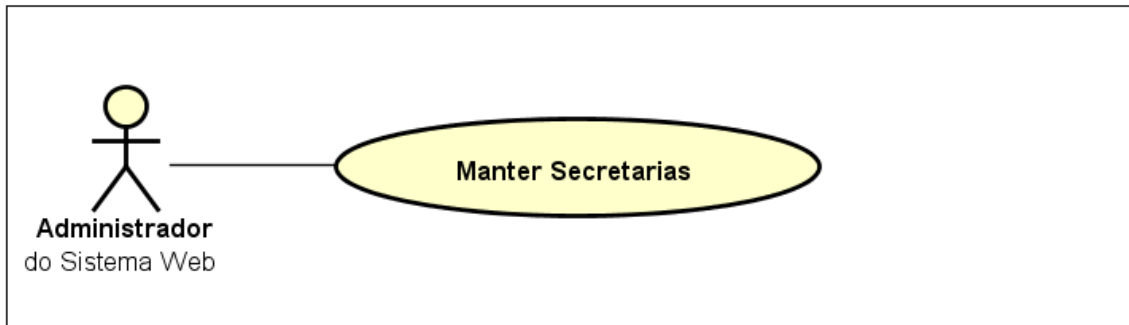


Figura 7: Use Case Manter Secretarias

Finalidade/Objetivo	Permite atualizar o cadastro de Secretarias, que auxiliará na organização das notícias e eventos municipais, bem como contribuirá para distinguir as mensagens na rotina de canal direto de comunicação.
Ator(es)	Administrador do Sistema Web
Pré-condições	O Administrador do Sistema Web precisa estar autenticado
Fluxo Principal	<p>A rotina apresenta uma listagem contendo as secretarias já cadastradas. A partir daí o administrador poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 3: Especificação da Use Case Manter Secretarias

3.5.3. Manter Instalações Públicas

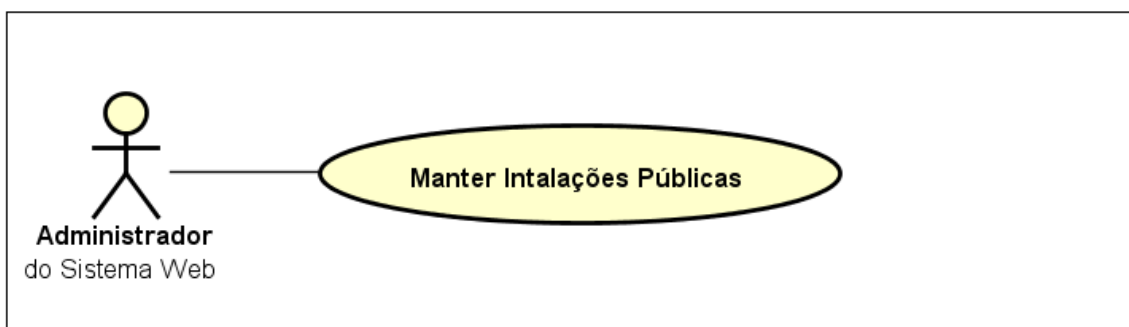


Figura 8: Use Case Manter Instalações Públicas

Finalidade/Objetivo	Permite organizar o cadastro de Instalações Públicas do município, o que contribuirá para organização dos eventos locais.
Ator(es)	Administrador do Sistema Web
Pré-condições	O Administrador do Sistema Web precisa estar autenticado
Fluxo Principal	A rotina apresenta uma listagem contendo as instalações públicas já cadastradas. A partir daí o administrador poderá escolher um dos botões: <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 4: Especificação da Use Case Manter Instalações Públicas

3.5.4. Manter Políticos

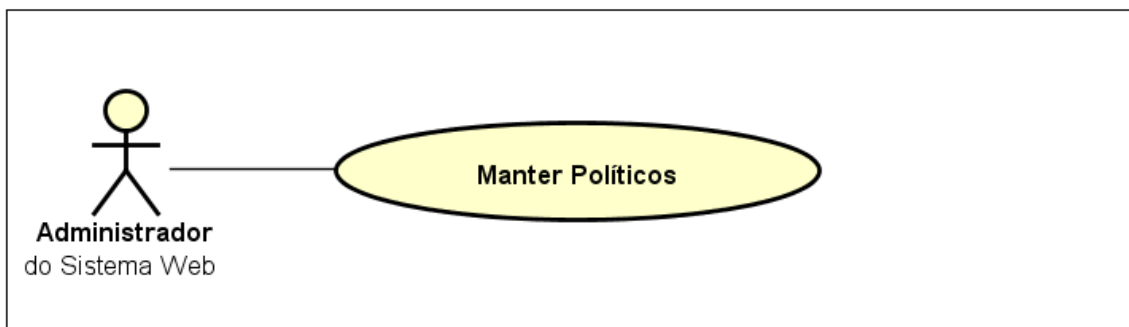


Figura 9: Use Case Manter Políticos

Finalidade/Objetivo	Permite organizar o cadastro de Políticos da cidade, o que contribuirá para as informações do município.
Ator(es)	Administrador do Sistema Web
Pré-condições	O Administrador do Sistema Web precisa estar autenticado
Fluxo Principal	A rotina apresenta uma listagem contendo Políticos já cadastrados. A partir daí o administrador poderá escolher um dos botões: <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 5: Especificação da Use Case Manter Políticos

3.5.5. Manter Utilidades Públicas

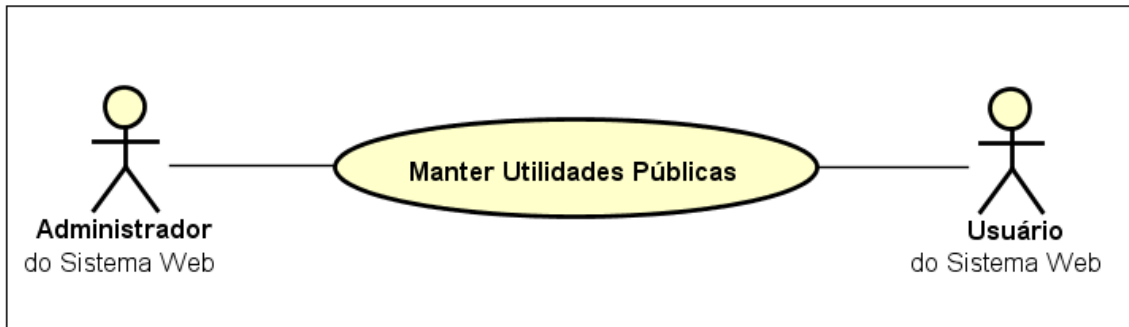


Figura 10: Use Case Manter Utilidades Públicas

Finalidade/Objetivo	Permite organizar o cadastro de Utilidades Públicas do município, órgãos públicos, empresas de saneamento, energia elétrica, secretarias e muito mais, que contribuirá na consulta de informações da cidade.
Ator(es)	Administrador e Usuário do Sistema Web
Pré-condições	O Administrador e o Usuário do Sistema Web precisam estar autenticados
Fluxo Principal	<p>A rotina apresenta uma listagem contendo as utilidades públicas já cadastradas. A partir daí o administrador ou o usuário poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 6: Especificação da Use Case Utilidades Públicas

3.5.6. Manter Usuários

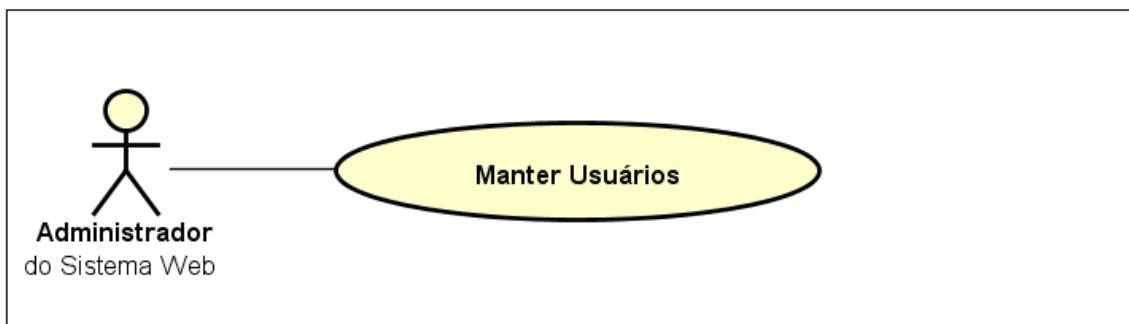


Figura 11: Use Case Manter Usuários

Finalidade/Objetivo	Permite organizar o cadastro de Usuários, o que permitirá o acesso de usuários para
---------------------	---

	as funcionalidades do Sistema Web.
Ator(es)	Administrador do Sistema Web
Pré-condições	O Administrador do Sistema Web precisa estar autenticado
Fluxo Principal	<p>A rotina apresenta uma listagem contendo os usuários já cadastrados. A partir daí o administrador poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 7: Especificação da Use Case Manter Usuários

3.5.7. Manter Calendários

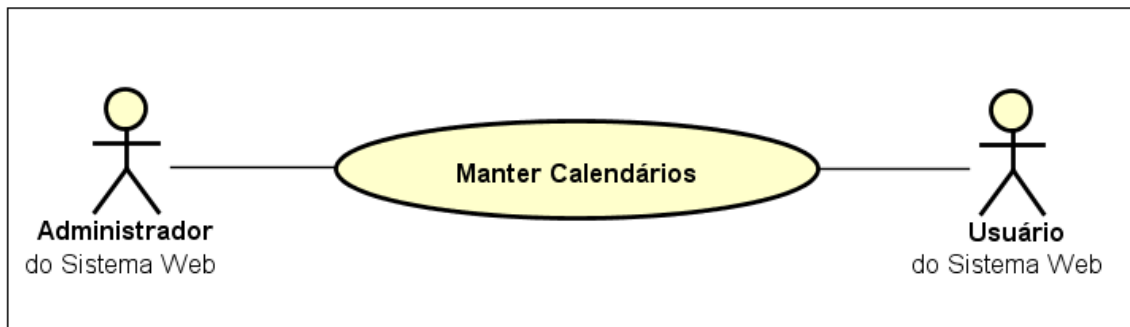


Figura 12: Use Case Manter Calendários

Finalidade/Objetivo	Permite organizar o cadastro do Calendário municipal, o que contribuirá para as informações do município.
Ator(es)	Administrador e Usuário do Sistema Web
Pré-condições	O Administrador e o Usuário do Sistema Web precisam estar autenticados
Fluxo Principal	<p>A rotina apresenta uma listagem contendo os compromissos municipais já cadastrados. A partir daí o administrador ou o usuário poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 8: Especificação da Use Case Manter Calendários

3.5.8. Manter Eventos

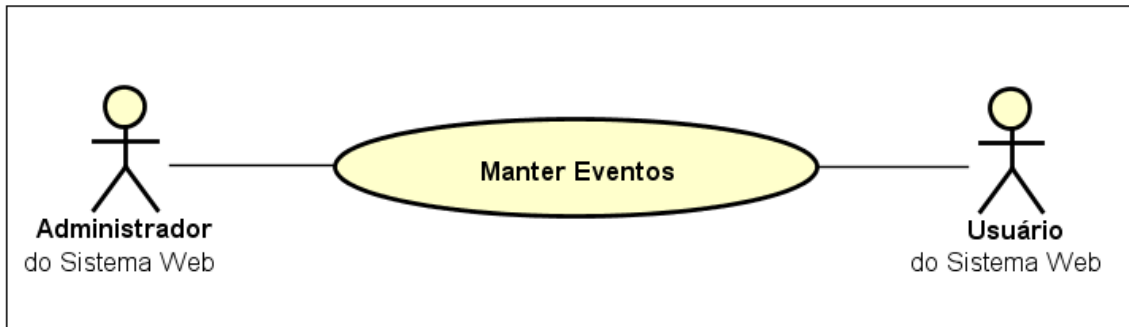


Figura 13: Use Case Manter Eventos

Finalidade/Objetivo	Permite organizar o cadastro de Eventos da cidade, tais como feiras, exposições e shows entre muitos outros, o que contribuíra para as informações do município.
Ator(es)	Administrador e Usuário do Sistema Web
Pré-condições	O Administrador e o Usuário do Sistema Web precisam estar autenticados
Fluxo Principal	<p>A rotina apresenta uma listagem contendo os eventos já cadastrados. A partir daí o administrador ou o usuário poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 9: Especificação da Use Case Manter Eventos

3.5.9. Manter História

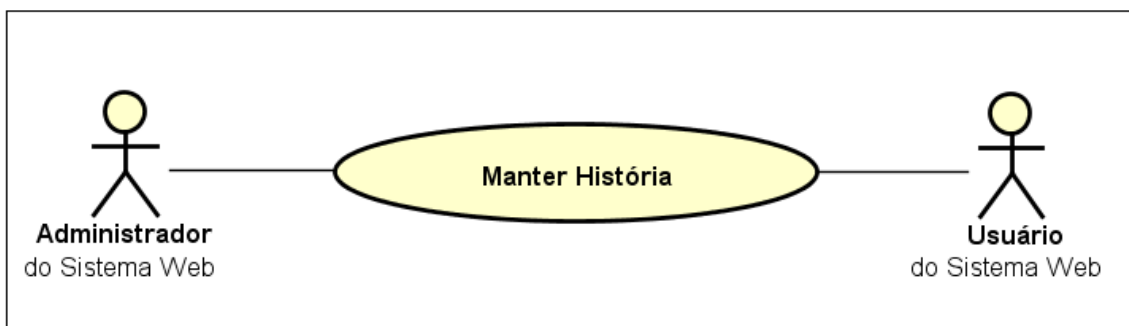


Figura 14: Use Case Manter História

Finalidade/Objetivo	Permite organizar o cadastro da História da cidade, o que contribuíra para as informações do município.
---------------------	---

Ator(es)	Administrador e Usuário do Sistema Web
Pré-condições	O Administrador e o Usuário do Sistema Web precisam estar autenticados
Fluxo Principal	<p>A rotina apresenta um espaço de texto para guardar a história da cidade. A partir daí o administrador ou o usuário poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Alterar: carrega os dados de um registro existente e permite sua atualização; • Fechar: encerra a rotina e volta à tela principal.

Tabela 10: Especificação da Use Case Manter História

3.5.10. Manter Notícias

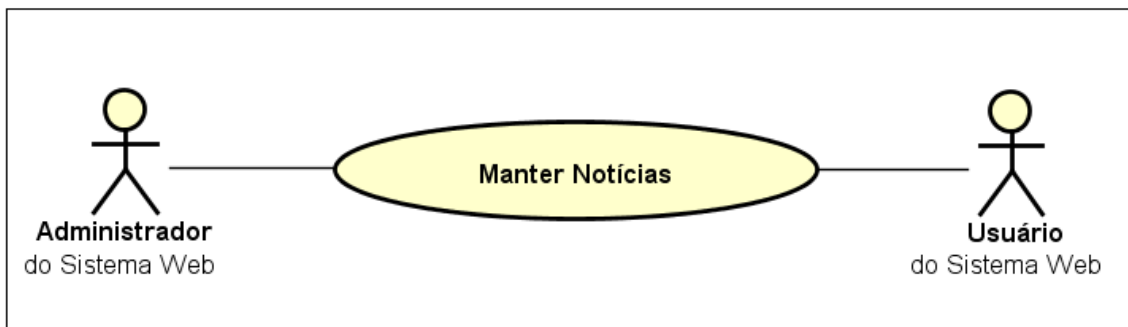


Figura 15: Use Case Manter Notícias

Finalidade/Objetivo	Permite organizar o cadastro de Notícias da administração municipal, o que pode conter comunicados, ofícios e leis municipais dentre outros, o contribuirá para as informações do município.
Ator(es)	Administrador e Usuário do Sistema Web
Pré-condições	O Administrador e o Usuário do Sistema Web precisam estar autenticados
Fluxo Principal	<p>A rotina apresenta uma listagem contendo as notícias da administração municipal já cadastradas. A partir daí o administrador ou o usuário poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Incluir: adiciona um novo registro; • Alterar: carrega os dados de um registro existente e permite sua atualização; • Excluir: exclui um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 11: Especificação da Use Case Manter Notícias

3.5.11. Responder Canal Direto

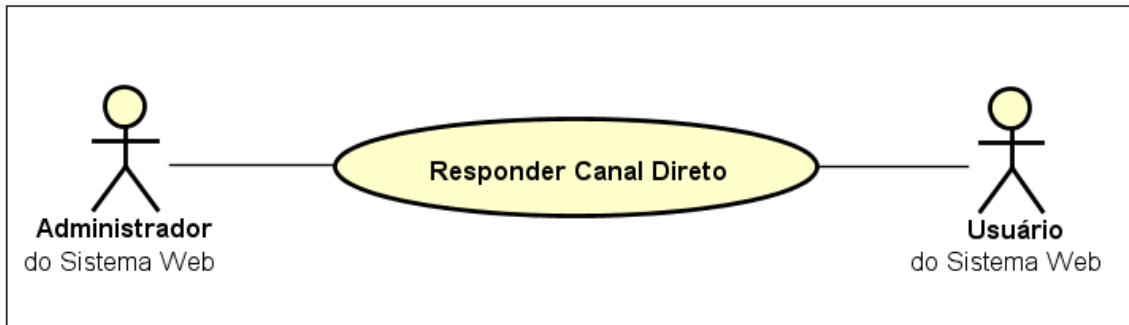


Figura 16: Use Case Responder Canal Direto

Finalidade/Objetivo	Permite atualizar o cadastro do Canal Direto da cidade, que contém as mensagens enviadas pelos usuários do App e os secretários por sua vez podem responder o questionado, o que contribuíra para as informações do município.
Ator(es)	Administrador e Usuário do Sistema Web
Pré-condições	O Administrador e o Usuário do Sistema Web precisam estar autenticados
Fluxo Principal	A rotina apresenta uma listagem contendo as mensagens dos usuários do app já cadastradas. A partir daí o administrador ou o usuário poderá escolher um dos botões: <ul style="list-style-type: none"> • Responder: responde um registro existente; • Fechar: encerra a rotina e volta à tela principal.

Tabela 12: Especificação da Use Case Manter Canal Direto

3.5.12. Manter Publicações

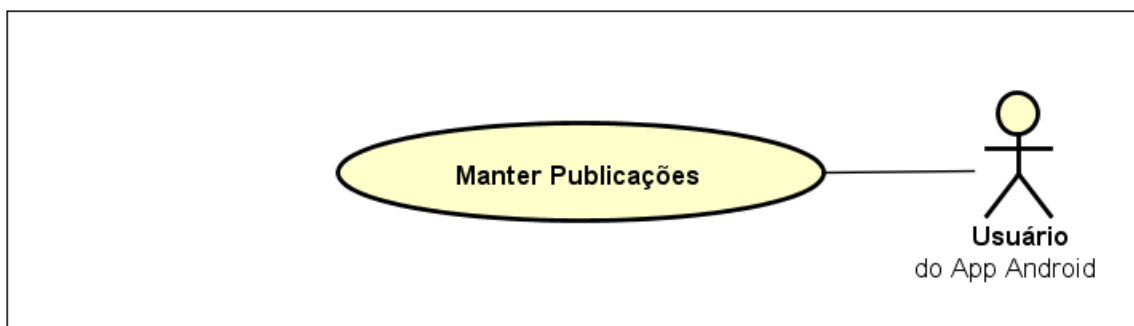


Figura 17: Use Case Manter Publicações

Finalidade/Objetivo	Permite organizar o arquivo de Publicações de imagens da cidade, onde o usuário do app poderá adicionar apoio ou comentários, inclusive compartilhar com outros aplicativos, o que contribuirá para as informações do município e a interação da rede social.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	<p>A rotina apresenta as publicações postadas na rede social em ordem de data descendente já cadastradas. A partir daí o usuário do app poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • ☰ (sub-menu): abre um submenu com as opções: <ul style="list-style-type: none"> • Editar: permite ao usuário do app modificar as publicações; • Excluir: permite ao usuário do app excluir suas publicações. • Apoiar: adiciona ou elimina apoio à publicação; • Comentar: adiciona, altera ou elimina um comentário à publicação; • Compartilhar: compartilha a publicação atual com outro aplicativo.

Tabela 13: Especificação da Use Case Manter Publicações

3.5.13. Manter Minhas Publicações

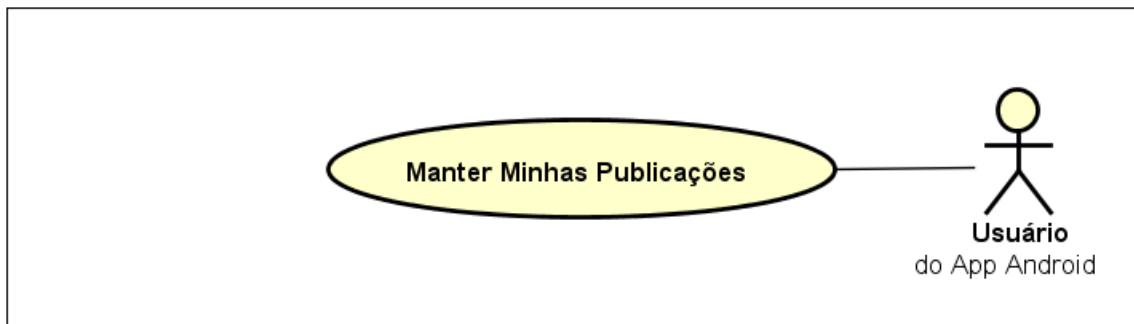


Figura 18: Use Case Manter Minhas Publicações

Finalidade/Objetivo	Permite organizar arquivo de Publicações da cidade, onde o usuário do App poderá atualizar ou eliminar apenas as publicações feitas por ele, o que contribuirá para as informações do município e a interação da rede social.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	<p>A rotina apresenta as publicações postadas pelo usuário do app na rede social em ordem de data descendente já cadastradas. A partir daí o usuário do app poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • ☰ (sub-menu): abre um submenu com as opções: <ul style="list-style-type: none"> • Editar: permite ao usuário do app modificar as publicações; • Excluir: permite ao usuário do app excluir suas publicações. • Apoiar: adiciona ou elimina apoio à publicação;

	<ul style="list-style-type: none"> • Comentar: adiciona, altera ou elimina um comentário à publicação; • Compartilhar: compartilha a publicação atual com outro aplicativo.
--	---

Tabela 14: Especificação da Use Case Manter Minhas Publicações

3.5.14. Inserir Registros

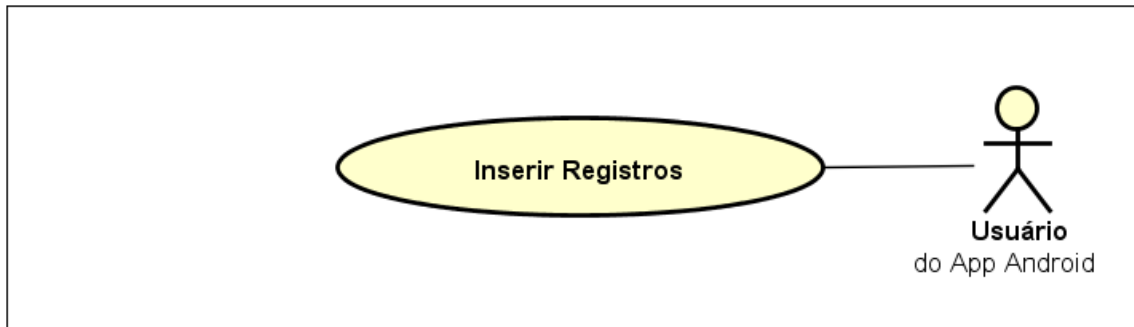


Figura 19: Use Case Inserir Registros

Finalidade/Objetivo	Permite adicionar imagens ao arquivo de Publicações da cidade, com imagens capturadas pelo dispositivo móvel ou imagens armazenadas na memória do mesmo, o que contribuíra para as informações do município e a interação da rede social.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	<p>A rotina permite adicionar novas publicações (de qualquer espécie) à rede social. A partir daí o usuário do app poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Capturar Foto: adiciona um novo registro (imagem) à partir da câmera digital do dispositivo móvel; • Carregar Foto: adiciona um novo registro (imagem) à partir da galeria de imagens do dispositivo móvel.

Tabela 15: Especificação da Use Case Inserir Registros

3.5.15. Inserir Idéias

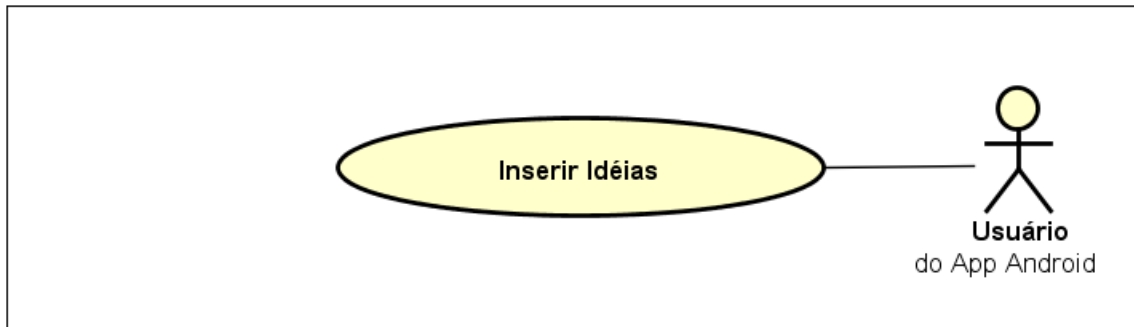


Figura 20: Use Case Inserir Idéias

Finalidade/Objetivo	Permite adicionar imagens ao arquivo de Publicações da cidade, com o intuito de apresentar boas idéias a espaços inutilizados, o que contribuirá para as informações do município.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	<p>A rotina permite adicionar novas publicações (de natureza sugestiva) à rede social. A partir daí o usuário do app poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Capturar Foto: adiciona um novo registro (imagem) à partir da câmera digital do dispositivo móvel; • Carregar Foto: adiciona um novo registro (imagem) à partir da galeria de imagens do dispositivo móvel.

Tabela 16: Especificação da Use Case Inserir Idéias

3.5.16. Consultar Notícias

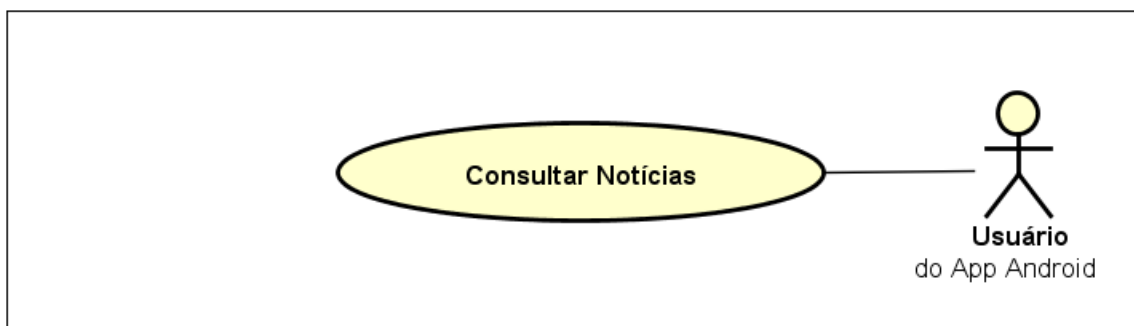


Figura 21: Use Case Consultar Notícias

Finalidade/Objetivo	Permite ao usuário do app consultar notícias publicadas pela administração
---------------------	--

	municipal, o que contribuíra para as informações do município.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	<p>A rotina apresenta as notícias cadastradas pela Administração municipal. Mostra o tema e as primeiras linhas do texto da publicação. A partir daí o usuário do app poderá escolher um dos botões:</p> <ul style="list-style-type: none"> Expandir: mostra o texto completo da notícia desejada;

Tabela 17: Especificação da Use Case Consultar Notícias

3.5.17. Consultar Calendário

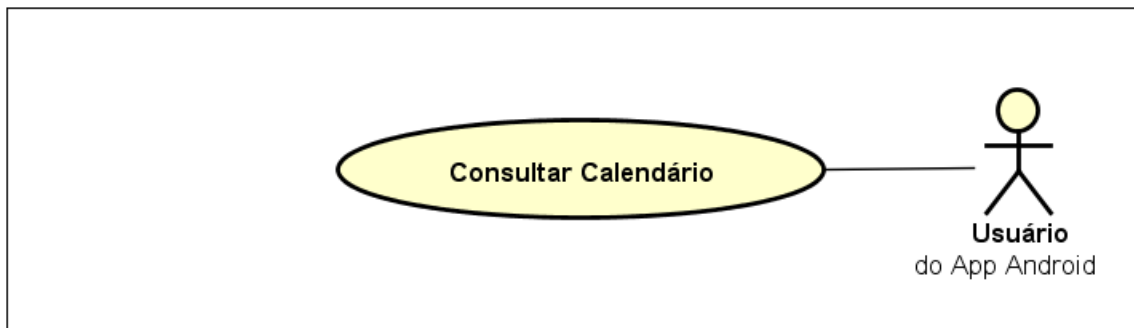


Figura 22: Use Case Consultar Calendário

Finalidade/Objetivo	Permite ao usuário do app consultar o calendário publicado pela administração municipal, o que contribuíra para as informações do município.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	A rotina apresenta o calendário do mês atual primeiramente, onde o usuário do app pode acompanhar todos os compromissos municipais e deslizar a tela para visualizar os demais meses e seus respectivos detalhes.

Tabela 18: Especificação da Use Case Consultar Calendário

3.5.18. Manter Canal Direto

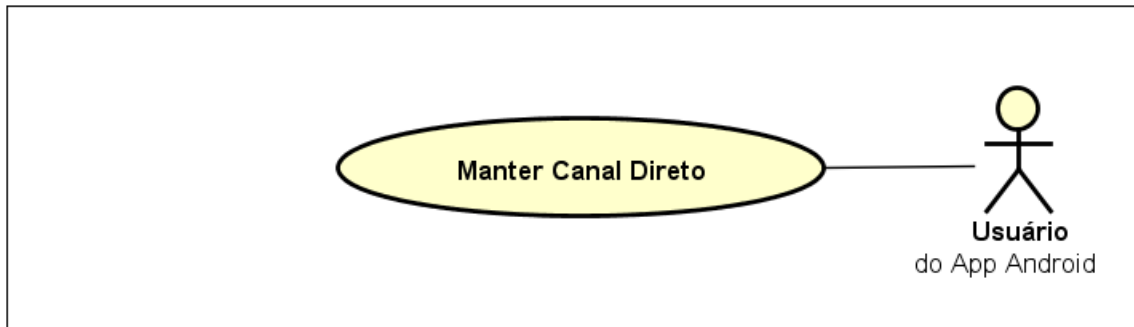


Figura 23: Use Case Manter Canal Direto

Finalidade/Objetivo	Permite ao usuário do app organizar as mensagens enviadas para as secretarias pelo mesmo, o que contribuíra para transparência das atividades.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	<p>A rotina enviar e receber mensagens do usuário do app e as secretárias do município. A partir daí o usuário do app poderá escolher um dos botões:</p> <ul style="list-style-type: none"> • Adicionar: adiciona um nova mensagem à secretaria desejada; • ⋮ (sub-menu): apresenta um submenu com as opções: <ul style="list-style-type: none"> • Editar: permite ao usuário do app alterar mensagens; • Excluir: permite ao usuário do app eliminar mensagens.

Tabela 19: Especificação da Use Case Manter Canal Direto

3.5.19. Consultar Secretarias

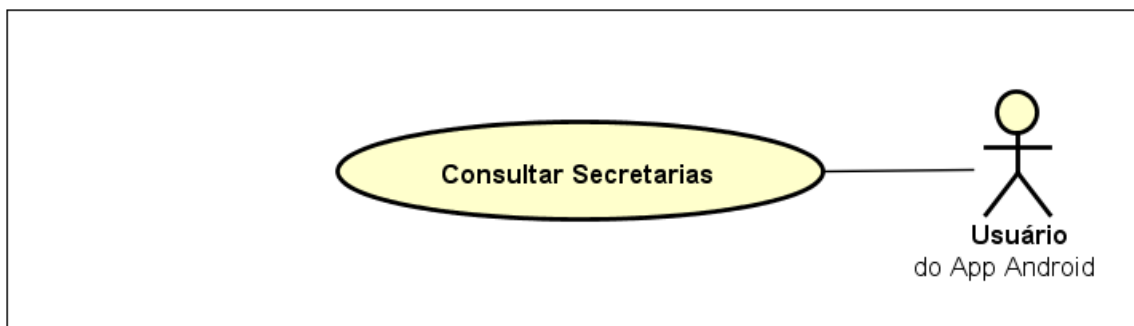


Figura 24: Use Case Consultar Secretarias

Finalidade/Objetivo	Permite ao usuário do app consultar o telefone, endereço e encarregado das
---------------------	--

	secretarias municipais, o que contribuíra para a melhor comunicação e localização.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	Esta rotina apresenta as secretarias municipais e o usuário do app pode procurar a secretaria desejada deslizando o dedo sobre a tela. Ela não apresenta nenhum tipo de interação, é meramente para consulta.

Tabela 20: Especificação da Use Case Consultar Secretarias

3.5.20. Consultar Utilidades Públicas

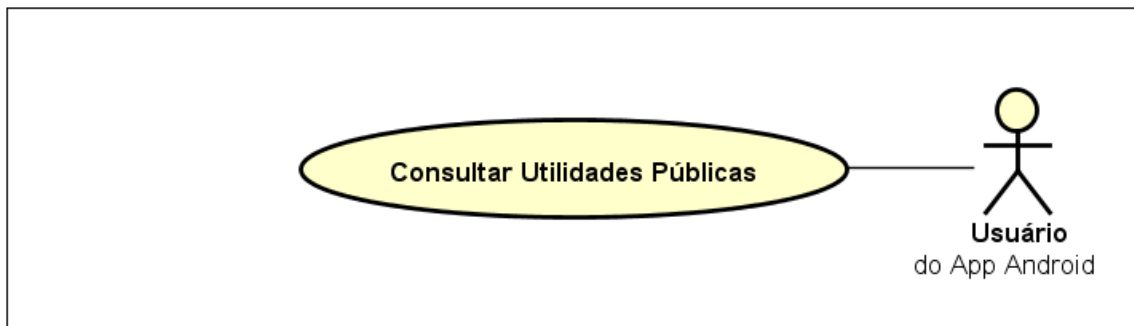


Figura 25: Use Case Consultar Utilidades Públicas

Finalidade/Objetivo	Permite ao usuário do app consultar principais órgãos, instituições e empresas de utilidades públicas do município, o que contribuíra para as informações do município.
Ator(es)	Usuário do App Android
Pré-condições	O Usuário do App Android precisa estar autenticado
Fluxo Principal	Esta rotina apresenta as principais utilidades públicas municipais e o usuário do app pode procurar a informação desejada deslizando o dedo sobre a tela. Ela não apresenta nenhum tipo de interação, é meramente para consulta.

Tabela 21: Especificação da Use Case Consultar Utilidades Públicas

4. CONCLUSÃO

O SUPERCIDADE foi projetado pra servir como uma poderosa ferramenta de comunicação e desenvolvimento municipal. Os cidadãos poderão acompanhar de perto tudo o que acontece na cidade em todas as áreas. Fará de cada usuário um fiscal, que contará com o apoio de outros membros do aplicativo. Outra possibilidade é que as pessoas mostrem o que a cidade possui de bom e belo e publicar também o que o seu povo tem feito de interessante. Que o software seja usado com bom senso e sirva de trampolim para o crescimento da sociedade.

É importantíssimo justificar que as tecnologias envolvidas no projeto, levaram em consideração o que existe atualmente de estável e confiável. A linguagem de programação, os frameworks e o banco de dados, são com certeza, o que existe de melhor e principalmente, recursos sob licença de software livre, ou seja, livre de qualquer impedimento de utilização e comercialização.

Pessoalmente, concluo que a construção do projeto foi de grande ajuda para o conhecimento de novas tecnologias e fundamental principalmente na questão organizacional no desenvolvimento de softwares.

5. REFERÊNCIAS

- CARVALHO, Vinícius: MySQL – Comece com o principal banco de dados open source do mercado. Casa do Código, 2015, 154 p
- GEARY, D; HORSTMANN, C. Core: JavaServer Faces. Third Edition Redwood Shores: Prentice Hall, 2010. 636 p.
- MASSOL, Vincent; O'BRIEN, Timothy: Maven: A Developer's Notebook. Editora O'Reilly, 2005, 224p.
- CORDEIRO, Gilliard: CDI: Integre as dependências e contextos do seu código Java. Editora Casa do Código (Digital), 2014, 238p.
- MONTEIRO, Vitor: ANDROID – Desenvolvimento de Aplicações com Android Studio. FCA Editora, 2016, 304p.
- SANTOS, Rui Rossi dos: Programação de Computadores em Java. Editora NovaTerra, 2ª Edição, 2014, 1456p.
- FARIA, Thiago: Java EE 7 Com JSF, PrimeFaces e CDI. eBook AlgaWorks Softwares, Treinamentos e Serviços, 2013, 197p.
- BAUER, Christian; KING Gavin: Java Persistence com Hibernate. Editora Ciência Moderna, 2007, 872p.
- SAULETE, Alexandre: REST Construa API'S Inteligentes de maneira simples. Editora Casa do Código, 2013, 315p.
- WEISSMANN, Henrique Lobo: Vire o Jogo com Spring Framework. Editora Casa do Código, 2012, 329p.